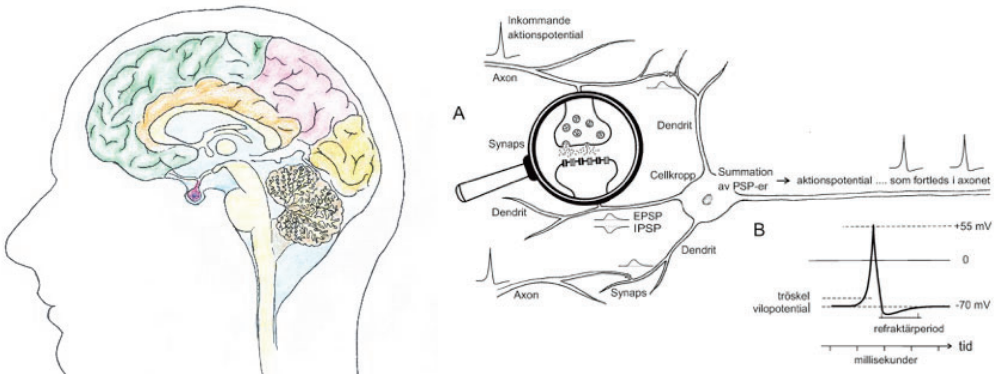
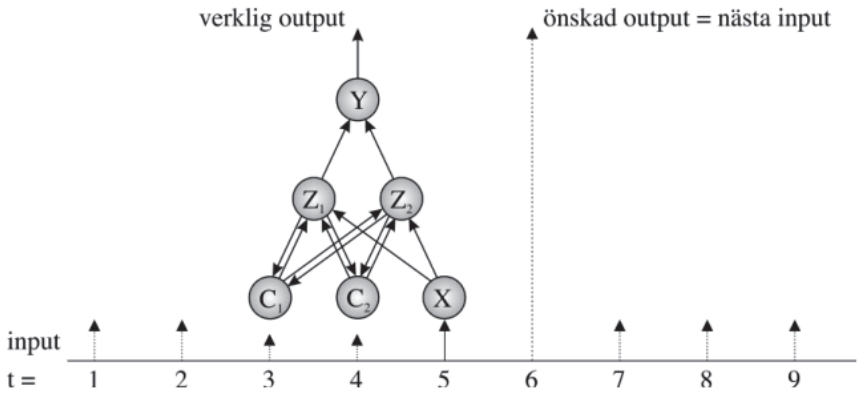
Inlärning och minne i neurala nätverk





Helge Malmgren

Andra (preliminära) upplagan

Helge Malmgren

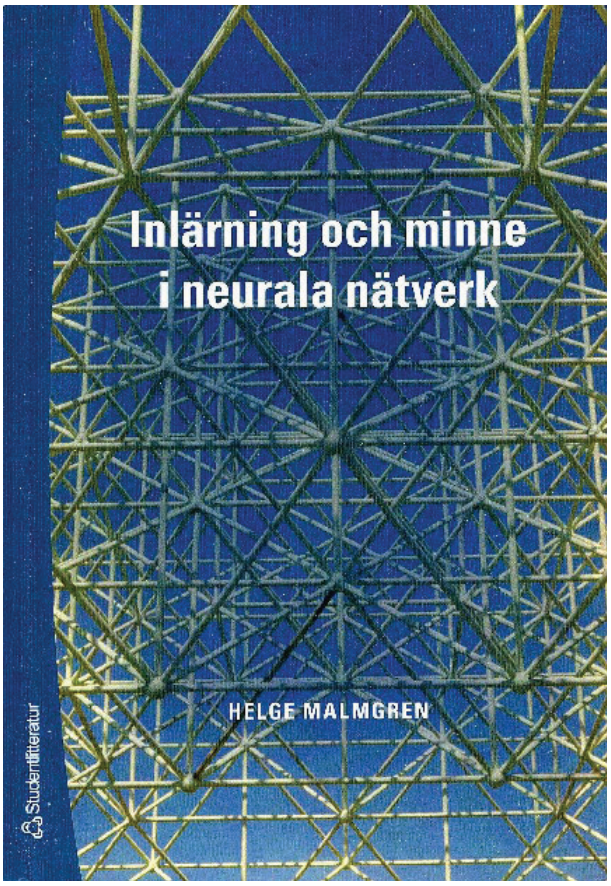
Inlärning och minne i neurala nätverk

Andra (preliminära) upplagan

Göteborg 2020
ISBN 978-91-519-5170-6

Innehållsförteckning

Förord till andra (preliminära) upplagan



Inlärning och minne i neurala nätverk kom ut 2007 då intresset för artificiella neurala nätverk (ANN) var i en djup vågdal, både nationellt och internationellt. Detta bidrog säkert till att boken inte fick någon större spridning och fortfarande är praktiskt taget okänd bland tekniker, matematiker och neurovetare. Efter 2010 har intresset för ANN exploderat, främst tack vare upptäckter och uppfinningar som kan sammanfattas under begreppet *deep learning*. Fortfarande finns ingen lättillgänglig och samtidigt fyllig introduktion till ämnet på svenska, och det är därför läge att ge ut en ny upplaga av den gamla boken. Här presenteras en preliminär andra upplaga, där texten inte ändrats men försetts med detta nya förord.

Några ord om bokens karaktär är på plats. Den kom till under en lång följd av år då författaren undervisade i ämnet vid Filosofiska Institutionen (senare FLOV: Institutionen för Filosofi, Lingvistik och Vetenskapsteori), Göteborgs Universitet. Detta skedde inom ramen för en utbildningsgren som vi kallade ”kognitionsfilosofi”, och som fokuserade på gamla och nya filosofiska problem kring begrepp som kognition, minne, föreställning och mental representation. Inkluderandet av en mer teknisk del i form av teorin för artificiella neurala nätverk var lite av ett vågspel men experimentet föll väl ut och ANN-kursen blev populär. För att passa deltagarnas i allmänhet inte alltför starka matematiska bakgrund fick alla grundbegrepp och basala resultat förklaras extremt noga, både vid undervisningen och i de preliminära upplagor av boken som användes. En annan sak som sticker ut med boken jämfört med

många andra framställningar av ANN är alla utvikningar till biologi och psykologi som den innehåller. Detta understryker den kognitionsfilosofiska kursens starka tvärvetenskapliga prägel, samtidigt som det återspeglar författarens bakgrund. I dag ser jag ingen nackdel med ett sådant perspektiv på ANN, och den definitiva andra upplagan kommer att innehålla ännu mer om hjärnans egna nätverk. Liksom många fler nya referenser än de som anges nedan.

De flesta av de personer som jag avtackar i det gamla förordet har också varit behjälpliga vid det fortsatta arbetet med boken och dess frågeställningar, och jag räknar inte upp dem igen. Dessutom vill jag nu tacka Carl-Martin Allwood, Birgitta Johansson, Niklas Klasson, Bertil Thomas och Björn Vickhoff för de kunskaper och idéer som de vidarebefordrat till mig under de senaste drygt 10 åren.

Det nya omslaget innehåller bilder som ritats av Lena Struck Malmgren och Sigun Bergstedt. Tack igen till er!

Nu följer några kommentarer till och kompletteringar av den gamla texten som senare ska införas i den nya upplagan.

1.1 Forskning om minne – från Aristoteles till ANN (ss 9–18)

En ny allmän referens är här Allwood, C.M. & Malmgren. H. (2012). Minne och kognition. I: Allwood, J. & Jensen, M. (utg.), *Kognitionsvetenskap*, ss. 159–73. Lund: Studentlitteratur.

En ny referens under avsnittet *Artificiella neurala nätverk* (s. 14) är Aggarwal, C. (2018). *Neural Networks and Deep Learning*. Cham: Springer Nature. Det är en grundlig och inte alltför svårläst framställning av ANN-teoriernas utveckling fram till våra dagar.

1.2 Vad är egentligen minne? (ss 18–22) och 1.3 Representation och information (ss 22–37)

Diskussionen om kognitivistiska vs icke-kognitivistiska förklaringar är fortfarande högaktuell. Ett exempel från den kognitionspsykologiska litteraturen är debatten kring teorin om ”predictive coding” (PC) i varseblivningen. Denna populära teori säger att vår hjärna kodar apriorisannolikheter för ett stort antal olika hypoteser om vad som kommer att hända närmast, och sedan uppdaterar sannolikheterna på ett matematiskt korrekt sätt utifrån felsignaler från det perceptuella systemet. Frågorna har ställts vad det betyder att hjärnan innehåller en *kod* för alla dessa sannolikheter och faktiskt *utför kalkyler* av det nämnda slaget. Se Bowers, J.S. & Davis, C.J. (2012). Bayesian Just-So Stories in Psychology and Neuroscience. *Psychological Bulletin 138*:3, ss. 389–424, och följ upp artikeln på t.ex. Google Scholar för att få en balanserad bild av debatten. Jfr också Malmgren, H., Hemeren, P., Haglund, B. & Svensson, H. (2012). Begrepp

och mentala representationer. I: Allwood, J. & Jensen, M. (utg.), *Kognitionsvetenskap*, ss. 175–90. Lund: Studentlitteratur.

Not 32 (s. 26): Malmgren (2006) finns nu tillgänglig som https://gup.ub.gu.se/file/208082. Beträffande Malmgren (1991) se nedan, kommentaren till sid 71 ff.

1.4 Dynamiska system och systemteori (ss 37–46)

En svaghet med boken är att den inte behandlar icke-deterministiska system. Det kanske blir bättre i nästa upplaga. Men där kommer antagligen också att stå mer om kaotiska system, ett begrepp som är högst relevant för hjärnforskningen (jfr. ss. 45f). Ett system av hjärnans komplexitet har en stark inneboende tendens att gå till kaos. Men kaos kan inte råda i beteendet. Hur går det ihop? Med vilka mekanismer kontrolleras kaos? Och har episoder av kaos någon biologisk funktion? En liten artikel om detta, i lättsmält form men med ett seriöst syfte, är H. Malmgren, H. (2010). En fågelskådares drömmar. I: *Drömmar. En vänbok till Ingemar Nilsson*, , ss 51–60. Göteborg: Göteborgs Universitet. Tillgänglig som https://gup.ub.gu.se/file/208061.

2.1 Icke-associativt minne (ss 47–52)

Malmgren (1984) (not 45, s. 50) finns nu tillgänglig på http://hdl.handle.net/2077/59605.
Om Ashby se också Malmgren, H. (2013). From Fechner, via Freud and Pavlov, to Ashby. *Constructivist Foundations, 9* (1), ss. 104–5.

2.2 Operant betingning (ss 52–64)

Avsnittet om olika former av kontroll (ss 55–57) kan illustreras som följer, hämtat ur min presentation *Adaptation och inlärning i komplexa system* vid en kurs för zoofysiologer vid Göteborgs Universitet 2007. Till den definitiva andra upplagan av boken planeras en liknande bild.

Artiklarna Malmgren (1985) (not 53, s. 58) och Malmgren & Östensson (1989) (not 54, s. 61) finns nu tillgängliga på http://hdl.handle.net/2077/59603 respektive http://hdl.handle.net/2077/59606.

Till den definitiva upplagan planeras ett någorlunda utförligt avsnitt om relationerna mellan operant inlärning i biologiska system, ANN och *reinforcement learning* (s. 63). Se också nedan, kommentaren till not 111, s. 147.

2.3 Klassisk betingning (ss 64–76)

Om representation av tid (s. 68–70): se kommentarerna nedan till avsnitt 10.4 (ss. 331 ff).

Det påstås på ss. 71ff att klassisk betingning (liksom habituering) kan ges en abstrakt, systemteoretisk förklaring. Idén, och argument för den, publicerades först i Malmgren, H. (1991). Learning by natural resonance, *Göteborg Psychological Reports* 21 (6), tillgänglig på http://hdl.handle.net/2077/59604. Se också not 32 på bokens s. 26.

3.2 Perceptuellt, episodiskt och semantiskt minne (ss 78–82)

Här hoppas jag kunna foga in fler aktuella filosofiska referenser i den definitiva andra upplagan.

Avsnitten 3.3–3.10 kommer att omarbetas i samarbete med kollegor som i dag arbetar med minnesprövning och/eller med demenssjukdomar.

3.7 Korsakoff och hans syndrom (ss 92–95)

Det kan inte nog betonas att det omedelbara minnet, manifesterat som förmågan att direkt upprepa en presenterad siffersekvens fram- eller baklänges, oftast fungerar perfekt vid minnestörningar av Korsakoff-typ (ss 93 f), t.ex. vid en skada på båda hippocampi (s. 95). Detta talar starkt mot alla teorier som innebär att hippocampusarna är nödvändiga för memorering och ”inre uppspelning” av korta tidsliga sekvenser. Se Malmgren, H. (2015). In What Sense, if Any, do Hippocampal “Time Cells” Represent or Encode Time? Poster vid konferensen *The Brain's Networks: Which are they, and what do they do?* September 18-20, 2015, Göteborg. https://neurophys.gu.se/digitalAssets/1553/1553055_helge_malmgren.pdf.
Jämför också kommentarerna till avsnitt 10.4 nedan.

3.8 Asteno-emotionellt syndrom (ss 96–99)

Syndromet är i dag mest känt under den mer lätthanterliga beteckningen ”hjärntrötthet”. För en auktoritativ översikt, som också tar upp några alternativa hypoteser om mekanismerna bakom tillståndet, se Johansson, B. & Rönnbäck, L. (2019). *Den ofattbara hjärntröttheten.* Lund: Studentlitteratur.

4.2 Något om verkliga nervceller (ss 105–112)

En mer utförlig beskrivning av LTP och LTD finns i Malmgren, H. & Wigström, H. (2012). Var sitter minnet? I: Allwood, J. & Jensen, M. (utg.), *Kognitionsvetenskap*, ss. 203–18. Lund: Studentlitteratur.

Förloppet vid aktivering och inhibering av nervceller påverkas av ett stort antal olika faktorer: signalsubstanser, modulerande substanser, hormoner, extracellulära elektriska fält… och nervsystemet (som i sig består av två, det centrala och det autonoma nervsystemet) interagerar hela tiden med ett annat stort kroppsligt system: det hormonella. Gränserna mellan dem är delvis svåra att dra, t.ex kan samma substans vara både ett hormon och en neurotransmittor. I boken står knappast något om detta, men i den definitiva andra upplagan ska jag försöka ge en översikt.

4.4 Nätverksarkitekturer (ss 120–126)

Om NeuralWorks (fig 22, s. 123 med flera): den som har kvar en gammal Mac-dator med G3/G4/G5-processor kan försöka få tag i det utmärkta utbildningsprogrammet *NeuralWorks Explorer* på begagnatmarknaden. Det håller fortfarande toppklass, och manualen är förnämlig. Tyvärr finns inte företaget kvar som producerade och sålde programmet.

Om feedback-nätverk (ss. 124 ff) kan man bland annat läsa i S Grossberg, S. (2013). Recurrent neural networks. *Scholarpedia, 8*(2):1888

4.5 Vad kan neurala nätverk göra? (ss 126–139)

Begreppet *mönsterassociation* (s. 129, se också nedan) blev ganska oklart i min framställning. Bättring utlovas. Samtidigt ska jag försöka reda ut vad man kan och bör mena med ”oövervakad”, ”övervakad” och ”styrd” inlärning i ANN-sammanhang.

4.6 Inlärningsalgoritmer för ANN (ss 139–149)

S. 130f: Ett annat exempel på gles kodning ges av M. Borga, H Malmgren & H Knutsson, (2000), Feature Selective Edge Detection. *Proceedings of 15th International Conference on Pattern Recognition, Barcelona.* https://www.researchgate.net/publication/2237134_FSED_-_Feature_Selective_Edge_Detection

Not 111, s. 147: en senare och ännu mer spektakulär kombination av reinforcement learning och ANN uppvisas av datorprogrammet *Alpha Zero*, som år 2014 visade sig överlägset både andra program och mänskliga schackspelare. Se Silver, D. et al (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science 362*, Issue 6419: 1140–4.

5.1 Neurala nätverk och sannolikheter (ss 151–173)

Detta är sannolikhetsteori och statistikteori för filosofer och andra icke-matematiker, och alltså mycket basal men nödvändig för att förstå resten av

boken. Men det är också (ss 168–173) lite statistikfilosofi för statistiker och andra icke-filosofer. Inte minst medicinare, som ofta bara känner till den Neyman-Pearsonska statistiktraditionen, kan ha nytta av de sidorna.

5.2 Vektorer och matriser (ss 173–190)

I detta avsnitt letar man däremot förgäves efter filosofiska inslag. Det är som helhet en extremt detaljerad framställning av den linjära algebrans elementa.
Formeln 5.2.16 (s. 189) är resultatet av ett feltryck. Det ska vara: $\mathbf{T = M_2 M_1^{-1}}$, som det så riktigt står på s. 202.

6.1 Den enkla perceptronen (ss 191–200)

Det som på s. 195 kallas ett ”direkt” bevis för att den enkla perceptronen inte kan lösa XOR-problemet är ju ett bevis genom *reductio ad absurdum,* och därför i en mening ”indirekt”. Jag borde omnämnt det som ett ”algebraiskt” bevis, i motsats till det geometriska.

Not 123 (s. 199): För att kunna använda den datormodell som omtalas här, liksom en liknande modell av Hopfieldnätet (se nedan), räcker det inte med att ha en gammal Mac. Man måste också ha en som kan köra Classic (System 7.6–9.1), vilket i sin tur kan göras inom OSX 10.4 men inte i senare OS. Den som är lycklig nog att ha tillgång till en sådan maskin kan vända sig till författaren för att få tillgång till programmen. Länken i noten fungerar inte längre.

6.2 Mönsterassociation, med mera, i linjära nätverk (ss 200–215)

s. 213: Termen ”autokorrelation” används inte sällan i samband med auto-associativa ANN, eftersom träningen av dem resulterar i korrelationer mellan komponenterna i inputvektorerna. Det är en olämplig terminologi eftersom ”autokorrelation” i andra sammanhang ofta syftar på en korrelation *över tid*, alltså mellan successiva tillstånd i ett system. Se också nedan, avsnittet om Hopfieldnätverket.

6.3 ”Linjära” ANN i vid mening (ss 215–217)

Begreppet linearitet är inte så lätt att greppa som kan kanske kan tro, och jag lyckas väl inte helt i boken. I den definitiva andra upplagan ska det förhoppningsvis vara någorlunda tydligt.

7.1 Lärande system med attraktorer (ss 219–232)

Not 129, s. 221: för att inte strida mot en vanlig betydelse av ”autokorrelation” (se ovan) är det bättre att tala om auto-association hos dessa attraktornätverk. Hopfieldnätet kan då kallas *den diskreta auto-associatorn*.

Ss. 228 ff, rubriken och texten: samma kommentar

Not 130, s. 232: jfr kommentar till not 123, s. 199. Kontakta författaren för tillgång till denna modell.

7.3 Kontinuerliga system och kontinuerliga attraktorer (ss 232–253)

Ss. 243–53: Min modell för online-inlärning av graderade responser i kontinuerliga system är ett komplement till den teori om ”naturlig resonans” i diskreta system som presenterades på ss. 71 ff. I den teorin har systemets ”sökande” efter en attraktor ingen riktning utan sker helt slumpmässigt. I den kontinuerliga teorin finns en sådan riktning.

Not 136, s. 244: Malmgren (2002) finns nu tillgänglig som https://gup.ub.gu.se/file/208083.

8.1 Kooperation och kompetition i neurala nätverk (ss 255–259)

Beträffande kantdetektion och kantfilter (s 257) se kommentar till ss. 130f, referensen Borga, M. et al. (2000), samt nedan om faltningsnätverk.

8.2 SOM – den självorganiserande kartan (ss 259–270)

En blick på den nyare litteraturen antyder att SOM håller ställningarna väl trots att det finns många alternativa metoder för dimensionsreduktion av data. För några aktuella perspektiv se Vellido, A., Gibert, K., Angulo, C. & Guerrero, J.D.M. (utg.) (2020). *Advances in Self-Organizing Maps, Learning Vector Quanitization, Clustering and Data Visualiation.* Cham: Springer Nature Switzerland. – WEBSOM (s. 266) har fått en mycket intressant uppföljare i ett projekt som siktar på att beskriva ämnesstrukturen i mängden av *alla* medicinska artiklar som finns upptagna i *Medline*. Se Skupin, A., Biberstine, J.R. & Börner, K. (2013). Visualizing the Topical Structure of the Medical Sciences: A Self-Organizing Map Approach. *PLoS ONE 8*(3): e58779.

8.4 ART (Adaptive Resonance Theory) (ss 270–274)

Två nyare artiklar om ART som biologisk modell är Grossberg, S. (2013). Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks 13,*1–47 och Grossberg, S. (2019). The resonant brain: How attentive conscious seeing regulates action sequences that interact with attentive cognitive learning, recognition, and prediction. *Attention, Perception, & Psychophysics 81*:2237–64. Den senare artikeln är open access och finns på https://doi.org/10.3758/s13414-019-01789-2. Kopplingen mellan perception och handling i den är särskilt intressant. Och man behöver inte vara övertygad om att Grossbergs ambitiösa projekt har lyckats för att ha stor behållning av att läsa hans artiklar.

9.1 Kraftfulla – och kanske för kraftfulla – modeller (ss 285–295)

Det har hänt mycket på området sedan boken skrevs, inte minst i samband med utvecklandet av nätverk för *deep learning*. En översikt över hur dessa

nätverk löser problemet med val av modellstyrka kommer att ges i den definitiva andra upplagan. Jfr också ss. 307f.

9.3 Inlärning i flerlagrade perceptroner (ss 298–308)

Intressant nog är varianter av *back propagation of error* centrala även inom fältet deep learning. Detta förhållande kommer att beskrivas närmare i den definitiva upplagan.

9.4 Bayesianska neurala nätverk (ss 308–311)

Med termen ”bayesianska nätverk” och förkortningen BN avser man som nämns oftast Bayesian *belief* networks, förkortat BBN. Dessa nätverk behandlas mycket kortfattat i boken men kommer att få en lite mer utförlig beskrivning i den definitiva andra upplagan. De är trots allt också lärande system. En diskussion av användningen av BBN för analys av stora datamängder är Dybowski, R. (2017). *Bayesian Networks and Big Data*. Föreläsning, Göteborgs Universitet, tillgänglig på https://vimeo.com/232254302.

9.5 LVQ, Learning Vector Quantization (ss 312–318)

Denna Kohonen-klassiker är i dag kanske ännu mer aktuell än SOM. En sökning på Google Scholar ger över 2000 träffar för 2019. Nya versioner av LVQ erbjuder kanske ett alternativ till deep learning för stora, heterogena datamängder när datorresurserna är begränsade. Se Villman, T., Bohnsack, A. & Kaden, M. (2017). Can learning vector quantization be an alternative to SVM and deep learning? - recent trends and advanced variants of learning vector quantization for classification learning. *Journal of Artificial Intelligence and Soft Computing Research 7*:1, 65–81, men även Vellido, A. et al. (2020) (se ovan, avsnittet om SOM).

9.6 Radialbasnätverk och supportvektormaskiner (ss 318–320)

Här har hänt en hel del ”sedan sist” och både RBF-nätverk och SVM används mycket i dag. Detaljer får anstå till den defiitiva upplagan. Men det ska poängteras att det kan vara bra att förstå hur LVQ och RBF-nät fungerar när man ska begripa sig på nätverk för deep learning.

Därmed är tiden kommen för att säga något om de senare, nu så berömda storheterna.

En pedagogisk föreläsning om djupinlärning för bildanalys är Dybowski, R. (2016). *Deep learning for Analysis of Medical Images*. Göteborgs Universitet. Tillgänglig på https://vimeo.com/201475940.

För att undvika dubbelarbete just nu citerar jag här, med tillstånd från copyrightinnehavarna, ur mitt eget kapitel i en kommande bok: Petersson, G., Rydmark, M. & Thurin, A. (2020). *Medicinsk Informatik.* Stockholm: Liber. Den definitiva andra upplagan av *Inlärning och*

minne i neurala nätverk kommer förstås att vara mycket mer utförlig om djupinlärning.

Djupinlärning
På 2010-talet gjordes upptäckter/uppfinningar som återförde ANN till händelsernas centrum när det gäller beslutsstöd, och sedan dess har utvecklingen gått snabbare än någonsin tidigare. De nya modellerna brukar sammanfattas under namnet djupinlärning (deep learning). ”Deep” syftar på att nätverken har mer än ett dolt lager. Fler dolda lager gör det möjligt att lösa ett givet problem med hjälp av ett lägre totalt antal noder. Dock kan sådana nätverk ta längre tid att träna. Dagens djupinlärningsnät har många fler dolda lager än vad som tidigare varit vanligt, ibland över 1000. Förr var sådana nätverk närmast omöjliga att hantera givet dåtidens begränsade datorkraft. Nya träningsalgoritmer, nya val av egenskaper hos noderna och nya sätt att förhindra överinlärning har bidragit till att förbättra nätverken. Framtiden för de artificiella neurala nätverken ter sig därför ljusare än på mycket länge.

Faltningsnätverk
Gemensamt för alla bilder är att det finns en rumslig struktur i dem, något som de flesta äldre neurala nätverk är mycket dåliga på att exploatera. För en klassisk MLP är en bild inget annat än en inputvektor, där ordningen dessutom är godtycklig. När en bild tas in i ett ANN behövs lika många neuron på inputsidan som antalet bildelement. Detta innebär att en MLP för bilder blir mycket komplex och kräver ett extremt stort antal exempel för att kunna tränas på rätt sätt, något som sällan eller aldrig är tillgängligt. I viss utsträckning kan detta hanteras genom olika former av förprocessande som förenklar uppgiften. Vid ansiktsigenkänning kan man till exempel börja med att normalisera ansiktets position i bilden, så att nätverket inte behöver tränas på uppochnervända och sneda bilder. Om en lokal struktur, till exempel ett ansikte, förekommer flera gånger i bilden kan man ”be” nätverket att först analysera denna struktur. Ett sådant förprocessande måste dock designas för varje specifik typ av bild.

Så kallade faltningsnätverk för bilder har inbyggda, förprocessande steg i form av filter som själva letar efter och hittar lokala drag i bilden. Den operation som då används kallas just ”faltning” och innebär en jämförelse mellan en filtermatris och områden i bilden. Filtren kan till exempel leta efter kanter och hörn i bilden. I nästa lager noteras då bara var de här lokala dragen finns. I djupare lager av filter och andra operationer syntetiseras de funna strukturerna till större objekt eller delar av objekt. Mekanismen påminner om neurofysiologiska processer i vårt visuella system. De allra djupaste lagren i ett faltningsnätverk påminner om en klassisk MLP, och där görs den slutliga klassifikationen eller igenkänningen. Den input som de lagren får är då inte på långt när så komplex som bilden själv, och de sista stegen kräver inte särskilt mycket datorkraft.

10.4 Tidsfönster i perception och motorisk kontroll? (ss 329–332)

Diskussionen om hur tid representeras i nervsystemet, inte minst i form av korttidsminne, omedelbart minne och motoriska planer, är mycket intensiv i dag och i den definitiva andra upplagan av boken kommer detta avsnitt att ta betydligt större plats.

Not 162, s. 329: Malmgren, H. (2004). *Why the past is sometimes perceived, and not only remembered* finns tillgänglig som https://gup.ub.gu.se/file/208065.

I postern Malmgren, H. (2015) – se kommentaren till ss 92–95 ovan – skisserade jag en helt ny teori om inlärning av sekvenser. Sedan dess har jag tillsammans med en yngre kollega utarbetat olika versioner av ett artikelmanus om den nya modellen, men ännu inte publicerat mer än ett abstract till en konferens, som vi dessutom inte kunde närvara vid. Abstractet Malmgren, H. & Klasson, N. (2017). Modelling complex Hebbian reverberations with sets of spiking oscillators, avsett för *The 21th annual meeting of the Association for the Scientific Study of Consciousness, Beijing 13–16 June 2017*, lyder så här (med den centrala frågan och essensen av svaret betonade):

> In 1949, Donald Hebb suggested that sensory information can be held in the form of reverberating neural activity before being coded as synaptic modifications. The idea has won widespread acceptance, but one central question remains: **by what mechanism can an incoming sensory sequence force the output of a neuronal assembly to enter a closed trajectory that mimics the sequence**? We here offer a radically new answer. Our model assumes the existence of a large set of neural oscillators, having a wide spectrum of frequencies, that interact with the input through a resonant layer. In the learning phase the activity in the resonant layer mirrors input. At each moment, oscillators that are close to firing threshold are set either to be silent or to fire depending on the present input. **This entrainment gives rise to a frequency-phase transform, coded as the current activity in the set of oscillators, of the input sequence.** In the retrieval phase, the input signal is shut off from access to the resonant layer. This layer now mirrors the activity of the oscillators, or of a chosen subset of oscillators. If, in the simple discrete-time case, only oscillators with a period of N are allowed to influence the resonant layer, the input during the last N moments will be replayed in this layer during the following N moments. The choice of oscillator period may correspond to a conscious decision to recall the immediately past N moments. We show results from two simulations of the proposed mechanism.

Allt detta kommer att förklaras närmare i den tryckta andra upplagan av boken, då vår artikel förhoppningsvis också kommer att vara publicerad.

Förord

Texten i denna bok har gradvis vuxit fram under mer än tio säsongers undervisning om neurala nätverk vid Göteborgs Universitet – i första hand inom ramen för den kognitionsfilosofiska grundkursen *(Modeller för Mänskligt Tänkande)* vid Filosofiska Institutionen, men också i form av kortare kursmoment inom ämnena psykologi och zoologi och enskilda föreläsningar vid ett antal andra institutioner. Vill man undervisa om neurala nätverk för humanister, samhällsvetare, medicinare och andra icke-specialister på det matematiska området kan man inte bara plocka en lärobok från bokhandelshyllan. Det finns en uppsjö av introduktioner till neural nätverksteori (de flesta på engelska), men nästan alla kräver rejäla matematiska förkunskaper och mycket få sätter in teorin i ett större filosofiskt, psykologiskt och biologiskt sammanhang. De är kort sagt oftast anpassade till behoven hos en annan grupp av studenter, nämligen de tekniker och matematiker som läser neural nätverksteori med tanke på dess tekniska tillämpningar.

Därför börjar den här boken med fyra kapitel som ger: en kort historisk och filosofisk bakgrund till dagens forskning om inlärning och minne; en översikt över inlärningsfenomen hos andra djur än människan och några förklaringsmodeller för dessa fenomen; en selektiv sammanfattning av modern psykologisk och neuropsykiatrisk forskning om mänskligt minne; respektive några basala fakta om neurala mekanismer. Även studenter med förkunskaper i psykologi bör läsa dessa avsnitt, eftersom de i mycket innebär en kritisk granskning, utifrån författarens och hans medarbetares egen filosofiska och empiriska forskning, av centrala begrepp i kognitionspsykologi och neuropsykologi. – Först därefter (från och med avsnitt 4.3) kommer bokens huvudtema, som är grunderna i teorin för *artificiella neurala nätverk* (ANN). Jag har försökt skriva dessa kapitel om ANN på ett sätt som inte förutsätter mer än allmän gymnasiekompetens i matematiska ämnen, vilket har varit en utmaning i många avseenden. Eftersom teorin i vissa stycken trots allt fordrar något större matematiska förkunskaper än så, har jag infogat några orienterande avsnitt om diskreta och kontinuerliga system (avsnitt 1.4 och 7.3), statistisk infe-

rensteori (5.1) och elementär linjär algebra (5.2) i boken. Man kan kanske tycka att vissa av dessa avsnitt är överdimensionerade i förhållande till boken i övrigt, men de är inte bara avsedda att underlätta läsningen av resten av boken, utan också att göra det möjligt för läsaren att ta del av mer avancerad litteratur om ANN. Den som vill ha en detaljerad matematisk förståelse av området har som sagt en stor engelskspråkig litteratur att vända sig till.

Det händer att studenter med examen från teknisk fakultet som gått vår ”filosofiska” ANN-kurs säger ”Jag visste redan hur man räknar med neurala nätverk, men först nu förstår jag vad det handlar om.” Det är min förhoppning att boken på motsvarande sätt ska kunna vara till nytta även för teknologer, matematiker och statistiker som vill ha en djupare förståelse av neural nätverksteori, dess idémässiga förutsättningar och dess relevans för livsvetenskaperna än vad den matematiska formalismen i sig erbjuder.

Två områden som jag måst lämna utanför framställningen är neuroanatomi och neurofysiologi, bortsett från basala neuronala mekanismer och vissa anatomiska detaljer av speciell relevans för minnesforskningen. Dels skulle boken snabbt blivit alldeles för tjock om den också hade innehållit en generell introduktion till dessa ämnen, dels finns det redan många lättillgängliga framställningar om nervsystemet på marknaden, även på svenska och även skrivna för icke-medicinare. Det rekommenderas att den icke medicinskt skolade läsaren repeterar grundläggande fakta om nervsystemet i någon sådan framställning innan hon/han kastar sig över den här boken.

Läsaren kommer också, kanske till sin besvikelse, att finna att boken inte innehåller särskilt mycket om konkreta tillämpningar av ANN-teorin. Detta gäller både ANN som modeller för hjärnfunktioner och tekniska tillämpningar av den matematiska teorin för lärande system. En förklaring till detta är att en representativ redogörelse för sådana tillämpningar skulle fordra minst varsin egen bok utöver denna. Jag har därför istället försökt guida läsaren vidare till böcker som redan finns på dessa områden. När det gäller ANN-modeller av hjärnan tillkommer att en framställning av dem, för att inte riskera att bli ytlig och trivial, måste bygga på en betydligt djupare kunskap om neuroanatomi och neurofysiologi än vad denna bok kan förmedla. Om jag någonsin lyckas skriva en bok om det ämnet (vilket ingår i livsplanen) kommer den därför att ha en något annorlunda målgrupp än vad den föreliggande framställningen har.

Efter denna uppräkning av vad som *inte* står i boken kan det vara på sin plats att nämna att det finns material i den, som inte kan inhämtas i andra framställningar om neurala nätverk eller i vanliga psykologiska läroböcker. Jag tänker inte bara på de filosofiska kommentarerna om begreppet ”representation” (1.3), utan också på mycket av det som står om organisk psykiatri, dvs. psykiska följder av hjärnskador och hjärnsjukdomar (3.6–3.10). Detsamma gäller de abstrakta inlärningsmodeller som beskrivs i slutet av avsnitten 2.1–2.3 samt i avsnitt 7.3 och 10.4. I samtliga dessa fall rör det sig i stor utsträckning om resultat av min egen och mina medarbetares forskning. När det gäller det organiskt-psykiatriska momentet kan en potentiell läsare av boken tänkas invända att detta inte hör hemma i en bok om artificiella neurala nätverk. Men det finns ett viktigt motiv för att ha med det, utöver det egoistiska motivet att vilja presentera sin egen forskning. En anledning till att man överhuvudtaget sysslar med ANN-teorier är nämligen att man vill modellera hjärnfunktioner, och hjärnskadesyndromen utgör en av våra huvudkällor till kunskap om hjärnfunktioner. Det är därför angeläget att ANN-forskaren har en något så när korrekt bild av det organiskt-psykiatriska fältet, och en sådan bild är enligt min mening inte helt lätt att hitta i standardlitteraturen.

Jag är stort tack skyldig många av mina medarbetare vid Göteborgs Universitet och andra akademiska lärosäten, liksom alla studenter som genom åren har kommit med goda uppslag och välmotiverad kritik. Särskilt vill jag nämna Björn Haglund och Holger Wigström. Björn och jag skrev artiklar om minnesteori tillsammans redan på 70-talet, och har haft många spännande samtal om dylika ting till och från sedan dess – även i anknytning till manuset till denna bok. Holger lärde mig en gång de första grunderna i ANN-teorin, och vårt senare samarbete har för mig inneburit en ovärderlig kontaktyta med den empiriska forskningen kring hjärnans minnesmekanismer.

Min förste och bäste psykologilärare Gösta Fröbärj har jag att tacka för många insikter om människosjälens natur. Utan det långvariga samarbetet med Göran Lindqvist hade mina kunskaper om hjärnans reaktion på skador och sjukdomar förvisso varit mycket begränsade. Många års bordssamtal med min tidigare hustru Kristina Malmgren hjälpte mig också att hålla mig någorlunda ajour med utvecklingen inom neurovetenskaperna. Andra seniora kollegor som betytt mycket är Sven Carlsson, som bland annat hjälpte mig att inse hur viktig den djurexperimentella psykologin är för vår förståelse av den mänskliga hjärnan; Alvar Ellegård, som tidigt bidrog till att inrikta min verksamhet åt det neurala

nätverkshållet; Bo Berndtsson, konsult i diverse matematiska frågor som övergått min egen horisont; Martin Rydmark och Lars Lindström som varit stimulerande samarbetspartners i flera forsknings- och utbildningsprojekt med anknytning till neurala nätverk; Richard Dybowski, suverän teoretiker, god pedagog och god vän; slutligen Magnus Borga, både uppskattad forskningspartner och (föga adaptivt) filter för dåliga matematiska formuleringar.

Bland doktorander och andra yngre medarbetare måste jag lyfta fram Filip Radovic, Erik Olsson, Daniel Ruhe, Mikael Jensen och Carl-Gustaf Wikstrand. Filip och Erik var flitiga hjälplärare på vår ANN-kurs i många år, och konstruerade också den trevliga modell för stereopsi som jag presenterar i bokens avsnitt 8.5. Daniel tog sedan över våra datorlaborationer och, inte minst, våra datorer, som under hans kompetenta tillsyn skött sig alldeles utmärkt. Erik har fortsatt sin verksamhet som doktorand i min forskargrupp, och det samarbetet har nog givit handledaren minst lika mycket som det givit doktoranden. Mikael läste, liksom Carl-Gustaf, Daniel, Erik och Holger, en version av hela manuset och lämnade många viktiga kommentarer till det.

Otaliga studenter, vars namn i stor utsträckning försvunnit in i mina neurala nätverks mer oåtkomliga skrymslen, har hjälpt till att förbättra texten genom sina kommentarer till texten och till lektionerna. Förvisso har jag glömt, eller annorledes utelämnat, ytterligare ett stort antal betydelsefulla personer i denna uppräkning, men för att inte förordet ska svälla ut över alla gränser måste jag nöja mig med ett kollektivt Tack! till alla er.

Tack också till John Wavle vid NeuralWare, Inc. för tillståndet att använda skärmfoton från NeuralWorks, och till Sigun Bergstedt som (liksom Daniel) stått för värdefull teknisk hjälp i arbetet med illustrationerna. Mina kontaktpersoner vid Studentlitteratur, det vill säga Ann Wirsén-Meurling, Eva Broberg och Kajsa Persson, har visat stort tålamod inför många uppskjutna deadlines. Och vad min nuvarande hustru Lena betytt för slutförandet av det här arbetet har jag redan berättat för henne.

Hålanda den 4 juli 2007

Författaren

1. Introduktion

1.1 Forskning om minne – från Aristoteles till ANN

När man ber en person ge några exempel på minne och inlärning, är det mycket sannolikt att hon beskriver företeelser av följande slag. Förmodligen berättar hon om en minnesbild som hon har från en tidigare händelse – i barndomen, förra sommaren, eller i går. Dessutom nämner hon några fakta som hon fick lära sig i skolan och som hon fortfarande minns, till exempel att gullviva heter *Primula veris* på latin, att det finns svarta svanar i Australien eller att Martin Luther King mördades 1968. Dessa två typer av minne har fått varsitt namn i modern psykologisk teori: *episodiskt* och *semantiskt* minne. Episodiskt minne är individens ofta mycket åskådliga (bildmässiga) minne av självupplevda händelser, medan semantiskt minne är kunskaper av icke självbiografisk karaktär, ofta av generellt slag och som regel formulerbara i språkliga termer (bland annat beroende på att inlärningen normalt skett via andras beskrivningar istället för genom självupplevande). Distinktionen, som den här formulerats utgående från typiska läroboksframställningar i kognitiv psykologi, är inte särskilt klar, mycket beroende på att den bygger på tre olika klassifikationsprinciper: en kodningsprincip (bildmässigt vs språkligt minne), en innehållsprincip (minnet handlar om enskilda händelser i individens liv, respektive om något annat) och en kausal princip (minnet har uppkommit genom direkt observation respektive på annat sätt). Men det hindrar inte att den kan vara nyttig som en första sorteringsprincip och för att peka på olika, kontrasterande typfall av minne.

Episodiskt och semantiskt minne är dock bara två i en hel rad av minnestyper. Ett sätt att ge en första bild av dem alla är genom en kort historik över milstolpar i minnesforskningen. En sådan historik följer därför nu. Den är inte bara komprimerad utan också vinklad, eftersom den ska leda läsaren fram till dagens forskning om inlärning i neurala nätverk. De flesta av de områden av minnesforskningen som berörs här kommer att tas upp till fördjupad behandling i de två följande kapitlen.

Associationsfilosofi och associationspsykologi

Filosofin har intresserat sig för minnets mekanismer åtminstone sedan Aristoteles' tid.[1] Aristoteles var den som först framhävde *association* som en grundläggande princip för själens arbetssätt: genom att en person iakttagit två fenomen i regelbundet nära tidsligt samband, kommer en iakttagelse av det ena fenomenet att automatiskt framkalla en tanke på det andra. Under 1700- och 1800-talen utvecklades en detaljerad teori om associativ inlärning inom den brittiska empiristiska filosofin; två förgrundsfigurer var David Hume och James Mill. Om, säger teorin, vi iakttar en association mellan händelsen A och händelsen B, så medför en senare iakttagelse av händelsen A en förväntan om att också B ska inträffa (vi "associerar" från A till B). Om A inträffar före B talar man om *successiv* association. Observera att "association" används dels för att beteckna det faktiska sambandet mellan händelser, dels för vår "mentala sammankoppling" av dem. *Simultan* (eller *synkron*) association, dvs. association mellan iakttagna samtidiga händelser eller ting, fick förklara bland annat *perceptuell komplettering*, det fenomen som innebär att vi i en viss mening "ser" även baksidan av ett föremål även om vi bara "direkt" ser framsidan. Flera spekulativa fysiologiska förklaringar till associativa fenomen såg också dagens ljus under dessa århundraden, inte minst sådana som baserades på en mekanistisk uppfattning om nervimpulsen som ett slags fortledd vibration.[2]

Runt det förra sekelskiftet blev den empiriska forskningen kring minnets mekanismer en viktig del av såväl experimentalpsykologin som fysiologin. Den introspektiva metod som filosoferna hade använt kompletterades således av kontrollerade humanpsykologiska experiment med inlärning av exempelvis successiv association mellan meningslösa stavelser.[3]

Klassisk och operant betingning

Pavlovs inlärningsförsök på hundar i början av 1900-talet och hans lära om betingning av reflexer blev ett slags biologisk verifikation av teorier-

[1] För en ovärderlig översikt över associationsbegreppets historia, med relevanta utdrag ur viktiga originaltexter, se Mandler & Mandler (1964).

[2] David Hartley, en annan av associationspsykologins fäder, utformade en sådan teori. Se vidare Mandler & Mandler (1964), kap. 3.

[3] Det stora namnet här är Ebbinghaus. Om denna tradition kan man gärna läsa i en klassisk framställning av inlärningspsykologins historia, nämligen Bower & Hilgard (1981), kap. 6.

na om associativ inlärning.[4] En hund som upprepade gånger varseblir ett samband mellan en betingad stimulus, CS (Conditioned Stimulus) och en obetingad sådan, UCS (UnConditioned Stimulus), kommer så småningom att reagera på CS med en reaktion CR (Conditioned Response) som liknar dess reaktion UCR på den obetingade stimulus.[5] I de mest kända försöken är CS en ringsignal, UCS en visuellt presenterad matskål och R salivutsöndring. Vår beskrivning av resultaten av Pavlovs försök är mycket förenklad och vi kommer att få anledning att modifiera den senare.

Vid ungefär samma tid började man också uppmärksamma en annan mekanism för inlärning, nämligen *effektprincipen*, som först formulerades av djurpsykologen Thorndike. Om ett beteende (en respons) R i en viss situation S följs av en belöning B^+ så kommer R sannolikt att förstärkas (dvs. djuret blir mer benäget att uppvisa responsen R i S), medan ett beteende som följs av en bestraffning B^- istället tenderar att försvagas. Psykologer som Skinner gjorde senare systematiska experiment på djur och människor kring dylik "operant" eller "instrumentell" betingning.[6]

Hebbs princip

Vid mitten av det förra seklet lade den kanadensiske psykologen Hebb fram ett förslag till fysiologisk förklaring av associativ inlärning, som blev mycket uppmärksammat och som också har haft stor betydelse för teorin om artificiella neurala nätverk. En vanlig formulering av "Hebbs princip" lyder som följer: Om ett neuron A är aktivt vid en viss tidpunkt, A har en förbindelse med ett annat neuron B, och B aktiveras samtidigt eller omedelbart efteråt, så kommer förbindelsen från A till B att förstärkas (bli effektivare). Detta medför i sin tur att senare aktivitet i A kommer att ha en ökad sannolikhet att ge upphov till aktivitet i B.[7] I kortversion: *samtidig aktivitet hos A och B leder till att förbindelsen från A till B förstärks*. Vi kommer att tala rätt mycket om Hebbs princip, och

[4] Pavlov är en mycket njutbar författare att läsa i original. Se Pavlov (1960) [1927)] och (1928).

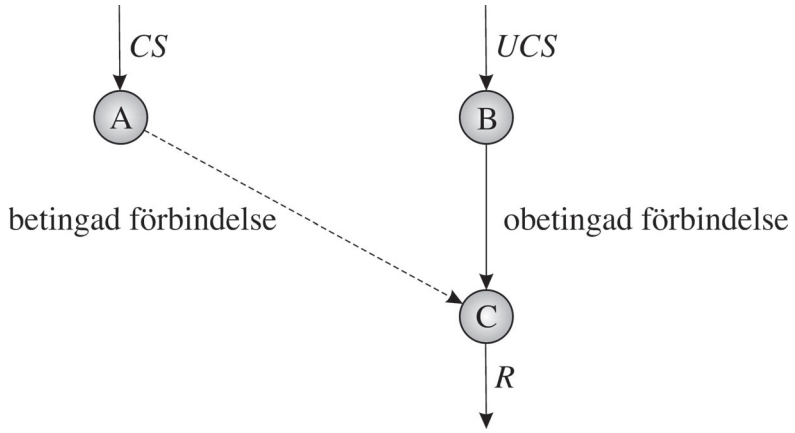
[5] Detta är inget feltryck. Det heter faktiskt "*en* stimulus" (och "flera stimuli"), inte "ett stimulus".

[6] Om Thorndike och Skinner kan man också med fördel läsa i Bower & Hilgard (1981), kap. 2 respektive 7.

[7] Hebb (1949). Ursprungsformuleringen innehåller kravet att A skall "bidra till" aktiveringen av B, men denna förutsättning behövs inte för förklaringen av klassisk betingning och används sällan eller aldrig i ANN-teorin. – Teorier som mer eller mindre anteciperar Hebbs idéer finns hos bland annat Konorsky (1948).

olika varianter av den, senare. Låt oss här bara, som ett första smakprov på en modell av ett neuralt nätverk, presentera en enkel (och i många avseenden orealistisk) skiss till förklaring av Pavlovsk inlärning med hjälp av Hebbs hypotes.

I figur 1 betecknar cirklarna A, B och C nervceller som antas vara involverade när djuret uppfattar CS och UCS, respektive avger responsen R. Vi utgår från att storleken på den signal som når fram till C beror dels på storleken hos signalerna som går ut från A och B, dels på styrkan (effektiviteten) hos förbindelserna från A till C och från B till C. Vidare antar vi att det finns ett tröskelvärde som den inkommande signalen till C måste överstiga för att C skall aktiveras och R utlösas. Den heldragna pilen från B till C beskriver det förhållandet, att nervcellen där UCS representeras redan från början kan aktivera nervcellen som utlöser R – det ligger ju i själva definitionen av en obetingad stimulus. Den streckade pilen från A till C beskriver att förbindelsen från A till C däremot (från början) är för svag för att CS ensam skall kunna utlösa R.



Figur 1. Ett mini-nätverk för klassisk betingning. Förklaring: se text.

Vad händer nu om man ett flertal gånger presenterar CS och UCS tillsammans? Jo, vid varje presentation kommer neuronen A och C att vara aktiva samtidigt (eller med en mycket liten tidsdifferens), och om Hebb hade rätt så stärks förbindelsen då mellan A och C. Därmed ökas storleken hos den signal som kommer fram till C från A, och efter tillräckligt många försök har den blivit tillräckligt stor för att en signal härstammande från enbart A ska överstiga tröskeln för aktivering av C. Detta betyder förstås inget annat än att CS *själv* kan utlösa R.

I modern neural nätverksteori använder sig man ofta av olika varianter av Hebbs postulat, men man binder sig inte vid den förenklande idén att ett enda neuron är ansvarigt för representation av en stimulus respektive av en respons. Mer om detta nedan, särskilt i avsnitt 4.6.

Djurexperimentell inlärningspsykologi förblev en central del av psykologin fram till omkring 1970. Många teorier för associativ inlärning presenterades.[8] De flesta psykologer stannade för uppfattningen att "klassisk" (Pavlovsk) och operant inlärning är två olika saker (som förvisso kan existera i blandningar). Men man studerade inte bara associativ inlärning utan även fenomen som exempelvis *habituering*. Habituering innebär att en organisms lokala eller globala respons på en stimulus – t.ex. ett högt ljud – minskar i storlek när denna stimulus upprepas. Sådan icke-associativ inlärning förekommer hos praktiskt taget alla djur man känner till, alltså även hos encelliga organismer som inte uppvisar någon associativ inlärning.[9]

Den kognitiva revolutionen och neuropsykologin

När den "kognitiva revolutionen" i psykologin inträffade runt 1965–70 minskade psykologernas intresse för djurexperimentella studier av inlärning. Istället började man åter forska mycket kring människans minne. En hel del av denna forskning inriktade sig på studiet av hur hjärnskador och hjärnsjukdomar påverkar minnet. Tillstånd som varit kända länge i psykiatrin, såsom Korsakoffs amnestiska syndrom (där det episodiska minnet är särskilt drabbat – se avsnitt 3.7) rönte ett förnyat intresse bland neuropsykologer.[10] Man visade bland annat att patienter med detta syndrom, som typiskt har en omfattande minneslucka bakåt i tiden och synbarligen en total oförmåga att lära sig något nytt material, ofta kan lära sig nya *praktiska färdigheter.* Inte minst detta fynd gör att man i moderna minnesteorier skiljer noga mellan teoretisk och praktisk inlärning, eller mellan *deklarativt* och *procedurellt* minne. Mer om allt detta i kapitel tre.

På 70-talet kom också ett genombrott i det *neurofysiologiska* studiet av minnesmekanismer, i och med de första beskrivningarna av fenomenet

[8] Mackintosh (1967) och (1983) erbjuder utomordentliga sammanfattningar av såväl data som teorier på detta område.
[9] Se Wyers et al. (1973).
[10] Jämför t.ex. Milner et al. (1968).

LTP (långtidspotentiering) i hippocampusneuron.[11] Detta fenomen, som vi också ska tala mer om senare, stämmer väl med Hebbs princip och anses av många vara en möjlig grund för associativ inlärning. På senare tid har man också fått god evidens för att LTP äger rum i samband med inlärning.[12] LTP och besläktade mekanismer har därför varit föremål för många empiriska studier under det senaste kvartsseklet, inte minst av svenska forskare. Även habituering har studerats flitigt av neurofysiologerna, en forskning som bland annat hjälpt till att motivera ett Nobelpris.[13]

Artificiella neurala nätverk

I början av 1980-talet tog utvecklingen inom en viss teoretisk disciplin ny fart, nämligen teorin om artificiella neurala nätverk (ANN). Teorin har, som namnet antyder, i hög grad inspirerats av vad vi vet eller tror oss veta om verkliga neurala system. Ett artificiellt neuralt nätverk är i första hand en rent matematisk konstruktion eller modell, som inte behöver vara realiserad genom någon ”hårdvara”. Nätverket antas ta emot en komplex signal, omvandla den och avge resultat i form av en ny signal, allt på ett sätt som liknar hur ett system av nervceller i hjärnan arbetar. Ett typiskt ANN modifierar också kontinuerligt sitt arbetssätt i ljuset av de signaler det får, dvs. det *lär sig* (i en viss mening) av erfarenheten. Framsynta forskare föreslog teoretiska modeller av nervnät redan på 1940- och 50-talen, men av olika skäl – som vi ska återkomma till – kom dessa teorier inte att spela någon stor roll i forskarsamhället förrän på 1980-talet.[14] I dag är ANN dock en väletablerad forskningsgren, eller rättare sagt flera forskningsgrenar, inom vilka man ofta ser ett intimt samarbete mellan matematiker, statistiker, psykologer, kognitionsvetare, hjärnfysiologer och filosofer.

Ett av de klassiska motiven bakom att ställa upp ANN-modeller är att genom beräkningar och simuleringar bättre förstå hur inlärning i verkliga nervsystem fungerar. Tillsammans försöker forskarna kartlägga psykiska och neurala mekanismer, och i centrum för intresset står minnets och inlärningens gåta. Redan tämligen enkla ANN-modeller ger tänkbara om än schematiska förklaringar av vissa minnes- och inlärningsfenomen som

[11] Bliss & Lømo (1973).
[12] Se Whitlock et al. (2006).
[13] Se Kupferman & Kandel (1969), Kandel (2001).
[14] För en historik se Rumelhardt & McClelland (1986). Appendixet till Simpson (1990) ger också en god historisk överblick.

exempelvis klassisk betingning (jämför ovan), perceptuell komplettering (se under *autokorrelation* i avsnitt 6.2 och 7.1) och begreppsbildning på empirisk grund (se vidare om *självorganiserande kartor*, avsnitt 8.2). Även enligt en optimistisk bedömning kan vi dock ännu bara förklara en försvinnande liten del av de kända minnesfenomenen med våra neurala modeller.

ANN-modeller kan syfta till förståelse av nervsystemetes arbetssätt på flera olika nivåer. Mycket detaljerade och realistiska modeller för hur enskilda nervceller bearbetar signaler – bl.a. i anslutning till LTP och liknande fenomen – har utarbetats av flera forskargrupper på senare år, medan andra forskare föredrar att arbeta på systemnivå och då oftast nöjer sig med förenklande antaganden om de enskilda elementen. ”Brain theory” och “computational neuroscience” har föreslagits som beteckning för fältet i dess helhet, och den senare termen har fått stor spridning.[15] ANN-forskning för neurobiologisk modellering går naturligtvis mer eller mindre kontinuerligt över i neurofysiologiskt, neuroanatomiskt och neuropsykologiskt teoribyggande, och exakt var gränserna skall dras är ointressant.

Det skall poängteras att biologiskt orienterad ANN-forskning inte bara handlar om inlärning och minne; artificiella neurala nätverk kan också beskriva andra aspekter av hjärnans förmåga till informationsbehandling. Ett exempel på ANN-modeller *utan inlärning* ges av de artificiella nervnät för stereoseende som vi skall beskriva senare (avsnitt 8.5). Med ANN kan man för övrigt simulera inte bara system av nervceller. Modellering av cellers (inte bara nervcellers) interna signalsystem är en annan livaktig gren av forskningen kring artificiella neurala nätverk. Relationen mellan modell och verklighet är här dock av en annan art än när ANN används som beskrivningar av verkliga nervnät; mer om detta i avsnitt 4.5.

Tekniska användningar av neurala nätverk

Modellering och simulering av biologiska strukturer och processer, särskilt nätverk av nervceller och särskilt deras förmåga till inlärning, är alltså en viktig aspekt av ANN-forskningen. En annan, och lika viktig, användning av artificiella nätverk bygger på idén att vi genom att efterlikna hjärnans sätt att arbeta skulle kunna hitta bättre metoder än de vi redan har för att lösa besvärliga problem i vetenskap och vardag. Vi

[15] För en relativt lättillgänglig översikt av ANN ur detta perspektiv, se Trappenberg (2002).

kommer att referera till tillämpningar av denna typ som ”tekniska” användningar, även om de ofta inte har med teknik i snäv mening att göra utan t.ex. rör biologi, medicin eller affärsrelationer.

Signalbearbetningen och inlärningen i ett ANN kan för varje given nätverkstyp beskrivas med ett antal matematiska formler. Ur en viss aspekt kan man därför säga att ett ANN inte är något annat än en matematisk procedur – en algoritm – för bearbetning av data. Många har slagits av idén att försöka använda just ANN-algoritmer för att lösa tekniska och andra problem som vi ställs inför. Tanken har varit att eftersom ett ANN i viss mån efterliknar hjärnans sätt att behandla signaler, så kanske det också kan lösa problem på samma grundläggande sätt som hjärnan – *fast i vissa avseenden ännu bättre*.

Den kanske mest anmärkningsvärda egenskapen hos typiska ANN är att de lär sig av sina ”erfarenheter”. Vill man till exempel sortera data i två klasser – det kan röra sig om elektrokardiogram (EKG) från friska resp. sjuka patienter – behöver man alltså inte programmera in en färdig, generell algoritm för detta i nätverket. Istället kan man mata sitt ANN med exempel på data från de två klasserna och *låta det självt hitta den regel* som bäst diskriminerar mellan dem. Utifrån denna regel kan nätverket sedan ta sig an nya exempel, dvs. generalisera till nya fall. Ett mycket stort antal sådana experiment har verkligen gjorts inom många olika applikationsområden, och försöken har inte sällan krönts med framgång.[16] Det har ofta handlat om inlärd klassifikation av komplicerade mönster (av typ fingeravtryck,[17] elektrokardiogram[18] och satellitbilder[19]), men också till exempel om förutsägelser av väder eller aktiekurser[20] på grundval av långsiktiga trender och om träningsbara system för styrning av industriprocesser och mobila robotar.[21] Tyvärr har dock en del överentusiastiska ANN-förespråkare ofta blundat för de svårigheter som metodiken innebär. Vi skall återkomma till både möjligheterna och svårigheterna i sinom tid.

---

[16] En trevlig översikt över sådana tillämpningar, med givande utblickar åt andra algoritmer än neurala nätverk, är Thomas (2003).
[17] Jämför Kung et al. (2005).
[18] Se t.ex. Olsson et al. (2006). Om medicinska applikationer av ANN se också Dybowski & Gant (utg.) (2001), Husmeier et al. (utg.) (2005) samt Malmgren et al. (utg.) (2000).
[19] T.ex. Cherkassky et al. (2006).
[20] En ny översikt här är Kamruzzaman et al. (2006).
[21] För en populär introduktion se Nordin (2003).

Lärande system och statistisk inferens

Med ökande mognad hos forskningsfältet ANN har insikten blivit klarare hos de flesta utövare, att artificiella neurala nätverk betraktade som matematiska modeller bara är en speciell variant av ett mer allmänt slag av metoder. Den mer allmänna klassen brukar man idag kalla *lärande algoritmer* (eller *lärande system*). Den innefattar alla de matematiska procedurer med vars hjälp man drar slutsatser utifrån ett antal observationer, och där slutsatserna modifieras successivt allteftersom nya exempel tillkommer. (”Machine learning” är en term med närbesläktad innebörd.) Begreppet lärande algoritmer innefattar många etablerade statistiska metoder för *inferens*, dvs. slutsatser från enskilda data till generella samband och till förväntade, framtida data. Teorin om artificiella neurala nätverk, uppfattade som algoritmer för inferens, är helt enkelt en del av vetenskapen statistik.[22] En annan sak är att den något överdrivna entusiasmen för ANN-metoder under 1980- och 90-talen (plus en god portion konservatism hos en del statistiker) ledde till att många statistiker av facket fortfarande hyser skepsis mot ANN. I ljuset av hur ANN-forskningen ser ut idag finns det dock knappast längre anledning till någon sådan misstro.

Syntetiska neurala nätverk och neuromorfa datorer

Ytterligare en forskningslinje inom ANN-fältet i stort skall nämnas, nämligen vad man lämpligen kallar *syntetiska neurala nätverk.* Här handlar det om att bygga fysiska modeller av biologiska nätverk. Detta slag av verksamhet kan också vara motiverat av en önskan att förstå nervsystemet, men ett ännu viktigare motiv är den framtida möjligheten att ersätta ”trasiga” delar i det mänskliga nervsystemet med syntetiska komponenter.[23] Ett annat forskningsområde där man också ofta arbetar på den fysiska nivån är *neuromorfa datorer*. Här designar man datorer, och konstruerar hårdvarukomponenter för dem, med en sidoblick på nervsystemets uppbyggnad. En neuromorf dator har ofta ett stort antal processorer som arbetar parallellt. En tänkbar användning för neuromorfa datorer är givetvis som ”artificiella hjärnor” i mer eller mindre människolika robotar.

---

[22] Bishop (1996) är en mycket gedigen och samtidigt pedagogisk genomgång av den sannolikhetsteoretiska basen för inferens med artificiella neurala nätverk och närbesläktade metoder. En alldeles ny lärobok av samme författare är Bishop (2006). Mycket värdefullt material finns också i Husmeier et al. (2005).

[23] Jämför t.ex. Wessberg & Nicolelis (2004), Berger & Glanzman (utg.) (2005) samt Lebedev & Nicolelis (2006).

Inget av de två fält som nämnts i det sista stycket behandlas närmare i denna bok – men de förutsätter båda den teori som kommer att presenteras. För att läsaren ska bli mogen att ta del av denna teori ska vi nu (avsnitt 1.2–1.4) diskutera några centrala filosofiskt-begreppsliga problem och introducera några grundläggande sätt att beskriva komplexa system. Sedan (kapitel 2–3) tar vi en rundtur i relevanta delar av den psykologiska och biologiska verkligheten och presenterar några allmänna tankemodeller för att beskriva inlärningsfenomen. Först därefter, från och med kapitel 4, kommer ANN-teorin att presenteras i detalj.

1.2 Vad är egentligen minne?

Kognitivistiska och icke-kognitivistiska förklaringar

I inlärningsteori, liksom i beteendevetenskapliga sammanhang i övrigt, möter man flera olika huvudtyper av förklaringar. En första typ kan vi kalla den *kognitivistiska*. I dylika förklaringar hänvisas till kognitiva tillstånd och processer som exempelvis kunskaper, tänkande, målföreställningar, trosföreställningar, slutledningar, sammanvägning av evidens och rationella beslut. Man möter dem oftast i kognitivt-psykologisk litteratur men också t.ex. i äldre texter om perceptionspsykologi eller djurexperimentell inlärningspsykologi. Den andra huvudtypen, som i mycket är den förstas motsats, formulerar istället hypoteserna i ett biologiskt språk – biokemiskt, neurofysiologiskt och/eller neuroanatomiskt – som inte innehåller några kognitiva termer. Vi kan kalla denna förklaringstyp för den *icke-kognitivistiska*, och den föredras av de flesta neurovetenskapliga forskare. Vissa termer som är vanliga i både kognitions- och neurovetenskap har en oklar status; det gäller särskilt ”representation” och ”information”. Mer om dessa i nästa avsnitt (1.3).

Det pågår sedan länge en filosofisk diskussion om relationen mellan kognitivistiska och icke-kognitivistiska förklaringstyper. Denna diskussion beror inte minst på den problematik som ska behandlas i nästa avsnitt, nämligen att det är osäkert om, och i så fall hur, begreppet *mental representation* kan översättas till icke-kognitivistiska begrepp. Om det visar sig att det finns en systematisk översättning mellan de två begreppsområdena, eller åtminstone en systematisk empirisk motsvarighet mellan dem, så kommer kognitivistiska och icke-kognitivistiska förklaringar att kunna inleda en fredlig samexistens. Som det nu är råder istället en

antagonism mellan företrädare för det ena och det andra sättet att förklara psykiska funktioner. Man använder ibland termen ”det subsymboliska paradigmet” för uppfattningen att många grundläggande förklaringar i beteendevetenskaperna måste vara av icke-kognitivistisk natur. Flera ANN-teoretiker har tagit ställning för detta subsymboliska paradigm. Just nu ska vi inte titta på de generella argumenten i denna stridsfråga, utan istället försöka konkretisera problemet genom en undersökning av begreppet *minne*.

Minne som kunskap om det förflutna

Resten av detta avsnitt kommer att ägnas frågan om det går att hitta en övergripande definition av begreppet *minne*.[24] Om vi tänker på alla möjliga former av inlärning och minne som finns, kan vi då hitta något gemensamt drag? Vad har episodiskt och semantiskt minne att göra med t.ex. klassisk betingning?

Ett – inte särskilt bra – kriterium för vad som skall kallas ”minne” tar fasta på att minnet på något sätt har med det förflutna att göra. Kanske minne helt enkelt är *kunskap som refererar till det förflutna?* Detta kriterium tycks stämma såväl för ett episodiskt minne av en madeleinekakas smak som för vår kunskap om vilket år mordet på Martin Luther King begicks, och likaså för hundens kunskap att ljudet från ringklockan hittills har följts av mat. Men det finns massor av motexempel. Till exempel minns de flesta av oss att två plus två är fyra, många av oss kommer ihåg att nästa totala solförmörkelse i Sverige äger rum år 2021, hunden har lärt sig att *nästa* ringsignal sannolikt kommer att följas av mat, och vanligtvis (men tyvärr inte alltid) kommer vi ihåg när vi går från jobbet att vi ska handla på hemvägen. I ingetdera fallet handlar minnet om något förflutet.

Ett annat sätt att nalkas frågan om minnet på något väsentligt sätt har att göra med det förflutna är att fråga sig vilken biologisk funktion det kan ha. Svaret blir rimligen att minnesmekanismer har utvecklats för att vi (och andra djur) ska kunna lära oss av erfarenheten att hantera saker bättre *i framtiden*. Detta är rätt uppenbart i fallet klassisk betingning (det är en fördel att veta i förväg att maten ska komma) men gäller också t.ex. episodiskt minne. Den biologiska bakgrunden till att vi har förmågan till livliga minnen av episoder vi varit med om är säkert inte att moder Natur

[24] Författaren förutsätter att en sådan definition måste vara kopplad på ett uppenbart sätt till definitioner av begrepp som *inlärning* och *komma ihåg*, och för därför ibland diskussionen i termer av de senare begreppen.

i sin generositet velat utrusta oss med en inbyggd video, så att vi kan njuta av gamla filmer från det förflutna. Ändamålet är istället att vi med hjälp av dessa minnen ska kunna hantera liknande episoder bättre i framtiden. Att det episodiska minnet *också* har ett visst underhållningsvärde och kan tjäna som tillflykt under grå dagar gör inte denna analys mindre sann.

Minne som erfarenhetsbaserad kunskap

Vi har ännu inte fått något svar på hur begreppet minne skall avgränsas. Ett bättre förslag än det vi nyss förkastade är att inlärning är *erfarenhetsbaserat förvärvande av kunskap* och att minne är *bibehållen sådan kunskap.*[25] Ett minne är helt enkelt en kunskap som vi förvärvade genom vår erfarenhet tidigare i livet, och som alltjämt består. Aspekten ”förflutet” kommer nu in endast genom att kunskapen har ett *ursprung* tidigare i livet. Denna definition är alltså neutral i fråga om vilken tidpunkt minnet *refererar till,* och drabbas därför inte av våra invändningar ovan. Den stämmer bra på de exempel vi tagit upp och även t.ex. på fall av procedurellt minne. Att jag kommer ihåg hur man simmar betyder enligt definitionen att jag en gång i livet genom mina sura (våta) erfarenheter förvärvade kunskap om hur man simmar, och att jag har kvar denna kunskap.

Det finns dock stora problem med denna nya definition. För det första är det inte självklart att den täcker sådana former av inlärning och minne hos djur, som vi huvudsakligen känner genom deras manifestationer i det yttre beteendet. Är det verkligen så att en råtta som undviker att trampa på en viss pedal, efter det att denna tidigare alltid har åsamkat råttan smärta genom att utlösa en elektrisk stöt, gör det därför att den *vet att* (eller *tror att*) det kommer att göra ont om den trycker på pedalen igen? Kanske är operant inlärning bara fråga om ett förvärvande av ett automatiskt reaktionssätt, som inte har någon motsvarighet i kognitiva processer? Och vad *vet* en amöba som habituerats till en repetitiv stimulus? Problemet är inte att råttan inte kan vara helt *säker* i sin tro på hur framtiden kommer att te sig, och inte att *vi* inte kan veta vad amöban vet. Hudvudsvårigheten är att det är oklart vad det *innebär* att tillskriva kognitiva tillstånd som ”kunskap” och ”tro” till en råtta eller en amöba.

För det andra är det tveksamt om någon definition som hänvisar till kunskapsbegreppet stämmer ens på alla minnesformer som återfinns hos oss

[25] Se Holland (1974).

människor. Perceptuellt minne, till exempel, yttrar sig som förändrad varseblivning, men behöver detta innebära *kunskap*? Tänk på det fall då man just hittat ett nytt sätt att se en flertydig figur. Och när det gäller habituering är det tveksamt om denna innebär ett förvärvande av kunskap ens när den förekommer hos människor (exempelvis när man vänjer sig vid ett bakgrundsljud).

En icke-kognitiv definition av minne

Ställd inför dessa svåra frågor frestas man gärna att ta till en helt annan sorts definition av inlärning och minne. Enligt denna är inlärning ett *modifierande, genom individens erfarenhet, av organismens respons (eller responstendens) i en återkommande situation.* En alternativ, kort formulering är att ett minne är en *historikberoende responstendens*. Habituering och perceptuell inlärning är exempel på inlärning i denna betydelse – ett förvärvande av specifika responstendenser. *Genom individens erfarenhet* är väsentligt i sammanhanget, eftersom responser som tillkommit genom naturlig selektion inte bör räknas som inlärda – såvida man inte, mer metaforiskt, vill karakterisera dem som utslag av ”artens minne”. Likaså utesluter frasen ”genom individens erfarenhet” de fall, där en förändring i hjärnan som direkt orsakats av en hjärnskada omedelbart leder till nya responsmönster. Inte heller då vill vi förstås säga att de nya, under livet förvärvade beteendena är ”inlärda”.

Baksidan med en sådan ”icke-kognitiv” definition av inlärning och minne är förstås att den inte på något uppenbart sätt fångar den kognitiva dimensionen av mänskligt minne. Läsaren kanske till och med tycker att detta är ett understatement: ser man minne som en förvärvad responstendens kan man väl *överhuvudtaget* inte beskriva kunskapen *att* Martin Luther King dog 1968 som ett minne? Deklarativt, semantiskt minne är väl ett inre psykiskt tillstånd med ett slags språklig struktur, och då kan det inte samtidigt *vara* en responstendens – kan man tycka.

Många filosofer har genom att erbjuda en s.k. ”naturalistisk” analys av kognitiva termer försökt visa att ett minne visst kan vara både ett kognitivt tillstånd och en responstendens. Sådana teorier kommer att beröras i nästa avsnitt. En tänkbar, alternativ lösning på problematiken är att tolka om ordet ”responstendens” i vår föreslagna definition av begreppet minne på ett så liberalt sätt att det får innefatta vilket slags tillstånd som helst (kognitivt eller inte), bara det är någorlunda varaktigt. Att en organism lär sig något skulle då innebära *att den genom en erfarenhet försätts i ett*

nytt tillstånd, som varar utöver själva sinnesintrycket. Exempel på sådana tillstånd kan vara den minskade responstendensen hos amöbor vid habituering, men också det hjärntillstånd som ligger bakom din tro att Martin Luther King mördades 1968. På så vis får vi en definition som skulle kunna omfatta inlärning hos såväl människor som så kallade ”lägre” djur.

Ett problem med alla försök att definiera minne i termer av resultat av erfarenhet eller sinnesintryck är att inte alla bestående resultat av sinnesintryck är att räkna som minnen – exempelvis är det väl inte så ovanligt, att en chockartad upplevelse kan få oss att *glömma* andra saker som hände den dagen. Det verkar alltså som om det, åtminstone för vissa minnesformer, skulle behövas något mer än en rent *kausal* förbindelse mellan en erfarenhet och ett bestående tillstånd för att det senare skall kunna räknas som ett minne. Och vad är naturligare än att anta att det där extra som behövs är av kognitiv art...

Med denna fundering måste vi lämna frågan om en generell definition av minne och inlärning. Att vi nu ger upp försöket att formulera en sådan definition betyder inte att försöket har varit meningslöst: vi har ju kunnat peka på såväl viktiga likheter som viktiga skillnader mellan olika minnesformer. På så vis har vi trots allt fått en bättre förståelse för det omfattande fält som vi i vanligt språkbruk sammanfattar med våra termer *minne* och *inlärning*.

Vi ska nu anställa några reflektioner över en uppsättning närbesläktade filosofiska problem, som är av hög relevans när vi vill modellera kognitiva processer med hjälp av artificiella neurala nätverk, eller överhuvudtaget knyta psykologiska begrepp till neurofysiologiska. Det handlar om begreppen *mental representation*, *neural representation*, *information* och *informationsbehandling*. Avsnittet avslutas med en diskussion om tänkande datorer och om användningen av datorn som analogi för att beskriva mänskliga, psykiska funktioner.

1.3 Representation och information

Intentionalitet

Detta avsnitt handlar också om relationen mellan ”kognitiva” och ”icke-kognitiva” beskrivningar av organismer, eller som filosofer ofta uttrycker

saken, mellan *intentionala* och *icke-intentionala* beskrivningar. I psykologin såväl som i vardagen talar vi om att en människa *tror* någonting, att hon *minns* någonting, att hon *hoppas* någonting, att hon *tänker på* någonting. Alla dessa beskrivningar handlar om kognitiva eller intentionala tillstånd, där personen ifråga har ett slags mental *inriktning mot ett objekt*. Detta *intentionala objekt* kan vara av varjehanda slag: när man minns att två plus två är fyra är objektet ett matematiskt faktum, när man tror att Martin Luther King mördades 1967 är det ett historiskt ickefaktum, och när man hoppas få en cykel på sin födelsedag är objektet helt enkelt en önskad födelsedagspresent som kanske blir verklighet, kanske inte. Det intentionala objektet behöver tydligen inte existera.

Ett klassiskt filosofiskt sätt att tala om dylika kognitiva tillstånd är alltså i termer av *intentionalitet*; i dagens kognitiva psykologi använder man hellre begreppet *mental representation*. När jag hoppas få en cykel, representerar jag mentalt en cykel. Som vi ska se nedan har termen ”mental representation” flera olika möjliga betydelser, men tills vidare använder vi den helt enkelt som en synonym till ”intentionalitet”. I denna användning är det okontroversiellt att mental representation existerar. ”Föreställa sig” och ”föreställning” är andra, mer vardagliga termer, som ofta är användbara för att beskriva intentionala fenomen.

Mental representation genom inre bilder?

Diskussionen om intentionalitetens/den mentala representationens karaktär har varit intensiv i både psykologin och filosofin i över ett sekel, men vi måste tyvärr gå förbi det mesta av denna debatt.[26] Låt oss dock komma ihåg att de gamla empiristiska filosoferna, t.ex. Hume, ofta utgick från att alla föreställningar uppstår ur varseblivningar och därför liknar dessa – ”enkla idéer är kopior av impressioner”.[27] Denna tanke utvecklades ibland som att en föreställning om ett objekt alltid är en *inre, mental bild* av sitt objekt och *representerar det genom konkret likhet*. Både psykologin och filosofin gjorde runt förra sekelskiftet upp med såväl Humes grundtanke som den sistnämnda preciseringen av den. Inte så att man sedan dess har förnekat att det finns ett bildligt, *åskådligt* tänkande. Det hör till de allra flesta människors vardagserfarenheter att de ibland ”tänker bildmässigt”. Däremot är det viktigt att göra två påpekanden, menar de flesta i dag. För det första finns det också *oåskådliga*

[26] För översikter över den problematik som diskuteras i detta avsnitt se Crane (2004), och för en mer avancerad framställning t.ex. Clapin et al. (utg.) (2004).
[27] Hume lämnade också plats för komplexa idéer, sammansatta av enkla idéer.

föreställningar, ett icke-bildligt tänkande.[28] Icke åskådligt tänkande är till exempel regeln vid normal, vardaglig språkförståelse. Åskådliga moment kan visserligen vara inblandade vid språkförståelse, men man hinner knappast med att för sitt inre bildligt illustrera *allt* det som man faktiskt lägger in i en skriven eller talad text. Man ska inte låta sig förvillas av att det kan förekomma ett åskådligt uppfattande av *orden,* när man ”tänker i ord”. Det intressanta är förstås om man åskådligt uppfattar *det man tänker på*.

För det andra har många kommit till slutsatsen att bildligt tänkande inte kan vara en inre analog till varseblivandet av en vanlig, yttre bild (t.ex. ett fotografi). Det finns bildligt tänkande, men inga inre bilder – och även om det funnes sådana, skulle de inte utgöra en *tillräcklig* förklaring av fenomenet mental representation. Varför inte det? Ett argument som härstammar redan från Humes samtid är påpekandet att våra föreställningar ofta har ett *generellt* innehåll, medan bilder är konkreta och specifika. En färgad bild måste exempelvis alltid ha en *specifik* färg (eller flera specifika färger). Den är exakt lik andra föremål med samma specifika färg(er), och om representation enbart bygger på likhet så borde det vara denna specifika färg (dessa specifika färger) som representeras. Det är svårt att se hur mer allmänna karakteristika som till exempel *rödhet* eller *att vara färgad* alls kan representeras.[29]

Ett annat argument har senare framförts bland annat av den kände filosofen Wittgenstein.[30] Antag att bildligt tänkande består i att vi uppfattar en inre bild. Genom den uppfattar vi ett annat objekt – nämligen det som vi tänker på. Men hur går detta till? Exakt likhet mellan bilden och objektet duger, har vi just sett, inte som förklaring. Vi ska också komma ihåg att en bild kan föreställa olika saker, beroende på hur den tolkas. För att vi ska uppfatta ett *visst* objekt genom den inre bilden, måste alltså den inre bilden tolkas på ett bestämt sätt (av flera möjliga). Hur går denna tolkning till? Tolkning är ju också ett intentionalt fenomen. Måste vi inte anta *ytterligare* en inre bild, som bestämmer hur den första bilden ska tolkas? I så fall måste också denna andra bild tolkas – etcetera. Detta är vad man i filosofin kallar en ”oändlig regress”, och sådana bör givetvis undvikas.

---

[28] Denna uppfattning fick sin definitiva plats i filosofin genom fenomenologins portalfigur Edmund Husserl – se särskilt Husserl (1900-01) – och fick samtidigt stöd i experimentalpsykologin genom forskningen inom den s.k. Würzburgskolan; se Mandler & Mandler (1964) kap. 4.

[29] Detta problem formulerades på ett pregnant sätt redan av 1700-talsfilosofen Berkeley.

[30] Jfr. Wittgenstein (1953), §§ 139 ff.

Dessa nya insikter gav under 1970-talet upphov till ett förnyat intresse (särskilt i filosofin) för hur det i detalj går till när vi mentalt representerar ett objekt, vare sig detta sker åskådligt eller ej. En populär modern teori postulerar att intentionala processer bygger på ett slags inre symboler, som är analoga med språkliga symboler snarare än med bilder. Man talar om ett ”language of thought”, som även skall förklara vår förståelse av naturliga språk.[31] I den mån denna teori innebär att man tänker, förstår etc. *genom att uppfatta* dessa inre symboler, så hotas även den av det ovan nämnda regressargumentet. Språkliga tecken behöver ju tolkas, och det borde gälla även symbolerna i *language of thought*. Hur går tolkningen av dessa symboler till – fordrar den att man uppfattar fler symboler?

Flera olika teorier antar alltså att mental representation går till så, att man uppfattar tankens objekt *via ett uppfattande av inre bilder eller symboler*. Inte sällan använder man termen *mentala representationer* just om sådana inre bilder eller symboler. Med ”representationalism” menar man då tesen att all intentionalitet kräver mentala representationer av detta slag. Regressargumentet tycks vara ett hot mot alla sådana teorier. Långt ifrån alla tänkare har dock accepterat detta arguments giltighet, och representationalismen är fortfarande en livskraftig åskådning, särskilt då den just nämnda ”språkliga” versionen.

En tanke som kan tyckas ge stöd åt representationalismen är följande. Mentalt representerande finns, det är alla eniga om. Men detta representerande kan inte komma från ingenstans. Det måste finnas *någonting* i huvudet som gör att vi mentalt representerar ett objekt snarare än ett annat! – Ja, så är det säkert. Varje gång vi tänker på något, föreställer oss något eller varseblir något så har detta säkerligen sin grund i ett specifikt hjärntillstånd. Vi kan kalla detta tillstånd för ”representationens bas”. Men för att vi ska kunna ta existensen av en sådan bas till intäkt för att representationalismen är riktig, måste vi anta att representationsbasen har karaktären av en bild eller symbol som vi faktiskt *uppfattar och tolkar*. Och det är långt ifrån självklart att så är fallet.

Till förvirringen bidrar att termen ”mental representation” inte sällan används om det som vi just kallat representationens bas. Termen har alltså tre vanliga betydelser. För det första kan den stå för ett intentionalt fenomen, vilket som helst. Det är i den innebörden som det är självklart att mental representation finns. För det andra används den om en neural bas

[31] Fodor (1975).

för det intentionala fenomenet – en sådan torde också finnas. För det tredje kan den syfta på en speciell, hypotetisk företeelse (den inre bilden eller symbolen) som av vissa teorier (nämligen de representationalistiska) postuleras som förklaring av mental representation i den första betydelsen. I det följande kommer vi, om inget annat sägs, att använda termen (och från den härledda termer som t.ex. ”mentalt representerande”) på ett teorineutralt sätt, alltså i den första innebörden.

Representation som simulering eller re-representation

Någon konsensus i frågan om det åskådliga tänkandets natur, eller om mental representation överhuvud, har ännu inte uppnåtts, vare sig i filosofin eller i psykologin. En inte helt ovanlig uppfattning, som författaren också förespråkar, är att mental representation, och då i första hand bildligt tänkande, är en *simulering* av varseblivningen och har samma funktionella egenskaper som denna.[32] Grundtanken kan också formuleras som att åskådligt tänkande är ett *substitut* för varseblivning i de situationer när varseblivning inte är möjlig. Typexemplet är då man navigerar i ett mörkt sovrum efter att ha haft lampan tänd en mycket kort stund. Den åskådliga rumsföreställningen fungerar under flera sekunder nästan lika bra som den visuella varseblivningen som grund för att kunna undvika hinder och hitta dörrhandtaget. Istället för en konkret likhet mellan en inre bild och ett yttre objekt är det här fråga om en funktionell likhet mellan två mentala processer, tänkande och varseblivning. Simuleringsteorin är oftast kopplad till tanken att åskådliga föreställningar åtminstone ibland har sitt upphov i tidigare, motsvarande perceptioner, och en alternativ beteckning för den är därför *representation som re-presentation.*[33]

Simuleringsteorin implicerar att frågan om hur objekt representeras genom åskådliga föreställningar inte torde få något svar förrän vi vet hur det går till när samma objekt representeras i varseblivningen. Det senare är också en filosofisk och kognitivt-psykologisk stridsfråga av rang. Dessutom är det inte självklart att samma slags lösning gäller för icke åskådlig representation som för åskådlig dito... för tankar är väl inte särskilt lika varseblivningar? Författaren menar dock att simuleringsteorin kan generaliseras till all mental representation. Den likhet mellan tanke och varseblivning som teorin primärt postulerar är ju en *funktionell*

[32] Se Malmgren (1991, 1996, 2006) och Hesslow (2002).
[33] Det är t.o.m. möjligt att det historiskt sett är rimligt att tolka Hume som simulationsteoretiker, men eftersom detta inte är en idéhistorisk framställning ska vi inte driva den frågan vidare.

likhet, inte en bildlikhet. Dessutom bör man komma ihåg, att en tanke kan omvandlas i en åskådlig föreställning. Med andra ord, även om kanske inte varje representation av något man varseblivit förtjänar att kallas en *re-presentation* av det varseblivna, så kan den i varje fall *ge upphov till* en sådan re-presentation.

Vi kan inte fördjupa oss mer i denna filosofiska fråga just nu. Jämför dock med den modell som presenteras i slutet av avsnittet om klassisk betingning (2.3) samt resonemanget om system med lärande kontinuerliga attraktorer (avsnitt 7.3).

Kan nervceller representera verkligheten?

Det är alltså rätt svårt att förstå på ett djupare plan vad mental representation är, hur välbekanta vi än är med fenomenet. Det är inte heller självklart vilken mening man skall ge åt termen ”*neural* representation”. Många neurala modeller för psykologiska processer förutsätter dock att vi har ett sådant begrepp. När man t.ex. modellerar mönsterklassifikation i mänsklig varseblivning med ett artificiellt neuralt nätverk (för detaljer i sådana modeller se nedan), så tänker man sig vanligen att insignalen till nätverket representerar mönstret som skall klassificeras, medan utsignalen innebär en klassificering av mönstret (*som* t.ex. en kvadrat). Vid tekniska tillämpningar av ANN-modeller är ett sådant tänkesätt oproblematiskt – användaren av modellen kan låta signalerna stå för vad han/hon behagar. Men hur ska detta tänkande appliceras på ett biologiskt nätverk? I synnerhet, vad skall det betyda att utsignalen från ett neuralt nätverk i hjärnan innebär en klassificering, och alltså *representerar* en viss klass (i vårt exempel: klassen av kvadrater)?

Ett tänkbart svar är att den neurala utsignalen är vad vi ovan kallat den mentala representationens ”bas”, i det aktuella fallet alltså det neurala underlaget för ett mentalt representerande av klassen av kvadrater. Att ett sådant underlag existerar torde de flesta vara överens om. På samma sätt skulle insignalen kunna representera det mönster som skall klassificeras genom att signalen är det neurala underlaget för ett mental representerande av mönstret ifråga. Det svaret innebär att man följer strategin att definiera neural representation genom mental sådan.

Bortsett från det inte obetydliga problem som uppstår genom att man kanske inte riktigt kan förklara vad mentalt representerande är (jämför ovan), medför denna strategi bekymmer i de fall när det inte är självklart

att det *finns* något mentalt korrelat. Ett exempel ges av de välbekanta analyser av celler i det visuella systemet som ledde till ett Nobelpris för ett antal år sedan.[34] Vissa celler i detta system, så kallade *off center–on surround-celler* har en maximal respons när ljusmönster av ett visst slag (mörker omgivet av en zon av ljus) projiceras på näthinnan. Andra, så kallade *komplexa* celler svarar maximalt på t.ex. linjesegment med en viss orientering. Man läser inte sällan att responserna hos dessa typer av celler *representerar* respektive typer av mönster. Men det är knappast rimligt att anta att det finns någon korresponderande *mental* representation av mönstren i alla dessa fall, eftersom vissa av de relevanta cellerna återfinns mycket tidigt i kedjan från näthinna till synbark. Innebörden i termen ”neural representation” måste alltså förklaras på något annat sätt.

Kausala representationsteorier och felrepresentationsproblemet

Det har inte sällan föreslagits, att redan det faktum att cellerna ger sitt maximala svar på vissa stimuli innebär att de representerar just dessa stimuli. Detta förslag är ett slags *kausal* teori för neural representation. En formulering av en närliggande, mycket enkel variant av sådana kausala teorier är: en händelse i nervsystemet representerar en stimulus, om och endast om händelsen *orsakas* av ifrågavarande stimulus. En något mer sofistikerad variant säger att ett neuralt tillstånd *x* representerar en stimulus S, om och endast om S är en *nödvändig kausal betingelse* för *x.* I detta fall är ju förekomsten av *x* i viss mening ett ”tecken på” förekomsten av S. Filosofer talar gärna om ”naturliga tecken”, och ger som andra exempel sambandet mellan rök och eld, eller mellan ett visst slags spår i snön och en björns besök på platsen. Maximal respons i en viss komplex cell kan ses som ett naturligt tecken på förekomsten av ett linjesegment med viss orientering och därför som representerande detta, i den sofistikerade kausala meningen av ”representation”.

Genom att använda sig av någon sådan kausal teori kommer man undan den neurala representationens beroende av den mentala. Kanske man till och med skulle kunna definiera vad mental representation är i termer av kausal representation hos de underliggande neurala processerna, dvs. återföra mental representation på neural sådan, istället för tvärtom?

Tyvärr strandar många versioner av sistnämnda förslag på vad man kan kalla ”felrepresenterandets problem”. Det är nämligen utmärkande för

[34] Hubel & Wiesel (1965).

mentalt representerande att det kan vara antingen riktigt eller felaktigt. (Detta är egentligen bara ett annat sätt att påpeka att det man tänker på, eller tror sig se, inte alltid behöver finnas.) Om du t.ex. tittar på en tiger i dåligt ljus och tycker dig se ett lejon, så har du representerat omgivningen felaktigt: genom din varseblivning representerar du ett lejon, medan du *borde* representera en tiger. Men den enkla kausala teorin för mental representation säger att eftersom varseblivningen är *orsakad* av en tiger, så *är* den en representation av en tiger. Den enkla kausala teorin är med andra ord inte förenlig med att man ser fel. Av liknande skäl är den inte förenlig med att man *tänker* fel, eller *minns* fel. Man kan ju som bekant tänka sig att det finns ett lejon i rummet utan att denna tanke orsakats av ett lejon i rummet, men teorin implicerar att det är omöjligt.

Det har föreslagits en modifikation av den sofistikerade teorin som klarar av de senare slagen av ”felrepresentation”. Man säger då att x är en representation av S, om och endast om S *i varseblivningssituationer* är ett nödvändigt villkor för x. Men denna modifikation är inte förenlig med att det finns felperception. Om du tycker dig se ett lejon när du faktiskt tittar på en tiger, så är tittande på ett lejon inte ett nödvändigt kausalt villkor för att ha den varseblivning du har – det kan därför, enligt teorin, inte vara ett lejon som du representerar i varseblivningen.

För att komma förbi detta problem – som alltså har en stor räckvidd också utanför varseblivningens område – har filosofer föreslagit andra, mer avancerade varianter av den kausala teorin för mental representation.[35] Vi kan tyvärr inte fördjupa oss i dem här utan får nöja oss med slutsatsen att inget av de kausala representationsbegrepp som vi introducerat torde ha alla de egenskaper som är önskvärda hos ett rimligt begrepp *mental* representation. Därmed inte sagt att de inte skulle vara användbara som preciseringar av begreppet *neural* representation i neurofysiologin och i ANN-modeller av biologiska nätverk. Vi ska nu titta närmare på denna möjlighet, och då ta omvägen via en diskussion av begreppet *information*. Spänningen mellan begreppen *mental* och *kausal* representation har nämligen en parallell i en spänning mellan vad man har kallat det ”semantiska” och det ”tekniska” informationsbegreppet.

[35] Ämnet är hett i modern kognitionsfilosofi och det finns en stor litteratur. Nämnas bör kanske framförallt Fred Dretske; se t.ex. Dretske (1981). Det ska också poängteras att Wittgenstein lägger stor vikt vid felrepresentationsproblemet i Wittgenstein (1953); se t.ex. § 51.

Det semantiska informationsbegreppet och de tekniska

Psykologer beskriver ofta kognitiva processer i termer av *informationsbehandling;* likaså sägs det inte sällan att nervsystemet ”behandlar information”. Vad menas med ”information”?[36] I en betydelse, ibland kallad den ”semantiska”, är termen nära relaterad till det vida begreppet mental representation. Det är denna betydelse av ”information” som är involverad när vi till vardags säger att vi fått information om det ena eller det andra, eller att vi söker information på Internet. I denna mening kan man tala om *riktig* kontra *felaktig* information.

Men det finns andra möjliga preciseringar av termen. Det begrepp information som *informationsteorin* använder sig av är ett matematiskt mått på *osäkerhet*, definierad i termer av sannolikheterna i en viss situation för att olika alternativa händelser ska inträffa.[37] Ju mer jämnt fördelade sannolikheterna är för de alternativa händelserna, desto mer information (också kallad ”entropi”) innehåller situationen. *Relativ* eller *betingad* information kan definieras som den *reduktion* av osäkerheten som uppstår, när situationen specificeras genom något ytterligare villkor, t.ex. att någon av de tidigare möjliga händelserna inte kan inträffa. Man måste här komma ihåg att ”osäkerhet” har en strikt sannolikhetsteoretisk definition, som inte nödvändigtvis har med (bristande) kunskap eller andra intentionala tillstånd att göra. Informationsbegreppet har t.ex. en viktig tillämpning i den statistiska mekaniken. Men förvisso kan det också tillämpas på processer som har mycket att göra med information i den semantiska meningen.[38]

Det tekniska begreppet betingad information är nära besläktat med den mer sofistikerade kausala teorin för representation, som vi nämnde ovan. För båda är begreppet *betingad sannolikhet* centralt, dvs. hur sannolik en händelse A är *givet* att en annan händelse B inträffar. Närmare bestämt kan förhållandet att A är ett nödvändigt villkor för B skrivas om som att den betingade sannolikheten för A, givet B, är = 1. Om det i detta fall från början råder en stor osäkerhet om A, och B inträffar, så reduceras osäkerheten om A så mycket som det överhuvudtaget är möjligt. Det

[36] Jfr. för detta avsnitt gärna Sayre (1976), (1986) samt Adams (2003).
[37] Shannon & Weaver (1949).
[38] Lägg också märke till att det som vi ur en vardagsspråklig synvinkel kanske skulle vara mest benägna att kalla information, det vill säga en reduktion av osäkerheten, i informationsteorin beskrivs som en *minskning* av den absoluta informationsmängden. Men detta är bara en matematisk konvention, som man inte behöver fästa någon principiell vikt vid.

finns alltså anledning att räkna det sofistikerade kausala representationsbegreppet som en variant av det tekniska begreppet information.

Informationsbehandling i nervsystemet?

Många moderna framställningar poängterar att hjärnan och nervsystemet *behandlar information*. Det går att ge preciseringar av detta påstående i termer av de tekniska informationsbegreppen. Här är några exempel på hur man kan beskriva hjärnan ur ett informationsteoretiskt perspektiv:

De varierande stimuli som kommer från omvärlden leder till en variation i input till sinnesorganen, som i sin tur åstadkommer en variation hos tillstånden i hjärnan. Beroende på vilka stimulusvillkor som är nödvändiga för att åstadkomma förändringarna i hjärnans tillstånd kommer dessa förändringar att i större eller mindre utsträckning återspegla variationen i omvärlden. Denna återspeglingsrelation kan definieras i termer av det sofistikerade kausala representationsbegreppet. Mer generellt kan informationsflödet från yttervärld till hjärna beskrivas i termer av betingade sannolikheter för yttervärldstillstånd, givet hjärntillstånd, och som den reduktion av osäkerhet om yttervärldens tillstånd som impliceras av att vissa hjärntillstånd föreligger.

Begrepp som *kodning* och *omkodning* kan också ges preciseringar i termer av kausala relationer mellan olika uppsättningar av sådana speglande tillstånd i hjärnan. Det kanske till exempel är så, att variationen i sådana händelser som vi alldeles nyss var med om återspeglas i skilda *aktivitetsnivåer* hos vissa nervceller, men att de vid en senare tidpunkt kommer att speglas i skilda egenskaper hos *förbindelser* mellan andra nervceller. Vi kan då säga att en ”omkodning” av information ägt rum mellan det omedelbara minnet och långtidsminnet – utan att implicera att det som är ”kodat” är ett budskap med semantiskt innehåll.

Vi ska närmast tillämpa dessa tankar på några fenomen av särskild relevans för neural nätverksteori.

Distribuerad och integrerad representation; redundans

Om en nervcell har flera utåtgående förbindelser med andra neuron, kommer den i princip att kunna påverka aktiviteten i alla dessa neuron. Antag att en viss variation i organismens omgivning speglas på ett enkelt

sätt av tillståndet i den förstnämnda cellen. Men i nästa processteg återspeglas samma variation bara av de *samlade* aktiviteterna i *flera* neuron. Enkelt uttryckt har bara delar av signalen nått de olika neuronen, men delarna är inbördes olika och uttömmer tillsammans den ursprungliga signalen. (Som att skicka ett meddelande genom att dela upp det på flera olika brev.) Vi har här ett exempel på *distribuerad representation*, ett fenomen som är av stort intresse för ANN-teorin. Observera att det tekniska begreppet ”distribuerad representation” bara syftar på det sätt som en aktivitetsnivå i en viss enhet återspeglas i andra enheter som den står i kausal förbindelse med. Om fenomenet, i ett givet fall, har något som helst att göra med ”representation” i betydelsen *intentionalitet*, och i så fall vad, är därför en öppen fråga.

Det finns också mer komplexa former av distribuerad representation, som det kan vara befogat att samla under rubriken *integrerad representation*. Det är kanske enklast att förklara begreppet genom en metafor. Antag att jag ska meddela två olika telefonnummer till en god vän. Av någon outgrundlig anledning gör jag detta genom att först skicka summan av telefonnumren, och därefter skillnaden mellan dem. Min intelligenta vän förstår genast hur hon ska rekonstruera ursprungsnumren, nämligen genom att först summera budskapen och dividera med två, och sedan ta skillnaden mellan dem och dividera med två. Vad jag gjorde var, uttryckt i mer allmänna termer, att blanda flera samtidiga meddelanden på flera kanaler på ett sådant sätt att det i slutänden gick att rekonstruera meddelandena. ANN-teorin, liksom det mänskliga nervsystemet, erbjuder många möjligheter till sådan integrerad informationsöverföring. För att kunna utföra denna typ av omkodningsoperationer måste ett neuralt nätverk först divergera (skicka signaler åt olika håll) och sedan konvergera (integrera signaler från olika källor) på ett sådant sätt att ingen informationsförlust uppstår. Andra, liknande operationer innebär ett större eller mindre mått av informationsförlust. Se vidare nedan om koordinattransformationer och dimensionsreduktion, avsnitt 4.5.

Ett annat viktigt fenomen är *redundans*, som också kan uppstå om en nervcell har flera utåtgående förbindelser. I kognitivt-metaforiska termer kan man förklara redundans som att samma budskap skickas via flera oberoende kanaler (flera kopior av samma brev, helt enkelt). Mer neutralt kan man säga, att hela variationen i en faktor speglas av variationen i flera andra faktorer, tagna var för sig. Vissa delar av det mänskliga centrala nervsystemet har en hög grad av redundans, vilket bidrar till att göra det tolerant mot vissa slag av störningar. Blandformer av distribuerad re-

presentation (inklusive integrerad representation) och redundans förekommer säkerligen ofta, men begreppen ska inte förväxlas.

Informationsbehandlingen i hjärnan är enligt det allmänna synsätt som här skisserats i första hand en *systematisk samvariation,* som kan ta sig olika, specifika former. Den medför att variationer i insignalerna till hjärnan kommer att återspeglas på olika sätt i olika neurala strukturer. Men har vi inte tappat bort någonting nu? Informationsbehandling i denna tekniska mening förekommer ju överallt, även i icke levande system. Exempelvis speglar positionen hos en planet, såsom Saturnus, vid en given tidpunkt alla planetens positioner under en lång tid tillbaka. Visserligen kan *vi använda* detta faktum för att skaffa oss information om planeternas tidigare positioner – men vi kallar inte solsystemet *självt* för ett ”informationsbehandlande system”. Varför karakteriserar vi då hjärnan så? Ett plausibelt svar är hjärnan tillhör en person, och att den tekniska informationsbehandlingen i hjärnan därför i sista änden *resulterar i semantisk information* – dvs. kunskaper, föreställningar och uppfattningar som denna person har. Detta svar undviker tankefelet att den information som hjärnan behandlar alltid själv *är* semantisk information. Men *hur det går till* när personen gör semantisk information av de kausala sammanhangen ingår inte i svaret.

Biologiskt-funktionell information

Det finns ett mer generellt motiv än det just föreslagna för att tala om hjärnan som ”informationsbehandlande”. Det inser man om man tittar på andra medicinska och biologiska områden än det centrala nervsystemet. Exempelvis omsätter vissa receptorer i väggen till halspulsådrorna (baroreceptorer) variationer i blodtrycket till signalvariationer i vagusnerven. Dessa signalvariationer reglerar i sin tur hjärtats frekvens och kraft så att blodtrycket inte blir för högt eller för lågt. Den här mekanismen beskriver man ofta som att vagusnerven ”skickar information om blodtrycket” från receptorerna ifråga till hjärtat. Varför gör man det, när man inte säger att Saturnus processar information om sina tidigare tillstånd? Det går inte, som när det gäller det centrala nervsystemet, att hänvisa till att semantisk information uppstår i slutänden, eftersom signalen från baroreceptorerna normalt inte ger upphov till att personen får några föreställningar eller uppfattningar om sitt blodtryck. Anledningen är snarare, att baroreceptorsystemet ingår i en biologiskt funktionell organisation. Den kausala transmissionen av variation genom detta system hjälper organismen (och släktet) att överleva, och därför kallar vi den ”informationsbearbetning”.

Ordvalet, som ju lätt leder tankarna till semantisk information, är säkerligen ett uttryck för samma antropomorfistiska tendens som gör sig gällande när vi säger att baroreceptorernas ”syfte” är att hålla blodtrycket konstant. Det må vara hur som helst med den saken; både ”information” och ”syfte” kan här preciseras i termer av organismens och släktets överlevnad.

Vi har alltså hittat ytterligare ett informationsbegrepp förutom det semantiska och det (de) tekniska. Låt oss tala om *biologiskt-funktionell* information. Författarens uppfattning är att talet om nervsystemet som informationsbehandlande i första hand bör tolkas i termer av biologisktfunktionell information. Att det visuella systemet behandlar information om ljus betyder då *mer* än att det överför variation i ljus till variationer i nervtillstånd, men innebär ändå inte att dessa tillstånd nödvändigtvis bär semantisk information. Det betyder helt enkelt att den förmedling och omkodning av variationer i stimuli som systemet står för har en överlevnadsfunktion.

Nu är det förvisso så att det centrala nervsystemets informationsprocessande, i teknisk och funktionell mening – med andra ord, dess biologiskt funktionella bruk av kausala representationer – så småningom ger upphov till en hel del semantisk information och mentala representationer hos respektive hjärnas ägare. Vi får trots allt vår kunskap om yttervärlden, och tyvärr också många av våra *falska* föreställningar om den, genom att aktivitetsmönster i periferin av nervsystemet på olika, intrikata sätt ger upphov till andra neurala aktivitetsmönster. Vilken är då den närmare relationen mellan å ena sidan teknisk och biologisk information, och å andra sidan semantisk information? Låt mig än en gång poängtera, att återförandet av semantisk information på teknisk eller biologisktfunktionell sådan inte kan ske genom att man identifierar begreppen. Att säga att hjärnans ägare *använder* informationen om sina hjärntillstånd, och gör det precis i samma mening som en forskare kan sägas ”använda” informationen om Saturnus nuvarande position för att räkna ut dess tidigare positioner, fungerar inte heller. Ingen av oss vet nämligen tillräckligt mycket om sina hjärntillstånd för att kunna räkna fram de nödvändiga kausala betingelserna för dem i form av tidigare stimuli.

En modern angreppsvinkel på hela den här problematiken representeras av den s.k. ”teleo-semantiken”.[39] Som namnet antyder, går den ut gå att försöka definiera semantisk information (och mental representation) i ter-

[39] Millikan (1987). För en aktuell doktorsavhandling i ämnet, se Almér (2007).

mer av vad vi här har kallat ”biologiskt-funktionell information”. Men vi kan inte fördjupa oss mer i frågan här.

Kan datorer tänka?

Vi ska nu kort beröra ytterligare en relaterad frågeställning, som är relevant såväl för diskussionen kring kognition i allmänhet som för användandet av en viss typ av modeller för kognitiva processer. Det är frågan om representation och informationsbehandling i datorer.

Som bekant kan datorer addera. Antag att datorn lägger ihop 23 och 24 och att svaret blir 47. För enkelhets skull kan vi anta att resultatet skrivs ut genom att symbolen ”47” visas på skärmen; detta antagande är dock inte nödvändigt för det fortsatta resonemanget. Vad betyder det nu att säga att datorn har *adderat* och att den output den ger representerar just talet *fyrtiosju*? Frågorna kan verka konstlade och ”filosofiska”: det är klart att ”47” betyder fyrtiosju, och vi kan ju bekräfta med huvudräkning att datorn adderat de tal vi gav den! Men betänk då att vi skulle ha kunnat använda samma procedur för att multiplicera 10^{23} med 10^{24}; det resultat som datorn ger är i så fall inte 47 utan 10^{47}! Ett lite mer extremt exempel är en användning av datorn för att åstadkomma originell musik. Man trycker på tangentbordet lite hipp som happ och skriver ner resultatet på notpapper. I denna användning betyder ”47” tonföljden F–H.

Vad en dators output betyder är tydligen beroende av *hur vi använder* datorn. Med andra ord, datorn representerar inte *själv* något bestämt tal med symbolen ”47”. Ett motsvarande resonemang kan föras angående inre tillstånd och inre processer hos datorn. Den mängd av magnetiska tillstånd som vi kallar ”filen på hårddisken med denna boks text” representerar inte *i sig själv* det som bokens text handlar om, och den process som gör att vi kan använda en dator för att addera två tal är inte *i sig själv* addition. Annorlunda uttryckt: förvisso behandlar datorn information, i en teknisk mening, men den innehåller inte i sig själv någon semantisk information. Förvisso innebär övergångarna mellan en dators inre tillstånd att variationen i dess input kommer att avspeglas på olika, komplicerade sätt, som gör att vi med fördel kan tala om ”representationer” och ”information”. Till yttermera visso är det ju så, att det är just datorns stora förmåga att i denna tekniska mening ”behandla information” som gör att *vi* gärna använder den för att behandla information – i den semantiska meningen.

Dessa tämligen självklara poänger glöms alltför ofta bort när man använder sig av datoranalogier för att beskriva och förklara kognitiva funktioner. Glömmer man dem, så kan man få för sig att *mental* representation enkelt kan förklaras genom hänvisning till att hjärnan arbetar precis som en dator. Våra resonemang har förhoppningsvis visat att en sådan hänvisning är cirkulär, eftersom de representerande funktioner hos datorn som den kräver uppkommer först i och med att en människa använder den.[40]

Dessa kritiska anmärkningar utesluter förstås inte på något sätt att man kan använda tekniska begrepp som först definierats i samband med datorer för att beskriva hur nervsystemet, eller för den delen ett artificiellt neuralt nätverk, processar information. Ett exempel på ett sådant begrepp är *innehållsadresserbart minne* (se avsnitt 7.2 nedan). Men man måste komma ihåg att man därmed inte automatiskt har givit en beskrivning av någon motsvarande *kognitiv* funktion.

Problematiken kring mental och neural representation samt kring representation och informationsprocessande hos datorer kan (särskilt av en icke filosofiskt indoktrinerad person) tyckas vara konstlad och utan intresse för psykologin, kognitionsteorin och neurovetenskapen. Visst förstår vi, kan man argumentera, vad som menas med att ett visst mentalt, neuralt eller elektroniskt tillstånd representerar ett objekt: det är helt enkelt något som är analogt med att språk, gester eller bilder representerar saker och ting! Ett sådant argument har dock, menar författaren, ett mycket begränsat värde eftersom det inte specificerar hur analogin ska se ut i detalj. De flesta psykologer, kognitionsteoretiker och neurovetare är alltför omedvetna om den filosofiska problematik som döljer sig här, och detta leder dem ofta vilse. Vare sig denna diagnos är korrekt eller inte, är det bra om läsaren försöker behålla lite medvetenhet om problemen när hon läser den följande framställningen.

Detta är inte i första hand en lärobok i medvetandefilosofi eller vetenskapsfilosofi, och därför ska vi inte fördjupa oss mer i diskussionen. Det är inte heller på något sätt nödvändigt att ta någon bestämd ståndpunkt i de filosofiska frågorna för att kunna följa med i den följande framställningen. Man behöver således inte omfatta det ovan nämnda ”subsymboliska paradigmet” (se början av avsnitt 1.2) för att ta till sig teorin om artificiella neurala nätverk. Men det är viktigt att inse att de ANN-modeller som beskrivs nedan primärt skall tolkas *rent tekniskt* – dvs. icke-kognitivistiskt och icke-funktionalistiskt. En helt annan sak är att boken

[40] För liknande argument se Dreyfuss (1992).

också kommer att handla både om den möjliga *biologiska funktionen* hos sådana system i den mån de är realiserade i hjärnan, inklusive deras möjliga betydelse för *kognitiva* funktioner, och om de möjliga *användningarna* av modellerna som abstrakta instrument för att utvidga vår *kunskap*.

1.4 Dynamiska system och systemteori

Ordet ”systemteori” syftar här på en heterogen uppsättning matematiska metoder som har det gemensamt, att de kan användas för att beskriva system som utvecklar sig över tiden. I föreliggande avsnitt introduceras några grundläggande systemteoretiska begrepp och matematiska redskap som ofta används i neural-nätverksteori.

System i konkret och abstrakt mening

Med ett *(dynamiskt) system i konkret mening* menar vi vilket föremål som helst (enkelt eller sammansatt) som följer bestämda lagar för sin utveckling över tiden. Lagarna behöver dock inte vara deterministiska utan kan innehålla element av slumpmässighet. Med ett *system i abstrakt mening* avser vi här de egenskaper som en viss beskrivning av ett konkret system tar fasta på. Det finns några viktiga huvudtyper av sådana beskrivningar.

Diskret och kontinuerlig tid i system

Den första skillnaden mellan olika abstrakta system som vi skall uppmärksamma är den mellan *diskret* och *kontinuerlig tid*. En beskrivning i termer av diskret tid betraktar varje ändlig tidsräcka som bestående av ett ändligt antal på varandra följande tidpunkter, medan en kontinuerlig tidsbeskrivning antar att det mellan två tidpunkter alltid finns en tredje. Observera nu att en beskrivning i termer av diskret tid inte innebär att man måste tro att det konkreta systemet faktiskt utvecklas på ett diskret sätt (dvs. att dess tillstånd ”hoppar” från en tidpunkt till en annan). Den innebär bara att man nöjer sig med att beskriva systemets tillstånd vid dessa diskreta tidpunkter.

En diskret tid gör det abstrakta systemet på flera sätt mer lätthanterligt ur matematisk synvinkel. Detta måste vägas mot att viktig information kan gå förlorad genom att man inte beskriver systemets tillstånd vid alla tid-

punkter. Ett diskret systems utveckling kan karakteriseras genom en *övergångsfunktion*, som beskriver hur systemet förändras från en tidpunkt till nästa – för exempel se nedan – och denna övergångsfunktion kan exakt simuleras av en dator. Neurala nätverksmodeller kan ha antingen diskret eller kontinuerlig tid, men vår framställning kommer nästan uteslutande att syssla med de diskreta varianterna.

System med kontinuerlig tid är matematiskt mer svårhanterbara. Ett vanligt sätt att beskriva deras utveckling är genom *differentialekvationer,* dvs. ekvationer som beskriver hur *tillståndsförändringen* vid en viss tidpunkt beror av olika egenskaper hos systemet. Exempelvis kan banan för en raket som skjutits upp från jorden, men vars motor nu är avstängd, beräknas genom att raketens acceleration (dvs. andra derivatan av positionen) i varje ögonblick är omvänt proportionell mot kvadraten på avståndet till jorden. Ur detta förhållande, tillsammans med vissa konstanter, kan man matematiskt härleda en lag för var raketen kommer att befinna sig t.ex. om en timme från nu. Mer om detta i avsnitt 7.3.

Så snart det blir fråga om någorlunda komplexa system med kontinuerlig tid är det dock mer undantag än regel att man kan göra sådana exakta kalkyler rörande framtida beteende. Man blir därför ofta hänvisad till *simuleringar*. Sådana kan innebära att man bygger en *konkret modell* av det abstrakta systemet (som i sin tur ofta kommit till som beskrivning av ett annat konkret system). Modellen skall vara konstruerad så att det är sannolikt att den uppfyller lagarna för det system som man vill lära sig något om. Sedan iakttar man helt enkelt hur modellen uppför sig över tid. Ett exempel är vindtunnelexperiment vid konstruktion av flygplan. Vanligare är *datorsimuleringar,* som innebär att man med datorns hjälp stegvis räknar sig fram till hur systemet beter sig över tid. Eftersom dagens datorer nästan undantagslöst är digitala innebär en datorsimulering att man använder sig av ett system med diskret tid. För system med kontinuerlig tid blir beräkningarna därför som regel approximativa.

Diskreta och kontinuerliga tillstånd hos system

Ytterligare en möjlig skillnad mellan abstrakta system har att göra med karaktären hos systemens *tillstånd*. Det kan ibland vara lämpligast att beskriva ett konkret system i termer av ett *ändligt* antal olika tillstånd: en lampa kan t.ex. vara *tänd* eller *släckt*. Att vi väljer en sådan beskrivning innebär förstås inte att lampan bara *har* två tillstånd (den har ju t.ex. också en viss temperatur), utan innebär helt enkelt att vi nöjer oss att be-

trakta två tillstånd hos den. Ett sådant betraktelsesätt kan vara alldeles tillräckligt i många sammanhang. Andra konkreta systems tillstånd tjänar absolut på att beskrivas som *kontinuerliga* variabler, exempelvis positionen hos vår raket.

Många av de modeller av neurala nätverk som vi ska tala om senare innehåller *både* diskreta och kontinuerliga tillståndsvariabler. De flesta arbetar dock med diskret tid och är därför matematiskt hanterbara på ett annat sätt än vad system med kontinuerlig tid är. Mer om detta nedan.

Tid och ”tid”

Teorin för dynamiska system är inte bunden till att man tolkar de abstrakta systemens ”tid” som just *tid.* Teorin kan med andra ord också appliceras på sekvenser av händelser som inte utspelas i tiden. Vi kan till exempel betrakta en viss språklig text som ett abstrakt dynamiskt system, som har de enskilda bokstäverna som diskreta tillstånd och som förlöper i diskret ”tid” i enlighet med textens ordning av dessa bokstäver från vänster till höger. I just detta fall finns det en specifik anknytning till verklig tid i och med att man vanligen läser en text från vänster till höger, men detta förhållande är oväsentligt för det systemteoretiska betraktelsesättet. Man kan, i princip, lika gärna betrakta texten som ett dynamiskt system som löper från höger till vänster.

System med och utan input

Beroende på hur man i sin beskrivning avgränsar ett konkret system från omvärlden, kan en viss variabel som har relevans för systemets utveckling komma att räknas som en *tillståndsvariabel* eller en *inputvariabel*. Exempelvis kan man vid beskrivningen av hur temperaturen inne i ett hus utvecklas betrakta utetemperaturen antingen som tillhörig systemet eller som en extern input. Det är dock för många syften lämpligt att hänföra ett för systemet relevant fenomen till inputsidan, om fenomenets *egen* dynamik inte kan beräknas utifrån systemet som helhet. Utetemperaturen om en timme kan inte förutsägas från tillståndet inne i huset tillsammans med den nuvarande utetemperaturen. Att räkna utetemperaturen bland systemets tillstånd skulle därför nödvändiggöra ett indeterministiskt betraktelsesätt (jämför nedan).

System med och utan ”minne”

Ett diskret system kan vara utan ”minne” i den meningen, att vilket tillstånd systemet går till i nästa tidpunkt bestäms av vilket tillstånd som det för tillfället befinner sig i, eventuellt i kombination med den input som systemet får. I sannolikhetsteorin kallar man sådana system för *Markov-kedjor* (av första ordningen). ”Bestämmandet” kan vara probabilistiskt, dvs. tillståndet vid en tidpunkt bestämmer (tillsammans med eventuell input) entydigt *sannolikheterna* för de alternativa tillstånden vid nästa tidpunkt. Tills vidare ska vi dock främst intressera oss för *deterministiska* Markov-system, dvs. sådana där nästa tillstånd är entydigt bestämt av det föregående (plus eventuell input). Ett extremt enkelt exempel ges av en spelpjäs som flyttas ett steg i taget på en slinga av fält numrerade från 1 till 10. Ett annat exempel, som inkluderar input, uppstår om pjäsen varje gång flyttas så många steg som det finns personer i rummet plus siffran på den plats pjäsen står.

Man talar också om system ”med minne”. Vad som då avses är att *tidigare* tillstånd hos systemet än det nuvarande påverkar vilket nästa tillstånd blir. Kunskap om tidigare tillstånd ger alltså ytterligare information om *nästa* tillstånd, utöver den information som det nuvarande tillståndet ger. Ett exempel ges av orden i en svensk ordlista, om varje ord betraktas som ett system med diskreta tillstånd och diskret ”tid” (jämför ovan). Från att en bokstav i ett ord är E kan vi inte med någon särskilt hög säkerhet sluta oss till vilken nästa bokstav skall vara. Om vi däremot vet att E:et föregås av sekvensen SNORK (och vi inte har hämtat bokstäverna ur en av Muminböckerna), kan vi räkna ut att nästa bokstav efter E högst sannolikt är ett L.

Varför talar man om ”minne” i sammanhanget, och varför sätter vi ordet inom citationstecken? Jo, termen används ju om system där nästa steg (”respons”) är *historieberoende*, liksom en organisms reaktion på en stimulus som den habituerats till (eller lärt sig att associera en viss respons till) är historieberoende. Vi har satt termen ”minne” inom citationstecken för att poängtera att vi inte använder den i någon kognitivistisk eller i övrigt antropomorfiserande betydelse. Men från och med nu tar vi bort citationstecknen, för att inte krångla till saker och ting för mycket!

Inte minst när man diskuterar system med respektive utan minne är det väldigt viktigt att komma ihåg distinktionen mellan konkreta och abstrakta system. Samma konkreta system kan nämligen ofta beskrivas genom

båda de nämnda typerna av abstrakt system. Valet beror oftast på hur mycket vi *vet* om det konkreta systemets funktion. Om vi bara har sparsam information om ett konkret systems natur, kanske vi behöver ta in information om fler tidpunkter än den nuvarande för att kunna förutsäga dess framtida beteende. Det utesluter inte, att det konkreta systemet har egenskaper med vars hjälp vi *skulle* kunna göra perfekta förutsägelser från enstaka tidpunkter, om vi kände till dem. Man skulle, med viss risk för missförstånd, kunna uttrycka saken som att systemet *i sig* kanske inte har minne, men att vi på grund av bristande kunskap ändå behöver beskriva det som att det har ett.

Ett näraliggande exempel som kan vara värt att fundera över för dess egen skull är följande. Författaren till denna bok tror att biologiska organismer lyder under vanliga naturlagar. Han har därför som ambition att visa, hur det som vi vanligen beskriver som mänskligt *minne* kan vara grundat i ett konkret biologiskt system som i den systemteoretiska meningen *inte* har minne – nämligen hjärnan.

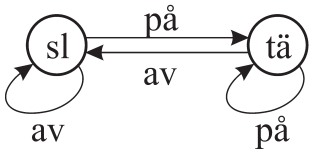
Vi ska nu ta en närmare titt på en speciell grupp av abstrakta system med diskret tid, för att därefter kort betrakta några intressanta egenskaper hos vissa system med kontinuerlig tid och kontinuerliga tillstånd.

Deterministiska ändliga system

Antag att vi beskriver en maskin, en del av en organism eller något annat komplext fenomen i termer av ett visst *ändligt* antal tillstånd som det kan befinna sig i. Antalet tillstånd betecknar vi med bokstaven n och de enskilda tillstånden med a_1, a_2, ..., a_n. Det kan till exempel vara fråga om att vara *tänd* eller *släckt* för en lampa (n = 2), eller om 10 olika, oberoende aktivitetsnivåer hos var och en av nervcellerna i ett system av två celler (n = 10 x 10 = 100). Låt oss också beskriva den tid som vårt system arbetar i som en *diskret* tid. Det innebär att en tidsrymd består av ett ändligt antal på varandra följande tidpunkter. Vid varje tidpunkt antas systemet ta emot någon av ett ändligt antal (m) inputs, b_1, b_2, ..., b_m. För lampans del kan det kanske vara fråga om tryckningar på knapparna *av* och *på* (m = 2), och för nervnätet ett större antal möjliga inputsignaler.

Vilket tillstånd systemet går till i nästa tidpunkt antas vara helt bestämt av den aktuella input i kombination med det tillstånd som systemet för tillfället befinner sig i. Det är med andra ord ett *deterministiskt* system utan ”minne”. Regeln för övergången från ett tillstånd till ett annat, eller

övergångsfunktionen för systemet, kan beskrivas genom en figur där varje cirkel representerar ett tillstånd och pilarna vilken väg systemet tar, givet olika inputs. De kan också sammanfattas i en *övergångstabell.* Figur 2 innehåller en grafisk och en tabellmässig representation av lampan, betraktad som ett deterministiskt ändligt-tillstånds-system.

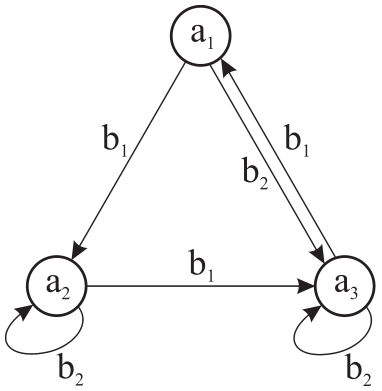


		Input:	
		av	på
Tillstånd:	sl →	sl	tä
	tä →	sl	tä

Figur 2. Ett ändligt, deterministiskt system. Förklaring: Se text.

Vad grafen och tabellen framställer är helt enkelt att ett tryck på ”på”-knappen leder till att lampan är tänd (tä) i nästa ögonblick, oberoende av om den var tänd eller släckt (sl) tidigare. Motsvarande gäller för ”av”-knappen.

Figur 3 visar ett annat, helt abstrakt exempel (med tre tillstånd och två inputs).



		Input:	
		b_1	b_2
Tillstånd:	a_1 →	a_2	a_3
	a_2 →	a_3	a_2
	a_3 →	a_1	a_3

Figur 3. Ett ändligt, deterministiskt system med input. Förklaring: se text.

Grafen och tabellen visar bl.a. att systemet, om det befinner sig i tillståndet a_1, går till tillståndet a_2 om det får input b_1 men till a_3 om det får input b_2. Lägg märke till att vi i figur 2 och 3 rör oss på en abstrakt beskrivningsnivå. Cirklarna betecknar *tillstånd*, inte konkreta delar av en lampa eller enheter i ett neuralt nätverk, och pilarna betecknar övergångar mel-

lan tillstånd, inte konkreta förbindelser mellan enheter! Jämför gärna med figur 1 (s. 12) ovan, där cirklarna och förbindelselinjerna mellan dem faktiskt står för konkreta enheter.

Attraktorer

Ur diagrammet och/eller tabellen i figur 3 kan man bland annat utläsa att om systemet startas i tillståndet a_1 och får en sekvens av input b_2, så hamnar det strax i ett visst stationärt tillstånd, a_3. Detta tillstånd är en *punktattraktor* under input b_2. Hade man istället startat systemet i tillstånd a_2, så hade det med samma input ”fastnat” i en annan punktattraktor, nämligen just a_2. Ger man däremot systemet en lång sekvens av input b_1, så kommer det (oberoende av startpunkt) att hela tiden ”gå runt” i en viss sekvens av tillstånd, $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_1 \rightarrow \ldots$ Denna sekvens sägs vara en *gränscykel* för systemet under input b_1. Både punktattraktorer och gränscykler är *attraktorer* för systemet. Man kan också säga att de är ett slags *jämviktstillstånd* för det, men begreppet attraktor och begreppet jämviktstillstånd sammanfaller inte helt. De sekvenser av tillstånd som ett system genomlöper innan det kommer in i en attraktor brukar kallas för *transienter*, eftersom systemet bara tillfälligtvis gästar dem.

Vi har här förklarat attraktorbegreppet för system med olika men konstanta inputs; vilka attraktorer ett system har beror då på vilken input vi betraktar. Oftare ser man attraktorbegreppet användas om system vars beteende är helt bestämt av det inre tillståndet, dvs. system som funktionellt sett är *utan* input. Dylika system är dock, vilket lätt inses, ekvivalenta med sådana som har en enda, konstant input. *Varje* system av den allmänna typ vi nu beskrivit kommer under konstant input (eller om det är utan input) så småningom att hamna antingen i en punktattraktor eller i en gränscykel. *Vilken* attraktor den kommer till beror dels på utgångstillståndet, dels på vilken input systemet får.

Några egenskaper hos kontinuerliga system

System med kontinuerliga tillstånd och kontinuerlig tid – *kontinuerliga system* – uppvisar ibland en del märkliga egenskaper. Visserligen har de inte sällan punktattraktorer och gränscykler, eller åtminstone något som mycket liknar sådana. Den viktigaste skillnaden mellan sådana attraktorer i diskreta system och deras motsvarigheter i kontinuerliga system är att

de kontinuerliga systemen oftast inte når ända fram till sina attraktorer, utan bara asymptotiskt approximerar dem (alltså närmar sig dem som gränsvärden när tiden blir lång). En pendel som dämpas stannar enligt teorin för kontinuerliga system aldrig i den lägsta punkten, utan närmar sig den i en oändligt avtagande sicksackrörelse; två kringirrande himlakroppar som möts uppnår enligt Newtons lagar aldrig exakt den stabila bana kring varandra som definieras av systemets gränscykel. För praktiska ändamål kan man dock ofta bortse från denna skillnad. Vi ska inte heller göra så stort väsen av den i den följande framställningen.

Andra kontinuerliga system uppvisar helt nya former av attraktorer jämfört med de diskreta system som vi nyss tittade på. Det finns för det första *kontinuerliga attraktorer*, som egentligen är sammanhängande mängder av punktattraktorer. En kula på ett plant golv med viss friktion är ett exempel. Utsätts den inte för några krafter ligger kulan där den ligger, men får den en puff ger den sig iväg till ett nytt ställe där den blir liggande. Det intressanta är att *varje* punkt på golvet är ett potentiellt viloläge, dvs. en punktattraktor, varför golvet som helhet är en kontinuerlig attraktor. Ett annat exempel är en helt horisontell stupränna, vars botten utgör en *linjeattraktor*.

System med kontinuerliga attraktorer har på senare tid uppmärksammats i neural-nätverksteorin, inte minst genom de möjligheter till inlärning av *graderade* responser som de erbjuder. Det står därför mer att läsa om dem i avsnitt 7.3 nedan.

Kaotiska system och egendomliga attraktorer

Ytterligare andra system, nämligen de som kallas *kaotiska*, har attraktormängder av ett mycket speciellt slag. Kaotiska system närmar sig inte någon bestämd punktattraktor eller någon bestämd gränscykel ens på mycket lång sikt. De kan däremot komma in i begränsade regioner av sitt tillståndsrum, som de inte lämnar. Inom dessa regioner kan de bete sig på sätt som kan verka slumpartade men som inte är det. Små differenser vid en tidpunkt kan leda till mycket stora differenser vid en senare tidpunkt, men systemet följer fortfarande deterministiska lagar. Kaotiska system är *inte* slumpartade, bara egendomliga! Man talar faktiskt om ”egendomliga attraktorer” (*strange attractors*).

Ett mycket enkelt exempel på ett kaotiskt system utgöres av den *forcerade pendeln*. Om man låter fästpunkten för en redan svängande enkel

pendel genomgå en regelbunden oscillerande rörelse i vertikalled, kommer positionen hos den andra änden av pendeln under vissa villkor på oscillationen att utgöra ett kaotiskt system. Ett annat känt exempel utgörs av vissa ganska enkla, abstrakta system som ställts upp för att beskriva svängningarna i biologiska populationer. För en del parametervärden beter sig dessa system helt ”beskedligt”, medan andra värden ger kaotiska system. En praktisk implikation är förstås, att man inte ska bli förvånad om ett biologiskt system visar sig vara mycket svårförutsägbart.

Det finns all anledning att tro att hjärnan, sedd ur vissa aspekter, är ett kaotiskt system. Det innebär att den klassiska begreppsapparaten (punktattraktorer och gränscykler) sannolikt inte är tillräcklig för att beskriva hur nervprocesser förlöper på lång sikt. Att lägga till begreppet kontinuerlig attraktor är också viktigt, men inte ens med hjälp av det klarar vi av att att hantera de kaotiska systemen.

Dessa påpekanden ska dock inte tas till intäkt för att tro att det är principiellt omöjligt att beskriva hjärnans sätt att fungera. Det finns faktiskt en exakt matematisk teori för kaotiska system, och, framförallt, kaos i denna mening är inte indeterminism! Däremot är det viktigt att man inser att en del (eller alla) neurala system kanske inte alls har *stabilitetsegenskaper* som liknar dem som ändliga system har. Hur det i detalj förhåller sig med detta är en empirisk fråga som ingen kommer att kunna svara på på länge – men låt oss tills vidare hoppas att evolutionen har selekterat fram nervsystem med en på sin höjd måttlig grad av kaotiskhet och oförutsägbarhet!

System med diskret tid och kontinuerliga tillståndsvariabler

Huvudanledningen till att vi tar upp denna ”hybridtyp” av system här är, att de flesta av våra ANN-modeller kommer att formuleras som sådana. Modellerna utgår således från att skeendet förlöper i diskret tid, men kontinuerliga storheter kan ingå bland tillståndsvariablerna, ofta tillsammans med diskreta variabler. Detta gör att de neurala nätverken i vissa avseenden beter sig som helt diskreta system, i det att de når hela vägen fram till attraktorpunkter eller gränscykler. Det diskreta Hopfieldnätets signaldynamik under givna vikter (avsnitt 7.1) är ett exempel på detta.

I andra avseenden beter sig många av nätverken som ”beskedliga” kontinuerliga system i det att de asymptotiskt närmar sig attraktorer; felfunk-

tionen under inlärning med deltaregeln i linjära system (avsnitt 4.6) utgör ett exempel på detta.

Vissa av de allmänna ANN-modeller som vi kommer att beskriva, särskilt de egentliga återkopplade nätverken (avsnitt 10.3), lämnar rum för specialfall som har ett kaotiskt beteende. Dessa specialfall kommer dock inte att beröras explicit.

Efter dessa filosofiska och matematiska utflykter kan det kanske kännas skönt att återvända till en något mer konkret psykologisk verklighet. Låt oss därför i de två följande kapitlen betrakta några viktiga fakta och teorier, som forskningen om inlärning hos människor och andra djur har kommit fram till genom åren.

2. Inlärning hos andra djur

2.1 Icke-associativt minne

Vi ska nu ge en lite noggrannare beskrivning av några viktiga former av inlärning (och inlärningsliknande fenomen) i den del av djurvärlden som inte inkluderar oss människor. Därmed inte sagt, att de inlärningsformer som detta kapitel handlar om inte *också* förekommer hos människor! De minnesfenomen som man i *första* hand har studerat hos människa behandlas däremot i nästa kapitel.

Parallellt med genomgången av inlärningsfenomenen kommer det att föras en rätt abstrakt diskussion om olika typer av förklaringsmodeller för minne och inlärning. Denna diskussion är till dels för att placera de neurala nätverksmodellerna i ett större intellektuellt sammanhang, dels för att introducera några tankeredskap som kommer att behövas vid analysen av ANN-modellerna.

Först ska vi se på det som kallas *icke-associativt* minne. Här är det inte fråga om att en stimulus A kopplas ihop med en annan stimulus B eller med en godtycklig ny respons R, utan det som händer är att organismens reaktion på en stimulus A förändras genom upprepad exposition för A.

Sensorisk adaptation

En vanlig form för sådan förändring är *sensorisk adaptation*. Sensorisk adaptation klassificeras visserligen långt ifrån alltid som inlärning, men fenomenet har så stora likheter med typisk inlärning att det bör nämnas här. Det innebär, att de perifera receptorernas (sinnescellernas) svar på samma stimulus kan ändras vid upprepad exposition. Ett exempel ges av ögats anpassning till ljus och mörker. Känsligheten för ljus ökar som bekant under en ganska lång tidsperiod efter det att man börjat vistas i mörker, och omvänt minskar känsligheten gradvis när man går tillbaka till en omgivning av ljus. Här spelar pupillförändringar visserligen en roll men förändringar i näthinnecellernas reaktivitet är också av betydel-

se. De senare beror i sin tur bland annat på ändrad tillgång på fotopigment.

Habituering

Sensorisk adaptation definierades här i termer av sinnescellernas reaktioner. Vi kan också betrakta förändringar av organismens yttre beteende. *Habituering* är en sådan förändring; som redan nämnts innebär den en gradvis minskning av en reaktion när en stimulus upprepas. Habitueringens omfattning uppskattas när det gäller ”högre” djur ofta genom storleken hos den så kallade *orienteringsresponsen,* som är ett vanligt beteende vid nya stimuli.[41] Orienteringsresponsen yttrar sig som att djuret plötsligt ter sig mer alert, lyssnar aktivt, ser sig om m.m. Man kan alternativt titta på styrkan av ett aversivt (avvärjande) beteende, t.ex. hur mycket djuret drar tillbaka en extremitet vid beröring. Hos encelliga organismer kan man t.ex. studera hur mycket organismen förflyttar sig eller ändrar sin form efter en given stimulus.[42]

Ett typexempel på habituering ser man när en lång tystnad bryts av ett kort ljud, som inte alls behöver vara starkt. Djuret (eller människan) reagerar ofta kraftigt på detta. Om ljudet återkommer (efter en inte alltför lång paus) blir den åtföljande orienteringsreaktionen eller aversiva responsen inte alls så stark, för att med upprepad exposition som regel försvinna helt. På vanlig svenska kallas detta att man ”vänjer sig vid” ljudet. Att fenomenet är vanligt och välkänt och har ett namn i vardagsspråket betyder dock inte att det är ointressant eller att förklaringen av det skulle ligga i öppen dag. Mer om detta nedan.

Sensitisering

Det omvända förhållandet, att styrkan hos en respons *ökar* med upprepad exposition för en stimulus, förekommer också men fenomenet är inte lika vanligt och sällan lika uttalat som habituering. En initial sensitisering följd av en långdragen habituering är dock inte sällsynt.

En del fall av ökad respons vid upprepad exposition kan antagligen förklaras som ett resultat av att organismen känner igen ett objekt och därför aktiverar olika associationer till det, vilket i sin tur ökar reaktivite-

[41] Sokolov (1963).
[42] Jämför Wyers et al. (1973).

ten. Objektet blir helt enkelt mer intressant när man väl förstått vad det är. Det finns förmodligen också sensitisering av ett i grunden icke associativt slag, men vi ska inte närmare beröra denna möjlighet här.

Att förklara habituering

Minskningen av responsen vid habituering kan inte (i varje fall inte generellt) förklaras som ett specialfall av sensorisk adaptation, bland annat eftersom habitueringen kan vara stark för svaga stimuli som kommer med långa mellanrum. Många teorier om habitueringens mekanismer har föreslagits. De utgår ofta från specifika cellulära mekanismer, t.ex. en nedgång i mängden frisatt signalsubstans i sensoriska synapser.[43] Andra forskare formulerar hellre sina teorier på ett något mer abstrakt plan, i termer av dynamiska egenskaper hos system av nervceller, utan att ta ställning till vilka de underliggande mekanismerna kan vara.[44]

Vi ska inte ge någon översikt av dessa teorier utan istället ta tillfället i akt att peka på möjligheten att ge en annan, ännu mer abstrakt förklaring av habituering, nämligen i termer av jämviktsbeteenden hos en stor klass av dynamiska system. Resonemanget har ett intresse för den här bokens tema därigenom att det understryker, att samma inlärningsfenomen hos olika organismer kan tänkas förklaras med olika, specifika mekanismer som har samma dynamiska egenskaper. Med andra ord, även om habituering hos nakensnäckor skulle bero på presynaptiska förändringar i synapser, så kanske den hos människor beror på förändringar i andra parametrar hos de perceptuella nervnäten – och hos amöbor, som inte har några nervnät, på åter andra mekanismer. Ändå skulle det väsentliga strukturella momentet i alla dessa förklaringar kunna vara gemensamt.

Avsnittet anknyter också till de viktiga begreppen *dynamiskt system*, *ändligt-tillstånds-system*, *punktattraktor* och *gränscykel*. Men det *går* att hoppa över det (fram till början av 2.3, *Operant betingning*) utan att detta omöjliggör läsandet av resten av framställningen.

Habituering är när delsystemen går till jämvikt

Grundtanken i den modell som vi nu ska titta på är, att en organisms habituering till en respons kanske helt enkelt avspeglar *att en mängd delsy-*

[43] Se t.ex. Kandel & Spencer (1968) och Wang (1993).
[44] Dragoi (2002).

stem hos organismen går till sina attraktorer ("till jämvikt") när input är konstant. En renodlad version av denna idé är följande.[45]

Låt oss tänka oss ett ändligt deterministiskt system S som består av ett antal, säg N, delsystem. Det konkreta system som det i grunden handlar om kan vara ett stort nätverk av nervceller som består av många delnät, eller en cell och dess biokemiska delsystem, och vi förutsätter alltså att en beskrivning i termer av ändliga system fångar väsentliga egenskaper hos det. Vart och ett av delsystemen kan vara i n olika tillstånd, kalla dem a_i. För enkelhets skull kan man anta att alla delsystemen delar de tillstånd de kan vara i, men det antagandet är inte alls nödvändigt för resonemanget. "a_i" kan med andra ord beteckna *olika* tillstånd i olika delsystem. Delsystemen tar alla i varje ögonblick på något sätt del av det globala systemets input; låt oss anta att det finns m olika inputs b_j. Delsystemen följer vart och ett sin bestämda dynamik, som dock skiftar från delsystem till delsystem; vi antar för enkelhets skull att reglerna för varje delsystem tillkommit helt slumpmässigt. Matematiskt sett innebär det en övergångstabell där "nästa tillstånd" har bestämts slumpmässigt (och med inbördes oberoende val) för varje plats i tabellen.

Vi antar vidare att output hos det globala systemet (som alltså är konstruerat av slumpmässigt sammansatta delsystem) bestäms "demokratiskt" av alla delsystemens tillstånd. Närmare bestämt är storleken av en viss global output G direkt proportionell till *hur många delsystem som ändrar tillstånd* i ett givet ögonblick. Tänk gärna på G som systemets orienteringsrespons! Det intressanta är nu att ett dylikt slumpmässigt konstruerat system "automatiskt" uppvisar habituering av responsen G till upprepade presentationer av en och samma input.

Antag nämligen först att det globala systemet S startar i ett slumpmässigt valt tillstånd och utsätts en enda gång för en viss input b. En viss proportion av delsystemen kommer då att förbli i *samma tillstånd* som förut; om delsystemen verkligen är slumpmässigt sammansatta kommer denna proportion att vara ungefär 1/n. I ungefär så många fall specificerar nämligen en slumpmässigt tillkommen övergångsregel att systemet skall stanna i det aktuella tillståndet. De övriga delsystemen kommer att gå till ett nytt tillstånd. Om vi presenterar stimulus b en gång till, kommer ungefär samma proportion 1/n av dessa senare delsystem (alltså de som ändrade

[45] För en mer detaljerad framställning se Malmgren (1984). Modellen har inspirerats av tankar hos W.R. Ashby, en av cybernetikens pionjärer. Jämför Ashby (1952) och (1956).

tillstånd nyss) att reagera på b genom att förbli i samma tillstånd. Detta följer också av den slumpmässigt sammansatta övergångstabellen. Och givetvis kommer de delsystem som nyss *inte* ändrade sitt tillstånd, givet input b, inte att göra det andra gången heller! De är ju redan i en punktattraktor under input b. Vi får alltså ett större antal delsystem som förblir i samma tillstånd vid den andra presentationen av b än vid den första. Med andra ord, färre delsystem ändrar tillstånd, och därmed blir den globala responsen G mindre. Med samma resonemang visar man lätt att G med största sannolikhet kommer att fortsätta att minska vid fortsatt exposition för samma input b, ända tills n stycken stimuli givits.

Det är en aning svårare att direkt se att samma mekanism fungerar om de upprepade presentationerna av b äger rum med tidslig separation, vilket ju är fallet i de typiska fallen av habituering, men det är inte heller särskilt svårt att bevisa.[46] Likaledes kommer responsen G att öka igen om man avbryter en konstant stimulering eller en serie av tidsligt separerade, upprepade stimuleringar (dishabituering).

Denna högeligen abstrakta modell duger givetvis inte *i sig* som förklaring av adaptation och/eller habituering i verkliga, biologiska organismer. Dessa är inte slumpmässigt sammansatta, och – ännu viktigare – det är inte heller på något sätt självklart att det är fruktbart att beskriva dem som ändligt-tillstånds-system. Det är vanligare att betrakta dem som *kontinuerliga* system med ett potentiellt oändligt antal tillstånd. Sådana system kan, men behöver inte, ha punktattraktorer (eller ens gränscykler). Men det finns anledning att ställa frågan för varje givet biologiskt system, om det inte har *tillräckliga* likheter med den abstrakta modell som vi ställt upp här för att en förklaring i termer av system med *liknande* dynamiska egenskaper skall vara giltig. Det är här värt att notera att kontinuerliga system som består av delsystem med gemensam input och en kollektiv output organiserad på det principiella sätt som antas i modellen, men som *inte* uppvisar någon habituering av denna kollektiva output, måste vara mycket speciellt konstruerade (de måste nämligen ha ont om både punktattraktorer och sådana icke punktartade attraktorer som innebär *små* tillståndsförändringar).

Huvudavsikten med modellen var dock, som redan nämnts, att visa hur vissa abstrakta, dynamiska systemegenskaper *kan* vara tillräckliga för att förklara ett inlärningsfenomen; systemegenskaper som i olika konkreta

[46] Se Malmgren (1984).

fall kan realiseras – exakt eller approximativt – av många olika mekanismer. Detta gäller, vilket vi ska se nedan, även förklaringar av associativ inlärning.

2.2 Operant betingning

Beskrivning av fenomenet

Operant betingning innebär, som nämndes redan i inledningskapitlet, att ett djurs tendens att avge en respons R i situationen S förstärks om R (i S) följs av en belöning B^+, och försvagas om R (i S) följs av en bestraffning B^-. Belöningssignalen kan ha fler grader än två, men vi ska närmast koncentrera oss på fallet med två alternativ. Ett typiskt experimentellt paradigm innebär att en råtta får mat om den trycker på den vänstra knappen i buren men en elektrisk stöt om den trycker på den högra. Råttan kommer, efter ett antal misstag, snart att bara trycka på den vänstra knappen. Operant inlärning kallas också ”instrumentell betingning”, ”inlärning genom belöning och bestraffning”, eller ”inlärning genom försök och misstag” (den sistnämnda beteckningen har dock oftast en mer allmän betydelse, jämför nedan). En vanlig engelsk term är ”reinforcement learning”. Fenomenet är uppenbarligen av mycket stor betydelse för djurs anpassning till tillvaron. Det utgör också en viktig inlärningsform hos oss människor.

Operant betingning handlar inte bara om kontroll av ”beteenden” i vanlig mening. Även fysiologiska processer som inte står under medveten kontroll tycks kunna styras på operant väg.[47] Så kallad biofeedbackterapi kan sannolikt till största delen analyseras i termer av operant inlärning.[48] I djurförsök kan enskilda neuron betingas till att ändra sin responsfrekvens genom att deras svar direktkopplas till stimulering i vissa ”belöningscentra” i hjärnan.[49]

[47] Det har t.ex. ofta visats att hjärtrytm och hjärtfrekvens kan påverkas genom procedurer som plausibelt kan klassificeras som operant inlärning.
[48] Biofeedback är en metodologi vars kliniska värde ofta har ifrågasatts, men det tycks åtminstone klart att den har en plats i behandlingen av smärta. Se t.ex. Carlsson & Gale (1976), Nestoriuc & Martin (2007).
[49] Olds & Milner (1954)

Förklaringsmodeller

Ett stort antal teoretiska modeller av och förklaringar till operant inlärning har föreslagits. En del av dessa är av ”högnivåig”, kognitiv natur och i första hand avsedda att gälla mänsklig inlärning genom belöning och bestraffning. Dessa modeller är oftast utvecklingar av den vardagliga förklaringen ”individen *tror* att R följs av belöning och utför därför R *för att* uppnå denna belöning” (och omvänt för bestraffningsfallet). Åtminstone en ledande teoretiker på området menar att denna typ av kognitivt laddade förklaringar gäller även för ”lägre” djur, t.ex. för råttor som lär sig att trycka på rätt knapp i en bur.[50] Den antropomorfiserande beskrivningen av råttan som att den ”trycker på den vänstra knappen *för att* få mat” skulle med andra ord vara bokstavligen sann och utgöra en vetenskapligt legitim förklaring av beteendet ifråga. Vi ska inte ta ställning för om så är fallet – det skulle nämligen leda till en omfattande filosofisk diskussion om innebörden och giltigheten hos dylika förklaringar, vilket i sin tur skulle ta oss tillbaka mitt in i debatten om mental representation – utan istället titta på möjliga förklaringar av icke-kognitivistisk typ. Sådana tycks vara särskilt svåra att komma ifrån om man vill förstå hur till exempel hjärtfrekvens eller output från ett enstaka neuron kan styras genom operant inlärning. Råttan, som har ett neuron som ökar sin signalfrekvens på grund av att detta leder till ökad stimulering i lustcentrum, *tror* knappast någonting alls om sina nervceller.

För att bättre förstå vilka speciella egenskaper icke-kognitivistiska förklaringar av operant inlärning måste ha skall vi först studera några liknande – men ändå delvis annorlunda – fenomen från biologiska och tekniska sammanhang.

Kinesis, taxis och tropism

Bland encelliga organismer och växter kan man iaktta beteendeformer, som uppenbarligen är biologiskt ändamålsenliga och som dessutom på ett mer eller mindre naturligt sätt kan beskrivas som ”målinriktade beteenden”.[51] Två sådana typer av beteenden brukar i biologin klassificeras som *taxis* och *tropism*. Båda innebär att någon aspekt av en organisms beteende direkt styrs av en viss gradient i omgivningen. (En gradient är en systematisk och gradvis variation i rummet, eller i ett abstrakt rum, av värdet

[50] Se Mackintosh (1983), ss. 111f.
[51] För några begreppsbildningar och många vackra exempel ur djurvärlden, se Hinde (1970).

hos en variabel.) En del djur styrs att röra sig i riktning mot en ljuskälla genom att det visuella systemet ”jämför” input från vänster och höger öga – en form av *fototaxis*. En växt som växer i riktning mot en ljuskälla, eftersom skillnaden i ljusexposition för olika partier av växten leder till olika tillväxthastighet hos de olika delarna, visar *fototropism*. Vi säger gärna till vardags att den växer som den gör *för att* få mer ljus. Så länge vi inte syftar på några medvetna eller omedvetna avsikter hos växten med en sådan beskrivning, så är den okontroversiell och väl förenlig med exempelvis biokemiska förklaringar. I ett naturvetenskapligt sammanhang är det också helt klart att det inte är fråga om en kognitivistisk förklaring, utan en metafor för en funktionell sådan.

Den form av gradientstyrt, ”målinriktat” beteende hos encelliga djur som vi framförallt ska intressera oss för är dock ännu mer primitiv än taxis och tropism, och kallas *kinesis*. Ett slag av kinesis innebär att både hastigheten hos en organisms rörelse och sannolikheten att den byter riktning står i (ungefär) omvänd proportion till koncentrationen av en faktor som är av livsviktig betydelse för organismen.[52] Låt oss för enkelhets skull kalla denna faktor ”mat”. Kinesis betyder alltså: ju mindre mat, desto snabbare och mer varierat i fråga om riktning rör sig organismen; ju mer mat, desto mindre och mer enkelspårigt rör den sig. Man inser att detta i det långa loppet leder till att organismen tenderar att uppehålla sig i närheten av större matkoncentrationer, medan den tenderar att ”fly” från områden där det finns mindre mat. Betraktar man en sådan organism kan det verkligen se ut som om den ”letar” efter mat och ”kommer till ro” först där den ”finner” mat.

Kinesis innehåller en primitiv version av *felkorrigering*. Om organismen tar ett steg bort från maten, dvs. ”gör fel”, kommer den nästa gång att ta ett större steg i en annan riktning och har då bättre chans att nå maten än om den behöll steglängden och riktningen. Går den mot maten, dvs. gör rätt, ökar istället sannolikheten för att den ska bli kvar i närheten av maten.

Felkorrigering och kontrollsystem

Felkorrigering är ett begrepp som har stor betydelse inte bara i biologin utan också i tekniska sammanhang. Tekniska system för *kontroll* av olika storheter kan oftast beskrivas i termer av felkorrigering. Det klassiska ex-

[52] Detta är närmare bestämt både *orthokinesis* och *klinokinesis*. Hinde (1970), ss. 148 ff.

emplet är en enkel inomhustermostat i ett hus som utsätts för spontant varierande yttertemperatur. Om temperaturen i ett rum är högre än det *önskade värdet* skickar termostaten en *styrsignal* till elementet att sänka temperaturen; om temperaturen istället är för låg går signalen ut på att höja temperaturen. Styrsignalen har här alltså motsatt tecken mot differensen mellan aktuell och önskad temperatur. Dess styrka är i det allra enklaste fallet omvänt proportionell mot storleken på samma differens (linjär kontroll), och i många andra kontrollsystem är den på något annat sätt växande med felets storlek. Kinesis, som vi just beskrivit detta fenomen, kan också ses som ett kontrollsystem där styrsignalens styrka växer med storleken på matbristen.

Om en inomhustermostat fungerar bra får den innetemperaturen att gradvis närma sig önskevärdet, och efter en stund ligger temperaturen någorlunda stabilt kring detta värde. En generellt för stark styrsignal kan dock leda till att systemet inte stabiliserar sig utan kommer i svängning runt önskevärdet, medan en generellt för svag signal kan medföra att systemet inte fullt kan korrigera för de svängningar i temperaturen som beror på externa faktorer (och som ju är anledningen till att man har en termostat). I system med mer komplicerad dynamik kan mycket specifika olinjära styrsignaler behövas för att systemet inte ska komma i en svängning som för det långt bort från alla önskevärden. Läran om tekniska kontrollsystem är därför en matematiskt avancerad vetenskap. De grundläggande begreppen i den är dock både lätta att ta till sig och högst relevanta inte bara för inlärningsteorin (inklusive teorin om ANN), utan också för förståelsen av hur enskilda handlingar styrs och kontrolleras. Låt oss utveckla det påståendet närmare.

Löpande felkorrigering av aktuella handlingar

När man skjuter med pilbåge mot en måltavla får man ”feedback” efter skottet genom att titta på hur långt från tavlans mitt man träffade (och i vilken riktning från mitten), och kan justera utskjutningsvinklarna i enlighet med detta. Det här är förstås ytterligare ett exempel på en kontrollkrets och en styrsignal. Vi kan kontrastera det mot ett annat exempel, nämligen när vi *pekar med en pekpinne* på en bestämd punkt på en tavla. I det senare fallet behöver vi inte vänta med att justera vårt beteende tills vi ser hur långt från den avsedda punkten vi träffar, utan kan ändra armens och handens läge redan när vi märker att pekpinnen inte *kommer att* träffa rätt punkt. Skillnaden gentemot pilbågsfallet kan uttryckas som att vi när det gäller pekandet använder *löpande* kontroll, eller *on-line-*

kontroll. Denna förutsätter att vi har tillgång till en *feluppskattning i förväg.* Man använder ibland termen *antecipatorisk feedback* för den signal, med vars hjälp vi under handlingens utförande gissar hur väl den kommer att lyckas. I fallet med pekpinnen kan denna signal förstås vara av visuell karaktär, men kan t.ex. också innebära att man känner att pekpinnen var tyngre än man trodde varför man kommer att träffa tavlan för lågt om man inte tar i lite mer. Ett fysikaliskt exempel på antecipatorisk feedback ges av en termostat med s.k. utegivare, där utetemperaturen används för att uppskatta kommande förändringar i innetemperaturen.

Fysiologisk forskning har visat att vi i många av våra beteenden, till exempel när vi snabbt fångar ett annalkande föremål med handen, inte utnyttjar någon feluppskattning *under* exekveringen av beteendet. Man talar i dessa fall – i analogi med pilens flykt – om en *ballistisk* rörelse. I åter andra fall, t.ex. när man genskjuter en springande person, kontrollerar man som regel rörelsen on-line och utnyttjar antecipatorisk feedback. Denna bygger delvis på extern sensorisk information (hur långt har han kommit, och hur långt har jag kommit?) men också på intern feedback från receptorer i muskler, leder och senor (hur trött är jag? kan jag hålla farten?). Givetvis ingår även ballistiska *moment* i de rörelsesekvenser som vi kontrollerar on-line.

En uppenbar anledning att föredra en ballistisk rörelse framför on-line-kontroll är, att sådan kontroll tar extra tid på grund av all den information som ska skickas åt olika håll. Ballistiska rörelser är snabbare, men det gäller ju att ”göra rätt från början”!

Både enkel ballistisk styrning och kontroll genom antecipatorisk feedback klassificeras ibland som former av *feed-forward-kontroll*. Oftast används dock denna term för styrning av ett beteende genom att responsen på en förväntad stimulus ”programmerats i förväg”. Ett exempel ges av när man med handen skall fånga en boll som man ser komma i ovanligt hög fart. Man spänner då i förväg armens muskler extra mycket för att bollmottagandet ska bli optimalt. Ett annat exempel: om man vill kunna lägga handen på en het spisplatta så att den ligger kvar, så måste man redan i förväg bestämma att den förväntade reflexen att rycka bort handen skall undertryckas. Sådan feed-forwardkontroll (i snäv mening) kan ingå i en ballistisk rörelse. I avsnitt 10.4 kommer ballistiska rörelser att diskuteras närmare, och en ANN-modell presenteras där som medger en i viss mening *dynamisk* ”förprogrammering” av rörelsen.

Inlärd kontroll

Relationerna mellan å ena sidan styrning av aktuella beteenden, å andra sidan operant inlärning är många och intressanta. Till exempel kan såväl ballistiska som icke-ballistiska rörelser läras in, och det sker bland annat på operant väg. Det finns förvisso medfödda ballistiska beteenden såväl hos människor som hos andra djur, och en del av dem är inte modifierbara genom inlärning, men när en tennisstjärna returnerar en hård serve ser vi ett typexempel på en *inlärd ballistisk rörelse.* Inlärning av dylika beteenden sker på flera olika nivåer och med olika tidsskalor; tennispelaren har således tränat sin allmänna servereturförmåga i många år, men han har också under matchens gång lärt sig att slå servereturerna med hänsyn till banans egenskaper och den aktuella motståndarens speciella sätt att serva.

Att lära sig köra bil innebär bland annat *inlärning av on-line-kontroll.* Man lär sig ju hur bilens framfart skall kontrolleras på grundval av löpande feed-back-information – exempelvis hur snabbt och hur mycket man behöver vrida på ratten om man av misstag råkar komma ut på vägrenen. Att sedan köra en ny bil på en ny väg innebär en fortlöpande uppdatering och specificering av dessa inlärda, generella kontrollförmågor (”styrförmågor”, här i en mycket konkret mening).

Operant inlärning och kontroll av aktuella beteenden

Det är också nyttigt att jämföra kontrollen av ett aktuellt beteende, vare sig felkorrektionen sker ”efter” eller ”under” beteendet, med inlärning genom belöning och bestraffning. I båda fallen är det ju fråga om felkorrektion genom styrsignaler. Kontroll ”efter beteendet” (off-line-kontroll) kan till och med naturligt sägas *vara* en form av inlärning.

Dock verkar det inte vara självklart riktigt att generellt analysera off-line-kontroll av beteenden som just *operant* inlärning. Detta beror på karaktären av styrsignalen. Vid operant inlärning är ju denna *ospecifik* i så måtto, att *samma* slags belöning eller bestraffning kan fungera som förstärkning av helt olika beteenden, enbart beroende på det tidsliga sambandet mellan beteende och bestraffning/belöning. (Man kanske inte kan få en människa att göra riktigt vad som helst för pengar, men det finns knappast några gränser för vad man tränat råttor till genom mat och elchocker.) De fall av ”off-line”-kontroll av aktuella beteenden som vi hittills har nämnt har istället utnyttjat mycket mer specifika styrsignaler med en intuitivt mer

tydlig relevans för beteendena ifråga: hur långt från mitten pilen träffade (och i vilket väderstreck), respektive vilken riktning tennisbollen tog efter slaget. Om bågskytten inte ser var pilen träffar utan bara får en elektrisk stöt när han missar, är det inte så lätt för honom att veta hur han ska sikta nästa gång.

Ser man dessa exempel på off-line beteendekontroll som inlärning så är de alltså i varje fall inte uppenbara fall av *operant* inlärning – däremot är de onekligen inlärning genom *försök och misstag,* i en allmän mening. En annan sak är att sådan inlärning, sedd i ett vidare perspektiv, kanske är *beroende* av belöning och bestraffning för att kunna fungera. Bågskytten och tennisspelaren kanske lär sig sikta bara för att vinna stora prispengar. Vetskapen om de hägrande prispengarna kanske finns i bakhuvudet även när bågskytten tränar. Och vem vet, kanske det är en njutning att träffa även om man inte vinner några pengar eller ens någon ära? Det må vara hur det vill med det; kontrasten mellan specifika styrsignaler (termostatens feedback, platsen pilen träffade på och vart bollen tog vägen) och ospecifika sådana (belöning eller bestraffning) är ändå giltig, och verkningsmekanismerna för de olika typerna av signaler torde skilja sig åt på många sätt.

Som en förberedelse för en närmare diskussion av mekanismerna bakom operant inlärning ska vi nu titta på en en liten och maximalt enkel förklaringsmodell. Förutom som ytterligare en övning i systemteori är den avsedd som en abstrakt skiss till förklaring av hur en bestraffningssignal egentligen fungerar. Liksom den abstrakta modellen av habituering ovan är den inte oundgänglig för förståelsen av den följande framställningen.

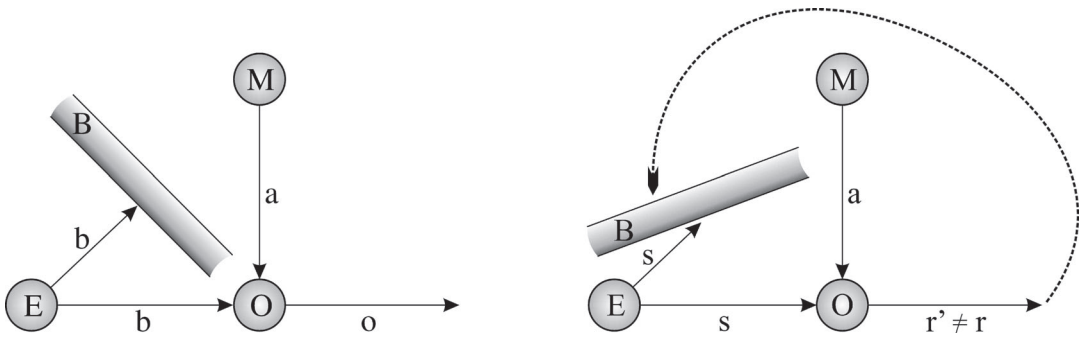
En abstrakt modell av inlärning genom bestraffning

Grundtanken i denna modell är, kan man säga, densamma som i vår miniteori för habituering.[53] Mer specifikt beskriver modellen ändringen av organismens beteende genom en bestraffningssignal som att systemet *lämnar en punktattraktor* på grund av att bestraffningen innebär ett *ökat inflöde av information*.

Vi tänker oss en organism som består av en minnesenhet M och en outputenhet O, och som rör sig ett steg i taget på en yta i form av ett rutnät. Varje plats på spelplanen är associerad med en speciell input till organis-

[53] För detaljer i modellen se Malmgren (1985).

men; man kan säga att organismen i varje ögonblick ”tittar” på sin omedelbara omgivning E. Spelplanen antas vara ganska rikt varierad (dvs. det finns många olika möjliga inputs för organismen). Minnesenheten M är ett slumpmässigt sammansatt, ändligt deterministiskt system av den typ vi mötte ovan. (Nu handlar det om ett enda system, inte ett kollektiv.) Den skickar hela tiden information om sitt tillstånd, a, till outputenheten O. Outputenhetens svar o på en viss input b bestäms dels av denna input, dels av signalen a från minnesenheten. Det finns fyra möjliga outputs, vilka motsvarar steg i de fyra väderstrecken. Normalt sett är det bara outputenheten som ”ser” omgivningen och reagerar på dess inputs, eftersom minnesenheten är skyddad från inputs genom en informationsbarriär B. Under exceptionella omständigheter (jämför nedan) kan dock också minnesenheten ”se” omgivningen, dvs. ta emot input. Modellen åskådliggörs i figur 4.

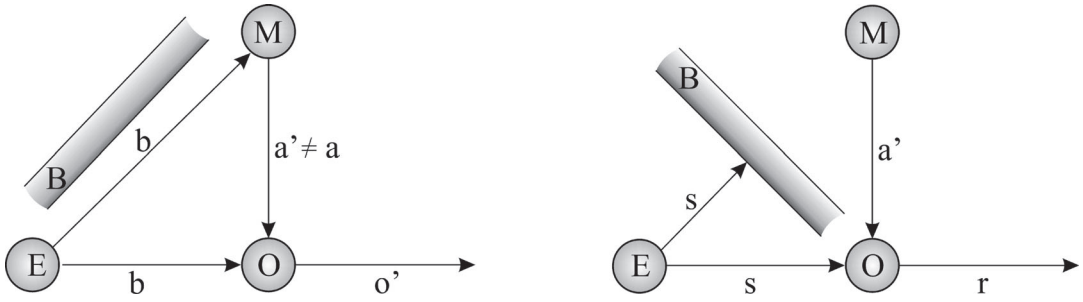


Figur 4. En minimal modell av inlärning genom bestraffning. Förklaring: se text.

Organismen rör sig alltså i varje ögonblick ett steg åt något av de fyra väderstrecken, beroende på vilken respons outputenheten ger. Efter varje steg ”tittar” den på den nya plats den kommit till och ger en ny respons, som beror av den nya input och minnesmodulens aktuella tillstånd a. Det senare kan också ha förändrats medan varelsen tog ett steg eftersom minnesmodulen är ett dynamiskt system; chansen är dock rätt stor att a *inte* har förändrats. Minnesenheten har ju normalt inte någon varierad input och hamnar därför lätt i en punktattraktor.

Vi låter organismen hålla på så här en stund... men nu invaderas spelplanen av en angripare, som signalerar sin omedelbara närvaro för organismen genom en speciell input, s (för ”signal”). För att organismen ska undvika att bli biten av angriparen måste den gå *ett steg åt vänster,*

vilket förstås hänger på att outputenheten ger rätt respons (låt oss kalla den ”r”) på input s. Om den *inte* gör det så blir alltså organismen biten. I det abstrakta systemet innebär detta att den ovan nämnda informationsbarriären lyfts bort under det följande ögonblicket (eller en kort följd av ögonblick). Detta illustreras i figur 5.



Figur 5. Ökat informationsflöde som bestraffningssignal. Förklaring: Se text.

Tänk gärna på vad som händer nu som en orienteringsfas: organismen chockas av det plötsliga angreppet och försöker snabbt ta in information om vad som händer. (Men tänk inte för mycket i antropomorfistiska termer... modellen är i sig helt mekanistisk.) Denna orienteringsfas får i sin tur som följd att minnesmodulen får betydligt större sannolikhet än tidigare för att ändra sitt tillstånd. Detta beror helt enkelt på att *minnesmodulens beteende under ett eller flera ögonblick påverkas av en varierad input från omgivningen*. Och den självklara följden blir i sin tur, att organismen med en inte obetydlig sannolikhet *ändrar* sin respons på nästa attack av angriparen. Denna respons beror ju delvis av signalen från minnesmodulen. Med lite tur undviker organismen med framgång nästa attack.

Vi ser här en process som har likheter med en *kinesis i minnessystemets tillståndsrum* (mängden av tillstånd som detta system kan vara i). Bestraffning leder till ökat informationsinflöde, som leder till större tendens än vanligt att ändra minnestillstånd; detta leder i sista änden till en ökad sannolikhet för att organismen skall undvika nästa bestraffning. Den här organismens beteende är dock ännu mer primitivt (och mer likt vad man i sannolikhetsteorin kallar en *slumpvandring*) än den typ av kinesis som vi beskrivit ovan. Den encelliga varelsen som sökte sig till mat styrdes ju av en graderad felsignal och graderade i viss mån sitt svar efter denna felsignal. Vår lilla organism beter sig i enlighet med något som liknar den

maximalt enkla strategin ”om det blev rätt, ändra ingenting; om det blev fel, ändra åt vilket håll som helst”. Man kan, om man vill, kalla strategin för ”diskret kinesis”.

Det speciella med den här modellen är att den förklarar bestraffning i termer av ett ökat informationsinflöde. Detta kan kanske verka väldigt onaturligt – man vill gärna föreställa sig en bestraffning som innebärande smärta, olust e.d. – men å andra sidan är det en både vardaglig och vetenskaplig erfarenhet att orienteringsresponser tenderar att avbryta pågående beteenden. Den största svagheten med modellen är förstås, att bestraffningsmekanismen är för ospecifik för att (i sin ursprungliga form) kunna fungera i ett mer komplext system. Men det förefaller inte omöjligt att överföra denna basala idé om bestraffningens natur till en kontext där det ökade informationsinflödet begränsas till en för det aktuella beteendet relevant struktur. Vi ska inte försöka utveckla denna tanke närmare här.[54] Nästa avsnitt skall istället behandla den ospecifika signalens problematik på ett mer allmänt sätt.

Problemet med ansvarsfördelning

När råttan trycker på fel knapp och får en stöt ändrar den, om den är läraktig, sitt beteende så att den slipper stötar i fortsättningen (och istället får mat). Men hur vet den *hur* den ska ändra sitt beteende? Ser man på problemet från en antropomorfiserande synvinkel ligger svaret nära till hands: råttan inser att elstöten hade att göra med tryckandet på knappen, och eftersom den inte vill ha fler stötar låter den bli att trycka på knappen nästa gång. Om man inte är nöjd med en sådan kognitivistisk förklaring utan vill ha en mer mekanistisk sådan (till exempel för att man så småningom vill göra en ANN-modell av operant inlärning) är det inte riktigt lika lätt.

Man kan då ställa frågan i termer av kontrollsystem: vad gör att styrsignalen korrigerar systemets (råttans) beteende *i rätt avseende och åt rätt håll* när det önskade värdet inte har uppnåtts? Vi vet ju att nervsystemet är ett oerhört komplext system, långt mer komplext än termostater och andra simpla kontrollkretsar ur läroböckerna (för att inte tala om den lilla modellorganism som vi beskrev i föregående avsnitt). Samtidigt tycks styrsignalen (den elektriska stöten) vara mycket ospecifik. Termostater använder sig av en graderad signal som härrör från en temperatur-

[54] Men jämför gärna Malmgren & Östensson (1989).

skillnad, och en bågskytt tittar på träffbilden i två dimensioner för att finjustera nästa skott, men i operant inlärning kan (som redan nämnts) samma yttre signal användas för att i olika kontexter åstadkomma en mängd olika beteenden – det enda som är viktigt tycks vara sambandet i tiden. Hur kan denna ospecifika signal styra ett så komplext system som en råtthjärna åt rätt håll i ett visst avseende, utan att samtidigt påverka en hel mängd *andra* variabler som *inte* är relevanta för det önskade beteendet?

Problemet som vi just formulerat går i teorin för lärande system under beteckningen *problemet med ansvarsfördelning* (”the problem of credit assignment”), och det är av central betydelse för denna teori.[55] Låt oss därför förklara det en gång till, nu i mer vardagsspråkliga termer. Hur kan den elektriska stöten klara av att ändra på *just det* neurala tillstånd som är ansvarigt för tendensen att trycka på den vänstra knappen snarare än den högra, och vad beror det på att den ändrar detta tillstånd *åt rätt håll?* I varje ögonblick händer ju en enorm mängd saker i råttans nervsystem, och en enorm mängd beteendetendenser finns lagrade (kanske som effektiviteten hos olika neurala förbindelser, kanske på andra sätt). Den elektriska chocken ger förstås en kraftig insignal till nervsystemet, en signal som borde kunna ställa till oreda både här och där. Att den fungerar som beteendemodifierare *överhuvud* är med andra ord kanske inte så konstigt. Men hur lyckas den *selektivt* träffa just det ställe som lagrar det beteende som orsakade responsen?

Fördröjd belöning och bestraffning

Allra mest svårbegripligt blir det hela i de fall då bestraffningen kommer *långt efter* den handling som gav upphov till den. Det är till exempel känt att råttor kan lära sig att undvika mat som gör dem illamående först många timmar senare.[56] Hur lyckas obehagssignalen från magen träffa och ändra i just den neurala struktur som åstadkom valet av mat *flera timmar tidigare?*

Återigen, om man tänker kognitivistiskt så går det sistnämnda fenomenet kanske att förklara med att råttan *minns* vad den gjorde för några timmar sedan och *förstår* att det var valet av mat som ledde till illamåendet. Men tänk dig istället att du ska designa en robot som uppvisar operant inlärning med tidsfördröjning. Då hjälper det inte att veta vad roboten ska

[55] Beteckningen torde härstamma från Minsky & Selfridge (1961).
[56] Mackintosh (1974), ss. 53ff.

”veta” och ”förstå”, för vad innebär robotens insikt och förståelse i elektroniska termer? Kan man verkligen bygga roboten så, att de ospecifika bestraffningssignalerna (styrsignalerna) träffar den struktur som genom sin (ibland mycket tidigare) aktivering ledde fram till det bestraffade beteendet, och dessutom träffar den så att styrsignalerna verkar åt rätt håll och med rätt styrka?

För att göra en lång historia kort så är det här problemet av sådan dignitet, att de flesta ANN-algoritmer som används för att lära robotar att bete sig rätt är så kallade *evolutionära algoritmer*. De ger förvisso ofta mycket bra resultat, men man måste notera att de inte kan tjäna som modeller av operant inlärning. De innebär nämligen en selektion av beteenden på populationsbasis, inte på individnivå. Man sorterar helt enkelt bort de neurala nätverk som inte beter sig rätt. Se vidare avsnitt 4.6 och 10.3.

Som lärande algoritmer på individnivå kan man däremot klassificera en del av de metoder, som i den matematiskt orienterade litteraturen ibland sammanfattas under beteckningen *reinforcement learning*.[57] Relevansen av dessa algoritmer för förståelsen av det som i psykologin ibland går under samma namn (alltså just det som vi här kallat ”operant” eller ”instrumentell” inlärning) är långtifrån klarlagd. Jämför också avsnitt 4.6.

Operant inlärning som selektion av beteenden

Psykologer och neurovetare har ibland föreslagit att operant inlärning faktiskt har sin grund i en selektionsprocess, men inte på populationsnivå utan på individnivå – det vill säga, man tänker sig att det är de enskilda beteendena hos en individ som selekteras bort.[58] Om man laborerar med sådana modeller måste man dock tänka på risken att ens påståenden blir triviala och inte förklarar någonting. På en rent beskrivande nivå är det ju nämligen klart att operant inlärning innebär elimination av vissa beteendetendenser till förmån för andra, och därmed en *selektion av beteenden.* Därav följer emellertid inte att den korrekta bakomliggande förklaringen måste vara ”selektionistisk” i den meningen att vissa konkreta neurala strukturer dör eller sätts ur funktion under inlärningen, medan andra förblir opåverkade av inlärningen och därför kommer att dominera beteendet. En dylik selektionistisk förklaring förutsätter att de alternativa res-

[57] Sutton & Barto (1998).
[58] Se särskilt Skinner (1981).

ponserna redan från början finns kodade i var sina neurala strukturer.[59] Men det kan ju lika gärna vara så, att egenskaperna hos en enda neural struktur ändras åt ”rätt” håll genom bestraffningen, och då har vi inte att göra med någon selektionsprocess i konkret mening. Det sist sagda utesluter inte alls möjligheten av icke-triviala selektionistiska förklaringar av operant betingning, utan författaren vill bara påpeka att man också bör leta efter andra typer av modeller.

Med dessa funderingar måste vi lämna området operant betingning tills vidare. Vi återkommer dock strax med några reflexioner kring förhållandet mellan operant och klassisk betingning. Senare i boken skall vi tala mycket mer om felkorrigering, och lite mer om design av system som lär sig genom fördröjd belöning och bestraffning.

2.3 Klassisk betingning

Likheter och skillnader mellan klassisk och operant inlärning

I Pavlovs försök och i andra paradigmatiska exempel på klassisk inlärning föreligger redan vid inlärningens början en viss respons UCR (Un-Conditioned Response) på en stimulus, den obetingade stimulus UCS. Mycket förenklat kan klassisk betingning beskrivas som att upprepad koppling mellan den betingade stimulus CS och UCS leder till att djuret ”överför” responsen på CS. Det har då uppstått en betingad respons, CR. I Pavlovs mest kända experiment med hundar saliverar hunden när ringklockan ljuder, inte bara när matskålen visas.

Ovanstående beskrivning är förenklad i flera avseenden. Vi skall återkomma till detta strax. Men först några ord om relationerna till operant betingning. Den viktigaste skillnaden är att det vid Pavlovsk inlärning inte finns något essentiellt samband mellan djurets respons och en eventuell belöning eller bestraffning. Visserligen är UCS i Pavlovs hundförsök en omedelbar signal om belöning (mat), *men hunden får mat oavsett om den saliverar vid ringsignalen*. I andra försök kan UCS dessutom vara en mycket mer neutral signal. I situationen med operant inlärning letar man å andra sidan förgäves efter en motsvarighet till den styrande funktion som UCS har i det klassiska paradigmet. Råttan som står inför

[59] Edelmann (1987) utgår faktiskt från selektion på neuronal nivå som en grundläggande förklaringsprincip för inlärning.

valet att trampa på höger eller vänster pedal i sin bur får ingen ledning av någon UCS som redan från början utlöser ett dylikt trampande på endera pedalen, utan det är den kausala relationen till mat eller elchock som ensam gör att råttan lär sig att trampa rätt.

Till det sagda måste genast sägas att många situationer torde vara en blandning av klassisk och operant inlärning. Man ser t.ex. inte sällan att en duvas sätt att picka på en knapp för att få mat delvis styrs av dess karakteristiska sätt att picka efter mat (”autoshaping” – dvs. situationen är en UCS för en respons som liknar den slutliga, adekvata responsen), och det vore ju konstigt om inte Pavlovs hundar lärde sig så bra som de gjorde *bland annat* eftersom de alltid fick mat mot slutet av försöken. Operant inlärning med tidsfördröjning kanske på ett väsentligt sätt bygger på en klassisk inlärning av tidssambandet, etc. Men detta hindrar inte att man bör försöka hålla isär begreppen. Exempelvis bör man inte (något som är alltför vanligt) tala om UCS i ett renodlat klassiskt experiment som en ”förstärkare” (reinforcer) utan att samtidigt poängtera att experimentet inte använder UCS som en förstärkare i den operanta meningen, dvs. som en bestraffning eller en belöning.

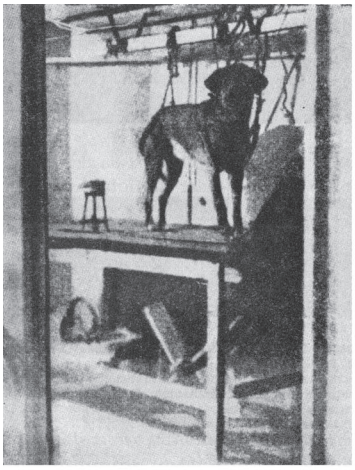
Hur ”djup” distinktionen mellan de två inlärningsformerna egentligen är är en svår och omdiskuterad fråga. Inte så få teoretiker har försökt reducera den ena inlärningsformen till den andra. Men min mening är att så länge någon sådan reduktion inte genomförts så skall klassisk och operant inlärning betraktas som två skilda processer. Med andra ord, förstärkning genom en ospecifik belönings/bestraffningssignal och förstärkning genom en styrsignal av typ UCS är två olika saker.

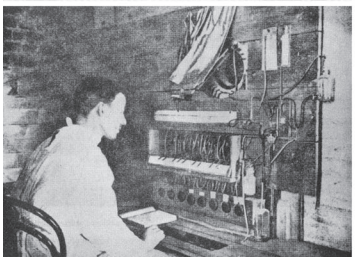
I de följande avsnitten skall vi gå in på några problem som är gemensamma för klassisk och operant betingning, men där vi oftast hämtar exemplen från klassiska situationer.

Fördröjd betingning

Försök med s.k. fördröjd betingning, när CS kommer en viss tid före UCS och inte samtidigt eller omedelbart före, visar för det första att den betingade reaktionen normalt sett *inte* kommer omedelbart efter CS, utan först *vid den tidpunkt då UCS är att vänta.* Tidigt i inlärningen och vid speciella motivationsförhållanden dyker CS däremot gärna upp ”för tidigt”. Figur 6, som är sammanställd från ett av Pavlovs originalarbeten, visar hur den betingade responsen hos ett vältränat och välnärt djur lo-

kaliseras till den förväntade tidpunkten för UCS, medan ett hungrigt djur tenderar att reagera ”för tidigt”.[60]





Experiment of 13th December, 1907.
Previously to the experiment the dog was fed at the usual time.

Time	Stimulus	Salivary Secretion in drops per 30 secs. during the isolated action of the conditioned stimulus
2.40 p.m.	Whistle	0, 0, 0, 0, 2, 6
2.54 ,,	,,	0, 0, 0, 2, 3, 6
3.30 ,,	,,	0, 0, 0, 0, 2, 5

Experiment of 15th December, 1907.
Conducted upon the same dog after two days deprivation of food.

Time	Stimulus	Salivary Secretion in drops per 30 secs. during the isolated action of the conditioned stimulus
3.5 p.m.	Whistle	0, 2, 2, 4, 4, 6
3.20 ,,	,,	2, 5, 3, 3, 4, 6
3.40 ,,	,,	1, 6, 4, 3, 5, 5

Figur 6. Från Pavlovs laboratorium: en av hans berömda hundar, kontrollrummet, och resultat från två försök med fördröjd betingning. Tabellen är självförklarande.

Timingen av responser är ofta mycket viktig i djurvärlden, t.ex. när ett djur ska göra en undvikande rörelse gentemot en anfallare. Att röra sig för sent är givetvis inte bra, men lika ödesdigert kan det vara att röra sig för tidigt, vilket ju ofta skulle bli fallet om en CR direktkopplades till sin CS. Man inser att mekanismen med fördröjd betingning har ett mycket stort överlevnadsvärde!

Det faktum att det välträ nade djuret normalt sett ”väntar” med responsen till den förväntade tidpunkten för UCS gör att det är vilseledande att beskriva situationen som att saliveringen nu blivit en ”reaktion på CS”. Saliveringen, CR, är snarare en reaktion på en *anteciperad (förväntad) UCS*. Förväntan om UCS är förvisso i sin tur utlöst av CS, och i viss mening är CR därför en ”reaktion på CS”. Men fördröjningen av responsen talar för att den förväntade UCS är ett väsentligt mellanled i processen.

[60] Ur Pavlov (1960) [1927].

En annan omständighet, som bidrar till att man inte bör beskriva klassisk betingning som ett överflyttande av en respons från en stimulus till en annan, är att klassisk betingning kan äga rum även när de relevanta perifera responserna är blockerade. Genom att i efterhand tillåta responser kan man visa att djuret under träningen ändå fått en förmåga att antecipera UCS utifrån CS.

Teorin om stimulussubstitution

Att tala om CR som en reaktion på en *förväntad* händelse (UCS) är att använda sig av en kognitivt laddad term, som många anser sig böra undvika i djurpsykologin. En förklaring av vad som händer vid klassisk betingning, som tar hänsyn till de fakta som nu presenterats och som dessutom undviker kognitiva termer, använder sig istället av begreppet *stimulussubstitution*. Att djuret reagerar med CR just vid den tidpunkt då UCS kan förväntas beror på att en ”inre” stimulus, med samma (relevanta) egenskaper som UCS, uppträder i djurets sensorimotoriska system vid just denna tidpunkt och därigenom *ersätter* UCS för djuret.[61] Den tidsliga kopplingen mellan CS och UCS gör, menar man alltså, att det så småningom uppkommer en ”inre UCS” som har samma tidssamband med CS. Tanken anknyter nära till idén om mentala representationer som simuleringar av perception (se avsnitt 1.3 ovan samt slutet av detta avsnitt).

Det skall tilläggas att många så kallade *antecipatoriska responser* också har ett stort överlevnadsvärde. En antecipatorisk respons är en betingad reaktion som utlöses *innan* den obetingade stimulus är att vänta, och som i typfallet har funktionen att göra organismens respons på denna stimulus mer effektiv. En katt som skall fånga en råtta just när denna passerar på stigen framför den gör rätt i att spänna musklerna till språng strax före den tidpunkt vid vilken den förväntar sig att råttan skall komma. Däremot ska den inte *ta språnget* innan råttan kommer!

En del teoretiker hävdar fortfarande att betingning inte innebär en association av en respons R till en förväntad UCS, eller till en stimulus som ersätter UCS, utan istället en association av R till CS. En anledning till att de håller fast vid denna teori (trots ovanstående argument) kan vara att de koncentrerat sitt intresse på de antecipatoriska responserna, som ju typiskt kommer *före* den förväntade tiden för UCS och som därför snarare kan förefalla vara direkta responser på CS. En bidragande orsak till detta

[61] För en diskussion av S-S respektive S-R-teorier se Mackintosh (1967).

kan i sin tur vara, att just de responser som Pavlov valde att i första hand studera – salivering och insöndring av bukspott – *också* har ett värde som antecipatoriska responser. Det kan faktiskt vara effektivt att börja producera saliv några sekunder innan maten anländer!

Betingning till sekvenser

En annan tidslig aspekt av inlärning genom klassisk betingning är, att ett djur kan lära sig att värdera en stimulus olika beroende på vad som föregått den. Detta gäller för övrigt också operant betingning – en hund kan tränas att hämta tidningen på en visselsignal och att ligga ner på en annan, även om signalernas sista fas är densamma i de två fallen. Annorlunda uttryckt kan djuret lära sig att associera inte en enstaka stimulus, utan en *sekvens* av stimuli, med en annan stimulus (eller med en respons).

Fördröjd betingning och betingning till sekvenser ställer till samma slags problem för simplistiska förklaringar i termer av förbindelser mellan två nervceller eller nervcentra. Den tillämpning av Hebbs princip som vi illustrerade i figur 1 ovan (avsnitt 1.1) förutsätter således att CS-neuronet har en *direktkoppling* till R-neuronet. Och sekvensen AAB kan inte *specifikt* associeras med UCS bara genom att ett neuron som representerar B får förbindelsen stärkt till ett neuron som representerar UCS. I så fall kommer djuret ju att reagera likadant på sekvenserna ABB och BAB, som på AAB.

Representation av tid

Det mesta vi sagt i detta avsnitt om fördröjd betingning och betingning till sekvenser gäller också situationer som involverar operant inlärning. Djur kan exempelvis lära sig att trampa på en pedal exakt fem sekunder efter en signal för att få mat, eller omedelbart efter signalen för att få mat fem sekunder senare.

Fenomenen fördröjd betingning och betingning till sekvenser är en stor teoretisk utmaning eftersom de tycks förutsätta en *representation av tid*. Hur går det till när ett djur inom sig mäter den tid som gått sedan CS, eller när det håller reda på skillnaden mellan sekvenserna AAB och CAB? Problemets dignitet inses om inte annat om man frågar sig hur man ska få in fördröjd betingning i vår tidigare, enkla modell av Hebbs princip. När förbindelsen mellan CS-neuronet och R-neuronet stärks enligt denna

princip, så ökas CS-neuronets tendens att själv framkalla R. Men hur ska vi kunna modellera att CS framkallar R efter just den tid som brukat gå mellan CS och UCS? Den enkla modellen kan ju bara förklara omedelbara responser.

En naturlig utvidgning av modellen är att föra in ett *tidsregister* där den aktivitet som CS åstadkommer lagras i ett antal kopior av CS-neuronet. Signaler skickas hela tiden bakåt i en kedja av sådana kopie-neuron. Ett neuron på plats *n* i kedjan ”representerar” därför den stimulus som presenterades för *n* tidssteg sedan. (Ett annat sätt att åstadkomma ett tidsregister beskrivs i avsnitt 10.4.) Om vi antar att intervallet mellan CS och UCS är fem tidssteg så är det de indirekta förbindelserna till R-neuronet från neuron nr 5 som kommer att tränas enligt Hebbs princip i Pavlovs försök med fördröjd betingning, och just detta neuron kommer så småningom själv att kunna aktivera R.

Antagandet om tidsregister förefaller också vara användbart för att förklara betingning till sekvenser. Ett sådant tidsregister kan ju samtidigt lagra hela sekvensen AAB, och betingning till sekvensen ifråga blir då inte mer problematisk än betingning till komplexa, samtidiga stimuli. Det senare fallet har vi visserligen inte modellerat, men det är enkelt att klara av med ANN-modeller av standardtyp (se t.ex. avsnitt 6.2 och 9.3).

Det förefaller inte omöjligt att djurs och människors nervsystem innehåller dylika tidsregisterneuron – de skulle också kunna förklara en mängd andra fenomen som har att göra med det mänskliga korttidsminnet, se avsnitt 3.4 och 10.4 – men en betänklig sida av modellen som *generell* förklaring av hjärnans behandling av sekventiell information är att den kräver ett mycket stort antal registerneuron. Tänk för det första på hur fina tidsdiskriminationer djur och människor kan göra. Den modell vi bygger ska t.ex. kunna förklara en orkestermusikers förmåga att följa dirigentens instruktioner om nyanseringar av tempo och rytm, en förmåga som handlar om hundradelar av sekunder. Det behövs många exakt stämda registerneuron för att detta skall fungera. För det andra fordras givetvis mycket långa neuronkedjor för att representera långa tidsintervall. För det tredje är det ju inte bara *en* stimulusdimension som ska in i neuronkedjan, utan *allt* som vi ska minnas. Hur många parallella kedjor behövs inte för att lagra det vi upplever när vi bevistar ett symfoniframförande?

Huvudalternativet till idén om tidsregisterneuron är att förklara tidsrepresentation med hjälp av processer i återkopplade neuronnät, dvs. nerv-

kretsar där signalen som ett neuron sänder ut kommer tillbaka till samma neuron efter en tid (eventuellt i förändrad form). Detta gör, som vi ska visa nedan (avsnitt 10.4), att man kan slippa antagandet om separata neuron för separata tidssteg. Redan ett mycket enkelt återkopplat neuronnät kan representera många olika tidsavstånd; diskriminationen mellan tidpunkter blir dock sämre och sämre ju längre dessa avstånd är. En annan fördel med dylika nätverk är att de på ett naturligt sätt kan modellera *kontinuerlig* tid.

Den abstrakta, diskreta modell som presenteras sist i föreliggande avsnitt visar också hur ett antagande om feedback kan göra en explicit tidsrepresentation onödig.

Element, helhet och kontext i betingningsförsök

När man försöker beskriva och förklara ett fall av klassisk betingning är det viktigt att man utvidgar sitt perspektiv till att omfatta mer än CS, UCS och R under den beskrivning som experimentledaren givit dem. (Motsvarande resonemang kan förstås föras om operant betingning.) Man måste ta hänsyn också till det sammanhang i vilka stimuli förekommer – och till hur djuret själv sannolikt uppfattar dem.

Det har till exempel visats upprepade gånger att djur kan betingas till ganska abstrakta drag hos stimulussituationen. En råtta klarar t.ex. att lära sig att maten finns bakom den dörr som signaleras av den *större* av två fyrkanter. Den ”effektiva stimulus” är då inte den absoluta storleken på denna fyrkant, utan den storleksrelation den har till den *andra* fyrkanten. Vad djuret ”egentligen” har lärt sig kan man pröva genom att studera det sätt på vilket det *generaliserar.* Antag att en råtta har tränats i en bur (bur 1), där det finns två dörrar märkta med olika stora fyrkanter. I bur 1 döljer sig, som sagt, maten bakom den dörr som är märkt med den större av fyrkanterna. I bur 2, som råttan nu får gå in i, har fyrkanten på den dörr som leder till maten samma absoluta storlek som fyrkanten på matdörren i bur 1, men den är den *mindre* av fyrkanterna i bur 2. Råttan kan då antingen välja symbolen med *samma storlek* som den symbol som signalerade mat i bur 1, eller symbolen som har *samma storleksrang* som den symbolen hade. Om den gör det första valet kan vi dra slutsatsen att den lärt sig att en viss *egenskap* signalerar mat, i det andra fallet att den blivit betingad till att en viss *relation* signalerar mat.
En annan kontextuell faktor av största betydelse är den tidsliga. Nu talar vi inte bara om korttidsperspektivet (fördröjd betingning, betingning till

sekvenser etc.), utan om det faktum att djuret före den aktuella inlärningssituationen också varit utsatt för en mängd associativ inlärning. Denna inverkar givetvis på resultatet av den aktuella träningen. Om djuret exempelvis nyligen har exponerats för CS upprepade gånger utan att denna följts av UCS, så går betingningen långsammare. Detta fenomen (”latent inhibition”) har framstått som ett mysterium för en del teoretiker, men kan enkelt förklaras genom att djuret före experimentet lärde sig att CS *inte* följs av UCS. Nu måste det lära sig något nytt, men det är självklart att denna inlärning inte kan innebära att den tidigare informationen genast kastas bort. Betingning är ju en process som ackumulerar information över tid, inklusive den tid som förlöpte alldeles före experimentet!

En del välkända och omdiskuterade fenomen i samband med *extinktion* kan sannolikt ges liknande förklaringar. Extinktion, eller utsläckning, av en respons äger rum när CS under en längre tid inte längre följts av UCS. Efter utsläckning av en association är det i allmänhet mycket lättare att betinga djuret till samma respons än vad det var första gången. Om man t.ex. antar att de allra senaste betingningsförsöken väger mycket tungt vid bestämmandet av den aktuella betingningens styrka och riktning medan de tidigare väger mycket lätt men ungefär lika, och tänker på att ett extinktionsförsök egentligen innebär betingning till frånvaro av UCS, blir detta begripligt.

En abstrakt modell för klassisk betingning

Övervägandena i de föregående avsnitten har inte minst varit avsedda att visa att klassisk och operant betingning inte är så enkla mekaniska processer som man kanske skulle kunna tro. Både klassisk och operant inlärning kan involvera ett uppfattande av exakta tidsliga samband och av abstrakta förhållanden, och de innebär båda en komplicerad sammanvägning av ny och gammal information till ett aktuellt beslut. Därmed inte sagt att vi behöver använda dylika kognitivt laddade termer (”uppfattande”, ”beslut”) i våra ultimata förklaringar av klassisk och operant betingning. Personligen lutar författaren snarare åt att det kommer att visa sig att kognitiva processer i själva verket *är* mekaniska, om än inte *enkla.*

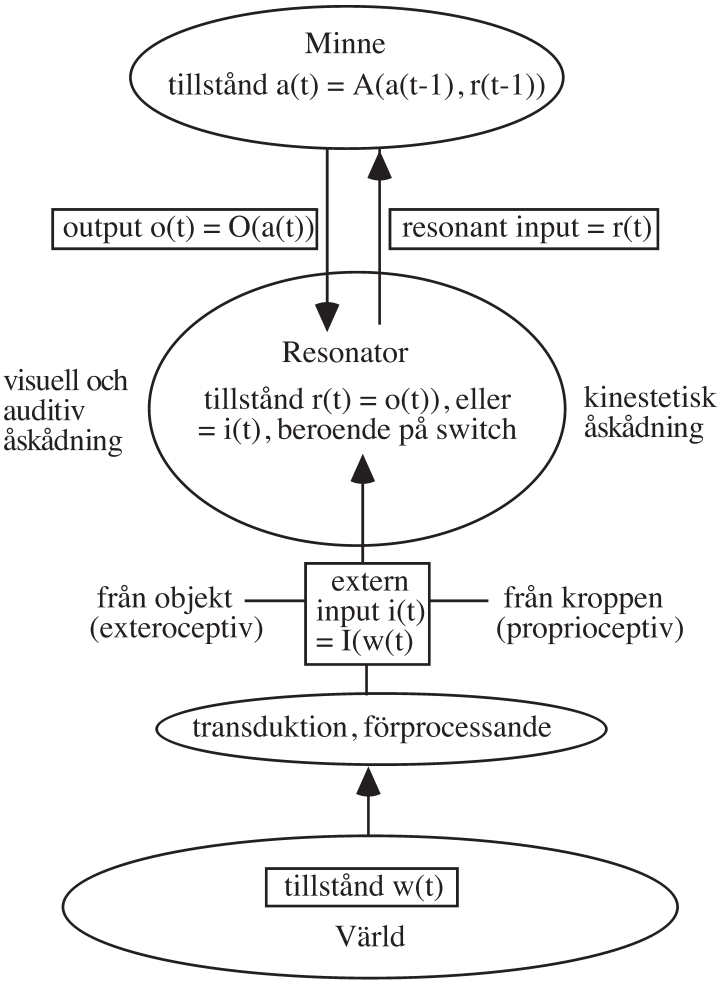
Som en avrundning av detta kapitel ska vi nu presentera ytterligare en högeligen abstrakt, systemteoretisk modell som kan förklara klassisk betingning – inklusive betingning till sekvenser – genom mycket generella egenskaper hos system med återkoppling. Liksom för de två tidigare försöken i samma genre – alltså modellerna för habituering respektive

operant inlärning – gäller dels att modellens relevans för mer konkreta, biologinära förklaringar förvisso återstår att klarlägga, dels att avsnittet inte är absolut nödvändigt för den följande framställningen. Den läsare som vill öva sitt systemteoretiska tänkande, och den som tycker om spekulativa teorier, kanske ändå vill följa med på färden. Annars ses vi i början av kapitel 3!

Denna tredje modell utgår från ett ändligt system som är kapabelt att simulera input från omgivningen. Det betyder för det första att (en del av) dess output kan anta samma värden som dess input, för det andra att det omväxlande tar input från omgivningen och från (denna del av) sin egen output. Vi ska nu visa att ett sådant system under mycket allmänna förutsättningar kommer att tendera att anpassa den simulerade output till input från omgivningen. Låt oss säga att ett sådant system är ”naturligt resonant”.

Antag ett ändligt, deterministiskt system vars input växlar på det nämnda sättet. Med andra ord bestäms det tillstånd som systemet går till vid en given tidpunkt dels av det nuvarande tillståndet – a(t) i figur 7 nedan – dels av en ”resonant” input r(t) som kommer *antingen* från omgivningen *eller* via feedback från systemets output o(t). Vad som bestämmer växlingen mellan de två operationsmodi kan vi lämna därhän tills vidare. Vi döper dem till ”erfarenhetsmodus” respektive ”tankemodus”.[62]

[62] Jämför Malmgren (1991) och (1996). Figur 7 och 8 är modifierade från det senare arbetet. Modellen i figur 7 skiljer mellan input från kroppen och input från omgivningen i övrigt (och därför också mellan kinestetisk och audiovisuell feedback). Denna distinktion är inte väsentlig för resonemanget i föreliggande avsnitt. Notera också att tillstånden i resonansmodulen betecknas som ”åskådning”. Detta beteckningssätt har inte heller någon betydelse för det närmast följande formella resonemanget, men binder samman modellen med vad som tidigare sagts om åskådning som simulering av varseblivning.

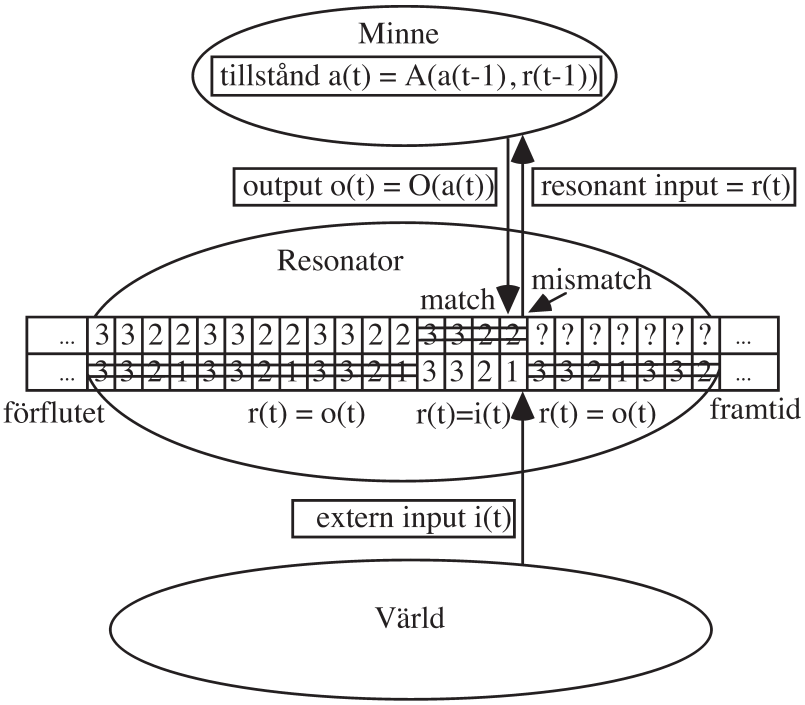


Figur 7. En"naturligt resonant" maskin som ibland läser sin egen output. Förklaring: se text.

Låt oss nu betrakta ett slumpmässigt sammansatt sådant system (se avsnittet om habituering). Vi ska nu visa att det kommer att tendera att simulera erfarenheten, när det arbetar i tankemodus. Vi antar först att systemet får en konstant input b och växlar mellan perioder av erfarenhetsmodus (då den tar denna stimulus som input) och perioder av tankemodus (då det läser sin egen output r). *Om* systemet går till en punktattraktor under input b, och *om* dess output r vid början av nästa period av tankemodus (det vill säga den simulerade input) råkar vara just b, *så kommer systemet att vara stabilt för obegränsad tid.* Om output när tankemodus startar däremot är skild från b, så riskerar feedbackslingan att ta systemet ur dess attraktor. Då kommer det dock att finnas en ny möjlighet för stabilitet genom att systemet senare går till en *annan* attraktor under b, där det faktiskt har b som output – etcetera.

I en stor uppsättning av slumpmässigt sammansatta system kommer många av systemen därför så småningom att vara i en *gemensam* attraktor för input från omgivningen respektive via feedback, och *den vanligaste output i tankemodus, dvs. den simulerade input, kommer att vara identisk med omgivningsstimulus.* Om man är entusiast kan man uttrycka detta som att systemen gärna fortsätter att tro, medan de tänker, att omvärlden har den beskaffenhet som systemet tidigare har erfarit. I ickekognitiva termer kan man, mer modest, säga att systemen tenderar att substituera för, eller simulera, en konstant stimulus.

Detta var en blygsam början. Man kan nu visa ytterligare två ting, i stigande intresseordning. Först utsätter vi upprepade gånger ett system av den typ vi just beskrivit för en repetitiv *sekvens av omgivningsstimuli*, och låter det växla på ett lämpligt sätt mellan att ta denna input från omgivningen och att läsa sin egen output. Se figur 8! Den övre raden av siffror betecknar sekvensen av outputs från minnesenheten, och den undre betecknar inputsekvensen. De icke överstrukna siffrorna står för den input respektive output som faktiskt avläses.



Figur 8. En naturligt resonant maskin som lär sig en repetitiv sekvens. Förklaring: se text.

Med ett resonemang av samma karaktär som det vi förde nyss inser man ganska lätt, att systemet i tankemodus kommer att tendera att på ett stabilt sätt producera en *sekvens av outputs* som är identisk med den yttre sekvensen. Anta nämligen, som i figur 8, att systemet en tid stabilt åstadkommit sekvensen 33223322... men att det nu får input 3321. Systemet kommer då att ha en god chans att komma ”ur balans” och har därför en viss möjlighet att, så småningom, hitta ett stabilt tillstånd där det självt producerar 33213321...

Observera gärna att ”match” och ”mismatch” i figur 8 inte står för konkreta jämförelser mellan sekvenser, utan bara markerar att en deterministisk maskin som befinner sig i ett givet tillstånd alltid reagerar likadant på samma input men tenderar att reagera olika på olika inputs. Modellen förutsätter alltså *varken* en intern, explicit representation av tid *eller* en samtidig förekomst av verklig och simulerad input! Vi kan däremot säga att maskinen gör en *virtuell jämförelse* mellan verklig och simulerad input.

Slutligen konstruerar vi ett stort antal system helt slumpmässigt, förutom att en viss bakgrundsinput, kalla den 4, till dem är *inert* och tenderar att inte ändra på systemens tillstånd. (Det sistnämnda antagandet är inte nödvändigt för principresonemanget, men gör resultaten tydligare.) Vi ger systemen denna bakgrundsinput men bryter den då och då för att istället presentera en sekvens 12 av två omgivningsstimuli. Oftast följs 12 av ytterligare en omgivningsinput 3, men ibland får det efter input 12 istället gå ett steg i tankemodus. När vi studerar vilka ”interna” responser systemen avger omedelbart efter input ab i det senare fallen finner vi, att en med tiden ökande andel av dessa responser består av 3!

Hur kan det vara så? Vi har ju inte byggt in någon associationsprincip i systemen från början. Jo, med en viss sannolikhet (som ackumuleras med stigande antal episoder) kommer en episod av 123 att föra systemen till ett tillstånd som är en punktattraktor under 123 – i betydelsen av ett tillstånd, som systemet skulle återkomma till om man upprepade sekvensen 123. Även om systemen under en period istället får en bakgrundstimulus 4, kommer (eftersom 4 är inert) en substantiell proportion av dem att vara kvar i en sådan attraktor när nästa presentation av 123 börjar. Nu ger vi bara 12 som input. Om ett system omedelbart efter 12 ger output 3, och läser denna som sin input, kommer responsen *inte* att föra systemet ut ur attraktorn. En annan output, avläst som input, riskerar däremot att bryta systemets stabilitet. Därför kommer den vanligaste reponsen på 12

att så småningom bli 3. Systemen tenderar med andra ord att lära sig att när 12 presenteras simulera, eller substituera för, den stimulus som erfarenhetsmässigt varit associerad just med 12.

För att verifiera att systemen lärt sig att associera 3 till just sekvensen 12, och inte till stimulus 2, utsätter vi dem slutligen en gång för sekvensen 22 och låter dem sedan gå till tankemodus. Vi ser då bara en mycket liten ökning av antalet 3 som outputs. Detta beror givetvis på att system som är i en punktattraktor under 123 inte alls behöver vara i en sådan attraktor under 223.

För en utförligare analys och några numeriska resultat, jämför referenserna i not 62.

Liksom för våra tidigare abstrakta modeller gäller att denna tankemodell inte har någon omedelbar applikation på den biologiska verkligheten. Däremot vore det intressant att undersöka, om verkliga nervsystem har dynamiska egenskaper som liknar modellens tillräckligt mycket för att en strukturellt analog förklaring av klassisk betingning skulle kunna vara giltig.

3. Modeller för mänskligt minne

3.1 Deklarativt och procedurellt minne

Antalet teorier om det mänskliga minnet är mycket stort, och varje pedagogisk framställning behöver göra ett snävt urval av fakta, begrepp och teorier. Man måste ju börja någonstans!

En grundläggande dikotomi, som vi redan stött på i inledningskapitlet, är den mellan *deklarativt* och *procedurellt* minne. Distinktionen har en motsvarighet i begreppsparet veta *att*/veta *hur* (närmare bestämt *hur man gör*, se nedan). Läsaren vet antagligen *att Göteborg ligger i Västsverige*, och har lärt sig en gång att så är fallet. Detta är deklarativt minne. Men läsaren vet antagligen också *hur man cyklar*, och har under en sommar i barndomen lärt sig hur man cyklar; detta är procedurellt minne. Många inlärningssituationer involverar förstås båda slagen av minne; exempelvis lär man sig i skolgeografin också *hur* man hittar Göteborg på kartan, och när man tränar cykling lär man sig också *att* man bör hålla båda händerna på styret.

Är deklarativt och procedurellt minne former av betingning?

Det är frestande att anta att procedurellt minne är nära besläktat med betingade reaktioner. Mycket av det som vi kallar procedurellt minne hos människor etableras faktiskt i situationer som liknar experiment med antingen klassisk eller operant betingning. Det finns tydliga likheter mellan å ena sidan den situation där en råtta lär sig att trampa på rätt pedal för att inte få en elektrisk stöt, å andra sidan den där ett barn lär sig att trampa rätt på cykelpedalerna för att inte ramla och slå sig. Det innebär *inte* att barn är lika enkla som råttor, och inte heller att förmågan att cykla kan förklaras genom någon enkel ”mekanisk” modell för betingning. Man ska dessutom inte glömma att vissa ledande teoretiker i djurexperimentell inlärningspsykologi menar, att även förklaringen av *råttans* färdigheter fordrar ett antagande om kognitiva processer!

Relationerna mellan begreppen *deklarativt minne* och *betingning* är inte lika uppenbara, främst därför att det inte finns något enkelt sätt att definiera den ”respons” som skulle vara ett avgörande kriterium på att en person har lärt sig *att* si och så är fallet. De främsta kandidaterna till sådana responser är språkliga beteenden, t.ex. ett yttrande av ”Göteborg ligger i Västsverige” vid ett läxförhör om Sveriges geografi, men det finns många argument (inte minst av filosofisk art) som underminerar förtroendet för en sådan lösning. Kan man t.ex. inte träna en papegoja (eller ett franskspråkigt barn) att säga ”Göteborg ligger i Västsverige” som svar på frågan var Göteborg ligger, utan att man därför vill medge att papegojan (barnet) har lärt sig *att* Göteborg ligger i Västsverige? Vi måste tyvärr lämna frågan i detta olösta skick. Men det ska tilläggas att även de situationer i vilka deklarativt minne uppstår ofta har stora likheter med de situationer i vilka klassiska och operanta betingade reflexer lärs in, och det vore oklokt att från början avvisa hypotesen att de grundläggande inlärningsmekanismerna är åtminstone delvis gemensamma.

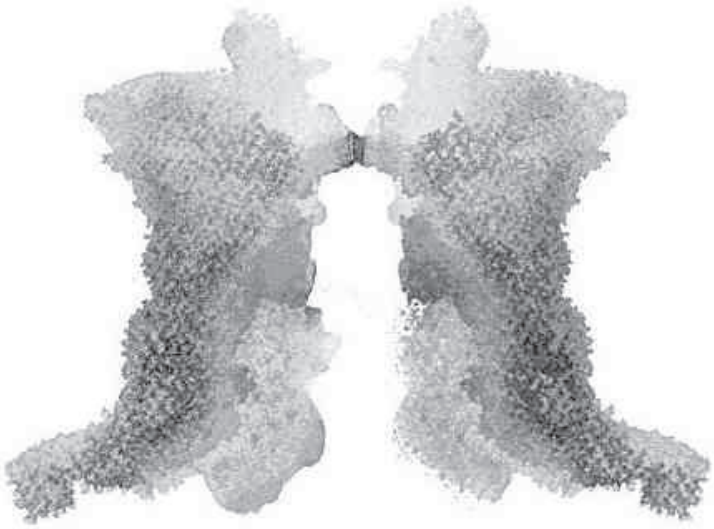
3.2 Perceptuellt, episodiskt och semantiskt minne

Perceptuellt minne

Ett viktigt begrepp med något oklara relationer till distinktionen mellan deklarativt och procedurellt minne är *perceptuellt minne*. John Stuart Mill och flera filosofer före honom diskuterade fenomenet; jämför vår tidigare diskussion om simultan association och perceptuell komplettering (avsnitt 1.1). Eleanor och James Gibson är två av de moderna forskare som ägnat sig åt det.[63]

Inte vilken form av minne som helst som har att göra med varseblivning faller under begreppet perceptuellt minne. Fenomenet innebär att *själva varseblivandet* modifieras genom inlärning. Det kan vackert exemplifieras med lösningen av fixeringsbilder. När man väl har blivit visad på att bilden i figur 9 kan ses som två människoliknande gestalter som står och tar spjärn med fötterna bakåt och händerna mot varann, så är det nästan omöjligt att senare se bilden utan att också se denna lösning.

[63] E.E. Gibson (1969), J.J. Gibson & E.E. Gibson (1955).



Figur 9. En mångtydig bild.

Men fenomenet perceptuell inlärning är givetvis verksamt även i vardagen. Allteftersom vi blir mer och mer bekanta med en viss typ av natur- eller kulturföremål så förändras vår varseblivning av objekten, inte bara våra trosföreställningar angående dem. Bland annat påverkas vårt sätt att *särskilja* och *kategorisera* föremål. Det är klarlagt att förmågan att perceptuellt skilja två stimuli från varandra blir bättre dels genom ökad exposition för dessa stimuli, dels om urskiljningsförmågan belönas på ett eller annat sätt.[64]

Perceptuellt minne yttrar sig också som *enhetsbildning*. Genom träning kan vi få förmågan att uppfatta komplexa konfigurationer av stimuli ”i ett enda slag”. Det är denna form av perceptuell inlärning som är verksam när vi gradvis lär oss att känna igen en blomma eller en fågel. De klassiska empiristiska filosoferna fäste stor vikt vid detta fenomen. De tillmätte det rentav en betydelse som det säkert inte har, i och med att de menade att *allt* uppfattande av komplexa enheter är inlärt. Under det tidiga 1900-talet argumenterade gestaltpsykologerna, på ett för många övertygande sätt, att man måste anta vissa medfödda principer för perceptuell helhetsbildning.[65]

Det kan kanske förefalla frestande att klassificera perceptuellt minne som en form av *deklarativt* minne, men vid närmare eftertanke framstår detta

[64] Jfr. Goldstone (1998).
[65] En nyanserad framställning av gestaltpsykologin ges i Sundqvist (2003).

som problematiskt. Visserligen kan man uttrycka en aspekt av sin varseblivning av fixeringsbilden i figur 9 genom att säga, ”Jag ser nu *att* det kan vara två människor”. Men det är samtidigt klart att den varseblivning man har innehåller mycket *mer* än det som uttrycks i orden ”det kan vara två människor”, och det är inte alls självklart att *allt* som den innehåller kan uttryckas genom de begrepp som vårt språk erbjuder.[66] Vi klassificerar därför, tentativt, perceptuellt minne som en tredje huvudform av minne vid sidan av deklarativt och procedurellt minne.

Episodiskt och semantiskt minne

E. Tulving införde begreppsparet *episodiskt* och *semantiskt* minne.[67] Med ”episodiskt minne” avser han en persons konkreta, mer eller mindre bildmässiga minnen av episoder i livet, t.ex. minnet av när man första gången ramlade av cykeln. Centralt för Tulvings begrepp är att episodiska minnen involverar ett självbiografiskt medvetande om förfluten tid. Att aktivera ett episodiskt minne är ”som att resa tillbaka i sin egen tid”. ”Semantiska” är istället de minnen som är lagrade i form av påståenden om världen, exempelvis minnet att Gustav II Adolf dog i slaget vid Lützen. Jag var inte med vid Lützen och har därför inte några minnesbilder därifrån. Mitt minne av *när jag hörde talas om* slaget första gången (i den mån jag har ett sådant minne kvar) kan däremot antagligen klassas som episodiskt.

Som redan antytts i inledningskapitlet kan Tulvings distinktion tolkas på flera sätt. Man skulle kunna läsa den som en skillnad mellan olika sätt att *etablera* ett minne (genom egen erfarenhet, respektive i andra hand), eller som en *innehållsbaserad* skillnad mellan å ena sidan minnen som handlar om episoder i personens eget privata liv, å andra sidan kunskap om fakta utanför denna krets. I den rimligaste tolkningen av Tulvings egen text rör distinktionen dock inte primärt någondera av dessa två alternativ utan *i vilken form (genom vilken ”mental kod”) innehållet presenteras.* Detta formmässiga kriterium, som alltså tar fasta på skillnaden mellan direkta, åskådliga föreställningar och språkligt kodade sådana, sammanfaller vare sig med det etiologiska (orsaksmässiga) eller med det innehållsliga. Exakt vad det ska innebära att semantiskt minne är kodat i ”språklig

[66] Det finns en omfattande diskussion i dagens filosofi, som vi tyvärr inte kan gå närmare in på, om relationen mellan innehållet i våra varseblivningar och innehållet i våra begrepp. Som ledtråd till den som vill ta del av litteraturen kan nämnas att diskussionen ofta förs under rubriken ”non-conceptual content”.
[67] För en aktuell diskussion av det förstnämnda begreppet se Tulving (2002).

form", i "påståendeform" eller "semantiskt" är förvisso inte glasklart. Vi ska dock inte gå in på någon djupare analys av de problem som uppstår när man försöker avgränsa de antydda begreppen närmare.

Av ungefär samma skäl som anfördes ovan beträffande perceptuellt minne är det inte heller klart huruvida allt episodiskt minne skall klassificeras som deklarativt (även om Tulving tycks göra det). Filosofer har nyligen föreslagit en ny kategori av kunskap, nämligen *att veta hur något är*, som komplement till de ovan nämnda, alltså *att veta att* respektive *att veta hur man gör*.[68] Kanske är det mest naturligt att säga att ett åskådligt minne av en stund vid stranden i första hand förmedlar kunskap om *hur* det var att ligga vid stranden – inte *att* det var på ett visst sätt, ett sätt som skulle kunna beskrivas genom ett påstående. Och det vore inte helt onaturligt att föra perceptuellt minne till samma huvudkategori, eftersom episodiskt minne i den ovan föreslagna preciseringen har nära relationer till varseblivning.

Begreppen explicit och implicit inlärning

I kognitionspsykologisk litteratur möter man ofta två parallella distinktioner mellan *explicit* och *implicit* inlärning, respektive mellan explicit och implicit minne. Tyvärr är definitionerna av begreppen ofta mycket oklara, och gränsdragningen skiftar från framställning till framställning. Det enda gemensamma tycks vara att huvudkriteriet för att föra ett inlärnings- eller minnesfenomen till den ena eller den andra kategorin ska vara i vilken mån *medvetna* processer är inblandade. Men begreppet medveten process är notoriskt oklart, och dessutom kan man mena olika saker med att medvetandet ska vara *inblandat* (jämför nedan).

Efter denna kraftfulla reservation vill författaren ändå medge att distinktionerna kan ha ett berättigande. Det är inte orimligt att säga, att semantiska minnen typiskt är explicita, eftersom de oftast har mer med medvetandet att göra – på flera sätt – än vad till exempel procedurell kunskap och betingade reaktioner har. De senare minnesformerna är alltså i typfallen implicita. Hur man ska klassificera episodiskt och perceptuellt minne i detta avseende är oklart för mig; slutresultatet berör medvetandet (vårt upplevande) i högsta grad, men inlärningen är "automatisk" och påminner på många sätt om procedurell inlärning.

[68] Se t.ex. Tye (2000).

Innan man använder termerna explicit/implicit i en vetenskaplig text om minne och inlärning bör man uppenbarligen se till att precisera dem! Författaren har valt att avstå från att använda orden i den följande framställningen.[69]

Vi ska nu övergå till att göra en kort översikt över ett annat mycket komplext och omdiskuterat område. Översikten är bland annat till för att underlätta beskrivningen av typiska minnesstörningar vid hjärnskador, och en mängd kontroverser av mer teoretisk art måste lämnas därhän.

3.3 Omedelbart minne och uppmärksamhetens betydelse

Tidsliga gestalter

Låt oss betrakta hur ett typiskt episodiskt minne uppkommer och vilka öden det sedan genomgår. Du står på en stubbåker en dag i början av april och hör plötsligt en lärka sjunga en kort drill. Eftersom det är vårens första lärka lyssnar du intensivt. Du uppfattar redan en liten bit in i drillen en *tidslig gestalt* av toner; denna gestalt förändras dessutom allteftersom drillen fortsätter att ljuda. Intressant är nu att sekunderna efter det att lärkan slutat sjunga, men medan du fortfarande är uppmärksam på sången, så är det ändå som om du fortfarande hörde den: drillen ljuder en kort stund i ditt inre nästan lika tydligt – och med samma gestalt – som vid slutet av den sista tonen. Detta åskådliga, för att inte säga kvasi-sensoriska, kvardröjande av den akustiska gestaltupplevelsen kan man klassificera som tillhörigt det *omedelbara minnet*. För alternativa termer se nedan.

I en del äldre psykologisk och filosofisk litteratur används termen ”retention” gärna *antingen* för det omedelbara minnet *eller* för den allra ”närmaste” delen av detta (jämför nedan om sensoriska buffertar). Numera använder man oftare ”retention” för allt kvarhållande av ett minne, även på längre sikt och i andra former, och vi skall ansluta oss till detta språkbruk. Det omedelbara minnet definierar vi som *den åskådliga retentionen inom ramen för ett oavbrutet uppmärksamt medvetande.*

Varför är en sådan definition lämplig? Jo, antag att lukten från köket får dej att släppa uppmärksamheten på lärkans sång och istället fundera över

[69] För en aktuell diskussion om implicit minne se French (utg.) (2002).

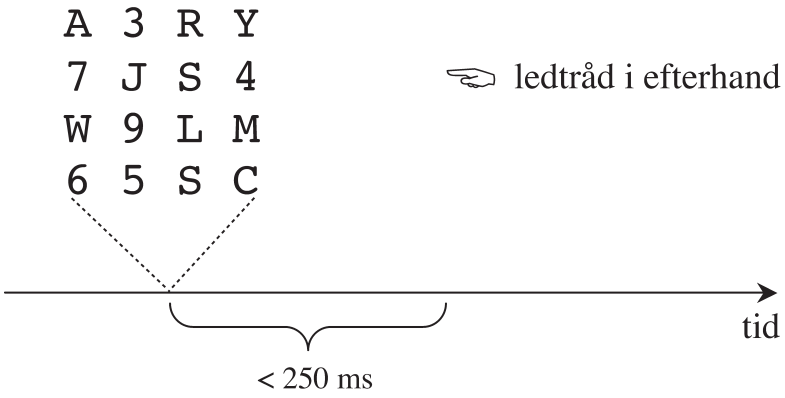
vad det ska bli till middag. Den åskådliga akustiska gestalten ersätts då abrupt av en kulinarisk smakfantasi. Visserligen kan du i normalfallet återkalla ett åskådligt episodiskt minne av drillen en stund senare eller kanske rentav långt i efterhand, men den neuropsykiatriska erfarenheten visar entydigt att mekanismen för ett sådant senare återkallande måste vara en annan än den minnesmekanism som var verksam medan uppmärksamheten bibehölls på drillen. Patienter med vissa former av hjärnskador uppvisar nämligen en total förlust av förmågan att återkalla även ganska nyinlagrade episodiska minnen som faller utanför den obrutna uppmärksamhetens räckvidd. Det är ju också en vardaglig erfarenhet att man, även om man med lätthet håller kvar samma tanke i minnet en lång stund, kan ha stora svårigheter att återerinra sig den *föregående* tanken man hade (innan man senast flyttade uppmärksamheten). Vi ska utveckla detta mer nedan, men redan nu kan vi använda den kliniska och vardagliga kunskapen till att motivera en gräns för begreppet omedelbart minne just vid den kontinuerliga uppmärksamhetens slut.

Sensoriska buffertar

Innan vi går vidare skall det också nämnas att många teoretiker vill skilja ut en mycket tidig (eller, om man vill, sen)[70] fas av det omedelbara minnet som alldeles speciell. Man kan här använda termen *sensorisk buffert.*

Sperlings berömda experiment på 1960-talet tycktes visa att den allra första representationen av stimuli innehåller information som är åtkomlig endast under en mycket kort stund; hjärnan ”väljer” sedan vad som ska bevaras, och när urvalet väl skett är det som ”valts bort” inte längre tillgängligt. Experimenten tillgår så, att en liten matris av bokstäver visas i sin helhet under mycket kort tid; försökspersonen har som uppgift att tala om vilka bokstäverna i en viss rad var, men vilken rad det gäller anges inte förrän strax *efter* expositionen. I figur 10 illustreras detta genom en pil som markerar raden ifråga just efter det att matrisen med bokstäver släckts:

[70] Tidig, om man följer en och samma stimulus öde medan den passerar genom nervsystemet; sen, om man betraktar det omedelbara minnets innehåll vid en given tidpunkt och frågar sig när just denna stimulus blev representerad där. Vi väljer i texten det förstnämnda betraktelsesättet.



Figur 10. Sperlings experiment. Förklaring: se text.

Det visar sig att försökspersonen som regel kan någorlunda korrekt återge den utpekade raden, oberoende av vilken denna är, men om hon får uppgiften att återge *alla* raderna, så klarar hon inte alls detta. Enligt Sperlings förklaring, som blivit rätt allmänt accepterad, visar det första fyndet att *hela* materialet lagras en kort stund i den sensoriska bufferten, medan det andra fyndet beror på att återerinringsprocessen tar för lång tid: en del av materialet har redan hunnit försvinna innan återerinringen är klar. Vilken relevans Sperlings experiment har för vår förståelse av senare faser av det omedelbara minnet är oklart, men givetvis måste varje minnesteori ta hänsyn till hans resultat. De visar ju inte minst, liksom de nyss nämnda fakta om det omedelbara minnet, på *uppmärksamhetens* stora betydelse för retentionen.

Sperling använde termen ”visuell buffert” för den tidiga lagringsmekanismen. Termen används inte så mycket i dag; istället talar man numera oftare om *ikoniskt minne*.[71] Motsvarigheten på det auditiva området, den auditiva bufferten, kallas numera oftast *eko-minne (echoic memory)*. Åsikterna om den auditiva buffertens varaktighet går i sär bland forskare, men en vanlig uppfattning är att den är betydligt större än den visuella buffertens. Två sekunder är en vanlig uppskattning. Det skulle betyda att en liten melodigestalt, kanske ett par toner av lärkans sång, får plats i den auditiva bufferten. Det åskådliga, omedelbara minnet av sången har betydligt längre räckvidd, och sträcker sig alltså en bra bit utanför den auditiva bufferten.

Många filosofer har intresserat sig för de sensoriska buffertarna, exempelvis fenomenologins grundare Edmund Husserl. Husserl använder termen ”retention” för den sensoriska bufferten och menar att den inte är

[71] Se t.ex. Massaro & Loftus (1996).

en form av minne, utan snarare något mitt emellan varseblivning och minne. Både fenomenologer och empiristiska filosofer som Bertrand Russell har ofta hävdat att de sensoriska buffertarna utgör vår ursprungliga tillgång till det förflutna. Även för en ANN-teoretiker erbjuder fenomenet en intellektuell utmaning. Hur är upplevelsen av det omedelbart förflutna materialiserad i nervsystemet? Frågan kommer att ägnas ett särskilt avsnitt (10.4) i samband med diskussionen av representation av tid i neurala nätverk. Författaren kommer där att lägga fram en ANN-hypotes som harmonierar med de just nämnda filosofernas tankar.

Repetition och minne

En annan uppsättning fakta som visar på uppmärksamhetens betydelse har att göra med effekten av repetition. Det som avses här är ”repetition i huvudet”, inte upprepad läsning eller annan upprepad stimulering. Att hålla kvar ett material i det omedelbara minnet länge leder till bättre retention senare, men också att mentalt repetera det med större mellanrum. Detta är ju en vardagserfarenhet och något som vi alla använder oss av, t.ex. för att bättre komma ihåg det som vi föresatt oss att göra under dagen.

Ett vackert experimentresultat med samma tendens är effekten av *seriell position*.[72] Experimentet går ut på att man presenterar en serie meningslösa stavelser för en person, och därefter presenterar dem en och en i slumpvis ordning och låter försökspersonen svara på om hon sett respektive stavelse. Efter ett stort antal sådana försök kan man skatta hur sannolikheten för hågkomst av en stavelse är relaterad till stavelsens position i serien. Det visar sig, inte oväntat, att de stavelser som presenterats sist i serien är de som försökspersonerna minns bäst. Mindre väntat är kanske, att försökspersonerna också minns de *först* presenterade stavelserna mycket bra, medan de som presenterades i mitten tycks vara de som lättast glöms! En vanlig förklaring (vilket inte betyder att förklaringen accepteras av alla) av detta fenomen är att de först presenterade stavelserna får flest chanser att repeteras mentalt, dvs. gör flest ”besök” i uppmärksamhetens fokus.

Varför mental repetition leder till bättre retention är en fråga av stort intresse, som vi här bara ska kommentera genom att nämna att simuleringsteorin om åskådliga föreställningar öppnar en möjlighet till förklaring.

[72] Jämför Bower & Hilgard (1981), ss. 138ff.

Det gör den nämligen om man antar att inlärning kan äga rum inte bara under verklig input, utan även under simulerad sådan. Ett sådant antagande skulle ha vittgående konsekvenser på flera andra områden, och skulle behöva kvalificeras på många sätt, men det skulle föra för långt att gå in på dessa konsekvenser och reservationer här.

3.4 Korttids- och långtidsminne; arbetsminne

STM och LTM

I de flesta äldre läroböcker i kognitiv psykologi (före 1980) görs en distinktion mellan *korttids-* och *långtidsminne* (STM respektive LTM). Korttidsminnet bestäms då ibland som sammanfallande med det omedelbara minnet enligt vår definition ovan (alltså minnet inom ramen för den obrutna uppmärksamheten), men karakteriseras vanligen som arbetande i ett längre tidsperspektiv än detta; ofta talar man om några minuter (medan det omedelbara minnet för det mesta varar i upp till några tiotal sekunder). I denna senare, vanligare innebörd av ”korttidsminne” innefattar begreppet alltså även förmågan att återerinra sig material som nyss försvann ur det omedelbara minnet. Däremot innefattar det inte hågkomst av äldre material; där träder långtidsminnet in.

En fråga som diskuterats mycket är om det verkligen finns någon intressant *kvalitativ* skiljelinje mellan å ena sidan korttidsminnet i den nu avgränsade meningen, å andra sidan långtidsminnet. Om man till exempel hade kunnat påvisa helt skilda anatomiska substrat för minnen som uppstod för två minuter sedan, respektive för tjugo minuter sedan, hade distinktionen uppenbarligen haft ett stort värde. Det finns dock inga goda empiriska belägg vare sig för en sådan anatomisk uppdelning eller för existensen av någon annan principiell gräns just mellan korttids- och långtidsminne, som dessa begrepp vanligen defineras. Det finns utan tvekan systematiska, gradvisa skillnader mellan minnen som lagrades för länge sedan och sådana som kom till mer nyligen. Men skall man tala om en kvalitativ gräns någonstans, talar empiriska fakta (särskilt från neuropsykiatrin) snarare för att den går *mellan omedelbart minne och övrigt korttidsminne*. Mer om detta nedan.

Arbetsminnet

För tjugofem år sedan lanserade Alan Baddeley en detaljerad teori om det som han kallar *arbetsminnet.*[73] Denna teori dominerar nu de kapitel i böcker om kognitiv psykologi som förut handlade om STM. Även Baddeleys teorier har utlagts på olika sätt: som en teori om det omedelbara minnet (dvs. inom den kontinuerliga uppmärksamhetens ram), eller som en teori om korttidsminnet i vidare mening. Oavsett Baddeleys intentioner på denna punkt – och kanske är det så att hans huvudintresse inte är inriktat på frågan om den tidsliga räckvidden – är det intressant att notera att han skiljer ut tre, eller numera till och med fyra, subsystem inom arbetsminnet. Det mest innovativa inslaget i hans teori är att han räknar in medvetandets högsta kontrollinstans, som han kallar ”Central Executive”, i arbetsminnet. Denna centrala exekutiv (CE) har två ”slavsystem” till sin hjälp, varav ett av fonologisk-auditiv natur och ett visuo-spatialt. Det första slavsystemet har nyligen försetts med ett eget subsystem. – Dessutom har CE tillgång till material som lagrats i långtidsminnet.

Begreppet *central executive* hos Baddeley fyller många funktioner som i äldre teorier fylldes av begreppet *uppmärksamhet*, och det är som redan antytts känt sedan länge att uppmärksamheten är inblandad i det intima samspelet mellan varseblivning och minne. Baddeleys teorier gör dock bättre reda för detaljer i denna interaktion än vad många av de tidigare teorierna klarade av. Trots det skall vi nedan försöka klara oss med de gamla beprövade begreppen så länge det går, och inte använda begreppen arbetsminne och central executive mer än tillfälligtvis.

Baddeleys teorier är för övrigt långt ifrån okontroversiella, och det finns många konkurrerande teorier och begreppsbildningar som alla använder sig av termen ”arbetsminne”.[74] Bland annat har en forskargrupp introducerat ett begrepp *långtidsarbetsminne,* som sammanfattar de funktioner som är involverade när man bearbetar relevant minnesinformation under en sammanhållen sekvens av kognitiva tillstånd.[75] Ett typexempel kan vara när man utför en komplicerad arbetsuppgift som består av flera av varandra beroende moment, eller för att anknyta till vardagen, när vi går ut och handlar mat utan att ta med någon minneslapp. Författaren menar

[73] För en tidig version av teorin se Baddeley (1987). För en mycket aktuell version, jämför Baddeley (2007).
[74] Tack till Helene Karlqvist, som för några år sedan skrev en uppsats i kognitionsfilosofi med titeln ”Verbalt arbetsminne och artificiella neurala nätverk”. Nästa referens har hämtats från den.
[75] Ericsson & Kintsch (1995).

att en sådan funktionell bestämning av begreppet är lämplig, eftersom den poängterar kontroll och framtagande av information och inte säger något om tidsrymdens storlek. I princip borde begreppet kunna innefatta handlingssekvenser som hålls samman mentalt över mycket lång tid. Samtidigt sammanfaller begreppet långtidsarbetsminne inte med det klassiska begreppet långtidsminne. Den senare termen syftar på all den minnesinformation som är lagrad under längre tid, snarare än på ett särskild typ av tillgänglighet för minnesinformation.

3.5 Minnestest och minnesmekanismer

Igenkänningstest och återgivningstest

I neuropsykologin och neuropsykiatrin testas minnet på olika sätt. En viktig distinktion är den mellan *igenkänningstest* (tests of recognition) och *återgivningstest* (tests of recall). Det är allmänt sett en lättare uppgift att känna igen, dvs. att svara ”Ja” eller ”Nej” på frågan om det nu presenterade materialet har visats tidigare, än att återge, dvs. att tala om vad som visats tidigare utan att något material presenteras nu. Man talar också om ”återgivning med ledtrådar” (*cued recall*). Det innebär att en del av det presenterade materialet visas som ledtråd för återgivningen. Återgivning med ledtrådar kan också kallas *associativ återgivning*. Det material som presenteras i minnestest är av skiftande natur; för återgivningstest inklusive associativ återgivning används ofta verbalt material (i auditiv eller visuell form), medan igenkänningstest oftare använder sig av bilder av olika ting. Man skall inte underskatta betydelsen av presentation av verkliga föremål snarare än bilder. En sådan presentation kan användas för både igenkänning och återgivning.

De olika sätten att undersöka minnet kan vara olika värdefulla vid olika minnesdefekter. I resten av detta avsnitt ska vi anlägga några principiella synpunkter på hur resultat av minnestest bör, och inte bör, tolkas.

Minnestest och minnesmekanismer

Vi nämnde redan ovan, att man inte får dra förhastade slutsatser från karaktären hos de minnestest man använder till egenskaper hos de minnessystem vars funktioner man försöker komma åt genom testen ifråga. Ett vanligt och mycket användbart test i neuropsykiatriska sammanhang är

således det s.k. ”femsaksprovet”: man visar fem små vardagsföremål för patienten, täcker över dem med en duk och frågar sedan patienten vilka föremålen var. Frågan kan ställas omedelbart, och/eller efter (t.ex). trettio minuter med mellanliggande samtal om något annat. Testet skall i det senare fallet inte slentrianmässigt beskrivas som ”test av långtidsminnet” utan som ”test av retention efter 30 minuter, med distraktion”. Den senare beskrivningen antyder inte, som den förra gör, att man tror att begreppet ”långtidsminne” står för en distinkt funktion hos hjärnan. Och även om man tror att omedelbart minne är en distinkt sådan funktion bör man inte beskriva det första sättet att använda femsaksprovet som ”test av det omedelbara minnet” utan helt enkelt som ”test med omedelbar återgivning”.

Minnestest och ospecifika mekanismer

Ännu viktigare är att man inser att *minnesprov inte bara prövar minnesmekanismer.* Våra minnesprestationer är nämligen beroende av diverse allmänna psykiska mekanismer, som var och en har betydelse också för flera andra mentala prestationer, och som inte rimligtvis kan sägas tillhöra just *minnet.* Exempelvis gör en patient nästan alltid bättre ifrån sig på ett minnestest när hon är vaken än när hon är sömnig, och bättre när hon är starkt motiverad för undersökningen än när hon inte är intresserad av den. Visserligen kan man beskriva skillnaderna som så, att den sömniga och omotiverade patienten *lär sig* och *minns* sämre än den vakna och motiverade, men att i en klinisk situation rapportera den sömniga och omotiverade patientens resultat som att hon har en *minnesstörning* kan vara vilseledande eftersom det för tankarna till mycket mer specifika störningar av minnesförmågan. Resultatet skall givetvis rapporteras som en *nedsatt prestation på minnestestet* i fråga, men neuropsykologen skall i detta fall också tillägga att nedsättningen sannolikt är beroende på ospecifika faktorer.[76]

Det är inte så vanligt att dylika ospecifika faktorer undersöks med psykologisk testmetodik, och ibland finns inte heller någon anledning till detta. Sömnighet och nedsatt motivation kan i allmänhet bedömas kliniskt, och en sådan klinisk bedömning måste alltid göras för att testresultaten skall kunna tolkas rätt. Vår slutsats blir att *neuropsykologen alltid måste kommentera resultatet av minnestestet utifrån sin kliniska helhetsbedömning.* Det finns i dag ”processorienterade” testförfaranden att tillgå, som

[76] Jämför Howieson & Lezak (1995) samt Lindqvist & Malmgren (1990), ss. 57ff.

ger en bättre bild av orsakerna till en nedsatt prestation än vad många traditionella test och testbatterier gör. Men man kan nog lugnt påstå att även dessa nya procedurer behöver en klinisk helhetsbedömning som ram för att resultaten skall kunna tolkas rätt.

Uppmärksamhetens betydelse – igen

En särskilt viktig familj av psykiska mekanismer som påverkar minnesprestationen är, som vi redan påpekat, de som har att göra med uppmärksamhet. Både inlärning (”lagring”, eller ”kodning” i en del kognitiva psykologers terminologi) och återerinring är starkt beroende av uppmärksamheten. Till vardags såväl som i kliniken finns otaliga exempel på hur en störd uppmärksamhet kan försämra minnesprestationen. Förutom de vardagsexempel som givits ovan kan vi tänka på hur svårt det är att lära sig något på en föreläsning om man är distraherad av en svår tandvärk, och hur svårt det kan vara att komma ihåg att göra dagens småärenden efter jobbet om man fortfarande är mycket koncentrerad på en komplicerad uppgift i arbetet och tar med sig den i huvudet på vägen hem.

En klok neuropsykolog minnestestar därför inte en patient utan att försäkra sig om att patienten är så koncentrerad som möjligt på uppgiften. Ännu klokare är psykologen om hon också gör en oberoende undersökning av patientens koncentrationsförmåga, och särskilt då av *uppmärksamhetens uthållighet*. Detta är framförallt relevant vid misstanke om det vi kallar ”asteno-emotionellt syndrom” (se vidare avsnitt 3.8 nedan). Patienter med detta syndrom har en bristande uthållighet i sin uppmärksamhet, och nedsättningen av deras minnesprestationer är tydligt beroende av detta förhållande. Även i dessa fall bör neuropsykologen undvika att tala om en ”minnesstörning” utan att närmare specificera den sannolika mekanismen.

3.6 Sekundära minnesstörningar

Av det sagda följer att de störningar av minnesprestationer som är sviter av hjärnskador och hjärnsjukdomar – dvs. som i en mycket handfast mening är *organiskt* betingade – inte alltid beror på att specifika minnessystem är skadade.[77] I själva verket torde den vanligaste formen av minnes-

[77] Framställningen i detta och de närmast följande avsnitten är om inte annat sägs baserad på Lindqvist & Malmgren (1990). Observera att vår avgränsning av vissa psyki-

störning, nämligen den man ser vid det asteno-emotionella syndromet (se vidare nedan), huvudsakligen vara orsakad av nedsatt förmåga till uthållig uppmärksamhet. I dylika fall kan man klassificera minnesstörningen som *sekundär*. Det betyder helt enkelt att den mekanism som primärt är skadad eller funktionsnedsatt inte *bara* har med minnet att göra, men ändå leder till en nedsatt minnesprestation.

Andra sekundära minnesstörningar ser man bl.a. vid organiskt betingade nedsättningar av patientens motivation, inte minst då vid det tillstånd som i klassisk psykiatri (dvs. före DSM-III, den första versionen av det modernaste amerikanska systemet för klassifikation av psykiska sjukdomar)[78] oftast kallades ”frontallobssyndromet”. Termen är olycklig eftersom syndromet förekommer ganska ofta vid skador i tinningloberna och mellanhjärnan; en tillräcklig betingelse för dess uppkomst är nämligen att det drabbar någon av strukturerna i det s.k. limbiska systemet.[79] Den etiologiskt neutrala beteckningen ”emotionellt-motivationsmässigt personlighets-syndrom”, med förkortningen *EM-syndrom*, är därför att föredra.

Vid EM-syndrom ser man en reduktion av patientens initiativ och motivation, och denna kan visa sig antingen generellt eller på enstaka områden av livet. Parallellt sker en förändring av känslolivet så att det uppstår en förflackning och insnävning av de emotionella reaktionerna; patienten blir i svåra fall okänslig för allt utom sina egna omedelbara behov. Man ser vidare en oförmåga att planera för framtiden samt vissa mer subtila kognitiva förändringar, som bl.a. yttrar sig som en oförmåga att förstå metaforer. EM-syndromet, som kan uppstå på grund av många olika orsaker men är en vanlig följd av tumörer och andra lokala processer i frontolimbiska områden, är inte sidobundet utan ses i lika svåra former vid vänster- och högersidiga skador. Motivationsnedsättningen kan bland annat yttra sig som ett dåligt resultat på ett minnestest, och då bör neuropsykologen veta hur tillståndet testmässigt ska differentieras från andra typer av minnesstörningar. Det kan nämligen hända att dessa patienter *också* har en annan och mer primär form av minnesstörning (se nedan).

---

atriska störningar som ”organiskt betingade” inte implicerar att övriga psykiska störningar saknar underlag i hjärnan. Jämför Malmgren (2005), (2007).

[78] DSM betyder ”Diagnostic and Statistical Manual of Mental Disorders” och den senaste upplagan (2000) heter DSM-IV-TR. Se American Psychiatric Association (1980) respektive (2000).

[79] Att det klassiska frontallobssyndromet inte förutsätter frontala skador har varit känt länge i psykiatrin.

För beskrivningar av andra kliniskt viktiga tillstånd där man ser sekundära minnesstörningar, se den första referensen i not 77.

Därmed är tiden mogen att berätta närmare om de två viktigaste formerna av minnesstörningar i den organiska psykiatrin, nämligen *Korsakoffs amnestiska syndrom* och *asteno-emotionellt syndrom*.

3.7 Korsakoff och hans syndrom

Terminologiska överväganden

Vi ska nu beskriva ett tillstånd som vi kallar *Korsakoffs amnestiska syndrom.* Detta tillstånd betecknas i den nyare litteraturen oftast som *amnestiskt syndrom*. ”Amnesi” betyder förvisso minnesförlust, men beteckningen är ändå olycklig. Dels finns det flera andra viktiga former av amnesi, dels saknas anknytning till syndromets upptäckare Korsakoff, som under 1880-talet beskrev tillståndet på ett än i dag giltigt sätt.[80]

I bland annat svensk psykiatri har man sedan länge reserverat beteckningarna ”Korsakoffs syndrom” och ”Korsakoffs psykos” för amnesier utlösta av kroniskt alkoholmissbruk. Men detta är historiskt oegentligt, eftersom inte alla de patienter som Korsakoff själv beskrev hade ett alkoholbetingat tillstånd. Det finns inte heller någon kliniskt betydelsefull skillnad mellan de Korsakoff-amnesier som orsakas av alkoholmissbruk och de som beror på exempelvis traumatisk hjärnskada, tumör, herpesencefalit (en form av hjärninflammation), syrebrist eller Alzheimers sjukdom – för att bara nämnda några av de vanligare orsakerna till det tillstånd, som vi här alltså med ett gemensamt namn kallar Korsakoffs amnestiska syndrom, eller kort *KA-syndrom*.[81] Inte heller finns det några stora skillnader mellan den amnesi som orsakas av skador i båda sidors hippocampus – jämför den berömda patienten HM (se not 10 ovan) som fick sina hippocampi bortopererade p.g.a. mycket svår epilepsi – och de som uppkommer vid andra bilaterala skador i det så kallade limbiska systemet, exempelvis främre thalamus eller mammillarkropparna (en klassisk skadelokal för alkoholrelaterad Korsakoff).[82]

---

[80] Se t.ex. Korsakoff (1890).

[81] En helt annan sak är att t.ex. Alzheimers sjukdom som regel också ger upphov till en mängd *andra* symptom än de amnestiska. Se vidare avsnitt 3.10.

[82] Jfr. Victor, Adams & Hope (1971).

Retrograd och anterograd amnesi

Vid ett KA-syndrom har patienten en *retrograd* och en *anterograd* amnesi. Den retrograda amnesin, populärt ”minnesluckan”, består i att patienten inte har några (eller bara har sporadiska) minnen från en viss period, som alltid sträcker sig fram till och med skadetillfället (insjuknandet). Denna minneslucka, som i första hand drabbar episodiska minnen, kan vara av högst olika längd beroende på tillståndets svårighetsgrad. Vid ett svårt fall hos en femtioårig patient kan den sträcka sig trettio år tillbaka i tiden, vilket betyder att hon endast har minnen från tjugoårsåldern och tidigare. I lätta fall kanske minnesluckan bara omfattar ett par timmar. I de fall då tillståndet senare förbättras krymper minnesluckan ”bakifrån”, dvs. vår 50-åriga patient börjar åter komma ihåg händelser från sin tjugofemårstid, sedan från trettioårsåldern, etc. I de flesta fall av förbättring blir dock en viss minneslucka för tiden omedelbart före skadetillfället kvar för gott. Till och med vid de lätta fall av KA-syndrom som uppkommer vid lindriga traumatiska hjärnskador (t.ex. vid trafikolyckor) är det vanligt att något minne av själva olyckstillfället aldrig återkommer.

Den anterograda amnesin[83] innebär att patienten efter skadan/insjuknandet har svårigheter att lära sig nytt material. Visserligen fungerar det omedelbara minnet ofta perfekt, så att patienten t.ex. kan återge en rad med siffror direkt efter det att den presenterats för henne, men om uppmärksamheten flyttas så tycks minnesmaterialet inte längre vara tillgängligt. Om den som undersöker patienten går ut ur rummet och strax kommer in igen, måste hon alltså i allmänhet presentera sig igen. Det ovan nämnda femsaksprovet ger därför dramatiskt olika utslag vid test med omedelbar respektive fördröjd återgivning.

Om och när en patient förbättras radikalt händer det att hon minns händelser från tiden för den anterograda amnesin, trots att hon inte kunde återerinra sig dem under sjukdomstiden. Det normala är visserligen att hela sjukdomsperioden för alltid är förlorad för minnet, men det faktum att vissa episoder ibland återerinras i efterhand visar att den anterograda amnesin inte *bara* kan bestå i en oförmåga att *lagra in* (koda) minnen. Det måste finnas en defekt på återerinringssidan också. Beträffande den retrograda amnesin är det uppenbart att det är fråga om en erinringsdefekt och att (de allra flesta) minnena från tiden för amnesin inte är utraderade, bara (mer eller mindre temporärt) oåtkomliga.

[83] Hos Lindqvist och Malmgren (1990) klassificerad som en ”närminnesstörning”.

Praktisk inlärning, med mera, vid KA-syndrom

Forskning från 1960-talet och framåt har visat att den anterograda amnesin inte hindrar all inlärning av *praktiska färdigheter*. Patienter kan exempelvis lära sig att spela ett spel genom att träna på det vid upprepade och i tid separerade träningstillfällen, trots att de vid varje tillfälle anger att de aldrig sett spelet förut.[84] Detta har ibland beskrivits som en dissociation[85] mellan deklarativt och procedurellt minne, men det är möjligt att den viktigaste gränsen går mellan episodiskt och icke-episodiskt minne. Man har nämligen också funnit att åtminstone en del patienter kan lära sig associationer till meningslösa stavelser, utan att de i testsituationen kan minnas att man tidigare har visat stavelserna för dem. Denna förmåga till associativ inlärning är visserligen också nedsatt, men inte alltid lika mycket som förmågan att bilda episodiska minnen från inlärningssituationerna. Det händer vidare att patienterna vet en hel del om händelser i sitt tidigare liv utan att ha några direkta, episodiska minnen av det. Procedurellt minne, associativ inlärning, semantiskt minne och episodiskt minne drabbas alltså på olika sätt vid KA-syndrom, och säkert kan olika fall av KA-syndrom ha något olika profil i dessa avseenden.

Läser man Korsakoffs egna uppsatser noga finner man att ingenting är nytt under solen. En av hans amnestiska patienter utvecklade under sjukdomstiden gradvis en betingad aversion mot en viss behandlingsapparat – fastän hon inte vid något av behandlingstillfällena kom ihåg att hon sett apparaten förut.[86] Det är alltså Korsakoff som borde krediteras för upptäckten av den nämnda dissociationen mellan minnesformer!

Desorientering och konfabulation

Det säger sig självt, att den retrograda och den anterograda amnesin tillsammans alltid leder till en svår *desorienteringstendens*. Patienten vet som regel vare sig var hon är eller vilken dag det är. Ett annat fenomen som man ofta kan iaktta är *konfabulation*. Patienten berättar om händelser och sammanhang som uppenbarligen inte hör verkligheten till, kanske för att ”fylla ut” sina minnesluckor. Graden av konfabulation tycks dock inte vara kopplad på något enkelt sätt till graden av minnesstörning.

---

[84] Warrington & Weisskrantz (1982).

[85] Termen ”dissociation” används här för det faktum att en psykisk funktion men inte en annan har fallit bort vid skadan, och har inget med dissociation i psykiatrisk mening att göra.

[86] Korsakoff (1890).

Samband med motivationsstörningar

Ett KA-syndrom, särskilt ett svårt sådant, åtföljs nästan alltid av ett mer eller mindre uttalat EM-syndrom. Detta är kanske inte så underligt, eftersom de anatomiska förutsättningarna för de två syndromen är så lika. Observera dock att medan EM-syndromet kan utvecklas till fullo även från en enkelsidig skada, så kräver Korsakoffs amnestiska syndrom en dubbelsidig sådan. Patienter som fått ena sidans hippocampus bortopererad kan ha ett i det närmaste fullgott minne. Det verkar med andra ord som om de två hjärnhalvornas limbiska minneskretsar kan fungera som backup för varandra. Vid modern epilepsikirurgi är man ytterst noga med att kontrollera att den tinninglob som man *inte* skall operera i fungerar som den skall. Komplikationer som de som HM drabbades av uppkommer därför inte i dag.

Den genom EM-syndromet nedsatta motivationen kan förvisso inte förklara minnesproblemen vid KA-syndrom. Inte heller är det fråga om att minnessvårigheterna skulle vara sekundära till koncentrationssvårigheter; patienter med ett ganska svårt KA-syndrom kan ofta ha en förvånansvärt god koncentrationsförmåga. Överhuvudtaget gör minnesstörningen vid KA-syndrom intryck av att vara en självständig, primär defekt i ett minnessystem. Att förklara vad denna defekt i grunden består i är en värdig uppgift för forskningen om neurala nätverk.

KA-syndrom och andra amnesiformer

Det är inte särskilt svårt att skilja ett uttalat KA-syndrom från andra typer av minnesstörningar. Den anterograda amnesin är påtaglig och av en sådan grad att eventuella samtidiga motivations- eller koncentrationsdefekter inte rimligtvis kan förklara den. Test med exempelvis bildigenkänning visar i allmänhet på en kraftig konfabulationstendens. Minnesluckan är också typisk, och något som liknar den ser man annars bara i de sällsynta fall när en person inte minns något alls av sitt förflutna och upplever sig ha helt ”förlorat sin identitet”. Om patienten inte vet vem hon är (och alltså inte heller sitt namn), men i övrigt verkar någorlunda intellektuellt bibehållen, är det dock sannolikt inte fråga om ett KA-syndrom eftersom minnesluckan vid ett sådant mycket sällan sträcker sig ända till tidig barndom. Den högst sannolika diagnosen hos en patient som helt förlorat sin identitet är istället *psykogen* amnesi.

3.8 Asteno-emotionellt syndrom

Inledning

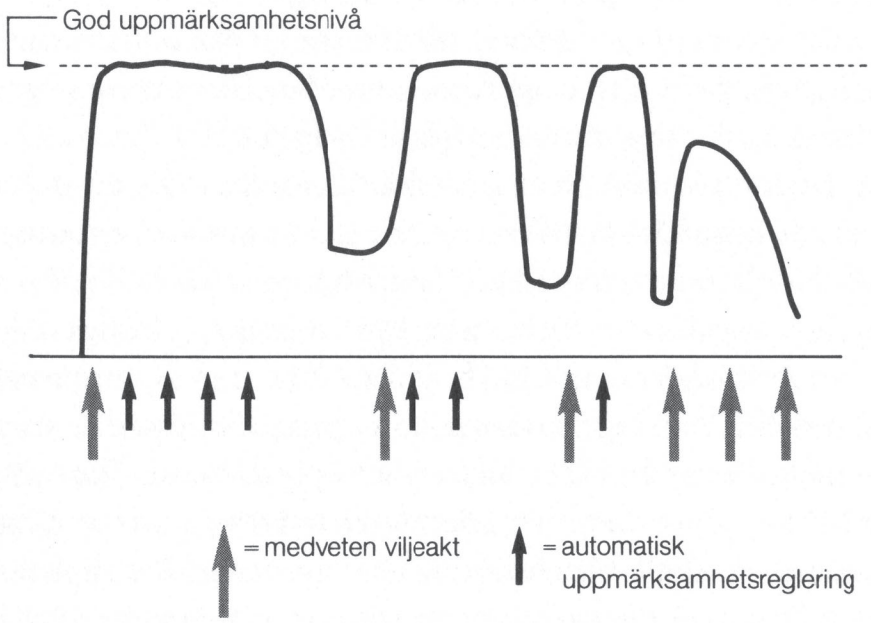
På det område som vi nu ska behandla är den terminologiska situationen om möjligt ännu mer oklar än i övriga delar av den organiska psykiatrin. Termen ”asteno-emotionellt syndrom” (med förkortningen ”AE-syndrom”) är relativt ny, och begreppet erkänns än så länge inte av särskilt många forskare eller praktiker.[87] Men den *verklighet* det täcker är välkänd av alla psykiatrer som arbetar på det organiskt-psykiatriska fältet. Visserligen är den följande framställningen centrerad just kring begreppet asteno-emotionellt syndrom, men även den som föredrar att arbeta med andra aktuella, närliggande begrepp som *mild cognitive impairment, mild neurocognitive disorder* och/eller *dysexekutivt syndrom* bör kunna tillgodogöra sig den följande framställningen.

Vid hjärnskador och hjärnsjukdomar av de mest varierande slag och lokalisationer – traumatiska skallskador, tumörer, infektioner, degenerativa sjukdomar i tidigt skede, etcetera – ser man mycket ofta en symptomkonstellation bestående av *koncentrationssvårigheter, psykisk uttröttbarhet, minnessvårigheter, emotionell labilitet* och *irritabilitet.* Koncentrationssvårigheterna har framförallt karaktären av en mer eller mindre uttalad svårighet att bibehålla uppmärksamheten på samma uppgift under längre tid. Förmågan till maximal koncentration under kort tid (och därmed till intellektuella topprestationer) behöver inte vara nedsatt; det är uthålligheten som drabbas.

Den typiska uppmärksamhetsstörningen

När en person med AE-syndrom tar sig an en kognitiv uppgift – t.ex. att läsa en tidningsartikel – går det till en början bra, men snart ger den automatiska kontrollen av uppmärksamheten vika. Patienten måste då medvetet anstränga sig för att hålla uppmärksamhetsnivån uppe. Den automatiska kontrollen sviktar dock oftare och oftare, och de upprepade medvetna ansträngningarna leder till en trötthetskänsla. Denna kan bli svår och leda till att arbetet med uppgiften måste avbrytas. Förloppet kan illustreras som följer:[88]

[87] För en aktuell avhandling om syndromet se Rödholm (2003).
[88] Efter Lindqvist & Malmgren (1990), s. 144.



Figur 11. Det typiska uppmärksamhetsförloppet vid AE-syndrom. Förklaring se text. Källa: Lindqvist & Malmgren (1990).

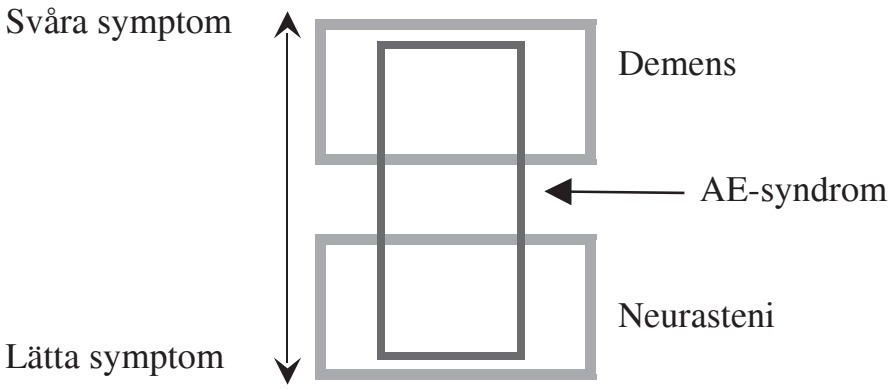
Känner man till denna typiska uppmärksamhetsstörning är diagnosen av ett AE-syndrom oftast inte särskilt svår. Man har också ledning av att minnessvårigheterna vid AE-syndrom drabbar både inlagring och framtagning av minnen. Någon distinkt minneslucka bakåt föreligger inte vid ett renodlat AE-syndrom, och konfabulation vid minnestest är mycket ovanlig. Vidare är den emotionella symptomatologin karakteristisk. Den emotionella labiliteten vid AE-syndrom behöver inte vara särskilt framträdande, men kan i andra fall vara så svår att patienten gråter så snart man t.ex. kommer in på något ämne som rör hans familj. Irritabiliteten kan yttra sig som en överkänslighet för plötsliga ljud och/eller som en retlighet i umgänget med andra personer.

Det är viktigt att inse att störningarna på det emotionella området vid AE-syndrom är av en helt annan karaktär än vid EM-syndrom. Vid det senare ser man en *förflackning och förändring* av emotionerna men vid AE-syndrom ”bara” en *bristande kontroll* av dem. Ett AE-syndrom berör alltså inte på samma sätt som ett EM-syndrom personlighetens kärna.

Liksom vid EM-syndrom finns det inga belägg för att de emotionella störningarna vid AE-syndrom skulle vara lateraliserade till den ena sidan av hjärnan snarare än till den andra.

Några alternativa beteckningar

Har en patient detta syndrom i fullt utvecklad eller partiell form säger vi alltså att hon har ett asteno-emotionellt syndrom, eller AE-syndrom. En anledning till att vi har tyckt att det behövs ett nytt begrepp här är, att få av de alternativa begreppen täcker *både den kognitiva och den emotionella aspekten* av syndromet, fastän de kliniskt sett hänger ihop. Exempelvis innehåller begreppen *mild neurocognitive disorder* och *mild cognitive impairment* inte någon referens till emotionella symptom.[89] Vidare finns det inga andra begrepp som täcker *både lätta, medelsvåra och svåra fall* av syndromet. Lätta fall kan i och för sig klassificeras som ”mild cognitive impairment” (eller kanske som ”neurasteni”, förutsatt att denna term inte reserverats för psykogena tillstånd), och svåra fall kan beskrivas som ”demens”, men genom bytet av term förlorar man den kliniska kontinuiteten mellan tillstånden i sikte. Jämför figur 12.



Figur 12. Förhållandet mellan begreppen AE-syndrom, demens och neurasteni.

Uppkomstmekanismer

AE-syndromet kan som antytts uppkomma på många olika sätt. Mest känt är syndromet antagligen som effekt av hjärnskakning (commotio cerebri), då det oftast benämns ”postkommotionellt syndrom”. Exakt samma symptomatologi ser man dock till exempel vid begynnande hjärntumörer (lokaliserade praktiskt taget var som helst i hjärnan, och ofta långt innan de ger mer specifika neurologiska symptom), och i dessa fall

[89] Om några varianter av begreppet mild cognitive impairment, se Bischkopf et al. (2002).

vore det förstås helt oegentligt att kalla syndromet ”postkommotionellt”. En helt annan typ av orsak är inresekretoriska rubbningar, t.ex. för hög kortisonhalt i blodet. Den varierande etiologin är ytterligare en av de omständigheter som gör att det behövs ett helt nytt begrepp på området. Förhållandet borde också kunna fungera som en nyttig tankeställare för dem som eventuellt tror att det alltid går att koppla neuropsykologiska symptombilder till specifika skadelokaler.[90]

Men de många möjliga etiologierna är också av principiellt intresse för vår förståelse av syndromets hjärnfysiologiska bakgrund, och därmed för ANN-teorin. Vi tror att många av de lätta eller medelsvåra fallen av AE-syndrom kan beskrivas i termer av en *överbelastning av överordnade kontrollmekanismer, huvudsakligen grundad i defekta filterfunktioner*. Här är vår hypotes, än så länge delvis formulerad i kognitiva termer:

I hjärnan försiggår hela tiden en oerhört omfattande signaltrafik, men de högsta kontrollerande centra (som man kan kalla ”centra för uppmärksamhetsreglering” eller ”Central Executive”, alltefter sina teoretiska preferenser) får normalt bara del av sådan information som lägre centra inte klarar av att behandla på egen hand. Om nu något av dessa lägre centra är skadat och har nedsatt funktion, så släpper det igenom mängder av signaler som de högsta kontrollerande instanserna inte är vana att ta hand om. Jämför överkänsligheten för ljus, ljud och emotionella impulser! Detta leder till en överbelastning av de högsta centra, och därigenom till en sviktande funktion även hos dessa. Även andra patologiska signaler från de skadade lägre centra kan bidra till överbelastningen, men de som har med den defekta filterfunktionen att göra torde vara viktigast. – En precisering av denna hypotes i termer av neurala nätverk står högt på författarens önskelista och agenda.

Hypotesen om överbelastning förklarar också varför man kan få ett AE-syndrom på psykogen väg, t.ex. vid svåra smärttillstånd. Allra viktigast att veta är dock att även lätta symptom av AE-typ, exempelvis koncentrationssvårigheter och irritabilitet, ofta har en *organisk* bakgrund, dvs. beror på en hjärnskada, en hjärnsjukdom eller en allmän kroppslig sjukdom som direkt påverkar hjärnans tillstånd (t.ex. en inresekretorisk rubbning). Alltför många patienter har fått gå obehandlade alltför länge bara för att den behandlande läkaren eller psykologen nöjt sig med en (i och för sig kanske inte helt implausibel) hypotes om psykiska orsaker till besvären.

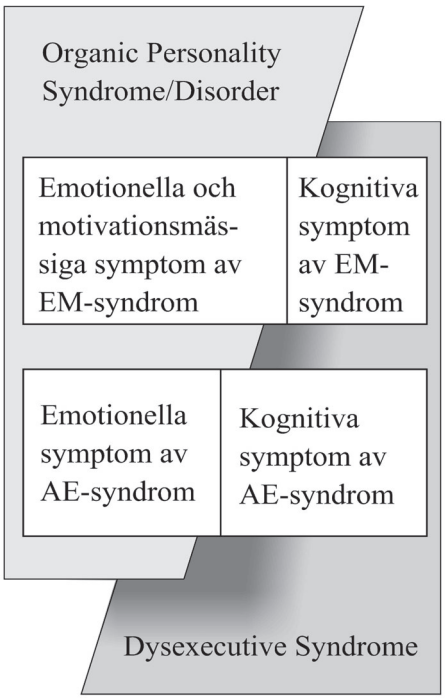
[90] Jfr. här Uttal (2001).

3.9 ”Frontala symptom” och cerebral lokalisation

Några avslutande kritiska reflektioner över termen ”frontal” i neuropsykologin kan också vara på sin plats här. Den klassiska psykiatrin rörde sig som nämnts med ett begrepp *frontallobssyndrom.* Vi påpekade ovan att beteckningen är olämplig, eftersom syndromet ofta utlöses från andra lokalisationer än frontalloberna. De nyare benämningarna ”Organic Personality Disorder” (DSM-III-R)[91] och ”Personality Change Due to a General Medical Condition” (DSM-IV-TR)[92] är inte heller lämpliga eftersom de omfattar en del fall av AE-syndrom (se ovan). Vårt begrepp EM-syndrom har tillkommit mot denna bakgrund.

Det klassiska frontallobssyndromet har under de senaste decennierna kommit i skymundan för en modern neuropsykologisk idé om ett annat ”frontalt syndrom”. En vanlig term i sammanhanget har varit *”dysexekutivt syndrom”*. Detta nya ”frontala” syndrom har definierats på varierande sätt, men de definitioner som man kan hitta i litteraturen lägger ofta tonvikten på kognitiva symptom.[93] Definitionerna är dock ofta oklara med avseende på *vilka* kognitiva symptom som skall hänföras till det dysexekutiva syndromet. Men framförallt är begreppsbestämningen otydlig när det gäller hur mycket av symptombilden vid det *klassiska* frontallobssyndromet som skall räknas in i det dysexekutiva syndromet. Nyligen har denna frågeställning rönt mer uppmärksamhet, och en samtida översikt antyder att två helt olika symptomfamiljer är inblandade.[94] Författaren kan inte annat än hålla med. Sannolikt innehåller *både* begreppet *dysexekutivt syndrom* (i standardtolkningen) och begreppet *Organic Personality Disorder* komponenter från *både* AE-syndrom och EM-syndrom, vilket illustreras i figur 13.

[91] American Psychiatric Association (1987).
[92] American Psychiatric Association (2000).
[93] Se till exempel Rabbitt (red.) (1997) och Roberts et al. (red.) (1998).
[94] Stuss & Levine (2002).



Figur 13. Förhållandet mellan några alternativa klassifikationer av kognitiva och emotionella störningar.

De kognitiva symptom som man ser vid AE-syndrom sägs ibland vara ”frontala”. Och visst ser man AE-syndromet vid de flesta frontala skador (faktiskt mycket oftare än man ser det klassiska ”frontallobssydromet”, alltså EM-syndromet), men det uppkommer lika regelbundet vid andra skadelokalisationer, liksom vid diffus påverkan på hjärnan på grund av till exempel störd inre sekretion. Är det då rimligt att kalla symptomen ”frontala”?

Ja, i en *viss* mening kanske det är rimligt, men i en annan mening inte. Det är nämligen inte omöjligt att många av symptomen vid AE-syndrom *är* ”frontala” i den meningen, att de överordnade kontrollmekanismer vars dysfunktion omedelbart förklarar symptomen i stor utsträckning är lokaliserade till frontalloberna. Dessa är ju trots allt fylogenetiskt sena strukturer, där man à priori kan förvänta sig att de högsta psykiska funktionerna har sitt säte. Men vi har som nämnts också stor anledning att förmoda att funktionssvikt hos dessa mekanismer mycket ofta utlöses av skador i andra delar av hjärnan. Så även om en *stor* del av den *omedelbara* förklaringen av symptomen vid AE-syndrom antagligen ligger i dysfunktion hos frontala mekanismer – *hur* stor del är alldeles för tidigt att gissa – så är de *bakomliggande* orsakerna till det mycket ofta att söka nå-

gon annanstans i hjärnan, utanför frontalloberna. Huvudsakligen av den anledningen, men också eftersom vi faktiskt inte *vet* i hur stor utsträckning de inblandade kontrollmekanismerna är frontalt lokaliserade, bör AE-syndromet inte kallas ett ”frontalt” syndrom.

Psykiatern och neuropsykologen måste alltså vara mycket öppna när det gäller *skadans* lokalisation vid ett AE-syndrom eller ett ”dysexekutivt syndrom”. En fixering till ”frontala symptom” och ”frontala test” i neuropsykologiska sammanhang kan vara direkt skadlig. Det gäller i ännu högre grad om man inte har gedigna kunskaper också om EM-syndromet och dess möjliga etiologier.

3.10 Något om demens och minne

Vad är demens?

Demenserna brukar behandlas utförligt i de flesta framställningar om minnets sjukdomar, och den här framställningen blir i motsvarande grad kortfattad. Med ”demens” brukar man mena en kronisk, organiskt betingad störning av kognitiva funktioner som är av sådan grad att den stör grundläggande sociala aktiviteter.[95] Inom detta begrepp ryms givetvis en hel mängd olika tillstånd, och någon symptomatologisk enhetlighet finns inte bland demenserna. Detta gäller också om man ser på de enskilda demenssjukdomarna, definierade genom sin specifika patologiska anatomi och fysiologi, exempelvis Alzheimers demens (termen här fattad i vid mening).[96] Sjukdomsbilderna och sjukdomsförloppen vid Alzheimer kan således vara mycket skiftande.

Demens är sammansatt av grundläggande syndrom

Vi har funnit det fruktbart att analysera de olika sjukdomsbilder som man kan se vid demenssjukdomarna som *sammansatta* av mer grundläggande syndrom, och då i första hand KA-, EM- och AE-syndromet. Alzheimers

[95] Fritt efter Lipowski (1980). Jfr. också Lindqvist & Malmgren (1990), ss. 158ff.
[96] Terminologin på demensområdet har skiftat över tiden. När författaren var kliniskt aktiv på 1980-talet var det viktigt att skilja mellan å ena sidan den ofta tidigt debuterande och ganska sällsynta *Alzheimers sjukdom*, å andra sidan den mycket vanligare, åldersrelaterade *(senila) demensen av Alzheimertyp.* Numer ser man oftast båda sammanfattade under begreppet *Alzheimers demens,* och vi följer den konventionen.

demens karakteriseras således vanligen av ett EM-syndrom, som i de senare stadierna blir mycket uttalat. I ett relativt tidigt skede kan en Alzheimer däremot ibland yttra sig som en nästan renodlad Korsakoff-amnesi. Vanligen ser man blandformer av KA- och EM-syndrom med påtagliga inslag av fokalneurologiska symptom (agnosier, afasi med mera). Som förstadium till Alzheimers demens, liksom till demens som betingas av förändringar i hjärnans blodkärl, ser man ofta smygande symptom som vid ett AE-syndrom. Detta har på senare år uppmärksammats mycket, om än vanligen i termer av ”mild cognitive impairment” (MCI).[97] Slutligen kan det nämnas att symptombilden vid sådan demens som orsakats av s.k. normaltryckshydrocephalus ofta beror på en kombination av AE-syndrom med vad vi kallar SSC-syndrom (somnolens-sopor-coma-syndrom, alternativ term ”patologisk vakenhetssänkning”).[98]

Destruktion av minnesspår vid demens

En viktig typ av förändring som inte fångas av vår hittillsvarande beskrivning är vissa störningar av långtidsminnet som uppträder vid många fall av avancerad demenssjukdom. Det är inte bara fråga om störd eller upphävd inlagrings- och återerinringsförmåga, utan det tycks röra sig om en irreversibel ”radering” av för länge sedan inlagrade minnen. Vi ser då inte en distinkt men ibland reversibel retrograd amnesi som vid ett rent KA-syndrom, utan istället ett till synes oåterkalleligt, mer fläckvis bortfall av minnen. Möjligen kan detta bortfall beskrivas som en svår form av AE-syndrom, men även om en sådan begreppsbildning kunde motiveras måste det framhållas att en förstöring av lagrade minnen är något helt annat än en av nedsatt koncentration störd lagring och/eller framhämtning av dem.[99]

Med dessa kommentarer har vi kommit till slutet av vår bakgrundsteckning av de klassiska forskningsfälten inom områdena minne och inlärning. Läsaren är nu förhoppningsvis mogen att bekanta sig med en ny, viktig familj av redskap för modellering av minne, inlärning och minnesstörningar: de artificiella neurala nätverken.

[97] Bischkopf et al. (2002).
[98] Se Lindqvist et al. (1993).
[99] Se vidare Lindqvist & Malmgren (1990), s. 148.

4. Artificiella neurala nätverk: grunderna

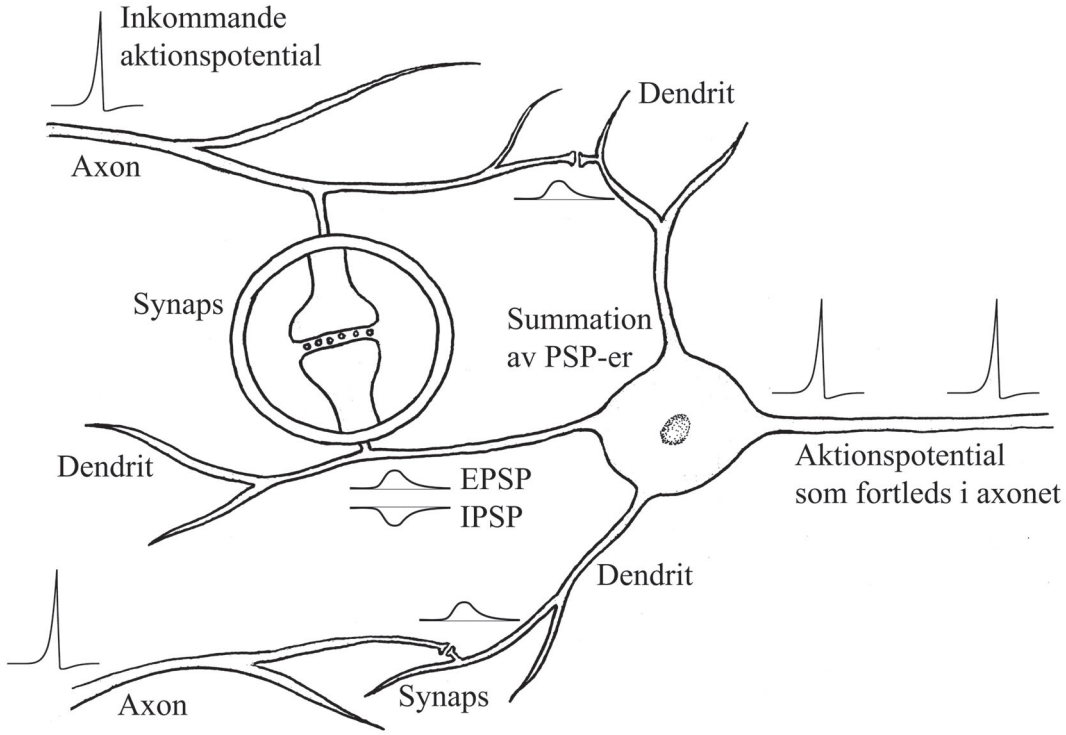
4.1 Val av beskrivningsnivå

Eftersom artificiella neurala nätverk är ett slags matematiska modeller, kan man i princip beskriva deras funktion uttömmande genom att ange de matematiska formlerna för den signalbearbetning som de utför och hur denna modifieras över tiden (när de lär sig genom exempel). En sådan abstrakt och kompakt beskrivning betonar släktskapet med andra matematiska, särskilt statistiska, modeller för databehandling. Om man, som vi, också vill betona släktskapen med biologiska neurala nätverk kan det vara lämpligare att använda en något mer konkret beskrivningsform och ett språk som lånar termer från biologin. En konkret beskrivning är också lättare att förstå för alla utom matematiker... så vi väljer att vara lite mer åskådliga än vad många matematiker skulle föredra. Och varför inte börja med den biologiska bakgrunden.

4.2 Något om verkliga nervceller

Den mänskliga hjärnan innehåller mer än 10^{10} (kanske 10^{12}) nervceller, som var och en är sammankopplade med ett antal (genomsnittligt troligen minst 1000) andra nervceller. De kommer i olika utföranden, men man kan beskriva de flesta av dem i termer av *cellkropp, dendriter, axon och synapser.*[100] Se figur 14!

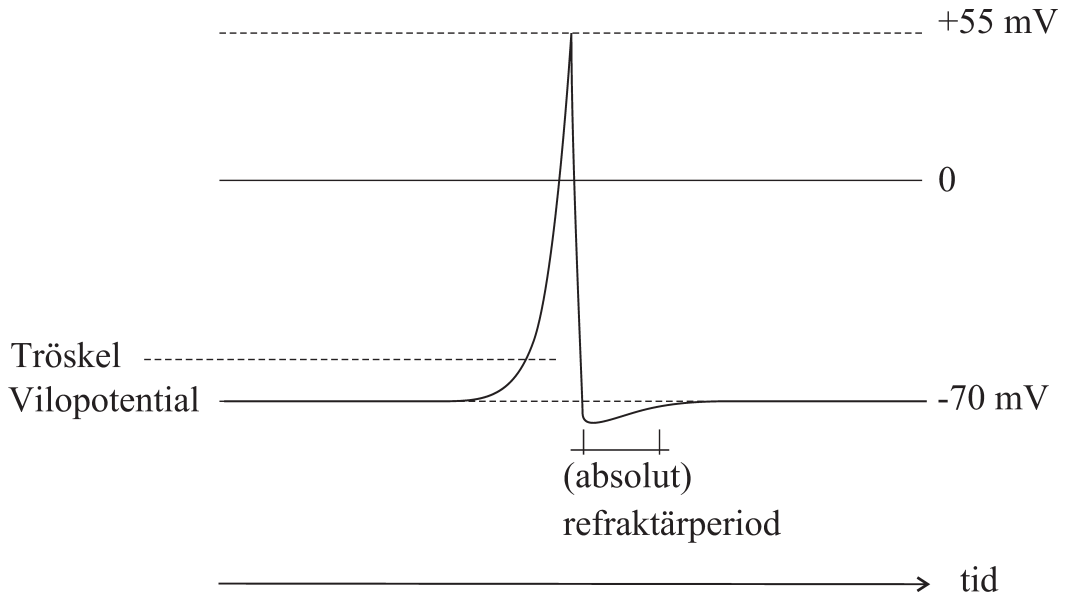
[100] Den här boken har inte plats för någon översikt av hjärnans funktioner, utom vad gäller grundläggande fakta om signalbearbetningen i nervceller och synapser. Den läsare som känner sig alldeles obekant med sin hjärna rekommenderas att läsa t.ex. Larsson (2000) eller O'Shea (2005).



Figur 14. En schematiserad, typisk nervcell och dess synapser. Förklaring: se text.

Signalbehandling i nervceller

Vi ska nu ge en kraftigt förenklad beskrivning av några grunddrag i typiska nervcellers funktionssätt. Låt oss börja långt upp till vänster i figur 14, vid *synapsen,* dvs. omkopplingsstället för signaler från en nervcell till en annan. Den signal som kommer in till synapsen, via det utskott från den sändande cellen som kallas *axonet,* är en elektrokemisk (*inte* rent elektrisk) impuls av allt-eller-intet-typ, kallad *aktionspotential.* Vi ska förklara mekanismerna bakom aktionspotentialen nedan, men redan nu behöver vi en kort beskrivning av den. Nervcellen har, liksom andra celler, överallt ett *cellmembran* som skiljer dess insida från omgivningen. I vila föreligger en elektrisk spänning ("vilopotentialen") mellan cellens insida och utsida om cirka –70 mV (millivolt; insidan negativ). Aktionspotentialen består i en snabb, temporär förändring av spänningen över en viss punkt på cellmembranet, från vilopotentialen till cirka +55 mV och sedan tillbaka igen. Se figur 15! Aktionspotentialen beror på jonflöden över membranen, är självpropagerande (fortplantar sig längs axonet), och har nu alltså anlänt till synapsen.



Figur 15. Aktionspotentialen mätt över en punkt på axonets membran.

Signalöverföringen i synapsen är kemisk. I synapsen frigörs nämligen, då aktionspotentialen anländer, kemiska substanser *(transmittorer, signalsubstanser),* som i sin tur påverkar den elektriska spänningen över cellmembranet hos den mottagande cellen – närmare bestämt påverkas först ändarna av dess inåtledande utskott, *dendriterna*. Dessa elektriska spänningsförändringar kallas *postsynaptiska potentialer* (PSP) och kan vara olika stora; de är alltså *inte* "allt-eller-intet". Det finns två slag: en PSP är excitatorisk (en EPSP) om den gör spänningen över membranet mindre negativ ("depolariserar") och därmed tenderar att framkalla aktionspotentialer i den mottagande cellen (se nedan). Den är inhibitorisk (en IPSP) om den har motsatt verkan, dvs. gör membranpotentialen mer negativ ("hyperpolariserar") och därmed minskar sannolikheten för att en aktionspotential skall uppstå. Om en PSP är excitatorisk eller inhibitorisk beror på vilken transmittorsubstans synapsen använder sig av.

Det sagda antyder att de postsynaptiska spänningsförändringarna kan utlösa aktionspotentialer om de har rätt tecken och är tillräckligt stora. PSP-erna sprider sig nämligen, på rent passiv (elektrotonisk) väg, ner till ett känsligt område i närheten av axonets början i cellkroppen. Cellmembranet i denna axonnära region har den märkliga och viktiga egenskapen, att *en aktionspotential startar där om depolariseringen blir tillräckligt stor*. Flera postsynaptiska potentialer kan anlända samtidigt eller nästan samtidigt dit, och då kombineras deras verkan. Man talar om *spatial summation* när samtidiga PSP-ar från olika synapser kombineras, och

om *temporal summation* när PSP-ar från samma eller olika synapser kommer så tätt på varann i tiden att de interagerar med varann. Den nivå för membranpotentialen där en aktionspotential utlöses kallas i figur 15 för ”tröskel”.

Aktionspotentialen är, som vi redan nämnt, en dramatisk men kortvarig depolarisering av membranet. Den är elektrokemisk till sin natur och betingas närmast av snabba flöden av joner över membranet, flöden som i sin tur är spänningsberoende och utlöses när membranet depolariserats tillräckligt mycket av EPSP-erna eller av andra faktorer. I vila är koncentrationen av kaliumjoner (K^+) hög inuti cellen medan det utanför cellen finns ett överskott av natriumjoner (Na^+). Dessa koncentrationer upprätthålles bl.a. genom att det finns aktiva mekanismer (”jonpumpar”) som hela tiden transporterar joner över membranet, samtidigt som jonerna endast diffunderar långsamt tillbaka. Om potentialskillnaden över membranet når tröskelnivån öppnas kanaler i membranet som tillfälligt släpper in stora mängder av natriumjoner. Detta innebär en ytterligare depolarisering som ger en lavineffekt, men som också åstadkommer att andra kanaler öppnas och släpper ut kaliumjoner ur cellen. Därmed återgår membranpotentialen snabbt mot viloläget, och det sker till och med en överkompensation i slutet av cykeln så att membranet för en tid blir “hyperpolariserat”. Under större delen av tiden för hyperpolariseringen kan ingen ny aktionspotential utlösas. Man talar om en “absolut refraktärperiod”. Så småningom har jonpunparna återställt den normala balansen över cellmembranet.

Aktionspotentialen är *självpropagerande* och *självbegränsande*. Detta betyder att den automatiskt sprider sig till angränsande icke-refraktära områden – dvs. längre ut i axonet – samtidigt som den snabbt upphör där den först startade. Det uppstår på så vis en signal, som ganska snabbt (dock inte alls så fort som en rent elektrisk impuls) fortplantar sig i axonet. Och så anländer signalen till nästa synaps – vi är tillbaka där vi började.

En nervcell kan skicka aktionspotentialer, eller som man säger, “fyra”, med en frekvens av upp till några hundra impulser per sekund. Det är inte bara mängden av inkommande aktionspotentialer som bestämmer om och hur fort ett neuron ska “fyra”; egenskaper hos cellmembranet och hos synapserna spelar en minst lika stor roll. För det första är tröskeln för aktionspotentialens utlösande inte densamma i alla neuron; vissa neuron fyrar till och med spontant i avsaknad av inhibitorisk input. För det andra är

synapser olika effektiva och enskilda synapsers effektivitet kan variera, både på kort och på lång sikt. Efter en kort period av intensiv aktivitet hos en nervcell ser man t.ex. ibland en kortvarig "post-tetanisk potentiering" då de aktiverade synapserna är mer effektiva än förut, kanske på grund av att mer signalsubstans tillfälligt har syntetiserats.

LTP, LTD och STDP

Ännu intressantare är att samtidig hög aktivitet hos två nervceller som har synaptisk kontakt ibland leder till att effektiviteten hos den synaptiska förbindelsen mellan dem specifikt ökar för en avsevärd tid (upp till veckor). Detta fenomen, *långtidspotentiering* (LTP), har observerats i bl.a. hippocampus och storhjärnbarken, och anses av många erbjuda ett plausibelt fysiologiskt underlag för associativ inlärning. (Jämför not 11–12, s. 14). Ett flertal andra typer för synaptisk modifiering är också kända. Således ser man vid vissa typer av stimulering istället en *långtidsdepression* (LTD) av synapser. LTD har iakttagits i hippocampus och lillhjärnan. Lågfrekvent stimulering har observerats oftare ge LTD än högfrekvent sådan. Under senare år har flera forskargrupper fått evidens för att den tidsliga relationen mellan enstaka pre- och postsynaptiska aktiveringar är väsentlig för om det skall bli en potentiering eller depression. Detta fenomen kallas STDP (*spike-timing-dependent plasticity*). Närmare bestämt tycks LTP i försök med enstaka stimuli kräva att den presynaptiska aktiveringen kommer före eller samtidigt med den postsynaptiska.[101]

Både LTP och LTD involverar sannolikt en speciell form av receptormolekyl för signalsubstanserna, den s.k. NMDA-receptorn, och calciumjonen (Ca^{2+}) tycks spela en nyckelroll. En vanlig hypotes i dag är att NMDA-receptorn kräver postsynaptisk elektrisk aktivitet för att reagera på signalsubstansen, och att kortvarig aktivering av NMDA-receptorerna leder till en relativt länge bestående ökning av den tillgängliga mängden av "vanliga" receptorer, s.k. AMPA-receptorer.

Informationsbehandlingen i nervsystemet sker alltså på grundval av flera olika kemiska, elektriska och elektrokemiska processer. Aktionspotentialen är bara *en* aspekt. Vidare bör man notera att fastän aktionspotentialerna är av allt-eller-intet-karaktär, så är den kontinuerligt varierbara *frekvensen* av dem sannolikt ofta avgörande för vilket budskap som förmedlas. Man kan med andra ord säga, att nervsystemet ofta använder sig av

[101] Vilket ju är i överensstämmelse med Hebbs princip. Se vidare Gerstner & Kistler (2002).

frekvenskodning. Med stor sannolikhet är det också ofta – jämför STDP – av betydelse *när,* i förhållande till andra relevanta händelser, som de *enskilda* aktionspotentialerna utlöses. Man talar i detta fall om *pulskodning*.

När man i de enklaste ANN-modellerna analyserar ett nervnäts funktion i termer av en stereotyp effekt av den enskilda aktionspotentialen, så förenklar man alltså bilden avsevärt. Andra artificiella neurala nätverk modellerar istället aktionspotentialernas frekvens, och under senare år har man formulerat ANN-teorier för pulskodade neurala nätverk.[102] De sistnämnda kommer inte att beröras närmare i denna bok. Däremot kommer vi att fortsätta att uppmärksamma andra möjliga mekanismer för inlärning än sådana som bygger på synaptisk plasticitet. Se särskilt avsnitt 7.3.

4.3 ANN-element och deras signalbearbetning

Noder och aktiviteter

Ett artificiellt neuralt nätverk består av ett antal *element* eller *noder*, X_i, som var och en tar emot signaler från andra noder (eller från yttervärlden), behandlar dem, och skickar resultatet vidare till andra noder (eller ut i världen). Varje nod X_i har i varje ögonblick en *aktivitet*, vars nivå vi betecknar med motsvarande lilla bokstav x_i.[103]

Aktivitetsnivån är i en del modeller diskret (t.ex. antingen 1 eller 0) men i andra kontinuerligt graderbar. En del ANN-enheter (de "binära") har alltså bara 0 och 1 som möjliga aktiviteter, de "bipolära" använder sig av –1, 0 och +1, medan slutligen andra ger en signal som kan variera kontinuerligt mellan bestämda gränser, exempelvis mellan 0 och 1, eller mellan –1 och +1. De olika modellerna har sina fördelar och nackdelar i olika matematiska sammanhang. För klassifikationssyften använder man ofta en binär aktivitet som representation av två alternativa klasser, men bipolära enheter är bra när man också vill representera en kategori "osäkra fall", och graderad aktivitet mellan 0 och 1 kan representera *sannolikheten* för tillhörighet till en klass (jämför avsnitt 9.3 nedan). Graderade signaler behövs också när nätverket används för regressionsändamål (se avsnitt 4.5), och är nödvändiga för att vissa viktiga inlärningsregler för

[102] Se t.ex. Maass & Bishop (red.) (1999).
[103] Observera att terminologin skiljer sig mycket mellan olika framställningar av teorin för ANN. Vi ansluter oss här till formalismen i Fausett (1994).

neurala nätverk skall vara tillämpbara (se vidare de följande avsnitten, särskilt 4.6).

Ur en mer biologisk synvinkel är det uppenbart att binära ANN kan modellera *den enskilda aktionspotentialens allt-eller-intet-karaktär*. Men även t.ex. en bipolär enhet kan vara en modell av allt-eller-intet-potentialer. Hur då, då? Det kan väl inte finnas negativ aktivitet? Nej, men i vissa sammanhang kan det vara matematiskt fördelaktigt att representera ”ingen aktivitet” med –1 istället för med 0, och det gäller inte bara i rent tekniska tillämpningar utan också när man modellerar biologiska förlopp. En kontinuerligt varierbar aktivitet som är begränsad både uppåt och neråt kan tjäna som modell av *aktionspotentialernas frekvens* i ett biologiskt neuron. Valet av 1 som övre gräns är en matematisk konvention, och att man ibland väljer –1 som nedre gräns är också en matematisk bekvämlighetsfråga.

Aktiviteten hos en enhet kallas inte sällan dess “output”. Den senare beteckningen kan vara vilseledande, eftersom det då är lätt att förväxla en sådan “lokal” output hos en enhet med den outputsignal som ett ANN i dess helhet avger (se nedan). Vi föredrar därför termen “aktivitet”.

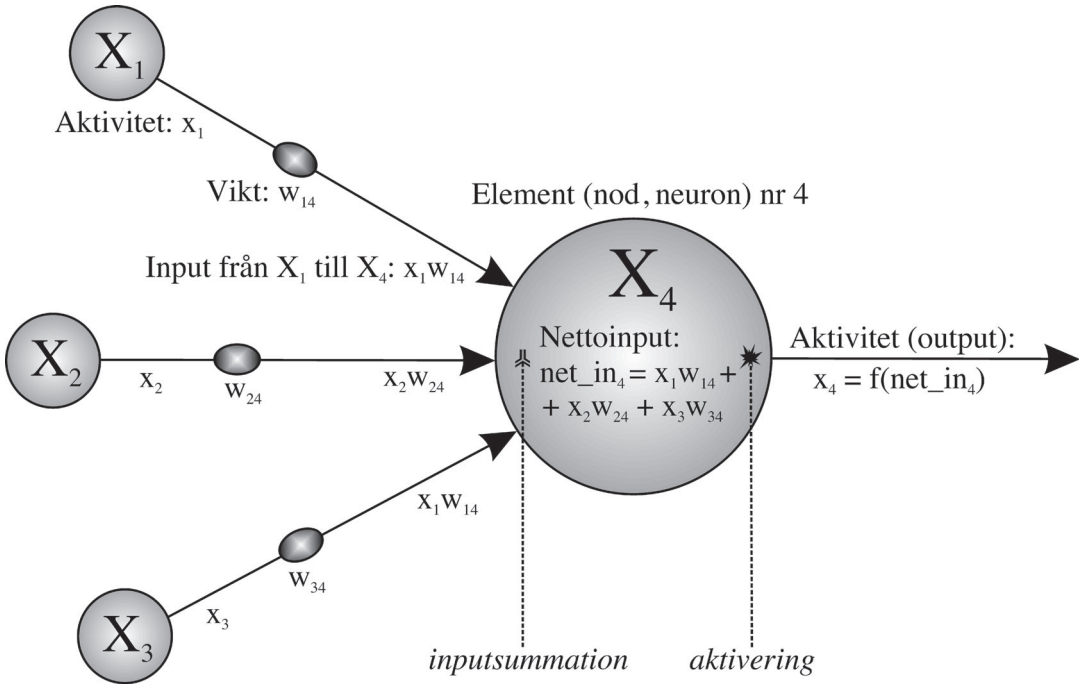
Vikter

Noderna är sammankopplade till nätverk av större eller mindre komplexitet genom riktade förbindelser, som var och en har en viss *vikt*, w. Styrkan på förbindelsen från den i:te till den j:te noden betecknas här med w_{ij}. Tänk på w_{ij} som den sammanlagda effektiviteten hos synapserna mellan två neuron! (Jämför not 104 nedan.) Förbindelsen åt andra hållet (från nod X_j till nod X_i), om det finns någon, kan ha en helt annan vikt. Frånvaro av förbindelse från en nod till en annan konceptualiseras ofta som en förbindelse med vikten 0.

Signalbehandling i ANN-noden

Signalbehandlingen i artificiella neurala nätverk kan som tidigare nämnts modelleras som en process i antingen kontinuerlig eller diskret tid; vi ska tills vidare bara behandla det senare, enklare fallet. Vi tänker oss att varje tidsrymd kan delas upp i ett ändligt antal små steg, och att varje sådant tidssteg innebär att en eller flera enheter i nätverket genomgår en viss processcykel. Låt oss ta en närmare titt på denna processcykel.

Betrakta figur 16, som föreställer en nod någonstans i ett artificiellt neuralt nätverk. Bilden skall givetvis jämföras med figur 14 ovan.[104] Vi antar att noden ifråga, som vi kallar X_4, mottar förbindelser från tre andra element (X_1–X_3).



Figur 16. Signalbearbetning i en typisk ANN-nod. Förklaring: se text.

Det som händer kan formaliseras i följande steg, som motsvarar de steg som vi delade in skeendet i hos verkliga neuron:

1) *Omvandling av andra elements aktiviteter till insignaler* (via vikterna)

2) *Integration (summation)* av insignalerna

3) *Aktivering av elementet* (via aktiveringsfunktionen).

I en beskrivning av ett *helt nätverks* signaldynamik måste man också tala om hur nätverket som helhet får sin input och hur det avger sin output;

[104] En viktig skillnad är att medan figur 14 visar att ett neuron kan ha flera synapser till ett givet annat neuron, så tillåter ANN-representationen bara en förbindelse (i en given riktning). Det är därför som man ska tänka på vikten som den *sammanlagda* effektiviteten hos ett antal synapser. Detta antal kan givetvis vara = 1.

med andra ord, hur signalen först kommer till nätverket överhuvudtaget, och på vad sätt det levererar ett slutresultat till omvärlden. (“Omvärlden” kan här vara andra neurala nätverk.) Dessutom återstår den centrala uppgiften att beskriva hur ett nätverk kan förändra sitt arbetssätt över tiden – dvs. hur det *lär* sig. Mer om allt detta strax – men låt oss först betrakta vad det enskilda elementet i figur 16 just nu har för sig. Här följer således några kommentarer till punkterna 1–3 ovan.

Ad 1: När signaler sänds från en uppsättning element till en annan, antar man att de förstärks i proportion till vikterna hos förbindelserna. Man modellerar oftast detta så, att den effektiva input till en enhet från en annan beräknas som *produkten* av *aktiviteten* i den sändande enheten och *vikten* hos förbindelsen mellan denna enhet och den mottagande. Förbindelsen från enhet X_i till enhet X_j har vikten w_{ij}; genom denna förbindelse kommer således bidraget $x_i w_{ij}$ till X_j:s totala input. Detta kan ses som en (primitiv) modell av inflytandet från synapsernas effektivitet på informationsflödet över synapserna.

Ad 2: I verkliga nervceller kan flera PSP-er samverka till uppkomsten av en aktionspotential. Motsvarigheten i ANN-noder är i standardmodellerna att de viktade insignalerna (inputs) till en given nod *summeras*. Resultatet blir vad man kallar *nettoinput* till enheten. (Förväxla inte insignaler och nettoinput till en nod med input från omvärlden till ett nätverk i dess helhet!)

Den totala formeln för hur nettoinput x_in_j till en nod X_j uppstår är, om x_i som vanligt betecknar aktiviteten i neuron x_i, således

$$x\_in_j = \sum_{i=1}^{n} x_i w_{ij} \qquad (4.3.1)$$

där summeringen sker över alla enheter i nätverket. (Om summaformler, se nedan!) Observera att eftersom en nod kan ha en förbindelse till sig själv, så ingår termen $x_j w_{jj}$ alltid i summan (men w_{jj} kan förstås vara 0).

Ad 3: Om en verklig nervcell aktiveras, och hur mycket (med vilken frekvens), bestäms som nämnts ovan inte bara av inflödets storlek och synapsernas effektivitet utan också av cellens övriga egenskaper. Detta modelleras i ANN-teorin genom att man antar en för enheten specifik *aktiveringsfunktion,* som tar nettoinput som sitt argument. Det värde som

kommer ut ur denna funktion är alltså enhetens resulterande aktivitet (eller output, om man föredrar den termen). Om aktiveringsfunktionen betecknas med f kan vi med andra ord skriva aktiviteten som

(4.3.2) $x_j = f(x\_in_j)$

vilket i ljuset av (4.3.1) är detsamma som

(4.3.3) $x_j = f\left(\sum_{i=1}^{n} x_i w_{ij}\right)$

Det är implicit i formel 4.3.3 att aktiviteten x_j (output) uppstår i *tidssteget efter* de aktiviteter som nämns i högerledet (inputs). Med andra ord, en explicit formel ser ut så här:

(4.3.4) $x_j(t+1) = f\left(\sum_{i=1}^{n} (x_i(t) w_{ij}(t))\right)$

Vi skriver dock fortsättningsvis inte ut tiden explicit.

Något om index och summaformler

För den läsare som känner ett främlingskap inför formlerna i detta avsnitt kommer här en kort sammanfattning av hur man använder index och summaformler.

Man numrerar ofta företeelser eller storheter genom att sätta ett index på deras namn; ett exempel var när vi kallade nod nr 4 för X_4 och dess aktivitet för x_4. En godtycklig nod kan betecknas med t.ex. X_i. Ibland kan det vara fördelaktigt att använda så kallade *dubbelindex*, som när vi betecknar en vikt med w_{14} eller väljer w_{ij} som beteckning på en godtycklig vikt. Observera att beteckningen w_{ij} inte utesluter att i = j; w_{ii} kan däremot bara stå för en vikt med samma första- som andraindex.

För att beteckna en summa av n stycken numrerade element x_1, x_2, ...x_n använder man sig med fördel av summationssymbolen:

$$\sum_{i=1}^{n} x_i$$

I den här framställningen kommer vi ibland att använda det förenklade skrivsättet Σx_i och låta det framgå av texten vilka värden i kan anta.

I många formler, t.ex formeln för summation av inputs

(4.3.1) $$x\_in_j = \sum_{i=1}^{n} x_i w_{ij}$$

står ett index (här: j) för ett utvalt element, medan ett annat (här: i) är ett variabelt namn för de företeelser som man summerar över. Formel 4.3.1 säger alltså, att man vid beräkningen av nettoinput till en viss nod nr j ska ta med alla vikter som har j som andra index.

I andra sammanhang kan man behöva summera över alla dubbelindexerade storheter i en viss uppsättning, och måste då markera på något sätt (i formeln eller i texten) att summan ska löpa över alla i och j. Inte sällan uppstår komplikationen att man beräknar summor av uttryck som själva innehåller summor. Vi kan inte här gå in i detalj på reglerna för hur man får manipulera sådana dubbelsummaformler, men reglerna är inte särskilt krångliga att förstå.

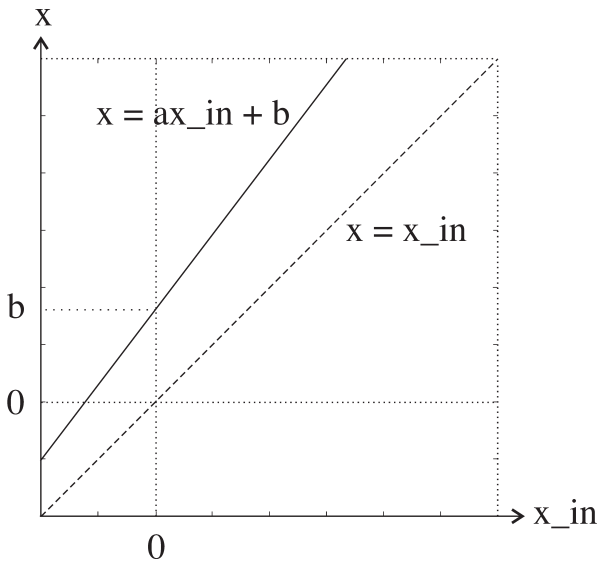
Olika aktiveringsfunktioner i ANN

Aktiveringsfunktionen bidrar i högsta grad till att bestämma det sätt på vilket ANN-noden, och därmed nätverket som helhet, behandlar signaler. I standardteorin för artificiella neurala nätverk laborerar man med några grundläggande typer av aktiveringsfunktioner:

• *Linjära funktioner,* dvs. funktioner av typen

(4.3.5) $$x = a \cdot x\_in + b$$

där a och b är konstanter. Se figur 17!



Figur 17. Två linjära aktiveringsfunktioner. x_in är nettoinput; x är den resulterande aktiviteten.

I det enklaste fallet väljer man a = 1 och b = 0, dvs. identitetsfunktionen:

(4.3.6) $x = x\_in$

Helt linjära nätverk är lätta att analysera och bra att ha, inte minst för att lära sig att förstå hur artificiella neurala nätverk fungerar. De har dock begränsat intresse både som biologiska modeller (eftersom biologiska neuron inte är linjära) och som matematiska analysinstrument (eftersom de inte är särskilt kraftfulla som sådana, och dessutom redan har exakta motsvarigheter i traditionell statistisk teori).

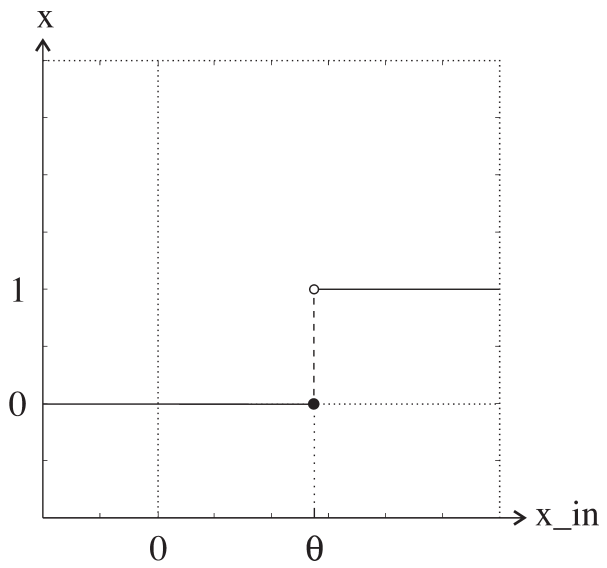
Däremot är det vanligt att man använder linjära enheter för att ge output (till omvärlden) från nätverk som i övrigt innehåller andra typer av element. Sådana linjära outputenheter är ett rationellt val då nätverken används för regression (om detta begrepp, se vidare avsnitt 4.5, 5.1, 6.2 och 9.1).

Det ska poängteras redan här att termen ”linjär” ofta används om vissa nätverk som också innehåller neuron med icke-linjära aktiveringsfunktioner. Detta har en rationell motivering, som vi ska gå igenom i avsnitt 6.3. Neurala nätverk med endast linjära element kallas i denna bok oftast för ”helt linjära” för att skilja ut dem från linjära ANN i vidare mening. Är outputenheternas aktiveringsfunktion dessutom identitetsfunktionen, kallar vi dem ”maximalt linjära”.

• *Stegfunktioner.* Stegfunktioner kallas också "tröskelfunktioner". De innebär i sin vanligaste version (den binära) att nettoinput ska nå över en viss *tröskel,* ofta betecknad med den grekiska bokstaven θ (theta), för att aktivitet skall uppstå. Formellt,

(4.3.7)
$$\begin{aligned} x\_in > \theta &\rightarrow x = 1 \\ x\_in \leq \theta &\rightarrow x = 0 \end{aligned}$$

Alternativt kan funktionen specificera att x är 1 även på själva tröskeln.



Figur 18. En binär stegfunktion. Förklaring: se formel 4.3.7.

Tröskelfunktioner av denna basala typ kan ses som modeller av det *enskilda* allt-eller-intet-svaret i ett biologiskt neuron. De ger genast intressantare egenskaper hos nätverket än vad de linjära funktionerna kan åstadkomma. Vi ska möta ett exempel redan i den första specifika ANN-modell vi betraktar nedan (den enkla perceptronen, avsnitt 6.1).

Bipolära varianter av stegfunktionen finns också, som under olika omständigheter ger enheten aktiviteten –1, 0 eller 1. Jämför Hopfieldnätet (avsnitt 7.1).

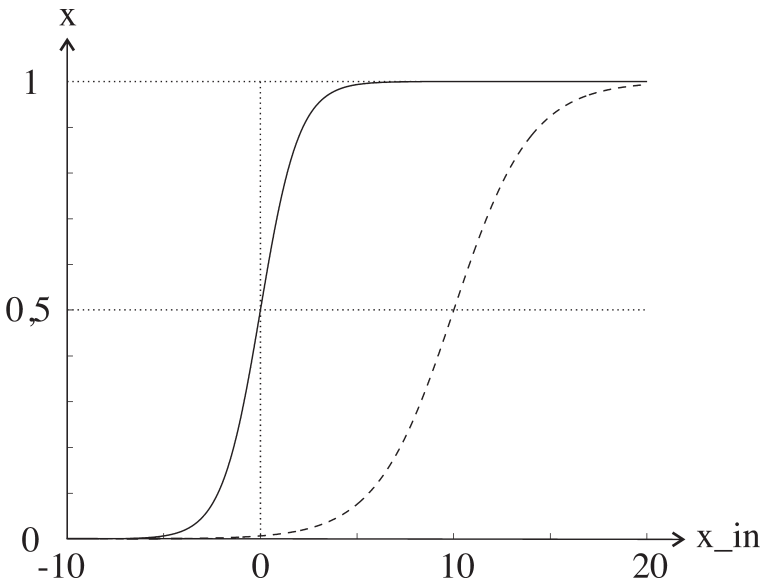
• *Tröskellinjära funktioner* är helt enkelt funktioner som är 0 under en viss tröskel men linjära över tröskeln.

• *Sigmoida (S-formade) funktioner*. I ANN-litteraturen används beteckningen “sigmoid” för det första om en speciell funktionsfamilj som också kallas de *logistiska* funktionerna, nämligen

$$y = \frac{1}{1 + e^{-kx}} \tag{4.3.8}$$

där specialfallet med k = 1 brukar kallas *den* logistiska funktionen. Men termen används ofta om en större klass av funktioner med liknande allmänna egenskaper. Vi ansluter oss här till den andra användningen av termen ”sigmoid”. En sigmoid funktion i denna allmänna mening är en funktion som är: (a) asymptotiskt begränsad både neråt och uppåt, exempelvis av 0 respektive 1 (ett annat vanligt val av gränser är –1 och +1); (b) monotont växande, med först växande och sedan avtagande derivata.

Dessa villkor medför att funktionens graf får ett S-format utseende. Någonstans på detta S (mitt på om funktionen dessutom är spegelsymmetrisk runt ett x-värde) finns en punkt där ökning av nettoinput får maximal effekt på aktiviteten:



Figur 19. Två logistiska funktioner. Heldragen kurva: $x = \frac{1}{1 + e^{-x\_in}}$. *Streckad kurva:* $x = \frac{1}{1 + e^{(10 - x\_in)/2}}$.

Väljer man en sigmoid funktion (i allmän mening) som aktiveringsfunktion i ett ANN får nätverket nya och ibland mycket kraftfulla signalbehandlande egenskaper. I så kallade flerlagrade perceptroner används därför oftast sådana aktiveringsfunktioner för de viktiga ”dolda” enheterna. Logistiska outputneuron ger även i enkla nätverk för klassifikation möjlighet till en sannolikhetstolkning av output (se särskilt avsnitt 4.5, 5.1, 6.2 och 9.3) och är därför ett naturligt val när det handlar om klassifikationsuppgifter där man inte bara vill ha svaren tillhör/tillhör inte klassen.

Sigmoida funktioner – som det finns ett obegränsat antal av – kan också användas som modeller av *signalfrekvensens* olinjära inputberoende i verkliga neuron. Ett biologiskt neuron har ju också ett golv och ett tak för sin aktivitet, och känsligheten för input är störst någonstans däremellan. I den sigmoida modellen för ett biologiskt neuron kan man också lägga in en tröskel där neuronet ifråga överhuvudtaget börjar aktiveras.

• *Aktivering genom konkurrens.* Detta sätt att aktivera enheter kan delvis reduceras till de redan nämnda, men måste delvis betraktas som en självständig typ av signalbehandling.

Många ANN-modeller har ett “kompetitivt” skikt av enheter (eller flera sådana skikt). Samma signal skickas från inputskiktet till alla enheterna i det kompetitiva skiktet. Nettoinput till en given enhet i det senare kommer att bestämmas av vilka vikterna är på förbindelserna mellan inputskiktet och denna enhet. Ett av neuronen utses, på ett sätt som varierar från modell till modell, till “vinnare” och får aktiviteten 1, medan övriga får aktiviteten 0. (Ett och noll är inte väsentligt; andra val av värden är givetvis möjliga.) Ett vanligt sätt att utse vinnare är att välja den enhet i det kompetitiva skiktet som har *störst nettoinput*.

Tänker man i biologiska termer kan det kanske vara svårt att förstå hur ett enskilt neuron ska kunna “veta” att det har störst nettoinput av alla i det skikt som det tillhör. Men man kan modellera kompetitiv aktivering genom att förse enheterna i ett ANN-skikt med tröskelsigmoida funktioner (jämför ovan) och koppla ihop dem med lämpliga *självexciterande* och *ömsesidigt inhiberande* förbindelser. Sannolikt finns en sådan organisation på flera håll i det mänskliga nervsystemet; jfr. avsnittet om Kohonens ”självorganiserande karta” (SOM, avsnitt 8.1).

För vissa andra varianter av kompetitiv aktivering som förekommer i ANN-teorin är en sådan reduktion till enheter med “vanliga” aktiverings-

funktioner inte teoretiskt möjlig. Vid andra tillämpningar av kompetitiva ANN än modellering av nervsystemet behöver man förstås inte bekymra sig närmare om detta. Se vidare om Kohonen-regler för inlärning i avsnitt 4.6, avsnittet om vektorer och matriser (5.2), samt om SOM och Learning Vector Quantization (LVQ, avsnitt 9.3)!

Vi har nu kastat en blick på det enskilda elementets sätt att behandla de signaler det får, och ska övergå till att studera hur flera element kan sättas samman till ett fungerande neuralt nätverk.

4.4 Nätverksarkitekturer

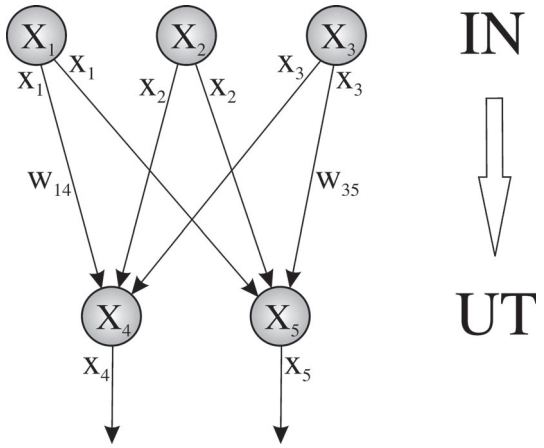
Enlagrade feedforward-nätverk

De enskilda elementen i ett ANN är anordnade i en struktur som kan ha olika uppbyggnad eller "arkitektur". Den kan för det första karakteriseras som *lagrad* eller *icke lagrad*. I lagrade nätverk kan man urskilja flera skikt av neuron i den specifika meningen att vissa enheter (som bildar skikt 1, eller inputskiktet) får signaler direkt från omgivningen, andra enheter (som bildar skikt 2) får signaler endast från inputskiktet, ytterligare andra (om det finns fler skikt än två) endast från neuron i skikt 2, etc. Lagrade ANN är enligt denna definition alltid av *feedforward-typ,* dvs. signalen genom nätverket är i viss mening enkelriktad. De står i detta avseende i motsats till *feedback-nät* (också kallat *återkopplade* eller *rekurrenta* nätverk), där det finns återkopplingar mellan elementen och där man alltså inte kan tala om "skikt" i den funktionella mening i vilken ordet används här.

Skiktningen som vi talar om definieras i termer av signalflödets riktning och inte har att göra med arrangemanget i rummet av noder som realiserar nätverket ifråga. Det är inte heller så, att ett anatomiskt skikt av neuron i hjärnan alltid skall representeras som ett funktionellt ANN-skikt. De anatomiska skikten kan ju mycket väl ha såväl dubbelriktade förbindelser med varann som inre förbindelser mellan sina neuron.

Figur 20 exemplifierar ett *enlagrat* feed-forwardnät. "Enlagrat" betyder att det finns *ett lager av (enkelriktade) förbindelser,* dvs. *två skikt av enheter*.[105]

[105] För lättare komma ihåg detta kan man välja att använda beteckningen "skikt" när



Figur 20. Enlagrat feedforward-nätverk. Förklaring: se text.

Signalflödet tänks här gå nedåt i systemet. I figur 20 förutsätts det därför att alla förbindelser mellan noder är riktade "nedåt", eller med andra ord att alla "uppåtriktade" vikter är = 0. Vi har bara noterat två av de sex "nedåtriktade" vikterna.

I ett dylikt enlagrat feed-forwardnät kan man skilja mellan ett *input-* och ett *outputlager* av enheter. Den inkommande informationen (från yttervärlden eller från ett annat nätverk) representeras först som inputnodernas aktiviteter. Denna *inputsignal* skickas därefter vidare till outputnoderna, och omvandlas på vägen via vikterna. Outputnoderna aktiveras i sin tur och levererar hela nätets output till omvärlden, till den mänskliga användaren eller till nästa neurala nätverk.

Man måste lägga märke till att den här modellen av ett feed-forwardnätverk inte specificerar hur aktiviteten i inputnoderna uppstår. Med "inputsignalen till nätverket" menar vi nämligen *inte* en signal som fungerar som en input, enligt schemat i figur 16, till inputnoderna. Vi avser istället inputnodernas aktivitetsmönster, dvs den *output* som de skickar till nästa skikt av enheter!

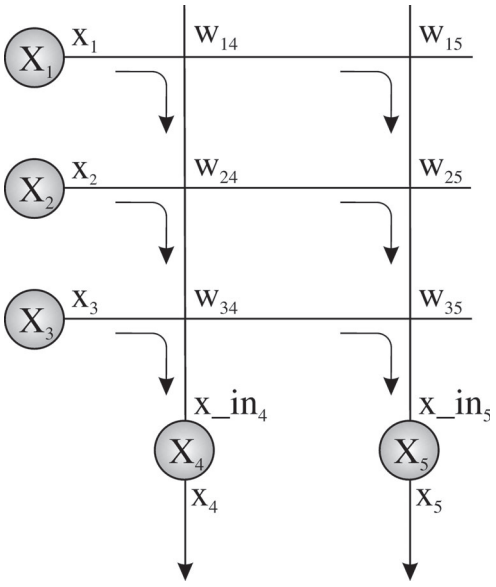
Ingenting hindrar förstås att man kompletterar modellen genom att också beskriva en mekanism för generering av inputnodernas aktivitet. Om modellen är avsedd att beskriva ett perifert sensoriskt nätverk och input-

man talar om olika lager av *enheter*. Vi har gjort så hittills, men kommer fortsättningsvis att tala både om "lager av förbindelser" och "lager av enheter" (t.ex. "inputlagret").

noderna motsvarar sinnesceller, blir det då fråga om att beskriva en viss transformation av fysiska stimuli till nervsignaler (*transduktion*). Om modellen istället handlar om ett nätverk som tar emot information från ett annat, så är det istället naturligt att använda sig av schemat i figur 16 för att också beskriva inputnodernas funktionssätt. Vid tillämpningar av feed-forward-nätverk som abstrakta modeller för databearbetning, slutligen, är nätverkets input helt enkelt de data som man som användare av modellen skickar till det andra skiktet av enheter, och inputnoderna är bara ”platshållare” för dessa data. Nog ordat om detta, men minns att ”input(signalen) till nätverket” i denna framställning alltid syftar på aktiviteten hos inputnoderna!

I matematiska beskrivningar av ANN tänker man sig inputsignalen till nätverket som en *vektor*, dvs. som en ordnad uppsättning tal. Vi talar alltså om en *inputvektor*; analogt om en *outputvektor.* (Mer om vektorbegreppet i nästa kapitel.) Arbetar man med feed-forwardnätverk för tekniska ändamål väljer man lämpligen ett ANN som har lika många inputnoder som antalet komponenter i den inputsignal man vill att nätverket skall bearbeta. Man kan också lägga till extra inputnoder för särskilda ändamål, t.ex. en så kallad *biasnod* (jämför diskussionen om linjär regression i avsnitt 6.2).

Ett annat framställningssätt för nätverket i figur 20 är dess så kallade *Hintondiagram.* Se figur 21.

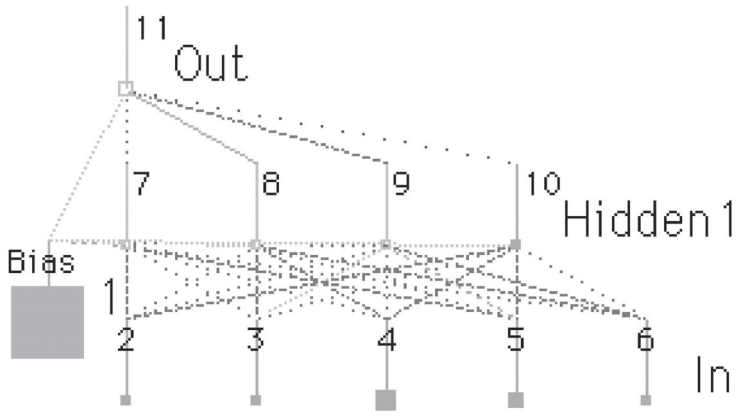


Figur 21. Hinton-diagram av feed-forward-nät. Förklaring: se text.

I detta diagram tänks aktiviteten först gå horisontellt från inputenheterna, ”fångas upp” vid korsningarna och sedan gå ner till outputenheterna. Tänk gärna på de horisontella ”utskotten” som axoner och de vertikala som dendriter! Observera att många framställningar ”vänder” på Hintondiagrammen; vårt sätt att rita är anpassat till vårt beteckningssätt för vikterna. En fördel med Hintondiagrammet är att nätverkets vikter är lätta att skriva ut utan att det blir “grötigt”, en annan att vikterna visualiseras i matrisform. Detta anknyter till de matematiska redskap man främst använder vid analysen av nätverkens funktion (vektor- och matrisalgebra). Se vidare avsnitt 5.2!

Flerlagrade feed-forward-nät

En naturlig och mycket viktig utvidgning av den enlagrade feed-forward-arkitekturen är de *flerlagrade* feed-forward-näten, där minst ett skikt av enheter interpolerats mellan input- och outputlagren. Dylika interpolerade lager kallas också “dolda” lager, eftersom deras aktiviteter inte är direkt tillgängliga utanför nätverket på det sätt som nätverkets input- och outputsignaler är det. Figur 22 visar hur datorprogrammet NeuralWorks framställer ett tvålagrat (två lager av förbindelser!) ANN med sex inputnoder (varav en biasnod), fyra dolda noder och en outputnod. Observera att informationsflödet här går uppåt i bilden. De varierande storlekarna på noderna representerar i NeuralWorks olika aktivitetsnivåer.



Figur 22. Ett tvålagrat feed-forwardnätverk. Förklaring: se text.

De dolda noderna ger flerlagrade ANN möjligheter till många former av informationsbearbetning som inte kan realiseras i enlagrade nätverk.

Redan ett tvålagrat nätverk kan, om det har sigmoida aktiveringsfunktioner i de dolda enheterna, i en viss mening utföra “alla” beräkningsuppgifter. Se kapitel 9!

De kraftfulla flerlagrade nätverk som just omtalats har *adaptiva* (modifierbara) vikter både till de dolda noderna och till outputnoderna. Om man till ett enlagrat nätverk lägger ett skikt av enheter utan att det tillkommer några adaptiva vikter, får man inte samma slags ökade beräkningskraft. Inte sällan klassificeras därför även sådana nätverk som ”enlagrade”; man syftar då på *antalet lager med adaptiva vikter.* Se vidare avsnitt 6.3.

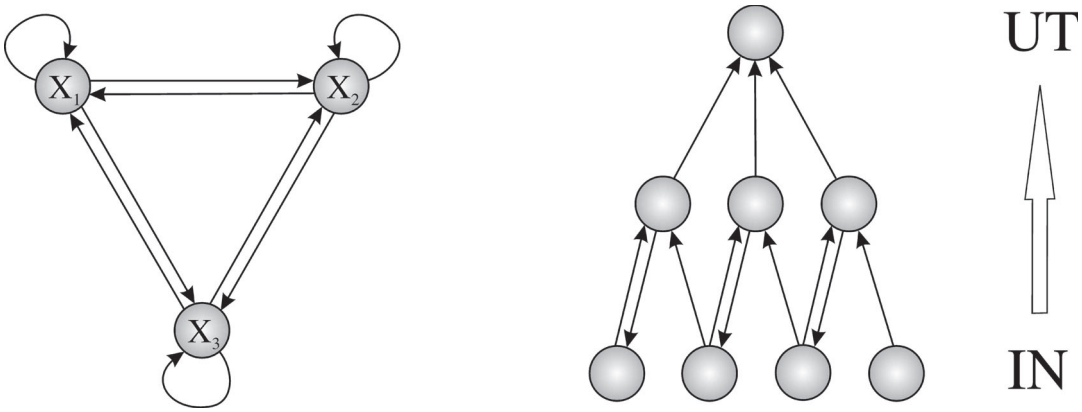
Feedback-nät (återkopplade eller rekurrenta ANN)

I återkopplade ANN är, som nämnts, signalströmmen inte enkelriktad utan går fram och tillbaka mellan enheterna i enlighet med mer eller mindre komplicerade mönster av förbindelser. Enskilda neuron kan därför vara aktiva flera olika tidssteg efter det att nätverket fått en enda inputsignal från omvärlden, och i vissa fall godtyckligt länge efteråt (åtminstone i teorin). I återkopplade nät kan man inte tala om ”skikt” av enheter i funktionell mening. En helt annan sak är att man kan symbolisera också rekurrenta nätverk med figurer där enheterna geometriskt framställs som liggande i olika lager. Och, som redan påpekats, funktionella skikt är något annat än anatomiska skikt.

Återkopplingarna i ett rekurrent ANN kan vara mer eller mindre *direkta.* I vissa nätverk skickar enheter signaler direkt till sig själva, i andra går signalen först från en enhet till en annan och sedan tillbaka till den första, medan ytterligare andra ANN använder sig av ännu längre feedback-kretsar. *Hur mycket* feedback det finns i ett ANN kan också variera, från det maximala fall då alla enheter får signaler direkt från alla (inklusive från sig själv), till de minimala fall då enstaka rekurrenta förbindelser har adderats till ett feed-forward-nät. I den sistnämnda typen av fall brukar man inte sällan, mer eller mindre oegentligt, fortsätta att tala om nätverket som “lagrat”, och tänker då på den ursprungliga feed-forward-strukturen. Två mycket studerade typer av återkopplade ANN, Elman- och Jordannäten, kan i denna terminologi karakteriseras som “tvålagrade perceptroner med feedback”. ART (Adaptive Resonance Theory) är en annan flerlagrad nätverksmodell med feedback. Se avsnitt 10.3 respektive 8.4.

Feedbacknätverk kan diagrammatiskt åskådliggöras på olika sätt. Figur 23 ger två enkla exempel på ett av dessa sätt. Vi ser dels ett nätverk med

tre noder där alla enheter är förbundna med alla, dels ett feed-forwardnät med tre adderade återkopplingar. I båda fallen används pilar för att ange riktningen på förbindelserna; ”ut” är i det senare fallet uppåt, och det förutsätts att vikterna på alla utritade förbindelser är skilda från 0.



Figur 23. Två rekurrenta nätverk. Förklaring: Se text.

Rekurrenta, särskilt maximalt rekurrenta, nät åskådliggörs dock bättre i ett *fullständigt Hinton-diagram*. I ett sådant diagram förekommer alla enheter såväl på input- som outputsidan. Se figur 34 i avsnitt 5.2!

Signaldynamik i rekurrenta nätverk

I ett feed-forward-nät passerar en signal genom hela populationen av enheter på ett antal tidssteg som motsvarar antalet lager av förbindelser. Varje enhet har då aktiverats en och endast en gång. Sedan händer inget mer förrän nätverket får nästa input utifrån. Så enkelt är det inte i feedback-nät. I sådana kan en signal i princip fortsätta att cirkulera, i mer eller mindre ursprunglig form, i all evighet utan att man behöver ge mer input till nätverket. Någon funktionell grund för en indelning i lager finns därför inte.

Man kan i många fall inte ens urskilja några enheter som levererar signal *bara till omvärlden*; i typiska feedback-nät skickar ju alla enheter signaler tillbaka till andra enheter i nätverket. Inte heller finns det alltid några enheter som representerar signaler *bara från omvärlden*. Frågor som uppstår är: Genom vilka enheter ger man input till, respektive tar ut output från, ett återkopplat nätverk? Och *när* ska man läsa av output från ett rekurrent nätverk?

Svaret på den första frågan varierar mellan olika nätverkstyper och mellan olika användningar av dem. Förenklat kan man säga, att mimimalt återkopplade nätverk (av typ modifierade feed-forward-nät) i tekniska användningar behåller de input- och outputlager som de skulle ha om man *bortsåg* från feedback-kretsarna, medan man i maximalt rekurrenta nät ofta väljer att låta *alla* neuron vara både input- och outputenheter. Den andra frågan kan besvaras på två principiellt olika sätt. Antingen läser man av det relevanta aktivitetsmönstret (i alla neuron eller i de dedicerade outputneuronen, om det finns sådana) i *varje* tidssteg, dvs. man betraktar output som en tidsserie, som en pågående process. Eller så väljer man att se det *slutliga* aktivitetsmönstret som nätverkets output.

Det senare valet är framförallt meningsfullt för sådana återkopplade ANN som alltid *har* ett slutmönster, det vill säga sådana för vilka signalen så småningom stabiliseras och blir konstant, oavsett vilket inputmönstret var. Med hjälp av de begrepp som vi introducerade i avsnitt 1.4 kan vi formulera detta villkor som att *alla attraktorer för aktivitetsvektorn måste vara punktattraktorer.* Man talar ofta om dylika nätverk som ”attraktornätverk”. Det mest kända exemplet på sådana attraktornätverk är Hopfield-nätet, som vi skall beskriva i kapitel 7.

4.5 Vad kan neurala nätverk göra?

Nätverksfunktioner

Vi har nu kastat en blick på hur den enskilda noden i ett ANN fungerar och hur sådana noder kan kopplas samman till nätverk. Vad kan då ett artificiellt neuralt nätverk uträtta? Låt oss först ge en övergripande, abstrakt beskrivning som i första hand är tänkt att vara applicerbar på feedforward-nät. En motsvarande definition kan ges för återkopplade nätverk, men vi ska inte genomföra denna här.

Rent matematiskt sett realiserar ett feedforwardnätverk med givna vikter ett visst *funktionssamband* mellan input och output. För varje värde på inputvektorn får ju outputvektorn ett bestämt värde, och det är detta som definierar en funktion. Alltså:

(4.5.1) $\mathbf{o} = F(\mathbf{i})$

(Vi föregriper här ett beteckningssätt som vi ska beskriva närmare i kapitel 5, nämligen att representera vektorer och matriser med fetstilta symboler.) Alltefter indatas karaktär och vilken aktiveringsfunktion enheterna i nätverket har, kan funktionen F ta emot antingen diskreta eller kontinuerliga argument (**i**) och lämna ifrån sig antingen diskreta eller kontinuerliga funktionsvärden (**o**).

Vi kommer hädanefter att referera till F som en *nätverksfunktion.* Ändrar man vikterna i nätverket kommer det förstås att bete sig i enlighet med en *annan* nätverksfunktion, eftersom vikterna har betydelse för hur signalerna behandlas. En mer fullständig beskrivning av funktionssambandet är alltså

(4.5.2) $$\mathbf{o} = G(\mathbf{i}, \mathbf{w}_{ij})$$

där $\mathbf{w}_{ij}$ är hela uppsättningen av vikter i nätverket (viktmatrisen, avsnitt 5.2) och där G beskriver nätverkets beteende under alla möjliga sådana uppsättningar.

Funktionsapproximation, träning och generalisering

För en stor grupp av neurala nätverk gäller nu, att man genom en lämpligt avvägd ändring av vikterna $\mathbf{w}_{ij}$ kan få nätverksfunktionen F att med godtycklig grad av noggrannhet approximera (nästan) vilken som helst *önskad* matematisk funktion, dvs. man kan i princip få det att transformera input till output på vilket som helst, i förväg stipulerat sätt. Till yttermera visso behöver man inte programmera in den relevanta viktändringen direkt (det kan vara mycket svårt att räkna ut hur den skulle se ut) utan kan åstadkomma den genom *träning*. Denna innebär att man *presenterar ett antal input- och outputdata som (mer eller mindre exakt) exemplifierar den önskade funktionen,* och låter nätverket självt modifiera sina vikter i ljuset av dessa data.

När man på detta sätt styr nätverkets prestation i riktning mot en förutbestämd, önskad prestation kan man tala om *styrd* eller *övervakad inlärning*. För skillnaden mellan dessa begrepp se nedan, avsn. 4.6 och 6.2!

Genom övervakad inlärning kan man, för det första, få nätverket att med hög noggranhet approximera en funktion som man redan har en explicit formel för. Då tränar man helt enkelt nätverket på en uppsättning data

som genererats i enlighet med denna formel. Viktigare är dock ett annat fall, nämligen då det finns en mängd givna data som förmodligen exemplifierar något funktionssamband, men där vi känner inte till någon formel för detta samband i förväg. Typexemplet är data som genererats i en empirisk, vetenskaplig undersökning. Nätverket kan i detta fall hjälpa oss att *hitta* generella samband i data, och kan utifrån de samband det hittar hjälpa oss att *förutsäga* kommande observationer.

Det sagda kanske låter märkvärdigt, men är inte konstigare än att vi med traditionella statistiska metoder och utifrån givna data kan formulera generella modeller som hjälper oss att förutse framtiden. Vad ANN tillför statistiken är ett antal nya sätt att hitta *komplexa* samband i data. Som redan nämnts kan neurala nätverk t.o.m. avslöja *godtyckligt* komplexa samband. Denna styrka är tyvärr delvis också en svaghet. Med den formuleringen antyds ett viktigt problem som man brukar kalla *generaliseringsproblemet,* och som är av fundamental betydelse i ANN-teorin likaväl som i traditionell statistikteori (och filosofisk kunskapsteori). Vi ska nu kort beskriva detta problem.

Givet en ändlig mängd av data är det i allmänhet inte särskilt svårt att hitta ett neuralt nätverk vars nätverksfunktion beskriver dessa data med hög precision. Om man tycker att ett enkelt ANN inte fungerar tillräckligt bra för denna uppgift, kan man alltid hitta ett mer komplext nätverk som utför uppgiften bättre. Oftast vill man emellertid ha nätverket till något mer. Man vill nämligen att det ska ge rätt output på inputs som *inte* hörde till den ursprungliga datamängden (*träningsmängden*). För att nätverket ska kunna göra det, dvs. prestera bra på inputs ur en *testmängd*, måste det ha hittat fram till en nätverksfunktion som stämmer tillräckligt bra med *både* träningsfallen och testfallen. Och detta är inte på något sätt en självklar följd av anpassningen till träningsdata, för hur ska nätverket ”veta” utifrån dem hur testfallen kommer att vara beskaffade? Beklagligtvis är det så, att komplexa, kraftfulla nätverk – som är duktiga på funktionsapproximering i träningsmängden – löper en särskilt stor risk att prestera betydligt sämre på testdata! Varför det är så, och vad man ska göra för att förbättra komplexa nätverks förmåga till generalisering, ska vi diskutera i avsnitt 9.1.

De närmaste avsnitten skall handla om några alternativa sätt att beskriva vad ett ANN gör, om olika slag av nätverksfunktioner och om grundläggande typer av biologiska och tekniska tillämpningar (med tonvikt på de senare).

Mönsterassociation: att koppla ihop saker och ting

Vad ett artificiellt neuralt nätverk av feedforwardtyp gör är alltså, som vi just förklarat, att det realiserar en nätverksfunktion. Detta allmänna förhållande kan också uttryckas som att neurala nätverk är *mönsterassocierande* strukturer eller algoritmer, i den meningen att varje inputmönster ger ett bestämt outputmönster. Beteckningen ”mönsterassociation” används dock framförallt i ANN-tillämpningar där intresset är koncentrerat på att nätverket kopplar ihop *enskilda* input-output-par. Ett exempel är modellering av associativ inlärning, som denna av tradition beskrivits hos människor och andra biologiska organismer (jämför avsnitt 2.2–2.3 ovan). Vi kommer senare, i anslutning till sådan modellering (avsnitt 6.2), att beskriva linjära nätverks förmågor i termer av ”inlärd mönsterassociation”.

Vi ska nu skilja mellan några olika specifika typer av funktionssamband som ett ANN kan realisera, approximativt eller exakt, och kort nämna vilken relevans dessa specifika funktionstyper kan ha i biologiska och tekniska sammanhang.

Koordinattransformationer: att säga samma sak i olika språk

Under vissa villkor kan ett neuralt nätverk behandla inputmönster på ett sätt som kan tolkas som en *koordinattransformation*. Det innebär att inputvektorn och motsvarande outputvektor kan uppfattas som ekvivalenta beskrivningar av samma datapunkt, men uttryckta i olika koordinatsystem.

Antag att vi beskriver läget för Sveriges städer i ett rätvinkligt koordinatsystem, som har origo i Stockholm och en y-axel som pekar mot norr. Sedan byter vi till ett annat system med origo i Göteborg och y-axeln mot nordväst. Varje tänkbar beskrivning av en svensk stad i det första systemet motsvaras då på ett visst sätt av en beskrivning i det andra, och vice versa. Den funktion som överför koordinaterna i det ena systemet till rätt koordinater i det andra kan realiseras av ett linjärt neuralt nätverk (se avsnitt 5.2).

Även icke-linjära nätverk med graderbar aktivering kan ofta uppfattas som att de utför koordinattransformationer. Villkoret är att nätverksfunktionen har en *invers,* dvs. är omvändbar. Punkter med *olika* beskrivning i det ena koordinatsystemet får ju inte ha *samma* beskrivning i det andra!

Men givet att en nätverksfunktion har en invers, så kan man se den som en koordinattransformation. Ett annat sätt att uttrycka saken är att dylika nätverk utför en *omkodning utan informationsförlust*.

Dimensionsreduktion: att förenkla beskrivningar

Ett feed-forwardnätverk med graderbar aktivitet och fler inputnoder än outputnoder utför vad man kallar en *dimensionsreduktion*. För att anknyta till resonemanget i föregående stycke så kan man geometriskt tolka dimensionsreduktion som en *projektion* av ett rum av högre dimensioner på ett rum av lägre dimensioner. Projicerar vi exempelvis alla punkter på Sveriges karta vinkelrätt på förbindelselinjen mellan Stockholm och Göteborg (och dess förlängning), och utnämner Göteborg till origo, så får vi ett endimensionellt rum från ett tvådimensionellt. En hel mängd punkter på Sveriges tvådimensionella karta kommer då att erhålla samma beskrivning i det nya, endimensionella koordinatsystemet.

En dimensionsreduktion innebär med andra ord att information går förlorad. Hur negativt detta är beror på inputdatas struktur och vad man vill använda outputdata till. Om vi i det lilla kartexemplet begränsar oss till att betrakta Sveriges städer, *kan* det ju hända att två städer aldrig hamnar på samma punkt på linjen. Det kan också hända att den ordinala och metriska information som läget på linjen innehåller kan utnyttjas för specifika ändamål. Kanske man till exempel kan förutse årsmedeltemperaturen för en stad ganska bra utifrån dess läge på linjen?

Det resonemang som vi just har fört har stor betydelse för ANN-teorin, inte bara på grund av att många neurala nätverk själva utför en dimensionsreduktion. Man ställs nämligen inte sällan inför frågan om man skall genomföra en dimensionsreduktion av något slag för de givna data, *innan* man låter det neurala nätverket ta hand om dem. Fördelarna med en dimensionsreduktion i ”förprocessandet” av data är att informationen blir lättare att hantera men, framförallt, att generaliseringsproblematiken (avsnitt 9.1) får en något mindre dignitet. De resultat man får kan med andra ord förväntas vara mer robusta.

Projektion på högre-dimensionella rum

I analogi med vad som sades i föregående stycke kan man beskriva ett ANN med *fler* outputnoder än inputnoder som att det projicerar inputdata

på ett rum av *högre* dimensioner. Denna projektion kan också uppfattas som en speciell form av omkodning, som man ibland kallar ”gles kodning” (datapunkterna hamnar ju glesare i det större rummet). En flerlagrad perceptron med en inputenhet och åtta outputenheter kan exempelvis, om den tränas med någon lämplig algoritm, koda om de binära talen 000, 001, 010, 011, 100, 101, 110 och 111 till exklusiv aktivitet i respektive outputneuron nr 1, 2, ... 8 (en s.k. 1 av N-kodning). En sådan omkodning av data kan vara av stor betydelse för möjligheten att lösa ett givet problem.

Diskussionen i detta och föregående stycken har också relevans för modelleringen av verkliga neurala nät. Det finns nämligen anledning tro, att biologiska neurala nätverk som utför transformationer av koordinater och liknande operationer har stor betydelse för samordningen av olika neurala strukturer. Här är två möjliga exempel: Vi kan utan svårighet när som helst sätta höger pekfingers topp mot vänster pekfingers topp, utan att först titta efter var vänster pekfinger befinner sig. Det är rimligt att förklara detta genom att ett neuralt nätverk utför en koordinattransformation mellan vänster och höger kroppshalvas inbördes spegelvända motoriska delsystem. Likaså måste nervsystemet hos en groda, som skickar iväg sin tunga mot en punkt i synfältet där en fluga just rörde sig, ”översätta” signalen som lokaliserar stimulus på näthinnan till en motorisk signal som gör att tungan träffar flugan. Även det mänskliga nervsystemet utför rutinmässigt sådana automatiska översättningar från det visuella till det motoriska systemet – ser vi ett vattenglas på bordet framför oss, så ”vet” vår hand automatiskt hur den ska göra för att gripa det.[106]

Ett tredje exempel ges av de i dag så omdiskuterade ”spegelneuron” som tycks vara inblandade i vår igenkänning av andras intentioner.[107] Åtminstone hos många apor, och med stor sannolikhet hos människor, finns det neuron som signalerar både då man utför en viss aktivitet (t.ex. griper ett äpple) och då man ser en annan individ utföra denna handling. Så här kanske det går till när vi ser en målvakt kasta sig efter en boll och tänjer på vår egen kropp för att hjälpa honom: först sker en transformation av rumsliga koordinater i det visuella systemet, så att vi uppfattar situatio-

[106] Kap. 10 i klassikern Churchland (1986) innehåller flera intressanta exempel på kända eller tänkbara koordinattransformationer i nervsystemet.

[107] Stamenov & Gallese (red.) (2002) är en innehållsrik antologi om spegelneuron. För en aktuell sammanfattning och diskussion om dessa frågor se Jeannerod (2006). Man kan notera att den franske filosofen Merleau-Ponty (1962) [1945] mycket tidigt förklarade imitation i termer av *koordinattransformationer* i det integrerade system för varseblivning och handling som han kallar ”kroppsschemat”.

nen som den ser ut från den andres perspektiv. Därefter äger den ovan nämnda automatiska översättningen från visuella till motoriska koordinater rum, dvs. vår kropp förbereder sig för att utföra den handling som vi ser den andre utföra.

Den detaljerade analysen av dylika biologiska nätverk hör hemma i en mer avancerad framställning än den här boken, men läsaren får gärna ha dem i minne som möjliga tillämpningsområden av de teorier som presenteras nedan. Detta gäller inte minst inlärningsaspekten. Biologiska neurala nätverk av den typ vi nu talat om kan i större eller mindre grad bygga på inlärning. Grodans flugfångande förlitar sig på ett genetiskt programmerat nätverk, medan motsvarande koordination mellan öga och hand hos oss människor delvis är inlärd. Utifrån dagens kunskapsläge är det svårt att veta om denna inlärning kan förklaras av någon av de ANN-modeller som vi ska tala om senare, men det är under alla omständigheter intressant att tänka på de givna exemplen som möjliga applikationer av modellerna ifråga.

Mönsterklassifikation: att sortera data

Många neurala nätverk realiserar alltså funktioner som *inte* är omvändbara. Det gäller också de nätverk som har *diskret* output kombinerad med kontinuerlig input och/eller fler inputnoder än outputnoder. Vi har också här att göra med en informationsförlust. Den information som är kvar kan dock vara tillräcklig för specifika syften, och rentav avgörande för våra beslut.

Sådana nätverk har en stor användning när vi vill *klassificera* data i två eller flera klasser. I fallet med två klasser vill man då att nätverksfunktionen ska hämta sina argument ur mängden av inputdata men som värden ha (till exempel) 0 och 1 i en enda outputnod. Nollan representerar då den ena klassen och ettan den andra. Andra kodningar av klasstillhörighet är också möjliga, exempelvis som aktivitet i den ena respektive den andra av två outputnoder. Den sistnämnda möjligheten är ett specialfall av 1-av-N-kodning. Denna typ av kod är ett mycket naturligt val i fallet med fler klasser än två. Genom inlärningen försöker man modifiera nätverkets vikter så att det klassificerar indata på ett i förhand bestämt sätt (övervakad inlärning).

Det ska redan nu poängteras att ANN med graderad output också kan användas för klassifikationsuppgifter. Då måste man givetvis ha någon re-

gel för att översätta outputvärdena till beslut om klasstillhörighet. Exempelvis kan, om outputneuronets aktivitet är begränsad mellan 0 och 1, värden $\geq 0{,}5$ få representera den ena klassen och värden $< 0{,}5$ den andra. Detta fungerar oftast bra oberoende av vilken nätverksalgoritm som det handlar om i övrigt. Om nätverkets outputenhet har en logistisk aktiveringsfunktion, och man har använt en viss typ av felfunktion (se avsnitt 4.6 och 9.3) vid träningen, får man dessutom på köpet att outputnivån hos det tränade nätverket under vissa omständigheter kan tolkas som *sannolikheten* för att input tillhör den första klassen. För fler än två klasser brukar man använda N outputnoder och låta den nod som får maximal aktivitet avgöra klasstillhörigheten. Med logistiska outputenheter och den speciella felfunktion som nämndes nyss kan den graderade output i varje outputneuron då ofta tolkas som sannolikheten för att den aktuella input tillhör den klass som outputenheten representerar. – Det följande resonemanget är giltigt för alla klassifikationsnätverk, oberoende av hur deras output organiserats.

De data som ska föras till en och samma klass kan inbördes vara mycket olika; man kan ändå ofta lära nätverket att föra samman dem. En viktig del av ANN-teorin handlar, enkelt uttryckt, om *hur* olika data kan vara och ändå föras till samma klass – och omvänt, hur *lika* de data kan vara som nätverket separerar. Flera typer av neurala nätverk, bland annat flerlagrade nätverk vars dolda enheter har sigmoida aktiveringsfunktioner, kan i princip tränas att lösa *alla* konsistenta[108] uppgifter som går ut på att separera två eller flera klasser av data. Problemet är här, som redan omtalats, att generaliseringsförmågan gärna minskar i takt med att metoden blir mer kraftfull. Mycket mer om detta nedan (särskilt i avsnitt 9.1)!

Mönsterklassifikation genom övervakad inlärning är en vanlig teknisk användning av ANN. Det kan därför vara lämpligt att här – innan vi går vidare till andra typer av ANN-modeller och andra möjliga tillämpningar – påminna om den mest grundläggande anledningen till att övervakad inlärning är så användbar i praktiken.

Att *explicit* programmera en maskin för en mönsterklassifikationsuppgift kan vara mycket svårt, eftersom det inte sällan är svårt eller omöjligt att med tillräcklig precision formulera de relevanta skillnaderna mellan de mönster som skall separeras. När det gäller att skilja mellan två typer av

[108] Konsistens betyder här att det får inte finnas två identiska inputs som hör till olika klasser.

tillverkade objekt, exempelvis äkta och falska hundrakronorssedlar, går det visserligen att säga exakt hur ett äkta objekt ska se ut. Men om det gäller att separera två typer av "naturgivna" objekt från varann, exempelvis röntgenbilder av friska respektive sjuka lungor, har vi inte tillgång till någon tillverkarspecifikation. Inte ens den röntgenläkare som i praktiken utan problem kan skilja sjukt från friskt kan i allmänhet tala om exakt *hur* han/hon egentligen gör denna bedömning – dvs. vad som egentligen skiljer de två klasserna av mönster från varann. Detta är för övrigt bara ett bland många exempel på att vi inte kan teoretiskt beskriva alla våra praktiska färdigheter. Jämför ovan om procedurell och deklarativ kunskap och om perceptuellt minne (avsnitt 3.1)!

Antag då att vi vill bygga en maskin som kan separera "friska" från "sjuka" röntgenbilder – är det överhuvudtaget möjligt, när inte ens experten kan tala om vad som definierar en frisk respektive en sjuk bild? Ja, det är faktiskt möjligt ändå. Kom ihåg att läkarens egen praktiska färdighet vad gäller mönsterseparation i stor utsträckning är *inlärd genom exempel.* Man blir inte en bra röntgenläkare enbart (eller ens huvudsakligen) genom att lära sig listor på röntgenkriterier för friskt och sjukt (fullständiga sådana listor finns ju inte), utan genom att utsätta sig för ett stort antal verkliga fall för vilka det finns ett "facit", dvs. information om vilken klass (frisk/sjuk) de tillhör. Facit tillhandahålls ibland av en mer erfaren röntgenläkare, men ofta av någon som har tillgång till en oberoende metod för att ställa diagnos. Efter att ha sett många fall har läkaren till slut förhoppningsvis förvärvat en god diskriminationsförmåga. Nåväl – om människor kan göra så, låt oss imitera människan och bygga en maskin som kan lära sig genom exempel ur ett facit! Detta är grundtanken bakom användandet av ANN för mönsterklassifikation.

Till denna grundidé kommer att ett ANN som simuleras med ett datorprogram oftast kan arbeta både snabbare och pålitligare (på ett mer konstant sätt) än en människa. Sist men inte minst finns det inga teoretiska gränser för hur många och hur varierande data ett ANN kan matas med. Mycket talar därför för att ett ANN kan tränas att bli inte bara *lika bra* som, utan rentav *bättre* än, människan när det gäller dylika klassifikationsuppgifter.

Kategorisering: att klumpa ihop data

I statistiken talar man om metoder för "clustering", och syftar då på algoritmer för att sammanföra *liknande* data i klasser eller hopar (clusters).

Dessa hopar skapas av algoritmen själv. Här är det alltså inte tal om att ”tvinga in” inbördes mer eller mindre olika data i på förhand bestämda klasser. Det finns ett stort antal clusteringmetoder i bruk och flera neurala nätverk som utför clustering. De senare är oftast av det kompetitiva slaget och har alltså ett antal outputneuron med diskret aktivering, varav endast ett blir aktivt för en given input.

Det som dessa nätverk gör kallas inte sällan ”oövervakad klassifikation”, men för att betona att det inte alls rör sig om sortering till i förväg bestämda klasser kan man istället tala om *kategorisering*. Kategorisering innebär förstås också reduktion av informationen i inputsignalen. Ett intressant specialfall är Kohonens ”självorganiserande karta”, som förutom att skapa kategorier placerar in dem på en tvådimensionell karta där rumslig närhet avspeglar likhet (se avsnitt 8.1). Här kan man därför tala om en dimensionsreduktion.

Mönsterigenkänning: att skilja välkänt från nytt

Särskilt vid användandet av attraktornätverk, dvs. återkopplade neurala nätverk där signalen alltid går till någon punktattraktor, använder man ofta termen *mönsterigenkänning*. Om ett diskret sådant nätverk presenteras med en input som råkar vara ett attraktortillstånd, och man iakttar hur aktiviteten utvecklas i de följande tidsstegen, så kommer nätverket inte att ändra tillstånd. Output blir därför densamma som input. Detta uttrycker man då som att nätverket ”känner igen” eller ”accepterar” inputsignalen i fråga. (Jämför gärna våra resonemang ovan om habituering till välkända stimuli, avsnitt 2.1!) Om input däremot inte svarar mot en attraktor, kommer signalen att förändras över tiden, ända till dess att nätverket kommer till en attraktor. Output blir då skild från input. Ibland beskriver man detta som att nätverket inte kände igen, eller förkastade, inputsignalen, men ibland (särskilt om slutmönstret ligger nära startmönstret) kan det ha en poäng att säga att nätverket kände igen inputmönstret som en förvanskad version av slutmönstret. Många attraktornätverk har nämligen (liksom många feedforwardnätverk) en inbyggd förmåga till *mönsterkomplettering* (jämför avsnitt 1.1 och 7.2).

Intressant är nu att man genom inlärning kan modifiera ett dylikt nätverks vikter så att det får *förutbestämda attraktorer*. Med andra ord, man kan träna det att ”känna igen” förutbestämda inputmönster (och förvanskade versioner av dem). Denna träning består i princip av upprepad exponering för mönstren i fråga. Se vidare nedan, avsnitt 7.1.

Regression: att rekonstruera en kontinuerlig funktion

Neurala nätverk som har en kontinuerligt graderad output kan användas för det som i statistiken kallas *regression*. Med denna term avses en anpassning till data av en kontinuerlig funktion (eller mer generellt till väntevärdet av en kontinuerlig betingad sannolikhetsfördelning, se avsnitt 5.1) genom att man modifierar funktionens parametrar. Välkänd för alla som läst elementär statistik är *linjär* regression, som innebär att man letar reda på den räta linje y = ax + b som (i en viss mening) bäst passar med en given datamängd i x-y-planet. Utifrån sannolikhetsteoretiska resonemang kan man visa, att denna linje (under vissa förutsättningar) representerar den (i viss mening) bästa hypotesen om vilket reellt samband som ligger bakom de iakttagna data. Se vidare avsnitt 5.1.

Om just linjär regression ska vi tala mer i avsnitt 6.2, eftersom den kan utföras med hjälp av linjära neurala nätverk. Men det finns också *olinjär* regression, av många olika slag. Man kan således välja att anpassa ett polynom av givet gradtal, till exempel en kvadratisk funktion, till datamängden och på så vis försöka hitta ett underliggande samband av just *denna* matematiska typ. Det kan visas att om man tillåter polynom av obegränsat gradtal i sin regressionsfunktion, så kan den anpassas till (nästan) varje konsistent mängd av talpar (x,y) och därför också approximera (i stort sett) varje möjligt funktionssamband. Polynomen sägs därför vara *universella approximatorer.*

En annan klass av universella approximatorer utgörs av flerlagrade neurala nätverk med sigmoida aktiveringsfunktioner i de dolda enheterna. (För att vara optimala för regressionsuppgifter bör de däremot ha linjära outputneuron.) Det är alltså denna egenskap som gör att de kan tränas till att modellera praktiskt taget vilket samband som helst mellan input- och outputvariabler. De är därför mycket kraftfulla redskap för att hitta ickelinjära samband mellan variabler. Dessutom är de på flera sätt överlägsna traditionella metoder, som t.ex. den polynomapproximation som just nämndes. Framförallt behöver de relativt få data för att ge välbestämda lösningar (givet samma krav på anpassning till data).[109] Men de är ändå

[109] Ett annat sätt att uttrycka detta är att neurala nätverk inte drabbas av ”dimensionalitetens förbannelse” (*the curse of dimensionality)* i samma mån som t.ex. polynomapproximation. När ett problem som man försöker lösa med en viss typ av matematisk modell växer i komplexitet, växer också kravet på komplexitet hos modellen. Vid polynomapproximation måste man ta till termer av högre gradtal, medan man i ett neuralt nätverk kan behöva lägga till dolda enheter. Men i det senare fallet växer alltså inte den nödvändiga komplexiteten hos modellen lika snabbt som i det förra,

behäftade med samma generaliseringsproblematik som andra typer av kraftfulla neurala nätverk; mer om detta senare.

Prediktion: att förutse framtiden

Givet en tidslig process, eller en annan enkelriktad sekvens av tillstånd, kan man ge ett neuralt nätverk delsekvenser av tillstånd som input och styra dess output så att den överensstämmer med *nästa* tillstånd i sekvensen. Denna uppgift, som ju innebär att approximera en funktion från ett antal på varandra följande tillstånd till nästa, kan inte helt oväntat lösas med (speciellt utformade) feedforward-nätverk. Ett intressant alternativ är återkopplade nätverk, som lagrar information om de närmast föregående inputs med hjälp av rekurrenta förbindelser. Dessa speciella typer av nätverk skall avhandlas i kapitel 10.

Inlärd kontroll

I en viss, inte helt onaturlig mening av ”kontroll” kan alla uppgifter som rör övervakad klassifikation eller funktionsapproximation (regression) klassificeras som ”kontrollproblem”. Vi vill ju nämligen kontrollera nätverkets output så att den överensstämmer med vissa önskade värden. När man i tekniska sammanhang talar om ”kontroll” handlar det däremot vanligen om system som genom sin output påverkar någon parameter i omgivningen, som i sin tur har ett ”önskevärde”. Systemet tar fortlöpande emot information om det aktuella värdet på den parameter som skall kontrolleras, eller om värdet på någon lämplig funktion av den (t.ex. avvikelsen från önskevärdet).

Det är en intressant uppgift för ett neuralt nätverk att modellera hur ett redan existerande kontrollsystem fungerar. Men man kan också ge ett neuralt nätverk uppgiften att *skapa* ett fungerande kontrollsystem genom att träna det mot systemets ”önskevärde” – alltså det värde som termostaten är inställd på att reglera temperaturen till, eller den plats man vill roboten skall gå till. Dylik *inlärning av kontroll* ställer stora krav på nätverkets arkitektur och inlärningsalgoritm. Värdet på den parameter som skall kontrolleras beror som regel på ett delvis okänt sätt av de tillstånd hos systemet, som dess användare har under mer omedelbar kontroll. Nätverket måste med andra ord läras att modellera även detta beroende.

---

varför det behövs färre inputdata för att nå fram till en lösning som generaliserar bra. För en förklaring av detta förhållande se Bishop (1996), avsnitt 1.4–1.7.

Det rör sig dessutom oftast om långsiktiga beroenden. Detta gör att en ”direkt” modellering av verkliga kontrollproblem med hjälp av övervakad inlärning (med önskevärdet på målvariabeln som önskad output) i ett feed-forwardnät långt ifrån alltid fungerar bra. Återkopplade nätverk som tränas med evolutionära algoritmer kommer ofta till användning.

Övriga optimeringsproblem

Som ”optimeringsproblem” kan man beteckna matematiska uppgifter där det gäller att hitta de variabelvärden som ger det “bästa” värdet på en utvald målvariabel. Att hitta ett system som kontrollerar en industriprocess så exakt som möjligt kan alltså sägas vara ett optimeringsproblem, liksom uppgiften att hitta den klassifikation av givna data som ger så få fel som möjligt. Det finns dock många optimeringsuppgifter som inte faller under några av de tidigare rubriker vi använt för vad ett neuralt nätverk kan göra, men som man prövat ANN för att lösa. Vi har i denna framställning valt lösningen av en enkel variant av det berömda TSP (Traveling Salesman Problem) med SOM (Self-Organising Map) som en illustration av de neurala nätverkens möjligheter på området. Se avsnitt 8.3.

Modeller med och utan strukturlikhet

Som framgått ovan kan man använda ett ANN som en modell av ett biologiskt neuralt nätverk. Då tänker man sig oftast *både* att ett visst ANN realiserar samma input-outputfunktion som ett visst biologiskt neuralt nätverk, *och* att detta sker på grund av att enheterna i modellen motsvaras av neuron i det biologiska nätverket, förbindelserna mellan modellens enheter motsvaras av faktiska (uppsättningar av) synapser mellan de verkliga neuronen, etcetera. Vi kan då prata om en *konkret strukturlikhet* mellan modell och verklighet.

I andra biologiska sammanhang kan sambandet vara av ett annat slag. En ANN-enhet behöver inte stå för ett biologiskt neuron, utan kan t.ex. motsvaras av en grupp av samverkande neuron. Aktiviteten i ANN-enheten i fråga står då kanske för den genomsnittliga aktiviteten i neurongruppen, medan modellen som helhet beskriver sambandet mellan de genomsnittliga aktiviteterna hos grupper av neuron i två anatomiska skikt. Sådana modeller kan sägas vara *abstrakt strukturlika* den verklighet som de modellerar.

Från abstrakt strukturlika modeller är steget inte alltför långt till de fall, då ett artificiellt neuralt nätverk modellerar relationen mellan input och output i ett system utan att de olika komponenterna i ANN-modellen är avsedda att kopplas till *några som helst* konkreta komponenter i det system som ska modelleras. Att artificiella neurala nätverk är universella approximatorer implicerar nämligen inte alls, att ett system vars beteende approximeras av ett visst nätverk måste ha samma inre struktur som nätverket självt. Detta följer redan av det förhållandet att samma system alltid kan modelleras genom neurala nätverk med olika struktur, exempelvis med olika antal enheter och t.o.m. med olika antal skikt av enheter.

I dessa fall kan man tala om modeller *utan anspråk på strukturlikhet,* eller, något förenklat, *icke strukturlika* modeller. Exempel ges av många av de neurala nätverk som tränas för kontroll av industriprocesser eller styrning av robotars beteende. Man är här i allmänhet inte särskilt intresserad av den inre strukturen hos den kontrollerande algoritmen, bara den fungerar. Neurala nätverksmodeller för intracellulär signalering kan också vara av denna icke-strukturlika typ, men kan ibland snarare klassificeras som abstrakt strukturlika modeller.

4.6 Inlärningsalgoritmer för ANN

Efter att således ha skisserat hur artificiella neurala nätverk behandlar signaler, och några sätt på vilka vi kan utnyttja deras signalbehandlande egenskaper, skall vi nu titta närmare på hur dessa egenskaper kan modifieras genom “erfarenheten”, alltså hur nätverken kan lära sig genom exempel. Detta är ett mycket intressant område ur både matematisk och biologisk synvinkel. Matematiskt, eftersom ett intelligent val av inlärningsalgoritm kan vara nyckeln till att lösa ett svårt problem. Ändringar i ett nätverks vikter ändrar det sätt på vilket det behandlar en given signal, och det gäller alltså att hitta ett klokt sätt att ändra vikterna så att nätverket kommer att göra det man vill. Biologiskt, eftersom det fortfarande är i hög grad okänt hur inlärning går till i verkliga nervsystem. Det finns alltså en stor efterfrågan på nya och bättre modeller för detta.

Det är värt att lägga märke till, att de två huvudtyperna av mål för ANN-forskning ger divergenta resultat när det gäller val av inlärningsalgoritm. De biologiskt mest realistiska algoritmerna av dem som hittills föreslagits är således inte särskilt bra för matematiska beräkningsändamål, och de

matematiskt mest kraftfulla algoritmerna är totalt orealistiska som modeller av verklig inlärning. Ingenting säger förstås att det alltid kommer att förbli på detta sätt. En vacker dag kanske någon hittar en inlärningsalgoritm som både är mycket kraftfull och biologiskt helt realistisk. För det måste ju finnas en sådan, med tanke på vår hjärnas förmågor...

De närmaste sidorna beskriver några grundläggande algoritmer för inlärning i ANN. (Fler kommer att nämnas senare i texten.) De kan i vissa stycken kanske vara svåra att förstå innan man fått detaljerade exempel på nätverk som använder sig av algoritmerna ifråga. Men det går ju bra att gå tillbaka och läsa om det följande partiet när du har kommit lite längre i boken!

Hebb-regeln

Denna algoritm, som finns i många varianter (se nedan), är som namnet säger en formalisering av Hebbs teori om förstärkning av synapsen mellan samtidigt aktiva neuron. Om vi symboliserar ändringen (i ett visst tidssteg) av vikten w_{ij} mellan enheterna X_i och X_j med Δw_{ij}, så säger Hebb-regeln (i sin enklaste form) att

$$\Delta w_{ij} = k \cdot x_i \cdot x_j \tag{4.6.1}$$

där k är en positiv konstant (inlärningskonstanten). Vikterna ändras med andra ord *i proportion till produkten av pre- och postsynaptisk aktivitet*. En konstant association mellan aktiviteterna i två element X_i och X_j leder då, precis som Hebb tänkte sig det, typiskt till att aktivitet i den ena enheten själv kan *åstadkomma* aktivitet i den andra. Man kan uttrycka detta som att nätverket har "lärt sig" associationen ifråga. Hebb-regeln erbjuder alltså, vilket vi visade redan i inledningskapitlet (avsnitt 1.1) ett enkelt förklaringsschema för associativ inlärning, exempelvis Pavlovsk betingning.

Hebbregeln är troligen den biologiskt mest realistiska av de i ANN-sammanhang vanliga inlärningsalgoritmerna. Långtidspotentiering (se avsnitt 4.2) har Hebb-liknande egenskaper.

Varianter av Hebb-inlärning

Med den vanliga Hebbregeln växer vikterna mellan aktiva enheter utan begränsning, vilket i många ANN-sammanhang är en nackdel (förutom att det är biologiskt orealistiskt). För att undvika detta har man föreslagit olika variationer av Hebbregeln i vilka vikterna begränsas av ett tak. Man kan då stipulera att en vikt alltid ska uppgraderas med en konstant fraktion av skillnaden mellan detta tak och den aktuella vikten. Ett annat, ofta använt alternativ är *normalisering* av vikterna. Detta betyder att summan av vikterna på de utgående förbindelserna från en enhet alltid sätts till 1. Båda förslagen kan kopplas till tänkbara biologiska mekanismer (t.ex. begränsad tillgång på transmittorsubstans).

En annan variant av Hebbregeln innebär att vikterna *minskar* om aktiviteten i det postsynaptiska neuronet är 0 (eller mindre än ett visst, positivt värde). På detta sätt kan man eventuellt bättre modellera det som i experimentalpsykologin kallas *utsläckning* av en inlärd association (se avsnitt 2.3). Ytterligare andra varianter har nyligen formulerats i samband med teorier för pulskodade neurala signaler. Man har, som nämnts i avsnitt 4.2, fått experimentellt stöd för att arten och graden av synaptisk förändring hos verkliga neuron är starkt beroende av tidsrelationen mellan presynaptisk och postsynaptisk aktivering. Detta inkluderar, att man i det fall då det postsynaptiska neuronet aktiveras *före* det presynaptiska får en *minskning* av synapsens styrka. Dessa idéer har dock än så länge inte blivit allmängods inom ANN-modelleringen, och att utveckla dem fordrar en tämligen avancerad matematisk begreppsapparat. Vi måste därför lämna dem därhän.

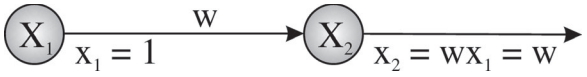
Delta-regeln

Vid styrd inlärning i ANN används ofta *felkorrigerande* algoritmer. Den enklaste av dessa är vad som ofta kallas "deltaregeln" eller "Widrow-Hoff-regeln". Låt oss anta att ett neuron X_i har en direkt förbindelse till ett annat neuron, X_j. Vi vill att en viss aktivitet i X_i skall ge en viss, förutbestämd aktivitet i X_j. Antag närmare bestämt att den förutbestämda (önskade) aktiviteten hos X_j är d_j ("d" står för "desired"), när aktiviteten hos X_i är x_i. Då säger deltaregeln att

$$\Delta w_{ij} = k \cdot x_i \cdot (d_j - x_j) \qquad (4.6.2)$$

där k är en positiv konstant. Vikten ändras med andra ord i proportion dels till aktiviteten i det "presynaptiska" neuronet, dels till *skillnaden* mellan önskad och verklig aktivitet i den "postsynaptiska" enheten. Poängen med detta arrangemang är att ändringen i w_{ij} blir mindre ju mindre felet ($d_j - x_j$) är, och om felet är noll ändras inte vikten alls.

Betrakta figur 24, som föreställer ett mycket litet och maximalt linjärt ANN (aktiveringsfunktionen är identitet), där aktiviteten hos X_1 hela tiden är 1, vikten w från början 0 och vi vill att output från X_2 skall bli 1:



Figur 24. Illustration av deltaregeln i ett linjärt mini-nätverk. Förklaring: se text.

I ekvationen ovan är x_2 i varje steg = w, och d_2 är från början = 1. Läsaren kan enkelt verifiera att med inlärningskonstanten k = 1/2 blir Δw vid första aktiveringen = 1/2, vid den andra = ¼, vid den tredje = 1/8, och så vidare. Vikten w kommer därför asymptotiskt att närma sig 1, liksom den verkliga output från X_2.

Deltaregeln ger goda resultat i helt linjära nätverk när dessa används för mönsterklassifikation eller mönsterassociation. Den kommer också till användning i andra nätverk som på ett väsentligt sätt använder sig av linjära neuron, exempelvis radialbasnätverk (avsnitt 9.6). Särskilt intressant är att man kan ge ganska enkla *konvergensbevis* för denna inlärningsalgoritm. Ett dylikt konvergensbevis innebär att man visar, att upprepad användning av algoritmen faktiskt så småningom leder till att det totala felet som hela nätverket gör minimeras. Detta är uppenbarligen fallet i mininätverket i figur 24, men så snart nätverket blir mer komplext är det inte längre trivialt. Mer om konvergensbevis nedan (avsnitt 6.2)!

Observera att deltaregeln i sig *inte* specificerar hur det går till när informationen om felet påverkar vikten. Särskilt: det finns inget neuron i vårt mininätverk som kodar storheten d_2, eller ($d_2 - x_2$). Att informationen om den senare storheten används vid modifierandet av vikten skall alltså *inte* tolkas som att någon neural enhet skickar en signal av storleken ($d_2 - x_2$) genom någon slags återkopplad förbindelse. Inlärning som på detta sätt använder sig av information som inte kodas in i själva nätverket

kan kallas *övervakad.* Begreppet *styrd* inlärning definieras då lämpligen så att det innefattar övervakad inlärning, men också algoritmer där styrinformationen representeras på något ”onaturligt” sätt i nätverkets aktivitet (jfr. avsnitt 6.2).

Det påpekade förhållandet är inget bekymmer i rent tekniska användningar av ANN, där användaren har full frihet att programmera in deltaregeln som ett moment i sin nätverksalgoritm utan att bekymra sig om ”hur det går till i verkligheten”. Det är däremot relevant för frågan om deltaregeln någonsin är förverkligad i verkliga nervsystem, vilket är omdiskuterat. Deltaregeln förutsätter ju, att synapserna i det lärande nätverket *på något sätt* får information om vilket det *önskade* resultatet (d_2) är, och att denna information *styr* den hastighet med vilken synapsernas effektivitet förändras. Någon biologisk mekanism som klarar av detta är inte känd. Dock kan ett återkopplat nätverk i princip göra samma sak som deltaregeln genom att en ”subtraktiv” neuronal krets först beräknar skillnaden ($d_2 - x_2$) och sedan återför denna storhet till systemet som en neuronal aktivitet. Därefter träder en Hebb-mekanism in och ändrar vikterna i enlighet med produkten mellan denna aktivitet och x_1. Ett sådant nätverk skulle alltså kunna göra samma sak som deltaregeln gör i ett övervakat nätverk, men genom en oövervakad mekanism. Visst skulle ett system som liknar detta kunna existera i hjärnan, och det kan vara mödan värt för neurofysiologerna att leta efter det!

Perceptronregeln

Denna regel är ett specialfall av deltaregeln för enlagrade binära nätverk med tröskel (x_i, x_j och d_j är alla 1 eller 0), och säger:

$$
\begin{aligned}
& x_j < d_j \;\rightarrow\; \Delta w_{ij} = k \cdot x_i \\
& x_j = d_j \;\rightarrow\; \Delta w_{ij} = 0 \\
& x_j > d_j \;\rightarrow\; \Delta w_{ij} = -k \cdot x_i
\end{aligned}
\qquad (4.6.3)
$$

där k är en positiv konstant. Man kan sammanfatta perceptronregeln så här: Om det blir rätt, ändra ingenting; om det blir fel, ändra vikten från aktiva presynaptiska enheter med en konstant storhet och i motsatt riktning mot felet.

För att göra perceptronregeln mer generellt användbar lägger man oftast till, att *tröskelns* storlek också skall ändras i en riktning som minskar

felet. Vi skall nedan berätta mer om regelns användning i perceptronen (avsnitt 6.1), och då utgår vi från en sådan formulering.

Back propagation of error

Deltaregeln är inte användbar i flerlagrade olinjära nätverk. Där ersätts den ofta av den mer komplicerade algoritm som kallas "back propagation of error", eller numer snarare av någon ännu mer sofistikerad variant av denna algoritm.

En grundläggande version av algoritmen kallas ofta den "generaliserade deltaregeln". Den innebär nämligen, liksom deltaregeln, att vikterna ändras *i proportion till deras bidrag till det totala felet i hela nätverkets output*. För den matematiskt kunnige kan den formaliseras på följande sätt. Låt $\partial E/\partial w$ beteckna den partiella derivatan av felet E med avseende på vikten w. Då säger regeln:

$$\Delta w_{ij} = - k \frac{\partial E}{\partial w_{ij}} \qquad (4.6.4)$$

Man beräknar $\partial E/\partial w_{ij}$ för olika w_{ij}, dvs. de enskilda vikternas bidrag till det totala felet, genom en stegvis procedur där man börjar med vikterna i det sista lagret och arbetar sig ner till det tidigaste. Det är denna procedur som utgör själva "back propagation"-idén. Vi skall berätta närmare om den i avsnitt 9.2.

Back propagation-algoritmen används i flerlagrade feedforwardnätverk för övervakad mönsterklassifikation, i återkopplade ANN för förutsägelse och kontroll, med mera. Även för denna regel finns intressanta konvergensbevis. Algoritmen betraktas oftast som biologiskt helt orealistisk, men det har gjorts försök att modellera den på ett biologiskt plausibelt sätt med hjälp av tillägg av rekurrenta förbindelser som skickar felinformation bakåt i nätverket, lager för lager (jämför vad vi nyss sade om deltaregeln). Man har utvecklat många alternativ till (4.6.4) för att göra den snabbare och effektivare, och inte minst för att säkra en bättre generaliseringsförmåga (se vidare avsnitt 9.2).

Vi skall slutligen mycket kort presentera två mycket speciella inlärningsalgoritmer, eller rättare sagt två klasser av algoritmer (eftersom de kommer i många versioner). Presentationen förutsätter delvis vissa matematiska begrepp som kommer att förklaras först i nästa kapitel.

Inlärningsregler för kompetitiva nätverk

Kompetitiva neurala nätverk är, som redan förklarats, nätverk där en given input aktiverar endast en nod, ”vinnaren”, i nästa lager. En typ av inlärningsalgoritmer för sådana nätverk kan kallas *Kohonen-regler*. För att förstå vad de går ut på behöver vi begreppet *viktvektor*. I ett neuralt nätverk med minst två skikt bildar vikterna mellan inputneuronen och ett givet neuron i det andra skiktet en vektor, som vi alltså kallar detta neurons ”viktvektor”. Kohonenreglerna förutsätter (i standardutförandet) att den vinnande noden i skikt nummer två utses med hjälp av *en jämförelse mellan å ena sidan inputvektorn, å andra sidan viktvektorerna för noderna i detta skikt*. Närmare bestämt vinner den nod vars viktvektor ligger *närmast*, i betydelsen: *på det minsta euklidiska avståndet från*, den aktuella inputvektorn. (För en förklaring av vad denna vinnarregel har att göra med den som vi beskrev i slutet av avsnitt 4.3, se avsnitt 5.2 och 8.1–8.2.) För vissa noder i det kompetitiva lagret – i första hand för vinnaren, vars viktvektor ju redan ligger nära inputvektorn i fråga – ”drar” man viktvektorn närmare eller i vissa fall längre bort från inputvektorn. Något informellt (jämför avsnittet om vektorer!) kan vi skriva, om $\mathbf{w}_j$ är vektorn av vikter från inputnoderna till noden X_j, $\mathbf{i}$ är inputvektorn och k är en positiv eller negativ konstant:

$$\Delta \mathbf{w}_i = k \cdot (\mathbf{i} - \mathbf{w}_i) \qquad (4.6.5)$$

Vilka element X_j som berörs, och hur, varierar mellan olika modeller. Se vidare avsnitten om Kohonen-nät nedan (8.1 och 9.5).

Om biologiska nätverk någonsin använder sig av inlärningsalgoritmer som liknar Kohonen-reglerna är inte känt. En annan grupp av inlärningsregler för kompetitiva nätverk används av ART-modellerna (Adaptive Resonance Theory). Dessa regler åstadkommer, liksom Kohonen-regeln, att den vinnande noden blir ännu bättre på att vinna på samma och liknande inputs. Eftersom de i grund och botten är baserade på Hebb-liknande mekanismer är det inte orimligt att de är biologiskt realiserbara. Se vidare avsnitt 8.4.

Genetiska (evolutionära) algoritmer

Dessa algoritmer innebär en fortlöpande *selektion* i *populationer* av nätverk. Nätverken som visar de bästa prestanda ”överlever” och får ”avkomma” där deras egenskaper rekombineras. Förbättring av populatio-

nens egenskaper åstadkoms också genom “mutationer”, dvs. slumpmässiga förändringar av nätverkens vikter. Till slut har man (i bästa fall) en population av neurala nätverk som alla klarar den förelagda uppgiften perfekt. Fler detaljer ges i avsnitt 10.4.

Genetiska algoritmer är mycket kraftfulla och kan användas oberoende av nätverksarkitektur, men fordrar mycket datatid och är rätt oförutsägbara, varför man i första hand skall välja andra algoritmer. Som konkret strukturlika modeller av ett enskilt nervsystems funktionssätt är de vanligen helt orealistiska, däremot (lika naturligtvis) är de mycket användbara som delvis realistiska modeller av biologiska organismers evolution. Och för ”black box”-modellering av helt okända system passar de utmärkt. Eftersom de har mycket litet att göra med inlärning, i vanlig mening, i neurala nätverk kommer de att behandlas styvmoderligt i denna bok.

“Reinforcement learning” som problem- och algoritmfamilj

Vi har i avsnitt 2.2 talat om operant eller instrumentell betingning, eller inlärning genom belöning och bestraffning. Gemensamt för all sådan inlärning är att den styrsignal som får organismen att modifiera sina beteenden *inte* omedelbart ger information om hur mycket, och på vilket sätt, det verkliga beteendet avviker från det önskade. Istället är det fråga om en *ospecifik* feedback på det verkliga beteendet i form av en binär (belöning/bestraffning) eller åtminstone endimensionell (grad av belöning) signal. Organismen rättar ändå sitt beteende så att det så småningom får en hög positiv förstärkningsnivå. Kan man modellera detta genom någon känd ANN-algoritm?

Det första alternativ som man kommer att tänka på är kanske övervakad inlärning med en felkorrigerande algoritm. Som felsignal får då differensen mellan maximal och verklig belöning fungera. En sådan felsignal ger dock betydligt mindre information till nätverket än vad som är fallet vid vanlig övervakad inlärning. Verklig operant inlärning innebär dessutom ofta ett långt tidsintervall mellan beteende och förstärkning. Detta, tillsammans med den ospecifika felsignalen, gör att problemet med ansvarsfördelning (se avsnitt 2.2) ofta blir oöverkomligt när man använder den nämnda, ”direkta” approachen till problem med operant inlärning.

Det finns nu andra lärande algoritmer på det enskilda systemets nivå, som är mycket bra på att optimera sluttillståndet i en tillståndssekvens med

hjälp av ospecifika förstärkningssignaler. Termen ”reinforcement learning” används i den matematiskt orienterade litteraturen inte sällan som ett samlingsnamn för dessa algoritmer.[110] Till dem hör exempelvis de så kallade tidsdifferensmetoderna (TD-algoritmerna). De kan integreras med neurala nätverks-algoritmer på olika sätt, men bör inte själva klassificeras som sådana.[111] I vilken utsträckning dessa algoritmer kan belysa operant inlärning hos människor och andra djur – det vill säga det som psykologerna kallar ”reinforcement learning” – är fortfarande en öppen fråga. Tyvärr kommer vi inte att säga mer om den i denna bok.

Lösningar som finns – och lösningar som nätverket hittar

Som avslutning på vår snabbgenomgång av olika nätverkstyper och inlärningsalgoritmer, och som förberedelse för den mer detaljerade redogörelsen för vad olika typer av neurala nätverk kan göra, ska vi göra en viktig distinktion som vi egentligen har förutsatt länge. Den har att göra med två olika innebörder av påståenden om att ett ANN ”kan göra” det och det:

1. Ett *existensbevis* för ett artificiellt neuralt nätverk är ett bevis för att nätverket i fråga *i princip* kan lösa, alternativt *inte* kan lösa, ett visst problem. “I princip kan lösa” betyder här att det *finns* värden för vikterna i nätverket som gör att nätverket utför den önskade uppgiften – t.ex. att separera en datamängd i två givna klasser. Med andra ord, om man “bara” snickrar till nätverkets vikter på rätt sätt så får man den önskade klassifikationen. Vi skall senare se att det finns många intressanta existens- och icke-existensresultat för neurala nätverk. Utan att gå in på detaljer kan det nämnas redan nu att lösningar av vissa relativt enkla men olinjära klassifikationsproblem (i en mening av “olinjär” som ska förklaras senare) *inte* existerar för enlagrade feedforwardnät med linjär aktivering eller tröskelaktivering, medan däremot flerlagrade icke-linjära nät *i princip* kan lösa *alla* konsistenta klassifikationsproblem.

Som framgått av det som sagts tidigare är idén att “för hand” snickra till vikterna i ett nätverk för att lösa en given uppgift oftast inte praktiskt tillämpbar. Det vore liktydigt med att explicit programmera modellen

[110] Se Sutton & Barto (1998).

[111] Ett fantasieggande exempel på användbarheten av kombinationen neurala nätverk och TD-algoritmen är det program för backgammon, TD-backgammon, som en forskare skapade genom att låta ett visst neuralt nätverk spela mot sig själv och lära sig av erfarenheten. Se Tesauro (2002).

med en regel som löser problemet, men en sådan regel har vi ju i allmänhet inte tillgång till. Med andra ord, ett bevis för att det i princip existerar en lösning av visst problem med ett ANN ger oss inte automatiskt kunskap om vilken denna lösning är. Och det är därför som idén om lärande system är så viktig: träna systemet genom exempel, och låt det försöka hitta lösningen själv!

Till denna ända har man konstruerat de olika inlärningsalgoritmerna. Givetvis är den *principiella* möjligheten till en lösning, i ett visst ANN, av ett problem en *nödvändig* förutsättning för att man genom att tillämpa en inlärningsalgoritm ska kunna *lära* detta ANN att lösa problemet. Existensbeviset är alltså grundläggande.

2. Men det är inte tillräckligt; algoritmen måste också kunna hitta lösningen i fråga, och det är inte alls självklart att den gör det. Just därför behöver man *konvergensbevis för inlärning*. Med ett ”konvergensbevis” menas här ett bevis för att en viss storhet i ett system under givna betingelser så småningom med säkerhet eller med viss sannolikhet når, eller åtminstone asymptotiskt närmar sig, ett stabilt värde. I teorin för neurala nätverk används dylika bevis främst i två sammanhang. För det första behöver man ibland visa att ett visst rekurrent (feedback-) nät med tiden når ett stabilt aktivitetsmönster, och alltså inte t.ex. för evigt oscillerar mellan ett antal olika mönster. Ett känt exempel på denna första typ av bevis kommer vi att stöta på i avsnittet om Hopfieldnätet (7.1). I detta bevis ingår som ett nyckelelement, att den s.k. “energin” hos nätverket så småningom når ett minimum. Detta implicerar i sin tur, att signalmönstret förblir stabilt.

Men konvergensbevis används också – och det är givetvis mer relevant just här – för att visa att det *fel* som ett nätverk gör, när man försöker lära det att lösa ett problem genom en viss inlärningsalgoritm, så småningom når ett stabilt och i viss mening minimalt värde. Kanske man till och med kan bevisa att felet alltid konvergerar mot värdet noll. I andra fall får man kanske nöja sig med att det med *viss sannolikhet* kommer att hitta en *ganska bra* lösning. Exempelvis kommer vi att se nedan, att enlagrade nätverk med stegfunktion och deltaregel (“enlagrade perceptroner”) med säkerhet hittar alla lösningar som existerar för dem. För de flesta av de (principiellt mycket kraftfullare) flerlagrade nätverken finns det inte fullt så starka konvergensresultat. För dem kan man tyvärr ofta bara säga, att inlärningsalgoritmerna *sannolikt* konvergerar mot lösningar som ligger någorlunda *nära* de teoretiskt optimala.

Vi är nu nästan mogna att diskutera ett antal mer specifika typer av ANN som kommit till stor användning, antingen som biologiska modeller eller som beräkningsinstrument, eller (i flera fall) som bådadera. Men först behöver vi ytterligare några matematiska instrument.

5. Sannolikheter, vektorer, ANN

5.1 Neurala nätverk och sannolikheter

Vi har nu flera gånger berört relationerna mellan ANN-teori och traditionell statistisk teori, så det är dags för en informell men kompakt sammanfattning av relevanta bitar av den senare. Den enda matematikkunskap som förutsätts hos läsaren är förtrogenhet med elementär mängdteoretisk symbolik.

Deskriptiv statistik och statistisk inferens

Statistik är en uppsättning teorier och tekniker för att beskriva och förklara data. Att extrahera statistiska mått från en datamängd, såsom medelvärde och standardavvikelse, tjänar för det första syftet att göra datamängden överskådlig och därmed förhoppningsvis intuitivt begriplig. Statistiska tekniker har dock också en *inferensteoretisk* motivering, dvs. de kan användas för att dra slutsatser om underliggande samband som kan ha givit upphov till de iakttagna data, och för att förutsäga nya data. Hur bra en viss statistisk teknik tjänar dessa syften är en fråga som behandlas i *teoretisk* eller *matematisk* statistik. Neurala nätverk används ibland främst deskriptivt, alltså för att göra data överskådliga och intuitivt fattbara (SOM är typexemplet). Men neurala nätverk används mycket ofta som inferensinstrument, och därför måste deras giltighet för detta ändamål granskas noga.

Intressant nog råder det ingen enighet inom den matematiska och statistiska forskargemenskapen om hur inferensteoretiska frågor i grunden ska lösas. Detta beror på en oenighet om vissa begreppsliga och kunskapsteoretiska frågor som ibland kallas ”statistikteoretiska”, men som lika gärna kan kallas sannolikhetsfilosofiska. De har nämligen att göra med naturen hos begreppet *sannolikhet* och hur man skall applicera detta begrepp på de situationer där man har *empirisk evidens* för eller emot en viss hypotes. Det är därför nödvändigt att bekanta sig med sannolikhetsfiloso-

fiska frågor, om man vill förstå den inferensteoretiska problematiken på djupet. Nu följer en kort diskussion av denna problematik, sammanvävd med en mycket kondenserad och informell framställning av sannolikhetsteorins matematiska elementa.

Grundläggande begrepp i sannolikhetsteori

Grundläggande i sannolikhetsteorin är begreppen *försök, utfall, stokastisk variabel, utfallsrum* och *händelse*. Med ett *försök* menar man vilken situation som helst där någonting kan utfalla på olika sätt: t.ex. ett kast med en tärning, en fotbollsmatch, eller en väderförutsägelse. *Utfallen* är de olika, specifika möjligheterna till resultat som föreligger. Försöket ett kast med tärning har typiskt de sex utfallen etta, tvåa, ..., sexa. Man uttrycker också detta som att den *stokastiska variabeln* antal prickar kan anta de sex olika värdena 1, 2, 3, 4, 5 eller 6. Tillsammans bildar dessa möjliga värden *utfallsrummet,* som i detta fall kan symboliseras som {1, 2, 3, 4, 5, 6}. Här står varje siffra för ett möjligt värde hos den stokastiska variabeln och klammern för mängden av alla dessa värden.

En *händelse* är en mängd av utfall, i exemplet t.ex. etta eller sexa eller jämnt antal prickar; i mängdteorins symbolik {1, 6} respektive {2, 4, 6}. Observera att händelsen etta eller sexa kan ses som den mängdteoretiska unionen av händelserna etta och sexa. Om vi symboliserar den stokastiska variabeln antal prickar med ”X” kan vi alltså (något informellt) säga att följande beskrivningar av en viss händelse är ekvivalenta:

[(X = 1) *eller* (X = 6)]
{1, 6}
{1} ∪ {6}

Vi blandar fortsättningsvis beskrivningar i termer av satslogiska konnektiver (”A eller B”, ”A och B” etc.) med strikt mängdteoretiska sådana (”A ∪ B”, ”A ∩ B” etc.) på ett fritt sätt, som innebär att ”A” ibland står för en egenskap och ibland för motsvarande mängd. Detta är formellt inte oklanderligt, men kan knappast leda till några missförstånd här.

Vad är sannolikheter?

Vad menas då med *sannolikheten* för en händelse? Frågan avser inte *vilken* sannolikhet en viss händelse, t.ex. regn i morgon, har utan vad det *in-*

nebär att t.ex. säga att det är si eller så sannolikt att det blir regn i morgon. Man kan svara på frågan på ett helt formellt och matematiskt korrekt sätt: det är ett tilldelande av tal till ett antal möjliga händelser på ett sätt som uppfyller sannoliketsteorins axiom (se nedan). Detta hjälper oss dock inte att förstå vad sannolikhet är i empiriska *användningar* av begreppet. För sådana användningar har några huvudtyper av förslag framställts, varav vi ska nöja oss med att titta på fyra.

Den så kallade *klassiska* sannolikhetsuppfattningen är tillämpbar på situationer där alla tänkbara utfall är *likvärdiga*, dvs. inte uppvisar några relevanta skillnader. Sannolikheten för en viss händelse A definieras då som antalet *gynnsamma* utfall (dvs. sådana utfall som ingår i A) dividerat med det totala antalet utfall. Exempel: ”trea eller femma” vid kast med tärning innehåller två av sex likvärdiga utfall, alltså har denna händelse sannolikheten 2/6 = 1/3. Ett stort problem med den klassiska sannolikhetsuppfattningen är givetvis att inte alla utfallsrum består av likvärdiga utfall. Ta t.ex. sannolikheten för att en viss tändsticksask skall ställa sig på kant när man kastar den på ett visst sätt. Här finns ingen rimlig utgångspunkt för att tillämpa den klassiska synen.

Många tar för givet att sannolikheter istället ska definieras som *frekvenser i långa loppet*. Kastar man en tärning väldigt många gånger, resonerar företrädare för denna teori, kommer frekvensen av etta i långa loppet att närma sig 1/6, och på samma sätt kommer frekvensen av ”på kanten” att närma sig ett visst värde när man kastar en viss tändsticksask väldigt många gånger. Det senare förhållandet gör, säger företrädarna, att frekvenstolkningen av sannolikhet har ett större applikationsområde än den klassiska uppfattningen.[112]

Ett grundläggande problem för frekvenstolkningen är att den inte alls kan tillämpas på försök (situationer) som inte *kan* upprepas. Skall t.ex. sannolikheten för atomkrig 1960 definieras som det tal som frekvensen av år med atomkrig skulle närma sig, om 1960 upprepades ett mycket stort antal gånger? Allra svårast för frekvensdefinitionen är det att man vill tala om sannolikheter för vetenskapliga teorier och hypoteser. Påståendet ”Det är mer sannolikt än inte att relativitetsteorin är sann” kan näppeligen

[112] En modern variant av frekvenstolkningen, med filosofen Karl Popper som upphovsman, ser sannolikheter som *benägenheter* hos objekt eller situationer. Vi tar inte upp denna tolkning separat, men kommer ibland att formulera frekvensteorin med hjälp av begreppet benägenhet.

preciseras i frekvenstermer. Inte heller ”Det är osannolikt att dinosaurierna dog ut av en virussjukdom”. För att inte tala om ”I går var det sannolikt att det skulle bli regn i dag, men nu vet vi att det inte blev så.”

De sistnämnda exemplen gör det förhoppningsvis tydligt, att sannolikhetsbegreppet ofta används som ett mått hur säker resp. osäker en viss åsikt är, alltså på vår *grad av kunskap.* Man talar här om *epistemiska* (kunskapsteoretiska) sannolikheter, och vi har fått en tredje, epistemisk tolkning av begreppet sannolikhet.

Med lite eftertanke inser man att de epistemiska sannolikheterna ofta skiljer sig från de frekvensteoretiska i de fall då frekvenstolkningen *är* tillämpbar. Om man har alla goda skäl i världen att tro att det mynt man har i handen är en vanlig enkrona, är den epistemiska sannolikheten för krona i nästa kast = ½. Detta utesluter dock inte, att myntet kan råka tillhöra en falskspelare och därför ha en verklig, inneboende tendens att i långa loppet ge krona i 90% av kasten. När vi säger att myntet kan ha en annan sannolikhet för krona än den mest sannolika, kan det vara rimligt att använda frekvenstolkningen (eller den relaterade ”benägenhetstolkningen”, se not 116) för den i myntet ”inneboende” sannolikheten. Men för inferensteoretiska diskussioner, alltså frågor om hur hypoteser får stöd av data och hur vi därigenom uppnår grundade uppfattningar om världen, är de epistemiska användningarna uppenbarligen centrala.

Tyvärr ger den sistnämnda iakttagelsen inget bra svar på frågan om vad en epistemisk sannolikhet egentligen är. Den klassiska uppfattningen *kan* betraktas som ett partiellt svar på denna fråga. Ett annat förslag har presterats av den *subjektivistiska* skolan, som ibland – missvisande, som vi ska se – också kallas den ”bayesianska”. Enligt detta förslag, som alltså är den fjärde tolkning av sannolikhetsbegreppet som presenteras här, är epistemiska sannolikheter inget annat än en persons *grad av tro* på en viss hypotes. Denna grad av tro definieras i sin tur (av subjektivisten) i termer av vilka *odds* personen skulle gå med på i ett spel, där utgången definieras av om hypotesen är sann eller ej.

Nackdelen med den subjektivistiska teorin är att den har svårt att skilja mellan *tro* och *berättigad tro*, eller annorlunda uttryckt, mellan *grad av tro* och *grad av kunskap*. Man kan enligt subjektivismen egentligen inte diskutera hur sannolik en händelse är, bara vad som följer logiskt-matematiskt ur det ena eller andra antagandet om sannolikheter. Om det t.ex. är mer sannolikt än inte (för författaren till denna bok, när boken skrivs)

att Filosofiska Institutionen i Göteborg låg på Dicksonsgatan år 2005, är i princip bara en fråga om graden av hans tro på detta påstående. Om denna grad av tro skiljer sig åt mellan honom och hans motpart i en diskussion så finns det inget att göra, utom att försöka hitta inkonsistenser i motpartens system av sannolikhetstilldelningar till olika händelser. Subjektivisten hotar med andra ord att hamna i en *relativism*. – Vi måste lämna detta problem tills vidare, men återkommer till det snart.

I den närmast följande framställningen ska vi inte förutsätta någon av de nämnda tolkningarna av sannolikhetsbegreppet, utan ta fasta på att sannolikhetsteorin är tillämpbar på konkreta *proportioner* eller *andelar*. Om man ser en händelses sannolikhet som andelen eller proportionen av något i något annat – andelen socker i en viss blandning av socker och salt, eller proportionen av rödhåriga i hela befolkningen – kan man lätt förstå varför sannolikhetsteorins axiom och teorem ser ut som de gör. Vi kommer därför ofta att hänvisa till andelar och proportioner både för att ge intuitiva förklaringar av olika grundbegrepp och för att göra olika axiom och teorem i sannolikhetsläran trovärdiga. Känsliga matematiker bland läsarna ombedes inta något lugnande preparat.

Sannolikhetsteorins axiom

Beteckna således andelen av de individer i en population (också kallad ”universum”) som har egenskapen A (tillhör mängden A) med P(A), och egenskapen att överhuvudtaget tillhöra populationen med U (för ”universell egenskap”). Andelen P(A) definieras (givetvis) som *antalet A* i hela populationen dividerat med *hela antalet individer*. Då gäller uppenbarligen följande räknelagar:

(5.1.1) $0 \leq P(A) \leq 1$

(5.1.2) $P(U) = 1$

(5.1.3) $A \cap B = \varnothing \rightarrow P(A \cup B) = P(A) + P(B)$

Den enda av dessa tre lagar som inte är helt trivial är den tredje. Den innebär att om det *inte* finns någon individ som är *både* A och B, så är andelen individer som är *antingen* A eller B lika med *summan* av andelarna av A respektive av B. Med lite eftertanke (tänk till exempel på rödhåriga och svarthåriga i befolkningen) inses dock, att också detta är självklart sant om alla proportioner eller andelar.

De tre lagarna 5.1.1–5.1.3 utgör ett tillräckligt axiomsystem för sannolikhetsteorin. Med andra ord, ur dessa tre lagar (och lämpliga definitioner av de begrepp som behövs i kontinuerlig sannolikhetsräkning, se nedan) kan man härleda alla sannolikhetslärans teorem.

Några basala teorem

Villkoret i 5.1.3 att mängderna är disjunkta är alldeles nödvändigt. Om det finns några individer som är både A och B, så kommer nämligen en summation av andelen av A med andelen av B att innebära att just dessa individer räknas två gånger. I det allmänna fallet måste man med andra ord subtrahera andelen av individer som är både A och B:

(5.1.4) $\quad P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Man inser omedelbart att 5.1.3 är ett specialfall av 5.1.4.

Nästa fråga är förstås om andelen för ett snitt, eller $P(A \cap B)$, på något enkelt och allmänt sätt kan beräknas utifrån de ingående mängdernas andelar. Svaret är nej, och detta är en annan hörnsten i sannolikhetsteorin. Det bästa sättet att inse att man *inte* kan hitta något sådant allmänt sätt att räkna ut $P(A \cap B)$ är att genom något exempel övertyga sig om att antalet element i snittmängden $A \cap B$ inte är entydigt bestämt av antalen element i A respektive B. Av detta följer i sin tur, via formeln 5.1.4, att inte heller $P(A \cup B)$ är entydigt bestämd av P(A) och P(B) (utom i de specialfall som beskrivs av 5.1.3 och 5.1.5, se nedan!).

Oberoende

Man definierar ofta begreppet *(statistiskt) oberoende* genom följande formel:

(5.1.5) $\quad$ *A och B är oberoende :* $\quad P(A \cap B) = P(A) \cdot P(B)$

Räkneregeln i 5.1.5 gäller alltså *inte* för godtyckliga A och B. Vi ska strax visa att 5.1.5 är matematiskt likvärdig med en annan, kanske mer intuitivt lättbegriplig definition av oberoende.

Relativa sannolikheter

Vi ska nu titta på andelar inom delmängder av populationen, eller *relativa* andelar. Betrakta uttrycket "de rödhårigas andel bland männen". Rödhåriga personer som *inte* är män kan förstås inte bidra till denna andel. Andelen *rödhåriga bland män* fås alltså inte som antalet av *alla* rödhåriga personer dividerat med antalet män, utan som antalet *rödhåriga män* dividerat med antalet män. Vi inför ett speciellt uttryck för denna relativa andel, nämligen $P(A \mid B)$, vilket utläses "andelen av A givet B".[113] Vi har just förklarat varför denna andel ska beräknas genom formeln

$$(5.1.6) \qquad \textit{Relativ sannolikhet :} \qquad P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

5.1.6 är den vanliga definitionen i sannolikhetsteori av begreppet *relativ sannolikhet.* Tänk, alltså, gärna på den relativa sannolikheten $P(A \mid B)$ som andelen individer med en egenskap A *bland* individer med egenskapen B, men tänk ännu hellre på den som sannolikheten för A, *givet* att B. "Andelen rödhåriga bland män" är ju samma sak som "andelen rödhåriga, givet att vi bara talar om män".

För att förstå de båda begreppen *oberoende* och *relativ sannolikhet* bättre ska vi nu relatera dem till varann. Antag att A och B är oberoende enligt 5.1.5. Då kan vi ersätta $P(A \cap B)$ i 5.1.6 med $P(A) \cdot P(B)$, och sedan förkorta bort P(B). Kvar blir uttrycket:

$$(5.1.7) \qquad P(A \mid B) = P(A)$$

vilket alltså gäller om och endast om A och B är oberoende. För att se att detta resultat stämmer bra med ett intuitivt begrepp *oberoende*, läs ut formeln 5.1.7 som följer: sannolikheten för A, givet att B, är lika med sannolikheten för A *utan* att något är givet om B. Med andra ord, att B faktiskt är fallet påverkar inte sannolikheten för A. Vad är detta, om inte oberoende mellan A och B?

Läsaren kan själv enkelt bevisa att 5.1.7 gäller om och endast om $P(B \mid A) = P(B)$.

---

[113] Förväxla inte symbolen $\mid$ med symbolen / för division, eller kvot.

Bayes sats

Vi kommer nu till en verkligt central punkt, nämligen förhållandet mellan de två storheterna $P(A \mid B)$ och $P(B \mid A)$. De två är nämligen inte nödvändigtvis lika. Detta följer direkt av definitionen 5.1.6:

(5.1.6a) $$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

(5.1.6b) $$P(B \mid A) = \frac{P(B \cap A)}{P(A)}$$

Täljarna i de båda högerleden har alltid samma värde, eftersom $A \cap B$ och $B \cap A$ är samma mängd. Nämnarna däremot har oftast olika värden. Dividera nu uttrycken (5.1.6a) och (5.1.6b) med varann. Resultatet blir:

(5.1.8) $$\frac{P(A \mid B)}{P(B \mid A)} = \frac{P(A)}{P(B)}$$

vilket också kan skrivas som

(5.1.9) *Bayes sats :* $$P(A \mid B) = P(B \mid A)\ \frac{P(A)}{P(B)}$$

Bayes sats har ett stort antal viktiga tillämpningar i såväl statistisk inferensteori som vetenskapsfilosofi. Vi ska återkomma till dem senare, men närmast gå vidare med ytterligare en räknelag.

Blandningsprincipen

Denna ytterst användbara princip säger följande. Om universum (hela populationen) *uttömmande* kan delas in i de *disjunkta* (åtskilda) delmängderna B_1, B_2,...,B_n, så kan varje sannolikhet $P(A)$ skrivas som

(5.1.10) $$\begin{aligned} P(A) = {} & P(A \mid B_1)\cdot P(B_1) + \\ & + P(A \mid B_2)\cdot P(B_2) + \ldots + P(A \mid B_n)\cdot P(B_n) \end{aligned}$$

Blandningssatsen säger med andra ord, att om de relativa sannolikheterna för en händelse under *alla* alternativ är givna, så får man den totala sannolikheten för händelsen genom att väga samman dessa relativa sanno-

likheter med alternativens respektive sannolikheter. Den har en lättfattlig modell i form av blandning av t.ex. vätskor. Antag att det i tre behållare B_i finns 2, 3 respektive 5 liter vätska; koncentrationerna $P(A \mid B_i)$ av alkohol A i de tre kärlen är 40%, 50% respektive 30%. Vilken styrka P(A) får blandningen om man häller ihop alltsammans? Det rätta sättet att räkna fram resultatet är att väga enligt blandningsprincipen:

$$P(A) = 0{,}4 \cdot 0{,}2 + 0{,}5 \cdot 0{,}3 + 0{,}3 \cdot 0{,}5 = 0{,}35 \tag{5.1.11}$$

Läsaren må själv testa slutsatsen empiriskt.

Blandningssatsen har en mycket viktig tillämpning i det som man kallar *statistiska blandningar av försök*. Antag att en falskspelare har tre mynt, som ger utfallet krona med sannolikheterna 1/3, 1/2 respektive 5/6. Nu tar han ett mynt slumpvis och kastar det. Vilken är sannolikheten för att utfallet ska bli krona? Jo, enligt blandningssatsen gäller:

$$P(\text{krona}) = \frac{1}{3}\cdot\frac{1}{3} + \frac{1}{2}\cdot\frac{1}{3} + \frac{5}{6}\cdot\frac{1}{3} = \frac{10}{18} = \frac{5}{9} \tag{5.1.12}$$

Blandningssatsen används också ofta i beräknandet av P(B) i nämnaren i högerledet av Bayes' sats. Ibland känner man nämligen både de relativa sannolikheterna $P(B \mid A_i)$ och sannolikheterna $P(A_i)$ för en uttömmande uppsättning alternativ A_i, och kan då beräkna P(B) med blandningssatsen. Mer om detta nedan.

Distributioner av sannolikheter

Om man vet vilka sannolikheterna är för alla alternativa utfall kan man beskriva dem som en *distribution* eller *fördelning* av sannolikhet över utfallsrummet. Termerna påminner oss om att hela den totala sannolikheten 1 måste vara fördelad på de olika möjliga alternativen. En distribution över ett diskret, ändligt utfallsrum är, matematiskt sett, en *funktion* p(x) som tillordnar en sannolikhet P(x) till varje utfall x. För att skilja mellan olika fördelningar indexerar man ibland funktionsnamnet med alfabetets stora bokstäver; $p_X(x)$ och $p_Y(x)$ står då för två olika fördelningar. Fortsättningsvis kommer vi, något oegentligt men mindre krångligt, i det diskreta fallet att använda stora P för *både* en distribution och de sannolikheter som den tilldelar olika händelser. Vilken fördelning det handlar om kommer att framgå av sammanhanget.

I många av de diskreta exempel som vi hittills talat om är utfallen ifråga inte beskrivna genom tal, och fördelningen ser inte ut som en matematisk funktion av den typ som man är van vid från den elementära matematiken. Distributionen av sannolikheter vid ett kast med ett vanligt mynt kan således enklast beskrivas så här:

(5.1.13) $\quad P(x)\text{: krona} \rightarrow \frac{1}{2}, \text{ klave} \rightarrow \frac{1}{2}$

I många sammanhang är emellertid utfallen karakteriserade genom ett tal, och då får sannolikhetsfördelningen en mer välkänd matematisk form. Ett exempel är följande. På ett nöjesfält är ett enkelt lyckohjuls omkrets indelad i fyra segment numrerade 1, 2, 3, 4 och med längderna 1, 2, 3 respektive 4 dm. Eftersom sannolikheten för att hjulet ska stanna inom ett visst segment är proportionell mot segmentets längd, och den sammanlagda omkretsen är 10 dm, kan sannolikheten P(x) för att det ska stanna inom segment nummer x beskrivas genom formeln:

(5.1.14) $\quad P(x) = \frac{x}{10}$

En ännu enklare diskret fördelning är den konstanta eller *likformiga* distribution av sannolikheter över utfallen x = 1, 2, ..., 6 som vi får när vi kastar en välbalanserad tärning, nämligen: $P(x) \equiv 1/6$.

En mycket välkänd distribution av sannolikheter över diskreta, ändliga utfallsrum är slutligen *binomialfördelningen*. Den säger, åskådligt uttryckt, att sannolikheten att få exakt x krona vid n kast med ett välbalanserat mynt följer formeln:

(5.1.15) $\quad \textit{Binomialfördelning } (p = \frac{1}{2})\text{:} \quad P(x) = \frac{n!}{x!(n-x)!} \cdot \left(\frac{1}{2}\right)^x$

där n!, som utläses n-fakultet, är $1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$.

Väntevärde

I de fall då utfallsrummet för en sannolikhetsdistribution kan beskrivas med tal, kan man också definiera begreppet *väntevärde* för distributionen. Väntevärdet E(x) definieras som ett slags medelvärde av utfallen,

där dessa sammanvägts med hjälp av sina sannolikheter. När utfallsrummet U är diskret har vi:

(5.1.16) *Väntevärde :* $E(x) = \sum_{U} x \cdot P(x)$

Väntevärdet vid kast med en vanlig tärning är uppenbarligen 3,5 – ett resultat som man sällan får... För en definition av väntevärde i det kontinuerliga fallet, se nedan.

Apriori- och aposteriorifördelningar enligt Bayes sats

Betrakta falskspelaren och hans tre mynt (se stycket ovan om blandningsprincipen). Han tar ett av dem slumpvis – dvs. vardera med sannolikheten 1/3 – och kastar det en gång, varvid han får en krona. Låt oss döpa detta utfall till ”e”. Rimligtvis bör det nu inte längre vara riktigt lika sannolikt som förut att det är fråga om det mynt som sällan ger krona, men istället lite mer sannolikt att det är det mynt som ofta ger krona. Fördelningen av sannolikheter över de tre alternativen borde med andra ord ha ändrats.

Vi kan faktiskt med hjälp av Bayes sats exakt bestämma den nya fördelningen $P(H_1 \mid e)$ – *aposteriorifördelningen* (ibland kallad fördelningen *ex post*), till skillnad från den tidigare *apriorifördelningen* $P(H_1)$ (eller fördelningen *ex ante*). Sannolikheten för att det ska vara mynt nummer 1 – låt oss kalla detta för hypotesen H_1 – måste nu nämligen vara:

(5.1.17) $$P(H_1 \mid e) = P(e \mid H_1) \cdot \frac{P(H_1)}{P(e)}$$

Av de ingående storheterna är $P(e/H_1)$ och $P(H_1)$ enkla att ange – de är båda 1/3 – och P(e) har vi redan räknat ut med blandningssatsen, se (5.1.12) ovan. Vi får:

(5.1.18) $$P(H_1 \mid e) = \frac{1}{3} \cdot \frac{1/3}{5/9} = \frac{18}{90} = 0{,}2$$

På samma sätt kan $P(H_2 \mid e)$ beräknas till 0,3 medan $P(H_3 \mid e)$ blir 0,5.

Det är lätt att inse, att om vi kastar myntet fem gånger och får fem krona så kommer fördelningen att förskjutas ännu mer till förmån för hypotesen

H_3. Det är inte heller svårt att tycka att detta är ett intuitivt rimligt resultat. Många av dagens statistikteoretiker menar faktiskt att hypotesprövning i vetenskap i princip går till just på detta sätt (se vidare nedan).

Kontinuerlig sannolikhetsräkning

Ett hjul med 6 decimeters omkrets är upphängt på en axel med mycket liten friktion. Kalla den punkt på hjulets omkrets som vid ett visst tillfälle befinner sig överst för ”utgångspunkten”. Ge hjulet en rejäl rotation och notera vilken punkt på hjulet som är överst när det slutligen stannar. Mät avståndet på hjulets omkrets medurs från utgångspunkten till denna punkt. Det är frestande att säga att sannolikheten för att avståndet skall vara x decimeter är lika hög för *varje* x mellan 0 och 6. Det vill säga, denna sannolikhet borde följa en likformig fördelning $P(x) = k$, för $0 \leq x < 6$. Hjulet kan emellertid stanna var som helst, det vill säga det finns i princip *oändligt många utfall*. Men då blir väl den sammanlagda sannolikheten för alla alternativen oändligt stor – såvida man inte sätter k i den likformiga distributionen till 0! Hur ska man behandla sådana här situationer, teoretiskt och matematiskt?

Jo, notera först att sannolikheten för att den översta punkten ska hamna *inom ett visst segment* av hjulets omkrets, t.ex. en viss halva av omkretsen, är större än noll. Betrakta ett godtyckligt segment, och dividera den sannolikhet som gäller för segmentet med segmentets längd. Som resultat får vi en storhet som vi kan kalla *segmentets genomsnittliga sannolikhetstäthet*. Vi gör om proceduren många gånger med mindre och mindre segment, och får när vi kommit till ett oändligt litet segment det som i sannolikhetsteorin kallas *sannolikhetstäthet i en punkt.*

Ett ekvivalent sätt att förstå denna sublima storhet är följande. Ta först ett vanligt tärningskast som exempel och fråga: hur fort *växer* sannolikheten för att få högst x prickar när x växer från 1 till 6? Jo, den växer med hastigheten 1/6 per prick. Återgå sedan till hjulexemplet. Tänk dig ett visst segment av hjulets omkrets, och låt det gradvis utvidgas åt ena hållet. Sannolikheten för att den punkt som är överst när hjulet stannat skall ligga *någonstans inom hela segmentet* ökar då förstås hela tiden. Den *hastighet* med vilken denna sannolikhet ökar när vi nått en viss punkt på omkretsen är ingenting annat än sannolikhetstätheten i denna punkt.

Detta resonemang leder på ett naturligt sätt fram till den formella definitionen av sannolikhetstäthet i termer av *derivatan av en total, eller acku-*

mulerad, sannolikhet (”sannolikhetsmassa”). Antag att en storhet X som är associerad med en viss sannolikhetsdistribution bara kan anta värden mellan a och b, och att sannolikheten för att den ska anta ett värde *mellan värdet a och värdet x* är en viss funktion F(x) av värdet x. Formellt:

(5.1.19) $F(x) = P(a < X \leq x)$

(I hjulexemplet är X = avståndet i decimeter, a är 0, och b är 6.) Det följer omedelbart att F(b) = 1 eftersom alla värden av X måste ligga mellan a och b. Tillväxthastigheten av F i punkten x, alltså *derivatan av F med avseende på x*, är återigen lika med sannolikhetstätheten i x. Den kallas också ”frekvensfunktionen” och brukar i svenska framställningar ofta betecknas med lilla ”f”. I engelskspråkig litteratur talar man oftast om ”density” och betecknar sådana densiteter med lilla ”p”. Vi följer den senare konventionen här. Liksom många andra författare skiljer vi ibland mellan olika frekvensfunktioner genom att använda olika argument för ”p”. Således kan p(x) vara en annan frekvensfunktion än p(y), när dessa uttryck förekommer i samma formel (se t.ex. 5.1.22 nedan).[114]

Observera att såväl en frekvensfunktion som en diskret fördelning över ett ändligt utfallsrum uttrycker hur fort den ackumulerade sannolikheten ökar vid ett visst utfall –skillnaden är bara att ökningen i det ena fallet går i diskreta steg (som *själva* kan beskrivas i termer av sannolikheter), medan den i det andra är kontinuerlig. Därför kallar vi också frekvensfunktioner för *kontinuerliga sannolikhetsfördelningar*.

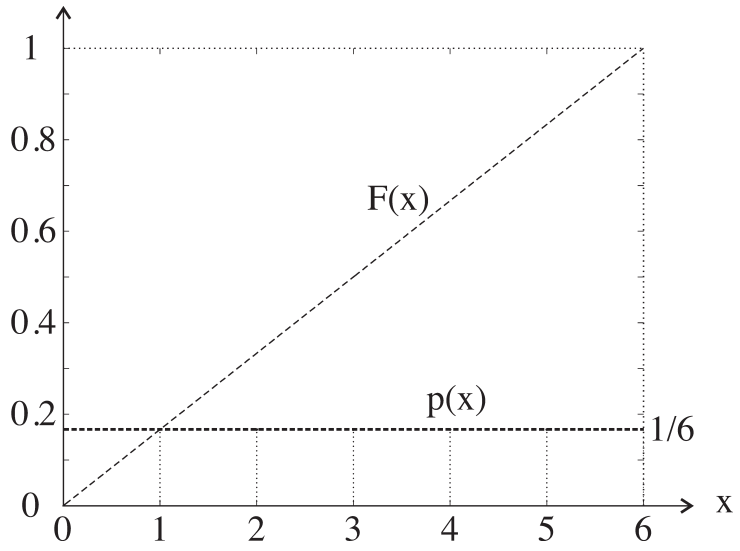
Väntevärdet för en kontinuerlig sannolikhetsfördelning defineras, i full analogi med det diskreta fallet, som

(5.1.20) *Väntevärde (kontinuerligt utfallsrum):* $E(x) = \int_U x \cdot p(x)\, dx$

Med dessa redskap i vår hand kan vi börja visualisera några kontinuerliga fördelningar. Den frekvensfunktion som beskriver sannolikheterna för att olika punkter på vårt hjul ska hamna överst kommer faktiskt att vara konstant, och om vi för exemplet skull fortfarande tänker oss att hjulet har en omkrets av 6 decimeter så måste det konstanta värdet vara 1/6 (enhet: sannolikhet per decimeter). Endast på detta sätt kan den totala sanno-

[114] Ett formellt mer korrekt beteckningssätt är att använda olika funktionsnamn. En kompromiss är (jämför ovan) att indexera funktionsnamnen och använda beteckningarna $p_X(x)$ och $p_Y(x)$ när det är fråga om två olika frekvensfunktioner.

likheten bli lika med 1. Den *ackumulerade* sannolikheten F(x) fram till punkten x är integralen av p(x) från 0 till x och kommer att följa en lutande rät linje. Detta åskådliggörs i figur 25.



Figur 25. En likformig, kontinuerlig sannolikhetsfördelning p(x) och den ackumulerade sannolikheten F(x). Förklaring: Se text.

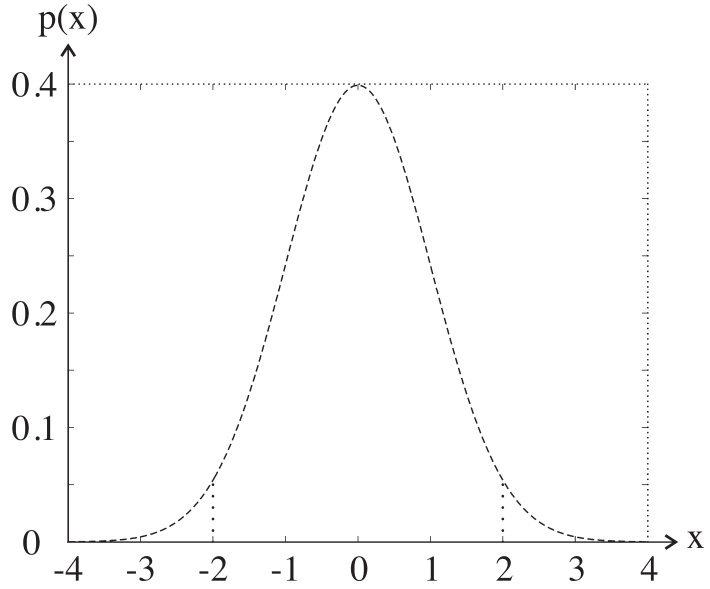
Vi har nu sett ett exempel på en *likformig* kontinuerlig sannolikhetsfördelning. *Icke* likformiga sådana fördelningar blir ganska snart matematiskt besvärliga att hantera. Vi ska därför inte tala så mycket om dem.

Normalfördelningar

En viktig och välkänd familj av kontinuerliga, icke likformiga fördelningar av sannolikhet är dock de *gaussiska* fördelningarna, eller normalfördelningarna, som var och en kan entydigt karakteriseras genom sitt *väntevärde* μ och sin *standardavvikelse* σ. Här är frekvensfunktionen för normalfördelningen:

(5.1.21) *Normalfördelning :* $$p(x) = \frac{e^{\frac{(x-m)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$$

Figur 26 visar dess graf när $\mu = 1$ och $\sigma = 0$.



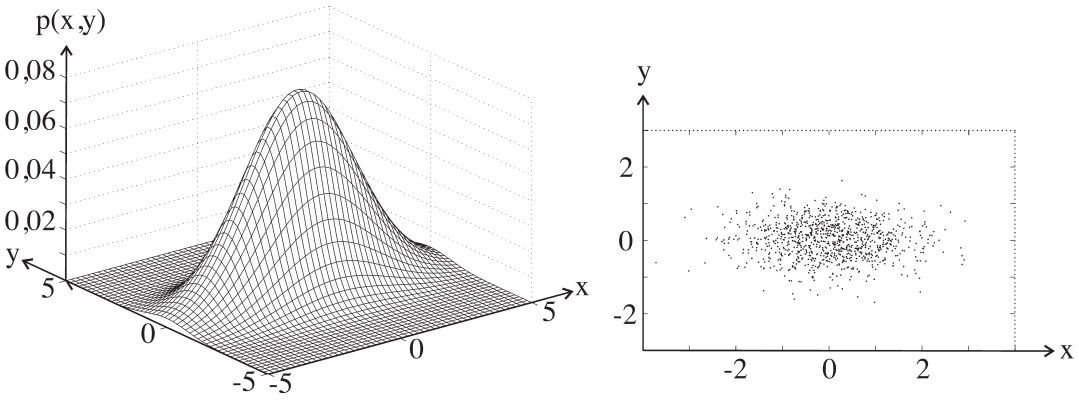
Figur 26. Normalfördelning med medelvärde 0 och standardavvikelse 1. Vertikala prickade linjer: 2 standardavvikelser på vardera sidan om medelvärdet.

Kurvan skall, i enlighet med hur frekvensfunktionen är definierad, uttydas som så: den totala sannolikheten för att en händelse X ska inträffa sådan att X ligger *till vänster* om en viss punkt x på den horisontella axeln (dvs. i intervallet mellan $-\infty$ och x) *ökar* snabbast där kurvan har sin topp, för att sedan öka långsammare och långsammare. Dock är inga värden på X helt uteslutna. Utfallsrummet är med andra ord oändligt i detta fall, vilket det inte kan vara vid en likformig fördelning.

Normalfördelningar spelar en central roll i statistisk inferensteori, dels eftersom många slumpprocesser kan visas leda till ungefär normalfördelade värden, dels eftersom normalfördelningar (trots den kanske avskräckande definitionen 5.1.21) är relativt lätta att behandla matematiskt.

Vi ska också nämna *flerdimensionella* normalfördelningar. Sådana fördelningar har ett enda, flerdimensionellt väntevärde men behöver inte ha samma standardavvikelse i de olika riktningarna. En tvådimensionell normalfördelning med olika standardavvikelser i de två olika riktningarna ser i ett tredimensionellt diagram ut som en puckel som är mer tillplattad i en riktning än i en annan. Följande figur visar en sådan tvådimensionell normalfördelning och ett stickprov från den, dvs. ett ändligt antal punkter

som placerats inom varje givet litet område med en sannolikhet som motsvarar sannolikhetsmassan inom detta område.



Figur 27. En tvådimensionell normalfördelning och ett stickprov från den. Variablerna x och y är oberoende, och båda har medelvärdet 0. Standarddeviationen i x-led är 1, och i y-led är den 0,5.

Bayes sats för kontinuerliga utfallsrum

Vi behöver bara ett sannolikhetsteoretiskt teorem till för den följande framställningen, nämligen den kontinuerliga varianten av Bayes sats. Antag att vi har en fördelning p(x) av en storhet X och en annan fördelning p(y) av storheten Y. Vi inför nu begreppet *relativ* (eller betingad) *fördelning,* symbol: p(x | y). Detta uttryck kan utläsas som fördelningen av X givet det specifika utfallet y av Y, och vi har

(5.1.22) *Bayes sats (kontinuerliga utfall) :* $$p(x \mid y) = p(y \mid x) \cdot \frac{p(x)}{p(y)}$$

där man i analogi med blandningssatsen för det diskreta fallet i princip kan beräkna p(y) som integralen (över hela utfallsrummet för X):

(5.1.23) $$p(y) = \int_U p(y \mid x) \cdot p(x)\, dx$$

Från kunskap om p(x), *aprioriförde*lningen av X, tillsammans med en observation av Y och kunskap om hur Y beror av X, kan man alltså med hjälp av Bayes kontinuerliga sats dra slutsatser om hur fördelningen av X ser ut i ljuset av denna observation – dvs. om *aposteriorifördelningen* av

X. Detta något abstrakta resonemangkommer att få mer kött på benen senare, då vi talar om bayesiansk inferensteori för det kontinuerliga fallet.

Konfirmation av hypoteser med Bayes sats?

Det är inte ovanligt att man formaliserar begreppet *evidens* som *ökad relativ sannolikhet*. Enligt detta tänkesätt får en hypotes H positivt stöd av evidensen (observationen) e om och endast om

(5.1.24) *Konfirmation :* $P(H \mid e \cap k) > P(H \mid k)$

där k är den bakgrundkunskap vi har *innan* vi observerar att e är fallet. (Ofta, t.ex. nedan, låter man k vara underförstådd.) Man säger också att e *konfirmerar* H om denna relation råder. Om sannolikheten istället sjunker när e tillkommer talar man om *diskonfirmation*. Frågan man nu måste ställa sig är: kan man med hjälp av sannolikhetsteorins lagar *beräkna* den grad av stöd som den ena eller andra hypotesen får av den ena eller den andra iakttagelsen? Det vill säga, kan man härleda regler för vetenskaplig verksamhet ur sannolikhetsteorin?

Vi har föreslagit ovan att så är fallet – nämligen med hjälp av Bayes teorem. Bayes sats ger oss ju $P(H \mid e)$ när vi känner $P(e \mid H)$, och $P(e \mid H)$ tycks inte sällan vara möjlig att beräkna. Vi gjorde det i exemplet med falskspelaren. På analogt sätt ger, under vissa förutsättningar, hypotesen

(5.1.25) H: placebo P botar sjukdomen S lika ofta som medicinen M

bestämda och beräkningsbara sannolikheter för de olika möjliga utfallen av ett experiment där 100 patienter får medicin och 100 andra får placebo. Alltså borde vi väl kunna räkna oss baklänges med Bayes sats till hur sannolikt det är, givet det utfall vi faktiskt får (e), att placebo P duger lika bra som medicinen M.

Går det att göra så? Om detta tvista de lärde. En mäktig problemfamilj indikeras av termen *apriorisannolikheternas problem*. Man måste ju också känna sannolikheten P(H) för att kunna beräkna $P(H \mid e)$ med Bayes' teorem. För att beräkna P(e) i Bayes sats med hjälp av blandningssatsen (se ovan, 5.1.23) behöver man t.o.m. känna till fördelningen {P(H')} över alla tänkbara alternativhypoteser H'. Varifrån kommer kunskapen om alla dessa sannolikheter? I exemplet med falskspelaren var sannolikheterna givna, men hur är det i exemplet med medicin vs. placebo, och andra

liknande fall? Ingen enighet råder bland statistikteoretiker eller filosofer om hur dessa problem skall lösas.

Statistisk inferens enligt traditionell teori

Det finns två huvudsätt att tackla apriorisannolikheternas problem. I traditionell, icke-bayesiansk statistikteori löser man problemet genom att undvika det; man avstår helt sonika från att beräkna $P(H \mid e)$. Man nöjer sig med att beräkna $P(e \mid H)$, eventuellt för olika hypoteser H_i, och fattar beslut om vilken hypotes som skall accepteras utgående från enbart dessa *sannolikheter för evidensen, givet hypoteserna* – de så kallade *troligheterna*, på engelska *likelihoods*.[115]

En central metodologi i traditionell statistik är således den så kallade *ML-metoden* (ML för *Maximum Likelihood*). Den går i det diskreta fallet ut på att man först beräknar troligheterna $P(e \mid H_i)$ för en uppsättning alternativhypoteser H_i. Sedan väljer man den alternativhypotes som har den *högsta* troligheten – det vill säga den hypotes som ger de givna data högst sannolikhet, förutsatt att den är sann. Utvidgningen till kontinuerliga sannolikhetsdistributioner är enkel. Alternativhypoteserna karakteriseras genom en eller flera parametrar, som var och en kan anta olika värden på ett kontinuum. ML-metoden innebär återigen att man väljer de parametervärden som ger de iakttagna data högst sannolikhet (alternativt: högst densitet, nämligen om utfallsrummet för data också är kontinuerligt).

En vanlig och för många välkänd tillämpning är skattning av väntevärde och standardavvikelse för en antagen normalfördelning. *Om* vi antar att vi observerar data från en normalfördelning med väntevärdet μ och standardavvikelsen σ, *så* kan sannolikheterna (eller densiteterna) för olika tänkbara utfall beräknas exakt. ML-skattningen väljer helt enkelt ut den normalfördelning vars tänkta parametrar μ och σ ger den *högsta* sannolikheten (densiteten) för de faktiskt iakttagna data. Det μ som åstadkommer denna maximala trolighet råkar sammanfalla med medelvärdet hos de iakttagna data, och det ”bästa” σ kan beräknas på ett nästan lika enkelt sätt, men detta läser man bättre om i läroböcker i statistik.

---

[115] Notera att troligheten förvisso skall beräknas som $P(e \mid H)$, dvs som en (relativ) *sannolikhet för evidensen*, men ändå i den statistiska litteraturen betecknas som ”troligheten *hos hypotesen*”! Detta har gissningsvis fått många avnämare av denna litteratur att tro att man faktiskt får ”vända på steken” och dra slutsatser om $P(H \mid e)$.

En ML-skattning behöver inte vara *väntevärdesriktig* (engelska: unbiased). En väntevärdesriktig skattning är en procedur som om den genomföres på ett allt större stickprov så småningom kommer godtyckligt nära den verkliga parametern – förutsatt att modellen som skattningen utgår från är riktigt vald. ML-skattningen av μ för en normalfördelning är väntevärdesriktig, men inte den av σ .

Flera algoritmer för artificiella neurala nätverk kan motiveras med maximum-likelihood-resonemang. Ett ännu otränat ANN som ska användas för att klassificera data i närvaro av brus uppfattas som en sannolikhetsprocess som kan generera olika klassifikationer. Denna process har ett antal okända parametrar, nämligen vikterna. De inlärningsregler som används går ut på att hitta de vikter som, inom ramen för de begränsningar som nätverkets struktur och brusets karaktär ger, gör att nätverket ger högsta möjliga sannolikhet åt evidensen, alltså åt de önskade klassifikationerna av träningsdata. Se vidare avsnitt 6.2 och 9.3.

Observera att man från en parameterskattning som följer ML-principen inte får dra några slutsatser om hur sannolikt det är att den ena eller den andra skattningen är riktig – framförallt får man inte dra slutsatsen att ML-skattningen skulle ange det *mest sannolika* värdet för den okända parametern! Problemet är *inte* att de alternativa hypoteserna om vilka parametrar den okända fördelningen har som regel bildar ett kontinuum, och att den enskilda hypotesen då knappast kan ha en sannolikhet skild från 0. Detta slags problem kan man ju enkelt tackla genom att tala om densiteter och kontinuerliga fördelningar. Problemet är att fördelningen *ex post* (i ljuset av evidensen) inte kan beräknas utan att man har tillgång till fördelningen *ex ante* (före evidensen), och fördelningar *ex ante* får man inte blanda in enligt den skola som vi talar om nu.

En annan välkänd, traditionell beslutsregel säger att man skall förkasta en hypotes H_0 i ljuset av evidensen e, om troligheten $P(e \mid H_0)$ är mindre än ett visst värde (signifikansnivån), t.ex. 0,05. Om således det faktiska utfallet av vår jämförelse mellan medicin och placebo, kalla det e, är *mindre* sannolikt än 0,05, *givet* att medicinen *inte* har bättre effekt än placebo, så ska man förkasta hypotesen att medicinen inte har bättre effekt – dvs. man accepterar tills vidare att medicinen *har* bättre effekt.[116]

[116] Vid beräkningen av $P(e \mid H_i)$ enligt denna metodologi avser man med ”e” inte det specifika utfallet av experimentet (t.ex. det exakta antalet som blev friska i försöks- respektive kontrollgruppen). Man beskriver evidensen på ett mer generellt sätt (som dessutom varierar beroende på om man väljer ett ”ensidigt” eller ett ”tvåsidigt” test).

Ett mycket vanligt missförstånd är att förkastande av en hypotes på signifikansnivån 0,05 betyder att *hypotesens* sannolikhet, givet data, skall bedömas som varande mindre än 0,05. Man får återigen inte glömma, att man enligt traditionell statistikteori inte får uttala sig om hypotesers sannolikhet, bara om hur sannolika data är, givet hypoteserna! En dansk läkare och vetenskapsteoretiker berättar att av 25 tillfrågade läkare, som alla gått statistikkurs och hade viss egen forskningserfarenhet, var det *ingen* som gav den rätta tolkningen åt signifikansvärdet...[117]

Varför ska man använda de icke-bayesianska beslutsreglerna?

Om man sålunda har insett att den icke-bayesianska traditionen inte ger rum för några omdömen om hypotesers sannolikhet, eller om hypotetiska parametrars sannolikhetsfördelning, är det naturligt att undra: varifrån kommer egentligen den traditionella statistikens beslutsregler, om *inte* från en bedömning att de genererar sannolika hypoteser och sannolikt korrekta skattningar? Mycken möda har lagts ner på argument som går ut på att användandet av dessa beslutsregler i det långa loppet måste generera fler korrekta hypoteser än vad andra regler skulle göra. Vi ska inte gå in på dessa – enligt min uppfattning inte särskilt lyckade – försök, utan istället titta närmare på det mer direkta angreppssättet, alltså det bayesianska.

Bayesiansk inferensteori

Bayesianen tar tjuren vid hornen och försöker explicit beräkna $P(H \mid e)$. Hennes stora problem är att hon för detta behöver apriorisannolikheten för H och (vanligen) för dess alternativhypoteser. I de enkla fall då det finns givna alternativhypoteser med bestämda apriorisannolikheter är det inte så svårt att räkna. Inte ens falskspelaren bjuder dock alltid på så goda betingelser. Du ser någon slå sex slag med en tärning och det blir 6 sexor. Ska du dra slutsatsen att tärningen är falsk? Ja, det beror i hög grad på hur du taxerar apriorisannolikheten för denna hypotes!

Samma problem uppstår givetvis om bayesianen vill gå vidare från en lyckad falsifiering av en hypotes H_0 enligt traditionell modell till att bedöma sannolikheten för att H_0 verkligen är falsk. Resultatet är helt

---

Detta är ett separat metodologiskt problem, som inte har någon betydelse för de filosofiska poänger vi gör här.

[117] Wulff (1981), s. 186.

avhängigt av vilka apriorisannolikheter som tilldelas H_0 och dess alternativhypoteser. Är det till exempel lika sannolikt, innan försöket görs, att medicinen har *samma* reella effekt som placebo (H_0), som att den har en *större* effekt (H_1)? Eller borde vi tilldela H_1 en större apriorisannolikhet? En mindre?

Frågan väntar också om hörnet då bayesianen skall motivera sitt alternativ till ML-skattning av en okänd parameter. ML-skattning går ju ut på att välja det parametervärde som ger högst sannolikhet åt observationerna, men bayesianen är betydligt mer ambitiös. Låt oss åter se på det enkla fall, då man antar att hon har ett stickprov från en normalfördelning men inte känner till väntevärdet. Låt oss för tydlighets skull döpa normalfördelningen ifråga till ”den ursprungliga fördelningen”.

Bayesianen betraktar nu det okända väntevärdet μ hos den ursprungliga fördelningen som en storhet med en *egen* sannolikhetsfördelning *ex ante.* Hon har före experimentet bestämt sig för att tro på en viss sådan fördelning. Antagligen är det en ganska flack normalfördelning, som bland annat har ett eget väntevärde μ'. Denna ”meta-fördelning” *ex ante* av μ leder, tillsammans med de iakttagna data och med hjälp av Bayes teorem för kontinuerliga fördelningar, till en viss meta-fördelning *ex post*. Toppen i denna fördelning *ex post* anger det värde μ_m för den ursprungliga, okända parametern μ som har *högst sannolikhet givet evidensen* (eller om man ska vara petnoga, det värde där frekvensfunktionen för μ nu har sitt maximum). Man inser lätt på intuitiv väg, att detta mest sannolika värde för parametern μ kommer att ligga någonstans mellan medelvärdet i stickprovet och väntevärdet μ' i *ex ante*-fördelningen, och närmare det förra ju flackare den valda *ex ante*-fördelningen var.

Bayesianen nöjer sig dock inte ens nu. Andra värden på μ är ju fortfarande mycket möjliga, och om aposterioridistributionen för μ' *inte* är en någorlunda toppig, symmetrisk fördelning kan det hända att en ganska liten del av sannolikhetsmassan ligger nära μ_m, alltså den ”bästa” bayesianska skattningen. Bayesianens gissingar beträffande *framtida* stickprov från fördelningen utgår därför från *hela* aposteriorifördelningen för den okända parametern, inte på en enskild ”bästa” skattning av dess värde.

De neurala nätverk som vanligen kallas ”bayesianska” bygger på samma princip.[118] Man antar återigen att de data som ska analyseras härrör från en slumpprocess med okända parametrar. De okända parametrarna är vik-

[118] Termen ”bayesianskt nätverk” kan betyda flera andra saker. Se avsnitt 9.4.

terna i ett neuralt nätverk, och de tilldelas en fördelning *ex ante*. ”Träningen” på en datamängd innebär en numerisk approximation av fördelningen *ex post*, givet dessa data. Resultatet blir alltså inte (som vid träning av ett ANN enligt ML-principen) den bestämda uppsättning vikter som ger den största sannolikheten åt data, utan *en kontinuerlig sannolikhetsfördelning över alla möjliga uppsättningar av vikter i nätverket*. Mer om allt detta i avsnitt 9.4.

Det går, som vi visat ovan, förvisso att göra de behövliga bestämningarna av fördelningarna *ex post* med hjälp av Bayes kontinuerliga sats – men bara om man valt en fördelning *ex ante* för de okända parametrarna. Och vilka skall kriterierna vara för detta val? Är man subjektivist i sannolikhetsfilosofin – se inledningen av detta avsnitt – så kan man ta ganska lätt på problematiken kring apriorisannolikheter: sannolikhet är grad av tro, så man tager den tro som man haver och använder den som apriorisannolikhet. Men om man inte är subjektivist, utan vill göra en åtskillnad mellan *tro* och *berättigad tro* samtidigt som man vill använda bayesiansk inferensteori, så är saken inte lika enkel.

En typ av argument som ofta förekommer i diskussionen är så kallade *okunnighetsargument*, som är nära besläktade med den klassiska sannolikhetsuppfattningen. De säger, att om man inte har någon evidens som talar för någon av hypoteserna H_1 och H_2 gentemot den andra, så skall man (om de är de enda möjliga alternativen) tilldela båda sannolikheterna 1/2. Det analoga resonemanget beträffande kontinuerliga sannolikhetsfördelningar säger, att man i brist på evidens angående en fördelnings utseende skall utgå från att den är någorlunda *platt* (antingen likafördelad i ett begränsat intervall, eller mycket flack i ett obegränsat intervall). Detta är som redan antytts en vanlig procedur bland praktiserande bayesianer, där man talar om dylika aprioriförd elningar som ”icke-informativa”.

En annan metodik är att försöka extrahera något slags genomsnittlig uppfattning bland auktoriteter på det område man forskar om. Detta tillvägagångssätt leder i allmänhet till mer distinkta, ”informativa” fördelningar *ex ante* av de okända parametrarna. Förvisso är denna metod i någon mening ”mindre subjektiv” än metoden att bara ta den tro som man *själv* haver, men traditionella statistikteoretiker accepterar den ändå inte som grund för statistisk inferens.

Lyckligtvis kan bayesianer och traditionalister samsas i en viktig fråga. Det är nämligen så, vilket man lätt övertygar sig om med hjälp av några

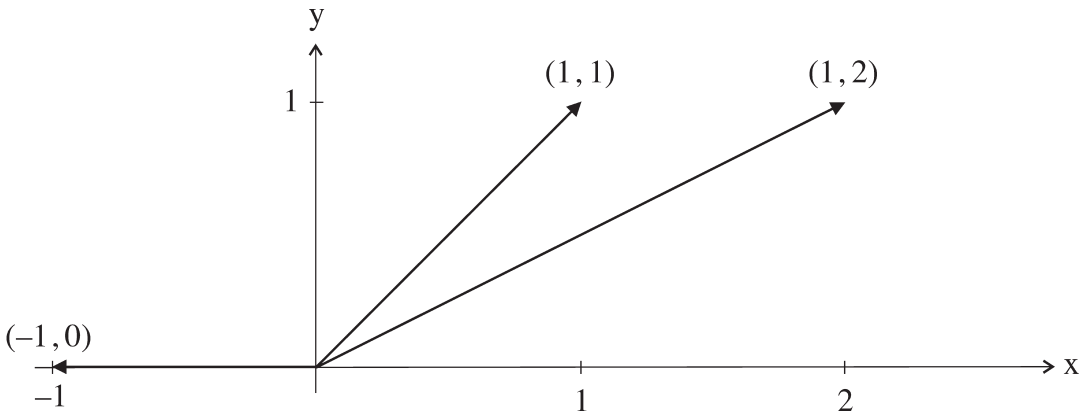
exempel, att *apriorisannolikheternas betydelse enligt Bayes formel minskar med ökande mängd data*. Antag en falskspelare ger oss ett mynt som antingen är äkta eller falskt. Antag också att om det är falskt, så har det en inbyggd sannolikhet om ¾ att ge krona i ett givet kast. Vi kastar myntet 1000 gånger och får 506 krona. Det måste nu till en *mycket* extrem aprioriförde lning för att bayesianen inte ska acceptera samma slutsats som traditionalisten – att myntet är äkta. Dock är denna ”super-empiristiska” lösning inte alltid praktiskt användbar i vetenskapen; tvärtom är det ofta en stor brist på data, och då måste konflikten lösas på annat sätt.

Vi ska inte försöka lösa dessa svåra tvistefrågor här, utan lämnar tills vidare inferensteorin och sannolikhetsfilosofin. Det kommer dock att finnas flera anledningar att återvända till ämnet i samband med den följande diskussionen.

5.2 Vektorer och matriser

För att underlätta beskrivningen av neurala nätverk skall vi nu introducera några elementära begrepp och resultat från s.k. linjär algebra. Matematiskt skolade läsare kan hoppa över åtminstone stora delar av detta långa avsnitt (men läs åtminstone utvikningen om kompetitiva nätverk).

En *vektor* bör egentligen definieras som ett objekt som uppfyller en viss axiomatisk teori – teorin för vektorrum. Det är dock vanligt att man identifierar vektorer med en viss representation av dem, nämligen *ordnade uppsättningar av positiva eller negativa tal*. Man kan då tänka på en vektor som en punkt i ett visst koordinatsystem (med lika många dimensioner som antalet komponenter i vektorn), eller som den riktade sträckan mellan systemets origo och dessa punkter. Se figur 28.



Figur 28. Tvådimensionella vektorer illustrerade i ett koordinatsystem.

Vi ska hålla oss till detta förenklade betraktelsesätt i det följande, utom i avsnittet om koordinattransformationer. Där kommer vi att införa termen *fysisk vektor* för de *objekt* som representeras genom talräckor (koordinater). Om man tänker på fysiska vektorer som *verkliga* riktade sträckor – exempelvis pilarna i figur 28 utan relation till det specifika koordinatsystemet där – så förstår man att vektorer kan få olika representationer i olika koordinatsystem. Men tills vidare låtsas vi alltså som om en vektor och en viss representation av den vore samma sak.

Radvektorer och kolumnvektorer

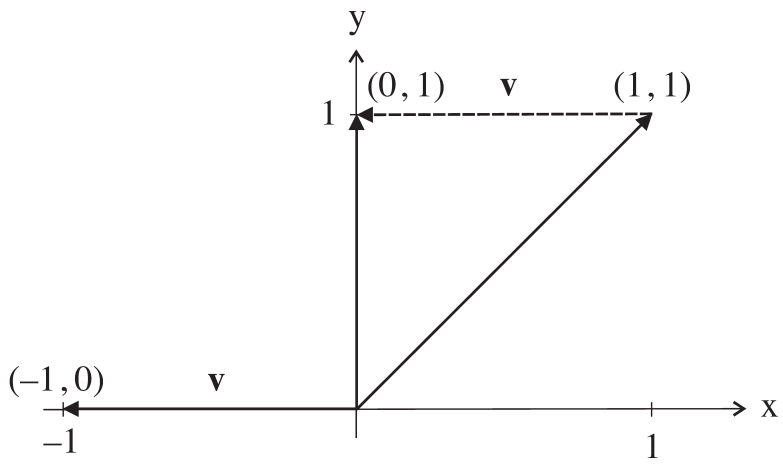
En vektor kan skrivas horisontellt (en radvektor) eller vertikalt (en kolumnvektor). Exempel på det senare är:

$$\begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

som är likvärdig med (2, 1, 3). När vi talar allmänt om vektorer betecknar vi dem med en fetstilt bokstav (oftast bokstaven **v**) med ett index, t.ex. $\mathbf{v}_2$.

Operationer med vektorer

Addition av vektorer sker genom komponentvis "vanlig" addition. Detta motsvarar att man lägger vektorerna "efter varann" i koordinatsystemet. Exempelvis är (1, 1) + (–1, 0) = (0, 1). Ordningen spelar ingen roll vid vektoraddition. Rita gärna in (–1, 0) + (1, 1), i den ordningen, i figur 29!



Figur 29. Addition av vektorerna (1, 1) och (-1, 0).

Multiplikation av en skalär (ett tal) med en vektor innebär att man multiplicerar alla komponenterna med det skalära talet. Operationen betecknas här med vanligt multiplikationstecken; exempel: $2 \cdot (3, 5) = (6, 10)$

Skalär multiplikation av vektorer betyder däremot att man multiplicerar komponentvis och lägger samman produkterna. Vi använder här symbolen $*$ för denna operation. Således gäller:

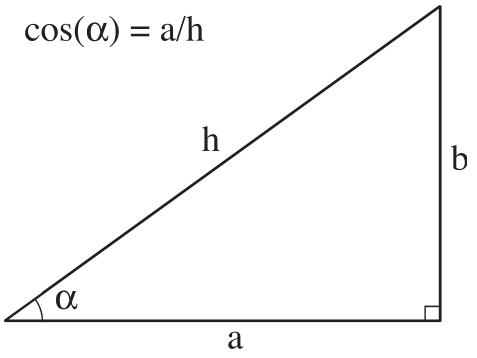
(5.2.1) *Skalärprodukt (exempel):* $(1, 1) * (2, 1) = 1 \cdot 2 + 1 \cdot 1 = 3$

Operationen resulterar alltså i ett enda tal (en skalär, därav namnet). Vektorerna måste ha samma komponentantal vid skalär multiplikation. Ordningen mellan vektorerna spelar däremot ingen roll, inte heller om de är rad- eller kolumnvektorer.

Man kan geometriskt bevisa, att för den skalära produkten $\mathbf{v_1} * \mathbf{v_2}$, definierad som summan av de komponentvisa produkterna enligt ovan, av två vektorer i planet gäller:

(5.2.2) *Skalärprodukt (alternativ):* $\mathbf{v}_1 * \mathbf{v}_2 = l(\mathbf{v}_1) \cdot l(\mathbf{v}_2) \cdot \cos\alpha$

där $l(\mathbf{v})$ är längden av vektorn $\mathbf{v}$ (som kan beräknas från komponenterna i $\mathbf{v}$ med Pythagoras sats) och α är vinkeln mellan $\mathbf{v_1}$ och $\mathbf{v_2}$. Cos α (cosinus för α) kan för $0 < \alpha < 90$ förklaras som förhållandet a/h mellan den närliggande katetern och hypotenusan i en rätvinklig triangel, där en vinkel är α. Se figur 30.

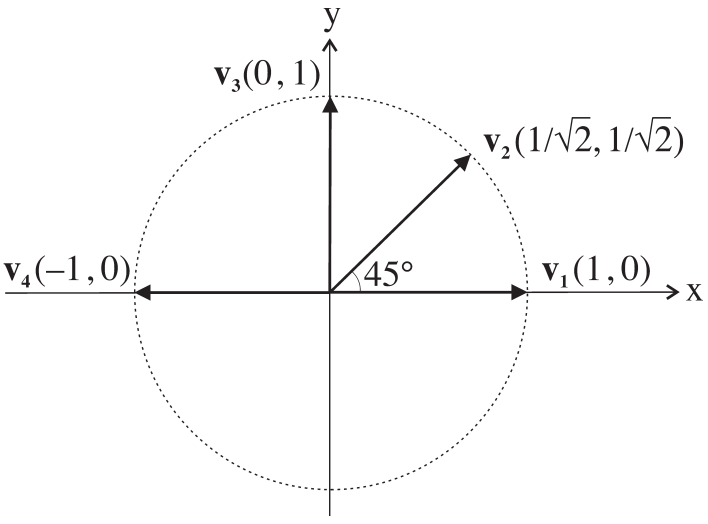


Figur 30. Förklaring av cosinusfunktionen.

I många framställningar *definieras* för övrigt skalärprodukten mellan vektorerna $\mathbf{v_1}$ och $\mathbf{v_2}$ som $l(\mathbf{v_1}) \cdot l(\mathbf{v_2}) \cdot \cos \alpha$, och då kan man istället bevisa att skalärprodukten kan beräknas genom komponentvis multiplikation följd av addition över resultaten.

Av den geometriska tolkningen av skalärprodukt följer flera intressanta saker. Eftersom cos α har maximum (= 1) vid $\alpha = 0°$, är 0 för $\alpha = 90°$ och har minimum (= –1) för $\alpha = 180°$, så är skalärprodukten av två vektorer med given längd maximal när de har samma riktning (dvs. är parallella), har värdet 0 när de bildar rät vinkel mot varann (dvs. är ortogonala) och har minimum när de går i motsatt riktning.

Betrakta vektorerna $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{v_3}$ och $\mathbf{v_4}$ i figur 31.



Figur 31. Parallellitet och ortogonalitet. Förklaring: Se text.

Tillämpning av den aritmetiska tolkningen av skalärprodukt ger vid handen att $\mathbf{v}_1 * \mathbf{v}_1 = 1$, $\mathbf{v}_1 * \mathbf{v}_2 = 1/\sqrt{2}$, $\mathbf{v}_1 * \mathbf{v}_3 = 0$ och $\mathbf{v}_1 * \mathbf{v}_4 = -1$. Verifiera gärna att detta stämmer med den geometriska tolkningen (observera att $\cos 45° = 1/\sqrt{2}$)!

I figuren ovan har vi att göra med *normerade* eller *normaliserade* vektorer, dvs. de har alla längden 1. Då gäller alltså att skalärproduktens storlek entydigt bestäms av vinkeln mellan vektorerna, och att den har maximum då vektorerna är identiska. En annan faktor som i det allmänna fallet bestämmer en skalärprodukts storlek är uppenbarligen vektorernas längd.

Linjärt oberoende

En uppsättning vektorer sägs vara *linjärt oberoende* om ingen av vektorerna kan skrivas som en linjär kombination av de andra. Ett exempel på motsatsen (dvs. linjärt beroende) är paret

(2, 4); (4, 8)

där den andra vektorn är lika med den första multiplicerad med 2. Ett annat är trion

(1, 2); (2, 3); (4, 7)

eftersom det gäller att

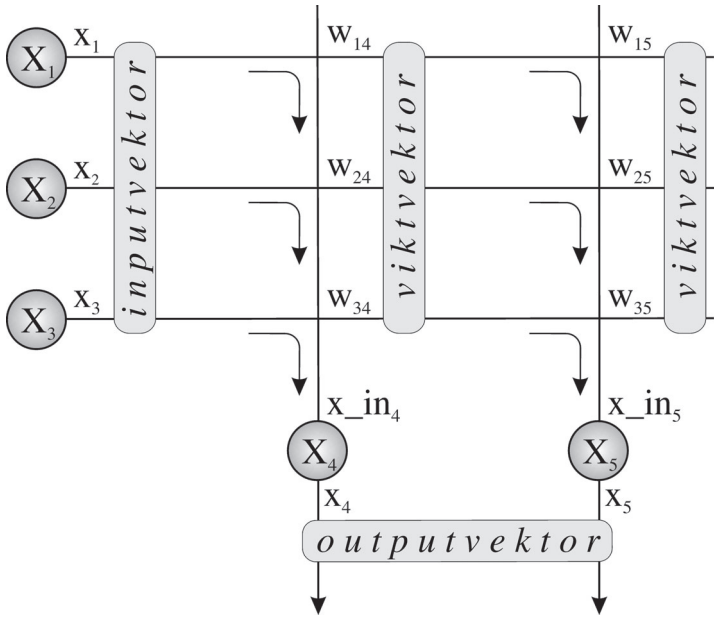
(5.2.3) $(4, 7) = 2 \cdot (1, 2) + (2, 3)$

De två vektorerna (1, 2) och (2, 3) är däremot inbördes linjärt oberoende.

Begreppet linjärt oberoende har nära relationer till begreppen *vektorrum* och *koordinatsystem.* Se vidare sist i detta avsnitt!

Input-, vikt- och outputvektorer i nätverk

Tänk återigen på ett enlagrat feed-forward-nätverk med 3 inputenheter och 2 outputenheter, och betrakta figur 32 som anger dess Hinton-diagram:



Figur 32. Hinton-diagram av ett feed-forward-nät. w_{ij}: vikter. x_i: aktiviteter. x_in_i: nettoinput.

Nätverket räknar ut ju nettoinput x_in_j till en outputenhet X_j så här: först multipliceras, för varje enhet X_i som bidrar till aktiveringen av X_j, aktiviteten x_i med vikten w_{ij}; därefter summeras alla bidragen. Men de olika inputs x_i bildar tillsammans en vektor, och detsamma gäller de vikter w_{ij} som hör till en given utnod X_j. Nettoinput x_in_j till varje enskild outputenhet X_j blir därför helt enkelt *skalärprodukten* mellan *inputvektorn* **x** och den *viktvektor* $\mathbf{w}_j$ som beskriver förbindelserna till outputenheten ifråga (dvs. den kolumnvektor av vikter som ligger rakt ovanför outputenheten i diagrammet). Inputvektorerna och viktvektorerna kan tyckas vara mycket olika slag av företeelser. Här gäller det dock att tänka på dem på ett abstrakt sätt, nämligen som ordnade talräckor med lika många komponenter.

Kom nu ihåg att resultatet av en skalärmultiplikation mellan vektorer är beroende av vinkeln mellan dessa! När en inputvektor multipliceras med ett nätverks viktvektorer, kommer storleken av nettoinput till en viss enhet i det andra skiktet av enheter således att bero dels av respektive viktvektors längd, dels av dennas riktning i förhållande till inputvektorn. Som ett specialfall gäller, att *om alla vektorer är normaliserade till samma längd, så fås maximal respons då inputvektorn och viktvektorn är identiska* ($\cos \alpha = 0$).

Man kan sammanfattningsvis säga, att ett ANN börjar sin signalbearbetning med att göra ett slags jämförelser mellan å ena sidan inputvektorn, å andra sidan viktvektorerna till de olika enheterna i nästa lager i nätverket. Särskilt träffande är denna beskrivning då vektorerna är normaliserade.

I ett maximalt linjärt, enlagrat nätverk (med identitet som aktiveringsfunktion) bildar de olika nettoinputs till det andra skiktet av enheter direkt nätverkets *outputvektor* **o**. I andra enlagrade nätverk krävs att en aktiveringsfunktion transformerar vektorn av nettoinputs till en vektor av aktiviteter; mer därom nedan. I flerlagrade feedforwardnät beräknas för varje lager en skalärprodukt mellan vektorn av detta lagers aktiviteter och viktvektorn för nästa lager, på det sätt som vi just beskrivit.

Utvikning om kompetitiva nät

I många kompetitiva nätverk låter man *den nod som får maximal nettoinput* av en viss inputvektor vara ”vinnaren” för denna input och därmed ensam få aktiviteten 1. Detta kan realiseras inom ramen för vanliga aktiveringsfunktioner genom att alla enheter i populationen, efter det att de fått input från omvärlden, genom feedbackförbindelser gradvis inhiberar varandra men exciterar sig själva på ett lämpligt avvägt sätt. Jämför nedan, avsnitt 8.2. Ofta – och särskilt ofta om man sysslar med matematiska tillämpningar snarare än biologisk modellering – hoppar man dock över detta steg, och vi ska göra detsamma här. Under förutsättning att man arbetar med normaliserade vektorer, eller en approximation av detta villkor, kommer den nod som har maximal nettoinput (och alltså utses till vinnare) att ha en viktvektor som relativt sett ligger *nära* inputvektorn, i en geometrisk mening. Detta följer omedelbart av vad vi sade ovan om hur ett neuralt nätverk ”jämför” inputvektor och viktvektorer. Om vektorerna inte alls är normaliserade, utan tillåts ha godtyckliga längder, följer det inte.

Ett alternativ (som alltså inte är *generellt* likvärdigt med att definera vinnaren som noden med maximal nettoinput) är att låta den vinnande enheten för en viss input vara *den nod vars viktvektor ligger närmast inputvektorn i vektorrummet*, där ”närhet” oftast definieras i termer av euklideiskt avstånd. Detta sätt att utse vinnare ger påtagligt avvikande resultat från det föregående när vektorerna har mycket olika längd. Det är viktigt vid kodningen av data till ett kompetitivt nätverk att man tänker på vilka faktorer det är som bestämmer storleken av en respons på en input, och man måste därför veta vilken vinnarmekanism nätverket använder sig av.

Matriser

En matris är en tvådimensionell struktur av tal, t ex:

$$\begin{pmatrix} 2 & 3 & 1 \\ 1 & 10 & 0 \end{pmatrix}$$

En matris kan sägas bestå av radvektorer arrangerade i en kolumn. I det här fallet är det två stycken trekomponents radvektorer, nämligen (2, 3, 1) och (1, 10, 0). Men matrisen kan också beskrivas som kolumnvektorer arrangerade i en rad. I det här fallet är det tre stycken tvåkomponents kolumnvektorer. Vår matris sägs vara en (2, 3)-matris eftersom den har två rader och tre kolumner. Förväxla inte beteckningen (2, 3) i detta sammanhang med en vektor, eller med beteckningarna på matrisens element! Som allmänna beteckningar på matriser väljer vi fetstilta versaler, t.ex. **A** och **B**.

En radvektor med n komponenter kan uppfattas som en (1, n)-matris, en kolumnvektor med n komponenter som en (n, 1)-matris.

Elementen i en matris betecknas med dubbelindex, där den första siffran står för raden och den andra för kolumnen där elementet hör hemma. Om matrisen ovan får heta **A** är således $a_{11} = 2$, $a_{21} = 1$ och $a_{23} = 0$. Observera att något element a_{32} inte existerar i **A**.

Ett allmänt beteckningssätt för en (m,n)-matris är:

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

vilket man ibland helt enkelt sammanfattar som (a_{mn}).

Transposition av en matris

Om man gör kolumnerna i en matris **A** till rader, och vice versa, får man den *transponerade* matrisen $\mathbf{A}^T$ (också kallad ”transponatet” av **A**). Exempel:

(5.2.4) $$\begin{pmatrix} 2 & 3 & 1 \\ 1 & 10 & 0 \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} 2 & 1 \\ 3 & 10 \\ 1 & 0 \end{pmatrix}$$

Det följer, att en radvektor alltid har motsvarande kolumnvektor som transponat, och vice versa.

Operationer med matriser

Två matriser med samma dimensioner kan *adderas* genom att motsvarande komponenter adderas. Således är

(5.2.5) $$\begin{pmatrix} 2 & 3 & 1 \\ 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 1 & -1 & -1 \\ 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 0 \\ 3 & 3 & 2 \end{pmatrix}$$

Observera villkoret att dimensionerna måste vara lika; en (2, 3)-matris kan alltså inte adderas med en (3, 4)-matris. Ordningen mellan matriserna spelar ingen roll vid addition.

Multiplikation av en matris med en skalär innebär att alla matrisens komponenter multipliceras med det skalära talet.

Matrismultiplikation av matrisen **A** med matrisen **B**, som vi här symboliserar med **AB**, är något helt annat. Operationen kan enklast definieras genom att *elementet c_{ij} i den resulterande matrisen **C** skall vara lika med skalärprodukten av radvektor i från **A** och kolumnvektor j från **B***. Det här innebär att raderna i **A** måste ha lika många komponenter som kolumnerna i **B**, eller, med andra ord, att det måste finnas lika många kolumner i **A** som rader i **B**. Låt oss försöka utföra

$$\begin{pmatrix} 2 & 3 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 & 1 & 1 \\ 1 & 5 & -1 & 0 \\ 2 & -2 & 1 & -3 \end{pmatrix}$$

Vi har en (2, 3)-matris och en (3, 4)-matris, så dimensionsvillkoret är uppfyllt. Elementet a_{11} i den resulterande matrisen är (2, 3, 1) ∗ (3, 1, 2) = $= 2 \times 3 + 3 \times 1 + 1 \times 2 = 11$. Vilket blir det sista elementet i matrisen? Rimligen skalärprodukten av andra raden i den första matrisen med fjärde kolumnen i den andra, alltså (1, 1, 0) ∗ (1, 0, –3) = 1. Detta är närmare

bestämt element a_{24}. Produktmatrisen har alltså 2 rader och 4 kolumner, och resultatet som helhet ser ut så här:

$$(5.2.6) \qquad \begin{pmatrix} 2 & 3 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 & 1 & 1 \\ 1 & 5 & -1 & 0 \\ 2 & -2 & 1 & -3 \end{pmatrix} = \begin{pmatrix} 11 & 13 & 0 & -1 \\ 4 & 5 & 0 & 1 \end{pmatrix}$$

Kontrollera gärna att detta stämmer!

Till skillnad från vanlig multiplikation är matrismultiplikation känslig för ordningen mellan termerna. Multiplikation av **A** med **B** är med andra ord inte samma sak som multiplikation av **B** med **A**. En (m, p)-matris **A** kan som nämnts multipliceras med en (p, n)-matris **B**, och resultatet blir en (m, n)-matris. Om $m \neq n$ går det ändå inte att multiplicera **B** med **A**. I exemplet ovan hade man ju då behövt multiplicera radvektorer ur matris 2 (med fyra komponenter) med kolumnvektorer ur matris 1 (med två komponenter), och det går inte. Även i de fall då det går att genomföra operationen åt båda hållen är **AB** i allmänhet $\neq$ **BA**.

Av denna anledning används ofta termerna ”vänstermultiplikation” respektive ”högermultiplikation”. Vi ska dock försöka undvika dessa uttryck, som också kan leda till missförstånd. I denna framställning syftar ”**A** multipliceras med **B**” därför på att man utför operationen **AB**. Vill vi tala om operationen **BA**, säger vi istället att **B** multipliceras med **A**.

Multiplikation mellan vektorer och matriser

Eftersom en radvektor med n komponenter kan uppfattas som en (1, p)-matris kan den multipliceras med en (p, n)-matris. Resultatet blir en (1, n)-matris, dvs. en ny radvektor. Exempelvis är

$$(5.2.7) \qquad \begin{pmatrix} 1 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 2 \end{pmatrix}$$

Varje komponent i resultatvektorn är produkten av den ursprungliga radvektorn och en kolumn i matrisen. Ett specialfall är när den matris man multiplicerar radvektorn med är en (p, 1)-matris, dvs. en enstaka kolumnvektor. Vi får då som resultat en (1, 1)-matris, som innehåller vektorernas

skalärprodukt. Analogt kan man multiplicera en (n, p)-matris med en kolumnvektor av typen (p, 1); resultatet blir en radvektor.

Om man prövar att transponera båda matriserna i vänsterledet av (5.2.7) och ändrar ordningen på multiplikationen (gör det!), så får man ut transponatet av högerledet, alltså en viss kolumnvektor. Detta exemplifierar en allmän räkneregel som säger

(5.2.8) $\quad (\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$

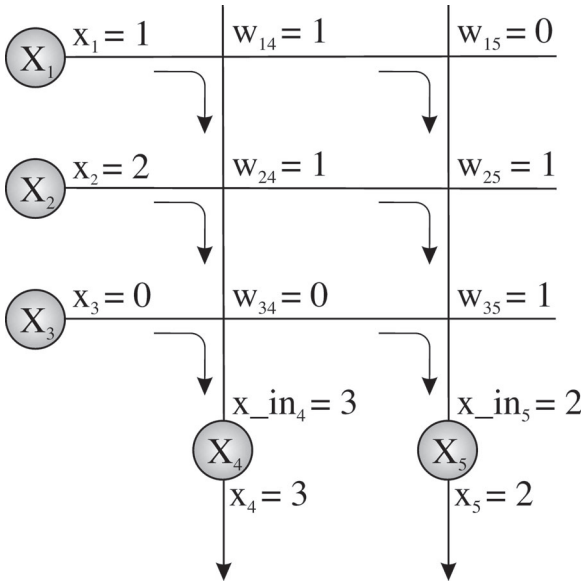
och som vi ska ha viss användning av senare.

Ett nätverks viktmatris

Det som händer i ett neuralt nätverk när det beräknar nettoinput till det andra skiktet av enheter är ju, att inputvektorn skalärmultipliceras med kolumner av vikter (jämför ovan) – med andra ord är det fråga om *en multiplikation av en vektor (inputvektorn) med hela matrisen av vikter*. Observera att vi formellt betraktar inputvektorn som en *radvektor* när vi multiplicerar in den i vikt matrisen (se 5.2.7), även om vi representerar den vertikalt i Hinton-diagrammet (fig. 32)!

Det följer av vad vi sade alldeles nyss och formel (5.2.8), att man lika gärna kan beskriva det som händer i termer av en multiplikation av en transponerad viktmatris med inputvektorn, betraktad som kolumnvektor. Tills vidare ska vi dock anlägga det första betraktelsesättet.

I ett maximalt linjärt nätverk är aktiviteten i en enhet = dess nettoinput; här blir därför resultatet = outputvektorn när vi multiplicerar inputvektorn med viktmatrisen. Låt oss sätta in värdena från vår lilla multiplikation 5.2.7 i ett Hinton-diagram för att illustrera hur det hela fungerar.

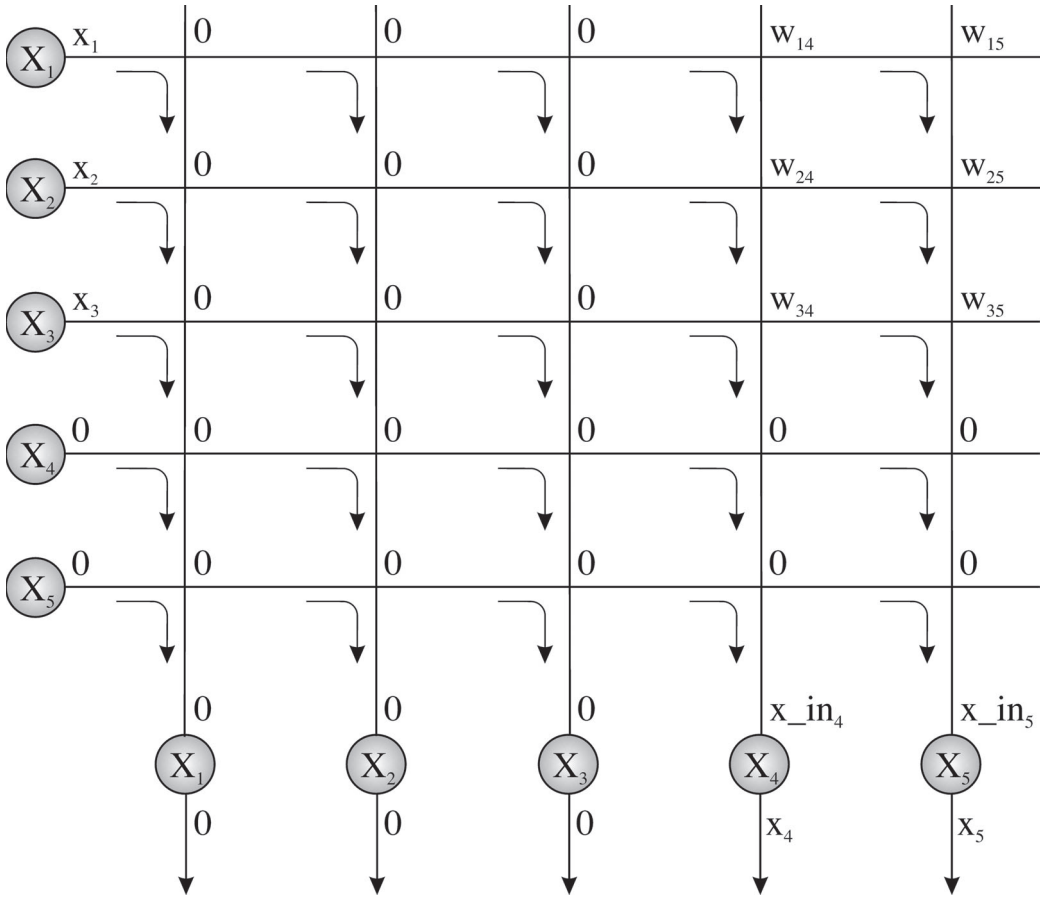


Figur 33. Skalärmultiplikation av inputvektor och viktmatris i ett maximalt linjärt, enlagrat nätverk. w_{ij}: vikter. x_i: aktiviteter. x_in_i: nettoinput.

Det enda som inte riktigt tycks stämma med den vektor- och matristerminologi som vi infört är numreringen av komponenter och kolumner. Outputvektorn har två komponenter, men de numreras inte från 1 till 2, utan från 4 till 5. Och var är till exempel den första kolumnen, som borde vara w_{11}–w_{31}, i viktmatrisen?

Man kan i och för sig döpa om outputenheterna i ett feedforwardnät och använda separata index för dem, och det görs ofta. I vårt fall kunde man t.ex. kalla dem Y_1 och Y_2; då skulle vi ju få en matris vars kolumner var ”rätt” numrerade. Men istället blir det fel på ett annat ställe, nämligen i viktbeteckningarna, eftersom w_{11} inte längre blir vikten hos en förbindelse från ett visst element till *samma* element! Vi kommer senare ibland (bland annat i avsnittet om den linjära associatorn) att ta oss friheten att skriva vikterna på detta sätt, men vi ska då vara observanta på att de har en för sammanhanget specifik numrering.

De här egenheterna kan sägas bero på att vi ”egentligen” opererar med större matriser, där vissa element är lika med noll. Vi kan visa detta i ett fullständigt Hinton-diagram, dvs. ett diagram där alla enheter är med både på input- och outputsidan. Nettoinput till inputenheterna (dvs. de enheter som aktiveras genom extern input), samt alla vikter utom de på förbindelserna från inputenheter till outputenheter, måste givetvis tilldelas värdet 0 i ett sådant diagram.



Figur 34. Fullständigt Hinton-diagram för ett maximalt linjärt nätverk. w_{ij}: vikter. x_i: aktiviteter. x_in_i: nettoinput.

Man inser lätt att det går bra att betrakta det som händer i detta nätverk som en skalär multiplikation av den ”fullständiga inputvektorn” (x_1, x_2, x_3, 0, 0) med hela viktmatrisen, inklusive alla dess nollor. Resultatet blir den ”fullständiga outputvektorn” (0, 0 , 0, x_4, x_5).

Det fullständiga Hinton-diagrammet har också följande vackra variant. Vi tittar på det helt allmänna fallet där det inte är bestämt att vissa vikter skall vara noll, dvs. där alla förbindelser är i princip tillåtna.

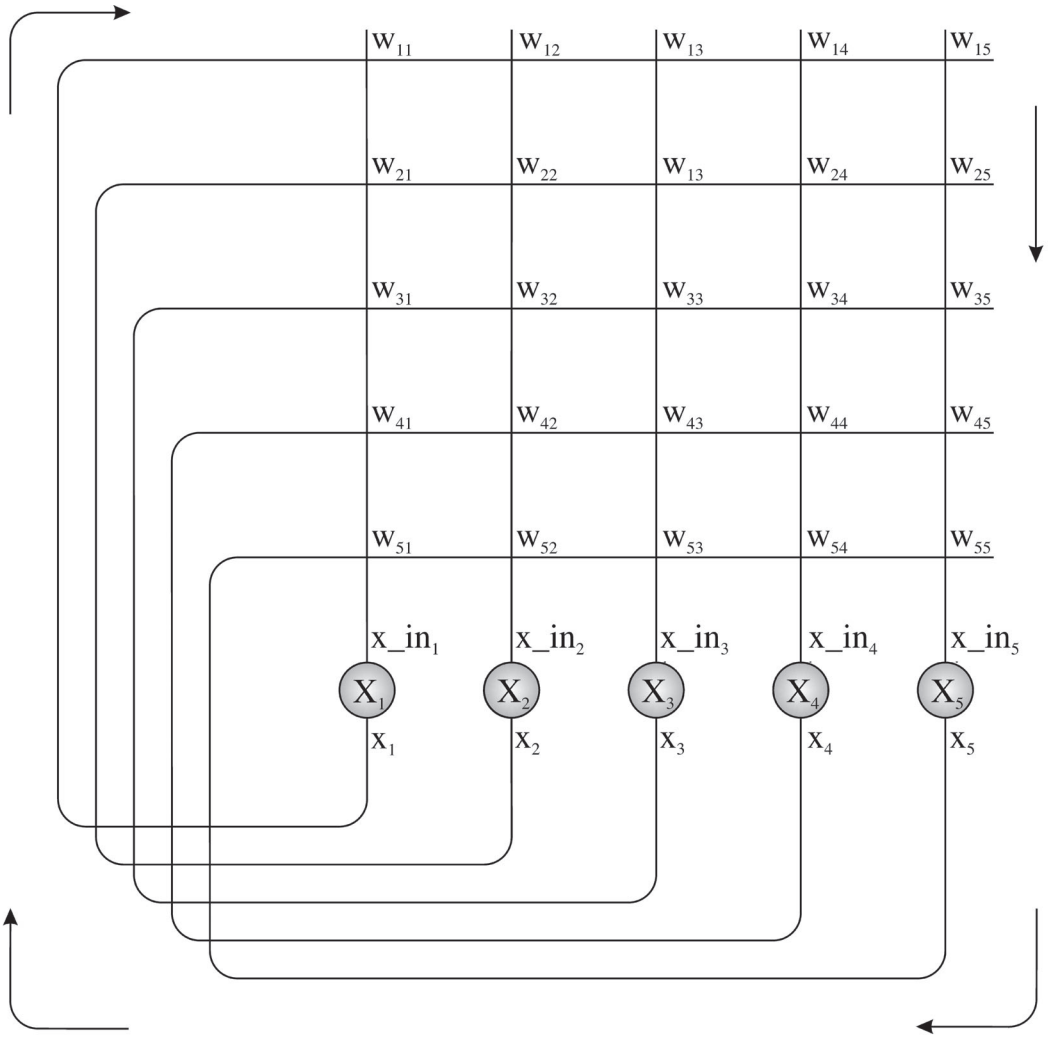


Fig. 35. Fullständigt Hintondiagram, variant. w_{ij}: vikter. x_i: aktiviteter. x_in_i: nettoinput.

Informationsflödets riktning har som vanligt symboliserats med pilar. Fördelarna med detta illustrationssätt är att man bara behöver rita varje neuronal enhet en gång, och framförallt att det kan användas även när det finns feedback och man alltså inte kan rita det förenklade Hinton-diagrammet. Likaså avbildar man på ett pedagogiskt sätt det förhållandet, att såväl input- som outputvektorn skall uppfattas som radvektorer.

Sammanfattning

Vi kan nu sammanfatta något av det ovan sagda i några ekvationer, där vi för tydlighets skull också tar med tiden som en variabel. Låt **x**(t) och **x_in**(t) vara aktivitetsvektorn respektive vektorn av nettoinputs i ett en-

lagrat nätverk vid tiden t, uppfattade som radmatriser, och låt **W** vara viktmatrisen. Då gäller att:

(5.2.9) $\mathbf{x\_in}(t+1) = \mathbf{x}(t)\mathbf{W}$

I det allmänna fallet har ouputenheterna en godtycklig aktiveringsfunktion, och vi kan sammanfatta vad som händer där genom formeln

(5.2.10) $\mathbf{x}(t) = f(\mathbf{x\_in}(t))$

Av de två senaste formlerna följer

(5.2.11) $\mathbf{x}(t+1) = f(\mathbf{x}(t)\mathbf{W})$

I de enklaste linjära nätverken är f identitetsfunktionen, dvs

(5.2.12) $\mathbf{x}(t) = \mathbf{x\_in}(t)$

och alltså

(5.2.13) $\mathbf{x}(t+1) = \mathbf{x}(t)\mathbf{W}$

Koordinattransformationer och inversa matriser

Vi ska nu berätta om en viss grupp av s.k. koordinattransformationer (på matematiskt språk de "affina" transformationerna), och påtala deras betydelse för förståelsen av vad ett artificiellt neuralt nätverk gör.

I ett n-dimensionellt vektorrum, dvs. mängden av möjliga vektorer med n komponenter, finns alltid högst n stycken linjärt oberoende vektorer. För en uppsättning *B* av n st oberoende men i övrigt godtyckliga vektorer $\mathbf{b}_i$ gäller med andra ord, att varje annan vektor kan skrivas som en linjärkombination av dem. Detta kan i sin tur läggas till grund för att välja vektorerna i *B* som *basvektorer i ett nytt koordinatsystem,* med samma origo som det ursprungliga men inte längre nödvändigtvis med räta vinklar mellan axlarna.

För att förstå hur ett byte av koordinatsystem går till är det bäst att tänka sig vektorer som fysiska storheter som vi kan beskriva med olika koordinater i olika system. Låt oss döpa det gamla och det nya koordinatsy-

stemet till K respektive K'. En viss (fysisk) vektors koordinater **v'** i det *nya* systemet anger hur den kan beskrivas som en viss viktad summa av basvektorerna för K'. Men varje basvektor för K' har ju en uppsättning koordinater $\mathbf{b}_i$ i det gamla systemet K. För att få den aktuella fysiska vektorns koordinater i det *gamla* systemet, **v**, beskriver man vektorn som *samma* viktade summa av de nya basvektorerna, men nu med de senare givna genom de koordinater som de har i det *gamla* systemet. Detta innebär *att man får en vektors gamla koordinater **v** genom att multiplicera dess nya koordinater **v'**, betraktade som en radvektor eller (1, n)-matris, med en (n, n)-matris där raderna är de nya basvektorernas gamla koordinater*. Denna matris, den s.k. transformationsmatrisen, karakteriserar entydigt vårt byte av koordinatsystem. Låt oss här kalla den **T**. Eftersom en koordinattransformation är entydigt bestämd av sin transformationsmatris identifierar man ofta transformationen (på svengelska: "transformen") och matrisen.

Vill man, i det fall vi talar om, istället gå från en fysisk vektors koordinater i det gamla koordinatsystemet till dess koordinater i det nya blir det hela bara en liten aning värre. Hade man tillgång till de gamla basvektorernas ekvationer i det nya systemet vore det en lätt match, men nu har vi inte det automatiskt. Man kan dock lösa problemet genom att leta reda på den transformation **M** som överför de nya basvektorerna från de gamla till de nya koordinaterna. Basvektorerna är ju normvektorer i det nya koordinatsystemet. Vi söker därför den matris **M** för vilken det gäller att

(5.2.14) $$\mathbf{TM} = \mathbf{I}$$

där **I** är identitetsmatrisen

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Den matris **M** som uppfyller (5.2.14) kallas *inversen* till **T** och betecknas med $\mathbf{T}^{-1}$. Den inversa matrisen till en transformationsmatris kallas ofta helt enkelt den inversa transformen. Inte alla (n, n)-matriser har en invers. Ett nödvändigt och tillräckligt villkor för existensen av en invers är att radvektorerna (ekvivalent: kolumnvektorerna) är linjärt oberoende.

Vi har hittills diskuterat hur man räknar ut ”nya” koordinater från ”gamla”, och omvänt, givet att man känner till hur basvektorerna i det ena systemet representeras i det andra. En intressant fråga är om man kan räkna fram de nya basvektorernas representation i det gamla systemet, och vice versa, givet att man vet hur *n* stycken *andra* linjärt oberoende vektorer representeras i de båda systemen. Detta är ekvivalent med frågan om man kan beräkna transformationsmatrisen ur kunskapen om hur den överför en matris $\mathbf{M}_1$ av linjärt oberoende vektorer till en annan sådan matris, $\mathbf{M}_2$. Svaret är *ja*. Ur

(5.2.15) $$\mathbf{TM}_1 = \mathbf{M}_2$$

följer nämligen (genom multiplikation av båda leden med $\mathbf{M}_1^{-1}$) att

(5.2.16) $$\mathbf{TM} = \mathbf{M}_1\mathbf{M}_2^{-1}$$

förutsatt att $\mathbf{M}_1^{-1}$ existerar. Men som vi vet är detta fallet, eftersom raderna/kolumnerna i $\mathbf{M}_1$ är linjärt oberoende.

Observera att det inte är något villkor för existensen av **T**, att $\mathbf{M}_2$ består av linjärt oberoende rader/kolumner. Däremot är detta en förutsättning för att det ska finnas en invers transformation. Om $\mathbf{M}_2$ *inte* består av oberoende vektorer, kan **T** uppfattas som en projektion av det n-dimensionella vektorrummet på ett vektorrum av lägre dimensionalitet.

För att se vad allt detta har att göra med neurala nätverk betraktar vi åter formeln (5.2.9), men nu för enkelhets skull utan explicit tidsreferens:

(5.2.17) $$\mathbf{x\_in} = \mathbf{xW}$$

som ju uttrycker att ett ANN multiplicerar inputvektorn **x**, uppfattad som en radvektor av typen (1, n), med en viktmatris **W** av typen (n, p). Här är n antalet inputnoder och p antalet noder i det andra lagret. Om **W** är en (n, n)-matris och består av linjärt oberoende vektorer, så uppfyller (5.2.17) villkoren för en inverterbar transformation. *Ett neuralt nätverk som har lika många inputnoder som noder i det andra skiktet, och dessutom linjärt oberoende viktvektorer, gör alltså en koordinattransformation när det från inputvektorn beräknar nettoinput till nästa lager*. Nettoinputvektorn kan närmare bestämt uppfattas som de nya koordinater för inputvektorn som man får, om viktmatrisens radvektorer väljs som basvektorer i ett nytt koordinatsystem. Om nätverket är helt linjärt gäller det

som just sagts om nettoinputvektorn också om aktivitetsvektorn i det andra skiktet.

Om viktvektorerna inte är linjärt oberoende kan man istället beskriva det som att nätverket projicerar inputrummet på ett rum med färre dimensioner. Detsamma gäller (förstås) om antalet noder i det andra skiktet är färre än antalet inputnoder.

Icke-linjära nätverk kan analogt uppfattas som algoritmer för andra, mer komplicerade koordinattransformationer/projektioner, men detaljerna i detta skall vi avstå från att gå in på tills vidare. Nu ska vi istället äntligen börja beskriva några av de viktigaste enskilda typerna av artificiella neurala nätverk.

6. Enlagrade nätverk

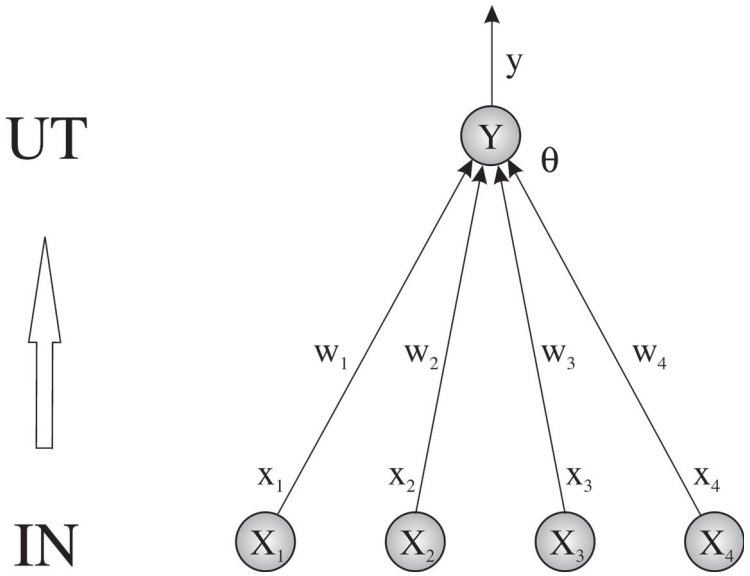
6.1 Den enkla perceptronen

Redan på 1940-talet beskrev McCulloch och Pitts artificiella neuron som kunde utföra logiska operationer med hjälp av en tröskelfunktion.[119] I slutet av 1950-talet konstruerade sedan Rosenblatt ett antal artificiella neurala nätverk för mönsterklassifikation, och implementerade dessutom flera av dem i den tidens hårdvara. Han kallade dessa nätverk "perceptroner" och bevisade intressanta ting om dem, dock utan att nå ända fram till de resultat som under 1980-talet skulle komma att göra (flerlagrade) perceptroner så populära.[120] Vi ska här beskriva en mycket enkel, enlagrad perceptron.

Signaldynamik

Vår enkla enlagrade perceptron innehåller, som namnet antyder, bara ett lager av förbindelser, eller med andra ord två lager av enheter: input- och outputlagret.[121] Enheterna är binära, dvs. deras aktivitet antar bara värdena 1 eller 0. De mönster som man vill klassificera skall alltså vara binärt kodade (dvs. som uppsättningar av 1-or och 0-or). Outputlagret består av en enda nod, "beslutsnoden" – så benämnd för att den signalerar nätverkets beslut om vilken klass den aktuella input tillhör. Förbindelserna mellan inputnoderna X_i och beslutsnoden Y har vikter w_i, som från början bestäms slumpvis. Se figur 36!

[119] McCulloch & Pitts (1943).
[120] Rosenblatt (1962).
[121] I de flesta utföranden har perceptronen ett "förprocessande" lager av förbindelser med fixa vikter. Detta ändrar dock inget principiellt i våra resonemang (jämför avsnitt 6.3).



Figur 36. En enkel perceptron. w_j: vikter. x_i: aktiviteter i inputnoderna X_i. y: aktivitet i outputnoden ("beslutsnoden") Y. θ: tröskel för Y.

Nettoinput till beslutsnoden beräknas på vanligt sätt som summan av inputnodernas aktiviteter x_i viktade med styrkan w_i hos respektive förbindelser. Beslutsnoden använder sig sedan av en stegfunktion med en tröskel θ för att besluta om dess aktivitet y skall vara 1 eller 0. Sammanfattningsvis gäller alltså för den enkla perceptronens signaldynamik, om n är antalet inputnoder:

$$\begin{aligned} &\sum_{i=1}^{n} x_i w_i > \theta \rightarrow y = 1 \\ &\sum_{i=1}^{n} x_i w_i \leq \theta \rightarrow y = 0 \end{aligned} \qquad (6.1.1)$$

Inlärning med perceptronregeln

Givet att man presenterar en uppsättning inputmönster för en otränad perceptron (dvs. en perceptron med slumpvisa vikter) med en viss tröskel, så ger den en bestämd utsignal, 0 eller 1, för varje enskilt mönster. Nu är det förstås inte i allmänhet så, att denna signal överensstämmer med den klassifikation man vill ha utförd. Perceptronen måste därför *tränas* att göra rätt. Denna träning går till så, att vikterna i nätverket ändras på ett visst sätt när nätverket gör "fel", dvs. klassificerar på det icke önskade sättet. Närmare bestämt arbetar perceptronen enligt följande enkla inlär-

ningsregel: (i) Om responsen är rätt, ändra ingenting. (ii) Om responsen är 1 men skulle vara 0, sänk vikterna på alla aktiva inlinjer och öka tröskeln. (iii) Om responsen är 0 men skulle vara 1, höj vikterna på alla aktiva inlinjer och minska tröskeln. Alla höjningar och sänkningar av vikter och tröskel sker med förutbestämda kvantiteter vars storlek bestämmer inlärningstakten. Det ger oss, med positiva konstanter α, β och d som symbol för önskad output:

$$
\begin{aligned}
& y = d \rightarrow \Delta w_i = 0;\ \Delta\theta = 0 \\
& y > d \rightarrow \Delta w_i = -\alpha x_i;\ \Delta\theta = \beta \\
& y < d \rightarrow \Delta w_i = \alpha x_i;\ \Delta\theta = -\beta
\end{aligned}
\qquad (6.1.2)
$$

Istället för en tröskel som ändras kan man ha en fix tröskel men lägga till en s.k. *biasnod*, som alltid antas ha input 1 och vars vikt justeras parallellt med övriga vikter. Dessa två beskrivningar är matematiskt helt ekvivalenta. (Jämför också not 123 nedan!)

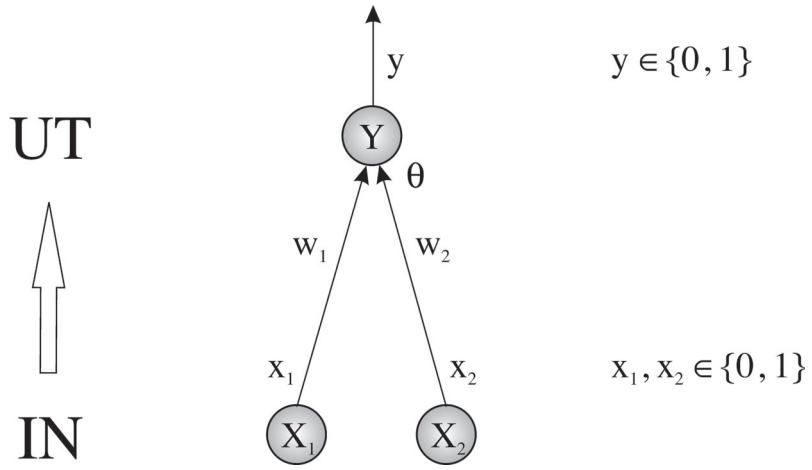
Låt oss nu titta närmare på hur den just nämnda inlärningsalgoritmen, ”perceptronregeln”, fungerar. Om en viss insignal ger 0 som output när vi önskade att den skulle ge 1, så måste nettoinput till outputnoden vara för låg. Alltså kommer man närmare en ”riktig” respons på detta mönster genom att öka vikterna på alla de införbindelser där mönstret ifråga hade en 1:a i inputnoden (de aktiva inlinjerna) och sänka tröskeln. Att öka vikterna på övriga linjer skulle däremot göra vare sig till eller från för responsen för just detta mönster. Det är inte svårt att se att man, om man bara hade att göra med *ett* inputmönster, vid upprepad presentation av mönstret snart skulle få en riktig klassifikation via perceptronregeln. Det omvända resonemanget gäller om responsen på mönstret är 1 men skulle vara 0.

Problemet är bara att man har att göra med många mönster, och de ska läras in på samma gång! Felkorrektionen för *ett* mönster kommer då att påverka svaret på många *andra* mönster. Det är inte på något sätt självklart att alla de höjningar och sänkningar av vikter, som blir resultatet av upprepade applikationer av perceptronens inlärningsregel på alla dessa mönster, leder till att alla – eller ens något av dem – någonsin kommer att klassificeras korrekt. Rosenblatts minnesvärda insats var nu att bevisa att perceptronen faktiskt alltid lär sig att klassificera korrekt, *förutsatt att klassifikationen ligger inom dess teoretiska möjligheter*. Förbehållet är lika viktigt (och lika berömt) som den positiva delen av resultatet, men låt oss spara det en stund.

Rosenblatts resultat säger, annorlunda uttryckt, att om vi har en uppsättning mönster som är indelade i två klasser, och det finns *någon tänkbar tilldelning av vikter* till perceptronen som skulle separera mönstren i just de klasserna, så kommer perceptronen alltid att hitta en sådan vikttilldelning om man tränar den enligt ovan. Detta är ett exempel på ett *konvergensresultat för inlärning:* det säger att *om* det finns en logiskt möjlig lösning av ett visst problem, *så* kommer nätverkets vikter, givet en viss inlärningsprocedur, att konvergera mot en sådan lösning. Vi ska inte referera detta bevis för perceptronen här eftersom det är ganska komplicerat, men vi ska senare möta fler liknande konvergensresultat.

XOR-problemet

Låt oss nu istället titta på förbehållet, ”om det finns någon tänkbar tilldelning av vikter som...”. För att på ett enkelt och åskådligt sätt förstå vad detta villkor innebär ska vi ta en titt på en ytterst simpel perceptron som har bara två inputnoder. Den illustreras i figur 37.



Figur 37. XOR-problemet i den enkla perceptronen. w_j: vikter. x_i: aktiviteter i inputnoderna X_i. y: aktivitet i outputnoden (”beslutsnoden”) Y. θ: tröskel för Y.

Denna maskin tar binära inputs och har alltså bara fyra möjliga inputvektorer – nämligen (1, 1), (1, 0), (0, 1) och (0, 0). Dessa inputmönster kan man vilja klassificera på olika sätt. Kanske vill man av någon anledning sätta etiketten 1 på mönstret (1, 1) och etiketten 0 på de övriga. Det är inte svårt att se att det finns många logiskt möjliga lösningar av detta problem. Exempelvis skulle $\theta = 1$ och vikterna $w_1 = w_2 = 2/3$ göra att per-

ceptronen klassificerade alla inputs på det önskade sättet. En annan fråga är om man kan träna en enkel perceptron till denna lösning. Rosenblatts resultat säger just att man alltid *kan* göra det, eftersom det är en teoretiskt möjlig lösning.

Om man har lite logisk skolning inser man snabbt att det går att tänka sig inputvektorerna till vår mini-perceptron som *sanningsvärden* hos två satser, med 1 = S(ANN) och 0 = F(ALSK). En given klassifikation av de fyra möjliga inputmönstren kan då uppfattas som en *sanningsfunktion.* Exempelvis motsvaras den nyss nämnda klassifikationen, som ger utsignalen 1 för inputmönstret (1, 1) men 0 för övriga, av sanningsfunktionen ”OCH”. Satsen ”p och q” är ju sann, om och endast om både p och q är sanna, dvs. OCH är den sanningsfunktion, som tilldelar (S, S) värdet S men (S, F), (F, S) och (F, F) värdet F. Vårt lilla nätverk, tränat att utföra den ovan nämnda klassifikationen, är helt enkelt en logisk OCH-krets.

Det exempel som vi närmast skall titta på kan analogt uppfattas som sanningsfunktionen för ”uteslutande eller” (eXclusive OR, XOR). Det är den sanningsfunktion som tilldelar (S, F) och (F, S) värdet S, men som i övrigt ger värdet F. Satsen ”p uteslutande-eller q” kräver med andra ord för att vara sann, att en av de två delsatserna skall vara sann, *men inte båda*. I termer av klassifikation av mönster vill vi alltså att perceptronen ska ge utsignalen 1 för inputmönstren (1, 0) och (0, 1), men 0 för de två övriga. Vi ska nu ge två olika bevis för att detta är teoretiskt omöjligt.

Direkt bevis

Antag att det finns en tröskel θ och två vikter w_1 och w_2 som ger den önskade klassifikationen. Villkoret för att perceptronen ska ge utsignal 1 är, enligt ovan, att den viktade summan $\sum_{i=1}^{2} x_i w_i$ är större än θ. För de två inputmönstren (1, 0) och (0, 1), som ju ska ge denna utsignal, ger detta:

(6.1.3)
$$1 \cdot w_1 + 0 \cdot w_2 > \theta$$
$$\text{Alltså: } w_1 > \theta$$

respektive:

(6.1.4)
$$0 \cdot w_1 + 1 \cdot w_2 > \theta$$
$$\text{Alltså: } w_2 > \theta$$

Input (0, 0) ska ge output 0. Det innebär:

(6.1.5)
$$0 \cdot w_1 + 0 \cdot w_2 \leq \theta$$
Alltså: $\theta \geq 0$

För input (1,1), som vi vill ska ge output 0, får vi av 6.1.3–6.1.5:

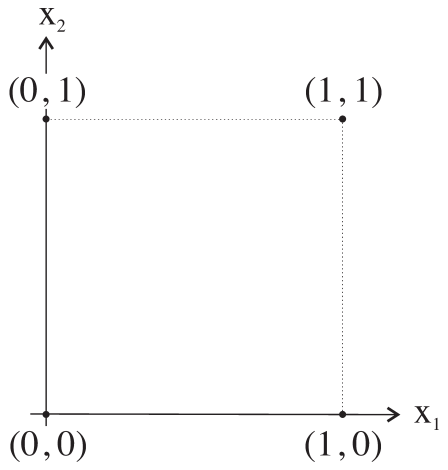
(6.1.6)
$$1 \cdot w_1 + 1 \cdot w_2 = w_1 + w_2 > \theta$$

vilket ger output 1, dvs. fel respons.

Vårt ursprungliga antagande var alltså felaktigt. Det går inte ens teoretiskt att hitta *någon* viktuppsättning och tröskel, som genomför den önskade klassifikationen. Givetvis kan man då inte heller träna perceptronen till att hitta en sådan!

Geometriskt bevis

Det geometriska beviset är mycket användbart eftersom man med dess hjälp kan införa de viktiga begreppen beslutsregion, beslutsgräns och linjär separerbarhet. Man utgår i detta bevis från en representation av inputvektorerna i ett cartesianskt koordinatsystem som har axlarna x1 och x2, dvs. aktivitetsnivåerna i de två inputnoderna. Inputs i XOR-problemet blir då fyra punkter på en kvadrat med sidan 1, enligt figur 38.



Figur 38. Inputrummet för XOR-problemet. x_i: aktiviteter i inputnoderna X_i. Markerade punkter: de fyra inputvektorerna.

Betrakta återigen perceptronens aktiveringsfunktion för 2 inputs, alltså

$$(6.1.1') \qquad \begin{aligned} &\sum_{i=1}^{2} x_i w_i > \theta \rightarrow y = 1 \\ &\sum_{i=1}^{2} x_i w_i \leq \theta \rightarrow y = 0 \end{aligned}$$

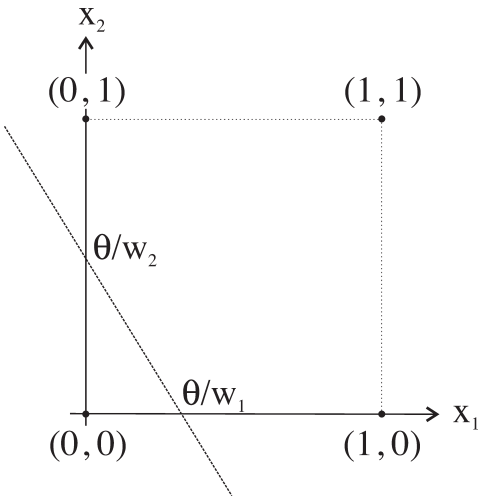
Givet två bestämda vikter w_1 och w_2 och en tröskel θ, avgränsar de här två villkoren var sin *beslutsregion* i x_1-x_2-planet, nämligen mängden av de inputs (x_1, x_2) som uppfyller den första respektive den andra olikheten. Gränsen mellan dessa två beslutsregioner, *beslutsgränsen,* definieras förstås av villkoret

$$(6.1.6) \qquad \begin{aligned} &\sum_{i=1}^{2} x_i w_i = \theta, \text{ eller:} \\ &x_1 w_1 + x_2 w_2 = \theta \end{aligned}$$

Hur ser nu den här beslutsgränsen ut geometriskt? Jo, skriver man den på formen

$$(6.1.7) \qquad x_2 = -x_1 \frac{w_1}{w_2} + \frac{\theta}{w_2}$$

inser man (om inte förr) att den är *en rät linje,* som har lutningen $-w_1/w_2$ och som skär x_2-axeln i punkten θ/w_2. På samma sätt inses att linjen skär x_1-axeln i punkten θ/w_1. Linjen kanske t.ex. ser ut ungefär så här:



Figur 39. En beslutsgräns för den enkla perceptronen med två binära inputs x_1 och x_2. Markerade punkter: de fyra inputvektorerna. w_j: vikter. θ: tröskel för outputnoden Y.

Poängen med det här framställningssättet är nu att man med hjälp av sin geometriska intuition lätt inser, att det inte *går* att dra beslutslinjen så att den löser XOR-problemet. För hur man än drar en rät linje linje så att (0, 1) och (1, 0) båda ligger på dess ena sida, så kan (0, 0) och (1, 1) uppenbarligen inte båda hamna på den andra sidan! Det finns med andra ord *inga* vikter w_1, w_2 som löser problemet.

Den enkla perceptronen med två inputnoder ger sammanfattningsvis alltid en rät linje som beslutsgräns, och om det inte går att dra en sådan linje mellan de två klasser som man vill separera, så kan inte perceptronen göra separationen ens med ”specialdesignade” vikter, än mindre genom inlärning. XOR-problemet är det enklaste exemplet på sådana *icke linjärt separerbara* klasser av mönster.

Begreppet *linjär separerbarhet* kan på ett självklart sätt generaliseras till mönster med fler dimensioner än två, och därmed till enkla perceptroner med fler inputnoder. I fallet med tre inputdimensioner motsvaras således den räta beslutslinjen av ett *beslutsplan*

$$x_1w_1 + x_2w_2 + x_3w_3 = \theta \qquad (6.1.8)$$

Allt som ligger på ena sidan om detta plan klassificeras av perceptronen ifråga som tillhörigt den ena klassen; allt på den andra sidan av planet

(eller på planet) hamnar i den andra klassen. Klasser som inte kan separeras av ett dylikt plan är inte linjärt separerbara, i den allmänna betydelsen av denna term, och kan alltså inte åtskiljas av en enkel perceptron.

Många klassifikationsproblem i levande livet är av icke-linjär karaktär, dvs. man behöver dra beslutsgränser som inte är linjer, plan eller hyperplan. Dessutom är det så att inte bara perceptroner, utan också flera klassiska matematiska metoder, enkelt klarar av alla linjärt separerbara problem. Det är alltså motiverat att undra vad man egentligen ska ha perceptroner till!

Det faktum att Rosenblatts enkla perceptroner inte kunde lösa icke linjärt separarerbara problem ledde också till att ledande företrädare för AI-forskningen på 1960-talet uttryckte starka tvivel på de artificiella neurala nätverkens möjligheter.[122] Visserligen hade redan Rosenblatt experimenterat med kraftfullare nätverk som inte hade samma principiella begränsning, utan som också (i princip) kunde dra mer komplicerade (icke-linjära) beslutsgränser, men han lyckades inte bevisa några viktiga konvergensresultat för dem. Han kunde med andra ord inte bevisa, att hans kraftfullare nätverk kunde tränas till att *hitta* de icke-linjära lösningar som de i princip var i stånd att leverera. Först när flera forskare på 1980-talet oberoende av varandra upptäckte ”back-propagation”-algoritmen för flerlagrade nätverk med kontinuerliga, sigmoida aktiveringsfunktioner i de dolda enheterna stod det klart att Rosenblatts intuitioner hade varit riktiga. De enkla perceptronerna skall ses som ett utvecklingssteg i riktning mot dessa mer komplicerade och verkligt användbara maskiner. Men de har inte bara ett historiskt intresse; de är också en naturlig pedagogisk inkörsport till ANN-området som helhet, och det är därför som vi ägnat dem så pass stor uppmärksamhet.[123]

---

[122] Minsky & Papert (1969).

[123] Till det utbildningspaket om ANN som finns tillgängligt vid Filosofiska institutionen vid Göteborgs Universitet hör en datormodell av den enkla perceptronen, som du kan ladda ner och köra på din egen dator (om du har en Macintosh eller kan emulera en, vill säga). Den har sju inputnoder och klassificerar LCD-siffror, kodade som sjusiffriga binära tal. Också i den illustrationen stöter du på icke linjärt separerbara problem, som alltså är analoga med XOR-problemet fast av högre dimensioner (beslutslinjer motsvaras av sexdimensionella besluts-hyperplan). Denna modell är ytterligare förenklad i och med att den varken har en justerbar tröskel eller en biasnod, och därför använder perceptronregeln i vår ursprungliga, enkla formulering (avsnitt 4.6). Detta gör att en inputvektor med alla komponenter = 0 inte kan hänföras till en godtyckligt bestämd klass, men förenklingen innebär inga andra principiella skillnader. Hämta modellen på http://www.phil.gu.se/kog/nerladd.htm.

Innan vi tar oss an de komplicerade, olinjära systemen ska vi titta på några nätverk som är "ännu mer linjära" än den enkla perceptronen.

6.2 Mönsterassociation, med mera, i linjära nätverk

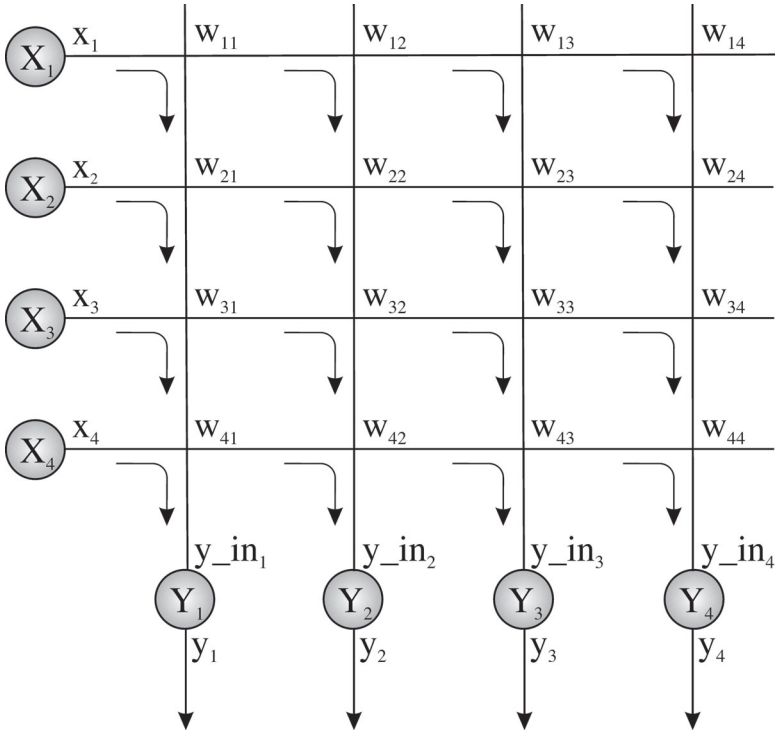
På 1960-talet studerade några forskare, inte minst amerikanerna Widrow och Hoff, lärande linjära system (de kallade dem för övrigt "adaptiva linjära filter") och gjorde upptäckter, som liksom Rosenblatts beskrivning av perceptronen är grundläggande för vår förståelse av alla neurala nätverk. Med ett helt linjärt nätverk menas i denna boks kontext ett artificiellt neuralt nätverk vars alla enheter har en linjär aktiveringsfunktion. I typfallet är denna funktion *identitet,* dvs. enheterna tar sin nettoinput som aktivitet. Vi ska se på tre närbesläktade nätverk av den sistnämnda typen – låt oss kalla dem "maximalt linjära". Det första lär sig vad man kan kalla *associationer mellan mönster* men kan ibland beskrivas som att det gör *koordinattransformationer*, det andra utför det som i klassisk statistik kallas *linjär regression*, medan det tredje gör så kallad *linjär diskrimination* vilket innebär att den sorterar mönster på ett sätt som liknar, men inte sammanfaller med, vad den enkla perceptronen gör. Gemensamt för alla nätverken är att de använder sig av deltaregeln för inlärning – också kallad "Widrow-Hoff-regeln". En outputenhet i ett dylikt nätverk kallas inte sällan "Adaline" (för ADAptive LInear NEurone).

I avsnittet ska vi också studera vad som händer om man använder Hebbregeln i ett linjärt nätverk för mönsterassociation, och i samband med detta introducera begreppet *autoassociation.*

Hela innehållet i avsnittet kan framställas mycket mer kortfattat än här, om man förutsätter goda kunskaper i linjär algebra hos läsaren. Det finns gott om sådana framställningar. Dock vänder sig denna bok också till en annan läsekrets, varför resonemangen nedan (särskilt de som rör deltaregeln) istället (liksom deltaregeln) går framåt i mycket små, men förhoppningsvis intuitivt fattbara steg.

Mönsterassociation

Låt oss ta ett maximalt linjärt ANN med lika många input- som outputnoder, exempelvis fyra, och illustrera det i ett Hinton-diagram. I figur 40 använder vi olika bokstäver för input- respektive outputneuron, och w_{ij} står därför för vikten mellan *inputelement* nr i och *outputelement* nr j.



Figur 40. En (maximalt) linjär associator. x_i: aktivitet i inputnod X_i. y_i: aktivitet i outputnod Y_i. w_{ij}: vikt från inputneuron X_i till outputnod Y_j.

Låt oss nu erinra oss kapitlet "Vektorer och matriser!". Eftersom nätverket är maximalt linjärt är aktiviteten y_j i en given outputenhet Y_j lika med dess nettoinput, vilken i sin tur är lika med inputvektorn **x** skalärt multiplicerad med outputenhetens viktvektor $\mathbf{w}_j$. Alltså:

(6.2.1) $$y_j = \mathbf{x} * \mathbf{w}_j$$

och hela outputvektorn **y** är lika med inputvektorn (betraktad som en (1, n)-matris) multiplicerad med viktmatrisen **W**,

(6.2.2) $$\mathbf{y} = \mathbf{xW}$$

Vi kan beskriva detta nätverk som att det "associerar mönster". Varje input är ju ett mönster, som i sin tur resulterar i ett outputmönster. Flera sådana associationer kan kodas in i nätverket oberoende av varann. Närmare bestämt kan ett linjärt nätverk med *n* inputnoder och *n* outputnoder i princip associera maximalt *n* st (i vårt exempel: fyra) inputmönster med *n* olika fritt valda outputmönster. Ett nödvändigt och tillräckligt villkor för

att detta ska vara möjligt är att inputmönstren är linjärt oberoende, vilket vi nu ska visa.

Linjärt oberoende hos inputvektorerna är givetvis ett *nödvändigt* villkor för att de skall kunna avbildas på fritt valda outputvektorer. Om en inputvektor är linjärt beroende av de andra måste ju output på den förstnämnda vektorn vara en linjär kombination av output på de andra, och kan således inte fritt bestämmas.

För att visa *tillräckligheten* av villkoret vill vi först ha ett *existensbevis,* dvs. vi behöver visa att det finns vikter med vilka nätverket faktiskt skulle utföra associationer av den önskade typen. Men detta följer av det vi ovan sagt om koordinattransformationer (avsnitt 5.2). En uppsättning $\mathbf{M}_1$ av n linjärt oberoende vektorer med n komponenter kan alltid överföras till en godtycklig annan sådan, $\mathbf{M}_2$, genom transformationsmatrisen $\mathbf{T} = \mathbf{M}_2\mathbf{M}_1^{-1}$. Detta uttryck anger således den önskade viktmatrisen. Är bara inputvektorerna, men inte outputvektorerna, linjärt oberoende finns det fortfarande en transformation av de förra till de senare men inte någon invers transformation, alltså i omvänd riktning.

Lägg märke till att detta bevis för existensen av en lösning också anvisar en direkt metod för att *hitta* lösningen – nämligen med hjälp av linjär algebra. Inversa matriser kan alltid beräknas i denna algebra. Artificiella neurala nätverk är alltså inte på något sätt nödvändiga för att vi ska kunna hitta transformationsmatrisen ifråga. Denna observation kan generaliseras till följande påstående: *för linjära problem behöver vi teoretiskt sett aldrig använda neurala nätverk, utan mer klassiska, mer direkta metoder är tillämpbara*. En praktisk fördel med ANN är dock att de anger enkla, stegvisa algoritmer för att hitta lösningar av linjära problem; matrisinvertering är svårare.

Även ur flera andra synvinklar är det intressant att undersöka om ANN kan lösa linjära problem, och hur de i så fall gör det. För det första är en sådan diskussion bra för förståelsen av hur andra artificiella neurala nätverk löser *icke-linjära* problem, där motsvarande klassiska metoder inte fungerar bra (se kapitel 9). För det andra är det mer sannolikt att *biologiska* neurala nätverk använder stegvis uppgradering av vikter, än att de direkt beräknar inversa matriser enligt den linjära algebrans algoritmer.

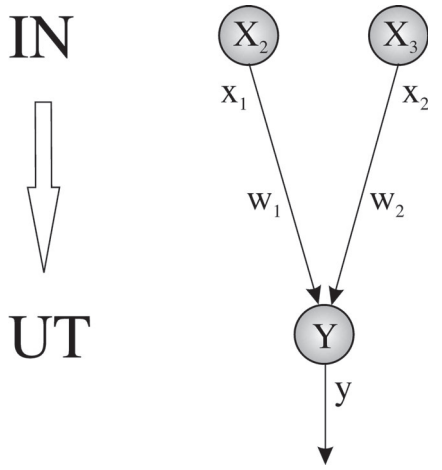
Låt oss, sålunda lugnade, återgå till frågan om ett linjärt ANN kan hitta de vikter som åstadkommer den önskade associationen mellan input- och

outputvektorer. Vi vet alltså att om inputvektorerna är linjärt oberoende, så måste det i princip finnas en matris av vikter som åstadkommer de önskade associationerna till lika många outputvektorer, men vi har inte visat att nätverket kan hitta dessa vikter genom inlärning. Vi ska därför skissera ett *konvergensbevis* med innebörden att deltaregeln verkligen leder fram till lösningen. Vi har tidigare (avsnittet om inlärningsalgoritmer, 4.6) sett hur deltaregeln fungerar i ett minimalt linjärt nätverk bestående av två neuron, men ska nu anlägga ett lite allmännare perspektiv. För att resonemanget inte ska bli för abstrakt ska vi dock gå omvägen om ännu ett exempel.

Deltaregelns konvergens i helt linjära nätverk

När ett nätverk tränas till ett visst, önskat beteende (till exempel en viss mönsterklassifikations- eller mönsterassociationsuppgift) brukar man ofta beskriva dess fortlöpande framsteg i termer av värdet hos en *felfunktion.* Felfunktionen ger ett globalt mått på avvikelsen mellan nätverkets önskade respektive verkliga prestation, och kan som sådant mått definieras på olika sätt. Den beror dock alltid av vilka vikter som nätverket för tillfället har, och kan alltså ses som en funktion $E(\mathbf{w})$ av dessa vikter. Ett bevis för att en inlärningsalgoritm hittar en lösning av klassifikationsuppgiften kan då formuleras som ett bevis för att vikterna, när algoritmen tillämpas, så småningom får (eller åtminstone asymptotiskt närmar sig) en uppsättning $\mathbf{w}$ av värden för vilka $E(\mathbf{w}) = 0$. Till yttermera visso använder sig inlärningsalgoritmerna själva gärna av felfunktionen, vilket vi strax ska illustrera.

Ett vanligt sätt att definiera felfunktionen är i termer av de *kvadrerade skillnaderna* $(d_j - o_j)^2$ mellan önskad output d_j och verklig output o_j (i de olika outputnoderna). Vi skall tills vidare hålla oss till denna typ av felfunktioner (för en annan viktig typ se avsnitt 9.3). Kvadreringen gör att felet aldrig blir mindre än noll. Betrakta således följande lilla, maximalt linjära associator som har en enda outputenhet Y:



Figur 41. En mini-associator. x_i: aktivitet i inputnoderna X_i. y: aktivitet i outputnoden Y. w_i: vikter.

och där $y = w_1 \cdot x_1 + w_2 \cdot x_2$. Antag att vi gett den uppgiften att associera fyra binära inputvektorer med output 1 eller 0 på följande sätt:

(6.2.3) $\quad (0, 0) \rightarrow 0;\ (1, 0) \rightarrow 0;\ (0, 1) \rightarrow 1;\ (1, 1) \rightarrow 1$

(Uppgiften är lätt: $w_1 = 0$ och $w_2 = 1$ är en uppenbar lösning.)

Nåväl, till felfunktionen. Med vikter w_1 respektive w_2 blir nätverkets respons på (1, 1) = $w_1 + w_2$, och det kvadrerade felet för dessa vikter och just denna inputvektor blir $(1 - (w_1 + w_2))^2$. På samma sätt kan vi räkna ut det kvadrerade felet för de andra tre inputvektorerna i termer av $w_1 + w_2$ (gör det!).

Vad vi är intresserade av är ju att nätverket skall göra rätt för *alla* inputvektorer, så för att få den mest relevanta felfunktionen skall vi lägga samman felen för alla inputs till ett *globalt* fel.[124] Det är denna globala funktion som man vill skall anta värdet 0. I detta fall inser man lätt att den globala felfunktionen blir:

(6.2.4) $\quad E(\mathbf{w}) = 0 + (0 - w_1)^2 + (1 - w_2)^2 + (1 - (w_1 + w_2))^2$

vilket efter uträkning av parenteserna blir

[124] I ANN-sammanhang brukar man också multiplicera med ½ för att få en formell likhet med definitioner av energi i fysiken, men det gör ingen skillnad för resonemangen att vi inte gör så.

(6.2.5) $E(\mathbf{w}) = 2 + 2\,w_1^{\,2} + 2\,w_2^{\,2} + 2\,w_1w_2 - 2\,w_1 - 4\,w_2$

Här har vi alltså det explicita uttrycket för hur det totala felet beror av vikterna i vårt lilla nätverk.

Innan vi går vidare och tittar på vad detta betyder för vår mini-associator ska vi se på en mer generell formulering av felfunktionen E. I många nätverk, exempelvis i den (4, 4)-mönsterassociator som vi beskrev nyss, har man fler än ett outputneuron och man vill förstås att det ska bli ”rätt” respons i dem alla. När man beräknar felet E_k för en given inputvektor $\mathbf{i}_k$ börjar man därför med att räkna ut det ”lokala” fel E_{kj} som den ger i outputnod nr j:

(6.2.6) $$E_{kj} = (d_{kj} - y_{kj})^2$$

Dubbelindexet $_{kj}$ i E_{kj} och y_{kj} markerar alltså att det handlar om inputvektor nr k och outputkomponent nr j. Därefter lägger man ihop felen i de olika outputnoderna. Låt oss säga att de är m st stycken:

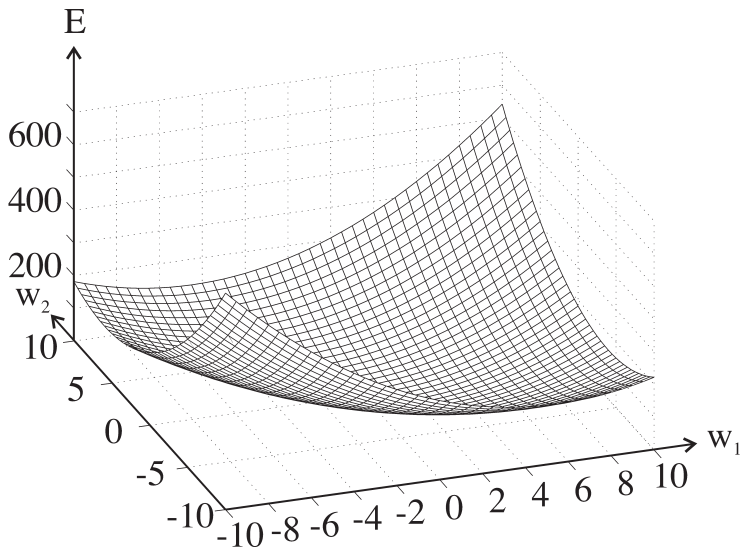
(6.2.7) $$E_k = \sum_{j=1}^{m} E_{kj} = \sum_{j=1}^{m} (d_{kj} - y_{kj})^2$$

Därefter måste vi för att få det totala felet E summera över alla de (säg) p stycken olika inputvektorerna, precis som vi gjorde nyss med mini-nätet. I det allmänna fallet får vi:

(6.2.8) $$E = \sum_{k=1}^{p} E_k = \sum_{k=1}^{p}\sum_{j=1}^{m} (d_{kj} - y_{kj})^2$$

Även i detta allmänna fall är E förstås en funktion av vikterna i nätverket, och felet är fortfarande alltid positivt eller 0 eftersom det är en summa av kvadratiska uttryck. Det explicita uttrycket för E kan vara mycket mer komplicerat än vad som är fallet för vår mini-associator. *Dock är E i ett linjärt nätverk alltid en kvadratisk funktion av vikterna.* $E(w_1, w_2)$ för mini-associatorn innehåller inga w_1- eller w_2-termer av tredje graden eller högre; motsvarande gäller, vilket lätt inses, oavsett hur många vikter nätverket innehåller.

Låt oss nu ta en titt på hur felfunktionen (6.2.5) för mini-associatorn ser ut i ett 3D-diagram med vikterna på två axlar och felet på den tredje:



Figur 42. Felfunktionen för mini-associatorn. w_i: vikter; E: totalt fel.

E är en icke-negativ kvadratisk funktion av två variabler och kan som sådan bara ha *ett* minimum. I diagrammet ser man tydligt att *felytan* är formad som en ”hängmatta” med en entydigt bestämd lägsta punkt. Det är kanske lite svårare att se exakt var denna är belägen. Minimum infaller faktiskt i (0, 1) och där har felet värdet 0, eftersom alla verkliga outputs här sammanfaller med de önskade (jämför 6.2.3 ovan).

Bilden ger oss en idé om hur en algoritm skulle kunna hitta till minimum för E från vilken punkt som helst på felytan: på grund av ytans enkla form är det bara att gå raka vägen neråt, så kommer man så småningom till den lägsta punkten.

Men vad är det egentligen, matematiskt sett, att ta ett steg i riktning ”raka vägen ner”? Jo, det är att gå en liten bit i w_1-led och en liten bit i w_2-led, *i proportion till hur mycket ytan lutar neråt i respektive led.* Formellt uttryckt betyder detta, att man minskar w_1 i proportion till den partiella derivatan av E med avseende på w_1, och likadant för w_2. Den partiella derivatan $\partial E/\partial w_1$ är ju ingenting annat än måttet på hur fort E ökar när man ökar w_1 (och håller allt annat konstant), dvs. just lutningen i w_1-led. Om man följer denna strategi och ser till att ta mindre och mindre steg, kommer man att gradvis närma sig den lägsta punkten.

Det här resonemanget om att följa felytans lutning neråt – den engelska termen är *gradient descent* och vi ska tala om *gradientnedstigning* – är i själva verket mycket generellt och är giltigt såväl för inlärning i mer komplexa neurala nätverk som för många andra optimeringsalgoritmer. Det viktigaste tillägget som man måste göra beträffande flerlagrade, ickelinjära nätverk (förutom givetvis att felytan har högre dimension än 2 eftersom nätverken har fler än två vikter) är, att felytan för dem i allmänhet inte har bara *ett* minimum utan istället flera ”gropar” av olika djup; de flesta av dessa karakteriseras alltså av att $E > 0$. Det betyder att man vid tillämpandet av gradientnedstigning riskerar att hamna i s.k. lokala minima, varvid man missar den rätta lösningen (om en sådan finns). För att undvika att fastna i lokala minima och vara någorlunda säker på att man hamnar i den djupaste gropen (det globala minimum för felfunktionen) får man ta till speciella knep, varav det enklaste är upprepade körningar med olika, slumpvis valda startvärden på vikterna. Det har också utvecklats många metoder som hittar minima betydligt *snabbare* än det enkla gradientnedstigande som vi beskriver här; se vidare avsnitt 9.3.

Låt oss nu fortsätta med det enkla fall då felytan har ett enda minimum, vilket alltså alltid är fallet med en linjär associator. Här kan vi, visar det sig, ganska lätt härleda explicita uttryck för de partiella derivatorna av E med avseende på de enskilda vikterna. Vi ska inte göra detta utifrån ekvation (6.2.5) som karakteriserar just vårt mini-nät, eftersom det är betydligt mer instruktivt att betrakta en godtycklig, maximalt linjär associator. (Generaliseringen till alla helt linjära associatorer är trivial.)

Hur ändrar sig E_k, felet i hela output för inputvektor nr k, om man ökar en viss vikt w_{ij} i en maximalt linjär associator med storheten Δw_{ij} (som kan vara negativ)? Notera att vikten w_{ij} är styrkan hos förbindelsen mellan inputenhet X_i och outputenhet Y_j. Vad som händer när man lägger till Δw_{ij} är därför att aktiviteten i Y_j ökar med $x_{ki}\,\Delta w_{ij}$, där x_{ki} är aktiviteten i X_i. (Dubbelindexet $_{ki}$ markerar som tidigare att det handlar om vad som händer under input nr k.) Den partiella derivatan $\partial y_{kj}/\partial w_{ij}$ av *outputaktiviteten* i Y_j med avseende på vikten w_{ij} är således gränsvärdet, när Δw_{ij} går mot 0, för $x_{ki}\,\Delta w_{ij}/\Delta w_{ij}$, vilket uppenbarligen är $= x_{ki}$. Ingen aktivitet i någon annan outputnod påverkas av en ändring i w_{ij}. Derivatan $\partial E_k/\partial w_{ij}$, dvs. av *hela outputfelet* med avseende på w_{ij} men betraktat bara för den aktuella inputvektorn $\mathbf{i}_k$, kan därför beräknas utifrån vad som händer med felet i noden Y_j. Detta fel är ju definierat som $(d_{kj} - y_{kj})^2$ (där d_{kj} = önskad output i Y_j för input nr k), och med hjälp av kedjeregeln för derivator får vi:

(6.2.9) $$\partial E_k / \partial w_{ij} = 2x_{ki}(y_{kj} - d_{kj})$$

Motsvarande gäller förstås för alla andra vikter.

Den flitiga läsaren associerar nu genast till deltaregeln, och det gör hon rätt i! Resultatet antyder att om vi ska gå ”raka vägen neråt” i outputfelets landskap, så ska vi ändra alla w_{ij} i omvänd proportion till respektive produkt $x_{ki}(y_{kj} - d_{kj})$.

Vi har dock ännu så länge bara räknat på felet för en enstaka input. Det *globala* felets derivata med avseende på w_{ij}, är derivatan av *summan* av alla fel under de olika inputs. Men detta är, inser man snabbt, inget annat än summan av de olika derivatorna $\partial E_k/\partial w_{ij}$ över alla p stycken inputs $\mathbf{i}_k$, eller med andra ord:

(6.2.10) $$\partial E / \partial w_{ij} = \sum_{k=1}^{p} 2x_{ki}(y_{kj} - d_{kj})$$

För att gå raka vägen ner mot felminimum ska man alltså räkna ut denna summa, och ändra w_{ij} i omvänd proportion till den.

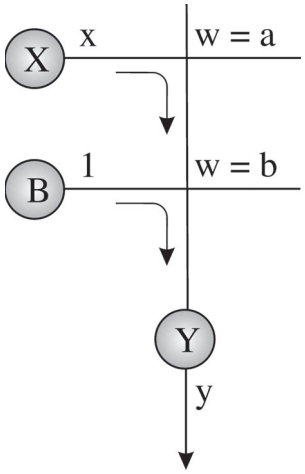
Det kan dock visas (vilket vi inte ska göra här), att man har rätt att ändra vikter i enlighet med felderivatan beräknad för varje *enskild* presentation av en input, så länge inlärningsfaktorn progressivt minskas och går mot noll under träningen. Denna procedur, som är identisk med deltaregeln som vi formulerat den ovan, leder också garanterat till att man uppnår felminimum; däremot följer den inte alltid raka vägen dit.

Motsvarande möjlighet till *on-line-inlärning*, alltså felkorrektion grundad på enskilda inputs bidrag till felet, finns i flera andra inlärningsalgoritmer. Vill man gardera sig bättre mot fel som beror på denna approximation, och/eller om man vill ha en jämnare nedstigning mot felminimum, väljer man istället att låta algoritmen titta på ett stort antal inputs innan den beräknar sina derivator – så kallad *batch-inlärning*.

Linjär regression med neurala nätverk

Vi har hittills koncentrerat oss på det förhållandet, att ett linjärt nätverk med deltaregeln som inlärningsalgoritm under vissa villkor kan exakt avbilda de givna inputvektorerna på önskade outputvektorer med samma di-

mensionalitet. Nu ska vi istället titta närmare på ett maximalt linjärt nätverk där outputvektorn bara har en komponent.



Figur 43. Ett nätverk för linjär regression. B: biasnod. w: träningsbara vikter.

I det maximalt linjära nätverket i figur 43 är B ett biaselement med konstant aktivitet 1. Det andra inputelementet, X, har däremot en variabel aktivitet x. Vikterna från noderna X och B till outputnoden Y är a respektive b. Aktiviteten hos noden Y blir därför alltid = ax + b.

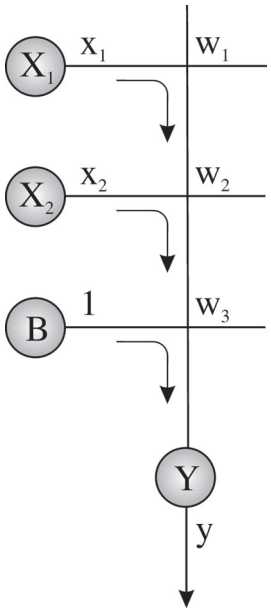
Antag att det föreligger en mängd av k talpar $<x_k, d_k>$. Träna nätverket med deltaregeln på alla inputvektorer av typen $(x_k, 1)$, med motsvarande d_k som önskad output i varje fall. Detta innebär ju, som vi visat ovan, att summan av de kvadrerade skillnaderna mellan nätverkets verkliga outputs $y_k = ax_k + b$ och dessa d_k så småningom kommer att minimeras. Detta är i sin tur likvärdigt med att nätverkets vikter asymptotiskt närmar sig värden a_0 och b_0, sådana att ekvationen $y = a_0x + b_0$ är den *linjära regressionsekvationen* för datamängden i fråga.

Felet är noll och lösningen av uppgiften exakt om och endast om alla talpar $<x_i, d_i>$ råkar ligga exakt på denna linje. Men även om de inte gör det, så finns det goda skäl att klassificera resultatet som den i en mening ”bästa” linjära lösningen av problemet att *anpassa en kontinuerlig kurva till en given mängd data*, eller (i en annan tolkning) av problemet att hitta ett *troligt underliggande matematiskt samband* mellan två variabler. Den linjära regressionsekvationen är nämligen, under förutsättningen att data härstammar från en linjär process y = ax + b plus ett normalfördelat

slumpfel med väntevärdet 0, en s.k. väntevärdesriktig skattning av parametrarna a och b.[125] En tredje term för vad nätverket gör är (linjär) *funktionsapproximation* från givna data.

Linjär diskriminering med linjära neurala nätverk

Ett helt linjärt nätverk kan också användas för övervakad mönsterklassifikation. Det kan naturligtvis finnas fler outputklasser, men låt oss koncentrera oss på det enklaste fallet med två outputklasser representerade i en outputnod, en tvådimensionell input och en biasnod. Nätverket i figur 44 är bipolärt, och varje input tilldelas en önskad output –1 eller +1 som representerar dess önskade klasstillhörighet. Vi tränar som förut nätverket med deltaregeln.



Figur 44. Ett linjärt nätverk för mönsterklassifikation. B: biasnod. w_i: träningsbara vikter.

Nu vet vi att alla möjliga input-outputfunktioner i detta nätverk är linjära, dvs. för varje given uppsättning vikter $<w_1, w_2>$ beskriver nätverksfunktionen ett plan i det rum som bestäms av x_1, x_2 och y:

(6.2.11) $\quad y = x_1 w_1 + x_2 w_2 + w_3$

[125] Detta blir trivialt sant om man, vilket är vanligt, *definierar* regressionsekvationen i termer av väntevärden för den process som data är ett stickprov från. Men eftersom vi istället har definierat den som resultatet av en minsta-kvadrat-anpassning till data är påståendet inte trivialt.

Naturligtvis kan ett sådant plan endast i undantagsfall lyckas med att tilldela rätt y-värden till alla inputs, eftersom funktionsvärdet y ändras kontinuerligt med x_1 och x_2 och det egentligen bara är y-värdena 1 eller –1 vi vill ha. Men vi kan välja att tolka nätverkets output så, att ett värde ≥ 0 står för den ena klassen och ett värde < 0 för den andra. Geometriskt betyder det en beslutslinje där planet i (6.2.11) skär x_1x_2-planet. Nätverket är därför mycket likt den enkla perceptronen i sitt funktionssätt, men en viktig skillnad är att outputfelet beräknats *innan* beslutströskeln appliceras. Det går att visa att den resulterande beslutslinjen, under vissa förutsättningar om den underliggande processens natur, minimerar risken för felklassifikation. Jämför också avsnitt 9.2.

De kontinuerliga värdena hos outputneuronen är inte heller ointressanta, eftersom de i viss mån avspeglar hur sannolikt det är att input hör till den ena respektive den andra klassen. För att bokstavligen kunna tolka output som en skattning av dessa sannolikheter måste man ersätta de linjära outputneuronen med logistiska enheter och använda en annan felfunktion (ömsesidig entropi). Detta ändrar dock inte beslutsgränsen, som alltså fortfarande är linjär. För fallet med flera klasser se avsnitt 9.3.

Hebbregeln i linjära system

Under vissa snävt definierade betingelser kan helt linjära system lära sig att associera mönster perfekt genom Hebb-regeln. I detta fall matar man under träningen in ett inputmönster i inputenheterna *och den önskade output i outputneuronen*, och låter Hebb-mekanismen verka. Man testar sedan nätverkets prestation genom att låta det producera output från input på normalt sätt. Under träningen växer, enkelt uttryckt, vikterna mellan de par av noder som är samtidigt aktiva när input och önskad output presenteras. Så småningom kommer därför inputmönstret att vid testningen kunna producera det motsvarande outputmönstret, förutsatt att inlärningskoefficienten i Hebbregeln har valts på ett lämpligt sätt (jämför nedan).

Denna form av inlärning brukar klassificeras som “oövervakad”, (engelska: *unsupervised*), eftersom inlärningsalgoritmen – till skillnad från exempelvis deltaregeln – inte kräver annan information än den som faktiskt är kodad i nätverkets aktiviteter och vikter. Däremot klassificeras den ofta som ”styrd”, och man syftar då på att användaren av nätverket på ett ”onaturligt” sätt matar in den önskade output i outputenheterna.

Man kan dock tänka sig ett arrangemang liknande det Pavlovska – jämför figur 1 i avsnitt 1.1 ovan – där denna ”onaturliga” inmatning får en rimlig biologisk tolkning. I så fall förvandlas den styrda inlärningen till ”icke styrd”, precis som tillägget av en subtraktiv feedbackkrets förvandlar övervakad inlärning med deltaregeln till oövervakad inlärning med Hebb-regeln. Jämför avsnitt 4.6.

I vår tidigare mini-modell av Pavlovsk betingning (som ju innehåller en tröskelenhet) är det helt uppenbart att träning med Hebbregeln kan ge det önskade resultatet. Om man istället använder motsvarande maximalt linjära mini-modell och ser till att inlärningskoefficienten i Hebbregeln = 1/n, där n är ett heltal, så kommer (lika uppenbart) output att bli = 1 efter inlärningssteg nummer n. Det är inte fullt lika självklart att det går att associera *vektorer* med varann på detta sätt, och i vilken omfattning det i så fall är möjligt. Man kan visa (men vi ska inte göra det här) att linjära nätverk som följer Hebbregeln och har alla vikter från början = 0 kan lära sig att associera *ortogonala* inputvektorer (men inte godtyckliga, linjärt oberoende vektorer) med fritt valda outputvektorer.

Tyvärr blir helt linjära Hebb-nätverk snabbt överträngade. Eftersom vikterna fortsätter att öka även *efter* det att output uppnått det önskade resultatet, så kommer output att återgå till att bli “fel” (fast åt andra hållet). För att komma till biologiskt mer realistiska modeller måste man *antingen* frångå antagandet om linjära aktiveringsfunktioner, *eller* välja någon variant av Hebb-regeln där vikterna inte växer obegränsat.

Korrelationsanalys med Hebb-nätverk

Ett helt linjärt Hebb-nätverk som tränas med input- respektive outputvektorer från två givna datamängder kan alltså ibland exakt avspegla sambandet mellan inputvektorerna och outputvektorerna. Men även när det inte lyckas med denna uppgift gör det något intressant. Vikterna avspeglar nämligen – redan efter *en* träningsrunda med hela datamängden – *korrelationerna* mellan komponenterna i input- respektive outputdata. Detta är inte särskilt märkligt i ljuset av att vikten på en förbindelse alltid ökar i proportion till aktiviteten hos båda de neuron som den förbinder. Linjära system med den ursprungliga Hebbregeln är visserligen inte perfekta avbilder av statistisk korrelationsanalys eftersom vikterna växer obegränsat, men detta kan avhjälpas genom en lämplig normalisering av vikterna.

Linjära nätverk med Hebbregler som modifierats på andra sätt har exakta motsvarigheter i andra kända statistiska metoder. Exempelvis kan nätverk som tillämpar sådana regler utföra en form av statistisk databearbetning som kallas *principalkomponentanalys* (PCA).[126] Vi ska dock inte gå in närmare in på denna eller andra liknande motsvarigheter här.

Auto-association och auto-korrelation i linjära system

Om man tränar ett maximalt linjärt (n, n)-nätverk med Hebbregeln enligt ovan, och för varje inputvektor väljer *samma* vektor som output, kommer nätverket att försöka associera var och en av dessa vektorer *med sig själv*. Om inputvektorerna som man tränar med dessutom är högst n st och ortogonala mot varandra, kommer nätverket efter träningen att som output på en given inputvektor ge precis samma vektor. Detta är *autoassociation*.

Observera att denna auto-association innebär inte bara att mönster associerats med sig själva, utan också att de olika *komponenterna* i varje enskilt mönster associerats *med varandra*. Eftersom vi använt Hebbregeln kommer vikterna närmare bestämt att avspegla korrelationen (i datamängden) mellan de olika komponenterna i inputvektorn. Man talar därför om *autokorrelation.*

Distribuerad kodning, parallellprocessande och feltolerans

Autokorrelationen mellan komponenter i ett Hebb-nätverk betyder i sin tur, att alla komponenter i inputvektorn (i den mån de är skilda från 0) kommer att bidra till aktiviteten i alla outputnoderna. Motsvarande förhållande gäller uppenbarligen även då output- och inputdata inte är identiska, dvs. vid *hetero-association.* Det innebär, att ett auto- eller hetero-associativt Hebbnätverk använder sig av *distribuerad representation* (närmare bestämt av *integrerad* representation, se avsnitt 1.3 ovan). Detsamma gäller förstås nätverk som tränats med deltaregeln, men för enkelhets skull ska vi här begränsa det följande resonemanget till sådana som använder Hebbregeln.

I en linjär Hebb-associator är, som just nämnts, *varje* komponent i input i princip relevant för att *varje* komponent i output blir vad den blir. Detta betyder i sin tur bland annat, att om man ger ett inputmönster **i**' som avviker från ett av de inträngade, **i**, med avseende på bara *en* komponent, så

[126] Jfr. t.ex. Principe et al. (2000), avsn. 6.3–6.4.

får man som regel ett outputmönster som avviker från output på **i** med avseende på *alla* komponenterna, men i mindre grad. Ett enkelt exempel är när man autoassocierat vektorn (1, 1, 1, 1) med samma vektor genom en tillämpning av Hebbregeln med inlärningsfaktorn 0,25 (men inte tränat nätverket på andra vektorer), och sedan ger (1, 0, 1, 1) som input. Output blir då (3/4, 3/4, 3/4, 3/4), eftersom alla vikter i nätverket är = 0,25. Detta fenomen är alltså en manifestation av det som kallas *distribuerad representation.* Termen står, som nämnts redan i avsnitt 1.3, för det förhållandet att ett nätverk ofta ”sprider ut” information, så att denna från att kodas i *en* enhet kommer att förmedlas av *flera* enheter.[127]

Existensen av distribuerad representation tas inte sällan till intäkt för påståendet, att neurala nätverk *automatiskt är toleranta mot fel i input*. Man kan ju tycka, att en liten förändring av alla komponenterna i ett mönster är en *mindre* förändring än en stor ändring av en komponent, och att t.ex. mönstret (3/4, 3/4, 3/4, 3/4) därför är *mer likt* mönstret (1, 1, 1, 1) än vad mönstret (1, 0, 1, 1) är. Nätverket skulle alltså på något sätt ha ”kompenserat” för inputfelet. Men detta är helt och hållet en fråga om vem som använder output, och hur. En mänsklig betraktare har kanske en tendens att bedöma likhet mellan vektorer på det antydda sättet. Men en maskin – till exempel det neurala nätverk som använder det första nätverkets output som sin input – kan ”se saken mer objektivt”, och behöver inte ha några principiella svårigheter med att hantera informationen i den ena formen snarare än den andra.

En källa till missförstånd är säkert att man tenderar att förväxla distribuerad representation med *redundans*, dvs. enkelt uttryckt det förhållandet att samma meddelande sänds på olika vägar samtidigt. Redundans innebär förvisso feltolerans, jämför åter avsnitt 1.3. Termen ”parallellprocessande” täcker tyvärr ofta både distribuerad representation och redundant informationsbehandling, och kan därför ytterligare bidra till förvirringen.[128]

Det finns emellertid en viktig mening, i vilken ett enkelt associativt nätverk *också* kan vara *feltolerant*. Antag att vi förser outputenheterna i vårt lilla (4, 4)-Hebbnätverk med en tröskelfunktion, sådan att aktiviteten = 1

[127] ”Information” och ”kodning” har här en strikt kausal innebörd och skall inte tolkas kognitivistiskt. Jämför avsnitt 1.3.

[128] Beteckningen ”parallel distributed processing” blev populär i ANN-sammanhang efter publicerandet av ett mycket inflytelserikt arbete: Rumelhardt & McClelland (red.) (1986).

om och endast om nettoinput är minst 1. Vi lär det sedan att associera (1, 1, 1, 1) med samma vektor, och för säkerhets skull *tränar vi det två gånger*. Givet inlärningskonstanten 0,25 kommer då alla vikter i nätverket att bli = 0,5, och svaret såväl på (1, 1, 1, 1) som på (1, 0, 1, 1) kommer att bli (1, 1, 1, 1)!

Vi ska tala mycket mer om denna form av feltolerans i avsnittet om Hopfieldnätet (7.2), men det finns anledning att redan nu påpeka följande. Man måste fortfarande noga skilja mellan beteendet hos vårt sista, ”feltoleranta” Hebbnätverk och det som ett redundant nätverk uppvisar. Feltoleransen hos Hebbnätverket har nämligen köpts till priset att information om *andra* inputs än den som nätet tränats på nu *kastas bort*. Detta är i sin tur egentligen bara en annan sida av att nätverket är ett system *med minne*, i den tekniska betydelsen av ”minne”: det tar i sin respons hänsyn inte bara till vilken den aktuella input är, utan också till vilka inputs den fått tidigare. Och detta betyder inte bara att vissa outputs ”gynnas” (feltolerans), utan naturligtvis också att andra ”missgynnas” (bortkastad information). Det finns kanske fria luncher, men inget gratis minne.

6.3 ”Linjära” ANN i vid mening

Linjära nätverk och linjära modeller

Med “linjära neurala nätverk” har vi ju avsett nätverk vars alla enheter har en linjär aktiveringsfunktion. Av denna egenskap följer på den positiva sidan att felfunktionens minimum är unikt och lätt att hitta; på den negativa sidan såg vi en stor begränsning i nätverkens prestationsförmåga som kunde sammanfattas i att de bara kan modellera linjära funktioner. Vi ska nu visa hur man kan få de neurala nätverken att modellera icke-linjära funktioner utan att man går miste om den egenskap som består i att felfunktionen har ett enda minimum. Det är nämligen så, att förekomsten av icke-linjära neuron på input- eller outputsidan inte *behöver* förstöra denna egenskap. Det viktiga är att felfunktionen har kvadratisk form. I statistikteorin är detta inte sällan ett kriterium på en “linjär modell”, eftersom det innebär att man kan hitta minimum med linjär algebra. En annan anledning att kalla de här nätverken ”linjära” i en vid mening är att de, trots allt, har begränsningar som påminner starkt om de helt linjära nätverkens. Vi ska nu precisera de här påståendena lite närmare.

Fixa transformationer av inputdata (”förprocessande”)

Antag att vi ersätter inputskiktet i ett linjärt nätverk med ett skikt av icke-linjära noder, som inputdata ska passera *via förbindelser med fixa vikter.* Detta ”förprocessande” lager kommer att transformera input I till en ny signal I’. Alla resonemang som vi tidigare fört kring linjära nätverk gäller givetvis fortfarande, så länge vi betraktar de *transformerade* data som input. Exempelvis kan nätverket nu modellera outputs linjära beroende av I’, om det finns ett sådant beroende, och vi kommer fortfarande att garanterat hitta regressionslinjen för output i förhållande till I’ med hjälp av deltaregeln. Men ett linjärt beroende mellan I’ och output innebär ju ett *icke-linjärt* beroende *mellan I och output*! Har vi alltså på detta enkla vis utökat nätverkets prestationsförmåga från att bara gälla linjära samband till att även gälla icke-linjära sådana?

Läsaren inser nog att det inte är så. Vad som har hänt är istället, att nätverkets förmåga att hitta linjära samband *ersatts* med en förmåga att hitta *en mycket speciell klass* av icke-linjära samband. Den viktiga reservationen att förbindelserna till de icke-linjära enheterna på inputsidan ska ha *fixa vikter* hindrar nämligen inlärningsalgoritmen från att leta efter *andra* transformationer av inputdata än den givna. Därför kan det bara realisera ett fåtal av alla möjliga icke-linjära funktioner. Detta innebär i allmänhet också, att det inte längre kan hitta *linjära* samband mellan den *ursprungliga* input och output!

Så om man bara ser till hur omfattande klassen av samband är som nätverket kan hitta, så har ingenting vunnits. Men det finns en annan aspekt som är lika viktig, och det är att vi kanske är ute efter en *speciell* typ av funktioner och vill anpassa metoden efter den typen. I denna situation kan en fix, icke-linjär transformation av inputdata förvandla ett tidigare svårlöst problem till ett enkelt, linjärt problem. Det finns ett otal exempel på denna sanning, som är ett specialfall av den viktiga regeln: *se alltid till att förbehandla data på bästa sätt!*

Icke-linjära transformationer av outputdata

Om man ger outputenheterna i ett helt linjärt nätverk en monotont växande, icke-linjär men kontinuerlig aktiveringsfunktion, så kommer felet fortfarande att vara en kvadratisk funktion *av vikterna* (om än lite mer komplicerad än i vanliga linjära nätverk), och felminimum kommer fortfarande att vara unikt och lätt att hitta. Signalbehandlingsegenskaperna

ändras återigen kvalitativt snarare än kvantitativt, dvs. nätverken kan inte lösa en *vidare klass* av problem än förut men däremot, med rätt val av outputenheter, *andra* problem än förut.

En intressant kvalitativ vinst finns att hämta när det gäller nätverk för klassifikation. Om man nämligen väljer *logistiska* outputneuron till ett i övrigt linjärt nätverk (det får också gärna innehålla en fix, icke-linjär inputtransformation enligt ovan) kommer man att kunna tolka output som en bästa skattning, givet modellens begränsningar i övrigt, av *sannolikheterna* för inputs tillhörighet till de olika klasserna. Detta förutsätter att man arbetar med en annan felfunktion än det vanliga summa-kvadratfelet, nämligen *ömsesidig entropi* (cross-entropy). Metoden kan utvidgas till att användas på icke-linjära ANN i egentlig mening, dvs. sådana som har icke-linjära neuron med *adaptiva* vikter i ett tidigare lager än outputlagret. Mer om detta i avsnitt 9.3!

Slutligen ska det poängteras att Rosenblatts ursprungliga perceptron kan ses som ett helt linjärt nätverk som försetts med *både* ett förprocessande inputlager *och* icke-linjära outputelement. Ingetdera av dessa förhållanden ändrar modellens principiella begränsningar till linjära problem, men som nyss nämnts kan ett väl valt förprocessande lager hjälpa oss transformera ett *visst* icke-linjärt problem till ett linjärt sådant.

7. Attraktornätverk

7.1 Lärande system med attraktorer

I detta kapitel skall vi befatta oss med några exempel på lärande system som på ett väsentligt sätt använder sig av *attraktorer*. För begreppet attraktor och olika typer av attraktorer, se avsnitt 1.4. Först (avsnitt 7.2) ska vi titta noga på ett klassiskt attraktornätverk, Hopfieldnätet, och fundera över om, och i så fall hur, det kan användas som modell för mänskligt minne. I detta avsnitt kommer vi också att bekanta oss närmare med några möjligheter och problem, som uppstår i samband med studiet av dynamiska system i allmänhet och av attraktorsystem i synnerhet.

Hopfieldnätet använder sig av *punktattraktorer*, och en central del av teorin om det utgörs av beviset för att det *har* sådana. I det efterföljande avsnittet kommer vi först att kort beskriva två aktuella modeller med biologisk framtoning, som använder sig av idén om *kontinuerliga* attraktorer. En av dessa är avsedd som en förklaring av hur hippocampus hos vissa djur kan fungera som en sorts karta över omgivningen, och utgår från ett kompetitivt nätverk. De möjliga stabila aktivitetsnivåerna i detta fungerar som en approximation av en kontinuerlig attraktor. Den andra modellen är menad att förklara hur vi kan ställa ögonen i en avsedd, exakt riktning av kontinuerligt många möjliga. Slutligen ska vi ta upp en hypotes som författaren till denna bok nyligen föreslagit, och som bland annat visar hur vissa system med kontinuerliga attraktorer skulle kunna förklara s.k. cellulärt minne. I denna applikation handlar teorin således inte omedelbart om neurala nätverk, men den skulle kunna komplettera existerande hypoteser om inlärning. Den har också andra tänkbara applikationer inom neural nätverksteori i strikt mening.

7.2 Autoassociation i Hopfieldnät

I flera tidigare avsnitt har vi kortfattat berört idéerna om *simultan association* och *mönsterkomplettering*. För att repetera: med ”simultan associ-

ation" menade de associationistiska psykologerna en process där olika element i erfarenheten genom att presenteras tillsammans också fogas samman i minnet, så att en aktualisering av det ena elementet sedan automatiskt kommer att väcka upp en bild av det andra ("mönsterkomplettering"). Även om man i modern psykologi är skeptisk mot den "elementism" som ligger i botten av associationisternas teori, arbetar man ändå gärna med någon hypotes som motsvarar deras teori om simultan association. Denna hypotetiska mekanism är nämligen mycket effektiv när det gäller att förklara två betydelsefulla fenomen, nämligen *perceptuell komplettering* och *innehållsadresserbart minne.*

Vad perceptuell komplettering är, är inte så svårt att inse. Om vi ser huvudet av vår katt sticka fram bakom en dörr så ser vi det inte som ett löst katthuvud, utan som en del av en större helhet – av en katt. Och det är verkligen så att vi *ser* resten av katten, även om vi en viss mening inte ser denna rest "direkt". Vi ser huvudet *som* en del av en levande katt. En del filosofer och psykologer har laborerat med tanken att vi uppfattar resten av katten på ett rent kognitivt, icke-perceptuellt sätt, nämligen genom *trosföreställningar* som åtföljer varseblivningen av katthuvudet. Men så är det inte. Att se ett katthuvud och *tro* att det sitter på en katt är något helt annat än att se ett katthuvud *som* sittande på en katt.

Vad är då innehållsadresserbart minne? Jo, "innehållsadresserbarhet" står för den förnämliga egenskapen hos ett minne, att man kan plocka fram det ur det stora minneslagret genom att ge *en partiell beskrivning av vad det innehåller.* Man behöver alltså inte ange på vilken hylla det ligger, så att säga, och man behöver inte heller specificera hela innehållet (hade man kunnat göra det, så hade man för övrigt inte behövt plocka fram minnet från lagret). Om någon frågar dej, "Kommer du ihåg vilket år den senaste riktigt varma sommaren var", så kommer det antagligen upp en minnesbild (nåja, ett episodiskt minne) av en sommar som inte bara var varm utan också (minns du) solig och fylld av bad och/eller andra trevliga aktiviteter. Till slut dyker det också upp något kännetecken som gör att du kan svara på frågan vilket år denna sommar ägde rum. Det har då ägt rum en *mönsterkompletterande* process i ditt minne. Tanken på den senaste varma sommaren väckte minnet av en mängd händelser som i din erfarenhet inträffade samtidigt med denna sommar, och som därför förknippats med den i ditt minne. Just på grund av denna förknippning kan du plocka fram det fullständiga minnet med ledning av blott några av dess kännetecken.

Perceptuell komplettering är i stor utsträckning – nämligen i den mån som den bygger på perceptuellt minne – också en manifestation av vårt minnes innehållsadresserbarhet. Vi använder oss av varseblivningens och (det övriga) minnets innehållsadresserbarhet dagligen och stundligen. Minnets egenskaper i detta avseende utforskas bl.a. av psykologer i forskning som använder sig av s.k. ”cued recall”, dvs. erinring med hjälp av ledtrådar. Men innehållsadresserbarhet och perceptuell komplettering kan även belysas av vad som händer när man tittar på fixeringsbilder och andra flertydiga bilder (jämför vidare avsnitt 3.2 ovan).

Vi ska nu titta på en tänkbar neuronal mekanism för innehållsadresserbarhet och på en mycket berömd ANN-modell av denna mekanism, nämligen det diskreta Hopfield-nätet. Låt oss först presentera grundtanken genom att åter följa samma banor som Hebb använde sig av (så att säga). Anta att stimulus A och stimulus B representeras genom aktivitet i var sitt neuronalt element X och Y, och att de två stimuli ges samtidigt. Vi antar vidare en helt symmetrisk situation, alltså att det finns såväl en förbindelse från X till Y, som en från Y till X. Efter ett antal upprepningar av den samtidiga presentationen av A och B kan, om Hebbs regel gäller, båda förbindelserna ha blivit så starka, att presentation av enbart A leder till aktivering inte bara av X-noden utan också av Y-noden; motsvarande för presentation av B. Vi har då fått vårt mini-nätverk att uppvisa mönsterkomplettering.

Hopfieldnätet

Den modell som man i ANN-sammanhang oftast hänvisar till när det gäller en mer strikt förklaring av mönsterkomplettering är Hopfieldnätet, så benämnt efter den amerikanske fysiker som 1982 gav ett intressant bevis för dess stabilitetsegenskaper.[129]

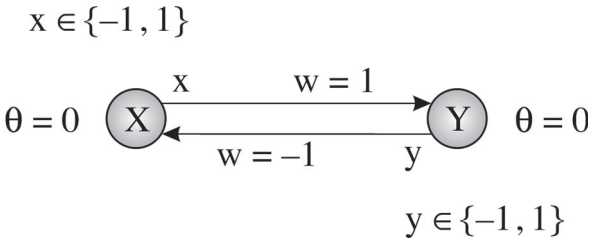
Stabilitet är ett problem som alltid aktualiseras när man arbetar med nätverk som har återkopplade förbindelser (feedback-nät, återkopplade/rekurrenta nät). Med att ett nätverk är stabilt menas här helt enkelt att det, givet ett godtyckligt inputmönster vid en viss tidpunkt, inom ändlig tid alltid hamnar i en punktattraktor. Anledningen till att det är bra att ha sådana statiska slutmönster i samband med modeller för associativt minne är att man då kan modellera minnessökningens *resultat* med hjälp av ak-

[129] Hopfield (1982). – Detta nätverk studerades av många teoretiker långt före Hopfield, och vill man ge det ett mindre personligt namn kan man kalla det ”den diskreta autokorrelatorn”.

tivitetsmönstrets *sluttillstånd*. Detta sluttillstånd betraktas alltså som nätverkets output. Om mönstret inte stannar i ett bestämt läge, har mönsterkompletteringen inte varit entydig, och nätverket har ingen väldefinierad output. När man använder ett återkopplat nätverk för mönsterigenkänning brukar man i samma anda säga att nätverket *känner igen* ett inputmönster om detta mönster också är ett sluttillstånd, dvs. om nätverket inte ändrar tillstånd efter det att inputmönstret lagts in i noderna. (Jämför gärna den abstrakta teorin om habituering ovan, avsnitt 2.1!)

I andra användningar av feedbacknätverk ser man hela det dynamiska förloppet hos den rekurrenta signalen som utsignal, men det är en helt annan historia; se kapitel 10.

För att se att aktivitetsmönstret i godtyckliga återkopplade nätverk inte alltid går till punktattraktorer, betrakta först den lilla modellen i figur 45 där X och Y antas vara bipolära tröskelenheter med samma tröskel 0 (dvs. x = 1 om och endast om x_in > 0, och –1 annars). De är kopplade till varann, men det finns ingen förbindelse från X direkt till X, eller från Y direkt till Y.



Figur 45. Ett återkopplat bipolärt mininätverk. θ: tröskelvärde för enheten X respektive Y.

Vi måste nu ta ställning till om nätverkets enheter skall *uppdateras synkront* eller *asynkront.* Att de uppdateras synkront innebär att omvandlingen av nettoinput till aktivitet äger rum samtidigt i alla enheter. I den tidigare framställningen av olika nätverk har vi alltid gjort ett implicit antagande om synkron uppdatering. I det aktuella lilla nätverket betyder synkroni att aktiviteterna uppstår i båda noderna samtidigt. Asynkroni innebär istället att aktivitet uppstår i ett element i taget, och att alla element sedan skickar ut sina signaler innan nästa element hunnit aktiveras. I det första fallet (synkron uppdatering) omvandlas ett helt mönster till ett nytt, helt mönster i ett slag. I det andra fallet uppdateras det gamla mönstret bara med en komponent i taget, *men å andra sidan används de nya en-*

skilda aktivitetsvärdena så fort de fallit på plats, det vill säga redan när nästa komponent i mönstret ska uppdateras. Detta gör, som vi strax skall se, ofta en stor skillnad för systemets dynamik. – I det asynkrona fallet kan man låta komponenterna uppdateras växelvis, eller låta slumpen bestämma vilken komponent som skall uppdateras vid en viss tidpunkt.

Låt oss nu starta vårt mini-nätverk genom att sätta både x och y till 1, och antag synkron uppdatering.

1. Det som först händer är att noden X får nettoinput –1 samtidigt som Y får nettoinput 1. Aktiviteten x slår därför om till –1 medan y behåller sitt värde 1.

2. I nästa steg får både X och Y nettoinput –1, och båda antar alltså värdet –1.

3. x_in = 1, y_in = –1, vilket leder till x = 1 och y = –1

4. Nu får både X och Y nettoinput 1, och deras aktiviteter antar därför båda värdet 1.

Men det fjärde steget innebär att vi är tillbaka vid utgångsläget! Eftersom nätverket också i övrigt ser likadant ut som från början (vikterna har t.ex. inte ändrats sig) så kommer det att genomlöpa samma sekvens av tillstånd en gång till – ja, i all oändlighet. Denna sekvens är ju vad vi kallar en *gränscykel,* som närmare bestämt ser ut så här:

(7.2.1) $(1, 1) \rightarrow (-1, 1) \rightarrow (-1, -1) \rightarrow (1, -1) \rightarrow (1, 1) \rightarrow ...$

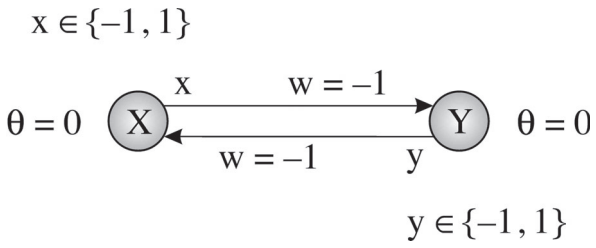
Det är av intresse att också se på asynkron uppdatering, med specifikationen att enheterna uppdateras *varannan* gång och att Y uppdateras först. Läsaren kan själv verifiera att nätverkets aktivitetsvektor kommer att genomgå sekvensen

(7.2.2) $(1, 1) \rightarrow (1, 1) \rightarrow (-1, 1) \rightarrow (-1, -1) \rightarrow (1, -1) \rightarrow (1, 1) \rightarrow$
$\rightarrow (1, 1) ...$

Det här ser nu inte alls ut som en gränscykel, eftersom (1, 1) ibland följs av (1, 1) och ibland av (–1, 1)! Sekvensen (7.2.2) av aktivitetsmönster är med andra ord inte en Markovprocess av ordning 1 (se avsnitt 1.4). Man inser snart att detta beror på att det finns en ”dold variabel” med i spelet,

nämligen *om X eller Y står i tur att uppdateras*. Tar man med denna variabel i tillståndsdiagrammet, så ser man tydligt att det är fråga om en gränscykel (visa det!).

I figur 46 har vi ändrat på en av vikterna i vårt lilla nätverk.



Figur 46. Andra vikter i mininätverket från figur 45.

Skillnaden gentemot det förra nätverket är att båda vikterna är = –1. Vikterna är med andra ord symmetriskt fördelade. Synkron uppdatering ger:

(7.2.3) $(1, 1) \rightarrow (-1, -1) \rightarrow (1, 1) \rightarrow (-1, -1) \rightarrow ...$

dvs. återigen en gränscykel (nu av längden 2). Väljer vi däremot asynkron uppdatering (varannan gång) och börjar med att uppdatera Y, får vi

(7.2.4) $(1, 1) \rightarrow (1, -1) \rightarrow (1, -1) \rightarrow (1, -1) \rightarrow ...$

dvs. nätverket ”stannar” i aktivitetsvektorn (1, –1). Börjar vi med att uppdatera X får vi istället (på grund av symmetrin):

(7.2.5) $(1, 1) \rightarrow (-1, 1) \rightarrow (-1, 1) \rightarrow (-1, 1) \rightarrow ...$

dvs. nätverkets aktivitetsvektor hamnar i det stabila tillståndet (–1, 1).

De första tre av exemplen som vi nu givit illustrerar att *rekurrenta nätverk med asymmetriska vikter*, liksom rekurrenta nätverk med *symmetriska vikter och synkron uppdatering*, inte nödvändigtvis behöver hamna i ett stabilt sluttillstånd.

Det fjärde exemplet visar i sig ingenting mer än att *det finns* rekurrenta nätverk med *symmetriska vikter och asynkron uppdatering* som har (ett eller flera) stabila slutttillstånd. Men det går faktiskt att bevisa en sådan

stabilitet för en stor klass av nätverk av denna typ. Vad Hopfield gjorde var just att lägga fram ett sådant bevis. Vi ska närmast koncentrera oss på *en* uppsättning villkor (bland många möjliga) som är tillräckliga för att ett återkopplat nätverks aktivitetsvektor alltid ska gå till någon punktattraktor, nämligen:

(H1) Noderna uppdateras *asynkront med slumpvis val av nod*

(H2) Nätverket har en *symmetrisk viktmatris*

(H3) Det finns inte någon effektiv förbindelse från något element till elementet själv (dvs. viktmatrisen har *värdet 0 överallt på diagonalen*)

(H4) Aktiveringsfunktionen är en speciell bipolär stegfunktion som ger aktiviteten $x = 1$ om nettoinput > 0, ger $x = -1$ om nettoinput < 0, och lämnar aktiviteten *som den var i förra steget* om nettoinput $= 0$.

När vi i det följande talar om ”Hopfieldnätet” utan närmare specifikation avser vi ett nätverk som uppfyller (H1) – (H4). Det finns många andra varianter av Hopfieldnätet, inklusive sådana med kontinuerlig aktivering, men vi kommer inte att gå in på dem här.

Hopfieldnätets tillståndskonvergens

Låt oss då betrakta det fullständiga Hopfielddiagrammet för ett godtyckligt Hopfieldnät med n enheter.

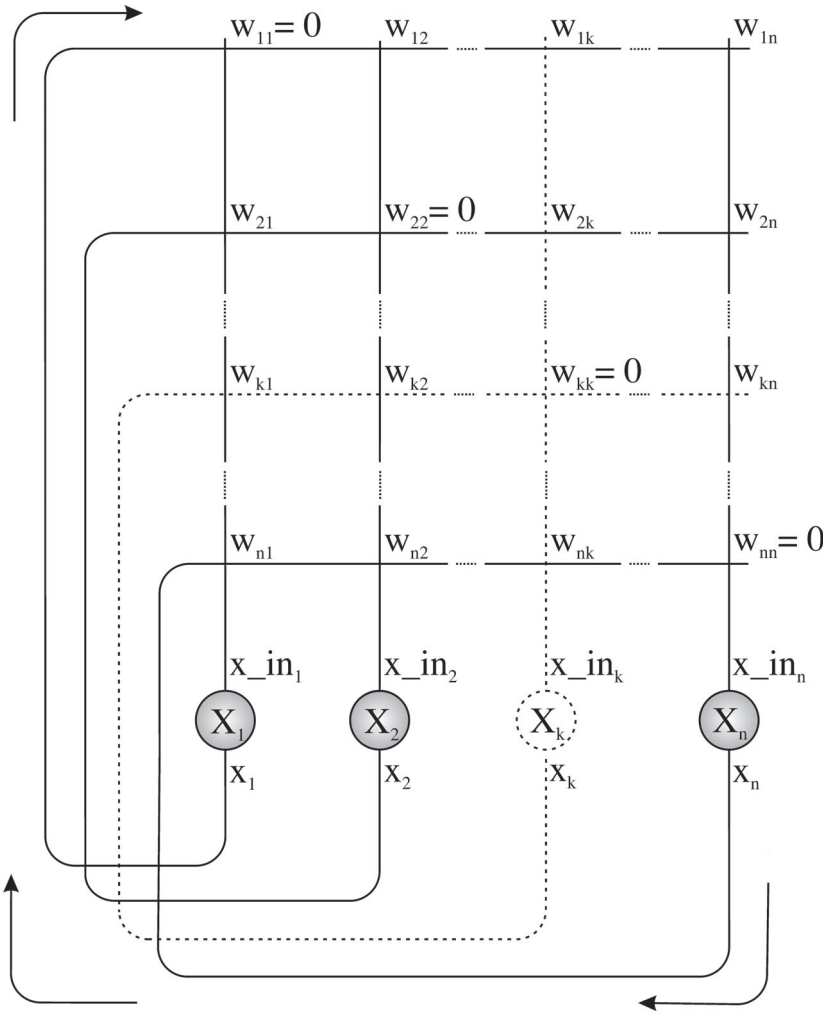


Fig. 47. Ett Hopfield-nät med n noder. X_k står för en godtycklig nod med aktivitet x_k, nettoinput x_in_k samt vikter w_{ik} och w_{kj} på inkommande respektive utåtgående ledningar.

Nätverkets vikter är i det närmast följande resonemanget fixa. Vi talar just nu inte om *viktuppdatering* genom inlärning, utan om *aktivitetsuppdatering* genom att en signal går runt i nätverket!

Betrakta en nod X_k som just ska uppdateras. Aktiveringsregeln säger ju, att x_k skall bli 1 om $x\_in_k > 0$; x_k ska bli -1 om $x\_in_k < 0$; om slutligen $x\_in_k = 0$ ska x_k inte ändras. Nettoinput x_in_k är i sin tur bestämd som $\sum_i x_i \cdot w_{ik}$. Låt oss nu definiera den *lokala energin* E_k för noden X_k vid tidpunkten ifråga genom

$$E_k = -\frac{1}{2}\sum_i x_i \cdot x_k \cdot w_{ik} = -\frac{1}{2} x_k \cdot \sum_i x_i \cdot w_{ik} = -\frac{1}{2} x_k \cdot x\_in_k \qquad (7.2.6)$$

Den lokala energin E_k kan enkelt visas vara *en indikator för om aktiviteten x_k kommer att ändras vid uppdateringen*. Man kan tänka på E_k som ett mått på *disharmonin* mellan tillståndet x_k och nettoinput x_in_k. Betrakta t.ex. fallet $x_k = 1$ och $x\_in_k < 0$. E_k är > 0 och x_k kommer att ändra tecken till -1. Genom att betrakta de övriga möjliga fallen inser man att E_k alltid är positiv om x_k kommer att ändra tecken vid uppdateringen, och negativ eller noll om x_k *inte* kommer att ändra tecken.

Det avgörande tricket är nu att titta på vad som händer med den lokala energin *över tiden* när X_k faktiskt uppdateras. Man inser snart att E_k kommer att byta tecken från positiv till negativ, det vill säga minska, om x_k ändras, men annars vara oförändrad. Den andra halvan av påståendet är trivialt sann eftersom högerledet i uttrycket (7.2.6) i detta fall är detsamma efter uppdateringen som före. Den första halvan är sann, eftersom det som händer i (7.2.6) i detta fall är att *x_k byter tecken åt just det håll som tecknet för x_in_k anger.* Vare sig detta tecken är plus eller minus *ökar* alltså produkten av x_k och x_in_k, dvs. E_k minskar.

Och därmed är vi nästan framme vid Hopfields konvergensbevis. Detta går nämligen ut på att man visar att varje ändring av nätverkets aktivitetsmönster under en uppdatering gör att energin, *sedd över nätet som helhet*, minskar. Med andra ord, den *globala* energin

$$E = \sum_k E_k = -\frac{1}{2}\sum_{i,k} x_i \cdot x_k \cdot w_{ik} \qquad (7.2.7)$$

minskar vid varje aktivitetsändring. Då nätet bara har ett ändligt antal möjliga aktivitetsmönster, som var och en entydigt bestämmer en global energi, följer att *energin så småningom måste nå ett stabilt minimum där inga aktivitetsförändringar sker.*

Beviset är dock inte klart än. Vi har visserligen visat, att den *lokala* energin E_k i en nod minskar så snart dess uppdatering innebär en förändring av dess aktivitet. Beviset för att detsamma gäller för den globala energin använder sig på ett väsentligt sätt av informationen att $w_{ik} = w_{ki}$, dvs. att viktmatrisen är symmetrisk. Utgå från någon nod X_k, och betrakta det ”inkommande” bidraget $-1/2 \cdot x_j \cdot x_k \cdot w_{jk}$ till dess lokala energi E_k från en

godtycklig annan nod $X_j \neq X_k$. Genom att $w_{ik} = w_{ki}$ har denna term samma värde som det ”utgående” bidraget $-1/2 \cdot x_k \cdot x_j \cdot w_{kj}$ från noden X_k till den lokala energin E_j i X_j. *Summan av alla bidrag från X_k till andra lokala energier än E_k måste därför vara lika med summan av alla bidrag till E_k från andra noder än X_k.* Men den senare summan är inget annat än den lokala energin E_k. Av resonemanget följer att om X_k uppdateras, så måste den globala energin ändras åt samma håll som E_k. Vilket enligt ovan i sin tur betyder, att Hopfieldnätets aktivitet alltid går till en punktattraktor.

Beviset behöver alltså antagandet av en symmetrisk viktmatris. Med lite eftertanke inser man också att antagandet om asynkron uppdatering är nödvändigt för att det ska gå igenom. Om nämligen andra aktiviteter än x_k kan ändras samtidigt med att X_k uppdateras, gäller inte längre att den lokala energin E_k minskar om och endast om x_k ändras. Därmed är inte sagt att *endast* sådana återkopplade nätverk som uppfyller dessa två villkor kan ha punktattraktorer, bara att deras stabilitet i så fall måste bevisas på något annat sätt.

Autokorrelation i Hopfieldnätet

Låt oss nu titta lite närmare på hur innehållsadresserbart minne och mönsterkomplettering uppstår i ett Hopfieldnät. Nätverket i figur 48 består av tre bipolära noder med tröskeln 0, som alla är förbundna med varann med slumpvis framtagna men symmetriska vikter och som uppfyller (H1) – (H4) ovan. Vi kan illustrera nätverkets förbindelser genom följande diagram, där det alltså gäller att $w_{ik} = w_{ki}$; $w_{ii} = 0$:

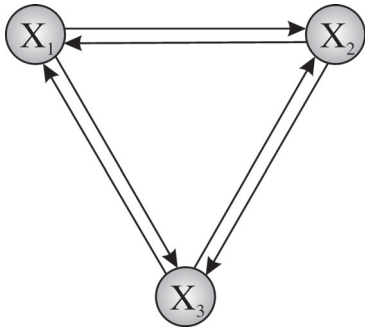


Fig. 48. Ett Hopfieldnät med tre enheter.

Om vi ger nätverket ett visst bipolärt inputmönster, t.ex. (+1 −1 −1), och sedan låter det uppdateras asynkront med slumpvis val av nod, kommer

det slutligen att hamna i något attraktormönster, som *kan* vara detsamma som inputmönstret men givetvis inte behöver vara det. Gör vi om försöket med samma mönster *kan* aktiviteten hamna i samma sluttillstånd, men det kan också hända – eftersom uppdateringssekvensen inte behöver vara densamma – att den hamnar i ett annat (vi såg ett exempel på det senare i mini-nätet i figur 46). Väljer vi så ett annat startmönster, t.ex. (+1 +1 –1), kan vi hamna i antingen något av de tidigare mönstren eller ett nytt. Antalet slutmönster är dock som regel mindre än antalet möjliga inputmönster.

Nu lägger vi på en Hebb-liknande mekanism på nätverket. Enligt Hebbs ursprungliga regel skall ju styrkan hos förbindelserna mellan samtidigt aktiva neuron öka. Hebb tänkte givetvis på neuron som varande antingen aktiva eller inte aktiva (möjligen med en graderad, positiv aktivitet mellan 0 och 1). Vår matematiska formulering av Hebbregeln (avsnitt 4.6, formel 4.6.1) tillåter dock även bipolär aktivering. Enligt (4.6.1) ökar vikterna mellan två noder i varje inlärningssteg med en viss storhet (inlärningskonstanten) multiplicerad med de två nodernas aktiviteter. Förbindelserna kommer då att *förstärkas* mellan noder som har *samma aktivitetsnivå*, vare sig den är +1 eller –1, men *försvagas* mellan de noder som har *olika* aktivitetsgrad. Detta är formellt sett en utökning av Hebbs regel men andemeningen är densamma.

Inlärningsregeln antas endast verka i en träningsfas, och under denna fas dessutom bara när man just lagt in inputmönstret i noderna – alltså inte medan Hopfieldnätet uppdaterar sig självt. På detta sätt garanterar man att nätverket lär sig de och endast de associationer som förefinns i inputdata under träningsfasen.

Vad får då tillämpningen av en sådan regel för slags effekter? Antag att vi har givit inputmönstret (+1 +1 –1) en eller flera gånger. De reciproka förbindelserna mellan nod X_1 och nod X_2 har då blivit starkare, medan de mellan X_1 och X_3 samt de mellan X_2 och X_3 blivit svagare (kanske rentav negativa). Det betyder för det första, att det är mer sannolikt än förut att inputmönstret (+1 +1 –1) också är ett sluttillstånd. De lika aktiviteterna i X_1 och X_2 tenderar nu nämligen att *upprätthålla varandra*, liksom de olika aktivitetsnivåerna i X_1 och X_3 (respektive X_2 och X_3). Det är inte så svårt att inse att så länge man bara tränar på ett enda inputmönster, så kommer det så småningom att med säkerhet bli ett slutmönster – det vill säga, man har lärt Hopfieldnätet att *känna igen* detta mönster!

För det andra gäller att om vi efter träningen ger mönstret (+1 0 –1) som input, så finns det nu en god chans att X_1 påverkar X_2 så mycket att X_2 byter aktivitet till värdet +1. Motsvarande gäller förstås om vi ger inputmönstret (0 +1 –1). Och ger vi input (+1 +1 0), så finns det en god chans att de positiva värdena i X_1 och X_2 leder till ett negativt värde i X_3. Vi har redan benämnt detta fenomen *autoassociation* eller *autokorrelation:* de olika komponenterna i inputmönstret har genom Hebb-inlärningen associerats med varandra på ett sådant sätt, att varje komponent nu tenderar att framkalla de andra.

Att ge ett inputmönster med en nolla i till ett annars bipolärt nätverk kan nu tolkas som att man ger nätverket ett mönster som är *ofullständigt.* Nätverket har, kan man säga, inte fått någon information huruvida det ”egentligen” ska vara +1 eller –1 på denna plats. Vad vi har gjort troligt med vårt resonemang är med andra ord att om man ger ett tränat Hopfieldnät ett inputmönster som avviker från träningsmönstret genom att vara ofullständigt på någon enstaka punkt, så tenderar nätet ändå att ge träningsmönstret som output. Och detta är ju inget annat än den mönsterkomplettering som vi efterlyser.

Mönsterkomplettering fungerar också (om än inte lika ”starkt”) om man istället för ett *ofullständigt* inputmönster presenterar ett som är *felaktigt* (+1 istället för –1, eller vice versa) på någon punkt – eller på flera punkter, om nätverket har någorlunda många noder.

Vi beskrev förvisso autokorrelation redan i föregående kapitel, i samband med autoassociation i de linjära feed-forwardnätverken. Fördelarna med Hopfield-nätet är dels att man inte behöver mata in två kopior av det mönster som ska tränas, dels att den inbyggda stegfunktionen gör att mönsterkompletteringen kan bli exakt (i de helt linjära systemen är det kompletterade mönstret alltid bara en approximation av originalet). Det var det senare som vi avsåg när vi sade att ett neuralt nätverk med stegfunktion kan vara i en stark mening *feltolerant* – nämligen om nätverket lärt sig en så stark association mellan ett inputmönsters komponenter, att trösklarna uppnås för alla outputkomponenterna även om en komponent saknas i input.

Som nämndes tidigt (i avsnitt 1.4) kan man beskriva ett attraktornätverks respons vid mönsterkomplettering som att det ”känner igen” även en förvrängd input. Men man måste då komma ihåg, att begreppet förvrängd input måste definieras i relation till vad vi betraktar som en *normal* input.

På en annan planet hade det kanske varit viktigare att känna igen mönstret (0 +1 –1) *för vad det är*... nämligen en helt okorrumperad input från en illasinnad marsian. Detta är bara ett mer drastiskt sätt att återigen påpeka att feltolerans genom autokorrelation åstadkoms på bekostnad av att *andra* inputmönster än de inlärda blir desto mer instabila. I en stabil omgivning där det faktiskt föreligger höga korrelationer mellan vissa inputkomponenter är detta pris förvisso värt att betala. Så länge vi håller oss kvar på jorden blir det ju ändå oftast rätt, när vi använder felkorrektion på grundval av autoassociation. Vad skulle vi annars ha vårt minne till...

Liksom för linjära associatorer gäller att man med fördel kan träna ett Hopfieldnät på flera mönster samtidigt, och att deltaregeln ger bättre resultat i detta avseende än Hebbregeln. Träningen gör, om man lyckas bra, en attraktor av varje träningsmönster; varje senare inputmönster tenderar att rekonstruera det träningsmönster som inputmönstret ”liknar mest”. (Att vi sätter citationstecken omkring ”liknar mest” indikerar att vår bedömning av likhet mellan mönstren inte behöver överensstämma med nätverkets.) En sådan ”inlagring” av flera mönster samtidigt förutsätter, precis som när det gäller association av mönster i en linjär associator, ett visst mått av oberoende mellan mönstren. Ligger mönstren för nära varandra, tenderar attraktorerna att smälta samman. Den formella beskrivningen av dessa villkor, och därmed av Hopfieldnätets begränsningar, måste vi lämna därhän. Det ska dock nämnas att Hopfieldnätet – trots att det är av stort historiskt och principiellt intresse – inte har särskilt stor mönsterlagringskapacitet jämfört med många andra återkopplade nätverk (inte ens om man använder deltaregeln).

Vi har här huvudsakligen begränsat oss till kvalitativa resonemang utan matematiska detaljer, men de torde ändå ha visat tydligt att Hopfieldnätet har egenskaper som starkt påminner om vårt innehållsadresserbara minne och om den mönsterkomplettering som detta ägnar sig åt. Kunskapen att hjärnbarkens nätverk är massivt återkopplade bidrar också till att göra modellen trovärdig som en första approximation av vad som händer i hjärnan vid simultan association.

En invändning mot användandet av Hopfieldnät som modell för biologiska system är antagandet om en symmetrisk viktmatris. Här ska man dock minnas att träning med Hebbregeln för simultana associationer tenderar att göra ett nätverks viktmatris allt mer symmetrisk, vilket innebär att även ett från början ”instabilt” nät med tiden gärna får punktattrakto-

rer.[130] Förvisso har hjärnbarken en komplexitet som vida överstiger den hos våra matematiska modeller, både dem vi förstår analytiskt och dem som vi bara kan studera genom simulering. Men något som i grunden *liknar* den autokorrelation som äger rum i enkla Hopfieldnät är säkerligen också verksamt i många av dessa mycket mer komplicerade, återkopplade biologiska nätverk.

Vi ska nu övergå till att titta på sådana neurala nätverk (och andra dynamiska system med relevans för inlärningsteorin) som har *kontinuerliga* attraktorer.

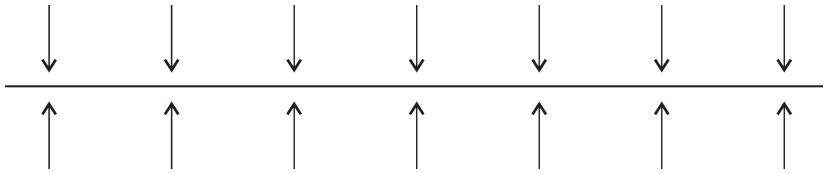
7.3 Kontinuerliga system och kontinuerliga attraktorer

Kontinuerliga attraktorer och ögonrörelsers dynamik

I avsnittet om systemteori (1.4) stiftade vi bekantskap med begreppen *gränscykel* och *punktattraktor*, och de användes nyss i kapitlet om Hopfieldnätet. För beskrivningen av en stor klass av biologiska (och andra) system är ett annat begrepp väsentligt, nämligen *kontinuerlig attraktor*. I neurala nätverkssammanhang har det uppmärksammats sedan mitten av 1990-talet, och det kan förväntas bli än mer betydelsefullt inom ANN-teorin i framtiden.

När du studerat kulan som rullar mot en punktattraktor i sin skål, ta upp den och lägg den istället i en horisontell stupränna som erbjuder viss friktion. Om du lägger den på en valfri punkt längst ner i rännans konkavitet kommer den att ligga kvar där; placerar du den på en punkt längre upp på stuprännans vägg så kommer den att söka sig mot punkten rakt nedanför sig. Varje punkt på den linje som definierar stuprännans lägsta nivå utgör således en punktattraktor, och linjen som helhet bildar *ett sammanhängande kontinuum av punktattraktorer* –vilket är just vad vi menar med en "kontinuerlig attraktor". I detta fall talar man om en *linjeattraktor*. Figur 49 illustrerar resonemanget.

[130] I den modell av Hopfieldnätet som ingår i utbildningspaketet från Filosofiska Institutionen vid Göteborgs Universitet kan man experimentera med denna idé, liksom med binär aktivering av nätverket, m. m. För webadress se not 123.



Figur 49. En linjeattraktor. Pilarna visar några möjliga vägar som systemets tillstånd kan ta mot ett tillstånd på attraktorn.

Man kan lägga märke till att en punkt i botten av stuprännan är en punktattraktor endast för kulor som befinner sig på "fallinjen" mot punkten. Det betyder bland annat att en kula som först ligger på botten, och som sedan genom en störning flyttas åt ett annat håll än vinkelrätt mot stuprännan, inte kommer att återvända och åter närma sig ursprungsläget. Jämviktslägena är med andra ord inte är stabila gentemot en godtycklig, liten störning på det sätt som den enskilda punktattraktorn i botten av en skål är det. En störning kan lika väl medföra att ett annat jämviktsläge uppnås.[131]

Egenskapen hos ett jämviktsläge att systemet återvänder till det efter en störning kallas *asymptotisk stabilitet,* och punkterna på stuprännans botten är tydligen inte (helt) asymptotiskt stabila. Det lätt att inse att detsamma måste gälla för *varje* kontinuerlig attraktor. Varje punktattraktor i en sådan har ju, definitionsmässigt, andra punktattraktorer i sin närmaste omgivning, vilket betyder att en del små störningar kommer att föra systemet direkt till en annan attraktor. I fallet med stuprännan gäller detta för alla störningar i rännans riktning. Däremot är detta system asymptotiskt stabilt gentemot störningar vinkelrätt mot rännans riktning.

I andra fall är det till och med så, att punktattraktorerna i en kontinuerlig attraktor inte är asymptotiskt stabila under *någon* av de störningar som är definierade för systemet. Placera till exempel kulan på ett perfekt horisontellt golv. Var du än lägger den på golvet kommer den att ligga kvar. Med andra ord, varje punkt på golvet är ett jämviktsläge, och golvytan som helhet bildar ett kontinuum av sådana jämviktslägen. Här gäller därför att kulan aldrig återvänder till sitt utgångsläge efter en liten störning.

Det har ibland förts fram som en invändning mot idén om kontinuerliga attraktorer i biologin, att system med sådana attraktorer inte är tillräckligt

[131] När vi säger att ett kontinuerligt system "uppnår" ett visst jämviktsläge innefattar vi alltid det fall då systemet asymptotiskt närmar sig läget ifråga utan att egentligen nå fram till det.

stabila för att kunna upprätthålla livsfunktioner under någon längre tid. Men lägg märke till att punktattraktorerna i en kontinuerlig attraktor inte är *labila* lägen i den meningen, att en godtycklig, liten störning automatiskt skulle få systemet att *avlägsna sig* ännu längre från det ursprungliga läget. En kula på toppen av en uppochnervänd skål bildar ett system som är labilt i denna mening. Men inte ens en kula på ett friktionsfritt golv bildar nödvändigtvis ett labilt system (eftersom man kan lägga kulan en liten bit bort och få den att ligga kvar), och kulan i en stupränna med friktion bildar som nämnts ett system som är delvis asymptotiskt stabilt – kulan återvänder faktiskt till ursprungsläget efter en godtycklig "vertikal" störning.

En annan viktig faktor som man måste ta hänsyn till när man diskuterar stabilitet är vilken insats som erfordras för att flytta systemet från en punktattraktor till en annan. En tung kula på ett klibbigt golv ligger stadigt, trots att systemet inte alls är asymptotiskt stabilt. Om ett system med en kontinuerlig attraktor reagerar med blott små förändringar i sitt attraktorläge på de yttre inflytanden som systemet brukar utsättas för, säger vi att det är *trögt*. Det finns anledning att tro att tröghet hos biologiska system är en lika viktig egenskap som asymptotisk stabilitet. Framförallt är det antagligen trögheten som i första hand förklarar att biologiska system kan vara långlivade trots förekomsten av kontinuerliga attraktorer.

Vad finns det då för skäl att tro att kontinuerliga attraktorer spelar någon väsentlig roll i biologiska system? Jo, framförallt är det deras egenskap att ha *godtyckligt många jämviktslägen* som är intressant. Ta till exempel vår förmåga att fixera blicken i en godtycklig riktning. Denna förmåga kan med fördel analyseras i termer av ett system som innehåller ett kontinuum av punkattraktorer – en för varje möjlig blickriktning – alltså en kontinuerlig attraktor. En av de första applikationerna av begreppet kontinuerlig attraktor till ANN-teorin var just en modell för ögonrörelser, och vi ska först ta en kort titt på den i en enkel formulering som inte förutsätter teorin för kontinuerliga system (dvs. den arbetar med diskret tid).

H.S. Seungs tidigaste teori för styrning av ögonrörelser bygger på observationen att ett system av två identiska, *maximalt linjära* neuron, som har kontinuerligt graderad aktivitet och som ömsesidigt exciterar varandra med vikten 1, har en kontinuerlig attraktor.[132] Antag nämligen att neuro-

[132] Seung 1996. Samma effekt, har Seung också påpekat, kan uppnås med ett enda linjärt neuron och en rekurrent synaps med vikten 1.

nen X och Y har aktiviteterna x respektive y och att de, när de endast påverkas av varandra, lyder följande lagar:

(7.3.1) $x(t+1) = y(t)$

(7.3.2) $y(t+1) = x(t)$

Detta system kommer, om det lämnas åt sig självt, i allmänhet att oscillera. Undantagen utgörs av de fall när systemet startas med samma aktivitet i båda neuronen. Då kommer det att stanna i utgångsläget – och det gäller oberoende av vilken aktivitetsnivå man tilldelat det från början. Det vill säga, systemet har en kontinuerlig attraktor som består av alla aktivitetspar av typ (a, a) inom det intervall som neuronen arbetar i.

Man inser också lätt att man kan få aktiviteten i systemet att stanna på en godtycklig nivå genom att utifrån sätta denna nivå i den *ena* nervcellen, t.ex. X, och hålla den konstant där även under nästa epok. Det vill säga, systemet ”minns” vilken nivå man fixerade X till under denna korta tid. Seungs idé är nu att kommandot om blickriktning skickas som en kort styrsignal från något överordnat centrum till neuronet X. Denna korta signal räcker för att neuronet X (eller för den delen Y) sedan ska kunna sända ut en konstant signal till ögonmusklerna ända tills nästa styrsignal kommer. Det enda felet med denna genialt enkla modell är att verkliga neuron inte är linjära! För senare forskning – inklusive Seungs – kring mer realistiska modeller av ögonrörelser, som också bygger på kontinuerliga attraktorer, bör läsaren konsultera någon modern framställning om återkopplade nätverk i nervsystemet.[133]

En annan tidig tillämpning var i analysen av de så kallade “platscellerna” i hippocampus. Det finns evidens som talar för att mönstret av neural aktivitet i en viss *population* av celler i en råttas hippocampus motsvarar den plats i en bur som råttan befinner sig på. Om aktivitetsmönstret kunde visas ha egenskaperna hos en kontinuerlig attraktor, så skulle man förstå hur det kan koda för en *godtycklig* plats i buren. Och faktiskt har sådana modeller föreslagits; vi ska beskriva en av dem i kapitlet om kompetetiva nätverk (avsnitt 8.1).

En tredje typ av möjliga applikationer har att göra med inlärning av responser med *graderad styrka*. Om det till exempel ska vara möjligt för en människa att lära sig att dra med en viss, förutbestämd styrka i ett hand-

[133] T.ex. Dayan & Roberts (2001), avsnitt 7.4.

tag, oberoende av vilken denna förutbestämda styrka är (inom ett visst intervall), och det relevanta neurala nätverket är ett attraktornätverk, så måste det innefatta en linjeattraktor som kodar för alla dessa olika styrkor. Mer om detta nedan.

Författaren är övertygad om att begreppet kontinuerlig attraktor har många andra viktiga biologiska tillämpningar, inte bara inom ANN-teori och allmän inlärningsteori utan också vad gäller *inlärning på cellulär nivå.* Det har nyligen visats att enskilda nervceller kan ha ett slags minne, oberoende av styrkan hos förbindelserna med andra celler. Detta cellulära minne kan också med fördel analyseras i termer av kontinuerliga attraktorer. Men innan vi går igenom de olika möjliga tillämpningarna måste vi förbereda marken genom att introducera några fler grundbegrepp och tänkesätt från teorin om kontinuerliga system. Repetera gärna avsnitt 1.4 innan du fortsätter!

Kontinuerliga system: tillstånds- och differentialekvationer

Om en beskrivning av ett dynamiskt systems storheter, *inklusive tiden*, är given i termer av kontinuerliga variabler så talar vi om ett *kontinuerligt system*. I en sådan beskrivning kan man modellera ett systems beteende under godtyckligt små tidsavsnitt. Formler som i en enklare modell relaterar tillstånd vid två på varandra följande, ändliga tidssteg till varann (som t.ex. i ekvationerna 7.3.1–7.3.2 ovan) ersätts då av mer ”finkorniga” formler som beskriver systemets utveckling vid varje tidpunkt.

Om beskrivningen av ett kontinuerligt system inkluderar en uppsättning variabler som är nödvändig och tillräcklig för att alla framtida tillstånd ska kunna förutsägas från tillståndet vid en given tidpunkt, så har vi att göra med ett *deterministiskt* kontinuerligt system. Vi ska här för enkelhets skull främst intressera oss för deterministiska kontinuerliga system med två variabler. Ett bra exempel på dylika system är beskrivningen av en pendel som svänger i ett givet plan. Variablerna *vinkel* (mot vertikalen) och *vinkelhastighet* bildar ett deterministiskt kontinuerligt system. Detta system utvecklar sig alltså, givet vissa initiala värden på de två variablerna, på ett helt förutbestämt sätt över tiden.

I princip kan ett sådant deterministisk system beskrivas genom en *tillståndsekvation*, där systemets tillstånd ges som en explicit funktion av initialtillståndet och tiden. Om vi låter **y** beteckna tillståndsvektorn kan vi rent allmänt skriva:

(7.3.3) $\mathbf{y}(t) = f(\mathbf{y}(t_0), t)$

En (tänkt) pendel utan friktion som startas i en liten vinkel x från lodlinjen (en s.k. linjär pendel) approximerar ekvationerna:

(7.3.4) $x = \omega_0 \cos(\omega t)$

(7.3.5) $\frac{dx}{dt} = \omega_0 \sin(\omega t)$

där ω (omega) är en systemkonstant som bestäms av pendelns längd, och ω_0 är pendelns ursprungliga avvikelse från vertikalen. Observera att vinkelhastigheten dx/dt här betraktas som en systemvariabel (inte bara som derivatan av en sådan, se nedan).

Det är ganska ovanligt att man har tillgång till ett systems tillståndsekvation, ens approximativt. Oftare har man på teoretiska och/eller empiriska grunder exakt eller approximativ kunskap om något samband mellan å ena sidan systemets *tillstånd*, å andra sidan dess *rörelsetendens*. Ett viktigt redskap för beskrivning som man då kan ta till är de så kallade *differentialekvationerna.* Dessa utvecklades först av Newton och Leibniz, och kom i och med Newtons mekanik att spela en fundamental roll för vår förståelse av fysikaliska system. De flesta teorier av idag som modellerar neural funktion på detaljnivå använder sig också av differentialekvationer.

Vad innebär då en sådan? Jo, en differentialekvation uttrycker någon form av matematiskt samband som inkluderar *derivatorna* (eventuellt även högre ordningens derivator) av systemets tillståndsvariabler, men ger inte något explicit uttryck för tillståndet självt. Ett välkänt exempel är ekvationen:

(7.3.6) $\frac{dx}{dt} = ax$

(där a är en konstant), som beskriver ett system i exponentiell tillväxt; ett annat är beskrivningen av pendeln från 7.3.4–7.3.5 med ekvationen

(7.3.7) $\frac{d^2x}{dt^2} = -x$

Förenklade beskrivningar av ett neuron X, betraktat som ett kontinuerligt system, ser ofta ut på ungefär följande sätt:

$$\frac{dx}{dt} = I(x_0 - x) - Ex \tag{7.3.8}$$

där x är den kontinuerligt graderade aktiviteten. Här står I för summerad nettoinput, x_0 för ett tak som aktiviteten i X antas ha, och slutligen E för en konstant faktor som gör att aktiviteten i X avtar spontant över tiden.

Antag slutligen att två lok har massorna m_1 respektive m_2 och löper på ett friktionsfritt, rakt järnvägsspår. De attraherar varandra enligt Newtons gravitationslag och påverkas inte av några andra krafter. Loken styrs då av följande system av differentialekvationer, där x och y är lokens positioner på rälsen och G den allmänna gravitationskonstanten:

$$\frac{d^2x}{dt^2} = G\frac{m_2}{(x - y)^2} \tag{7.3.9}$$

$$\frac{d^2y}{dt^2} = G\frac{m_1}{(x - y)^2} \tag{7.3.10}$$

Det är nu av stort intresse att man från differentialekvationen (-erna) för ett system *ibland, men långt ifrån alltid*, kan räkna fram tillståndsekvationen (alltså tillståndet som en explicit funktion av tiden). För de ekvationer som vi exemplifierat med ovan går det dock bra, givet tillräckliga initialvillkor. Lösningen kommer praktiskt taget alltid i form av en *familj* av tillståndsekvationer, eftersom systemets framtida tillstånd förutom av den inre dynamiken (som anges av differentialekvationerna) också bestäms av initialtillståndet (jämför termen ω_0 i ekvationerna 7.3.4–7.3.5 ovan).

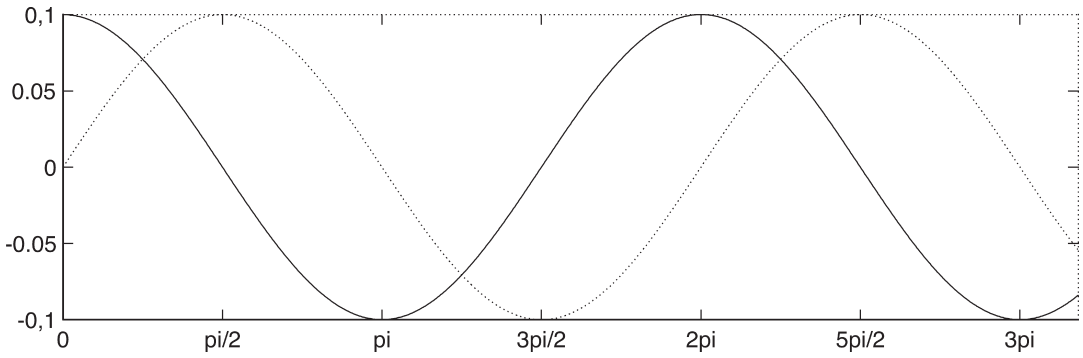
Sambandet mellan differentialekvationer och tillståndsekvationer utgör studieobjektet för en omfattande matematisk disciplin, och vi ska inte alls gå in på några detaljer. Det får räcka med anmärkningen att många till synes ganska enkla system av differentialekvationer är omöjliga att lösa med analytiska metoder, varför man när det gäller kontinuerliga system ofta är hänvisad till *simuleringar* för att bli klar över deras detaljförlopp över tiden.[134] Det sistnämnda förhållandet är mycket relevant för ANN-

[134] Ett berömt exempel ur den klassiska fysiken är den måttligt komplicerade variant

forskningen eftersom man där mycket snabbt kommer upp i en komplexitetsgrad hos systemen som motstår alla försök till analytiska lösningar av differentialekvationer. Simuleringar är då ofta en bra utväg, dock inte den enda, eftersom man inte sällan kan bevisa t.ex. allmänna stabilitetsegenskaper hos ett system utan att kunna ställa upp en explicit ekvation för dess förlopp över tiden. Men dessa komplikationer och möjligheter är något som faller utanför ramarna för vår framställning.[135]

Illustrationer av kontinuerliga systems dynamik

Man har när det gäller kontinuerliga system tillgång till (minst) tre olika sätt att illustrera utvecklingen över tiden. Dels kan man, om man har tillgång till tillståndsekvationen, rita in variabelvärdena som explicita funktioner av tiden. Som exempel väljer vi den idealiserade pendeln från ekvationerna 7.3.4–7.3.5 ovan, med $\omega = 1$ och $\omega_0 = 0{,}1$.



Figur 50. Tillståndsdiagram över en linjär pendel. Tiden är på den horisontella axeln. Heldragen linje: vinkel (radianer) mot vertikalen. Streckad linje: vinkelhastighet (radianer/sekund).

Bara en del av punkterna i tillståndsrummet genomlöps när systemet startas från ett visst initialtillstånd, och endast dessa punkter kommer med i tillståndsdiagrammet. Ett annat sätt att illustrera systemets dynamik i det möjliga tillståndsrummet är genom ett *flödesdiagram.* Ett sådant diagram omfattar inte tiden som en dimension, men genom pilar i det markerar man i vilken *riktning* tillståndsvektorn rör sig i ett representativt urval punkter. Det antyder därför hur systemets variabler tenderar att förändra sig i en godtycklig punkt i tillståndsrummet. Även föränd-

av ekvationssystemet (7.3.8)–(7.3.9) som uppstår när man har att göra med tre kroppar som rör sig i tre dimensioner – det så kallade ”trekropparsproblemet”.

[135] För vidare läsning rekommenderas Wilson (1999) och Izhikevitch (2007).

ringshastighetens *storlek* i olika punkter kan representeras i flödesdiagrammet.

Läsaren inser genast, att vår illustration av kulan i stuprännan (figur 49 ovan) kan tolkas som en del av ett flödesdiagram. Figur 51 är motsvarande diagram för den enkla linjära pendeln, där x är vinkeln och $\dot{x}$ (utläses: x-prick) är vinkelhastigheten.

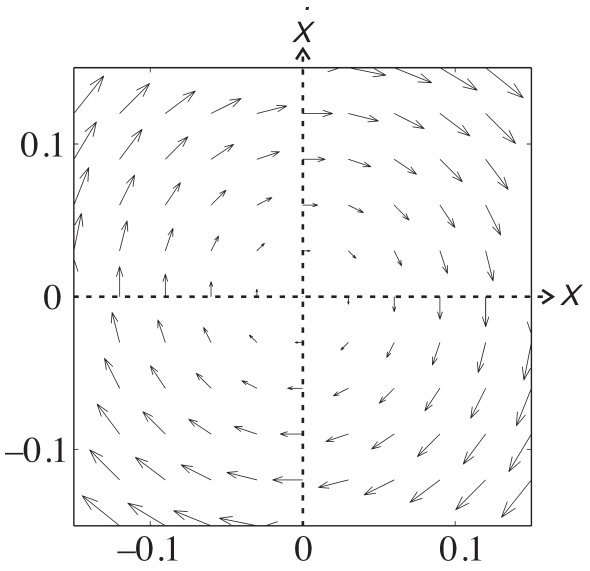
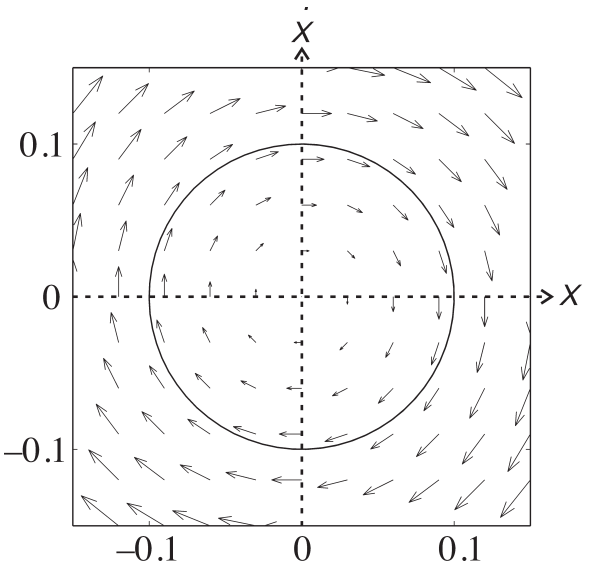


Fig 51. Ett flödesdiagram över den odämpade linjära pendeln. x: vinkel mot vertikalen. $\dot{x}$: vinkelhastighet.

Ett viktigt komplement för system med två variabler är *fasdiagrammet*, där man också eliminerat tiden men nu visar projektionen av systemets trajektoria (bana) på tillståndsrummet, givet ett visst initialtillstånd. Ur ett fasdiagram kan man inte, som ur tillståndsdiagrammet, avläsa hur fort systemet rör sig eller ens i vilken riktning. Därför är det lämpligt att komplettera det med ett flödesdiagram.

Figur 52 är således ett fas- och flödesdiagram för den enkla pendeln, släppt fri från en viss vinkel.



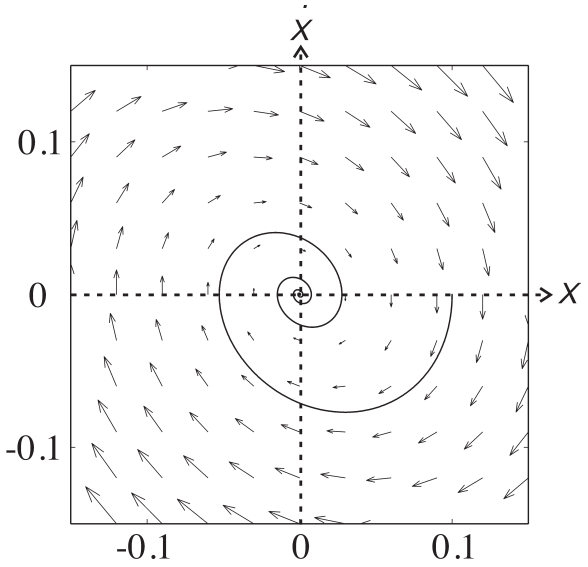
Figur 52. Ett fas- och flödesdiagram över den enkla linjära pendeln; x är vinkeln och $\dot{x}$ är vinkelhastigheten.

Fasdiagram lämpar sig särskilt väl för *animerade* framställningar av ett systems dynamik. I en sådan framställning är ju tiden återigen representerad, nämligen genom den tid som animeringen själv löper i. Men även de statiska fasdiagrammen utgör, liksom flödesdiagrammen, goda hjälpmedel för att förstå kontinuerliga system med två variabler.

Vad vi är särskilt intresserade av att förstå just nu är vilka attraktorer ett sådant system kan ha. Av figur 51 (eller 52) kan man utläsa att den linjära, odämpade pendeln alltid hamnar i en gränscykel. I figur 53 visas fas- och flödesdiagrammen för en *dämpad linjär pendel*, som lyder ekvationen

$$\frac{d^2x}{dt^2} = -x - b\frac{dx}{dt} \qquad (7.3.11)$$

där b är en friktionskonstant.



Figur 53. Den dämpade linjära pendelns punktattraktor. x: vinkel mot vertikalen. $\dot{x}$: vinkelhastighet.

Systemet går enligt diagrammet alltid till en punktattraktor, vilket stämmer med den vardagliga erfarenheten att en verklig, fritt svängande pendel med friktion till slut stannar (i ett nästan vertikalt viloläge).

Hur kan en *kontinuerlig* attraktor se ut i ett fas- eller flödesdiagram? Figur 54 visar ett exempel som är valt med en viss baktanke, nämligen flödesdiagrammet för det system som lyder ekvationerna:

(7.3.13) $$\frac{dx}{dt} = C(y - x)$$

(7.3.14) $$\frac{dy}{dt} = C(x - y)$$

där C är en positiv konstant. Det framgår av figuren, att attraktorn ifråga definieras av linjen y = x. Detta följer också omedelbart om man sätter båda tidsderivatorna till 0.

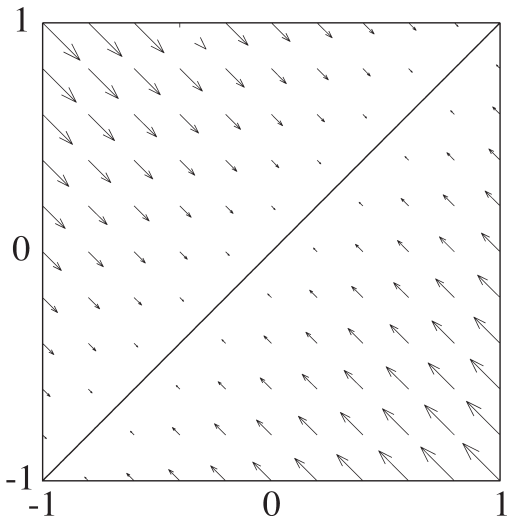


Fig. 54. En linjeattraktor. Förklaring: se text och ekvationerna 7.3.13–7.3.14.

Vi har alltså att göra med ett system som har samma kontinuerliga attraktor som Seungs par av linjära neuron. Det skiljer sig dock på ett viktigt sätt från det system som beskrivs av ekvationerna 7.3.1–7.3.2 ovan, nämligen därigenom att det gradvis och med avtagande hastighet närmar sig sina jämviktspunkter. Det kommer därför aldrig att oscillera. Med lämpligt val av konstanten C (stor!) kommer det kontinuerliga systemet dock att approximera det diskreta i det avseendet, att det givet en insignal a i neuronet X snabbt ”lär sig” att efter insignalens slut upprätthålla en nivå som ligger mycket nära (a, a).

Efter dessa förberedelser är vi mogna att ge oss i kast med några andra intressanta alternativ. Det allra närmaste stycket är ganska abstrakt, men det går att hoppa över det och ändå förstå resten av detta avsnitt.

Villkor för lärande kontinuerliga attraktorer

Huvudpoängen med den enkla linjära modellen för graderad kontroll av ögonrörelser är att det system som den beskriver kan ”minnas” vilken kortvarig signal som helst inom systemets operationsområde. Den skiljer sig därigenom från modeller med diskreta punktattraktorer, typ Hopfieldnätet, som visserligen kan ”minnas” en temporär input genom att bli kvar i samma excitationsmönster, men som ju är betydligt mer begränsat i fråga om *vad* det i denna mening kan minnas. En viktig fråga är nu om andra typer av system än de som vi just beskrivit också har förmågan att minnas valfria insignaler. Svaret är ja. Vi ska här kortfattat beskriva någ-

ra abstrakta villkor för en viss klass av sådana system, för att sedan skissera några relevanta biologiska tillämpningar.[136]

Antag ett tvådimensionellt system, där variablerna x och y interagerar med varandra enligt (de ännu inte specificerade) ekvationerna

$$\frac{dx}{dt} = f(x, y) \tag{7.3.15}$$

$$\frac{dy}{dt} = g(x, y) \tag{7.3.16}$$

Vi antar också att systemet har en linjeattraktor, och att denna kan beskrivas av en *strikt monoton* funktion $y = A(x)$, sådan att:

(Lx) Om en viss input x_0 ges till systemet under tillräckligt lång tid, dvs. om värdet på x fixeras till x_0 under denna tid, så kommer systemet att närma sig punktattraktorn $[x_0, A(x_0)]$.

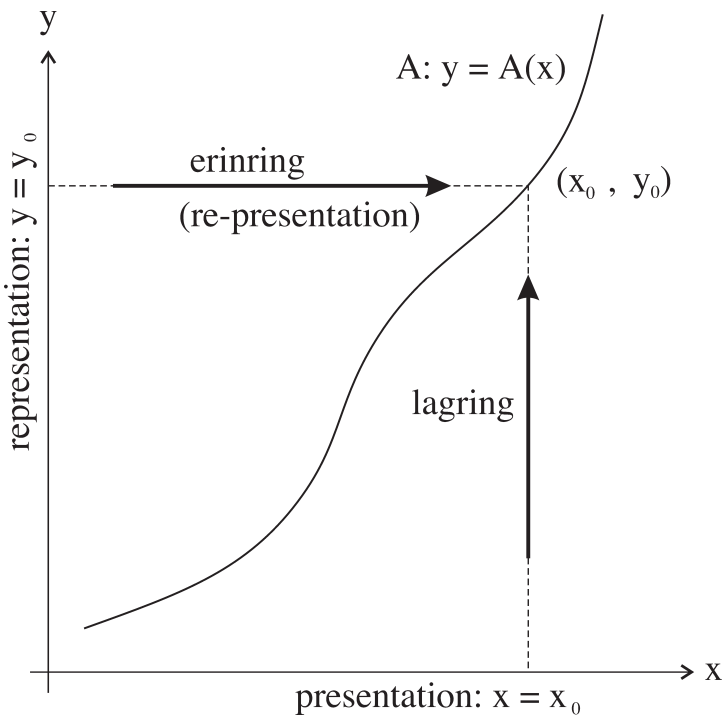
Villkoret (Lx) beskriver just systemets förmåga att ”minnas” en insignal x_0 som givits till det under tillräckligt lång tid. Av en viss anledning, som vi strax ska förklara närmare, önskar vi också att systemet har motsvarande minnesegenskap för variabeln y, det vill säga:

(Ly) Om en viss input y_0 ges till systemet under tillräckligt lång tid, dvs. om värdet på y fixeras till y_0 under denna tid, så kommer det att närma sig punktattraktorn $[A^{-1}(y_0), y_0]$.[137]

Ett system som uppfyller Lx och Ly sägs här ha en *lärande kontinuerlig attraktor*. Innan vi går vidare kan det vara värt att notera, att såväl Seungs enkla modell i 7.3.1–7.3.2 som det system som har ekvationerna 7.3.13–7.3.14 uppfyller båda dessa krav. Funktionen A är i båda fallen identitetsfunktionen.

I figur 55 illustreras ett system som uppfyller Lx och Ly. I figuren förekommer också ett flertal beteckningar som antyder hur modellen kan anknytas till inlärningsteori.

[136] Jämför också Malmgren (2002).
[137] A^{-1} betecknar inversen till funktionen A.



Figur 55. Presentation och representation i ett tvådimensionellt system med en lärande kontinuerlig attraktor.

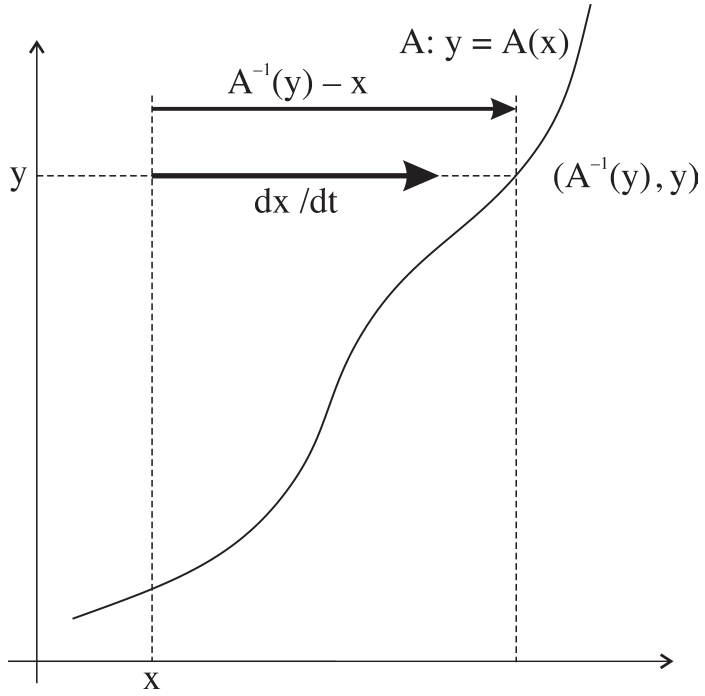
Vi vill alltså föreslå att läsaren tänker på systemet (x, y) som det neurala underlaget för *presentation* (varseblivning), *representation* och *re-presentation* (mental simulering). Jämför avsnitt 1.4 och 2.3 ovan! Variabeln x är presentationsvariabeln, som kan styras av signaler från sinnesorganen; variabeln y är representationsvariabeln, som istället kan styras från högre neurala centra. Båda variablerna antas följa ekvationerna 7.3.15–7.3.16, men bara i den mån de *inte* styrs av signaler från omvärlden respektive högre centra.

Vi ska närmast visa, att ovanstående specifikation (alltså genom Lx och Ly) av systemet är ekvivalent med antagandet att det finns en strikt monoton funktion A sådan att:

(L'x) $\frac{dy}{dt}$ har överallt samma tecken som [A(x) – y)];

(L'y) $\frac{dx}{dt}$ har överallt samma tecken som [A^{-1}(y) – x].

En konsekvens av L'x och L'y är ju att båda tidsderivatorna är lika med 0 när punkten (x, y) uppfyller sambandet y = A(x). Detta är dock inte i sig tillräckligt för att systemet ska ha de "minnesegenskaper" som vi tillskrivit det. De egenskaperna följer istället ur de angivna villkoren på tidsderivatorna när (x, y) *inte* ligger på grafen för A. För att inse detta, kontemplera figur 56. Den illustrerar L'y för det fall där a är en monotont *växande* funktion. Representationsvariabelns värde hålls konstant vid en viss nivå y från ett utgångsläge för systemet till vänster om grafen för A. Den tjocka pilen markerar den väg som systemet måste följa till attraktorpunkten $(A^{-1}(y), y)$.



Figur 56. Villkor på tidsderivatan för x i ett system med en lärande kontinuerlig attraktor. För närmare förklaring, se text.

Ett nödvändigt och tillräckligt villkor för att systemet över tiden ska kunna röra sig fram till grafen för A är förstås, att tidsderivatan för x är positiv under hela den väg som antyds av den kraftiga pilen. Annars stannar ju systemet på vägen. Hade startpunkten legat till höger om grafen för A skulle tidsderivatan av x istället behövt vara negativ. Och just dessa två fakta är det som sammanfattas i L'y.

Ett helt analogt resonemang gäller för L'x och för det fall då A är monotont *avtagande*.

Innan vi går vidare i det allmänna resonemanget kan vi återigen kollationera resultaten mot exemplet 7.3.13–7.3.14. Tidsderivatan dx/dt är i detta fall = C(y – x), men eftersom funktionen A nu är identitet så är detta ekvivalent med $C[A^{-1}(y) - x]$. Denna storhet har förvisso samma tecken överallt som $[A^{-1}(y) - x]$. Motsvarande gäller för dy/dt.

Villkoren L'x och L'y är ju ganska abstrakta, och det är inte omedelbart uppenbart vilka system som uppfyller dem. Vi ska därför formulera om dem, och börjar med att införa några nya beteckningar. Låt $f^{-+}(x, y)$ stå för att *funktionen f någonstans byter tecken från positiv till negativ när man gradvis ökar värdet på x, och någonstans går från negativ till positiv när man gradvis ökar värdet på y. Några andra teckenbyten får inte förekomma.* Innebörden av $f^{++}(x, y)$, $f^{+-}(x, y)$ och $f^{--}(x, y)$ torde omedelbart framgå.

Vi kan nu beskriva de förhållanden som illustreras i figur 56 på följande enkla sätt:

(L_0) $\quad \frac{dx}{dt}$ har alltid motsatt tecken mot $\frac{dy}{dt}$;

(L+) $\quad \frac{dx}{dt} = f^{-+}(x, y)$.

Av L_0 och L+ följer dels att $dy/dt = g^{+-}(x, y)$, dels att derivatornas teckenbyten sker i gemensamma punkter (derivatorna är noll samtidigt). Eftersom det bara sker ett teckenbyte per derivata och riktning definierar dessa punkter grafen för en monoton funktion. Denna måste i sin tur vara växande om teckenbytena ska ha den riktning som L+ anger. Det följer också av L+, att tecknen för dx/dt och dy/dt uppfyller villkoren L'x och L'y ovan. L_0 tillsammans med L+ utgör med andra ord *nödvändiga och tillräckliga villkor för existensen av en monotont växande funktion A, som är en lärande kontinuerlig attraktor till systemet (x, y).*

Genom ett helt analogt resonemang kan man visa att villkoren

(L_1) $\quad \frac{dx}{dt}$ har alltid samma tecken som $\frac{dy}{dt}$;

(L–) $$\frac{dx}{dt} = f^{--}(x, y)$$

utgör nödvändiga och tillräckliga villkor för existensen av en *avtagande* monoton funktion A, som är en lärande kontinuerlig attraktor.

Något som är värt att observera är att det i formlerna ovan inte någonstans figurerar ett villkor av typen $dx/dt = f^{++}(x, y)$. Intuitivt kan man förstå detta som att ett system med för mycket positiv feedback inte kan ha en kontinuerlig attraktor.

Utbytessystem

Avsikten med det föregående avsnittet var framförallt att visa, att många typer av system av två variabler kan ha lärande kontinuerliga attraktorer. Resonemangen kan på ett uppenbart sätt generaliseras till system med fler variabler, men vi ska inte ägna oss åt det här. Istället ska vi försöka konkretisera situationen genom att beskriva några system som uppfyller de abstrakta villkoren från föregående avsnitt. Av särskilt intresse är kanske de system som uppfyller L–, eftersom dessa sällan har diskuterats i litteraturen. Men dem spar vi till lite senare; först ska vi beskriva en enkel men mycket generell tillämpning av L+.

Ett ämne som diffunderar över en barriär, t.ex. ett biologiskt membran, gör det som regel med en hastighet som står i proportion till skillnaden mellan koncentrationen på ”hitsidan” respektive ”bortsidan” av membranet. Det är också så, eftersom det är fråga om ett *utbyte,* att den hastighet med vilken koncentrationen av ämnet avtar på den ena sidan är proportionell mot den hastighet med vilken koncentrationen tilltar på andra sidan. Det vill säga, om x är koncentrationen på ena sidan barriären och y är koncentrationen på andra sidan, så gäller:

(7.3.17) $$\frac{dx}{dt} = a(y - x)$$

(7.3.18) $$\frac{dy}{dt} = -b\frac{dx}{dt}$$

där a och b är positiva konstanter. Men dessa ekvationer innebär att villkoren L_0 och L+ ovan är uppfyllda. Ett diffusionssystem som uppfyller

7.3.17–7.3.18 – vi talar hädanefter om denna process som *enkel diffusion* – har alltså en kontinuerlig attraktor.

Många andra fysikaliska och kemiska system uppfyller samma, eller mycket liknande, villkor som enkel diffusion. Betrakta först ett system där ett stort kärl kan fyllas till en godtycklig nivå genom ett litet stigrör.

Figur 57. Ett hydrauliskt lärande system. Förklaring: se text.

Om vi här låter x och y stå för nivåerna i det stora kärlet respektive stigröret, så gäller 7.3.17 och 7.3.18 när kranen mellan de två kärlen är öppen.

En stor och viktig klass av system som har liknande kontinuerliga attraktorer definieras av reversibla reaktioner i kemin. Betrakta exempelvis

(7.3.19) $H_20 + HCl \leftrightarrows Cl^- + H_3O^+$

Här kan vi t.ex. sätta x = koncentrationen av HCl och y = koncentrationen av H_3O^+, varvid följande gäller:

(7.3.20) $$\frac{dx}{dt} = f^{-+}(x, y);$$

(7.3.21) $$\frac{dy}{dt} = -\frac{dx}{dt}$$

vilket gör att L_0 och L+ är uppfyllda.

Innan vi går vidare ska det poängteras, att ekvationssystemet 7.3.17–7.3.18 bara är ett *linjärt specialfall* av alla de system som har kontinuerliga attraktorer genom att uppfylla L_0 och L+. Med andra ord, *en diffusion eller en kemisk utbytesreaktion kan mycket väl ha en påtagligt ickelinjär dynamik och ändå ge en kontinuerlig attraktor.*

Nåväl, vad har dessa system, vare sig de är linjära eller ej, att göra med *inlärning?* Kanske en hel del. För att bereda marken rent begreppsligt, betrakta först det hydrauliska systemet i figur 57. Volymen i stigröret är liten i jämförelse med volymen i den stora behållaren, och det har två implikationer för hur systemet beter sig. För det första, om nivåerna i de två delarna är olika och man lämnar systemet åt sig självt, så kommer jämvikten som uppnås att ligga mycket nära den ursprungliga nivån i den *stora* behållaren. För det andra, om man envist fyller på stigröret till en förutbestämd nivå, så kommer nivån i behållaren så småningom att anpassa sig till denna. Låt oss nu tänka på detta i termer av presentation och representation (och jämför figur 57). Att hälla på vatten i stigröret under en period är att *presentera* en konstant stimulus. När vattnet så småningom har stigit till samma nivå i behållaren har en *representation* av denna stimulus uppstått. Om vi sedan startar med en godtycklig nivå i stigröret och låter slutnivån i detta bestämmas av utbytet med den stora behållaren, så *återskapas* (*re-presenteras*) (nästan exakt) den stimulus som systemet tidigare varit utsatt för. Har vi dessutom en mekanism för att öppna och stänga kranen kan vi återskapa denna stimulus vid ett valfritt, senare tillfälle.

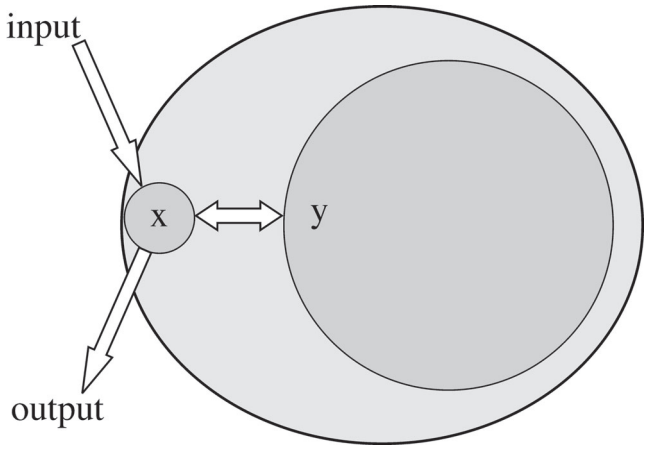
Det låter sig fortfarande sägas att vi rör oss på en rent metaforisk nivå. Här är två mer relevanta biologiska spekulationer.

För ett par år sedan visade en forskargrupp (med bl.a. svensk förankring) att en enskild nervcell kan ”minnas” den konstanta elektriska signal som den fått under en sammanhängande period.[138] Beroende på styrkan hos input som givits och den tid som expositionen varat bibehålls den förändrade membranpotentialen under en tid som är mycket längre än den som är aktuell t.ex. i samband med postsynaptiska potentialer.

Mekanismen för detta ”cellulära minne” är ännu så länge oklar, men man kunde tänka sig att rent principiellt förklara skeendet med kontinuerliga attraktorer i en diffusionsmodell.[139] Figur 58 nedan föreställer en cell, och de två cirklarna är två compartments (avdelningar i cellen) av mycket olika storlek mellan vilka ett visst ämne kan diffundera. Antag att koncentrationen x av ämnet i det lilla utrymmet bestämmer cellens membranpotential (”output”), och att x i sin tur kan direkt styras av den elektriska signal cellen får utifrån (”input”). När det inte finns någon input till

[138] Egorov et al. (2002).
[139] För argument *mot* att förklara experimentresultaten i termer av kontinuerliga attraktorer, se Franzen et al. (2006).

cellen styrs x dock av diffusionen från det stora utrymmet och koncentrationen y av ämnet i det.



Figur 58. Ett diffusionsbaserat minne hos en enskild cell. Förklaring: se text.

Man inser med lite eftertanke, att detta system kommer att bete sig helt analogt med vår hydrauliska modell ovan. Efter att ha utsatts för en konstant input under en period kommer systemet att ha uppnått en jämvikt där y = x, och när denna input upphör kommer systemet – i kraft av att compartments är så olika stora – att kunna upprätthålla samma output under en avsevärd period. Vidare inser man snabbt, att en kemisk jämviktsreaktion med lämpliga egenskaper kan fylla samma funktion som diffusionen gör i figur 58.

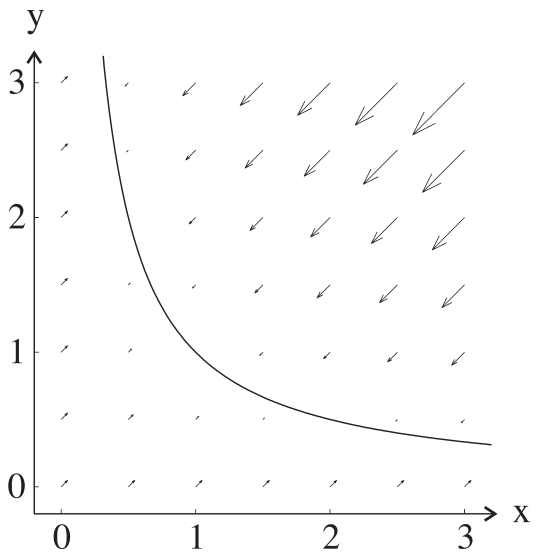
Negativa bilder och monotont avtagande attraktorer

Vi ska nu diskutera kontinuerliga attraktorsystem som uppfyller L_2 och L– (se ovan). Det vill säga, derivatorna dx/dt och dy/dt ska alltid ha *samma* tecken. Dessutom ska båda byta tecken en och endast en gång när man ökar x eller y, och då från att vara positiva till att vara negativa. För att läsaren ska bekanta sig med denna typ av system ska vi börja med att titta på det som styrs av följande ekvationer:

(7.3.22) $$\frac{dx}{dt} = (1 - xy);$$

(7.3.23) $$\frac{dy}{dt} = (1 - xy).$$

Detta system har en kontinuerlig attraktor som beskrivs av ekvationen y = 1/x, och dess flödesdiagram ser ut som i figur 59.



Figur 59. En monotont avtagande kontinuerlig attraktor. Förklaring: se text och ekvation 7.3.22–7.3.23.

Om vi återigen tänker i termer av presentation och representation så kan man säga, att ett dylikt system representerar en presentation x genom dess invers 1/x.

Det finns en uppenbar lärdom att hämta här när det gäller teorier om mental representation. En klassisk teori säger ju att dylik representation sker genom likhet – i moderna termer: som om vi hade fotografiska bilder i huvudet. Men om man tror att representationer har som funktion att *kunna re-presentera*, inser man att det skulle gå lika bra med fotografiska *negativ*! Det viktiga är ju, att man via representationen kan *återskapa* presentationen. Och det kan man som bekant göra med fotografiska negativ. Närmare bestämt inverterar man bilden en gång till – precis som man gör med 1/x.

Det finns flera andra enkla tvådimensionella system för representation via inverser. Ett exempel ges av funktionen y = 1 – x. Ett av de många system som har denna linje som attraktor är:

(7.3.24) $$\frac{dx}{dt} = 1 - x - y$$

(7.3.25) $$\frac{dy}{dt} = 1 - x - y$$

vilket omedelbart inses, liksom att ekvationernas högerled kan multipliceras med en godtycklig konstant utan att systemet får en annan attraktor. Varje avtagande *självinvers* funktion, dvs. en avtagande funktion f sådan att f(f(x)) = x, definierar på samma sätt en linjeattraktor för en hel klass av system.

Självinversa funktioner innebär en påtaglig *symmetri,* både i det att dx/dt = dy/dt och därigenom att den funktion som definierar derivatorna är symmetrisk i variablerna x och y. Inget av dessa villkor är nödvändigt för att det ska finnas en avtagande kontinuerlig attraktor, men villkoren är intressanta därigenom att de antyder att *symmetriska biologiska system som fungerar genom ömsesidig inhibition* kan misstänkas ha kontinuerliga attraktorer. Lägg särskilt märke till att medan det bara finns en självinvers växande funktion, nämligen y = x, så finns det oändligt många olinjära, självinversa, avtagande funktioner, nämligen alla de som är symmetriska kring linjen y = x. Om biologiska nervcellers olinjaritet på något sätt kan passas in i detta mönster är en öppen fråga, men om svaret på frågan är *Ja*, så innebär det att nervceller skulle kunna spara graderade aktivitetsnivåer i en symmetrisk inhibitorisk krets.

8. Kompetitiva nätverk

8.1 Kooperation och kompetition i neurala nätverk

Interaktionen mellan elementen i ett verkligt eller artificiellt neuralt nätverk kan ibland på ett naturligt sätt beskrivas genom termerna ”kooperation” och ”kompetition” – alltså samarbete respektive konkurrens. Termerna kan användas dels för att beteckna speciella former av interaktion mellan enheter i samband med att de reagerar på en inputsignal (som kan, men inte behöver, vara gemensam för alla enheterna), dels som beskrivningar av hur output från en uppsättning enheter integreras till en kollektiv respons. Inte sällan är båda termerna tillämpbara samtidigt; man kan till exempel tala om en kooperativ/kompetitiv organisation av output (se avsnitt 2.3). I detta kapitel skall vi koncentrera oss på kooperativ och kompetitiv organisation av ett nätverks omedelbara respons på input.

Lateral inhibition

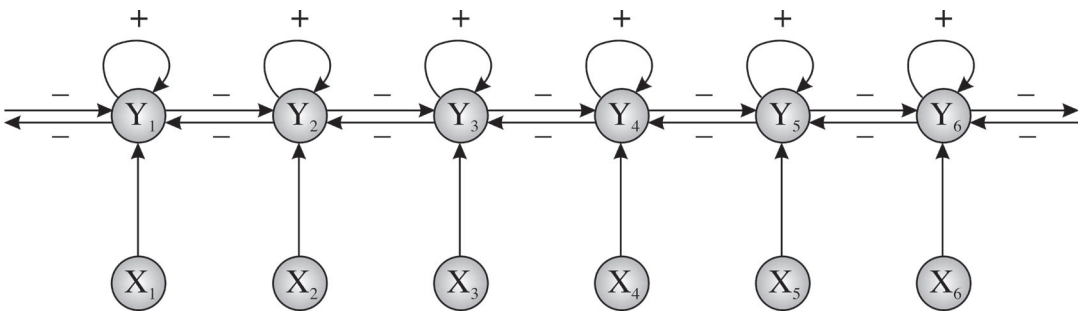
En vanlig typ av kompetitiv organisation i biologiska nätverk är den mekanism som kallas *lateral inhibition*. För att kunna förklara vad detta fenomen innebär och vilken biologisk relevans det (antagligen) har måste vi först skissera en bakgrund.

I de perceptuella system som ägnar sig åt representation av någon form av rumsliga sammanhang, till exempel de visuella och somatosensoriska systemen, hittar man vad man brukar kalla *topologiska kartor*. Dessa förefinns på olika nivåer av nervsystemet, dvs. både i hjärnbarken och i lägre omkopplingsstationer för informationen från receptorerna. Begreppet topologisk karta innebär att de rumsliga relationerna mellan neuronen på en viss nivå avspeglar rumsliga relationer på en lägre nivå, och i sista hand rumsliga relationer mellan stimuli. Ett välkänt exempel är den kortikala somatosensoriska kartan (i området strax bakom hjärnbarkens centralfåra), som finns avbildad i många framställningar av nervsystemets funktionella anatomi. Det framgår av dessa avbildningar (t.ex. handflatans storlek på kartan jämfört med rygghudens) att avstånds-

förhållanden bevaras dåligt, medan ordningsrelationer bevaras bättre; det är just detta som man vill betona med termen ”topologisk” (till skillnad från ”topografisk”). Dylika topologiska kartor finns alltså på många sensoriska nivåer, och är regel även på den motoriska sidan.

Nu är det inte så, att hierarkierna av topologiska kartor typiskt innebär att informationen från närmast lägre nivå avbildas entydigt, enhet för enhet, på närmast högre nivå. I så fall hade man kunnat betrakta kartorna på vägen mellan sinneorganen och hjärnbarken som helt passiva omkopplingscentra. Istället är det så, att varje omkoppling också innebär en omorganisation. Till viss del är denna omorganisation av en natur som gör att själva egenskapen att vara en *karta* mer eller mindre förstörs, eller åtminstone blandas ut med andra principer för representation. För att ta ytterligare ett välkänt exempel, så reagerar många celler i synbarken särskilt starkt på vertikala linjesegment som rör sig horisontellt. Detta beror uppenbarligen på att signaler om lokala händelser på näthinnan successivt integrerats medan de passerat uppåt i det visuella systemet, och gör att cellerna ifråga förlorat sin karaktär av punkter på en karta över dessa händelser.

En annan typ av omorganisation bevarar karaktären av topologisk karta, men förändrar kartans egenskaper så att vissa drag hos den framhävs, medan annan information kastas bort eller åtminstone förlorar i betydelse. Den laterala inhibitionen är här mycket betydelsefull. Den innebär i korthet, att varje element på kartan skickar signaler som inhiberar (hämmar) aktiviteten hos elementets rumsliga grannar, och (eventuellt) att alla element skickar en signal som förstärker deras egen aktivitet. En schematisk framställning av lateral inhibition kan alltså se ut som i figur 60 (vi bryr oss inte om någon eventuell intern organisation hos det underliggande skiktet):



Figur 60. Lateral inhibition. Förklaring: se text.

Vi antar för exemplets skull att alla elementen i det kompetitiva skiktet (Y-skiktet) är helt linjära, alltså med aktivitet = nettoinput, att vikterna från X-skiktet till Y-skiktet och de självexcitatoriska vikterna alla är 1, att de inhibitoriska vikterna alla är –0.5 och att alla neuronen i Y-skiktet uppdateras synkront. Vad leder då en sådan organisation av ett neuronalt skikt till?

Jo, anta att X-lagret skickar den konstanta signalen (0.2, 0.2, 0.2, 0.4, 0.4, 0.4) och betrakta vad som händer i enheterna Y_3 och Y_4. I första processteget uppstår en aktivitetsvektor i Y-lagret som är identisk med denna inputvektor. I nästa steg påverkas Y-neuronen också av andra Y-neuron och av den självexcitatoriska signalen. Enheterna Y_3 och Y_4 kommer att få precis *samma* sammanlagda inhibition från sina grannar, nämligen 0.3, och hamnar (med hänsyn tagen till självexcitationen och aktiveringen från X-lagret) på en total aktivering av 0.1 respektive 0.5. Neuronen till vänster om Y_3 kommer att få *mindre* inhibition än Y_3 (nämligen 0.2), och deras resulterande aktivitet blir 0.2 (precis som i första processteget). Elementen till höger om Y_3 kommer att få *mer* inhibition än Y_3 (nämligen 0.4), och deras aktivitetsnivå blir 0.4 (också precis som i föregående steg). Slutresultatet blir alltså aktivitetsvektorn (0.2, 0.2, 0.1, 0.5, 0.4, 0.4).

Kontrastförstärkning och kantdetektion

Hur ska vi beskriva det som hänt? Jo, det kompetitiva lagret har *förstärkt en kontrast*, nämligen kontrasten mellan de tre första och de tre sista komponenterna i input. Accentueringen av skillnaden mellan aktiviteterna i Y_3 och Y_4, jämfört med skillnaden mellan nivåerna i X_3 och X_4, kan antas göra det lättare för resten av nervsystemet att reagera specifikt på övergången mellan de två segmenten av input. I mer kognitivistiska termer kan man säga att organismen har skaffat sig ett redskap för att *upptäcka gränser* för objekt i yttervärlden; en ingenjör skulle hellre säga att nervsystemet använder sig av *kantfilter*. Lateral inhibition har helt säkert fler viktiga uppgifter att fylla i nervsystemet, men det är inte orimligt att anta att det just är kantdetektion som mekanismen är till för när den återfinns tidigt i de perceptuella systemen.

Vår modell i figur 60 är förvisso kraftigt förenklad jämfört med den biologiska verkligheten, men illustrerar ändå väl den grundläggande princip som gäller för lateral inhibition.

Den rumsliga kartan i hippocampus

Det finns god experimentell evidens för att vissa celler i råttans hippocampus (”platscellerna”) genom sin aktivitet specifikt indikerar den plats i en bur som djuret befinner sig på – även när den sensoriska input inte ger tillräcklig information om detta.[140] Det har föreslagits en modell för detta, som bygger på ett kooperativt-kompetitivt nätverk med vikter som gör aktiviteten i och runt vinnarnoden självuppehållande efter det att stimulus upphört. Vi ska nu beskriva resultaten och modellen i lite mer detalj, dock utan att gå in på några matematiska formuleringar.

Mäter man aktiviteten i ett antal neuron av det nämnda slaget under omständigheter när råttan kan se var den befinner sig, finner man för varje given plats i buren att en eller högst ett fåtal celler visar en maximal respons. Andra neuron visar en något lägre aktivitet, andra en betydligt lägre och majoriteten är tysta. Dessa andra neuron kan istället ge maximal respons för andra platser i buren. De neuron som ger liknande responser för en given plats befinner sig *inte*, generellt sett, i närheten av varandra i hippocampus. Det verkar med andra ord inte som om det rör sig om en topologisk karta av den typ som vi nämnt ovan.

Man kan dock ordna cellerna med utgångspunkt från deras likheter ifråga om responser på stimuli, och då framträder åter en bild som påminner om en sådan karta. Det neuron som ger maximal respons för en viss lokalisation är i denna karta omgivet av neuron som ger en något lägre respons för denna lokalisation, men som istället ger maximalt svar när råttan befinner sig på något av de angränsande ställena i buren. I ett diagram ser aktivitetsmönstret för en given stimulussituation ut som en distinkt, lokal ”puckel” av aktivitet; man talar ofta om ”aktivitetspaket”.

Detta responsmönster vill man i den aktuella modellen förklara genom ömsesidigt *exciterande* laterala förbindelser, som tenderar att vara starkare ju ”närmare” varann (i diagrammet) två neuron befinner sig. Neuron på ”långt avstånd” från varann kan istället ha starka inhiberande förbindelser. Under ganska allmänna övriga villkor på vikterna tenderar inputmönster som har ett maximum i en viss nod att ge upphov till ett lokalt ”aktivitetspaket” av den typ som vi nyss beskrev. Det har också visats, att man med ett lämpligt val av vikter kan få aktiviteten i ett dylikt paket att vara självuppehållande. Nätverket kan med andra ord ”minnas” vilken

[140] O’Keefe & Nadel (1978). Rolls & Treves (1998), kap. 6, argumenterar för att de aktuella rumsliga koordinaterna beräknas i neocortex, inte i hippocampus.

plats som den sensoriska input senast indikerade. Om nätverket består av många neuron kan det därför approximera ett system med en kontinuerlig, tvådimensionell attraktor.[141] Kanske är det ett sådant nätverk som hjälper oss att hitta i sovrummet även efter det att vi släckt lampan (jämför simuleringsteorin för representation, avsnitt 1.3).

Närmast ska vi beskriva en annan typ av kompetitiv organisation. Denna gång gör vi det inte i första hand genom en hänvisning till biologin, utan med utgångspunkt i välkänt artificiellt nätverk: Kohonens SOM.

8.2 SOM – den självorganiserande kartan

Få artificiella neurala nätverk, med undantag av flerlagrade perceptroner med varianter av algoritmen ”back propagation of error” (se nästa kapitel), har fått så många föreslagna praktiska tillämpningar, bl.a. i medicin och biologi, som finländaren Teuvo Kohonens ”Self-Organising Map”, eller SOM.[142] Detta nätverk bygger på idén att rumsligt representera de väsentliga likheterna mellan inputdata i färre dimensioner – oftast två – än vad dessa data har från början. För att uppnå detta använder Kohonen ett nätverk där noderna från början har en *rumslig närhetsrelation* till varandra. Träningsalgoritmen jämkar underhand vikterna hos rumsligt näraliggande noder så att *liknande* inputs till slut representeras *intill* varann i nätverket. En karta över inputdomänen har därigenom konstruerats.

SOM klassificerar data utan att användaren förutbestämt klasserna, dvs. på ett icke-styrt sätt. Eftersom metoden bygger på likhet mellan data kan man tala om en i viss mening “naturlig” klassifikation. Vi har dock argumenterat för termen ”kategorisering” som beskrivning av vad SOM gör (se avsnitt 4.5), och skall hålla fast vid det språkbruket. I ett statistiskt perspektiv är SOM bara en av många metoder för “clustering” av data, men SOM har unika egenskaper, kanske särskilt som instrument för visualisering av data.

---

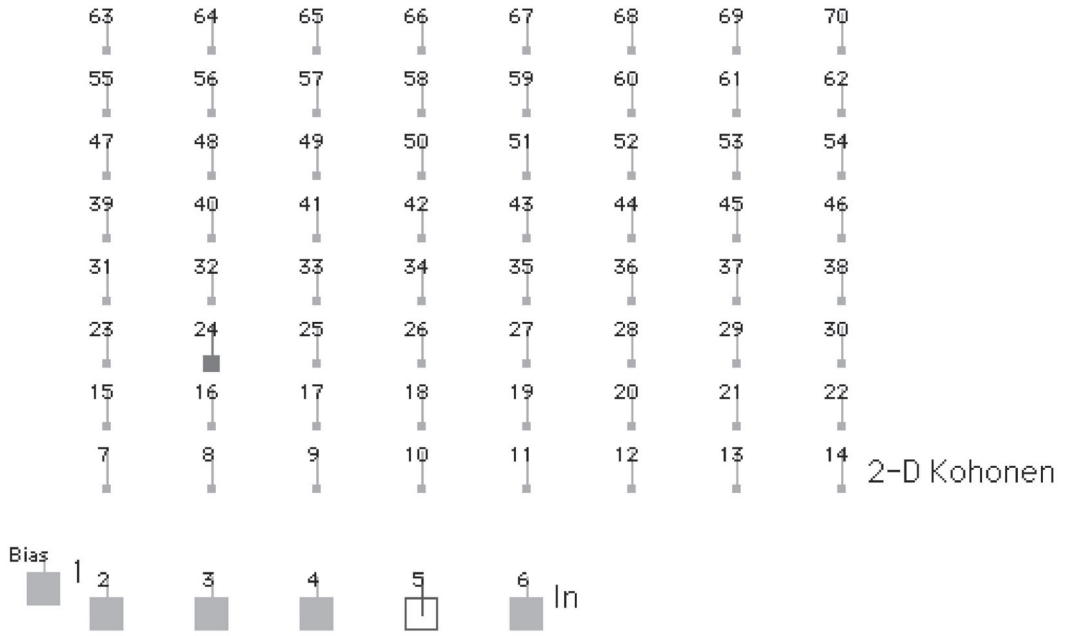
[141] För fler detaljer i modellen och för matematiska formuleringar, se Samsonovich & McNaughton (1997) samt Trappenberg (2002), kap. 9.

[142] En helt annan sak är hur många av dessa förslag som i praktiken har använts i någon nämnvärd utsträckning. Vad beträffar de medicinska applikationerna är svaret att mycket få, om ens någon, av dem har blivit klinisk rutin. Detta gäller även andra ANN-modeller som föreslagits för medicinska ändamål. Det finns många orsaker till detta stora glapp mellan teori och praktik. För en analys av en del av problematiken se Dybowski (2000).

Låt oss se på några detaljer i detta användbara nätverk. (Det kan vara en fördel att vara välbekant med innehållet i avsnitt 5.2.) SOM är ett feed-forwardnätverk med två skikt av enheter, det vill säga ett lager av förbindelser. Varje inputnod har förbindelser med varje nod i det andra, kompetitiva skiktet (också kallat Kohonenlagret), och vikterna mellan lagren slumpas ut vid träningens start. Kohonenlagret är som sagt organiserat i en rumslig struktur. Vi ska nu beskriva först hur noderna i detta lager aktiveras och sedan hur den inlärning går till, som gör SOM till en rumslig *karta* över inputmängden.

”The winner takes it all”

Kohonen-noderna sägs vara “kompetitiva” därför att de på ett visst sätt konkurrerar om att bli aktiverade av inputs. Endast en av noderna blir aktiv för en given input – ”the winner takes it all”. Figur 61 är en illustration av ett SOM med 5 inputs och 64 (8x8) kompetitiva enheter. Det har just fått en input, och en vinnarnod har utsetts.



Figur 61. Ett SOM där vinnaren (den fyrkantiga nod 24 i Kohonen-lagret) just utsetts. Skärmavbildning från Neural Works.

SOM aktiveras närmare bestämt som följer när en inputvektor presenterats: Den Kohonen-nod får aktiviteten 1 *vars vektor av vikter från inputlagret mest liknar inputvektorn ifråga*, medan alla de andra no-

derna får aktiviteten 0. Maximal likhet definieras genom att det euklidiska avståndet mellan vektorerna (mätt med Pythagoras sats) ska vara så litet som möjligt. En liten påminnelse kan vara på sin plats här, eftersom det kan kännas ovant att tala om likheter mellan en inputvektor och en viktvektor. En input och en uppsättning vikter på förbindelserna till en nätverksnod är ju helt väsensskilda saker! Matematiskt sett är de dock båda vektorer, och de har samma antal komponenter, varför det går att göra den erforderliga jämförelsen.

Nåväl, vi säger att den Kohonen-nod som får aktiviteten 1 (”vinnarnoden”) *representerar* ifrågavarande input; andra vanliga beskrivningar är att vinnarnodens viktvektor är en ”kodvektor” eller ”typvektor” för denna input. Uppenbarligen kan en Kohonen-nod i princip vara en kodvektor för en hel grupp av inputs. Man talar också om ”vektorkvantisering”, eftersom en potentiellt oändlig mängd inputvektorer på detta sätt kan få en förenklad representation i en ändlig mängd av typvektorer. Den inlärning som följer (se nedan) går ut på att anpassa kodvektorerna bättre och bättre till data, samtidigt som representationerna för input flyttas mellan Kohonen-noderna på ett sådant sätt att den ovan nämnda kartan uppstår. Men innan vi går in på inlärningsalgoritmen finns det mer att säga om den kompetitiva aktiveringen som sådan.

Är aktiveringsfunktionen i SOM biologiskt realistisk?

Idén om vektorkvantisering och kodvektorer anknyter till den teori om mental representation som brukar kallas ”template-teorin”, enligt vilken igenkänning av mönster i den mänskliga hjärnan tillgår som så att ett inputmönster matchas med avseende på sin likhet med ett antal typmönster (templates). Vi har redan beskrivit en nätverksmodell som kan sägas utföra sådan “template matching”, nämligen Hopfieldnätet, och har alltså sedan tidigare en tänkbar förklaring av mänsklig mönsterigenkänning till hands. Den självorganiserande kartan med kompetitiv aktivering erbjuder kanske en alternativ förklaringsram.

Men kan neuronal aktivering i hjärnan fungera på det sätt som Kohonen stipulerar för sitt SOM? Kan det mänskliga nervsystemet beräkna euklidiskt avstånd mellan en inputvektor och en uppsättning viktvektorer, och utser det en vinnare där detta avstånd är minst? Svaret på den frågan är: Den grundläggande algoritm som används i SOM (och för övrigt även i LVQ, Learning Vector Quantization, se avsnitt 9.3) har knappast en exakt biologisk motsvarighet, men det är lätt att ange en alternativ algoritm

som gör ungefär samma sak och som dessutom är mer biologiskt trovärdig. Med viss risk för upprepning av vad som sades i avsnitt 5.2 ska vi nu förklara detta närmare.

Tänk först på den alternativa formeln (ovan numrerad som 5.2.2) för skalärprodukt mellan två vektorer $\mathbf{v_1}$ och $\mathbf{v_2}$ som har längden $l(\mathbf{v_1})$ respektive $l(\mathbf{v_2})$ och bildar vinkeln α med varann:

(8.2.1) *Skalärprodukt (alternativ):* $\mathbf{v}_1 * \mathbf{v}_2 = l(\mathbf{v}_1) \cdot l(\mathbf{v}_2) \cdot \cos\alpha$

Denna formel implicerar, allt annat lika, att två vektorer som bildar en *liten* vinkel med varann (dvs. pekar åt ungefär samma håll) har en *större* skalärprodukt än om de bildar en större vinkel. Och, fortfarande allt annat lika, är det euklidiska avståndet mellan två vektorer mindre, ju mindre vinkeln mellan dem är. Skalärprodukt och euklidisk närhet följs alltså åt, och överensstämmelsen mellan dem blir exakt om alla vektorer är normerade till en viss längd, exempelvis 1.

Under förutsättning att alla inblandade vektorer har någorlunda lika *längd* återspeglar alltså *storleken på skalärprodukterna* mellan en inputvektor och de olika viktvektorerna ganska bra *euklidisk närhet* mellan denna inputvektor och dessa viktvektorer. Skalärprodukterna ifråga beräknas ju rutinmässigt i de flesta artificiella neurala nätverk, och den operationen är förmodligen en hyfsat realistisk modell för integrationen av den information som anländer till en nervcell.

Låt oss därför tänka oss att vi ger normerade inputvektorer till ett SOM, där vi bytt ut det vanliga Kohonen-lagret mot ett som fungerar enligt normala ANN-principer. Skalärprodukten mellan en inputvektor och en viktvektor som tillhör en Kohonen-nod är ju då inget annat än den nettoinput som kommer till denna nod. Om noderna har en positivt linjär eller sigmoid aktiveringsfunktion, kommer den nod i Kohonenlagret vars viktvektor ligger närmast inputvektorn att få den högsta aktiviteten. Det är sedan inte svårt att designa en biologiskt någorlunda realistisk algoritm som genom självexcitation och ömsesidig inhibition i det kompetitiva lagret (alltså den mekanism som vi nyss beskrev som *lateral inhibition*) resulterar i att just denna nod – vinnaren – slutligen får aktiviteten 1, medan övriga noder får aktiviteten 0.

Om input- och viktvektorer *inte* är normerade kommer det biologiska nätverket som vi just skisserat inte att fungera *precis* som SOM, men dess

verkningssätt kommer att *påminna* om SOMs. Detta nätverk kommer att göra en alternativ och, biologiskt sett, kanske mer naturlig kategorisering.

Inlärning i SOM

Låt oss nu, efter att ha givit det grundläggande funktionssättet hos Kohonen-noderna ett slags biologisk motivering, gå in på träningsalgoritmen för den självorganiserande kartan. Träningen går till som följer; steget (d) är det verkligt unika för SOM.

(8.2.2) (a) En input presenteras

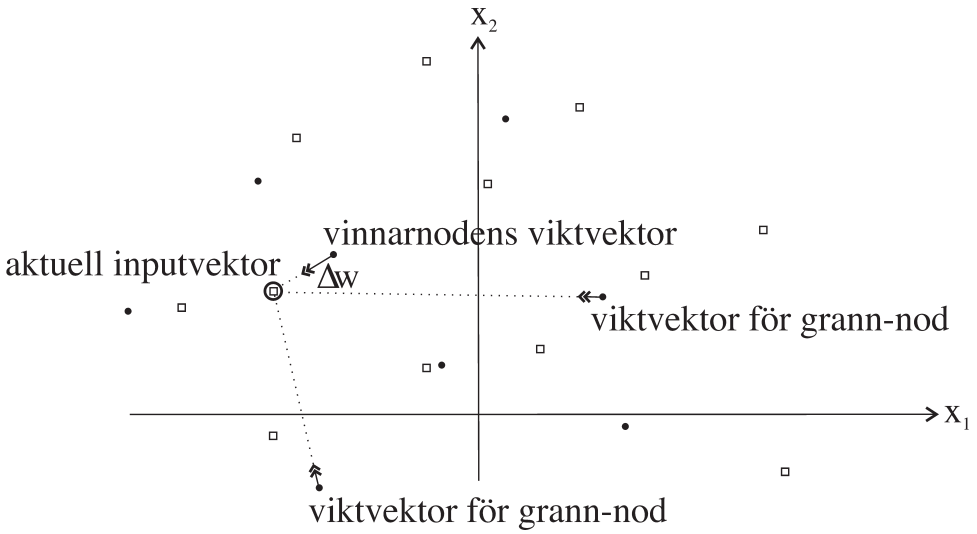
(b) Vinnaren utses (minimalt avstånd mellan input och viktvektor)

(c) Viktvektorn för vinnarnoden *makas närmare* inputvektorn ifråga

(d) Vikterna hos vinnarnodens *rumsliga grannar* makas också (lite) närmare denna inputvektor.

Proceduren (med en del tillägg som vi inte ska gå in på här) upprepas till dess att vikterna stabiliserats. Under träningens gång snävas kriterierna in för vilka noder som räknas som grannar, så att färre och färre noder berörs av steg (d).

Låt oss för att lättare kunna illustrera ett steg i algoritmen anta att också inputdata är tvådimensionella. Visserligen utför ett SOM för dylika data ingen sådan "dimensionsreduktion" som äger rum när data har högre dimensioner, men för att förmedla tanken bakom inlärningsprocessen duger exemplet bra. I figur 62 representeras några inputdata genom ofyllda kvadrater, medan viktvektorerna för några av Kohonen-noderna representeras av små fyllda cirklar. Pilarna illustrerar vad som händer då en input, markerad med hjälp av en större (ofylld) cirkel, presenteras första gången. Dels flyttas vektorn för vinnarnoden (den närmast belägna viktvektorn) ännu närmare inputvektorn, dels flyttas viktvektorn för vinnarnodens rumsliga grannar en liten bit närmare. Av dessa grannar representeras i figur 62 bara två. Observera att viktvektorerna för de rumsliga grannarna till vinnaren som regel inte alls ligger nära inputvektorn när SOM-nätverket inte är tränat, trots att vinnarens viktvektor gör det.

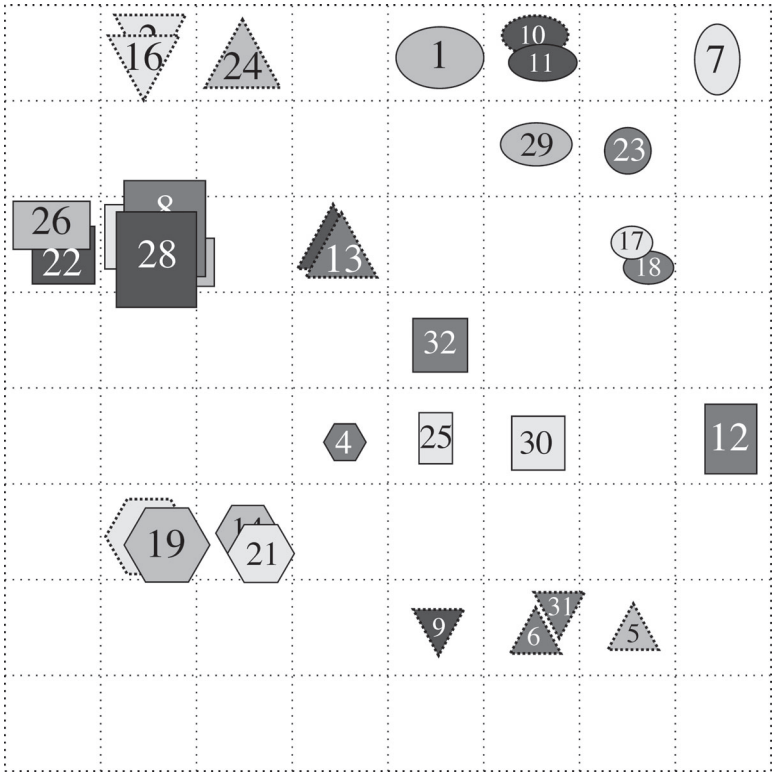


Figur 62. En del av ett steg i inlärningen hos ett SOM. De öppna fyrkanterna representerar inputvektorer, och de fyllda små cirklarna är viktvektorer för Kohonen-noderna. Δw representerar vinnarnodens viktändring.

Vad leder då detta steg till? Jo, momentet (c) i algoritmen (8.2.2) gör vinnarnoden *ännu bättre på att vinna på den input som den just "vann" på*. På sikt, och genom att samma nod förmodligen vinner även på andra inputs, åstadkommer detta moment normalt att nodens viktvektor till slut hamnar nära tyngdpunkten för en grupp av inputvektorer, som den representerar i egenskap av kodvektor.

Momentet (d) ger *vinnarens grannskapsnoder en något bättre chans att bli vinnare på inputs som liknar den just aktuella* – oavsett hur dåliga de var på att vinna på sådana inputs från början. Med upprepad användning av algoritmen kommer därför *liknande inputs oftare och oftare att representeras om inte av samma nod, så av fysiska grannar i nätverket.* Det är just så som "kartan" över inputdomänen uppkommer.

Här är ett exempel från ett övningsmaterial, där ett antal (artificiella) biologiska organismer som varierar i fem dimensioner – taggighet, ljushet, höjd, bredd och antal hörn (5 för ellips) – processats i ett 8 x 8-SOM. Organismerna har placerats ut i ett diagram, där positionerna för varje organism motsvarar den rumsliga placeringen i nätverket av den nod som representerar organismen (dvs. den nod som vinner på ifrågavarande inputvektor när träningen är avslutad).



Figur 63. SOM-karta över en inputdomän. Modifierat efter ett skärmfoto i NeuralWorks. Förklaring: se text.

I detta fall har man valt att låta kartan "gå runt" i kanterna, dvs. överkanten och nederkanten sitter egentligen ihop, liksom vänsterkanten och högerkanten. Med andra ord: vid användandet av algoritmen (8.2.2) på nätverket i figur 61 räknas t.ex. elementen 47 och 54 som grannar. Detta betyder att kartan måste vara en torus (en munk, eller en badring). Trianglarna upptill till vänster respektive nertill till höger i figur 63 befinner sig alltså egentligen relativt *nära* varann på kartan.

Kodningens betydelse för SOM

Beroende på vilka måttenheter man använder för de olika inputdimensionerna kan man få mycket varierande resultat av en kategorisering med SOM. Man kan lätt förstå varför det är så genom följande exempel:

A är 2 meter lång och väger 99520 gram
B är 1,95 meter lång och väger 99400 gram
C är 1,90 meter lång och väger 99500 gram

Ett Kohonen-nätverk som får input kodad på det nämnda sättet skulle fästa mycket stor vikt vid skillnaderna i vikt. I termer av euklidiskt avstånd ligger B längre från A än vad C gör, och det är mindre troligt att A och B hamnar i samma kategori än att A och C gör det. Kodar man däremot om grammen till kilo, så blir resultatet det motsatta.

Kodningen av data är således A och O när det gäller neurala nätverk för kategorisering. Men kodningen av inputdata har stor betydelse även för övervakad mönsterklassifikation, exempelvis när man arbetar med flerlagrade perceptroner, och kan där göra skillnaden mellan en bra och en dålig lösning av ett problem (se vidare kap. 9).

I detta sammanhang kan nämnas att normalisering av data inte måste innebära att information går förlorad. Man kan helt enkelt lägga till en extra vektorkomponent, som representerar den ursprungliga inputvektorns längd. (Nåja, det är inte *helt* enkelt, eftersom det är den resulterande vektorn som ska vara normaliserad, men det är inte särskilt svårt att lista ut hur det ska gå till.) Enheten för denna längd kan väljas godtyckligt beroende på vilken roll man vill att inputvektorernas längd skall spela i analysen. Detta är ett exempel på hur man kan vinna kodningsfördelar genom att projicera data på ett högredimensionellt rum.

Tekniska och biologiska tillämpningar av SOM

SOM är en enkel och intuitivt tilltalande metod för att sammanfatta annars svåröverskådliga datamängder på ett visuellt fattbart sätt, och har som redan nämnts fått ett stort antal sådana användningar. Metoden kritiseras ibland för att vara matematiskt inexakt. Det är sant i så måtto att de existerande konvergensresultateten för inlärningsproceduren inte är så generella som man skulle kunna önska, och (viktigare) att man inte kan dra några exakta inferensteoretiska växlar på resultatet av en analys med SOM. Men som heuristisk algoritm torde SOM ha ett stort värde, inte minst när det gäller att ge ett underlag för nödvändiga, snabba beslut utifrån stora och/eller svåröverskådliga datamängder. Ett exempel är sorteringen av Websidetexter med algoritmen WEBSOM.[143] Många utvidgningar av SOM har också föreslagits, bland annat för att kunna ta hand om spatio-temporala data på bästa sätt.[144] Med tanke på möjligheterna i dagens visualiseringsteknik kommer för övrigt *tredimensionella* SOM säkert att komma till praktisk användning ganska snart.

[143] Se Lagus et al. (2004), och sök upp WEBSOM på webben!
[144] T.ex. Wiemer (2003).

Slutligen ska man inte underskatta möjligheten att använda SOM för biologisk modellering. Givetvis måste man då börja med att ersätta beräkningen av euklidiskt avstånd med någon mer biologiskt realistisk variant, förslagsvis på det sätt som vi antytt ovan. Sedan får man leta efter exempel i nervsystemet på grupper av neuron som alla får input från precis *samma* inputenheter (vilket ju inte gäller för de vanliga topografiska kartorna i de perceptuella systemen), och som är inbördes organiserade med lateral inhibition (i två eller tre dimensioner). Om man hittar sådana neurongrupper, till exempel någonstans i hjärnbarken, finns det fog för att ställa upp hypotesen att de är ämnade för kategorisering av högdimensionella data genom avbildning av dem på en topologisk neuronal karta av lägre dimensionalitet.

8.3 Optimering med Hopfieldnät och SOM

En lite udda men riktigt rolig applikation av SOM är dess användning på ett klassiskt optimeringsproblem, nämligen TSP, *Traveling Salesman Problem*.

Ett optimeringsproblem är en uppgift där man vill att en viss storhet ska maximeras eller minimeras. Vissa sådana problem kan lösas analytiskt (dvs. man kan hitta en explicit formel för att räkna ut svaret), andra kan man tackla genom en stegvis algoritm – t.ex. av typ gradientnedstigning – som garanterat leder till svaret. Vi har redan tittat på ett optimeringsproblem som kan lösas på båda dessa sätt, nämligen att hitta minimum för felfunktionen i ett linjärt nätverk. Motsvarande problem för icke-linjära, flerlagrade nätverk kan också lösas med stegvisa algoritmer, även om man här aldrig kan vara helt säker på att hitta den bästa lösningen inom förutbestämd tid.

Åter andra optimeringsproblem går överhuvudtaget inte att lösa på något bra sätt, det vill säga sannolikheten är liten att de algoritmer man har hittar det optimala värdet inom en begränsad tid. Ofta beror detta på att en någorlunda fullständig lösning förutsätter att man går igenom ett mycket stort antal diskreta, uteslutande alternativ, så många att beräkningskraften hos tillgängliga datorer inte räcker till. TSP är ett sådant svårlöst optimeringsproblem. Det innebär att man skall hitta den kortaste resvägen genom ett antal städer, givet att resvägen mellan varje par av städer är känd. ”Resvägen” betyder här antalet kilometer mellan städerna, alternativt åtgången av tid, bensin eller någon annan resurs som man är in-

tresserad av att minimera. Problemet är av stor praktisk betydelse bland annat för flygföretag, som alltid måste arbeta på att mimimera resursåtgången i sin verksamhet. Därför finns det en efterfrågan på förbättrade algoritmer för att lösa TSP.

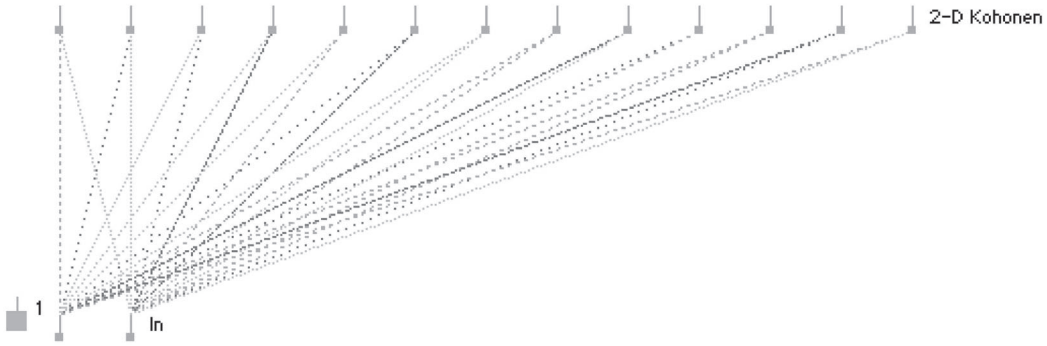
Det finns två uppslag till lösning med hjälp av artificiella neurala nätverk. Hopfield och Tank visade redan i mitten av 80-talet att man kan koda in problemets parametrar i ett Hopfieldnät på ett sådant sätt att nätverket, när det uppdateras, hittar en lokalt optimal lösning av problemet. Kodningen medför nämligen att nätverkets globala energi motsvarar det sammanlagda avståndet mellan städerna, och som vi vet går Hopfieldnätet alltid till ett lokalt minimum vad gäller energin.[145]

Det euklidiska TSP

Kohonens SOM kan användas för att lösa en enkel variant av TSP. Här måste städerna vara kodade med sina geometriska koordinater, inte med en avståndstabell, och man antar att resvägen definieras av det euklidiska avståndet på kartan. Detta – det euklidiska TSP – är en avsevärd förenkling av det ursprungliga problemet, eftersom det varken tar hänsyn till att vägar kan vara krokiga eller till att man måste köra långsammare på vissa vägar. Men det förenklade problemet, och dess lösning med SOM, kanske ändå kan vara av intresse i verksamheter där dessa hänsyn inte spelar *så* stor roll – t.ex. i vissa flygbolags planering.

Nåväl, hur löser man då det euklidiska TSP med SOM? Jo, man gör en *endimensionell* karta med två inputnoder och lika många Kohonen-noder som antalet städer. Det är lämpligt att låta kartan ”gå runt” i kanterna. Inputvektorerna skall vara städernas geografiska positioner, uttryckta som två koordinater i ett godtyckligt rätvinkligt koordinatsystem. Med 13 städer ser nätverket ut som i figur 64.

[145] Se Hopfield & Tank (1985). För en översikt över nya varianter av denna modell och andra neurala-nätverkslösningar av TSP, se Mérida-Casermeiro et al. (2001).



Figur 64. Travelling Salesman med SOM. Skärmfoto från NeuralWorks. Förklaring: se text.

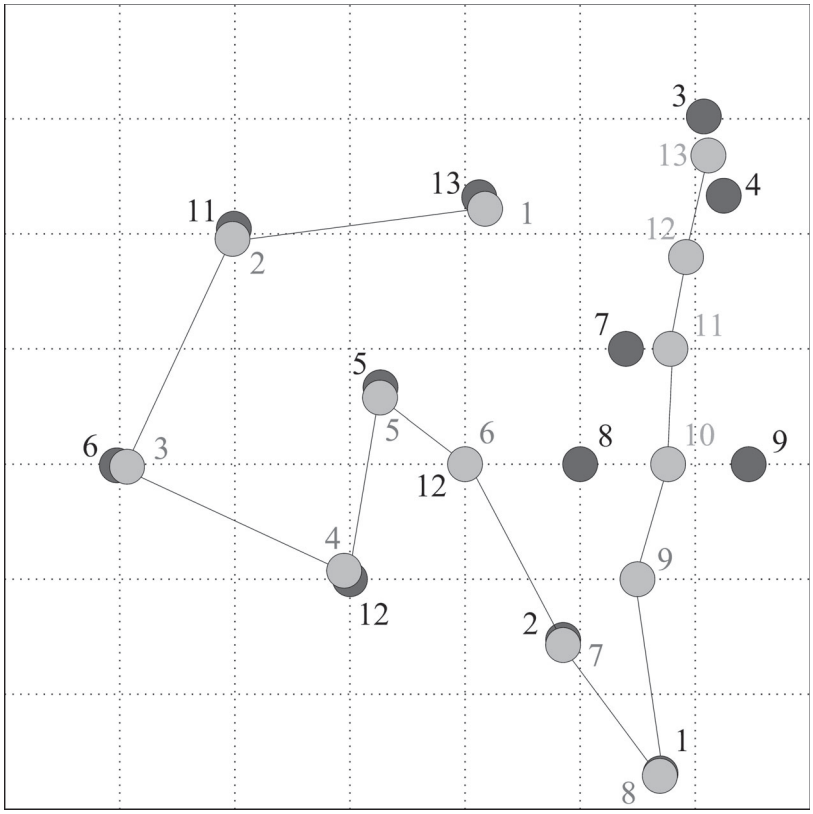
Sedan är det bara att träna nätverket med den vanliga SOM-algoritmen!

Hur fungerar nu detta, och varför ska vi tro att det ger oss ett bra förslag till lösning av det euklidiska TSP? Jo, SOM tenderar ju att representera *liknande* inputs, och därmed också liknande kodvektorer, i noder som är *grannar*. TSP-SOM kommer med andra ord att representera orter som ligger nära varann på den *geografiska* kartan i noder som ligger nära varann på *SOM-kartan.* Den ordning i SOM-kartan som bildas av städernas representationer kan således lämpligen tolkas som nätverkets förslag till (kort) resväg. För att hitta dessa representationer behöver man bara ge de olika inputs (alltså städernas geografiska positioner) i valfri ordning och notera vilken nod som vinner på respektive input.

Ett litet problem är att två städer inte sällan kommer att representeras i samma nod. Detta kan åtgärdas genom att man kör SOM med en högre ”samvetsfaktor”, en parameter som förvårar för noderna att vinna på flera inputs. Alternativt kan man finputsa den ursprungliga lösningen för hand (den kan ju sägas innebära att man ska åka genom båda städerna samtidigt, och det är ju inte helt enkelt).

Ett annat sätt att läsa ut nätverkets förslag till lösning av TSP är att tolka nodernas vikter som koordinater hos städer som skall besökas, fortfarande i den ordning som noderna ligger i nätverket. Då kan det istället hända att den föreslagna resvägen går mitt emellan två städer, vilket också uppmanar till nya försök med SOM om man inte nöjer sig med en manuell putsning. Figur 65 visar ett ganska typiskt resultat från ett försök med nätverket i figur 64; de grå cirklarna som är förbundna med en bruten linje definierar förslaget till resväg, och man ser att de oftast men

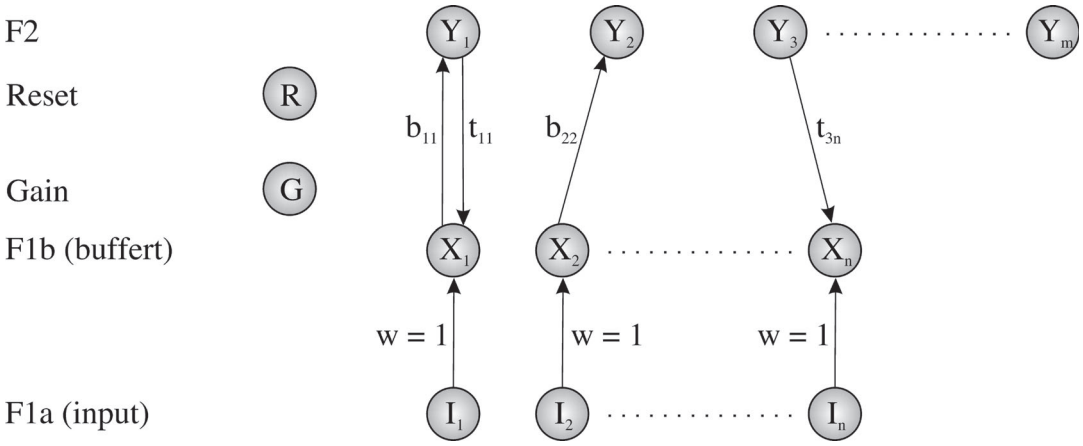
långtifrån alltid sammanfaller bra med de verkliga städerna (svarta cirklar utan förbindelselinjer). – Numreringen av de verkliga städerna är godtycklig.



Figur 65. En lösning med SOM av det euklidiska TSP för 13 städer. Modifierat efter ett skärmfoto i NeuralWorks. Förklaring: se text.

8.4 ART (Adaptive Resonance Theory)

Adaptive Resonance Theory, som konstruerats av en annan veteran på ANN-området (Stephen Grossberg), är sammanfattningsvis ett ganska komplicerat och relativt biologinära nätverk som kategoriserar mönster med hjälp av en kompetitiv algoritm. Det är tvålagrat, har feedback i andra lagret av förbindelser (mellan andra och tredje skikten av enheter) och tar i sin enklaste variant (ART 1) binära indata. Vi benämner här inputskiktet F1a och nästa skikt F1b, medan det kompetitiva skiktet får heta F2. Två extra noder, Reset och Gain, utför speciella uppgifter i nätverket som vi strax ska berätta om.



Figur 66. Strukturen hos ett ART1-nätverk. Förklaring se text.

I figur 66 har endast ett fåtal av förbindelserna i nätverket ritats ut. b_{ij} står för bottom-up-vikter och t_{ij} för top-down-vikter. Vikterna från F1a till F1b är alltid 1; de mellan F1b och F2 initialiseras så att alla bottom-up-vikter randomiseras, medan alla top-down-vikter initialt sätts = 1.

Aktivering och inlärning i ART1

Så här *aktiveras* nätverket:

(8.2.3) (a) Binär input (en räcka av 1-or och 0-or) presenteras i F1a. Denna signal antas kvarstå i F1a under hela aktiveringscykeln.

(b) Input passerar i obearbetat skick till F1b-lagret.

(c) F1b skickar vidare input till F2 via bottom-up-vikterna.

(d) Den F2-nod som nu har högst aktivitet utses till preliminär vinnare. Kalla denna nod för V.

(e) V skickar tillbaka en signal till F1b genom top-down-vikterna.

(f) Signalen från F2 ”jämförs” i F1b med signalen från F1a. Detta går till som så att endast de noder i F1b som har 1 som input från båda hållen får aktivitet = 1; aktiviteten i övriga

F1b-noder sätts till 0. F1b-noderna fungerar nu med andra ord som logiska ”OCH-kretsar” (genom en tröskelmekanism). Här är Gain-noden inblandad på ett väsentligt sätt; se nedan!

(g) Systemet tittar nu på en parameter som kallas ”vigilans” för att bestämma om V ska bli definitiv vinnare eller ej. Om antalet aktiva noder i F1b, sett som en andel av antalet aktiva noder i F1a, överstiger värdet på vigilansfaktorn så godkänns vinnaren. Annars träder Reset-noden i funktion och stänger av V tills vidare, varefter systemet börjar om vid steg (c) för att utse en ny vinnare. Exakt hur Reset-noden utför detta ska vi inte gå in på.

Om ingen vinnare hittas kan nätet ha olika strategier för att fortsätta.

Första gången en F2-nod preliminärt vinner på en input är alla top-down-vikter = 1. Noden kommer därför att skicka tillbaka precis samma signal som den tar emot, och resultatet av steg (f) måste därför bli att V utses till definitiv vinnare. Om samma nod vid ett senare tillfälle vinner på samma (eller på en annan) input är det emellertid inte alls självklart att den kan exakt reproducera mönstret, eftersom vissa top-down-vikter från noden då kan ha ”nollats” (se nedan). Därför blir jämförelserna mellan input och output snart icke-triviala.

Inlärning sker för den slutgiltigt vinnande noden genom en Hebb-liknande mekanism där vikternas värden är maximerade till 1. Vikten på en top-down-förbindelse från vinnaren till en viss nod i F1b sätts ner till 0, om och endast om top-down-signalen i denna förbindelse är 1 men motsvarande inputkomponent är 0. Bottom-up-förbindelserna kan däremot öka eller minska gradvis i styrka. Detta är så kallad ”snabb inlärning”; varianter finns som vi dock inte skall gå in på.

Dessa aktiverings- och inlärningsregler leder till slut till en kategorisering av data. I ett visst avseende kan den styras: om man har valt en hög vigilansparameter är kategorierna ”smala”, annars är de ”vidare”. Med maximal vigilans (= 1) kodar varje F2-nod bara för en enda input. ART ger på så vis en inte helt oplausibel modell av en aspekt av mänsklig begreppsbildning, nämligen vår förmåga att bilda snäva eller vida begrepp alltefter behov.

ART som modell av biologiska nätverk

Är modellen verkligen biologiskt trovärdig? Grossberg och hans medarbetare har lagt ner ganska stor möda på att visa att det är så. Bland annat härleder man aktiverings- och inlärningsalgoritmerna ur sådana mer fundamentala ekvationer för kontinuerliga system som av många anses karakterisera neuron (jämför ekvation 7.3.8, avsnitt 7.3), och Gain- och Reset-nodernas funktioner beskrivs i detalj på ett biologiskt rätt plausibelt sätt. Vi ska nöja oss med att förklara vilket jobb Gain-noden utför:

I steg (f) ovan antas F1b-noderna fungera som ”OCH”-kretsar, dvs. de aktiveras om och endast om de får signalen 1 både ”nerifrån” och ”uppifrån”. Men hur går det ihop med att de i steg (c) reagerar på en enda signal ”nerifrån”? Kan de byta aktiveringsfunktion mitt under aktiveringscykeln? Nej, de fungerar alltid som samma tröskelelement och kräver två ”ettor” in för att aktiveras, men de har en *tredje* inputlinje från Gain som ibland är ”på” och ibland är ”av”. Närmare bestämt är Gain ”av” om båda lagren F1a och F2 är aktiva, men ”på” om bara F1a är aktivt. Man inser med lite eftertanke att detta tillåter både ”jämförelsen” mellan F2 och F1a och den ursprungliga överföringen av signalen från F1a till F1b. Dessutom är Gain ”av” om F1a är inaktivt, dvs. om det inte föreligger någon input. Annars, säger Grossberg, skulle ju F2-noderna kunna få systemet att hallucinera!

En egenskap hos ART som talar för dess biologiska relevans är att modellen ger en lösning av vad Grossberg kallar *stabilitets-plasticitets-dilemmat*. Detta dilemma innebär att avvägningen kan vara svår mellan ett nätverks förmåga att lära sig radikalt nya saker och dess förmåga att komma ihåg det som den redan har lärt sig. Försöker man till exempel *lära* ett redan tränat SOM att kategorisera några helt nya, avvikande inputs – dvs. man ger inte bara de nya inputs och ser var de hamnar på kartan, utan man applicerar hela inlärningsalgoritmen (förutom den initiala randomiseringen av vikter) på sitt redan tränade SOM och den nya, utvidgade träningsdatamängden – så är risken stor att man river upp den gamla kategoriseringen så att resultatet blir helt oigenkännligt.

Detta händer inte i ART. Om en radikalt ny input presenteras för ett färdigtränat ART kommer sannolikt ingen av de noder som tidigare ”samlat på sig” inputs att vinna. Nätverket plockar istället fram en ny, fräsch vinnarkandidat ur förrådet av oförbrukade F2-noder och bildar en helt ny kategori, som sedan fortlever vid sidan om de gamla, oförändrade katego-

rierna. Kanske är detta en modell som kan relateras till vår nyvunna insikt, att nervceller ständigt nybildas i hjärnan?

ART är alltså intressant inte minst därför att nätverket i flera avseenden är biologiskt rimligt. Detta omdöme gäller för övrigt i ännu högre grad om de olika vidareutvecklingar av grundtankarna bakom det, som Grossberg och hans medarbetare nu använder i sina försök att modellera bland annat det visuella systemet.[146] ART-nätverkens tekniska värde som metod för kategorisering av data är mer omtvistat, men det verkar att finnas många välfungerande applikationer.

8.5 Ett kooperativt-kompetitivt nätverk för stereoseende

Djupseendets problem

I detta avsnitt ska vi göra en liten exkurs från vårt huvudtema, som ju är inlärning och minne, och titta på ett neuralt nätverk som gör intressanta saker utan att kunna lära sig något genom erfarenheten. De funktioner nätverket har är alltså inbyggda i det från början (men givetvis kan man tänka sig att också ge det inlärningsförmåga). Avsnittet är ganska detaljerat, eftersom det berör klassiska filosofiska och psykologiska frågor.

Nätverket som vi ska prata om är avsett vara en förenklad modell av en viss aspekt av vårt visuella system. Det handlar om hur vi uppfattar djup med synen, och vi börjar med ett citat från filosofen George Berkeley.

> ...I believe whoever will look narrowly into his own thoughts, and examine what he means by saying he sees this or that thing at a distance, will agree with me, that what he sees only suggests to his understanding that, after having passed at a certain distance, to be measured by the motion of his body, which is perceivable by touch, he shall come to perceive such and such tangible ideas, which have been usually connected with such and such visible ideas.[147]

Berkeley är som synes av uppfattningen att vi egentligen inte *ser* djup, utan bara uppfattar det indirekt, genom förståndet. Om man tolkar vad han säger i en fenomenologisk anda, dvs. som en beskrivning av vilka upplevelser och tankar som ingår i seendet av djup, är det mycket tvek-

[146] Se t.ex. Berzhanskaya et al. (2007).
[147] Berkeley (1901) [1709], ss. 148 f.

samt om han har rätt. Djupseende innebär inte att vi medvetet tänker eller tror, att en tvådimensionell struktur som vi ser är förbunden på ett visst sätt med möjliga känselupplevelser. Däremot är det uppenbart, att de i viss mening tvådimensionella signalmönstren från näthinnorna genomgår en avancerad bearbetning i hjärnan och där integreras med annan information för att vi rätt ska kunna hantera den tredimensionella världen. Berkeleys påstående är därför mer plausibelt om det tolkas som en metaforisk beskrivning av icke medvetna, fysiologiska processer. Hur ser dessa processer ut?

Kända mekanismer för djupseende

Vår förmåga att med synen uppfatta den tredje dimensionen av rummet bygger faktiskt på flera olika mekanismer. Några av dem är sannolikt *högnivåiga* i den meningen, att de i sin tur bygger på bearbetning i många steg av de primära signalerna. Om vi ser ett objekt av välbekant slag och ser *vad* det är, kan vi således med hjälp av vår kunskap om dess verkliga storlek och den vinkel det upptar i synfältet i princip räkna ut avståndet till det. Det är mycket troligt att en del av vår djupuppfattning bygger att hjärnan implicit utför en dylik beräkning. Andra mekanismer kommer sannolikt in på ett något tidigare stadium av bearbetningen av visuella signaler, bland annat den mekanism som bygger på så kallade *strukturgradienter.* När vi exempelvis ser ett fält med enhetlig vegetation, som ett sädesfält, avtar växternas (eller växtdelarnas, här: axens) skenbara längd på ett regelbundet sätt uppåt mot horisonten. Denna gradient ger god information om relativa avstånd, förutsatt av vi vet att längden på axen är någorlunda homogen över fältet. Vi behöver däremot inte veta *hur* långa axen är för att göra dessa relativa bedömningar. Ett flertal andra djupseende- och avståndsbedömningsmekanismer på dylik ”halvlåg” nivå har föreslagits, inte minst av forskare i traditionen kring Eleanor och James Gibson.[148]

Slutligen, och inte minst, använder sig det visuella systemet uppenbarligen av principer för djupseende som kommer in på ett mycket tidigt stadium av signalbearbetningen. Dit hör ackommodation, konvergens och stereopsi. *Ackommodation* är den mekanism för anpassning av linsens form med vars hjälp vi får en skarp monokulär (med ett öga) bild på ett visst avstånd, eller rättare sagt inom ett visst avståndsintervall, som vi här ska kalla ”fokalzonen”. *Konvergens* är när vi vinklar in ögonen för att få

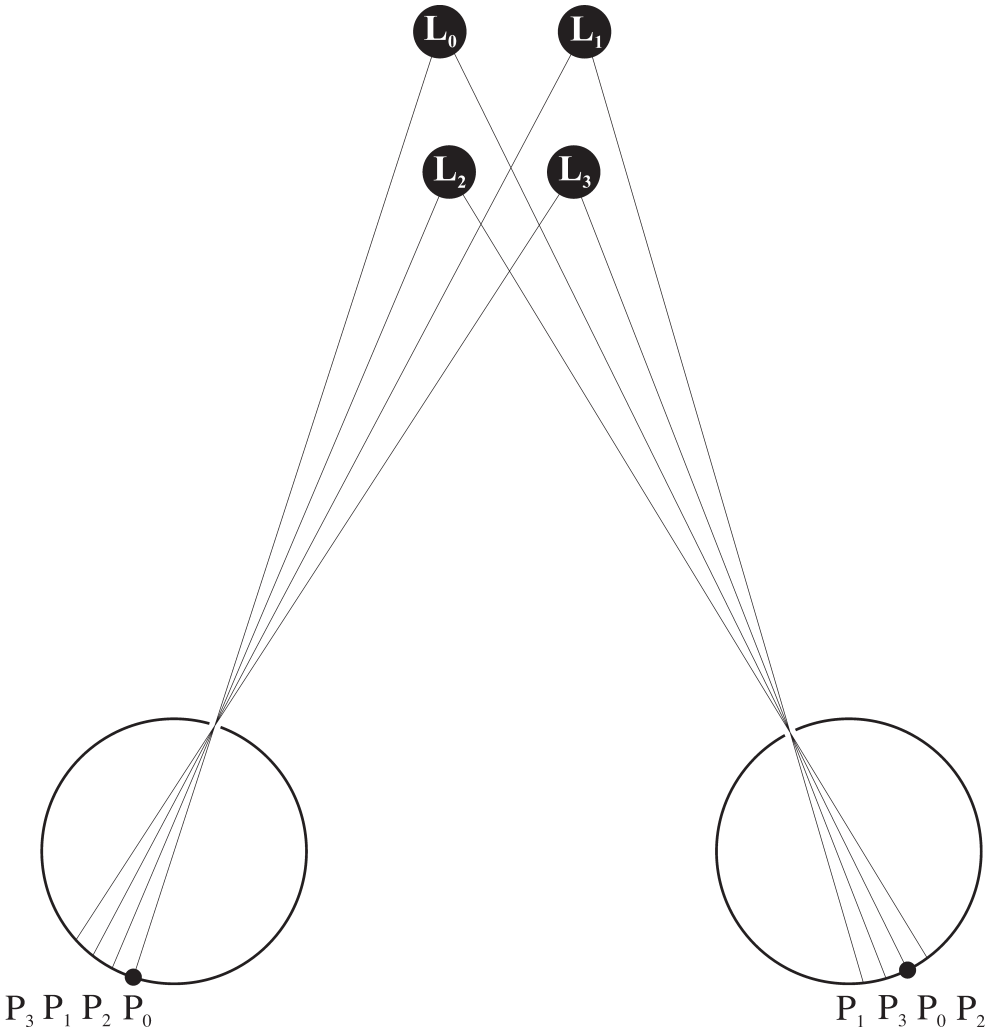
[148] Se J.J. Gibson (1978).

en enda bild trots att vi har två ögon. Information som återföres från de system som sköter ackommodation och konvergens spelar en mycket viktig roll för vårt avståndssende, framförallt på relativt nära håll. Denna information är dock ofullständig även när det gäller korta avstånd, dels för att vi som sagt ser maximalt skarpt inom ett avståndsintervall (inte bara på ett exakt avstånd), dels för att konvergensen som sådan bara bestämmer avståndet till en enda punkt i synfältet (för att uttrycka det enkelt).

Stereopsi

Den princip som kompletterar ackommodation och konvergens på ett väsentligt sätt kallas *stereopsi.* Stereopsi bygger på den *skillnad mellan de två näthinnebilderna* som uppstår när vi fokuserar på ett visst avstånd och konvergerar ögonen mot en punkt på detta avstånd. Det är med största sannolikhet stereopsimekanismen som ger oss vår mycket precisa djupdiskrimination inom fokalzonen, och som bland annat gör att vi kan hantera små objekt där med hög säkerhet. (Försök hitta skåran på en liten skruv med skruvmejseln när det ena ögat är skymt, så förstår du vad som avses.)

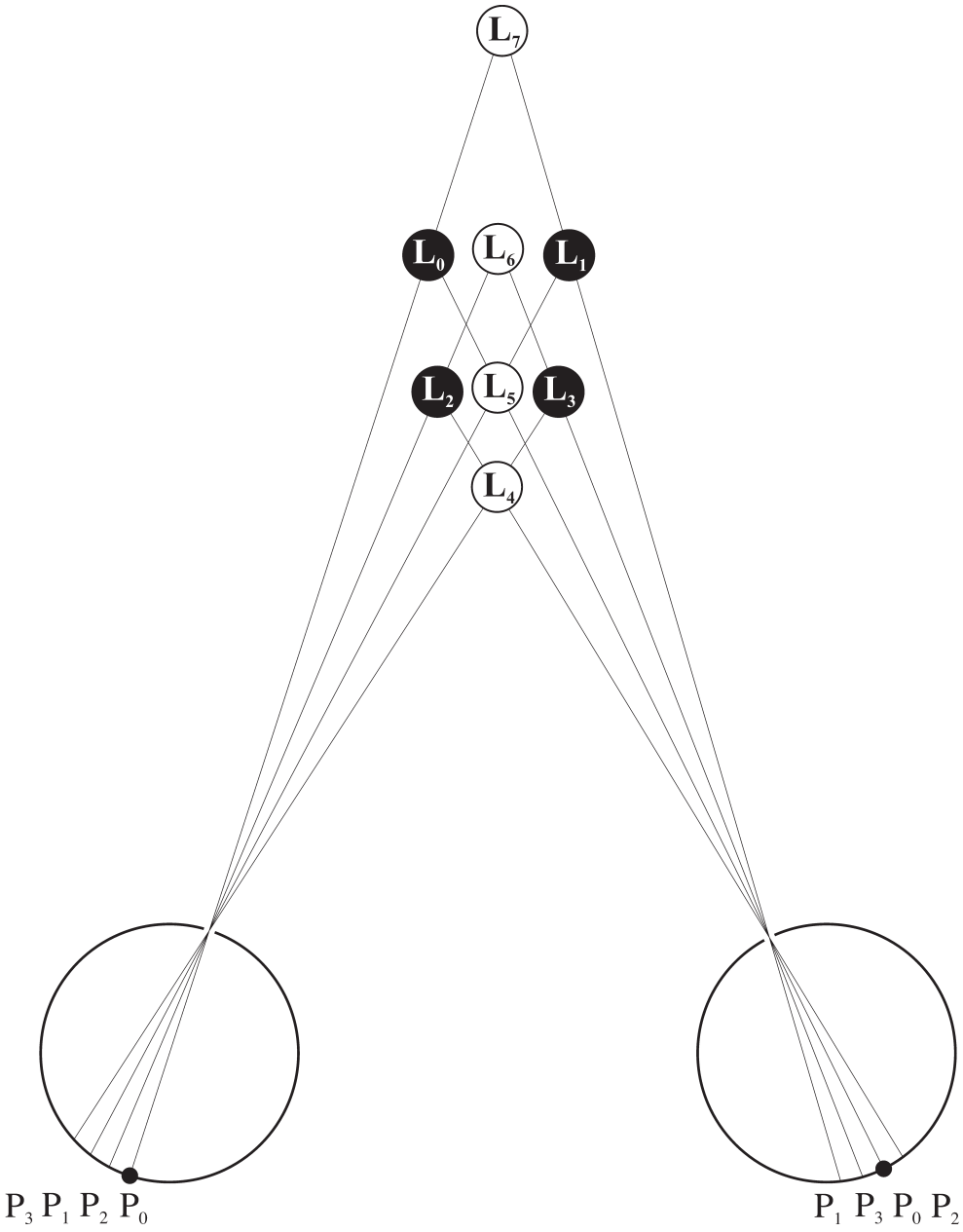
I figur 67 nedan antas iakttagaren fokusera på det avstånd på vilket ljuspunkterna L_0 och L_1 befinner sig, och konvergera ögonen så att ljuset från punkten L_0 hamnar i fovea (gula fläcken) i båda ögonen. Vi betecknar fovea i båda ögonen med P_0. Ljusstrålarna från L_1 hamnar då i två *motsvarande* punkter P_1 i de båda ögonen, och avståndet mellan P_0 och P_1 är (i det närmaste) lika stort på båda näthinnorna. Betrakta nu ljuset från punkten L_2, som ligger i ett annat plan än L_0 och L_1 men fortfarande inom fokalzonen (vars djup vi kraftigt överdrivit i figuren), och som lämnar ett avtryck i punkten P_2 på de båda näthinnorna. Det framgår omedelbart av figuren att förhållandet mellan P_0 och P_2 inte är detsamma på vänster och höger näthinna. Signalen från L_2 i *vänster* öga har närmare bestämt förskjutits åt *vänster* i förhållande till signalen i höger öga. Motsvarande skillnad föreligger för punkten L_3. Dessa skillnader är de *retinala disparitеter* som man tänker sig signalerar att L_2 och L_3 ligger *närmare* näthinnan än vad L_0 och L_1 gör. Visst borde nervsystemet kunna använda dessa disparitеter för att räkna ut hur långt L_2 och L_3 ligger från L_0 och L_1?



Figur 67. Retinal disparitet. P_0 betecknar fovea i respektive öga. Förklaring i övrigt, se text.

Problemet med falsk matchning

Här kommer det emellertid in en besvärlig komplikation, som beror på att det visuella systemet inte självklart ”vet” vilka signaler på de två näthinnorna som härrör från *samma* ljuskällor. Ett åskådligt exempel ges av figur 68, där ljuspunkterna på näthinnan har lånat sina beteckningar från figur 67.



Figur 68. Flera möjliga orsaker till samma retinala mönster. Förklaring: se text.

Om ljusmönstren på näthinnorna utgör den enda evidensen, är hypotesen att de kommer från punkterna L_4–L_7 en lika bra förklaring som att de kommer från punkterna L_0–L_3. Och observera att om denna alternativa hypotes är riktig, så kommer signalen i P_0 på vänster näthinna från samma punkt som signalen i P_1 i höger, nämligen från punkten L_7! Punktparen P_2(vä)/P_3(hö), P_1(vä)/P_0(hö) och P_3(vä)/P_2(hö) betecknar också signalpar som då kommer från samma punkter (L_6, L_5 respektive L_4).

Poängen med exemplet är förstås, att det visuella systemet inte har något sätt att ”veta” hur signalerna ska ”paras ihop” – de är ju inte märkta med några index när de anländer till näthinnorna. Om systemet utgår från att P_0 till vänster och P_0 till höger hör ihop, kan detta vara helt fel, dess slutsats att det finns en ljuspunkt i L_0 likaså. Vi kallar hädanefter ett sådant misstag för en *falsk matchning*. Möjligheten av falsk matchning gör att det visuella systemet inte kan bestämma några retinala dispariteter utgående från enbart näthinnemönstren, och därmed kan det inte heller räkna ut några avstånd i djupled. *Om* nervsystemet kände till hur signalerna faktiskt hör ihop parvis, dvs. vilka signaler på de två näthinnorna som kommer från samma punkt, skulle det kunna i princip avgöra vilka av de möjliga källorna L_0–L_3 eller L_4–L_7 som ljuset kommer från. Men nu saknas alltså den nödvändiga informationen.

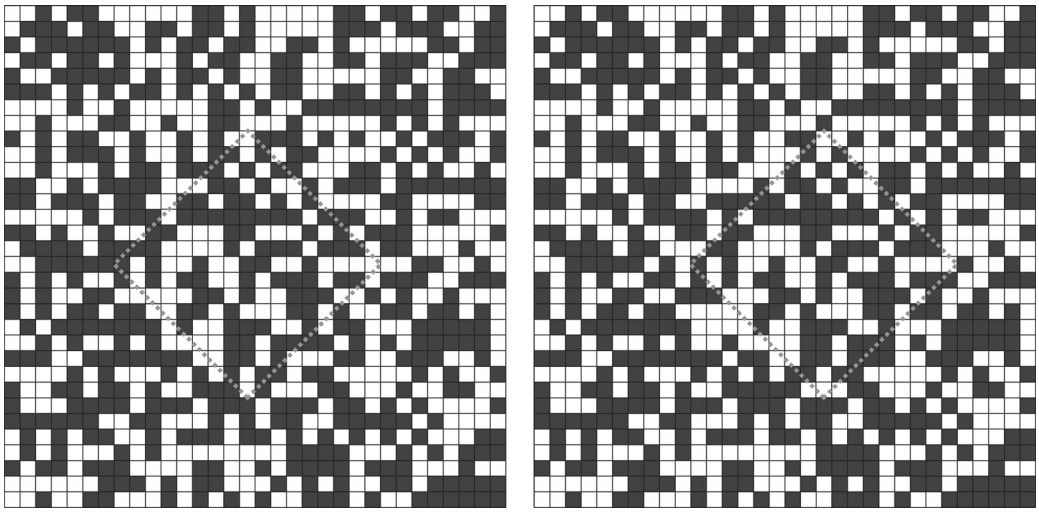
Det är lätt att inse att exemplet har generell räckvidd. (Rita gärna själv ett annat exempel!) Det betyder att *beräkningen av ett djup med hjälp av enbart näthinnedispariteter som regel inte har någon välbestämd lösning*. Om man tänker sig att det visuella systemet ska kunna lösa problemet måste det alltså till mer information.

Hjälpantaganden för att lösa stereoproblem

I traditionen från David Marr (som var den som först föreslog en algoritm för att lösa problemet) anser man att vad som kommer in nu är ett slags implicita antaganden eller ”constraints” om omgivningens beskaffenhet.[149] Om hjärnan exempelvis antar (vad *det* nu betyder ska vi strax återkomma till!) att *stora sammanhängande ytor* (på varje givet avstånd) är sannolikare än små ytor, så kommer lösningen L_0–L_3 i figur 67 att favoriseras framför den alternativa lösningen L_4–L_7 i figur 68.

Man förmodar att det är just ett sådant implicit antagande som gör att vi kan lösa s.k. slumpstereogram. Ett slumpstereogram består av två binära bilder varav den ena är uppbyggd av utslumpade svarta och vita pixlar. Den andra bilden är identisk med den första utom i så måtto att *alla pixlar* – svarta såväl som vita – *inom en viss tänkt figur*, t.ex. en liten romb, har förskjutits en liten sträcka åt samma håll, höger eller vänster. Den del av matrisen som blir ”tom” vid förskjutningen fylls åter i slumpvis. Principen illustreras i figur 69 (som på grund av den valda skalan tyvärr inte är lämplig för experiment med stereoseende).

[149] Jämför Marr (1982).



Figur 69. Principen för ett slumpstereogram. Området som har flyttats i den högra bilden jämfört med den vänstra har markerats.

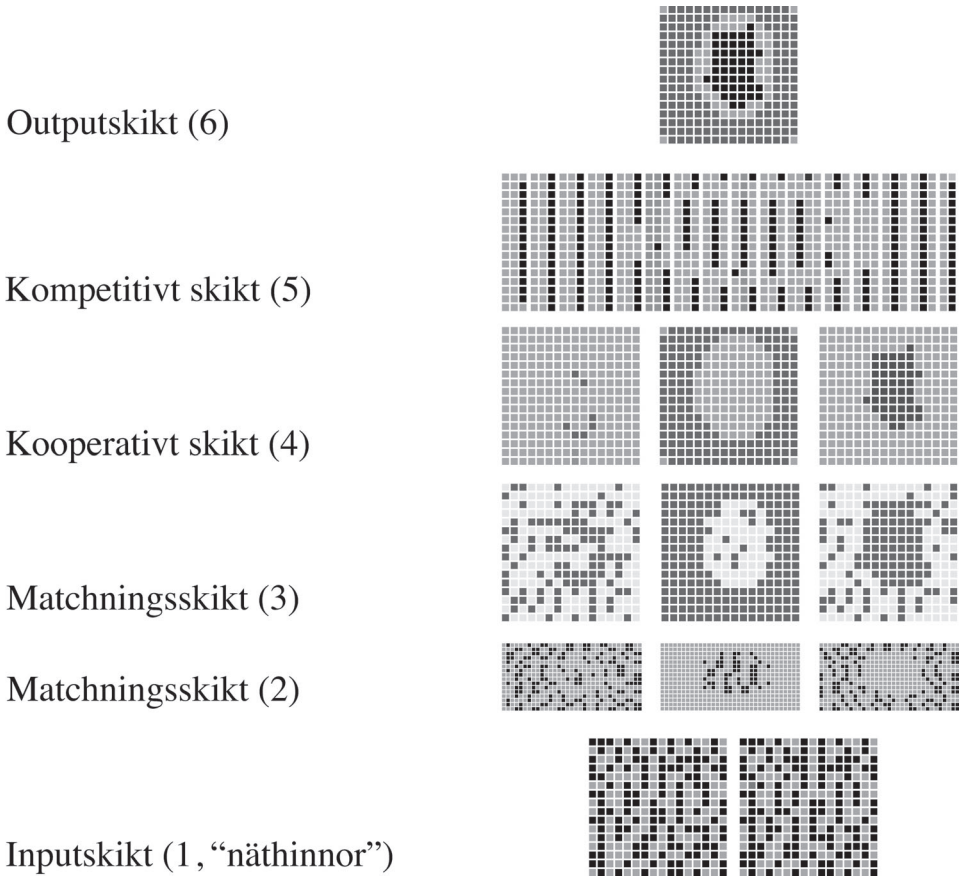
Det synintryck som uppstår när man ser på ett slumpstereogram (som då givetvis inte innehåller någon markering av den ”dolda” figuren!) genom ett stereoskop, eller med blotta ögonen när dessa står parallellt och vardera ögat ser bara den ena bilden, är ett intryck av en enda bild där en figur ligger över eller under resten av bilden. Hur går nu detta till?

Lägg först märke till det är *principiellt omöjligt* att extrahera den nödvändiga informationen om den ”dolda” figuren från bara den ena av bilderna! Även den bild som man skapat genom att flytta ett visst parti av den första innehåller nämligen ett i sig helt slumpartat mönster. Det är bara i relation till den första bilden som den inte är slumpmässig.

Det går däremot att extrahera figurinformationen från hela bildparet genom att räkna på de näthinnedispariteter som uppstår. Om det är så hjärnan gör, följer det att objektigenkännandet i detta fall bygger på djupseendemekanismen, och inte tvärtom. Beräkningen förutsätter dock att hjärnan arbetar med ett visst implicit antagande av det ovan nämnda slaget, dvs. favoriserar hypoteser om *stora* objekt på varje givet avstånd från näthinnan. Annars blir lösningen obestämd på samma principiella sätt som i figur 68, men i stor skala. Konstruktionen av problemet, med ungefär lika många svarta som vita punkter och en slumpmässig fördelning, ger nämligen ett mycket stort antal möjligheter till falsk matchning (se vidare nedan).

Ett neuralt nätverk som använder hjälpantaganden

Nåväl, till saken: kan man bygga in ett sådant antagande i ett artificiellt neuralt nätverk och därmed modellera hur våra egna nervsystem kan tänkas lösa uppgiften? Ja, det kan man, och flera alternativa nätverk har föreslagits i sammanhanget. De bygger dock alla på samma grundläggande princip. Här ska en variant skisseras som konstruerats av två doktorander vid Filosofiska institutionen vid Göteborgs Universitet, och som regelbundet används vid ANN-laborationerna där. Betrakta figur 70!



Figur 70. Stereomaskinen. Modifierat efter ett skärmfoto i NeuralWorks. Varje liten fyrkant motsvarar en ANN-nod, och dess ljushetsgrad motsvarar (någorlunda väl) nodens aktivitetsnivå. För förklaring i övrigt se texten.

Nätverket består av linjära enheter och tröskelenheter. Input är just ett slumpstereogram: dels en binär slumpmatris, dels samma matris modifierad genom enstegs höger- eller vänsterförskjutningar av punkterna inom en eller flera begränsade ytor. Stereogrammet representerar alltså objekt

som ligger i tre olika plan. Bilderna projiceras på två ”näthinnor”, som utgör inputskiktet (skikt 1).

Skikt 1 skickar tre versioner av bildinformationen vidare till nästa lager. Det första skiktet av neuron efter inputskiktet (skikt 2) består nämligen av tre delar, som metaforiskt kan sägas utgå från var sin hypotes om vad som är korresponderande punkter på de två näthinnorna – dvs. om vilka punkter som härrör från samma ljuskälla. Skikt 2a (till vänster i bilden) utgår från att alla signaler kommer från ett plan alldeles ”hitom” konvergensplanet, vilket konkret betyder att det tar emot alla signalerna från den *vänstra* näthinnan förskjutna ett steg åt *vänster* i förhållande till punkterna på den högra näthinnan. Skikt 2b (i mitten) tar emot näthinnesignalen utan förskjutning, och skikt 2c (till höger) erhåller signalen efter en förskjutning av alla punkterna på den vänstra näthinnan ett steg åt höger.

I delskikt 2a-2c samt de motsvarande delskikten i ett ytterligare skikt, nr 3, jämförs signalerna på *motsvarande* punkter på näthinnorna med varann. Detta är en uppgift av typ XOR. Låt oss säga att skikt 2–3 hittar en *match* om två motsvarande punkter har samma värde (svart/svart eller vit/vit). Eftersom de olika delskikten har olika ”utgångshypoteser” i frågan om vilka punkter som motsvarar vilka, jämför de olika punkter och ger (som regel) helt olika svar på frågan om vilka punkter på den ena näthinnan som har en matchande punkt på den andra.

Ut från skikt 3 kommer alltså tre signalmönster som representerar match/icke-match under de tre antagandena. Dessa tre mönster representeras i skikt 4 i tre olika uppsättningar neuron (tre *disparitetskanaler*). I detta skikt sker något avgörande. Här favoriseras nämligen homogena objekt i samma plan genom att *signalerna från närliggande enheter inom varje kanal förstärker varann*. Det är alltså fråga om en kooperativ mekanism. Nästa skikt (nummer 5) är däremot kompetitivt. Här förstärks skillnaderna mellan motsvarande enheter i de olika disparitetskanalerna genom att *inhiberande signaler skickas mellan motsvarande enheter i de tre kanalerna*. Signalerna i skikt 4 och 5 är anpassade så, att högst ett neuron av de tre som beskriver avståndet till en viss punkt har kvarvarande aktivitet när processen är avslutad. Detta neuron får då avgöra frågan. I enstaka fall (om inget neuron har någon aktivitet efter den slutliga tävlingen) kan den lämnas obesvarad.

I outputlagret syntetiseras ett enda mönster, där varje punkt har ett ljusvärde som anger hur långt bort punkten förmodligen ligger. Det mot-

svarar alltså ett synfält med djupseende. I de flesta fall är det en rätt bra bild av den 3-D-struktur som man kodat in i slumpstereogrammet.

I figur 70 kan man bl.a. se hur den kooperativa processen i skikt 4 eliminerar flera små inhomogeniteter i de mönster som kommer från skikt 3. Samarbetet i skikt 4 kodifierar nätverkets bias till förmån för stora sammanhängande objekt, som det alltså har jämförelsevis lätt att ”se”. De största olikheterna mellan ”den tredimensionella verkligheten” och ”den visuella representationen” får man, omvänt, i de fall inputobjekten är små och/eller mycket inhomogena.

Modellen som vi har beskrivit här är förstås en kraftig förenkling av verkligheten. Framförallt löser den bara den begränsade uppgiften att lokalisera objekt i ett av tre givna plan. Men det är inte svårt att generalisera modellen, och den åskådliggör på ett mycket tydligt sätt hur nervsystemet kan tänkas hitta en ”bästa” lösning av en uppgift (dvs. lösa en optimeringsuppgift) under givna begränsningar genom att använda sig av ömsesidig förstärkning och försvagning (kooperation respektive kompetition) mellan neuronala enheter.[150]

[150] För en översikt av stereopsi ur neurovetenskaplig synvinkel se Cumming & DeAngelis (2001).

9. Neurala nätverk för icke-linjära problem

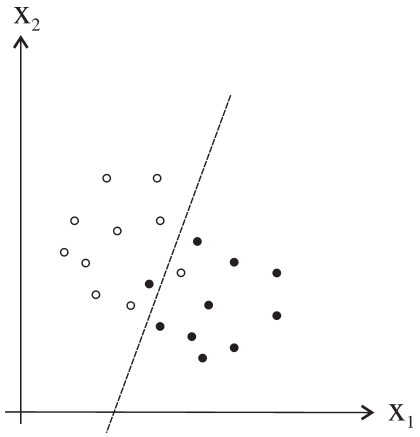
9.1 Kraftfulla – och kanske *för* kraftfulla – modeller

Den enkla perceptronen kan som vi sett lösa alla linjärt separerbara beslutsproblem, men inga andra. Likaså kan enlagrade fullt linjära nätverk som använder deltaregeln utföra linjär regression, men inte icke-linjär sådan. Lyckligtvis finns det ett stort antal nätverk som klarar av mer komplicerade problem – till och med godtyckligt komplicerade sådana. Vi ska nedan (9.2) genom några enkla exempel förklara hur lösningar av komplexa problem *rent principiellt blir möjliga* genom att man lägger till fler lager av adaptiva förbindelser och element med icke-linjära, kontinuerliga aktiveringsfunktioner. Därefter (fortfarande i avsnitt 9.2) beskriver vi den mest välkända typen av dylika kraftfulla nätverk, nämligen flerlagrade perceptroner (MultiLayer Perceptrons, MLP). Nästa avsnitt (9.3) ägnas åt några grundläggande sätt på vilka en MLP *genom inlärning kan hitta* de möjliga lösningarna av ett problem. I samband med de flerlagrade perceptronerna, men i ett särskilt avsnitt (9.4), diskuterar vi också *bayesiansk* inlärning i neurala nätverk. Slutligen skall vi i detta kapitel också kort (när det gäller SVM helt summariskt) beskriva tre andra typer av kraftfulla algoritmer, som i praktiken ofta är användbara alternativ till flerlagrade perceptroner, nämligen Learning Vector Quantification (LVQ, avsnitt 9.5) samt radialbas-nätverk och supportvektormaskiner (RBF-nätverk respektive SVM, avsnitt 9.6).

Kriterier för en ”bra” lösning

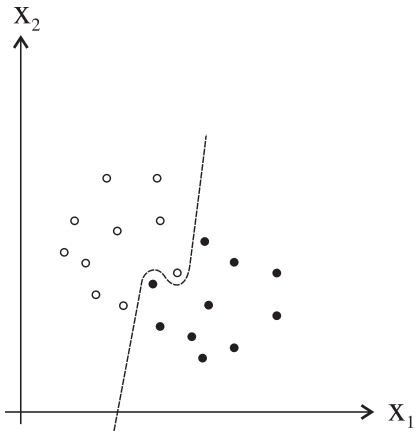
Allra först skall vi dock uppehålla oss ytterligare en stund vid den problematik beträffande *generalisering* som vi redan berört ett par gånger. Låt oss ställa en fråga som kanske kan tyckas egendomlig, nämligen, ”Är en perfekt separation av data alltid den bästa lösningen av ett klassifikationsproblem?”

Antag att vi har en medicinsk datamängd som har det utseende som framgår av figur 71, där fyllda och ofyllda cirklar betecknar (par av) mätvärden från patienter med en viss sjukdom, respektive från friska patienter. En enkel perceptron misslyckas med separationen.



Figur 71. Icke linjärt separerbara data och resultatet (den räta linjen) av en tänkt analys med en enlagrad perceptron.

Den räta linjen är, som vi vet, den bästa beslutslinje som en enkel perceptron hittar. Antag att vi istället använde ett mer kraftfullt nätverk som kunde göra en fullständig separation av data. Beslutsgränsen kunde kanske se ut som i figur 72.



Figur 72. En icke-linjär separation av data från figur 71.

Kan det i så fall finnas någon rimlig anledning att *inte* använda detta nätverk, snarare än den enkla perceptronen, för att ställa diagnos? Ja, det kan

det, och den anledningen har att göra med *generaliseringsförmåga*. Vi måste tänka på, att den yttersta målsättningen med en algoritm för automatisk mönsterklassifikation inte är att separera punkterna (patienterna) i den *ursprungliga* datamängden i olika klasser (sjuka och friska), utan att avgöra *kommande* fall. Algoritmen måste med andra ord kunna generalisera. Och det är faktiskt inte alls självklart, att det nätverk som presterar bäst på träningsmängden också är det som generaliserar bäst. Tvärtom ser man, om man i sitt val av nätverk fokuserar alltför mycket på prestationen vad avser träningsdata, ofta *ett inverst förhållande mellan "ursprungliga" prestanda och prestanda på "osedda" fall*. Hur kan det vara så? Vi ska utreda denna fråga ganska noga, eftersom den är av fundamental betydelse för förståelsen av vad artificiella neurala nätverk kan och inte kan göra.

Förklarbar och oförklarbar variation i empiriska data

Antag att vi (som i ovanstående exempel) försöker skilja två klasser av objekt utgående från empiriska data. De flesta empiriska observationer innehåller nu ett slumpelement, eller, mer generellt, ett element av variation som inte kan förklaras av variabler som vi har tagit med i analysen. Det kan vara fråga om en opålitlighet i mätmetodiken som ger ett slumpmässigt fördelat mätfel, eller om en verklig variation hos objekten som dock inte (enbart) beror på de faktorer som vi räknar med. Låt oss börja med två något långsökta ("filosofiska"), endimensionella exempel. Antag att vi vill diskriminera mellan bilar av två liknande modeller, och att beslutskriteriet ska grundas på *bilarnas längd, uppskattad med ögonmått på 50 meters håll*. De två klasserna av bilar överlappar, kan vi anta, inte i *verklig* längd, men mätfelet kommer säkerligen att göra att *varje* val av beslutsgräns medför några felklassifikationer. Eller antag att vi vill skilja mellan män och kvinnor, utgående enbart från *längd mätt med måttband*. Hur vi än väljer beslutsgränsen (t.ex. 170 cm) kommer vi att få en rätt hög procent felklassifikationer. Och detta *trots* att längd är en variabel som i viss mån kan användas för att skilja män från kvinnor, och *trots* att mätfelet kan anses vara litet. I detta fall beror de förväntade felklassifikationerna istället på att kroppslängd i hög grad är beroende av andra faktorer än könstillhörigheten. Man kan sammanfatta resonemanget som att det i empirisk vetenskap mycket ofta är så, att variationen i den iakttagna beroende variabeln – i våra exempel: fördelningen på två olika klasser – endast delvis är *förklarbar* med hjälp av de oberoende variabler som man tagit med i analysen.

”Bättre” metoder som generaliserar sämre

Antag att vi i det andra exemplet ovan, och i ett urval bestående av 10 män och 10 kvinnor, funnit att beslutsgränsen 175 cm ger den bästa separationen av klasserna. Närmare bestämt blir det i det givna materialet bara en felklassifikation av vardera sorten: en man var 165 centimer lång och en kvinna var 181 centimeter lång. (Alla mätvärden förutsätts för enkelhets skull vara avrundade till heltal.) Vi är väldigt angelägna om att hitta ett perfekt beslutskriterium. Turligt nog är ingen man i urvalet 181 cm lång och ingen kvinna 165 cm lång, varför vi kommer på den (till synes) geniala idén att förfina beslutskriteriet på följande sätt: en person är kvinna *om hennes längd antingen är mindre än 175 cm (dock ej 165 cm), eller också 181 cm*; annars är det fråga om en man. Voilà en perfekt beslutsregel!

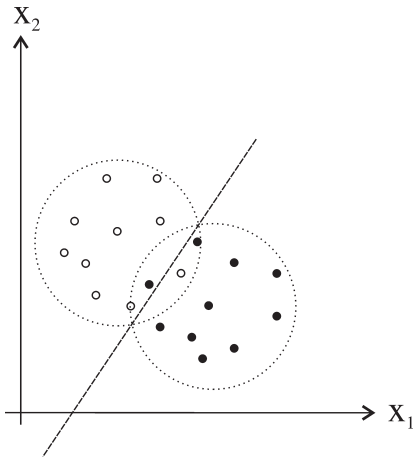
Om vi gör några mycket naturliga antaganden (se vidare nedan) om vad kroppslängd beror på och hur den varierar kan vi emellertid förutsäga, att denna ”perfekta” regel sannolikt inte kommer att fungera bra i nästa urval av personer. Data antyder ju att det finns faktorer *utanför* vår analys som avgör personers längd, och den slumpmässiga variationen i dessa faktorer kommer naturligtvis att göra att det i fortsättningen uppstår felklassifikationer även med den ”perfekta” regeln. Vi kan förvisso förvänta oss fler män som är under 175 cm, och fler kvinnor som är längre än 175 cm, och det finns ingen särskild anledning att tro att dessa personer ska vara just 165 respektive just 181 cm långa. Om en person i nästa urval är 181 cm lång *är det tvärtom mest sannolikt att det är fråga om en man*, medan en person med längden 165 cm sannolikt är en kvinna. Men den ”perfekta” regeln kommer att klassificera dem tvärtom! Det är just detta som gör att den ”perfekta” regeln till och med kan förväntas fungera *sämre* än den ursprungliga regeln. De specialdesignade undantagen kommer i det långa loppet inte att vara till nytta, utan till skada, och det fel som därigenom uppstår skall *läggas till* det förväntade fel som slumpmässigheten i sig innebär. Den enkla regeln (utan undantag) drabbas visserligen också av slumpmässigheten, och kan därför förväntas fungera ungefär lika dåligt i de kommande urvalen som i det första – men den kan förväntas fungera bättre i kommande urval än den till synes ”perfekta” regeln!

Att koda slumpvariationen

Det fel som det innebär att använda ”regeln med undantag” kan beskrivas som att regeln tenderar att *kodifiera den slumpmässiga komponenten* i

datamängden, och inte bara det samband som vi vill fånga – i exemplet, kroppslängdens beroende av kön. Och slumpmässigheten (variationen som beror på faktorer utanför analysen) slår, inte oväntat, ut på olika specifika sätt i olika urval. I det här exemplet har vi, som antytts, arbetet med några implicita förutsättningar om hur kroppslängden beror av kön och hur den beror av slumpmässiga faktorer utanför analysen. Ett sätt att specificera förutsättningarna så att resultatet följer med matematiskt stringens är att anta, att längd hos män respektive kvinnor är normalfördelade med olika väntevärden men samma standardavvikelse. Det går då att bevisa, att den i längden bästa beslutsregeln är att lägga en gräns mitt emellan de två medelvärdena i stickprovet. Vårt kvalitativa resonemang är dock robust inför stora variationer av dessa förutsättningar, och det finns ingen anledning att analysera detta enkla exempel närmare.

Låt oss istället ta en titt på det tvådimensionella fallet och analysera vårt fiktiva medicinska exempel. Vi börjar här med ett matematiskt resultat och betraktar sedan situationen mer principiellt. Datapunkterna i figur 71 och 72 skulle kunna vara tagna från två tvådimensionella, symmetriskt normalfördelade slumpprocesser som bara skiljer sig åt med avseende på väntevärdet. Jämför de streckade cirklarna i figur 73, som symboliserar spridningen runt dessa väntevärden. I så fall kan man ganska enkelt bevisa (vi gör det inte här) att den bästa beslutsgränsen är den räta linjen som går mitt emellan stickprovens medelvärden, vinkelrätt mot deras förbindelselinje:



Figur 73. Den bästa beslutsgränsen, under vissa antaganden, för datamängden från figur 71. Förklaring: se text.

Denna beslutsgräns klassificerar faktiskt ännu *sämre* i träningsmängden än den (tänkta) enkla perceptronens linjära separation i figur 72. Men under de givna antagandena kommer *varje* annan beslutsregel än den i figur 73 – inklusive den som vårt tänkta kraftfulla nätverk åstadkommit i figur 72 – att leda till sämre resultat i det långa loppet. *Färre* felaktiga beslut i den ursprungliga datamängden leder alltså till *fler* felaktiga beslut för senare data!

Man kan förstå detta resultat som följer, helt i analogi med det endimensionella ”längd”-exemplet. Låt oss anta att mätvärdena för de två variablerna x_1 och x_2 visserligen delvis beror av om personen ifråga är sjuk eller frisk, men också påverkas av slumpprocesser som i sin tur är betingade av mätfel eller av andra (okontrollerade) variabler. Den avancerade algoritmen kodar då fullt ut den variation i de aktuella data som beror på slumpen (dvs. på variabler som inte är med i analysen). Men denna komponent av variationen ger oss ingen sannolikt giltig information om framtiden, utan snarare *desinformation*. Den enklare algoritmen är alltså egentligen på ett sätt den mer sofistikerade. Genom att den inte klarar av att koda slumpens inflytande (desinformationen) perfekt, kommer den att lägga större vikt vid det som *inte* beror på slumpen, och kommer därför att förutsäga framtiden bättre.

Neurala nätverk och kurvanpassningsproblemet

Ovanstående resonemang kan också formuleras i termer av artificiella neurala nätverk uppfattade som metoder för *regression.* Det handlar alltså om att fånga sambandet, sådant det avspeglar sig i en viss datamängd, mellan en eller flera kontinuerliga “invariabler” och en eller flera likaledes kontinuerliga “utvariabler”. Vi har redan stött på begreppet *linjär regression* (avsnitt 4.5) och tittat på ett nätverk som kan utföra sådan (avsnitt 4.6). Man talar som nämnts också om *olinjär regression*, dvs. metoder som anpassar parametrarna i en icke-linjär funktion av en viss typ till en given datamängd. Ett exempel är *polynomapproximation,* där man försöker hitta de koefficienter i ett polynom (av ett visst gradtal) som ger den bästa anpassningen till data. Några av de nätverk som vi ska tala om nedan kan – liksom för övrigt polynomapproximationen – ses som *universalinstrument för olinjär regression.* Gör man dem bara tillräckligt komplexa kan de nämligen lära sig att approximera varje “snäll” funktion (varje funktion som har ett ändligt antal diskontinuiteter). Konkret betyder detta att ett sådant nätverk, när det tränas på en mängd av “indata” och motsvarande “utdata” som stämmer med en viss funktion, kan hitta

vikter som gör att nätverket för varje indatum producerar rätt utdatum med en godtycklig grad av noggrannhet.

Begränsar man sig till en dimension i vardera input och output får vi det man ofta brukar referera till som *kurvanpassningsproblemet:* att hitta den funktion y = f(x) som bäst förklarar en iakttagen mängd M av parvis värden <x,y>. De komplexa neurala nätverkens förmåga till universell approximation medför att man för varje konsistent sådan mängd (dvs. sådan att alla punkter med olika y-koordinater också har olika x-koordinater) kan träna ett neuralt nätverk till att hitta en kurva som går godtyckligt nära *alla* punkterna. Denna utmärkta egenskap måste dock ses i belysning av generaliseringsproblemet. Ju mer komplex den kurva är som nätverket dragit, desto större är risken att vi också fångat en slumpvariation som inte kan läggas till grund för generalisering. Det är därför ofta så att en mindre komplex kurva, som inte går exakt genom alla punkterna, är ett bättre underlag för generalisering.

Generaliseringsproblemet är inte på något begränsat till neurala nätverk, utan gäller i lika hög grad andra icke-linjära statistiska metoder. Men forskare som sysslar med neurala nätverk är inte alltid lika vana att tänka i dessa termer som de som arbetar med andra icke-linjära metoder – dvs. statistiker av facket. Därför finns det all anledning att poängtera problematiken i samband med en framställning av neurala nätverk.

Är den enklaste beslutsregeln alltså alltid *den bästa?*

Nu måste det poängteras att våra resonemang hittills förutsätter, att ”oregelbundenheterna” i de iakttagna i data faktiskt väsentligen beror på faktorer som vi inte har tillräcklig information om. Denna förutsättning gäller naturligtvis inte alltid. Att en viss datamängd som härstammar från två klasser av observationer inte kan separeras på det enklast möjliga sättet *kan* också bero på, att data faktiskt avspeglar ett underliggande mer komplext samband mellan klasstillhörighet och värdena på de variabler som vi har med i vår analys.

De enklaste åskådningsexemplen på detta är återigen endimensionella. Både mycket hög vilopuls och mycket låg vilopuls är således tecken på sjukdom. En automatisk algoritm som försöker separera friskt från sjukt genom att använda ett enda gränsvärde, analogt med längden 175 centimeter ovan, *kan* därför inte vara den bästa. Det behövs två gränsvärden i detta fall, så att sjuka personer kan hamna i en av två disjunkta beslutsre-

gioner. Dessa gränsvärden behöver i sin tur inte vara absoluta, utan det kan finnas en överlappning runt båda, men det är en helt annan historia.

På samma sätt *kan* naturligtvis vår medicinska låtsasdatamängd i två dimensioner avspegla ett verkligt icke-linjärt samband mellan sjukdom och variablerna x_1 och x_2, ett samband som approximativt fångas av beslutsgränsen i figur 72. Detta kommer i så fall att visa sig genom att det kraftfulla nätverket gör bättre förutsägelser än algoritmen i figur 73. Mer allmänt: om den iakttagna olinjariteten i en datamängd faktiskt är i hög grad bestämd av de variabler som vi har med i analysen, och inte i särskilt hög grad av en slumpvariation, kommer det *inte* att finnas ett inverst samband mellan prestation i denna datamängd och prestation i det långa loppet. Tvärtom kommer det nu att krävas en mycket god prestation på originaldata för att *generaliseringen* skall bli tillräckligt olinjär! Resonemanget är givetvis inte bara tillämpbart på den situation då vi ska finna ett klassifikationskriterium, utan i lika hög grad på problematiken kring valet av modell för regression.

Det stora problemet som nu dyker upp är förstås: Hur ska vi kunna veta *i förväg* vilketdera som är fallet – beror icke-linjariteten i data bara på en slumpprocess som vi inte kan extrahera information om framtiden ur, eller avspeglar den en underliggande, systematisk olinjaritet som vi ska försöka koda, eller beror den på en blandning av båda – och i så fall, *vilken* blandning? Kort sagt, *hur ska vi veta hur kraftfull modell vi ska välja för analysen?*

Aprioribedömningar av lämplig modellstyrka

På denna fråga finns inget enkelt svar, men följande anmärkning är relevant: vi vet ofta genom vår tidigare erfarenhet av det erfarenhetsområde som data är hämtade från vilken typ av samband som är vanliga där. Genom vår tidigare kunskap om sjukdomar vet vi således att sjuka patienter kan ha både hög och låg vilopuls, och därför skulle vi knappast ens komma på tanken att i ett urval av patienter med olika sjukdomar försöka hitta en enda beslutsgräns, i termer av vilopuls, mellan sjukt och friskt. Kanske råkar vi också veta att data i figur 71 kan härledas till vissa patofysiologiska processer, som är benägna att åstadkomma kraftiga olinjariteter. I den mån vi har sådan kunskap, så skall den vägas in vid valet av metod. Och *mycket* ofta har vi *mycket* kunskap om området i förväg, åtminstone i form av mer eller mindre sannolika hypoteser. Biologiska processer *är* i stor utsträckning olinjära, vilket betyder att kraftfulla

analysmetoder ofta är motiverade när data är hämtade direkt från den biologiska verkligheten.

En annan aspekt av vår förhandskunskap är givetvis våra uppskattningar av vilken inverkan slumpfaktorer, eller faktorer som inte är med i analysen, har. Inte sällan vet vi till exempel att mätfelet är stort (jämför bilexemplet ovan), vilket bör göra oss mindre benägna att använda olinjära modeller. I andra sammanhang kan samma slutsats följa av att vi valt att studera endast ett fåtal av de variabler som vi vet är verksamma.

När det sedan gäller att på rätt sätt väga in vår förhandskunskap beträffande källan till olinjaritet, är det mycket viktigt att ta hänsyn till urvalets storlek. En olinjaritet som är urskiljbar i grafen över 1000 data beror, allt annat lika, med mindre sannolikhet på slumpen än vad en olinjaritet av samma storleksordning, som vi tycker oss se i 20 datapunkter, gör. Omvänt är det inte ofta som man har grundad anledning att använda olinjära analysmetoder på mycket små datamängder. En stor del av det missbruk av ANN-teknik som faktiskt förekom under 1980- och 90-talen var just av den typen: för kraftfulla nätverk på för små urval.

Redan att låta valet av metodik influeras av sina förhandskunskaper på ett intuitivt och kvalitativt sätt är generellt sett bättre än att inte alls använda sina förhandskunskaper. Det har också i flera sammanhang formulerats kvantitativa tumregler som ger mer specifik vägledning, till exempel i form av hur många förbindelser mellan neuron som ett neuralt nätverk får ha, givet antalet av datapunkter som det ska analysera. Dessa tumregler har mycket begränsad giltighet, men det är bättre att gå efter dem än att inte fundera alls över generaliseringsproblematiken.

Det finns en matematisk teori som ger ett allmängiltigt och exakt, men ändå kontroversiellt, svar på frågan om hur kraftfull metod man bör välja för sina data. Vi talar om bayesiansk inferensteori, som beskriver hur man ska väga samman i förväg givna sannolikheter (apriorisannolikheter) med data (jämför också avsnitt 5.1). I neurala nätverkssammanhang har teorin sin konkreta tillämpning i form av *bayesianska inlärningsalgoritmer*. Bayesiansk inlärning har alltså bland annat den fördelen, att generaliseringsproblematiken får en exakt, sannolikhetsteoretiskt grundad lösning. En annan fråga är om de förutsättningar beträffande tillgängliga apriorisannolikheter som metodiken kräver någonsin är uppfyllda. Bayesianismens kritiker förnekar att så är fallet, utom möjligen i enstaka, ganska ointressanta situationer. Mer om detta i avsnitt 9.4 nedan.

Aposterioriprövning av modellstyrka

Det är långtifrån alltid som vår förhandskunskap ger oss ett särskilt gott underlag för att avgöra hur komplex nätverksmodell som vi bör välja. Lyckligtvis kan man också pröva sig fram, och det gör man helt enkelt genom att testa hur bra nätverket generaliserar. Man prövar det alltså, med den den viktuppsättning man nått fram till i den ursprungliga datamängden (”träningsdata”), på en *annan datamängd* än den ursprungliga (”testdata”). Om prestanda då försämras påtagligt har man sannolikt ett alltför komplext nätverk, och/eller man har tränat det för mycket (jämför nedan).

Denna prövning på testdata kan göras mer eller mindre omfattande, och kan byggas in som ett mer eller mindre automatiskt moment i konstruerandet av nätverket. I moderna, kommersiellt tillgängliga program för statistisk analys med ANN är basrutinen således ofta den, att de tillgängliga data delas in i *tre* delmängder. Programmet använder data i den *första* av dessa (träningsmängden) för att träna ett antal neurala nätverk av olika komplexitet (och eventuellt av olika typ). Det testar samtidigt hur vart och ett av dem, när det tränats färdigt, presterar på data från den *andra* delmängden (testmängden). Det ”bästa” nätverket kan väljas på olika sätt, men både prestationen i träningsmängden och förmågan att upprätthålla denna prestation i testmängden skall vägas in. Ibland erbjuds också alternativet att sätta samman en ”allra bästa” analysmetod genom att kombinera rösterna från flera nätverk. För att få en uppskattning av hur bra metodiken kommer att fungera utanför den ursprungliga datamängden prövar man slutligen detta bästa nätverk (eller detta forum av bästa nätverk) på den *tredje* delmängden av data, *valideringsmängden*. Givet den nämnda strukturen hos analysen kan man nämligen inte utgå från prestationen på tränings- och testdata när man gör en sådan förutsägelse. Prestationen på tränings- och testdata var ju själva grunden för urvalet av nätverk, och risken finns därför att det nätverk som väljs ut som ”bäst” är det som bäst lyckas fånga en *gemensam slumpvariation* i tränings- och testdatamängderna (jämför problematiken med ”mass-signifi-kans” i traditionell signifikanstestning av hypoteser)!

”Korsvalidering” refererar i ANN-sammanhang till ett speciellt sätt att dela in originaldatamängden i ett antal delmängder, och sedan *alternerande* använda dessa för träning och testning. Rent allmänt sett är korsvalidering en osäkrare metod än *extern validering* (dvs. prövning på helt nya data), men proceduren är ett acceptabelt alternativ.

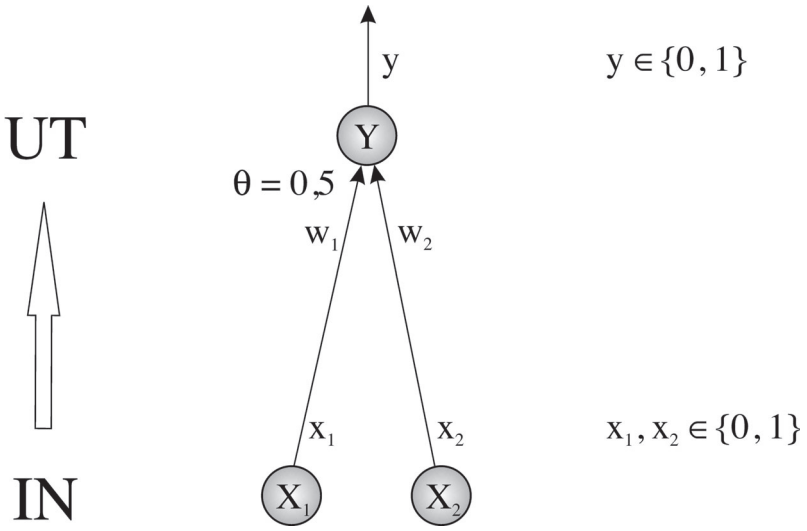
Hur gör man ett nätverk mer eller mindre olinjärt?

Det grundläggande sättet att skapa en komplex nätverksalgoritm är att välja ett nätverk med *flera lager av modifierbara vikter* och *olinjära element i något dolt skikt* (dvs. mellan input- och outputskikten) – till exempel en MLP, ett LVQ-nät eller ett RBF-nät. Inom ramen för en sådan modell blir nätverket mer kraftfullt *ju fler dolda noder det har*. Antalet dolda noder bestämmer man normalt när man sätter upp sin nätverksmodell, men det finns också automatiska algoritmer som kan få ett nätverk att växa eller krympa, allt eftersom resultaten anvisar att det är för enkelt respektive för komplext. Slutligen bestäms graden av olinjaritet av *hur höga vikterna i nätverket är*. Vikterna är normalt ganska små vid starten och växer med tilltagande träning, och man kan därför styra nätverkets styrka genom att *träna kortare eller längre tid*. En vanlig automatisk metod för ”on-line-kontroll” av ett nätverks komplexitet, *regularisering*, gör samma sak genom att ”straffa” alltför höga vikter. Detta kan enklast ske genom att vikterna får lämna ett bidrag till felfunktionens värde (se vidare nedan).

9.2 Betydelsen av dolda noder

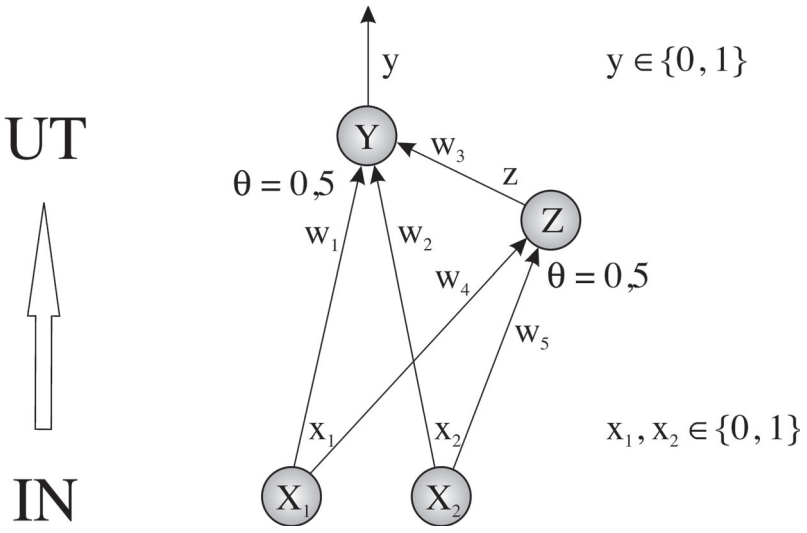
Låt oss efter denna långa men nödvändiga exkurs återgå till att bygga några nya neurala nätverk.

I feed-forward-nätverk med fler än ett lager av förbindelser, dvs. fler än två skikt av noder (element), brukar man referera till enheterna i skikten mellan input- och outputnoderna som *dolda* enheter. Beteckningen syftar till att användaren inte interagerar direkt med dessa noder. Ett dolt skikt av olinjära enheter ger kraftigt ökade möjligheter åt exempelvis en perceptron med stegfunktion. Figur 74 föreställer en enkel binär perceptron (med tröskeln 0,5) sådan som vi känner den från tidigare:



Figur 74. En enkel binär perceptron igen. w_j: vikter. x_i: aktiviteter i inputnoderna X_i. y: aktivitet i outputnoden (”beslutsnoden”) Y. θ: tröskel för Y.

I figur 75 har vi lagt till ett minimalt mellanliggande skikt i form av en extra nod Z (också den binär med tröskeln 0,5). Denna nya nod väger in aktiviteterna från X_1 och X_2 och signalerar sedan sin aktivitet z till outputnoden. Vi förutsätter att informationen som kommer till Y direkt från X_1 och X_2 anländer till Y samtidigt med signalen från Z.



Figur 75. Perceptronen i föregående figur utökad med en dold nod, Z, med tröskel 0,5. Beteckningar i övrigt som i figur 74.

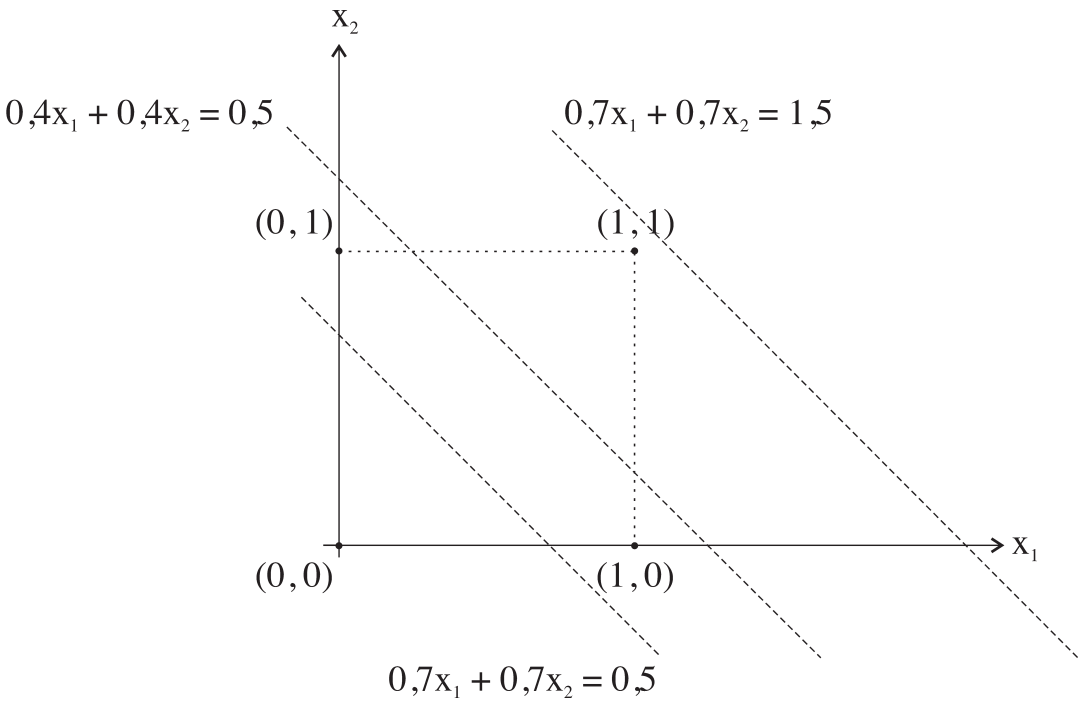
Det är lätt att inse att exempelvis följande vikter löser XOR-problemet:

(9.2.1) $w_1 = w_2 = 0{,}7$
$w_3 = -1$
$w_4 = w_5 = 0{,}4$

Nätverket ger då nämligen output 1 endast när följande ekvationer gäller:

(9.2.2) $0{,}7\ x_1 + 0{,}7\ x_2 > 1{,}5$ (nämligen om z = 1), *eller*
$0{,}7\ x_1 + 0{,}7\ x_2 > 0{,}5$ *och* $0{,}4\ x_1 + 0{,}4\ x_2 < 0{,}5$ (fallet z = 0)

Detta är ett splittrat område som dels ligger *mellan de två vänstra* linjerna, dels *till höger om den högra* i figur 76.



Figur 76. Beslutsgränserna för perceptronen i figur 75. x_i: aktiviteter i inputnoderna X_i. Markerade punkter: de fyra inputvektorerna. Förklaring i övrigt: se text.

För binära inputs faller nu bara (1, 0) och (0, 1) innanför det inputområde som ger 1 som utvärde.

Vad den nya dolda enheten Z i figur 76 gör kan beskrivas som att den signalerar för en viss klass av inputs (”kodar” denna klass) på ett sätt,

som (ur outputnodens synvinkel, så att säga) förvandlar det ursprungliga problemet till ett som *är* linjärt diskriminerbart. Noden eliminerar den svårighet som det ursprungliga nätverket hade att ge rätt signal på (1, 1) genom att i just detta fall skicka en extra signal som kompenserar för signalerna från inputnoderna. Det inses lätt att mer komplicerade perceptroner med stegfunktion kan ”finfördela” beslutsområdet på mycket mer raffinerade sätt genom fler dolda enheter, som på analogt sätt kodar särskilda klasser av inputs (man kan också säga: kodar för vissa egenskaper hos inputvektorn).

Samma princip gäller, ska vi se, även för andra flerlagrade nätverk. Genom att de olika noderna i det dolda skiktet (eller de dolda skikten) kodar för specifika egenskaper hos inputs, så kan nätverkets slutgiltiga beslut fattas på samma enkla sätt som i den enkla perceptronen eller den linjära associatorn. Outputenheterna i dessa nätverk kan därför vara linjära; den väsentliga olinjariteten ligger i de dolda skikten.

För att de flerlagrade nätverken ska bli verkligt kraftfulla är det, vilket påpekats flera gånger, också av central betydelse att vikterna på förbindelserna till de dolda skikten är modifierbara genom träning. Annars blir nätverkens prestationer begränsade till en förutbestämd, snäv klass av olinjära problem (jämför avsnitt 6.3 ovan).

9.3 Inlärning i flerlagrade perceptroner

Flerlagrade perceptroner med sigmoida aktiveringsfunktioner

Rosenblatt kände förstås mycket väl till att flerlagrade, olinjära perceptroner är mycket kraftfullare än enlagrade sådana. Problemet var bara att han inte kunde finna någon inlärningsalgoritm som fungerade för dem, dvs. en algoritm som hittade de principiellt möjliga lösningarna genom träning av vikter även i de dolda lagren. Än i dag finns ingen användbar sådan algoritm för flerlagrade perceptroner med stegfunktion. Däremot fann flera forskare på 1970- och 80-talen mer eller mindre oberoende av varann, att problemet i princip kunde lösas för flerlagrade perceptroner som arbetar med kontinuerliga, sigmoida aktiveringsfunktioner. Vi skall strax kort beskriva den algoritm som gav den första allmänt kända lösningen, och som ofta kallas ”back propagation of error”.[151]

[151] Werbos (1974) torde ha varit den förste som levererade en fullständig formulering

Låt oss först återigen formulera en viktig egenskap hos flerlagrade perceptroner med sigmoida aktiveringfunktioner: *Dessa maskiner kan, till skillnad från enkla perceptroner och linjära nätverk, i princip dra beslutsgränser av vilken önskad form som helst.* Detta beror i sin tur på att de kan godtyckligt noga approximera en godtycklig funktion, förutsatt att funktionen är kontinuerlig eller har ett ändligt antal diskontinuiteter. För att ha egenskapen av universalapproximator behöver nätverket i princip bara *ett* dolt lager av enheter, men det kan behövas många enheter i detta dolda lager.[152] För praktiska ändamål kan det ibland vara enklare att arbeta med fler dolda lager. Det räcker, som nämnts, i princip med att det dolda lagret (de dolda lagren) arbetar med sigmoida funktioner. Outputlagret låter man vid regressionsuppgifter som regel vara linjärt. Om man väljer en logistisk outputfunktion i en MLP är det inte för att nätverket ska bli kraftfullare, utan för att det gäller en klassifikationsuppgift och man vill kunna tolka output som *sannolikheter* för tillhörighet till olika klasser.

En MLP med sigmoida dolda enheter kan inte bara *i princip* dra mycket olinjära gränser. Den kan som regel också ofta *tränas att hitta* den gräns som användaren önskar ha dragen. En klassisk algoritm för detta går ofta under namnet ”back propagation of error” (BP). Denna term används i flera olika betydelser, och i en snäv mening står den istället för en viss *komponent* i den algoritm som vi nu ska beskriva. Den fullständiga proceduren innebär (jämför också avsnitt 4.6 ovan):

(9.2.3) (a) man beräknar det sammanlagda felet E i output, exempelvis som summan (över alla inputs och alla outputnoder) av de kvadrerade skillnaderna mellan önskad och verklig output;

(b) man använder en viss stegvis matematisk procedur för att beräkna hur outputfelet beror av vikterna i nätverket. Proceduren kan beskrivas som att felet fortplantas bakåt i nätverket, från outputenheterna till vikterna från inputskiktet.

(c) Vikterna korrigeras i omvänd proportion till deras respektive bidrag till detta fel (gradientnedstigning).

”Back propagation” i *snävare* mening syftar på momentet (b), dvs. den särskilda metod med vilken vikternas bidrag till outputfelet beräknas. av algoritmen.

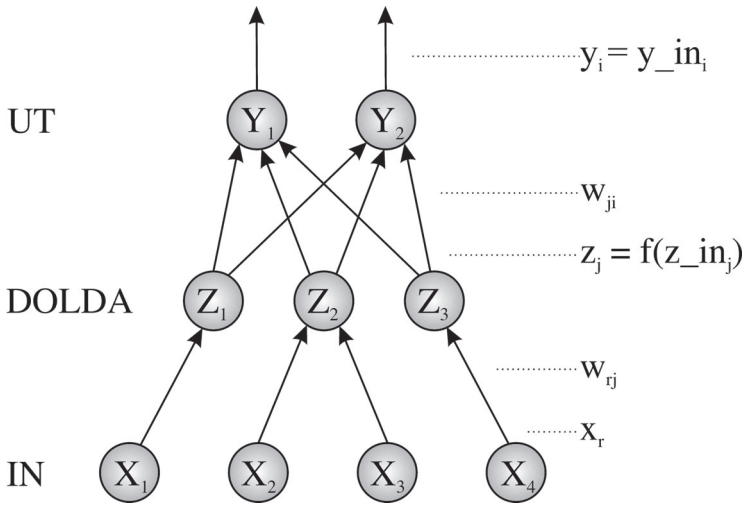
[152] För ett kort bevis se Bishop (1995), ss. 130f.

Denna metod kan kombineras med andra felfunktioner än den vanliga kvadratiska, och med andra sätt att söka felminimum än genom gradientnedstigning. Vi kommer att återkomma till båda dessa möjligheter nedan, men låt oss först beskriva den klassiska kombination som (a)–(c) står för.

Observera dock allra först att back propagation i snäv mening, dvs. sättet att räkna sig bakåt genom lagren för att räkna fram vikternas bidrag till outputfelet, är en rent beräkningsteknisk historia och inte (behöver) motsvaras av några återkopplade förbindelser i nätverket! Den flerlagrade perceptronen med back-propagation-algoritm är fortfarande ett typiskt feed-forward-nät, och såväl inmatningen av den önskade output och användningen av denna för att korrigera vikterna har ingen explicit motsvarighet i nätverket som sådant. Vi har alltså att göra med övervakad inlärning.

Back propagation of error – klassisk version

Mycket av det som följer nu är en repetiton av vad som redan sagts om deltaregeln i kapitel 4, och avsnittet ska därför gå att följa utan större svårigheter. Betrakta figur 77, som framställer ett tvålagrat nätverk med fyra inputenheter, tre dolda sigmoida enheter och två fullt linjära utenheter. (Av de 12 förbindelserna från inputlagret har endast 4 ritats ut.)



Figur 77. Ett nätverk för BP-algoritmen. Förklaring se text.

Vi väljer x, z och y som beteckningar på aktiviteter i inputskikt, dolt skikt respektive outputskikt. Antag att vi har en inputvektor $\mathbf{x}_k$ som vi vill skall

ge output $\mathbf{d}_k = (d_{k1}, d_{k2})$. Den ger de facto output $\mathbf{y}_k = (y_{k1}, y_{k2})$. Ett lämpligt mått på felet för denna input är nu summan

(9.2.4) $$E_k = (d_{k1} - y_{k1})^2 + (d_{k2} - y_{k2})^2$$

och mer generellt:

(9.2.5) $$E_k = \sum_{i=1}^{m} \left(d_{ki} - y_{ki}\right)^2$$

dvs. man summerar de kvadrerade skillnaderna mellan verklig och önskad output över alla de m st outputnoderna. Vi vill emellertid veta inte bara hur fel nätverket gör vid just denna input, utan hur fel det gör totalt sett i hela datamängden. Det betyder att vi ska summera en gång till, nu över alla inputvektorer som vi kan anta är p st. Alltså:

(9.2.6) $$E = \sum_{k=1}^{p} \sum_{i=1}^{m} \left(d_{ki} - y_{ki}\right)^2$$

Precis som vi gjorde i resonemanget om deltaregeln (avsnitt 6.2) skall vi nu betrakta *felet som en funktion av vikterna* i nätverket och sedan titta på *de partiella derivatorna av denna funktion med avseende på vikterna*. Att det totala felet E, givet nätverkets design, en viss mängd av inputdata samt de önskade outputs, är entydigt bestämt av nätverkets vikter (dvs. E är en funktion av dessa) är lätt att inse. Givet alla de nämnda faktorerna är det ju vikterna som bestämmer output, och därmed felet. Vi kan alltså skriva, förutsatt att det finns n vikter i nätverket:

(9.2.7) $$E = f(\mathbf{w}) = f(w_1, w_2, ..., w_n)$$

Om n = 1 eller 2 kan man visualisera E som en kurva, respektive som en yta, i ett 2- respektive 3-dimensionellt koordinatsystem med E på en axel och vikterna på de övriga. Vi gjorde detta tidigare (figur 42, avsnitt 6.2) när det gällde felfunktionen i en viss liten linjär associator med två vikter, titta gärna på den bilden igen! Redan vårt senaste lilla nätverk (figur 78) har dock 18 vikter, och det är tyvärr inte lätt att föreställa sig ett 19-dimensionellt rum... men felfunktionen är alltså en hyperyta i ett sådant.

Eftersom felet E är en funktion av alla vikterna kan vi nu fråga, *Hur ändrar sig felet, givet en viss uppsättning* $\mathbf{w}$ *av vikter, om man bara*

ändrar en enskild vikt w_i? Om f är en överallt kontinuerlig och deriverbar funktion är detta detsamma som att fråga efter den partiella derivatan av E med avseende på w_i, $\partial E/\partial w_i$, i punkten **w**. Denna partiella derivata är ett mått på hur mycket E-hyperytan *lutar i w_i-led* i punkten ifråga. Vi har beskrivit hur den kan användas för att hitta ett viktminimum för den linjära associatorn. En flerlagrad perceptron med sigmoida dolda enheter har, givet en viss inputdatamängd, en kontinuerlig och överallt deriverbar felfunktion, så de eftersökta derivatorna existerar, och de kommer till användning på i princip samma sätt här.

Man utgår från derivatorna av felen E_k (dvs. för en input i taget) och summerar sedan resultaten över alla inputs. Back-propagation-algoritmens kärna (back propagation-momentet i snäv mening) är just det speciella sätt som de partiella derivatorna $\partial E_k/\partial w_j$ beräknas på.[153] Proceduren för att beräkna en term av typen $\partial E_k/\partial w_j$ startar med observationen att en förändring i en förbindelses vikt kan påverka outputfelet *endast* genom att ändra nettoinput till den enhet som förbindelsen ifråga går till. Detta gör att man kan tillämpa den så kallade kedjeregeln för derivator på följande sätt. Man räknar först ut hur *förändringar i nettoinput till outputenheterna påverkar felet E_k*, dvs. man bestämmer $\partial E_k/\partial y\_in_i$ för de olika outputenheterna Y_i. Detta är lätt för linjära outputenheter och en kvadratisk felfunktion, vilket vi redan sett i härledningen av deltaregeln. Vi får nämligen

$$\frac{\partial E_k}{\partial y\_in_i} = 2(y_i - d_i) \qquad (9.2.8)$$

Om outputfunktionen inte är linjär och/eller vi har en annan felfunktion ersätts 9.2.8 av någon annan enkelt beräkningsbar formel. Sedan räknar man ut hur förändringar hos *vikterna hos förbindelserna till outputnoderna förändrar nettoinputs till dessa noder*, allt annat lika; dvs. $\partial y\_in_i/\partial w_{ji}$ beräknas för alla vikter i det sista lagret. Notera här att $\partial y\_in_i/\partial w_{ji}$ helt och hållet beror på aktiviteten z_j i den nod Z_j som skickar sin signal via vikten w_{ji}; närmare bestämt är $\partial y\_in_i/\partial w_{ji} = z_j$. En tillämpning av kedjeregeln för derivator ger oss omedelbart alla $\partial E_k/\partial w_j$ för det *sista* lagret av vikter:

---

[153] Den läsare som inte känner sig hemmastadd med partiella derivator kan hoppa över de närmaste två sidorna – fram till underrubriken *Vad gör man då med derivatorna?* – utan att sammanhanget behöver gå förlorat.

(9.2.9) $$\frac{\partial E_k}{\partial w_{ji}} = \frac{\partial E_k}{\partial y\_in_i} \cdot \frac{\partial y\_in_i}{\partial w_{ji}} = 2z_i\,(y_i - d_i)$$

vilket, inte oväntat, låter precis som deltaregeln.

Nästa steg innebär att vi gör om samma procedur för det *näst* sista skiktet av noder. Vi håller först alla vikter fixa och betraktar hur nettoinput till noderna i det näst sista lagret påverkar outputfelet. Det innebär en beräkning av $\partial E_k/\partial z\_in_j$, där z_in_j är en nettoinput i detta lager. För detta används också kedjeregeln. Nettoinput z_in_j påverkar ju bara outputfelet via förändringar i aktiviteten z_j som fortplantas genom vikten w_{ji}. Men $z_j = f(z\_in_j)$, där f är aktiveringsfunktionen för enheterna i Z-lagret. Vi kan därför börja med att beräkna alla $\partial y\_in_i/\partial z\_in_j$ genom formeln:

(9.2.10) $$\frac{\partial y\_in_i}{\partial z\_in_j} = f'\left(z_j\right) w_{ji}$$

Om vi har valt en lämplig aktiveringsfunktion, till exempel den logistiska funktionen, är också dess derivata f' lätt att beräkna. För att få fram en formel för $\partial E_k/\partial z\_in_j$ summerar vi nu över alla outputnoderna samtidigt som vi använder kedjeregeln:

(9.2.11) $$\frac{\partial E_k}{\partial z\_in_j} = \sum_{i=1}^{m} \frac{\partial E_k}{\partial y\_in_i} \cdot \frac{\partial y\_in_i}{\partial z\_in_j} = f'\left(z_j\right) \sum_{i=1}^{m} w_{ji} \cdot \frac{\partial E_k}{\partial y\_in_i}$$

vilket med en logistisk aktiveringsfunktion och linjär output blir ett tämligen enkelt uttryck:

(9.2.12) $$\frac{\partial E_k}{\partial z\_in_j} = f\left(z_j\right)\left(1 - f\left(z_j\right)\right) \sum_{i=1}^{m} w_{ji} \cdot 2(y_i - d_i)$$

Det som givit back propagation-algoritmen dess namn, och det som gör den särskilt beräkningseffektiv, är just att man på detta sätt kan erhålla bidraget till felet från nettoinput till en nod i ett lager ur motsvarande bidrag från nettoinput till noderna i *nästa* högre lager.

När vi på detta sätt beräknat hur nettoinput i de dolda noderna bidrar till outputfelet är det enkelt att (på samma sätt som nyss) räkna ut hur vikterna som går till de dolda noderna påverkar felet. De gör det nämligen i

proportion till aktiviteterna i de noder som finns i början av förbindelserna:

$$\frac{\partial E_k}{\partial w_{rj}} = x_r \frac{\partial E_k}{\partial z\_in_j} \tag{9.2.13}$$

Därmed är beräkningen avslutad om det rör sig om ett tvålagrat nätverk, bortsett från summationen över alla inputs. Den senare delen av proceduren upprepas om det finns ytterligare ett eller flera lager av vikter.

Vad ska vi använda derivatorna till?

Nåväl, låt oss anta att BP-algoritmen beräknat alla de relevanta partiella derivatorna. Hur ska då vikterna ändras för att nätverket skall prestera bättre? Den ursprungliga och enklaste varianten av BP använder sig av gradientnedstigning, dvs. man låter viktvektorn flytta sig en liten bit nedåt i den riktning i vilket vikthyperplanet lutar mest. Denna brantaste riktning hittar man direkt som riktningen, med omvänt tecken, för vektorn av alla de partiella derivatorna. *Att gå ett steg den brantaste vägen neråt från en punkt på vikthyperytan är med andra ord detsamma som att korrigera vikterna i proportion till deras respektive bidrag till felet.* Gör man på detta sätt, och dessutom ser till att ta mindre och mindre steg, är man garanterad att hamna i en minimipunkt. Precis som för deltaregeln med den linjära associatorn gäller att metoden fungerar även om man beräknar felet efter varje enskild presentation av en input.

Att inte fastna i lokala optima

Vi ska strax se att det finns metoder som är både snabbare och i andra avseenden bättre än enkel gradientnedstigning för att leta sig ner till felminima. Först måste vi dock uppmärksamma ett problem som är gemensamt för alla dessa metoder, om än inte riktigt lika svårbemästrat för dem alla. Detta problem är att det för en flerlagrad perceptron – till skillnad från vad som är fallet i en linjär associator – oftast finns *lokala minima* på viktytan, vilket innebär att den punkt i vilken man hamnar inte behöver stå för det *minsta* fel som perceptronen *i princip* kan uppnå för den givna datamängden. Exempelvis kan man mycket väl fastna i ett lokalt minimum där det kvarstår ett fel, trots att det *finns* en helt felfri lösning. Vad man kan bevisa om ett MLP-nätverk som korrigererar vikter med gradientnedstigning (eller någon av de vanliga modifikationerna av denna

procedur) är alltså bara att inlärningen garanterat konvergerar till ett *lokalt* felminimum, även i de fall då det finns ett lägre globalt minimum, kanske rentav med E = 0. Viktvektorns beteende enligt algoritmen som vi beskrivit är helt analogt med att en kula som läggs på en mycket gropig plan rullar ner i närmaste grop, vilken inte behöver vara den djupaste gropen på planen.

Detta verkar ju inte vara ett önskvärt beteende hos en neural-nätverks-algoritm. För att öka sannolikheten att nå det globala optimum (förhoppningsvis med E = 0) har man därför förbättrat BP-algoritmen på ett antal olika sätt, bland annat genom att lägga till en tröghetsfaktor (”moment”) till viktförändringarna. Momentet gör att risken att fastna i mycket små ”gropar” i fel-landskapet minskar. En annan enkel metod är att starta om nätverket med nya utslumpade vikter om resultatet inte var till belåtenhet. Ingen av de existerande modifikationerna ger en hundraprocentig garanti att man hittar det globala optimum inom en given tidsram. Detta problem är dock för de flesta problem av klart underordnad betydelse i förhållande till generaliseringsproblematiken, som vi snart ska återkomma till. Först dock några ord om några sätt att snabba upp den algoritm som vi beskrivit ovan.

Snabbare sätt att hitta minima

För praktiska ändamål använder man numera sällan enkel gradientnedstigning. Flera alternativa metoder har föreslagits, de flesta baserade på att man på ett eller annat sätt tar *andraderivatorna* av felfunktionen med i beräkningen, dvs. uttryck av typen $\partial^2 E_k / \partial w_i \partial w_j$. En intuitiv motivering för detta är att andraderivatan av en funktion ju ger ytterligare information, utöver förstaderivatan, om ”vart funktionen är på väg”. (Om man vet inte bara att marken lutar neråt, utan också att det håller på att bli brantare och brantare, så kan man anta att man kommer att hamna en bit uppe i luften om man bara fortsätter neråt i tangentens riktning.) När det gäller neurala nätverk handlar det om att beräkna en hel matris av andraderivator av felfunktionen, den så kallade *hessianska matrisen*. Lyckligtvis kan man använda varianter av back-propagation-resonemanget även vid dessa beräkningar, varför de inte kräver orimligt mycket beräkningskraft.

Vi kan inte gå in i detalj på hur de alternativa metoderna fungerar, långt mindre försöka härleda någon av dem, men det kan vara värt att nämna att tre av de mest använda undertyperna är *konjugat-gradientmetoder*,

kvasi-Newtonska metoder och *Levenberg-Marquart-algoritmen*. Matematiken kring dessa metoder hör närmast hemma i den allmänna teorin för optimeringsmetoder, och skillnaderna mellan dem är inte särskilt relevanta för den grundläggande förståelsen av hur ett artificiellt neuralt nätverk fungerar. Valet mellan dem handlar ju bara om att hitta det globala minimum för felfunktionen så snabbt och effektivt som möjligt. Däremot kan det vara bra att känna till beteckningarna, eftersom algoritmerna finns tillgängliga i de flesta färdiga program för neurala nätverk.

Felfunktioner och statistikteori

I vår framställning har vi hittills koncentrerat oss på den felfunktion som innebär att man summerar de kvadrerade skillnaderna mellan önskad och verklig output. Det kan visas att denna felfunktion är lämplig vid de flesta regressionsproblem. Om vi kort betraktar fallet *linjär* regression (som ju kan utföras av ett linjärt nätverk och deltaregeln), så gäller att minimering av den kvadratiska felfunktionen leder till en väntevärdesriktig skattning av den linjära modellens parametrar. Motsvarande resultat gäller för icke-linjär regression med neurala nätverk: givet ett normalfördelat brus kring en icke-linjär kontinuerlig funktion kommer en optimalt tränad flerlagrat MLP som använder kvadratsummefelet och linjära outputenheter att komma godtyckligt nära funktionen i fråga, bara det får tillräckligt med data. Med tanke på de praktiska svårigheter som finns när det gäller att träna en flerlagrad perceptron, plus osäkerheten i antagandet om brusets karaktär, måste man givetvis alltid vara försiktig med slutsatsen att man *faktiskt* kommit nära verkligheten.

När vi kommer till klassifikationsproblem blir bilden en annan. Här kan man, givet rätt val av felfunktion och aktiveringsfunktion hos outputenheterna, tolka output som *sannolikheten för klasstillhörighet.* För detta ändamål är kvadratsummefelet inte lämpligt. I samband med klassifikationsuppgifter bör man därför istället välja en annan felfunktion, nämligen *ömsesidig entropi* (*cross-entropy*). Denna skall för tvåklassuppgifter kombineras med en *logistisk* outputenhet. Antag att nätverket tränas med output 1 som önskat värde för de inputs som härrör från den ena klassen. Det verkliga värdet på en output kan då, under mycket allmänna förutsättningar, tolkas som en ML-skattning av *sannolikheten* för att respektive input härrör från denna klass. Alternativt kan man (och för fler än två klasser måste man) använda lika många outputnoder som antalet klasser man vill diskriminera mellan. Som aktiveringsfunktion i dessa noder väljer man den generalisering av den logistiska funktionen som

ofta går under namnet *softmax.* Softmax-funktionen garanterar att summan av aktiviteterna i outputnoderna är 1. Valet av denna funktion, i kombination med ömsesidig entropi som felfunktion, gör att man (under samma allmänna förutsättningar som ovan) kan tolka aktiviteten i outputnod nr k som sannolikheten att input hör till klass nummer k. Härledningen av dessa resultat faller utanför ramen för denna bok, men skall man använda icke-linjära neurala nätverk för praktisk problemlösning bör man känna till resultaten.[154]

Överträning, modellstyrka och regularisering

I ljuset av vad vi har sagt ovan om relationen mellan komplexitet och generaliseringsförmåga är det inte alltid önskvärt att man reducerar outputfelet till det absoluta minimum som nätverket i princip är kapabelt till. Kanske måste man för att hitta detta globala minimum leta sig ner i små branta gropar i fel-landskapet, vilket betyder att den resulterande input-outputfunktionen har kraftiga lokala olinjariteter. Då kodar man sannolikt sådana egenskaper hos datamängden som beror på slumpvariation, snarare än avbildar en underliggande struktur. Man ska alltså sluta när man hittat en *lagom* bra lösning...

Vi har redan i viss mån diskuterat hur man ska veta när en lösningen är lagom bra: ett nyckelord är testdatamängd. Man kan under träningen av ett nätverk löpande undersöka hur bra det presterar i en testdatamängd, och stoppa träningen om och när skillnaden i prestation mellan träningsmängd och testmängd börjar öka påtagligt. Eftersom man då använt testdatamängden för att välja sitt optimala nätverk, måste prestanda i ytterligare en oberoende datamängd (som man ibland kallar ”valideringsmängden”) kontrolleras innan man accepterar nätverket som beslutsunderlag. Det är alltså fråga om samma slags procedur som den som kommer till användning när man bestämmer nätverkets optimala storlek.

En klass av algoritmer för att mer eller mindre automatiskt undvika extremt olinjära lösningar går under namnet *(vikt-)regularisering*. Denna brukar gå till så, att en term som växer monotont med alla vikterna (enligt någon för ändamålet utvald funktion) adderas till felfunktionens värde. Det betyder givetvis att det uppkommer ett nytt fel-landskap, där de lägsta punkterna sannolikt inte längre innebär mycket höga vikter. Regularisering av vikter är en elegant metod för att hålla nere ett nätverks

[154] Se vidare Bishop (1996).

komplexitet, och kan på så vis eliminera en del arbete med manuellt stoppande. Metoden ger dock i sig inget svar på frågan *hur mycket* man ska regularisera vikterna.

Tyvärr måste vi lämna dessa viktiga ämnen här. Den läsare som inte nöjer sig med vår informella diskussion hänvisas återigen till den just nämnda litteraturen (men se också slutet av nästa avsnitt).

9.4 Bayesianska neurala nätverk

Vad är ett bayesianskt nätverk?

Termen ”bayesianska nätverk” används för flera olika saker förutom de neurala nätverk som vi strax ska tala om. För det första betecknar termen ofta vad som helst bör kallas bayesianska påstående-nätverk, på engelska *Bayesian Belief Nets*, förkortat BBN (eller BN). Ett BBN är en speciell form av grafisk framställning av ett system av påståenden, som bär bestämda evidensrelationer till varandra. Dessa relationer är formulerade i termer av relativa sannolikheter. Genom att mata in sannolikheter för ett antal av påståendena kan man för det första (föga förvånande) automatiskt få ut sannolikheter för andra påståenden i systemet. Intressantare är att sådana nätverk kan *tränas* på en uppsättning data och sedan användas som beslutsunderlag i nya situationer. Men de har i övrigt inte särskilt mycket med neurala nätverk att göra, och vi ska inte tala mer om dem här.[155]

För det andra ser man rätt ofta att ett *neuralt* nätverk kallas ”bayesianskt”, utan att träningen av det utgår från bayesiansk inferensteori. Givetvis finns det många sätt att anlägga ett bayesianskt perspektiv på såväl biologiska som artificiella neurala nätverk.[156] Vi ska dock begränsa termen ”bayesianskt neuralt nätverk” (förkortat: BNN) till sådana nätverk som beskrivs i nästa avsnitt.

[155] Se Husmeier et al. (utg.) (2005).
[156] Jämför Doya et al. (2007), där informationsbehandlingen i nervsystemet betraktas i belysning av hur signaler från omvärlden skulle behandlas av ett ur bayesiansk synvinkel perfekt arbetande system.

Apriorisannolikheter för vikterna

I avsnittet om sannolikheter och statistikteori (5.1) kontrasterades ett traditionellt sätt att se på statistisk inferens, maximum-likelihood-principen, med ett bayesianskt synsätt. ML-principen innebär, i korthet, att man väljer den modell för data som ger dessa data den högsta sannolikheten. Det mesta av den hittillsvarande diskussionen om neurala nätverk kan sägas ha hållit sig inom ramen för ML-principen eller nära den. Vi ska nu ta en mycket snabb blick på vad den alternativa, bayesianska approachen kan innebära för ANN-teorin. För en djupare analys och en exakt formalism hänvisas åter till andra framställningar.[157]

Bayesiansk inferensteori innebär, för att repetera, två väsentliga avsteg från traditionellt tänkande i statistiken. För det första nöjer man sig inte med att bedöma hur bra modellen förklarar data, dvs. *datas sannolikhet givet modellen*, utan man försöker beräkna *modellens sannolikhet givet data*. För detta måste man införa apriorisannolikheter för olika modeller, vilket i allmänhet innebär kontinuerliga distributioner av sannolikheter över ett kontinuum av modeller. För det andra levererar man resultat inte bara i form av vilken modell som är *mest* sannolik, utan i form av *en aposterioridistribution över alla de olika modellerna*. Som en följd av detta kan man också göra prediktioner där man tar hänsyn till vad varje möjlig modell förutsäger; varje förutsägelse från en modell viktas med modellens aposteriorisannolikhet.

En bayesiansk approach till neurala nätverk innebär alltså att man hela tiden arbetar med *en stor mängd av möjliga nätverk* och med fördelningar över deras sannolikheter. Nedan ska vi för enkelhets skull förutsätta att strukturen hos dessa nätverk är känd, och att det rör sig om vanliga icke-linjära, flerlagrade feed-forwardnät med en fix arkitektur som har en viss regressionsuppgift att lösa. Det som skiljer de olika nätverken åt är då bara vikterna, och Bayesianen börjar därför med att anta en (flerdimensionell) *apriorifördelning* av sannolikheterna för olika viktuppsättningar. I det enklaste fallet bestämmer hon sig i förväg för hur denna distribution skall se ut. Här kan man välja en ”platt” eller en ”toppig” distribution. En platt distribution innebär att apriorisannolikheten för alla vikter inom ett stort intervall bedöms som ungefär lika stor. En distribution med en klar topp kring 0 innebär istället att nätverk med små vikter är mer sannolika, och medför därför en automatisk regularisering (se nedan).

[157] T.ex. Bishop (1996).

Ett mer generellt angreppssätt är att använda en så kallad ”hyperparameter” för hur platt aprioridistributionen är, och sedan arbeta med en överordnad distribution av denna hyperparameter. Vi ska dock nöja oss med att beskriva den enklare varianten, dvs. man antar en bestämd distribution av vikter.

”Träning” av ett BNN

Hur tränas då ett bayesianskt neuralt nätverk? Jo, det är helt och hållet en fråga om beräkningar med hjälp av Bayes’ teorem. För att kunna använda detta, och därigenom beräkna aposteriorifördelningen av de olika viktuppsättningarna, måste man först beräkna sannolikheten för de givna data, givet varje sådan uppsättning (precis som man gör i ML-paradigmet). Nu ska vi komma ihåg att de modeller vi arbetar med alltid räknar med en slumpfaktor – vi siktar inte på en modell som förklarar data perfekt, utan som förklarar dem så bra som möjligt, givet bruset i data. *Hur* bra modellen förklarar data beror på hur mycket (och vilket slags) brus som föreligger, och detta måste bayesianen ta hänsyn till. För att återigen välja det enklaste fallet, så anser man sig ibland ha sannolik kunskap att slumpfaktorn följer en viss normalfördelning. Har man denna typ av kunskap så kan *sannolikheten för data, givet en viss viktuppsättning hos nätverket*, beräknas. Om man *inte* anser sig ha specifik kunskap om brusets distribution så finns det bayesianska metoder för att skatta den från data, men inte heller dessa ska vi gå in på här.

Låt oss alltså anta att vi kan beräkna sannolikheten för den givna datamängden, givet varje viktuppsättning. I princip tillåter då Bayes teorem att vi beräknar *aposterioridistributionen* av vikter för nätverket, dvs. *sannolikheterna för olika viktuppsättningar, givet data*. Låt oss för en kort stund anta att vi lyckats med detta. Distributionen vi har nu är säkert rätt ”toppig”, eftersom vissa viktuppsättningar förklarar datamängden betydligt bättre än andra. Den högsta toppen (dvs. den mest sannolika viktuppsättningen, givet datamängden) ligger troligen *ganska* nära den viktuppsättning (den viktuppsättning som bäst förklarar data) som vi i bästa fall skulle funnit, om vi tränat ett nätverk med den givna arkitekturen på traditionellt sätt. Men eftersom vi nu har vägt in aprioriförde lningen kan det mycket väl finnas en påtaglig skillnad mellan resultaten.

En annan skillnad är, som redan påpekats, att bayesianen har ett mycket rikare slutresultat i sin hand i form av aposteriorisannolikheter för en stor mängd av *alternativa* nätverk. Hon kan därför göra mycket mer nyanse-

rade förutsägelser i nästa fall, baserade på hela denna mängd och återigen i form av en distribution av sannolikheter för olika utfall.

Denna ljusa bild kompliceras, förutom av den principiella problematik om apriorisannolikheter som vi redan talat nog om ovan, av att beräkningarna med Bayes' formel för neurala nätverk är synnerligen komplicerade och i många fall omöjliga att utföra exakt. Det handlar om integraler av en form som inte är analytiskt hanterbara. Man använder sig därför av avancerade numeriska metoder.[158] Dessa går, i korthet, ut på att man försöker ta stickprov ur populationen av viktuppsättningar i de regioner där vikterna förklarar data någorlunda bra, för att sedan approximera integralen över alla viktuppsättningar med en summa över de stickproven. Metoderna är mycket tidsödande, vilket väl är den främsta anledningen till att bayesianska neurala nätverk inte kan rekommenderas för alla regressionsuppgifter. Trots allt ger de ungefär samma resultat, i viktiga avseenden, som mer traditionella metoder, om man bara har tillräckligt med data! Ju fler data man matar in, desto mindre roll spelar ju eventuella toppar i vikternas aprioriförd elning (jämför slutet av avsnitt 5.1).

Den bayesianska referensramen erbjuder också metoder för att mer eller mindre automatiskt bestämma ett nätverks optimala komplexitet utifrån den givna datamängden. Istället för att välja en bestämd aprioridistribution av vikter, och därmed bestämma i förväg hur mycket man ska "straffa" höga vikter, kan man (som redan nämnts) arbeta mer förutsättningslöst med en distribution över "toppigheten" hos aprioridistributionen. En närmare diskussion av detta ämne skulle dock föra oss långt in på ett visserligen mycket centralt men samtidigt mycket avancerat område av statistikteorin (val av modellstyrka), där författaren inte alls känner sig hemma.

Som en sammanfattning av detta avsnitt och slutet av det föregående kan sägas, att ett korrekt sannolikhetsteoretiskt perspektiv först och främst kan visa oss vilken plats neurala nätverk har om de betraktas som instrument för statistisk inferens i den klassiska ML-traditionens anda. Men i förlängningen visar ett sådant perspektiv – genom de bayesianska neurala nätverken – också på stora möjligheter både till en fördjupning av teorin för ANN och till framtida, ännu inte realiserade användningsmöjligheter för neurala nätverk.

[158] Den viktigaste av dessa går under namnet "Mixed Monte Carlo/Markov Chain".

9.5 LVQ, Learning Vector Quantization

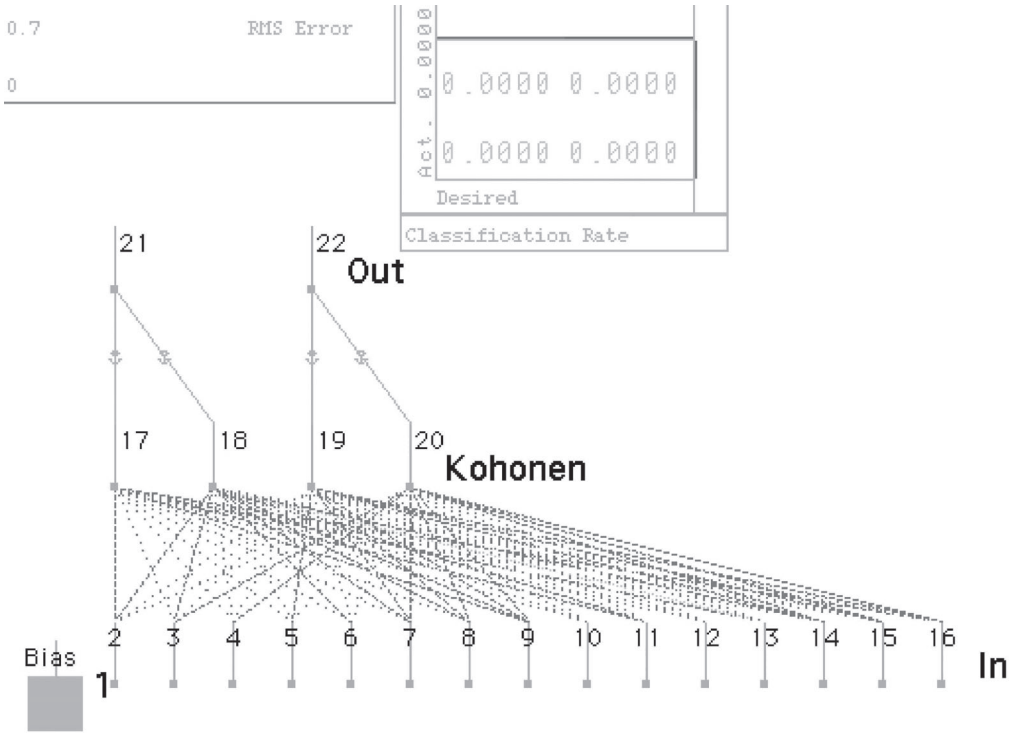
Arkitektur och signaldynamik

Detta nätverk är ytterligare en av Teuvo Kohonens genialt enkla uppfinningar. LVQ fungerar efter samma grundprincip som hans självorganiserande karta (SOM), men nu är det fråga om övervakad klassifikation.

Nätverket har tre skikt av noder och är således med vår terminologi tvålagrat. Om uppgiften är att indela en mängd inputmönster i N olika klasser, så skall nätverket ha N outputnoder och tillhörighet till klass nr i skall kodas som ”1 av N”, alltså genom att outputnod nr i har aktiviteten 1 medan övriga har aktiviteten 0. Antalet mellanliggande noder (i “Kohonenlagret”) väljs som en multipel M x N av N. De aktiveras kompetitivt på samma sätt som i ett SOM, dvs. minsta euklidiska avstånd mellan Kohonen-nodens viktvektor och inputvektorn avgör. Vikterna från inputnoder till Kohonen-noder är från början satta slumpvis.

Alla noderna i en viss grupp om M Kohonen-noder är fast kopplade till en och samma maximalt linjära outputnod medelst vikten 1. Varje nod i gruppen åstadkommer alltså (om den aktiveras) samma, förutbestämda output från nätverket. Denna output är ju samtidigt den önskade output för en viss delklass av inputmönstren (klass nr i om vi talar om grupp nr i av noder). *Träningen går därför ut på att åstadkomma, att vinnaren på inputs från denna delklass av mönster alltid hämtas från just denna grupp av noder*. Det betyder att viktvektorerna för noderna i gruppen måste justeras så att de blir kodvektorer för inputs från klassen ifråga. Vi ska strax beskriva hur detta går till.

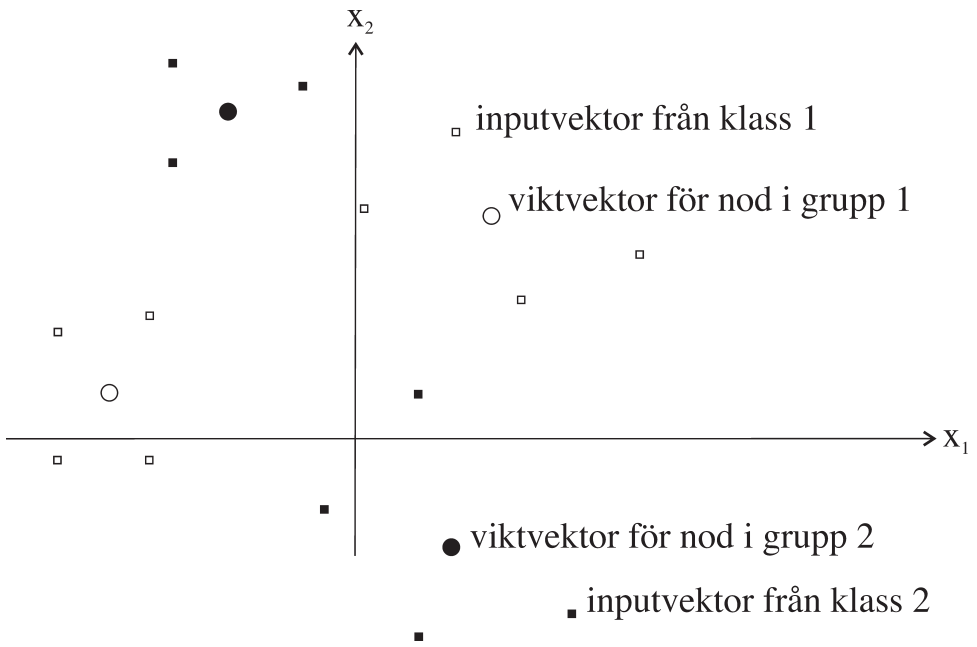
Figur 78 visar en simulering i NeuralWorks av ett LVQ-nätverk med 16 inputs, två outputklasser och 2 x 2 mellanliggande (Kohonen-)noder. Längst upp ser man instrumenten med vars hjälp det fel nätverket gör fortlöpande åskådliggörs.



Figur 78. Ett LVQ-nätverk. Skärmfoto från Neuralworks. Förklaring: se text.

LVQ för icke linjärt separerbara problem

För att förstå varför LVQ-nät *i princip* kan lösa icke linjärt separerbara klassifikationsuppgifter utgår vi för enkelhets skull från att input har två komponenter, att input skall separeras i två klasser och att LVQ-nätverket som i figur 78 har 2 x 2 Kohonennoder (dvs. M = 2). Inputvektorerna och viktvektorerna till Kohonen-noderna ligger nu i ett tvådimensionellt vektorrum. I figur 79, som åskådliggör förhållandena i det tränade nätverket, har ett antal inputvektorer markerats med ofyllda respektive fyllda kvadrater, beroende på vilken av de två önskade klasserna 1 respektive 2 de tillhör. Viktvektorerna för de fyra Kohonen-noderna (som motsvarar nod nr 17–20 i figur 78) har markerats på motsvarande sätt, men med ofyllda respektive fyllda cirklar. En ofylld cirkel betyder alltså en Kohonennod som genom sin grupptillhörighet är fast kopplad till den output, som vi vill att nätverket skall tilldela de inputs som representeras av ofyllda kvadrater. Figur 79 skisserar en möjlig placering av Kohonennodernas viktvektorer efter träningen. Denna placering ger en icke-linjär separation av de två klasserna.



Figur 79. LVQ-lösning av ett icke linjärt separerbart problem. Ofyllda och fyllda kvadrater: inputvektorer från klass 1 respektive 2. Ofyllda och fyllda cirklar: viktvektorer för noder i grupp 1 respektive 2.

De två ”ofyllda” Kohonennoderna är här, genom sina närhetsrelationer till inputdata, kodvektorer för *var sin delklass* av ”ofyllda” inputs. Dessa två inputdelklasser är separerade från varandra av en barriär av ”fyllda” inputs på ett sätt som påminner om XOR-problemet, men just genom att de två delklasserna tas om hand av var sin Kohonennod som tillhör *samma* grupp kan inputs i båda delklasserna ge samma output.

Det är lätt att inse att man med ett större antal Kohonennoder i varje grupp (större värde på M) *i princip* kan lösa också mycket mer komplexa problem.

Träning av ett LVQ-nät

Kan ett LVQ lösa klassifikationsproblem lika bra som en flerlagrad perceptron? I princip, ja, även om de inferensteoretiska resonemangen blir annorlunda (en fråga som vi inte kommer att penetrera).

Kan nätet tränas att hitta de lösningar som finns? Från början har ju de olika Kohonennodernas viktvektorer inte “rätt” relationer till de input-

vektorer som de ska vara kodvektorer för, dvs. de ligger inte där de ska i viktrummet. Kan de hitta rätt genom en träningsalgoritm? Svaret är också här jakande, och algoritmens viktigaste del ser ut som följer.

Man ger en input, och algoritmen utser (precis som i SOM) en vinnare bland noderna i Kohonen-lagret, nämligen den nod vars viktvektor ligger närmast inputvektorn. Sedan kommer övervakningen in: *Om och endast om den vinnande noden hör till den grupp av noder som ger den önskade output för den input som givits,"dras" nodens viktvektor närmare input-vektorn, i annat fall avlägsnas den från inputvektorn.* Anta till exempel för nätverket i figur 78, att den önskade output för en viss input är (1, 0), dvs. vi vill ha aktivitet i det *vänstra* outputneuronet (nod 21). I så fall ska vinnarnoden vara en av de två vänstra Kohonennoderna (17 eller 18) för att dess viktvektor ska dras närmare inputvektorn; annars dras den bort från input.

LVQ innehåller också ett antal andra mekanismer som underlättar en korrekt lösning, men dem ska vi inte beskriva här.

Obsevera att man inte tränar förbindelsen mellan Kohonennoderna och outputnoderna; de hålls fixa. Istället ändrar algoritmen på vilka inputs som aktiverar vilken Kohonennod. En vinnare som hör till "fel" grupp får sin viktvektor avlägsnad från inputvektorn, och är därmed mindre trolig vinnare nästa gång. Det blir därför gradvis allt mer sannolikt att en input ger en vinnare i den grupp av Kohonennoder som aktiverar "rätt" output. Om nätverket från början är tillräckligt komplext kommer så småningom varje input att aktivera en Kohonennod som ger rätt output.

Komplexitet och generalisering – igen

I princip kan LVQ lösa alla konsistenta klassifikationsproblem för en given träningsdatamängd. För att inse detta behöver man bara sätta M = antalet inputs i datamängden, varvid man förstår att varje input kan "tas om hand" av en egen kodvektor. Principen att alltför komplexa nätverk generaliserar dåligt gäller dock även LVQ-nätverk, och ett nätverk med M = antalet inputs kommer inte att kunna generalisera alls. Men just nu gällde det ju bara att visa på den principiella lösbarheten av en uppgift, inte att lösningen är bra för framtida bruk... För att få ett nätverk som både klassificerar utgångsdata bra och generaliserar bra gäller det, precis som för en MLP, att välja ett nätverk av lagom styrka. Samma grundläggande principer för att välja en "lagom" modellstyrka gäller för LVQ

som för MLP, även om redskapen för att genomföra valet kan skilja sig åt. Färre Kohonen-noder och mindre träning betyder således inte bara sämre diskriminationsförmåga i träningsdatamängden, utan också, som regel, bättre generaliseringsförmåga.

LVQ-nätverken är i praktiken ofta ett mycket bra alternativ till MLP-BP, särskilt för klassifikationsuppgifter som likt den i figur 79 innehåller separata "öar" av inputdata som ändå skall hänföras till samma klass.

9.6 Radialbasnätverk och supportvektormaskiner

En ny typ av basfunktioner

En perceptron med ett visst antal noder i en viss arkitektur och med bestämda vikter definierar ju, som vi redan uppmärksamnmat i avsnitt 4.5, en funktion från inputvektorn **x** till outputvektorn **y**,

(9.6.1) $\mathbf{y} = F(\mathbf{x})$

Denna funktion är i sin tur sammansatt av ett stort antal delfunktioner som härstammar från de enskilda elementens bearbetning av informationen och det sätt på vilket deras arbete summeras. För en typisk tvålagrad perceptron med n dolda noder och maximalt linjär output kan man skriva sambandet som

(9.6.2) $\mathbf{y} = g(h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_n(\mathbf{x}))$

där g är en viktad summation och de olika h_i, *basfunktionerna*, är sigmoida funktioner bestämda av vikterna till det dolda lagret. Med detta betraktelsesätt kan man se de neurala nätverkens anpassning till sin uppgift som en justering av de ingående funktionernas parametrar.

Radialbasnätverken arbetar med helt andra basfunktioner än perceptronerna, nämligen *radialbasfunktioner* (RBF). De påminner på många sätt om de metoder som arbetar med typvektorer, exempelvis SOM och LVQ. Ett RBF-nät är således tvålagrat. Outputlagret är linjärt och får input dels (eventuellt) direkt från inputlagret, dels från de dolda noderna (RBF-noderna). Hur en input aktiverar en dold nod bestäms av nodens "närhet" till input. Närhetsmåttet bestäms som någon monoton funktion av den

geometriska (euklidiska) närheten mellan nodens viktvektor och ifrågavarande input; vanligt är att man tittar på *vilken sannolikhet input ifråga skulle få enligt en Gaussisk fördelning runt viktvektorn*. Varje RBF-nod kommer att vara en ”kodvektor” för en viss delklass av inputmängden i den svagare meningen, att noden har det kortaste ”avståndet” av alla till dessa inputs och därför är den nod som aktiveras *mest* av dem. Däremot finns här inget som liknar ”winner takes all”, utan varje input kommer att ge upphov till aktivitet i *alla* noderna i RBF-lagret.

Träningen av ett RBF-nät är mycket mindre tidsödande än träningen av en MLP. Placeringen av RBF-noderna och deras övriga parametrar (exempelvis de gaussiska fördelningarnas spridning) bestäms först så att de på bästa sätt representerar inputdatamängden. Det finns flera olika algoritmer för att göra detta. Det är liksom när det gäller LVQ en poäng, att separerade grupper av inputs kan få olika RBF-noder som kodvektorer. De skall alltså inte placeras ut för glest, men inte heller för tätt, för då kan generaliseringsförmågan bli lidande. Vikterna till de linjära outputneuronen bestäms sedan på samma sätt som för ett helt linjärt system; det vill säga man kan använda deltaregeln eller, vilket förstås går ännu snabbare, en direkt beräkning med linjär algebra (jämför avsnitt 6.2).

Anledningen till att ett RBF-nätverk kan använda denna enkla träningsmetod är i grund och botten, att övergången till de nya basfunktionerna innebär ett slags ”gles kodning” av inputdata. Det finns många fler RBF-noder än inputdimensioner, och informationen sprids alltså ut i ett rum av mycket högre dimensioner än inputrummet. Detta gör att det oftast går att hitta bra linjära beslutsgränser mellan de önskade kategorierna. Detsamma gäller, men i ännu högre grad, om supportvektormaskinerna (se nedan).

RBF-nät har i princip samma styrka och svagheter som MLP-BP och LVQ vad gäller träning och överträning. De är, likt LVQ, bäst på uppgifter där samma klass innehåller separata ”öar” av inputs. För andra uppgifter skall de normalt sett inte vara förstahandsvalet.

Det är inte otänkbart att det finns biologiska nätverk vars grundläggande funktionssätt liknar ett RBF-nät. En excitatorisk, topologibevarande projektion av en receptoryta på ett annat skikt av neuron måste inte nödvändigtvis fungera enligt den modell som vi beskrev i avsnitt 8.1. Istället kanske sannolikheten för att ett sensoriskt neuron ska ha en excitatorisk förbindelse till ett neuron i skiktet ovanför avtar med avståndet i ”sidled”

till detta neuron, exempelvis enligt en gaussisk fördelning. Detta kunde i sin tur ha att göra med förhållanden under nätverkets tillblivelse. En sådan projektion skulle kunna fungera analogt med projektionen av inputdata på det dolda skiktet i ett RBF-nät.

Supportvektormaskiner

En ny utvecklingslinje som bara skall nämnas kort är SVM, Support Vector Machines.[159] SVM innebär, mycket enkelt uttryckt, en speciell träningsprocedur för RBF-nät och liknande ”kernel-baserade” metoder. En grundtanke är att noderna placeras ut nära gränsen mellan de områden som ska skiljas åt, snarare än centralt i öar av inputdata. En SVM kan visas ta hand om generaliseringsproblemen på ett ovanligt bra sätt därigenom att den mer eller mindre automatiskt anpassar sin storlek till komplexiteten hos inputmängden. Om man ska klassificera SVM som en ANN-metod är däremot mer tveksamt, eftersom likheten med kända neurala mekanismer är liten.

Därmed lämnar vi feed-forwardnätverken, och ska ägna det – tyvärr alltför kortfattade – sista kapitlet åt neurala nätverk med återkoppling.

[159] Schölkopf et al. (1999).

10. Representation av tid i neurala nätverk

10.1 Inledning

Vi har redan stiftat bekantskap med två nätverk vars arkitektur har ett mer eller mindre dominerande inslag av återkoppling, nämligen Hopfieldnätet (avsnitt 7.2) och ART (avsnitt 8.4). Trots olikheterna i uppbyggnad har dessa nätverk det gemensamt, att de rekurrenta mekanismerna används för att nätet ska hitta en stabil lösning av ett klassifikations- eller kategoriseringsproblem. Output är således, liksom input, en *statisk* storhet. I Hopfieldnätet är output ett stabilt mönster i alla enheterna som mer eller mindre liknar det inputmönster som lagts på samma enheter, i ART är output ett stabilt mönster i F1(b) som mer eller mindre liknar inputmönstret i F1(a).

Vi ska nu istället titta på några uppgifter och några lösningar där de återkopplade nätverkens dynamik används också på input- och/eller outputsidan. Dessa uppgifter har alla att göra med analys av tidsförlopp eller andra sekvenser. I resten av detta avsnitt ska vi definiera en av dessa uppgifter, nämligen *prediktion*, närmare. I avsnitt 10.2 ska vi se på ett sätt att lösa prediktionsuppgifter med en typ av ANN som *inte* involverar en rekurrent arkitektur, medan vi i 10.3 kommer att kort bekanta oss med några kända återkopplade nätverk som representerar tid på ett helt annat sätt. I det allra sista avsnittet av boken (10.4) spekulerar vi över hur nervsystemet representerar tid i två specifika sammanhang.

Rubriken för detta kapitel nämner ”kontroll”. Som nämnts ett par gånger tidigare innebär inlärning av kontroll i allmänhet att man måste beakta skeenden över tid. Det är ju ofta fråga om en fördröjd felsignal (eller förstärkning, om man talar i termer av operant inlärning). Detta innebär att de neurala nätverk som vi introducerar i detta kapitel också är lämpade för inlärning av kontroll. Denna sida av dem kommer inte att närmare behandlas här, utan tonvikten kommer att ligga på prediktion.

Kontrollproblematiken tangeras dock genom den modell av motorisk kontroll som läggs fram på bokens allra sista sidor.

Prediktionsproblemet

Betrakta problemet att från en bokstav i en text, om vilken vi inte vet mer än att den är på svenska, förutsäga nästa bokstav. Man inser strax att dessa förutsägelser oftast blir mycket osäkra. Visserligen är det någorlunda sannolikt, om bokstaven är ett (gement) ”x”, att nästa bokstav är ”y” (som i ”xylofon”), men säkert är det inte (det kan stå ”xenofobi”), och ännu mycket osäkrare blir det om bokstaven ifråga är ”t”. Ett otroligt stort antal ord innehåller ”t”, så fortsättningen är mycket obestämt. Men antag att vi också skulle få veta att den närmast föregående bokstaven är ”a”, den näst närmast föregående ett ”r”, dessförinnan ”t” igen och före det ett mellanslag. Vi har alltså tillgång till bokstavssekvensen ” trat” (observera mellanslaget), och på svenska är den enda möjliga fortsättningen på ett ord som börjar så ytterligare ett ”t” (som i ”tratt” och ”trattkantarell”). Vår möjlighet att predicera nästa element i bokstavssekvensen har alltså förstärkts avsevärt genom att vi fått tillgång till information om fler (konsekutiva) moment i sekvensen än ett enda.

Ett helt analogt exempel, men nu från tidsdomänen, är förutsägelse av aktiekurser. Ingen aktiespekulant skulle, när hon försöker förutsäga morgondagens kurs på Volvoaktien, nöja sig med att använda dagskursen som utgångsvärde. Istället tittar hon på s.k. ”trender” i utvecklingen, dvs. hon tar hänsyn till ett antal kursnoteringar före dagens värde. Därigenom gör hon säkrare (om än inte särskilt säkra) förutsägelser. Ännu något säkrare blir de förstås om hon tittar inte bara på Volvoaktien, utan också ser dess värdeutveckling i relation till den hos andra börspapper – detta sagt bara för att man inte ska glömma, att en analys som tar sekvenser som indata givetvis kan arbeta med flerdimensionella sådana sekvenser. Även en sekventiell output kan på detta sätt vara flerdimensionell.

Markovförlopp

Låt oss nu påminna om det viktiga begreppet *Markovförlopp* (jämför också avsnitt 1.4). Ett tidsförlopp, eller en annan ordnad sekvens, sägs vara ett Markovförlopp av ordningen 1, om alla sannolikheter som kan tillskrivas ”kommande” händelser utifrån kunskap om det ”nuvarande” elementet blir oförändrade om man tar in information från fler element

”bakåt” i sekvensen. Varken svenska språket (betraktat som en bokstavssekvens) eller Volvoaktiens utveckling över tid är alltså Markovprocesser av ordningen 1, eftersom förutsägelseförmågan ökar om man tar fler element i sekvenserna än det senaste som utgångspunkt. Ofta skriver man helt enkelt ”Markovprocess” när man menar en Markovprocess av första ordningen, och vi kommer ibland att göra så nedan. Markovförlopp av ordningen 2 är de, där prediktionerna inte förändras om man tar hänsyn till mer än *två* konsekutiva element, etcetera. Vi har här inte skilt mellan diskreta och kontinuerliga Markovprocesser, så för ordningens skull skall det påpekas att vi tills vidare bara talar om de diskreta fallen (också kallade ”Markovkedjor”).

Ett exempel på något som *är* en Markovprocess av första ordningen ges av följande fabel. Antag att en räv förföljer en skuttande hare mot norr. I varje skutt hoppar haren 2 meter antingen norrut, åt väster eller åt öster, men den väljer hoppriktning helt slumpvis. Räven, som har studerat harens beteende ett bra tag, kan med kännedom om harens nuvarande position förutsäga bl.a. att den efter nästa hopp med sannolikheten 1/3 kommer att befinna sig ytterligare 2 meter norrut. Det intressanta är att rävens förutsägelseförmåga inte förbättras om han också tar hänsyn till var haren var före det sista hoppet.

Det ska också nämnas att alla deterministiska förlopp är Markov-processer av ordningen 1. En deterministiskt system är nämligen ett system, vars framtida tillstånd i princip kan förutsägas med 100% säkerhet från det *nuvarande* tillståndet. Och om ett system verkligen är sådant, så kan information om tidigare tillstånd givetvis inte förbättra förutsägelserna. En helt annan historia är att vi, om vi inte har någon säker kunskap om det nuvarande tillståndet, istället kan ta ledning av de tidigare (jämför också nästa stycke).

Slumpmässighet och slumpmodeller

Innan vi går vidare och studerar hur neurala nätverk kan användas för att analysera tidsförlopp och andra sekvenser, kanske det också kan vara bra att fundera över en annan fråga som har att göra med determinism och indeterminism. När det gäller icke-tidsliga sekvenser är det inte så svårt att se relevansen av det allmänna begreppet Markovprocess, men är begreppet verkligen så användbart på tidsförlopp? Finns det så många slumpmässiga tidsförlopp i naturen att en sådan beskrivningsapparat är motiverad?

Här måste man noga skilja mellan våra iakttagelser och den underliggande verkligheten. Även när vi har att göra med ett i grunden deterministiskt naturfenomen är det långtifrån säkert att vi har observerat, eller ens *kan* observera, just de variabler från vilka framtiden i princip skulle kunna förutsägas. Och även om dessa variabler kan observeras är det inte troligt att vi kommer undan mätfel. Vi har därför i många fall nytta av att analysera de givna skeendena *som om* de utgjorde icke-deterministiska processer. Inte sällan är en analys i termer av en högre ordningens Markovprocess den bästa, även när vi har anledning tro att skeendet *egentligen* är deterministiskt. Jämför gärna ss. 40–41 ovan.

10.2 Flerlagrade perceptroner med tidsfönster

Analys av första ordningens stationära processer

Låt oss då äntligen komma till saken, alltså användandet av ANN för analys av sekvenser. Med hjälp av de begrepp vi infört kan vi först konstatera, att Markovkedjor av första ordningen lätt kan hanteras med hjälp av de redskap vi redan bekantat oss med. I dessa fall kan vi nämligen ta det enskilda elementet i sekvensen (tillståndet vid en enskild tidpunkt) som input till ett neuralt nätverk, och nästa element (tillståndet vid nästa tidpunkt) som output. Ett neuralt nätverk kan då, om det får de verkliga ”nästa” tillstånden i en träningsmängd som önskad output, lära sig att så gott data tillåter förutsäga nästa tillstånd från det föreliggande.

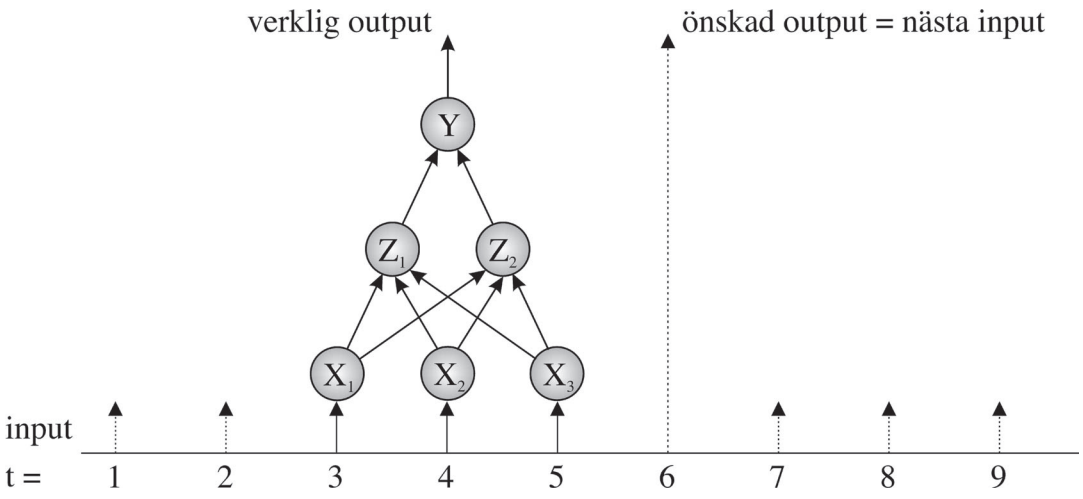
Träningen skall tillgå på vanligt sätt, dvs. data presenteras i slumpvis vald ordning ett stort antal gånger och en lämplig algoritm uppdaterar vikterna. Det finns absolut ingen anledning att mata in data i den ordning de kommer i sekvensen, eftersom all nödvändig information finns i det enskilda tillståndet. Metodiken förutsätter dock, att det funktionella sambandet mellan ett tillstånd och nästa kan antas vara konstant över tiden, dvs. vi har anledning tro att vi har att göra med en *stationär* process. Annars kan man naturligtvis inte använda träning på ”tidiga” data för att göra förutsägelser från ”sena” händelser, eller vice versa.

Tidsfönster för högre ordningens problem

Hur går det då i börsmäklarfallet? Låt oss göra en datafil där varje post har Volvoaktiens börsvärde de enskilda dagarna som input och nästa dags

värde som önskad output; sedan bygger vi en MLP och försöker lära den att förutsäga output från input. *Om* aktieutvecklingen vore en stationär Markovkedja av ordningen ett skulle detta vara den bästa strategin. Nu vet vi att det inte är så; samma kursvärde kan ju föreligga lika gärna vare sig aktien är på väg upp eller ner, och utan att känna till trenden över tid gör vi en dålig förutsägelse. Nätverket gör antagligen inte så väldigt mycket bättre ifrån sig om vi håller fast vid prediktion utifrån en enda tidpunkt, men lägger till ett antal andra aktiers kurser vid denna tidpunkt som komponenter i input.

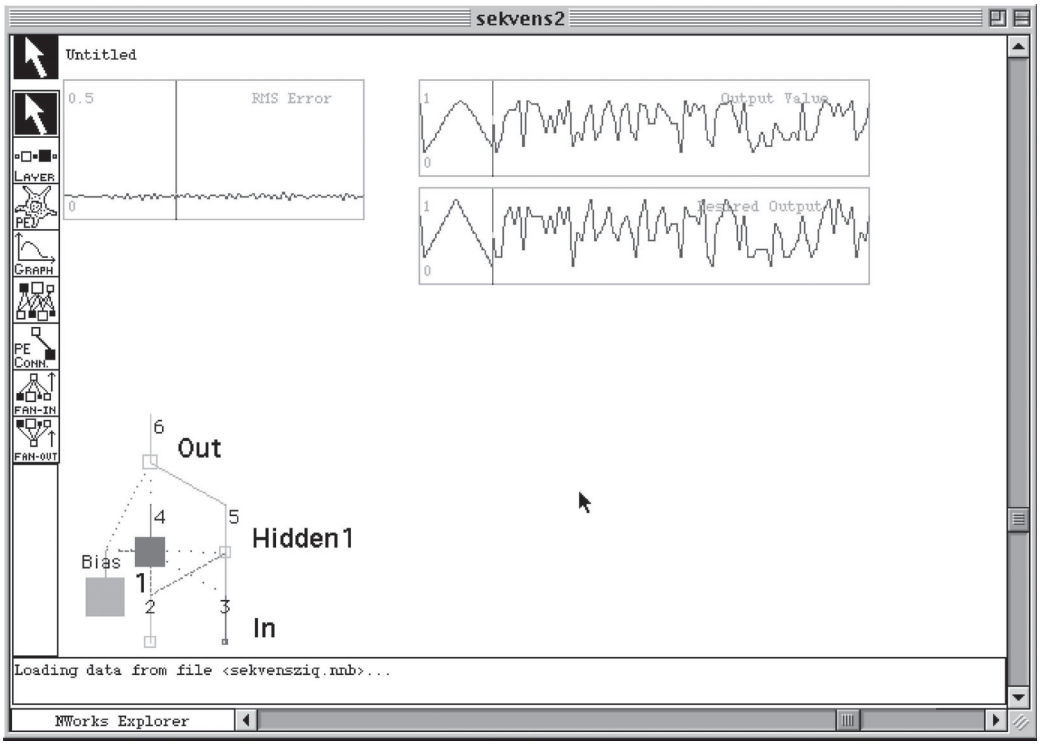
Exemplet leder dock på ett mycket naturligt sätt fram till en enkel och ofta nöjaktig lösning av prediktionsproblemet. Vi låter vårt neurala nätverk som input ta inte bara ett aktievärde vid en viss tidpunkt, eller en vektor av olika värden vid denna tidpunkt, utan *värdena (eller vektorerna) vid ett antal på varandra följande tidpunkter*. Man brukar uttrycka detta som att nätverket tittar på informationen inom ett helt ”fönster” av intilliggande händelser i sekvensen. Det betyder i fallet med en enda aktie att en input ges av dess kurs vid tre konsekutiva tidpunkter, medan den önskade output är värdet vid tidpunkten därefter. Se figur 80.



Figur 80. En flerlagrad perceptron med tidsfönster av längden 3. Förklaring: se text.

Återigen, inputvärdena och/eller outputvärdena vid varje tidpunkt kan själva utgöras av vektorer. Tidsfönstret definierar då en vektor över tiden av dessa vektorer. Om antalet komponenter i indata är n, antalet komponenter i utdata är m och tidsfönstrets storlek är t, måste nätverket alltså ha n * t inputelement och m outputnoder.

Tidsfönstret skall vid träningen av nätverket flyttas slumpmässigt över hela sekvensen (här: över tiden). Ett sådant MLP-nätverk med ”flyttbart tidsfönster”, tränat med någon av de vanliga inlärningsalgoritmerna, kan göra förvånansvärt bra ifrån sig på uppgiften att förutsäga förlopp som inte är av Markov-ordningen 1. Det behöver kanske inte påpekas att stationära förlopp av Markov-ordning 2 kan hanteras bra av nätverk med fönsterstorlek 2, etc. Figur 81 visar således resultatet av en enkel laboration i NeuralWorks med en MLP med tidsfönster 2, som just lärt sig att approximera den återkommande, upp- och nedstigande sekvensen ...0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.2 0.3.... Den vänstra delen av diagrammet längst upp till höger visar de senaste prestanda, med den önskade output nertill och den verkliga upptill.



Figur 81. En MLP med tidsfönster av längd 2, som lär sig ett Markovförlopp av ordning 2. Skärmfoto från NeuralWorks; förklaring: se text.

Ett stort problem är dock att man ofta inte känner till Markov-egenskaperna hos den studerade processen, och därför inte vet hur långt tidsfönster man behöver – samtidigt som ett större tidsfönster leder till större risk för att nätverket hittar skenbara korrelationer mellan in- och utdata. Det är alltså inte bara att ”brassa på” med ett långt tidsfönster.

Back propagation in time

Det finns ett visst sätt (”back propagation in time”) att träna MLP med tidsfönster, som ger dem ännu intressantare egenskaper – dock utan att lösa det just nämnda problemet. Det är användbart i de situationer när man inte alltid har tillgång till värdet i *nästa* steg efter tidsfönstret, utan först ett senare värde. Man kan då istället *simulera* nästa värde, dvs. använda nätverkets output som sista ledet i input till samma nätverk när det tittar på *nästa* tidsfönster. Processen kan upprepas i flera steg till dess att det finns ett verkligt värde (en verklig önskad output) att jämföra med. Back propagation-algoritmen (eller någon variant av den) tillämpas sedan på ett tänkt sammansatt nätverk, som består av det behövliga antalet strukturella kopior av det ursprungliga nätverket ”staplade ovanpå varandra” i sekvensens ordning, och med output och input arrangerade enligt ovan. Outputfelet i det sammansatta nätverket propageras hela vägen ”ner” genom delnätverken (vilket kan sägas motsvara *bakåt genom tiden* – därav algoritmens namn), via de simulerade outputvärdena och slutligen till det understa lagret av vikter. På detta sätt kan nätverket lära sig att förutse ett förlopp flera steg in i framtiden, och detta effektivare än om man tränade det ”rakt upp och ner” med sekvenser av indata som input och verkliga data några steg senare som önskad output.

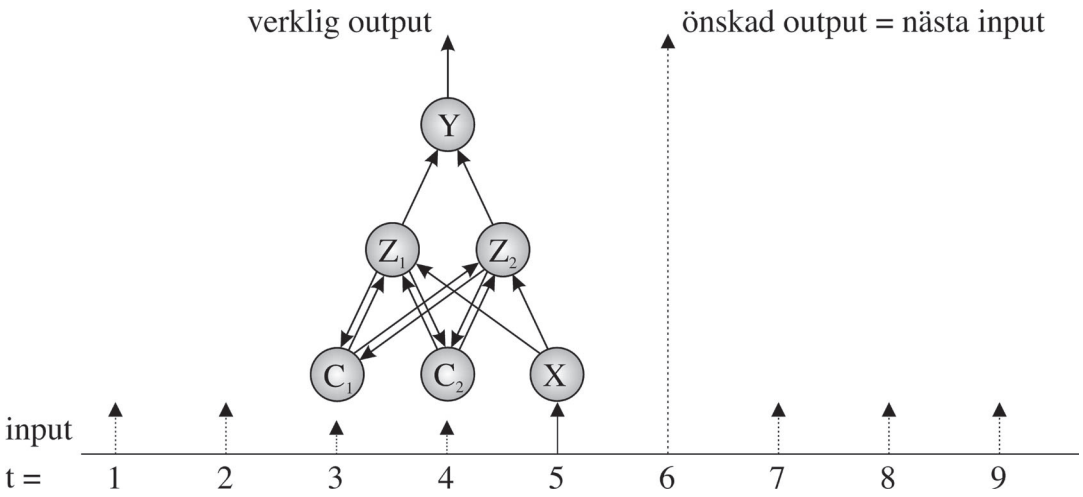
Flerlagrade perceptroner med tidsfönster som tränas med ”back propagation in time” beskrivs ibland som återkopplade, men det är vilseledande. Det är istället fråga om ett feed-forwardnät, uppbyggt av andra feed-forwardnät. Nätverket kan fortfarande inte ”se” längre än tidsfönstrets storlek, och om det finns fullständiga data från sekvensen ger algoritmen samma resultat som när man tränar på vanligt sätt. En helt annan sak är att man med fördel kan använda ”back propagation in time” för att träna verkligt återkopplade nätverk (se nästa avsnitt).

10.3 Egentliga återkopplade nätverk

Ett äkta rekurrent nätverk representerar inte sekvenser explicit. Input utgöres istället av data från ett enda steg i sekvensen, t.ex. en tidpunkt. Denna input matas in fortlöpande, och informationen om tidsföljd lagras implicit genom de rekurrenta förbindelser som finns i nätverket.

Elmans och Jordans nätverk

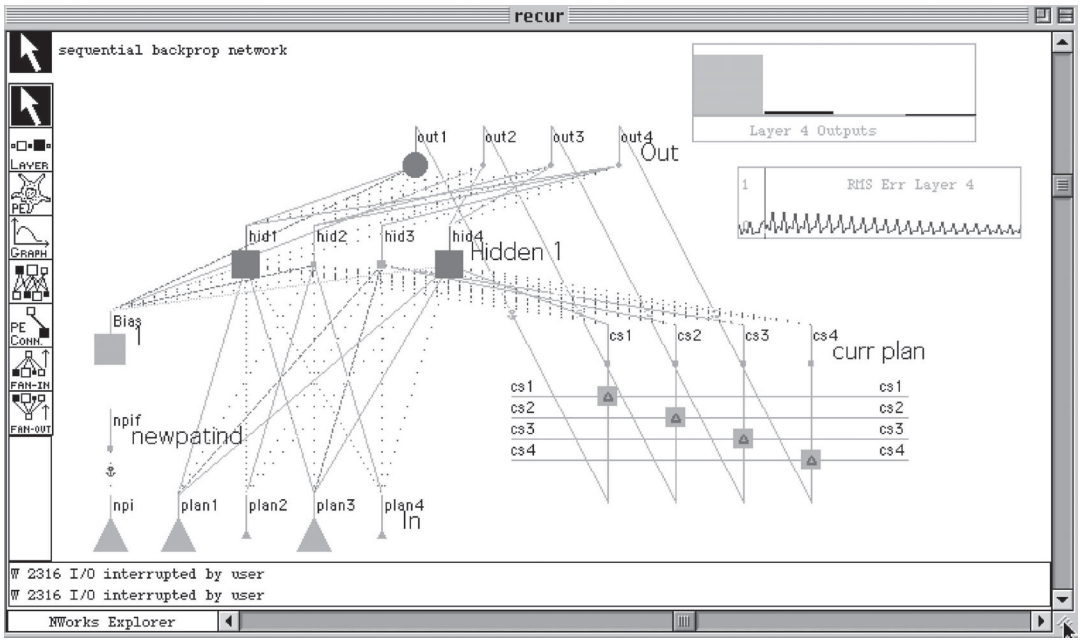
Bland de olika arkitekturer som kommit till användning bör man nämna Elman- och Jordannätverken. I Elmans version (också kallad SRN, Simple Recurrent Net) återförs informationen från de dolda noderna till motsvarande noder i ett särskilt skikt (kontextskiktet). Dessa noder har i figur 82 markerats med bokstaven C (för ”context”), och feedbackförbindelserna till dem med pilar (alla förbindelselinjer utan pilar är således feed-forward). Kontextnoderna skickar sedan tillbaka sina signaler till hela det dolda skiktet, pecis som om de varit inputnoder.



Figur 82. Principskiss av ett Elman-nät. C_i är kontextnoder. Förklaring i övrigt: se text.

Input till Elmannätet ges strikt sekventiellt, dvs. nätverket ”rör sig” åt höger över tidsaxeln med ett steg i taget. På det viset kommer kontextnoderna att vid varje tillfälle förmedla information om tidpunkterna före den aktuella input.

I Jordan-nätet är det istället outputnoderna som signalerar till kontextlagret, som i sin tur skickar information till de dolda noderna i nätet. Figur 83 visar ett Jordannät i Neuralworks; det har fyra inputnoder (”plan1” etc.) och lika många kontextnoder (”cs1” etc.), dolda noder och outputs. Observera att detta nätverk tar in en *samtidig* inputvektor med fyra komponenter; inputnoderna motsvarar alltså noden X i figur 82 och *inte* noderna X_1–X_3 i figur 80! På samma sätt motsvarar de fyra outputnoderna i figur 83 noden Y i figur 82. Dessa noder kodar nätverkets prediktion om nästa steg i sekvensen.



Figur 83. Ett Jordan-nät med fyra inputkomponenter, som det ser ut i NeuralWorks. Förklaring: se text.

Elman- och Jordannätverken kan i princip lagra inputinformation *godtyckligt* länge. I icke-kognitivistiska termer betyder det sagda, att deras tillstånd kan påverkas av vad som hänt i deras input godtyckligt långt tillbaka. De kan därför analysera funktionella samband som bygger på Markovegenskaper av godtyckligt hög ordning. Visserligen blir inflytandet från en viss input mindre och mindre ju längre tiden går, men vore det inte för att systemet hade en begränsad diskriminationsförmåga skulle något spår av varje input kunna leva kvar hur länge som helst.

Detta betyder, enkelt uttryckt, att ett Jordan- eller Elmannät inte behöver veta i förväg hur långa tidsfönster det behöver ta hänsyn till. Elman visade således i ett berömt försök, att ett litet nätverk med den av honom föreslagna arkitekturen kunde modellera strukturen hos en så kallad ändlig-tillstånds-grammatik i det närmaste perfekt.[160] En sådan grammatik innebär ett system av beroenden mellan tecken, som kan vara av godtyckligt hög Markov-ordning.

Jordan- och Elman-nät kan tränas med vanlig back propagation och dess varianter, med backpropagation in time, och med ytterligare andra metoder (t.ex. genetiska algoritmer, se nedan). I praktiken är Elman- och Jordan-nät mycket känsliga för störningar, och det kan vara en kinkig

[160] Elman (1990).

uppgift att träna dem. Det hindrar inte att de har en mycket stor principiell betydelse och att förståelsen av dem torde vara central för vår kunskap om nervsystemets sätt att bearbeta information. Lagring i egentliga återkopplade nätverk är sannolikt en av nervsystemets viktigaste principer för att representera tid och tidsliga förlopp.[161]

Genetiska algoritmer för träning av rekurrenta nätverk

För prediktionsuppgifter, och inte minst för inlärning av kontroll, vill man ofta använda mer komplicerade arkitekturer än Elman- och Jordannäten. Det kan för sådana arkitekturer vara oklart vilken variant av back propagation som är bäst att tillämpa, och hur den ska tillämpas. I det läget kan det vara bra att veta att det alltid finns en alternativ och mycket kraftfull typ av inlärningsmekanism, nämligen de *genetiska algoritmerna*. Dessa metoder, som också benämns *evolutionära algoritmer,* kan sägas vara analoga med naturligt urval snarare än med inlärning på individnivå. En hel population av neurala nätverk skapas, vart och ett med parametrar som skiljer sig något från de andras. Vanligtvis håller man arkitekturen konstant och varierar endast vikterna. Dessa representeras matematiskt som delsträngar i en lång sträng av tal, ofta binära men andra representationer är också vanliga. Alla nätverk i populationen prövas på det problem som ska lösas. De nätverk som lyckas sämst ”kasseras”, medan de andra får chansen att ”föröka sig” och till och med få gemensam avkomma genom att parameteruppsättningarna ”rekombineras”. Genom ”mutationer”, dvs. små slumpvisa parameterförändringar, garanterar man att det i långa loppet sker en positiv utveckling (och inte bara ett negativt urval).

Det finns många varianter av genetiska algoritmer för ANN, och att hantera dem rätt är en konst. Fördelen med angreppssättet är att varje problem som har en teoretiskt möjlig lösning med ett ANN av viss arkitektur också kan lösas i praktiken, om man väljer den genetiska algoritmen med omsorg. Nackdelarna är större tidsåtgång och högre krav på datorns kapacitet än för de algoritmer vi talat om tidigare. I de flesta sammanhang tar man därför till genetiska algoritmer endast då andra metoder misslyckats. Just när det gäller inlärning av kontroll har dock de genetiska algoritmerna fått en ordinarie plats i laget. Här arbetar man, som antytts ovan, gärna med komplexa rekurrenta arkitekturer som inte är särskilt lätta att träna med andra metoder. Genetiska algoritmer har då

[161] Jfr. Cleeremans (1993), French (utg.) (2002) samt Hertz (2006).

givit slående resultat, inte minst vid konstruktion av styrsystem för robotar. Av uppenbara skäl säger oss dessa resultat dock inte särskilt mycket om hur det går till när en *individ* lär sig att utföra en förelagd uppgift – såvida inte den genetiska algoritmen selekterat fram en *inlärningsförmåga* hos roboten. Experiment med den sistnämnda möjligheten i fokus är inte vanliga men utgör ett spännande forskningsområde.

10.4 Tidsfönster i perception och motorisk kontroll?

Vi ska avsluta denna framställning med en diskussion om nervsystemets sätt att representera tid i två specifika sammanhang.

Har sådana tidsfönster som vi diskuterade ovan (avsnitt 10.2) biologisk relevans i den meningen, att något biologiskt nätverk kan antas ta en hel tidslig sekvens som samtidig input? Det är väl mycket naturligare att tänka sig, att nervsystemet tar emot tidslig information ”ett steg i taget”? Ja, men även i det senare fallet kan ett tidsfönster konstrueras, nämligen om nervsystemet har tillgång till ett lämpligt sätt att korttidslagra den inkommande signalen i obehandlad form. Här ska vi särskilt uppmärksamma möjligheterna att åstadkomma detta antingen genom ett *skiftregister* eller genom förbindelser med *differentiell fördröjning.*[162]

Sensoriska buffertar och skiftregister

För att förklara den första av dessa möjligheter ska vi först återknyta till ett fenomen som diskuterades i kapitlet om korttidsminne, nämligen de ”sensoriska buffertarna” (avsnitt 3.3). Vi tycker oss t.ex. kunna *höra,* inte bara minnas, en ton en kort stund (upp till cirka två sekunder) efter det att den fysiskt upphört. En standardförklaring av fenomenet utgår från begreppet *skiftregister.* Man antar att det finns en kedja av neurala enheter (inte nödvändigtvis enstaka neuron) som har en ”enkelriktad” koppling till varann. Enheten i ena änden i kedjan antas i varje ögonblick ta emot input utifrån, samtidigt som varje enhet signalerar sitt tillstånd till nästa enhet i kedjan. I slutet av kedjan försvinner signalen. Ett skiftregister kan uppenbarligen ligga till grund för ett högre ordningens nätverk, som i varje ögonblick tar sekvensen av de senaste enskilda inputs som sin egen input – på samma sätt som ett MLP med tidsfönster gör.

[162] För utförligare diskussioner av ämnet för detta avsnitt se Malmgren (2004) och Herz (2006).

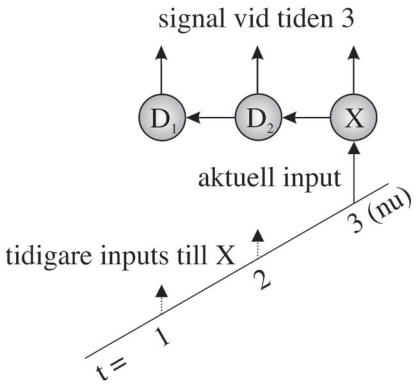
Skiftregister används mycket i kalkylatorer och datorer, och det är en rimlig hypotes att de bildar underlag för de sensoriska buffertarna. Man måste då också anta, att tillståndet i hela registret avspeglas samtidigt i medvetandet, och att tillståndet i enheter *nära inputenheten* upplevs som något som inträffat *senare* än de andra händelserna i bufferten.

Fördröjd input genom olika snabba förbindelser

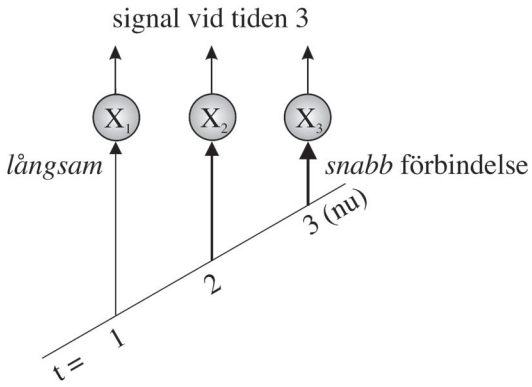
Emellertid finns det en annan, minst lika rimlig förklaring till buffertfenomenen, nämligen i termer av *differentiellt fördröjt input*. Antag att alla de neurala enheterna i den kedja som vi nyss antog tar emot den enskilda inputsignalen *direkt*, men förmedlad via flera inkommande förbindelser som ger *olika överföringstider* från sinnesorganet till de olika enheterna. Med lite eftertanke förstår man att denna mekanism kommer att resultera i exakt samma representation av input som skiftregistret. Dock behövs det nu ingen signalöverföring *mellan* enheterna, vilket betyder att ordningen mellan enheterna nu enbart definieras av hur snabba de förbindelser är som de mottar.

Figur 84 illustrerar de båda hypoteser som vi just talat om. Vi befinner oss vid tidpunkt 3, och neuronet längst till höger mottar i båda modellerna en aktuell inputsignal.

Skiftregister:



Fördröjd input:



Figur 84. Sensorisk buffring genom skiftregister resp. fördröjd input. Den lutande linjen representerar yttre tid. De tre neurala enheterna antas representera tre konsekutiva inputs som en samtidig signal.

Finns det någon anledning att tro att nervsystemet använder sig av fördröjd input snarare än ett skiftregister? Ett argument för att det skulle kunna vara så är, att mekanismen med fördröjd input är mer robust. Om input distribueras via flera oberoende förbindelser till inbördes oberoende enheter, så är det ingen katastrof om någon förbindelse eller enhet slås ut.

Antag exempelvis att den enhet som representerar att en ljudsignal ankom för en halv sekund sedan slås ut. Då borde vår upplevelse av den tidsliga gestalten få en konstant liten lucka på denna plats, men alla ljud kommer att höras vid nästan lika många tidpunkter som tidigare (på både kortare och längre tidsavstånd än ”buffertluckan”), och all sekventiell information kommer fortfarande att finnas representerad genom de andra enheterna. Detta är ett exempel på den biologiska betydelsen av *redundans* (jämför avsnitt 1.3).

Om däremot motsvarande neuron i ett skiftregister slås ut, så kommer ingen information att passera till enheterna längre bort i kedjan. Det vill säga, inga hörselupplevelser av tidsliga gestalter som är längre än en halv sekund blir möjliga, och de sekvenser som ett eventuellt högre ordningens nätverk kan utgå från blir mycket korta.

Fördröjd output i feed-forwardkontroll av rörelser

En relaterad hypotes är att nervsystemet använder sig av differentiellt fördröjd *output* i vissa former av motorisk kontroll. Vad vi ska diskutera är möjligheten av en extra snabb feed-forwardkontroll som är oberoende av alla stimuli i en sekvens utom den initiala. Vad menas med detta? Jo, låt oss först se på en vardaglig analogi. Antag att vi till exempel ska styra en bil från en gårdsplan in i ett välbekant garage. Förhållandena fordrar att vi först svänger vänster för att komma in genom dörren, och sedan omedelbart höger för att inte köra på en arbetsbänk. Om vi inte hade varit där förut hade vi måst vänta tills vi såg bänken innan vi initierade högersvängen, och det kanske hade fordrat att vi behövde bromsa kraftigt. Men eftersom förhållandena är välbekanta kan vi börja högersvängen ”på automatik” så snart vi rätat upp bilen efter den första svängen, vi behöver inte bromsa, och kommer därför snabbare på plats.

Det finns en uppenbar möjlighet att nervsystemet använder sig av fördröjda outputkretsar för att åstadkomma en liknande automatik i sådana beteendesekvenser som behöver vara mycket snabba. Antag ett arrangemang liknande det i högra delen av figur 84, men där förbindelserna

går åt andra hållet och slutar i ett effektororgan (en muskel). Nu kan ett enda sammansatt kommando skickas på en gång, som i förväg bestämmer en hel sekvens av rörelser. Om det är så, skulle man kunna tala om *motoriska buffertar* i analogi med de sensoriska. Liksom alla feed-forwardmeka-nismer borde dessa buffertar vara träningsbara, och till skillnad från vad som är fallet vid träning av enkla feed-forwardresponser kunde träningen inkludera prediktion av stimulussekvenser.

Är det inte, kära läsare, just så som vi ibland *upplever* att vi styr våra handlingar – dvs. som att hela rörelsen är bestämd i förväg och utförs utan att vi tar in någon sinnesinformation utöver den som behövs för att initiera rörelsen vid rätt tidpunkt? Och varför skulle inte vår upplevelse vara en god ledtråd till den neurala organisationen här, när den antas vara det i fråga om de sensoriska buffertarna?

Med denna lilla filosofiska fundering må vår framställning få ett slut, som antyder att det finns andra, lika viktiga gåtor att lösa i anknytning till neurala nätverk som de matematiska och de neurofysiologiska problemen.

Litteraturförteckning

Adams, F. (2003). The informational turn in philosophy. *Minds and Machines,* 13, ss. 471–501.

Almér, A. (2007). *Naturalising Intentionality.* Diss., Göteborgs Universitet. Göteborg: Univ.

American Psychiatric Association (1980). *Diagnostic and Statistical Manual of Mental Disorders*. 3:e uppl. (DSM-III). Arlington, VA: American Psychiatric Publishing.

American Psychiatric Association (1987). *Diagnostic and Statistical Manual of Mental Disorders*. 3:e reviderade uppl. (DSM-III-R) (Revised). Arlington, VA: American Psychiatric Publishing.

American Psychiatric Association (2000). *Diagnostic and Statistical Manual of Mental Disorders*. 4:e reviderade upplagan. (DSM-IV-TR) (Text Revision). Arlington, VA: American Psychiatric Publishing.

Ashby, W.R. (1952). *Design for a Brain.* London: Chapman & Hall.

Ashby, W.R. (1956). *An introduction to cybernetics.* London: Chapman & Hall.

Baddeley, A. (1987). *Working Memory.* Oxford: Oxford University Press.

Baddeley, A. (2007). *Working Memory, Thought and Action.* Oxford: Oxford University Press.

Berger, T.W. & Glanzman, D.L. (2005). *Implantable Biomimetic Electronics as Neural Prostheses.* Cambridge, MA: MIT Press.

Berzhanskaya, J., Grossberg, S. & Mingolla, E. (2007). Laminar cortical dynamics of visual form and motion interactions during coherent object motion perception. *Spatial Vision*, 20, ss. 337-95

Berkeley, G. (1901). An Essay towards a New Theory of Vision. I Fraser, A.C. (red.). *The works of George Berkeley*, Vol. I. Oxford: Clarendon. Ss. 93–210. [Ursprungligt publikationsdatum 1709].

Bischkopf, J., Busse, A. & Angermeyer, M.C. (2002). Mild cognitive impairment – a review of prevalence, incidence and outcome according to current approaches. *Acta Psychiatrica Scandinavica* 106, ss. 403–14.

Bishop, C.M. (1996). *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* New York: Springer Science+Business Media, LCC.

Bliss, T.V.P. & Lømo, T. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232, ss. 331–56.

Bower, G.H. & Hilgard, E.R. (1981). *Theories of Learning.* 5:e upplagan. Englewood Cliffs, NJ: Prentice-Hall.

Bower, J.M. & Bolouri, H. (2001). *Computational Modeling of Genetic and Biochemical Networks.* Cambridge, MA: MIT Press.

Carlsson, S.G. & Gale, E.N. (1976). Biofeedback treatment for muscle pain associated with the temporomandibular joint. *Journal of Behavior Therapy and Experimental Psychiatry,* 7, ss. 383–385.

Cherkassky, V., Krasnopolsky, V., Solomatine, D.P. & Valdes, J. (2006). Computational intelligence in earth sciences and environmental applications: Issues and challenges. *Neural Networks,* 19 (2006 Special Issue), ss. 113–21.

Churchland, P. (Patricia) (1986). *Neurophilosophy.* Cambridge, MA: MIT Press.

Clapin, H., Staines, P. & Slezak, P. (utg.) (2004). *Representation in Mind: New Approaches to Mental Representation.* Oxford: Elsevier.

Cleereman, A. (1993). *Mechanisms of Implicit Learning. Connectionist Models of Sequence Processing.* Cambridge, MA: MIT Press.

Crane, T. (2004). *Medvetandets mekanik.* Stockholm: Thales. (Engelska originalet 2003: *The Mechanical Mind.* 2:a uppl. London: Routledge.)

Cummin, B.C. & DeAngelis, G.C. (2001). The physiology of stereopsis. *Annual Review of Neuroscience,* 24, ss. 203–38.

Dayan, P. & Abbott, L.F. (2001). *Theoretical Neuroscience. Computational and Mathematical Modeling of Neural Systems.* Cambridge, MA: MIT Press.

Dragoi, V. (2002). A feedforward model of suppressive and facilitatory habituation effects. *Biological Cybernetics,* 86, ss. 419–26.

Doya, K., Ishii, S., Pouget, A. & Rao, R.P.N. (2007). *Bayesian Brain. Probabilistic Approaches to Neural Coding.* Cambridge, MA: MIT Press.

Dretske, F. (1981). *Knowledge and the Flow of Information.* Cambridge, MA: MIT Press.

Dreyfuss, H. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason.* Cambridge, MA: MIT Press.

Dybowski, R. (2000). Neural computation in medicine: perspectives and prospects. I Malmgren, H., Borga, M. & Niklasson, L. (red.). *Artificial Neural Networks in Medicine and Biology.* London: Springer-Verlag. Ss. 26–36.

Dybowski, R. & Gant, V. (red.) (2001). *Clinical applications of artificial neural networks.* Cambridge: Cambridge University Press.

Edelmann, G. (1987). *Neural Darwinism: The Theory of Neuronal Group Selection.* New York: Basic Books.

Egorov, A.V., Hamam, B.N., Fransén, E., Hasselmo, M.E. & Alonso, A.A., Graded

Persistent Activity in Entorhinal Cortex Neurons. (2002). *Nature,* 420, ss. 173–178.

Elman, J.L. (1990). Finding structure in time. *Cognitive Science,* 14, ss. 179–211.

Ericsson, K.A. & Kintsch, W. (1995). Long-term working memory. *Psychological Review,* 102, ss. 211–245

Fausett, L. (1994). *Fundamentals of Neural Networks.* Englewood Cliffs, NJ: Prentice Hall.

Fodor, J. (1975). *The Language of Thought.* Cambridge, MA: Harvard University Press.

Fransén, E., Tahvildari, B., Egorov, A.V., Hasselmo, M.E. & Alonso, A.A. (2006). Mechanism of Graded Persistent Cellular Activity of Entorhinal Cortex Layer V Neurons. *Neuron,* 49, ss. 735–4.

French, R. (red.) (2002). *Implicit Learning and Consciousness: An Empirical, Philosophical and Computational Consensus in the Making.* Hove, East Sussex: Psychology Press.

Gerstner, W. & Kistler, W.M. (2002). Mathematical formulations of Hebbian learning. *Biological Cybernetics,* 87, ss. 404–15.

Gibson, E.J. (1969). *Principles of perceptual learning and development.* New York: Appleton-Century-Crofts

Gibson, J.J. & Gibson, E.J. (1955). Perceptual learning: differentiation or enrichment? *Psychological Review,* 62, ss. 32–41.

Goldstone, R.L. (1998). Perceptual learning. *Annual Review of Psychology,* 49, ss. 585–612.

He, B. (red.) (2005). *Neural Engineering.* New York: Kluwer Academic/Plenum Publishers.

Hebb, D. (1949). *Organization of Behavior.* New York: Wiley.

Herz, A.V.M. (2006). How is time represented in the brain? I van Hemmen, J.L. & Sejnowski, T.J. (red). *23 Problems in Systems Neuroscience.* Oxford: Oxford University Press. Ss. 266–82.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences,* 6, ss. 242–7.

Hinde, R. (1970). *Animal behaviour. A synthesis of ethology and comparative psychology.* 2:a uppl. Tokyo: McGraw-Hill Kogakusha Ltd.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences,* 79, ss. 2554–8.

Hopfield, J.J. & Tank, D.W. (1985). Neural computations of decisions in optimization problems. *Biological Cybernetics,* 52, ss. 141–52.

Holland, A. (1974). Retained knowledge. *Mind,* 83, ss. 355–71.

Howieson, D.B. & Lezak, M.D. (1995). Separating memory from other cognitive pro-

blems. I Baddeley, A.D., Wilson, B.A. & Watts, F.N. (red.). *Handbook of Memory Disorders.* Chichester: Wiley. Ss. 411–26.

Hubel, D.H. & Wiesel, T.N. (1965). Receptive fields and functional architecture in two non-striate visual areas (18 and 19) of the cat. *Journal of Neurophysiology,* 28, ss. 229–89.

Husmeier, D., Dybowski, R. & Roberts, S. (red.) (2005). *Probabilistic Modeling in Bioinformatics and Medical Informatics.* London: Springer-Verlag.

Husserl, E. (1990–01). *Logische Untersuchungen.* Tübingen: Niemeyer.

Izhikevitch, E.M. (2007). *Dynamical Systems in Neuroscience.* Cambridge, MA: MIT Press.

Kamruzzaman, J., Begg, R. & Sarker, R. (2006). *Artificial Neural Networks in Finance and Manufacturing.* Hershey, London, Melbourne & Singapore: Idea Group Publishing.

Kandel, E. & Spencer, W.A. (1968). Cellular neurophysiological approaches in the study of learning. *Physiological Review,* 48, ss. 65–134.

Kandel, E. (2001). The molecular biology of memory storage: A dialog between genes and synapses. *Biosciences Reports,* 21, ss. 565–611.

Karniel, A. & Mussa-Ivaldi, F.A. (2003). Sequence, time or state representation: how does the motor control system adapt to variable environments? *Biological Cybernetics,* 89, ss. 10–21.

Kohonen, T. (2001). *Self-Organizing Maps.* 3:e uppl. Berlin: Springer-Verlag.

Konorski, J. (1948). *Conditioned Reflexes and Neuron Organization.* Cambridge: Cambridge University Press.

Korsakoff, S. (1890). Ueber eine besondere form psychischer Störung kombiniert mit multipler Neuritis. *Archiv für Psychiatrie und Nervenkrankheiten,* 21, ss. 669–704.

Kung, S.Y, Mak, M.W. & Lin, S.H. (2005). *Biomethric Authentication. A Machine Learning Approach.* Upper Saddle River, NJ: Prentice-Hall Professional Technical Reference.

Kupferman, I. & Kandel, E. (1969). Neuronal controls of a behavioral response mediated by the abdominal ganglion of *Aplysia. Science,* 164, ss. 847–50.

Lagus, K., Kaski, S. & Kohonen, T. (2004). Mining massive document collections by the WEBSOM method. *Information Sciences,* 163, ss. 135–156.

Larsson, L.-E. (2000). *Neurofysiologi.* Lund: Studentlitteratur.

Lebedev, M.A. & Nicolelis, M.A.L. (2006). Brain-machine interfaces: past, present and future. *Trends in Neurosciences,* 29, ss. 536–46.

Lindqvist, G. & Malmgren, H. (1990). *Organisk Psykiatri. Teoretiska och kliniska aspekter.* Stockholm: Almqvist & Wiksell.

Lindqvist, G., Andersson, H., Bilting, M., Blomstrand, C., Malmgren, H. & Wikkelsø, C. (1993). Normal pressure hydrocephalus: psychiatric findings before and

after shunt operation classified in a new diagnostic system for organic psychiatry. I Lindqvist, G. & Malmgren, H. *Classification and diagnosis of organic mental disorders. Acta Psychiatrica Scandinavica,* 88, Suppl. 373. Ss. 18-32.

Lipowski, Z.J. (1980). Organic mental disorders: introduction and review of syndromes. I Kaplan, H., Freedman, A. & Sadock, B. (red.). *Comprehensive Textbook of Psychiatry.* 3:e uppl. Vol. 2. Washington, DC: American Psychiatric Press. Ss. 1359–92.

Lisman, J. The CaM Kinase II Hypothesis for the Storage of Synaptic Memory. (1994). *Trends in Neurosciences,* 17, ss. 406–412.

Maas, W. & Bishop, C.M. (1999). *Pulsed Neural Networks.* Cambridge, MA: MIT Press.

Mackintosh, N.J. (1974). *The Psychology of Animal Learning.* London: Academic Press.

Mackintosh, N.J. (1983). *Conditioning and Associative Learning.* Oxford: Oxford University Press.

Malmgren, H. (1984). Habituation and associative learning in random mixtures of deterministic automata. *Göteborg Psychological Reports* 15:2. Göteborg: Göteborgs Universitet.

Malmgren, H. (1985). On the nature of reinforcement. *Göteborg Psychological Reports,* 16:3. Göteborg: Göteborgs Universitet.

Malmgren, H. (1991). Learning by natural resonance. *Göteborg Psychological Reports,* 21:6. Göteborg: Göteborgs Universitet.

Malmgren, H. (1996). Perceptual expectations and the learning of temporal sequences. *Philosophical Communications, Red Series,* 35. Göteborg: Göteborgs Universitet.

Malmgren, H. (2002). Forced learning of graded responses. *Philosophical Communications, Web Series,* 26. Göteborg: Göteborgs Universitet. Tillgänglig: http://www.phil.gu.se/posters/hmgraded.pdf

Malmgren, H. (2004). Why the past is sometimes perceived, and not only remembered. *Philosophical Communications, Web Series,* 31. Göteborg: Göteborgs Universitet. Tillgänglig: http://www.phil.gu.se/posters/HMbuffer.pdf

Malmgren, H. (2005). The theoretical basis of the biopsychosocial model. I White, P.D. (red.), *Biopsychosocial medicine. An integrated approach to understanding illness.* Oxford: Oxford University Press. Ss. 21–35.

Malmgren, H. (2006). The essential connection between representation and learning. *Philosophical Communications, Web Series,* 36. Göteborg: Göteborgs Universitet. Tillgänglig: http://www.phil.gu.se/posters/hmoxford.pdf

Malmgren, H. (2007). Är schizofreni en hjärnsjukdom? *Läkartidningen,* 105, ss. 2152-5.

Malmgren, H., Borga, M. & Niklasson, L. (red.) (2000). *Artificial Neural Networks in Medicine and Biology.* London: Springer-Verlag.

Malmgren, H. & Östensson, O. Selective attention is selective learning. *Göteborg Psychological Reports,* 19:2. Göteborg: Göteborgs Universitet.

Mandler, J.M. & Mandler, G. (1964). *Thinking: From Association to Gestalt.* New York: Wiley.

Marr, D. (1982). *Vision.* New York: Freeman.

Massaro, D.W. & Loftus, G.R. (1996). Sensory and perceptual storage: data and theory. I Bjork, E.L. & Bjork, R.A. (red.), *Memory*. San Diego, CA: Academic Press. Ss. 68–101.

McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics,* 7, ss. 115–33.

Mérida-Casermeiro, E., Galán-Marín, G & Muñoz-Pérez, J. (2001). An efficient multivalued Hopfield network for the Traveling Salesman problem. *Neural Processing Letters,* 14, ss. 203–16.

Merleau-Ponty, M. (1962). *Phenomenology of Perception.* London: Routledge. [Franska originalet 1945: *Phénoménologie de la Perception.* Paris: Éditions Gallimard.]

Millikan, R.G. (1987). *Language, Thought, and Other Biological Categories: New Foundations for Realism.* Cambridge, MA: MIT Press.

Milner, B., Corkin, S. & Teuber, H.-L. (1968). Further analysis of the hippocampal amnesic syndrome. 14 year follow-up study of H.M. *Neuropsychologica* 6, ss. 215–34.

Minsky, M. & Papert, S. (1969). *Perceptrons.* Cambridge, MA: MIT Press.

Minsky, M. & Selfridge, O. (1961). Learning in random nets. I Cherry, C. (red.), *Information Theory: Fourth London Symposium.* London: Butterworths.

Nestoriuc, Y. & Martin, A. (2007). Efficacy of biofeedback for migraine: a meta-analysis. *Pain,* 128, ss.111–27.

Nordin, P. (2003). *Humanoider. Självlärande robotar och artificiell intelligens.* Stockholm: Liber.

O'Keefe, J. & Nadel, L. (1978). *The Hippocampus as a Cognitive Map.* Oxford: Clarendon.

Olds, J. & Milner, P. (1954). Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain. *Journal of Comparative and Physiological Psychology,* 6, ss. 419–427.

Olsson, S.E., Ohlsson, M., Ohlin, H., Dzaferagic, S., Nilsson, M.L., Sandkull, P. & Edenbrandt, L. (2006). Decision support for the initial triage of patients with acute coronary syndromes. *Clinical Physiology and Functional Imaging,* 26, ss. 151–6.

O'Shea, M. (2005). *The Brain: A Very Short Introduction.* Oxford: Oxford University Press.

Pavlov, I.P. (1960) *Conditioned Reflexes.* New York: Dover. [Tidigare publicerad 1927. Oxford: Oxford University Press.]

Pavlov, I.P. (1928). *Lectures on Conditioned Reflexes.* New York: International Publishers Co.

Principe, J.C., Euliano, N.R. & Lefebvre, W.C. (2000). *Neural and Adaptive Systems. Fundamentals Through Simulation.* New York: Wiley.

Rabbitt, P. (red.) (1997). *Methodology of Frontal and Executive Function.* Hove, East Sussex: Psychology Press.

Ripley, B.D. (1996). *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press.

Roberts, A.C. & Glanzman, D.L. (2003). Learning in *Aplysia*: looking at synaptic plasticity from both sides. *Trends in Neurosciences,* 26, ss. 662–70.

Roberts, A.C., Robbins, T.W. & Weiskrantz, L. (red.) (1998). *The Prefrontal Cortex. Executive and Cognitive Functions.* Oxford: Oxford University Press.

Rolls, E.T. & Treves, A. (1998). *Neural Networks and Brain Function.* Oxford: Oxford University Press.

Rosenblatt, F. (1962). *Principles of Neurodynamics.* Washington: Spartan Books.

Rumelhart, D. & McClelland, J. (red.) (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* vol. I. Cambridge, MA: MIT Press.

Rödholm, M. (2003). *Astheno-Emotional Disorder after Aneurysmal Subarachnoid Hemorrhage.* Diss. Göteborgs Universitet. Göteborg: Univ.

Samsonovich, A. & McNaughton, B.L. (1997). Path Integration and Cognitive Mapping in a Continuous Attractor Neural Network Model. *Journal of Neuroscience,* 17, ss. 5900–5920.

Sayre, K. (1976). *Cybernetics and the Philosophy of Mind.* Atlantic Highlands, N.J.: Humanities Press.

Sayre, K. (1986). Intentionality and information processing: an alternative model for cognitive science. *Behavioral and Brain Sciences,* 9, ss. 121–160.

Schölkopf, B., Burges, C.J.C. & Smola, A.J. (1999). *Advances in Kernel Methods: Support Vector Learning.* Cambridge: MIT Press.

Seung, H.S. (1996). How the Brain Keeps the Eyes still. *Proceedings of the National Academy of Science USA,* 93, ss. 13339–13344.

Simpson, P. (1990). *Artificial neural systems*. New York: Pergamon Press.

Shannon, C.E & Weaver, W. (1949). *The Mathematical Theory of Communication.* Urbana, IL, Chicago & London: University of Illinois Press.

Skinner, B.F. (1981). Selection by consequences. *Science,* 213, ss. 501–504

Sokolov, E.M. (1963). Higher nervous functions: The orienting reflex. *Annual Review of Physiology,* 25, ss. 545–80.

Sperling, G. (1960). The information available in brief visual presentations. *Psychological Monographs,* 74, ss. 11ff.

Stamenov, M.I. & Gallese, V. (red.) (2002). *Mirror Neurons and the Evolution of*

Brain and Language. Amsterdam & Philadelphia: John Benjamins.

Stringer, S.M., Trappenberg, T.P., Rolls, E.T. & de Aranjo, I.E.T. (2002). Self-organising continuous attractor networks and path integration: one-dimensional models of head direction cells. *Network: Computation in Neural Systems,* 13, ss. 217–42.

Stuss, D.T. & Levine, B. (2002). Adult clinical neuropsychology: Lessons from studies of the frontal lobe. *Annual Review of Psychology,* 53, ss. 401–33.

Sundqvist, F. (2003). *Perceptual Dynamics.* Diss. Göteborgs Universitet. Göteborg: Univ.

Sutton, R.S. & Barto, A.G. (1998). *Reinforcement Learning. An Introduction.* Cambridge, MA: MIT Press.

Tesauro, G. (2002). Programming backgammon using self-teaching neural nets. *Artificial Intelligence,* 134, ss. 181–99.

Thomas, B. (2003). *Framtidens intelligens.* Lund: Studentlitteratur.

Trappenberg, T.P. (2002). *Fundamentals of Computational Neuroscience.* Oxford: Oxford University Press.

Tulving, E. (2002). Episodic memory: From mind to brain. *Annual Review of Psychology*, 53, ss. 1–25.

Tye, M. (2000). Knowing what it is like: the ability hypothesis and the Knowledge Argument. I: Preyer, G. & Siebert, F., *Reality and Humean Supervenience.* Lanham, MD: Rowman & Littlefield.

Uttal, W.R. (2001). *The New Phrenology. The Limits of Localizing Cognitive Processes in the Brain.* Cambridge, MA: MIT Press.

Victor, M., Adams, R. & Collins, G. (1971). *The Wernicke-Korsakoff Syndrome.* Oxford: Blackwell.

Wang, D. (1993). A neural model of synaptic plasticity underlying short-term and long-term habituation. *Adaptive Behavior,* 2, ss. 111–29.

Warrington, E. & Weiskrantz, L. (1982). Amnesia: A disconnection syndrome? *Neuropsychology,* 20, ss. 233–48.

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Diss. Harvard University. Boston: Harvard Univ.

Wessberg, J., Nicolelis, M.A. (2004). Optimizing a linear algorithm for real-time robotic control using chronic cortical ensemble recordings in monkeys. *Journal of Cognitive Neuroscience*, 16, ss. 1022–35.

Whitlock, J., Heynen, A., Shuler, M. & Bear, M. (2006). Learning induces long-term potentiation in the hippocampus. *Science,* 313, ss. 1093-7.

Wiemer, J.C. (2003). The time-organized map algorithm: extending the self-organizing map to spatiotemporal signals. *Neural Computation,* 15, ss. 1143–71.

Wilson, H.R. (1999). *Spikes, decisions and actions. The dynamical foundations of neuroscience.* Oxford: Oxford University Press.

Wittgenstein, L. (1953). *Philosophische Untersuchungen/Philosophical Investigations.* Oxford: Blackwell.

Wulff, H. (1981). *Rational Diagnosis and Treatment.* 2:a uppl. Oxford: Blackwell.

Wyers, E.J., Peeke, H.V.S & Herz, M.J. (1973). Behavioral habituation in invertebrates. I Peeke, H.V.S. & Herz, M.J. (red.). *Habituation. Vol. I: Behavioral Studies.* New York & London: Academic Press. Ss. 1–57.

Yi, Z. & Tan, K.K. (2004). *Convergence Analysis of Recurrent Neural Networks.* Boston, Dordrecht & London: Kluwer Academic Publishers.

Sakregister

Flerordsfraser har ofta, men långtifrån alltid, bokförts endast på huvudordet. Således "Aktiveringsfunktion, linjär" och "Betingning, klassisk", men "Mental representation" och "Återkopplat ANN". Detsamma gäller många sammansatta ord. "Frekvenskod" står exempelvis som "Kod, frekvens-".