UNIVERSITY OF GOTHENBURG

Option Modelling by Deep Learning

Niclas Klausson
niclas@klausson.se

Victor Tisell
victor@tritonic.se

Abstract

In this thesis we aim to provide a fully data driven approach for modelling financial derivatives, exclusively using deep learning. In order for a derivatives model to be plausible, it should adhere to the principle of no-arbitrage which has profound consequences on both pricing and risk management. As a consequence of the *Black-Scholes model* in Black & Scholes (1973), *arbitrage theory* was born. Arbitrage theory provides the necessary and sufficient formal conditions for a model to be free of arbitrage and the two most important results are the first and second *fundamental theorems of arbitrage*. Intuitively, under so called *market completeness*, the current price of any derivative/contingent claim in the model must reflect all available information and the price is unique, irrespective of risk-preferences. In order to arrive at an explicit arbitrage free price of any contingent claim, a choice must be made in order to simulate the distribution of the asset in the future. Traditionally this is achieved by the theory of *random processes* and *martingales*. However, the choice of random process introduces a type of *model risk*.

In Buehler et al. (2019), a formal theory was provided under which hedging and consecutively pricing can be achieved irrespective of choice of model through deep learning. However, the challenge of choosing the right random process still remains. Recent developments in the area of *generative modelling* and in particular the successful implementation of *generative adversarial networks (GAN)* in Goodfellow et al. (2014) may provide a solution. Intuitively speaking, a GAN is a game theoretic learning based model in which two components, called the *generator* and *discriminator*, competes. The objective being to approximate the distribution of a given random variable.

The objective of this thesis is to extend the *deep hedging* algorithm in Buehler et al. (2019) with a generative adversarial network. In particular we use the *TimeGAN* model developed by Yoon et al. (2019). We illustrate model performance in a simulation environment using geometric Brownian motion and Black-Scholes prices of options. Thus, the objective our model is to approximate the theoretically optimal hedge using only sample paths of the trained generator. Our results indicates that this objective is achieved, however in order to generalise to real market data, some tweaks to the algorithm should be considered.

Keywords: Deep learning, deep hedging, generative adversarial networks, arbitrage pricing.

Bachelor´s thesis in Economics, 15 credits
Fall Semester 2020
Supervisor: Andreas Dzemski

Department of Economics
School of Business, Economics and Law
University of Gothenburg

Acknowledgements

We would like to express our thanks and gratitude to our thesis supervisor Andreas Dzemski, for his constructive recommendations and comments.

Contents

Commonly used Mathematical Symbols

- Ω: *Sample space.* Non empty set on which random variables map into real numbers.
- $\mathcal{T}$: *Time index set.* Defines a sequence, countable or uncountable depending on the countability of $\mathcal{T}$.
- $\mathcal{F}$: *σ-algebra.* Defines which events that are measurable.
- $\mathbb{F}$: *Filtration of σ-algebras.* Constitutes a sequence of $\sigma-$algebras and can intuitively be thought of as the information available at each $t \in \mathcal{T}$.
- $\mathbb{P}$: *Probability measure.* Assigns the probability of a *measurable* event, i.e. a element of a σ-algebra. In the context of financial modelling, often referred to as the *physical/real-world* measure.
- $\mathbb{Q}$: *Equivalent martingale measure/pricing measure.* Probability measure that shares the same null sets as $\mathbb{P}$. Used in the context of pricing derivatives.
- *Random process*: A function on the set $\Omega \times \mathcal{T}$ which is a random variable for each $t \in \mathcal{T}$.
- W. *Brownian motion.* Specific type of random process that. Can be thought of as representing random noise.
- S: *Market prices of underlying assets.* A random process describing how asset prices in a market of $d+1$ for $d \geq 0$ assets evolves over time.
- T: Maturity of a specific derivative (depending on use case). $t \leq T \in \mathcal{T}$.
- X: Payoff/cash-flow of a given derivative on S^1. Can be thought of as a generalization of a call option.
- H: *Portfolio process.* Trading strategy on S describing how many *units* to hold in each asset.
- V: *Portfolio value process.* The wealth of the portfolio H, sometimes for clarity denoted V^H.
- *Self-financing portfolio.* A portfolio is self-financing if one does not consume or add capital beyond initial capital.
- $(H \cdot S)_T$: *Stochastic integral.* Gains of the trading strategy H up until time $T \in \mathcal{T}$.
- V_0: The amount of funding required for a trading strategy (wealth at time $t = 0$).
- Π: The arbitrage free price of a derivative with payoff X.
- Π_0: *Current* market price of a derivative with payoff X at maturity. All current information is embedded in price.
- $\mathcal{G}, \mathcal{D}$: *Generator and Discriminator* respectively. The essential components of a generative adversarial network.
- θ. Parameter vector. Usage depends on context. $\theta \in \Theta$ where Θ is referred to as parameter space.

1 Introduction

Risk management and pricing for portfolios of derivative contracts is of great importance to academics and practitioners alike. The global derivatives market is very complex and can have great societal impact, as seen in the 2008 financial crisis. New developments in technology has enabled new numerical methods for addressing both pricing and risk management. In particular the practical application of *deep learning* methods, which is a class of statistical algorithms with computational procedures similar to biological neural networks, like our brain. In Buehler et al. (2019) deep learning was successfully applied to hedging and pricing of simple derivatives. In order for the model to be complete and risk management to be realistic, many researchers agree that simulation should also be conducted by deep learning. Many solutions has been proposed, see e.g. Kondratyev & Schwarz (2019), Wiese et al. (2020) and Buehler et al. (2020), all of which use so called *neural samplers* for simulation. To our knowledge, no papers has been published on how to *combine* neural samplers with the *deep hedging* algorithm in Buehler et al. (2019) and its effect on hedging and pricing. Given the impressive performance and utility of the deep hedging model, this is in our view, likely to be of fundamental importance in the general area of derivative modelling in the future.

In this thesis we aim to shed some light on the combination of neural samplers with the deep hedging algorithm, by combining a special type of neural sampler with the deep hedging model. In order for a more precise understanding of our proposed model, further context is required.

Financial derivatives are contracts in a financial market that specifies an exchange of cash-flows between the holder and the seller according to some agreed upon scheme. A subclass of financial derivatives is that of *(simple) contingent claims* which are contracts where the payoff is specified at a single point in time. One of the more liquidly traded examples of a contingent claim is that of a *European option*. A European option is a bilateral agreement between two counter-parties to bestow the owner of the contract with the right to purchase/sell an *underlying asset* at a pre-specied time and price, called the *maturity* and *strike* respectively. As earlier mentioned, a large part of financial mathematics research is devoted to the pricing and risk management for portfolios of contingent claims. In arbitrage theory, the concept of *arbitrage* dictates the conditions for suitable derivative pricing models. Economically, no arbitrage means that no risk-free profits can be made above the risk-free rate. Arguably the most fundamental insight of the famous *Black-Scholes* in Black & Scholes (1973) was that the concept of *hedging* and arbitrage free pricing are actually equivalent. If one can trade in all risks dictating the payoff of a claim, the arbitrage free price of said claim is proportional to the *funding* required for a trading strategy to achieve the same payoff. This portfolio H is called a *hedging/replication* strategy and is unique. Using arbitrage theory, it is then easily shown that the price of the claim is also given by its discounted expected payoff at maturity, which is known as the general pricing formula for simple contingent claims.

However, suppose that not all risks are tradeable. Then the arbitrage free price of the claim is no longer uniquely given since a claim may exhibit *intrinsic* risk affecting its payoff at maturity. Hence, the concept of hedging/pricing is reduced from risk *elimination* to risk *minimization*. A market in which not all risks are liquidly traded is more generally referred to as an *incomplete market* and its implications on hedging and pricing has been studied for a long time, see e.g. Föllmer & Schweizer (1991), Schweizer (1995) and Föllmer & Leukert (1999). In Schweizer (1995), the hedging problem is characterised as a minimization over the profit-loss of a hedged position in the claim.

Combining the universal approximation results in Hornik (1991) and the empirical successes of *artificial neural networks (ANN)*, the *deep hedging* framework was developed. In essence, the deep hedging framework can be seen as a theoretical justification for the implementation of ANN´s to hedging in incomplete markets. In Buehler et al. (2019) they reduce the infinite dimensional problem of finding optimal hedging strategies, to a finite dimensional problem of finding optimal parameters for a neural network. Intuitively, the procedure can be described as predicting the hedging strategies such that the *risk* of the error in the hedged position is minimized.

Given the intuition presented above, it becomes apparent that a reasonable derivative pricing system should to be able to simulate the distribution of asset prices on which the hedging strategies are formed. Traditionally, this is achieved by specifying *random processes* which dictates how asset prices are allowed to evolve in the future. Furthermore, the deep hedging framework provides a formal theory in which optimal hedging strategies can be derived from any a given asset price process using machine learning. A natural consequence of this fact is the need for a more naturalistic approach to financial time-series generation. Otherwise, the deep hedging algorithm is not able to completely transcend traditional pricing and hedging models and is thus still naturally constrained by the insufficiency of traditional random processes. Recent research, see e.g. Kondratyev & Schwarz (2019), Wiese et al. (2020) and Buehler et al. (2020), suggests that such naturalistic cross asset simulation can be achieved by learning based generative models, called *neural samplers*.

As mentioned in the beginning of the introduction, this thesis explores the possibility of extending the deep hedging algorithm by a subclass of neural samplers, called *generative adversarial networks (GAN)*, originally introduced by Goodfellow et al. (2014). In particular, the *TimeGAN* architecture proposed by Yoon et al. (2019) is used for time-series generation and *recurrent neural networks (RNN)* for the implementation of the deep hedging algorithm. Conceptually, one can think of the proposed model as an iterative process where the financial time-series is first being embedded in a lower dimensional latent space representation. Then if random noise is mapped by a function such that the mapped noise is in some sense "dense" in the latent space, classification into fake and real samples is futile. By finding the inverse embedding map on the latent space representation of the noise process, the marginal distributions of the original time-series are approximated. We then use this distributional approximation to predict hedging strategies, with the objective being to minimize the error associated to a hedged portfolio formed by trading in the derivative and underlying market.

We test our proposed model in a simulation environment using the Black-Scholes model where the theoretically optimal hedge and arbitrage free price is known and has a analytical solution. Therefore, the objective of the model is to replicate the performance of the Black-Scholes model. Our results indicates that this objective is achieved by our model. However from a modelling perspective, the more interesting question of using real asset price time-series to infer their distribution and hedges was unattainable in our current model. We found that the TimeGAN algorithm could not sufficiently well learn the distribution of the underlying asset price processes to a satisfactory extent. However, this could be attributable to insufficient attention placed on the pre-processing of the data to suit the specific application of random process approximation. Therefore we suggest potential future researchers to pay closer attention to the pre-processing. Especially through signature transforms, such as Ni et al. (2020), Kidger et al. (2019) and Buehler et al. (2020), since they will also accelerate learning.

Outline. The thesis starts with a literature review in Section 2 which covers previous conducted research in the areas directly related to the objective of the thesis. Section3 extends the basic probability theory presented in Appendix A.2 by developing the mathematics required for the Arbitrage theory and portfolio dynamics covered in Section 4. Hence, the objective of Section 3 is mainly to fix notation, therefore the mathematically familiar reader can skip the first section. However, the unfamiliar reader is also recommended to read Appendix A, which develops the mathematical prerequisites for this thesis. As alluded to above, Section 4 formalises the concept of arbitrage free markets and its impact on pricing. Furthermore,Section 4 also show how arbitrage theory is effected by market imperfections. Therefore, the objective of Section 4 is to provide the formal theory required by the reader to intuit the objective of the thesis. In Section 5 the notions of *artificial neural networks (ANN)* as universal approximators is developed. Furthermore Section 5 introduces neural networks, adapted to sequential data and the approximation of probability distributions through *recurrent neural networks (RNN)* and *generative adversarial networks (GAN)* respectively. Hence, Section 5 develops the theoretical background to the proposed methodology, presented in Section 6. Lastly, we provide some experimental results in Section 7, which illustrates the performance of the proposed model and concluding remarks in Section 8.

2 Literature Review

As alluded to in the introduction, this thesis aims to extend the deep hedging algorithm, presented in Buehler et al. (2019) by a deep generative model. The intuition behind deep hedging, detailed formally in Section 6, is that one can hedge, i.e. reduce the risk, associated to a position in a derivative, by minimising the induced risk for the profit-and-loss of a portfolio. This portfolio is formed by trading on the general market and the units to be held in each asset is predicted by a artificial neural network. The deep hedging framework provides the formal justification to apply artificial neural networks for addressing the problem of hedging in incomplete markets. However, as discussed in the introduction, researchers in mathematical finance has long sought to address incomplete market pricing/hedging and as such there exists quite a large amount of literature on the topic. In Föllmer & Schweizer (1991), hedging in incomplete markets is characterised by a terminal condition placed on the formation of portfolios. In essence, the idea is that the objective of hedging corresponds to choosing the units held in each asset such that the risk of the profit-and-loss distribution is minimized. Formally, this means that a hedged portfolio will still contain risk, which is naturally referred to as *unhedgeable/intrinsic* risk, characterised by a orthogonal error in the profit-and-loss distribution of the hedged position. This idea was later extended to discrete-time in Schweizer (1995), in which it is shown that the variance optimal hedging position is derived from minimizing the squared length of the terminal hedging error vector in L^2. Furthermore, one popular area of research is what is called *super-replication* which attempts to address the problem of hedging derivatives in incomplete markets, see e.g. Föllmer & Leukert (1999) and Delbaen & Schachermayer (2006) theorem 2.4.2 for further details. However this thesis focuses more explicitly on the characterisation in Schweizer (1995) since the problem is clearly formulated in terms of a optimization that is attainable by a learning based model, such as the deep hedging algorithm.

Any reasonable derivative pricing model should have the ability of both pricing through hedging *and* asset price simulation. Inevitably, derivative prices will be a function of the future state of the simulated asset prices. Given the intuition gained above, any hedging portfolio will be formed by trading in the market and thus, one naturally needs to prescribe a model for the

market. For example, the *Black-Scholes* model, proposed by Black & Scholes (1973), uses so called *geometric Brownian motion*, which is a continuous random process which allows for the closed form nature of the Black-Scholes model. See e.g. Delbaen & Schachermayer (2006) ch. 4.4 for further analysis. In Buehler et al. (2019), the deep hedging model still rely on the specification of such a *classical* model which describes the possible states of the world by a, arguably simplistic, equation. Hence, extending the deep hedging framework by a generative model that is not limited by a parametric description is very natural and is a relatively active current area of research Vittori et al. (2020). To our understanding, there exists three main candidates in *deep* generative modelling of financial time-series, *restricted Boltzmann machines (RBM)*, *variational auto-encoders (VAE)* and the latest addition *generative adversarial networks (GAN)* invented by Goodfellow et al. (2014). In Kondratyev & Schwarz (2019) and Buehler et al. (2020), they apply restricted RBM and VAE respectively to define a *market generator*, which aims to preserve the multivariate dependency structure of asset price processes. Furthermore, Wiese et al. (2020) uses GAN to define so called *neural processes* with similar objective. Lastly, Ni et al. (2020) also uses generative adversarial networks in the context of financial time-series modelling. The central distinction of Buehler et al. (2020) and Ni et al. (2020) is that they utilise so called *signature transforms* to describe the closeness in distribution random processes. Furthermore, Kondratyev & Schwarz (2019) shows that the multivariate dependency structure can be preserved and especially non-linear correlations and auto-correlation. Recall that the objective of this thesis is to extend the deep hedging algorithm by a generative model, the results in Ni et al. (2020) and Buehler et al. (2020) are more relevant to this thesis. However, we consider a alternative architecture called *TimeGAN* proposed by Yoon et al. (2019) to approximate random processes describing market dynamics.

To our knowledge, no other articles has specifically applied a deep generative model to extend the deep hedging framework and thesis aims to provide insights to their joint applicability. Furthermore, this thesis also trivially differ from the theory proposed in Buehler et al. (2019) by considering fully *recurrent neural networks (RNN)* as opposed to *semi-recurrent* networks.

3 Stochastic Integration & Martingales

This section shows the minimum necessary mathematics required for arbitrage theory used in Section 4. Hence, this section is the natural extension of the probability theory covered in Appendix A to sequential dynamics. We strongly advise readers that are unfamiliar with probability theory to read Appendix A. Furthermore, for the mathematically initiated reader, this section will mainly fix notation and serve as an introduction to the subject.

Outline. The main objective of this section is to define the notions of semi-martingales and stochastic integration, which are essential for arbitrage theory. We start by a formal definition of random processes and in particular, *adapted* and *predictable* random processes from a *filtration* of σ-algebras. Then we proceed by defining so called *martingales* as a subclass of adapted processes. Lastly, the necessary definitions required for the formal understanding of a *semi-martingale* and stochastic integration are provided, both in continuous and discrete time.

The basic and most relevant objects of study are *random processes*. Before a random process can be defined it (trivially) has to be noted that for any *index* set $\mathcal{T}$, $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P})$ constitutes a probability space, see Definition A.8.

Definition 3.1. Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P})$ be a probability space and $(E, \mathcal{E})$ a measurable space, as per

Definition A.6. Then a random process $X = (X_t)_{t\in\mathcal{T}}$ is a map

$$X : \Omega \times \mathcal{T} \longrightarrow E$$

such that for all $t \in \mathcal{T}$, X_t is a $(\mathcal{F}, \mathcal{E})$-measurable random variable, see Definition A.9.

Example. A random process with many good analytical properties, as shall be seen later, is that of a *Brownian motion*/Weiner process.

Definition 3.2. A Brownian motion $W : \Omega \times \mathbb{R}^+ \longrightarrow \mathbb{R}$ is a random process satisfying

i . $W_0 = 0$.

ii . $W_{t+u} - W_t$ is *independent* of W_s for any $0 < s \leq t$ and $u \geq 0$.

iii . $W_{t+u} - W_t \sim \mathcal{N}(0, u)$.

iv . $W_\cdot(\omega)$ is a continuous function of t.

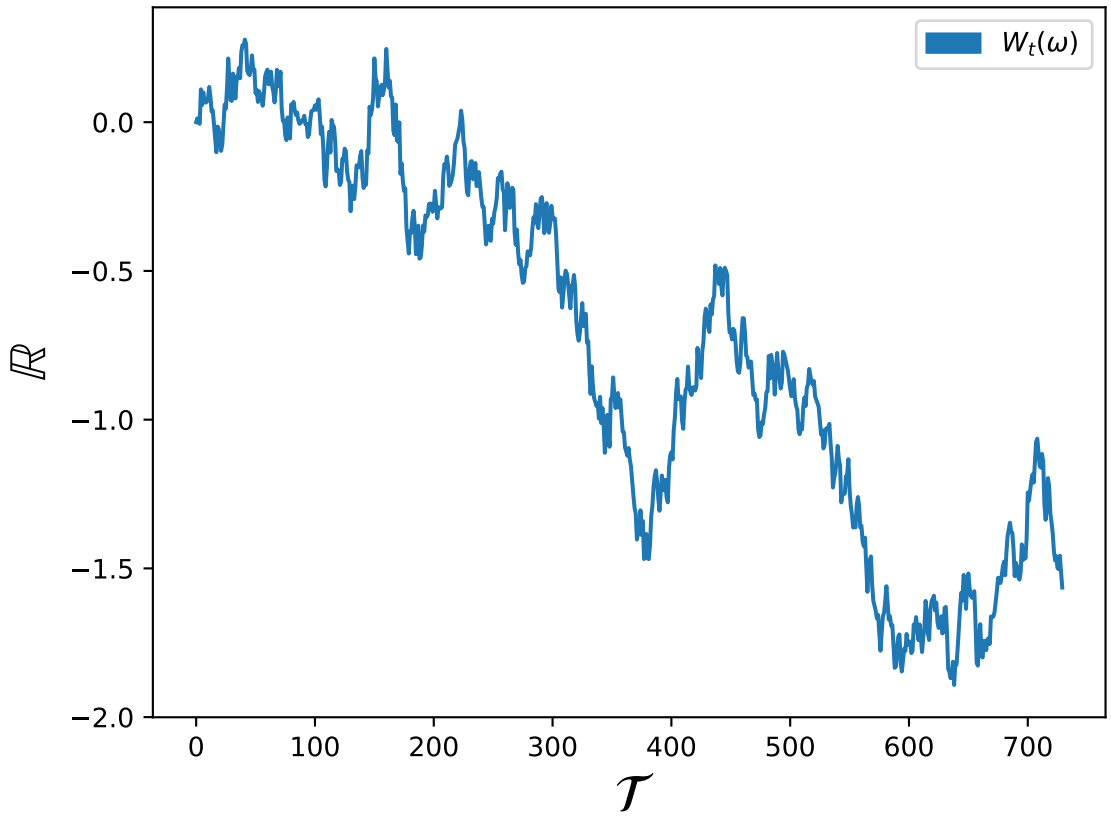

Figure 1: Values of a Wiener process/Brownian motion W for a fixed $\omega \in \Omega$ as a function of time. $\mathcal{T}$ is set to a countable index set of length 730 constituting 2 years of daily datapoints.

A sample path of Brownian motion is visualised in Figure 1. Brownian motion is a special case of a so called *martingale*.In order to develop the notion of a martingale, the notion of a σ-algebra, see Definition A.3, have to be extended to*sequences* of σ-algebras indexed by time, as a model for information-flows.

Definition 3.3. Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P})$ be a probability space. A *filtration* $\mathbb{F} = (\mathcal{F}_t)_{t\in\mathcal{T}}$ is a monotonic sequence of σ-algebras, i.e.

$$\mathcal{F}_s \subset \mathcal{F}_t \subset \mathcal{F}$$

where $\mathcal{F}_t$ is a σ-algebra on Ω for all $s \leq t \in \mathcal{T}$.

Terminology. The quadruple $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$ is called a *filtered* probability space.

The concept of generated σ-algebras in Definition A.4 is easily extendable to the theory of random processes.

Definition 3.4. Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P})$ be a probability space and $(E, \mathcal{E})$ a measurable space such that there exists a $(\mathcal{F}, \mathcal{E})$-measurable random process $X : \Omega \times \mathcal{T} \longrightarrow E$. The *generated filtration* $\mathbb{F}^X = (\mathcal{F}_t^X)_{t\in\mathcal{T}}$ of X is given by

$$\mathcal{F}_t^X = \sigma(X_s : s \leq t), \qquad \forall\, t \in \mathcal{T}.$$

Recall that $\sigma(X_s : s \leq t) := X_s^{-1}(\mathcal{E})$.

Terminology. Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a filtered probability space and $X : \Omega \times \mathcal{T} \longrightarrow E$.

i . A random process X is called *adapted* to the filtration $\mathbb{F}$ if X_t is $\mathcal{F}_t$-measurable for all $t \in \mathcal{T}$.

ii . A random process X is called *predictable* from the filtration $\mathbb{F}$ if X_t is $\mathcal{F}_{t^-}$ measurable for all $t \in \mathcal{T}$, where $\mathcal{F}_{t^-}$ is the left limit of $\mathcal{F}_t$, see e.g. Rudin et al. (1964) for details.

All results and definitions have now been stated to allow for the definition of a $\mathbb{R}^n$ valued martingale.

Definition 3.5. An $\mathbb{F}$-adapted random process $X : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^n$ is called a $\mathbb{F}$ - martingale (or just martingale for short) if it satisfies the so called martingale identity

$$X_s = \mathbb{E}(X_t|\mathcal{F}_s), \qquad \forall s \leq t \in \mathcal{T}.$$

Until further notice, let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a fixed filtered probability space to prevent clutter.

A class of martingales that are very important in arbitrage theory, since they prove to be so called good integrators, are *semi-martingales*. Therefore, the following sequence of definitions provide the prerequisites for their introduction. We begin by recollecting what a *cádlág* or *right continuous with left limits*-real valued function is.

Definition 3.6. Let $E \subset \mathbb{R}$ and M be a set. A map $f : E \longrightarrow M$ is called a *cádlág* function if for every $x \in E$ it is true that

i . the left limit $f(x^-)$ exists and

ii . the right limit $f(x^+)$ esits and equals $f(x)$.

Terminology. The set of all cádlág functions between two metric spaces is called a *skorokhod space*.

Definition 3.7. The *total variation* of a continuous real valued function on a interval $[0,T] \subset \mathbb{R}$ is the real number

$$\begin{aligned} V &: \mathbb{R} \times \mathcal{C} \longrightarrow \bar{\mathbb{R}}^+ \\ ([0,T], f) &\longmapsto V_0^T(f) := \sup_{\pi_n} \sum_{[t_i, t_{i+1}] \in \pi_n} |f(t_{i+1}) - f(t_i)| \end{aligned}$$

where $(\pi_n)_{n\in\mathbb{N}}$ is a sequence of partitions of $[0,T]$, i.e. $\pi_n = \{0 = t_0 < t_1 < \ldots < t_n = T\}$.

Intuition. In terms of random processes defined on an abstract sample space Ω, it only makes sense to talk about bounded variation in the time component. The total variation of a random process is the largest sum of the euclidean distance two points on the trajectory of the random process.

Terminology. A random process X is of *finite variation*/locally bounded if its total variation is finite over a fixed time interval $[0,T]$, i.e. $V_0^T(X(\omega)) < \infty$.

Definition 3.8. Let $\tau : \Omega \longrightarrow \bar{\mathbb{R}}^+$ be a $\mathcal{F}$-measurable random variable. Then we call τ a $\mathbb{F}-$*stopping time* if τ is $\mathbb{F}$-adapted.

Remarks. If the target is equipped with with the Borel σ-algebra $\mathcal{B}$ the following observations can be made.

i .$B = \{(0,t) : t \in [0,T]\}$, $B \in \mathcal{B}$ hence

$$\tau^{-1}(B) = \{\omega \in \Omega : \tau(\omega) \leq t\}.$$

ii . Let $\tau : \Omega \longrightarrow \mathcal{T}$. Then τ is a stopping time if and only if the random process

$$\begin{aligned} X &: \Omega \times \mathcal{T} \longrightarrow \mathbb{R} \\ (\omega, t) &\longmapsto X_t(\omega) := \begin{cases} 1 & t \leq \tau(\omega) \\ 0 & t > \tau(\omega) \end{cases} \end{aligned}$$

is $\mathbb{F}$-adapted.

Definition 3.9. Let $X : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}$ be an $\mathbb{F}$-adapted process. Then X is called a $\mathbb{F}$-*local* martingale if there exists a countable sequence of almost surely divergent and almost surely monotonic $\mathbb{F}$-stopping times $(\tau_k)_{k\in\mathbb{N}}$ such that

$$X_t^{\tau_k} := X_{\min\{t,\tau_k\}}$$

is a $\mathbb{F}$-martingale, i.e.

$$X_t^{\tau_k} = \mathbb{E}(X_m | \mathcal{F}_t^{\tau_k}), \qquad \forall\, m > \min\{t, \tau_k\}.$$

The concept of a *semi-martingale* is now easily defined:

Definition 3.10. Let $X : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^n$ be a $\mathbb{F}$-adapted random process. X is called a *semi-martingale* if it adheres to the decomposition

$$X = M + A$$

where M is a $\mathbb{F}$-local martingale and A is an $\mathbb{F}$-adapted, cádlág process of local bounded variation.

We can now give a definition of stochastic (Itô) integrals, which is one of the main notions of integration of stochastic processes and will be used throughout the thesis.

Definition 3.11. Let $H : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^n$ be a locally finite variation (FV) $\mathbb{F}$-adapted random process. Furthermore, let $X : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^n$ be a semi-martingale and $(\pi_n)_{n\in\mathbb{N}}$ a sequence of partitions of $[0, t]$. Then there exists a unique semi-martingale $Z : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^n$ such that

$$\lim_{n\to\infty} \mathbb{P}\left(\omega \in \Omega : \left| Z_t(\omega) - \sum_{[t_{i-1},t_i]\in\pi_n} H_{t_{i-1}}(\omega)\left(X_{t_i}(\omega) - X_{t_{i-1}}(\omega)\right)\right| > \varepsilon\right) = 0$$

for any $\varepsilon > 0$. We call Z the Itô integral of H with respect to X and denote by

$$Z_t = (H \cdot X)_t = \int_0^t H_s dX_s.$$

To prevent clutter we drop the time indexing in the integral, i.e.

$$(H \cdot X)_t = \int_0^t H dX.$$

Remark. Actually, in accordance with Protter (2005) the proper notation for the integral is

$$(H \cdot X)_t = \int_{0^+}^t H dX.$$

where 0^+ is the right limit of 0. However, to prevent clutter we always suppress the "+" in the lower bound of the integral. However the above statements has to be said at least once in order to prevent confusion.

Recall that, embedded in the objective of this thesis is to utilise machine learning for the construction of hedging portfolios and since a computer naturally operates on countable sets, one needs to extend Definition 3.11 of the stochastic integral to countable index sets.

Definition 3.12. Let $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a filtered probability space. Furthermore, let $(\pi_n)_{n\in\mathbb{N}}$ be a sequence of partitions of the interval finite interval $[0, t] \subseteq \mathcal{T}$ such that the *discretized* interval is the set

$$\pi_n = \{0 = t_0 < \ldots < t_n = t < \infty\}.$$

In addition let $H : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^n$ be a left continuous, locally bounded, $\mathbb{F}$-adapted random process and $X : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^n$ be a semi-martingale. Then for any $t \in \mathcal{T}$, the *discrete* Itô integral of H with respect to X is the unique semi-martingale

$$[H \cdot X]_t = \sum_{i=0}^{n-1} H_{t_i}(X_{t_{i+1}} - X_{t_i}).$$

Notation. From now on we identify

$$[H \cdot X]_t = (H \cdot X)_t$$

since we only operate in discrete time, unless explicitly stated otherwise.

Sufficient definitions and results has now been established in order to properly introduce the field of *Arbitrage theory* which provides the necessary and sufficient conditions for a model of a market to be *free of arbitrage*.

4 Arbitrage Theory

In this section we aim to provide an acceptable mathematical foundation to the theory regarding the pricing of derivatives, equivalently called *contingent claims*. Therefore, this section contains both the formal and financial theory required to intuit the objective of the thesis.

Outline. In Subsection 4.1 the notion of *self-financing* portfolios is developed utilising the theory of martingales from Section 3. Furthermore Subsection 4.1 provide the definition of arbitrage in terms of self-financing portfolios. Subsection 4.1 also contains the pivotal definition of *equivalent martingale measures*, the importance of which is illustrated by the first (fundamental) theorem of asset pricing.

We then proceed by extending the implications of the first theorem of asset pricing on arbitrage free pricing of contingent claims in Subsection 4.2. Here the previously developed theory of portfolio dynamics is used to define *replication/hedging* strategies and show its implications on arbitrage free pricing. Furthermore, we derive the *general* pricing formula for contingent claims and provide the familiar *risk-neutral* pricing formula as a corollary. We conclude Subsection 4.2 with the uniqueness conditions for the above mentioned equivalent martingale measure, characterised by so called *market completeness*, which are collected in the second (fundamental) theorem of asset pricing.

Lastly, in Subsection 4.3 the theoretical implications on pricing and hedging of moving into a more general market environment is discussed.

4.1 Portfolio Dynamics

When modelling financial markets, the following definition of a financial market is used.

Notation. A *financial* market is a collection of $d+1$ $\mathbb{F}$-adapted asset price processes defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ such that

$$\begin{aligned} S : \Omega \times \mathcal{T} &\longrightarrow \mathbb{R}^{d+1} \\ (\omega, t) &\longmapsto S_t(\omega) := (S_t^0(\omega), S_t^1(\omega), \dots, S_t^d(\omega)). \end{aligned} \tag{1}$$

In this section arbitrage theory in *discrete time* is developed, hence $\mathcal{T} \subseteq \mathbb{N}_0$. Let $\mathcal{F}_0$ be a non-trivial σ-algebra and for a fixed $T \in \mathcal{T}$, $\mathcal{F}_T = \mathcal{F}$ as in Delbaen & Schachermayer (2006). Furthermore, the asset S^0 is characterized by

$$S_t^0 \underset{a.s.}{>} 0, \qquad \forall t \in [0, T]$$

which is commonly referred to as the *numeraîre* asset. Furthermore, S^0 is required to be $\mathbb{F}$-adapted.

Definition 4.1. A *portfolio strategy* $H : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^{d+1}$ is a $\mathbb{F}$-predictable process of the units held in a collection of $d+1$ assets.

Remark. The value process induced by a portfolio strategy H is an $\mathbb{F}$-adapted random process V^H satisfying

$$\begin{aligned} V^H : \Omega \times \mathcal{T} &\longrightarrow \mathbb{R} \\ (\omega, t) &\longmapsto V_t^H(\omega) := \sum_{i=0}^{d} H_t^i(\omega) S_t^i(\omega) \end{aligned} \tag{2}$$

subject to the linear constraint

$$V_0^H = \sum_{i=0}^{d} H_1^i S_0^i$$

since H is predictable and $V_0^H \in L^\infty(\Omega, \mathcal{F}_0, \mathbb{P})$ For further information on , see e.g. Delbaen & Schachermayer (2006) ch 2.

Notation. From now on, the superscript H for V is suppressed in order to prevent clutter.

The pivotal concept in portfolio dynamics is that of a *self-financing* portfolio.

Definition 4.2. A portfolio strategy process H is called *self-financing* if

$$V_t = V_0 + \sum_{i=0}^{d} (H^i \cdot S^i)_t, \qquad \forall\, t \in \mathcal{T} \tag{3}$$

where $(H^i \cdot S_t^i)$ is the discrete stochastic integral as developed in Definition 3.12.

Proposition 1. *Let H be a portfolio strategy and V the associated value process. Then a portfolio satisfies the self-financing condition in Equation* (3) *if and only if the following* re-balancing *condition holds for all $t \leq T-1$:*

$$\sum_{i=0}^{d} H_{t+1}^i S_t^i = \sum_{i=0}^{d} H_t^i S_t^i$$

Proof. See e.g. Delbaen & Schachermayer (2006). □

We now use a change of coordinate system for reasons that will soon become very clear.

Definition 4.3. Let $S = (S^0, S^1, \ldots, S^d)$ be a market as in Equation (1), where S^0 is the numeraire asset satisfying Equation (4.1). Then the *normalized* market is defined as

$$\begin{aligned} \tilde{S} : \Omega \times \mathcal{T} &\longrightarrow \mathbb{R}^{d+1} \\ (\omega, t) &\longmapsto \tilde{S}_t(\omega) := \frac{S_t(\omega)}{S_t^0(\omega)} = \left(1, \frac{S_t^1}{S_t^0}(\omega), \ldots, \frac{S_t^d}{S_t^0}(\omega)\right). \end{aligned}$$

Lemma 1 (Price System Invariance)**.** *Let H be an $\mathbb{F}$-predictable portfolio process. Then H is self-financing in the S-market if and only if H is self-financing in the $\tilde{S}$-market.*

Proof. See e.g. Björk (2009). □

Implications. By Definition 4.3 and Lemma 1 we see that normalizing prices to units of the numeraire asset removes the linear constraint of V_0 in the self-financing condition for the portfolio strategy H. If we express the value process in the normalized price system, i.e.

$$\begin{aligned} \tilde{V} : \Omega \times \mathcal{T} &\longrightarrow \mathbb{R} \\ (\omega, t) &\longmapsto \tilde{V}_t(\omega) := \frac{V_t}{S_t^0}(\omega) = \frac{\sum_{i=0}^{d} H_t^i S_t^i}{S_t^0}(\omega) \\ &= H_t^0(\omega) + \frac{\sum_{i=1}^{d} H_t^i S_t^i}{S_t^0}(\omega) \end{aligned} \tag{4}$$

and recall the self-financing condition in Equation (3) one can immediately conclude that

$$\tilde{V}_t = \tilde{V}_0 + \sum_{i=0}^{d} (H^i \cdot \tilde{S}^i)_t = \tilde{V}_0 + \sum_{i=1}^{d} (H^i \cdot \tilde{S}^i)_t \tag{5}$$

since $d\tilde{S}^0 = 0$. Furthermore $\tilde{V}_0$ is a $\mathcal{F}_0$-measurable random variable. From Equations (4) and (5) one can conclude that there exists a unique H^0 defined by

$$H_t^0 = \tilde{V}_0 + (H \cdot \tilde{S})_t - \sum_{i=1}^{d} H_t^i \tilde{S}_t^i \tag{6}$$

which is a $\mathbb{F}$-predictable process since $\left(H^1, \ldots, H^d\right)$ if $\mathbb{F}$-predictable.

The above implications are now summarised in the form of a proposition.

Proposition 2. *For every $\mathbb{F}$-predictable process $\left(H^1, \ldots, H^d\right)$ there exists a **unique** $H^0 : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}$ such that*

i . For all $t \in \mathcal{T}$ it is true that

$$\tilde{V}_t = \sum_{i=0}^{d} H_t^i \tilde{S}_t^i = \tilde{V}_0 + \sum_{i=1}^{d} (H^i \cdot \tilde{S}^i)_t.$$

ii . $\left(H^1, \ldots, H^d\right)$ is self-financing.

Proof. The uniqueness follows from Equation (6) and *i-ii* follows from Equation (4). □

Example. If S^0 is chosen as a risk free bond with starting value 1, then

$$S_t^0 = e^{\sum_{k=0}^{t} r_k}, \qquad \forall\, t \in \mathcal{T}$$

hence the normalized price system $\tilde{S}$ is nothing but the *discounted* prices of assets since

$$\tilde{S}_t = \frac{S_t}{S_t^0} = e^{-\sum_{k=0}^{t} r_k} S_t.$$

Therefore, not only does discounting have an economic meaning, it also provides a change of coordinate system such that the initial, linear constraint on self-financing portfolios disappear.

One can now provide a simple characterisation of *arbitrage opportunities* in terms of self-financing portfolios.

Definition 4.4. Fix a positive $T \in \mathcal{T}$. A market model admits an *arbitrage opportunity* if there exists a self financing portfolio H, and an associated value process V, such that

i . $V_0 \leq 0$.

ii . $V_T \underset{a.s.}{\geq} 0$.

iii . $\mathbb{P}(\omega \in \Omega : V_T(\omega) > 0) > 0$.

A market model is called *free of arbitrage* if no arbitrage opportunities exists.

Notice that in Definition 4.4, we do not explicitly impose any conditions on the dynamics of the underlying market price process in order for the model to adhere to the no arbitrage principle. For example, one might expect the price process S to satisfy the martingale identity in Definition 3.5 or some variant thereof. However, as it turns out this is overly restrictive. Before stating the relevant theorem, equivalence of probability measures needs to be defined.

Definition 4.5. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Then a probability measure on $(\Omega, \mathcal{F})$, $\mathbb{Q} : \mathcal{F} \longrightarrow [0, 1]$ is called *equivalent* to $\mathbb{P}$ if

$$\mathbb{P}(A) = 0 \iff \mathbb{Q}(A) = 0$$

for some $A \in \mathcal{F}$. Furthermore, notice that $\mathbb{Q} >> \mathbb{P}$ and $\mathbb{P} >> \mathbb{Q}$ (absolute continuity of measures) as in Theorem 9. If $\mathbb{P}$ and $\mathbb{Q}$ are equivalent we write $\mathbb{Q} \sim \mathbb{P}$.

Definition 4.6. Let $X : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}$ be a random process, then a measure is an *equivalent martingale measure* if

i . $\mathbb{Q} \sim \mathbb{P}$ and

ii . $X_s = \mathbb{E}^{\mathbb{Q}}(X_t | \mathcal{F}_s)$.

Notation. Note that $\mathbb{Q}$ is not necessarily unique since it will depend on S^0, which will be central to the discussion in Subsection 4.3. For now, let $\mathcal{M}$ denote the set of martingale measures equivalent to $\mathbb{P}$, i.e.

$$\mathcal{M} := \left\{ \mathbb{Q} : \mathcal{F} \longrightarrow [0, 1] \,\middle|\, \substack{\mathbb{Q} \sim \mathbb{P} \\ \mathbb{Q} \text{ Martingale measure}} \right\} \tag{7}$$

We now provide a version of the first fundamental theorem of asset pricing in terms of arbitrage and martingales.

Theorem 1. *(First theorem of asset pricing) The model is free of arbitrage if and only if there exists an equivalent martingale measure $\mathbb{Q} \sim \mathbb{P}$ such that S is a $\mathbb{F}-$martingale under $\mathbb{Q}$.*

Proof. See e.g. Föllmer & Schied (2011). □

4.2 Pricing of Contingent Claims

Arbitrage consistent pricing of derivatives is one of the main practical implications of arbitrage theory. In this subsection, only so called *simple* contingent claims with a fixed maturity $T \in \mathcal{T}$ are considered.

Definition 4.7. Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a filtered probability space. Then a *contingent claim* $X : \Omega \longrightarrow \mathbb{R}$ with maturity $T < \infty$ is an $\mathcal{F}_T$-measurable random variable.

Remark. For some applications we need to impose integrability conditions on the claim, we follow Föllmer & Schied (2011) and Delbaen & Schachermayer (2006) and hence restrict ourselves to essentially bounded claims.

Example. In this thesis, we consider *European call options*. Let the claim X be an essentially bounded $\mathcal{F}_T$-measurable real valued random variable, i.e. $X \in L^\infty(\Omega, \mathcal{F}_T, \mathbb{P})$. Furthermore, following standard convention, the call option is considered a claim on the first component in the market S^1. If we represent the claim X by its *payoff / contract*-function $\Phi : L^\infty \times \mathbb{R} \longrightarrow L^\infty(\Omega, R)$ then

$$X = \Phi^K(S_T) = \max\left\{S_T^1 - K, 0\right\} =: (S_T - K)_+ \tag{8}$$

which encodes the fact that a European call option represents the right to purchase the underlying asset at maturity T for the pre-specified strike price K. Therefore, at maturity the contract holder will pocket the difference between the terminal price and the strike $S_T^1 - K$ or face a zero payoff, since the holder will not exercise the option.

Conditions 1. *To produce arbitrage consistent prices, in light of Subsection 4.1, the following conditions must be imposed on the model:*

i . The model for the market S is free of arbitrage.

ii . All simple contingent claims maturing at time T is a claim on the market S and are bounded $\mathcal{F}_T$-measurable random variables.

Definition 4.8. A $\mathbb{F}$-adapted random process $\Pi = (\Pi_t)_{t \leq T}$ is the *price process* for the claim X if $\Pi_T = X$ and is called the *arbitrage free* price process for X if the *extended* market

$$\left(S^0, S^1, \ldots, S^d, \Pi\right)$$

is arbitrage free for all $t \leq T$.

This provides a very natural theorem.

Theorem 2. *Let $\mathcal{M}$ denote the set of martingale measures, see Equation* (7)*. Then Π is the arbitrage free price for X if and only if there exists a martingale measure $\mathbb{Q} \in \mathcal{M}$ such that the extended market in the normalized/discounted price system*

$$\left(\tilde{S}^0, \tilde{S}^1, \ldots, \tilde{S}^d, \tilde{\Pi}\right) \tag{9}$$

is an $\mathbb{F}$-martingale under $\mathbb{Q}$.

Proof. See e.g. Föllmer & Schied (2011). □

Remark. By Definition 4.4 and Equation (4.8) if there exists a portfolio strategy H satisfying

$$\tilde{V}_T \underset{\mathbb{P}-a.s.}{=} \tilde{X} \tag{10}$$

which is called a *replication strategy*, then in order to preserve the no arbitrage condition in the normalized extended market

$$\left(\tilde{S}^0, \tilde{S}^1, \ldots, \tilde{S}^d, \tilde{\Pi}\right) \tag{11}$$

must be free of arbitrage. Therefore, by Theorem 2 Equation (11) needs to be extended to cover the addition of $\tilde{V}$ such that

$$\left(\tilde{S}^0, \tilde{S}^1, \ldots, \tilde{S}^d, \tilde{\Pi}, \tilde{V}\right) \tag{12}$$

is a $\mathbb{F}$-martingale under $\mathbb{Q}$. Therefore the following identity holds

$$\begin{aligned} \tilde{\Pi}_t &= \mathbb{E}^{\mathbb{Q}}\left(\tilde{\Pi}_T \middle| \mathcal{F}_t\right) \\ \tilde{V}_t &= \mathbb{E}^{\mathbb{Q}}\left(\tilde{V}_T \middle| \mathcal{F}_t\right). \end{aligned}$$

But both $\tilde{\Pi}_T = \tilde{X}$ and $\tilde{V}_T = \tilde{X}$ which means that

$$\tilde{V}_t = \tilde{\Pi}_t = \mathbb{E}^{\mathbb{Q}}\left(\tilde{\Pi}_T \middle| \mathcal{F}_t\right). \tag{13}$$

In other words, if a market is free of arbitrage, the price of any contingent claim is the value of a replication strategy that, almost surely, shares the same payoff. Hence one arrives at the following *general pricing formula* for simple contingent claims.

Theorem 3 (General Pricing Formula)**.** *The arbitrage free price process of a contingent claim X with maturity T is given by*

$$\Pi_t = S_t^0 \, \mathbb{E}^{\mathbb{Q}} \left(\frac{X}{S_T^0} \,\middle|\, \mathcal{F}_t \right),$$

where $\mathbb{Q}$ is a equivalent martingale measure for the market S.

Proof. Follows trivially from Equation (13) and can bee seen in e.g. Björk (2009) for the continuous time analogue. □

Remark. In particular if S^0 is a risk-free bond with and let $S_0^0 = 1$ we get

$$S_t^0 = e^{\sum_{k=0}^{t} r_k}$$

where r represents the short rate, Equation (3) becomes the *risk-neutral formula*.

Corollary 1 (Risk Neutral Pricing Formula)**.** *Choosing the numeraire as the risk free bond, the General Pricing Formula in Theorem 3 takes the form*

$$\Pi_t = e^{-\sum_{k=t}^{T} r_k} \, \mathbb{E}^{\mathbb{Q}} \left(X \mid \mathcal{F}_t \right).$$

In order for the above results to be applicable one needs to establish uniqueness conditions for the martingale pricing measure $\mathbb{Q}$.

As shown below, the condition needed to impose for the uniqueness of the martingale measure $\mathbb{Q}$ is called *market completeness*. Consider, as in Subsection 4.2, the nominal market S defined on the filtered probability space $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$. Implicit in this statement is that S is considered in its $\mathbb{P}$ dynamics. The statement in Equation (10) means that that there exists *hedging*/replication strategy H such that the value of the *hedging* portfolio at maturity T of a given contract X, is the same as the terminal payoff X. This intuition can be summarised by the following definition

Definition 4.9. A claim X maturing at time T is called *admissible*, if there exists a self-financing portfolio H such that

$$V_T \underset{\mathbb{P}-a.s.}{=} X. \tag{14}$$

In this case H is called the *hedge* for X or equivalently H can be called the *replication strategy* for X.

Before proceeding with the analysis, an economic intuition for market completeness is provided. The following statement originates from Björk (2009).

Intuition. Let M denote the number of underlying *traded* assets in the market *excluding* the risk free asset. Furthermore, let R denote the number of sources of risk. Generically, the following relations hold

i . Absence of arbitrage is equivalent to $M \leq R$.

ii . Market completeness is equivalent to $M \geq R$.

iii . Market completeness and absence of arbitrage is equivalent to $M = R$.

If a new asset S^{d+1} is being added to the market, one can construct a new hedging strategy for a claim X provided that S^{d+1} is a source of risk. Therefore, completeness requires the number of risky assets, here $d+1$, to be greater than the number of risk-sources. On the other hand, if a market is complete with absence of arbitrage then every new asset in the market that does provide a source of risk will also provide a potential arbitrage opportunity.

The following economic intuition to Definition 4.9 can be constructed; If a market is free of arbitrage $M \leq R$ and there exists at least one claim which is not admissible. Then there must exist some risk factor which cannot be accounted for by the hedging strategy which implies that $M \geq R$ cannot be true. Therefore we arrive at the following definition.

Definition 4.10. A market is *complete* if **every** claim is admissible as in Definition 4.9.

Economically, this means that market completeness is equivalent to; every claim has a unique arbitrage free price, which from our discussion in Subsection 4.2, must imply the uniqueness of the martingale measure $\mathbb{Q}$.

For a fixed martingale measure $\mathbb{Q}$ and if the normalized claim $\tilde{X}$ is integrable. Then, if $\tilde{V} = \tilde{\Pi}$, X can be hedged by V because of Lemma 1 and there exists a unique H^0 given by Equation (6). Therefore, the concept of completeness is equivalent to the existence of a martingale representation given by $\tilde{V}$ in Equation (5). Therefore the following theorem can be used.

Theorem 4 (Jacod & Shiryaev (1998))**.** *Let $\mathcal{M}$ be the set of equivalent martingale measures. Then for any fixed $\mathbb{Q} \in \mathcal{M}$ the following statements are equivalent:*

i . Every martingale M under $\mathbb{Q}$ has dynamics of the form

$$M_t = M_0 + \sum_{i=1}^{d} (H^i \cdot \tilde{S}^i)_t.$$

ii . $\mathbb{Q}$ is an extremal point of $\mathcal{M}$.

Which naturally brings us to the second theorem of asset pricing.

Theorem 5 (Second Theorem of Asset Pricing)**.** *Assume that the market is free of arbitrage and consider a fixed numeraire asset S^0. Then the market is complete if and only if the martingale measure $\mathbb{Q}$, corresponding to S^0 is unique.*

Proof. See e.g. Björk (2009). □

In conclusion, arbitrage pricing corresponds to choosing a numeraire for the market and if the market is complete, this choice induces a unique martingale measure under which pricing is performed. However, is the condition of market completeness a plausible one? If not, then that would imply that some claims exists which carry unhedgeable/intrinsic risk. Furthermore, in order for investors to agree on one unique arbitrage free price, when choosing the numeraire, or discount process, which measure should one choose. These questions are non-trivial and is one of the central concepts of incomplete market hedging, which will be the subject of Subsection 4.3.

4.3 Hedging in Incomplete Markets

In this Subsection, we consider hedging and pricing for *incomplete markets*, which we later will show is a more general market setting and thereby more realistic. Following Theorem 5, if complete markets are transcended, the martingale measure is no longer unique, which makes the objective of pricing more complicated. Hence, we focus on the impact on hedging strategies for the claim. We start by providing a definition of incomplete markets and provide subsequent intuition.

Definition 4.11. A market is called *incomplete* if there there exists at least one claim which is not admissible by a self-financing hedging strategy H on either S or $\tilde{S}$.

Intuition. As seen in Subsection 4.2, a complete market is free of arbitrage if the number of claims is equal to the number of states of the world. Market incompleteness means therefore that the number of claims is less than the number of states. In other words, some claims will remain unhedgeable by any dynamic strategy, since there does not exist claims on some state of nature which acts like a source of risk. Therefore, in incomplete markets, perfect hedging is no longer feasible for any contingent claim since all claims carry *intrinsic* risk. Hence, the task of hedging now becomes risk-minimization, as in Föllmer & Schweizer (1991).

With this intuition in mind, we now provide a formal analysis of the impact on hedging and thereby pricing of contingent claims. Firstly, by Theorem 5 for incomplete markets the equivalent martingale measure is not unique. Hence, we fix a martingale measure $\tilde{\mathbb{Q}}$ induced by the choice of a fixed numeraire S^0. Below we follow the same line of thought as Föllmer & Schweizer (1991) but adapted to suit our notation and applications. As alluded to above, any claim will carry intrinsic risk, therefore it is reasonable to modify the replication condition in Equation (14) such that

$$\tilde{X} \underset{\mathbb{P}-a.s.}{=} \tilde{V}_T + \tilde{Z}_T \tag{15}$$

where $\tilde{Z} : \Omega \times [0,T] \longrightarrow \mathbb{R}$ is a normalized $\mathbb{F}$-martingale under the fixed equivalent martingale measure $\tilde{\mathbb{Q}}$. The essential idea of Föllmer & Schweizer (1991) is that $\tilde{Z}$ is orthogonal to $\tilde{S}$, hence one needs an inner product space. Therefore consider S_T, Z_T, V_T as square integrable $\mathcal{F}_T$-measurable random variables and equip $(L^2(\Omega, \mathcal{F}_T, \tilde{\mathbb{Q}}), +, \cdot)$ with the inner product $\langle X, Y \rangle := \mathbb{E}_{\tilde{\mathbb{Q}}}(XY)$ for all $X, Y \in L^2(\Omega, \mathcal{F}_T, \tilde{\mathbb{Q}})$. Since only self financing strategies H are admitted, the normalized *hedging profit and loss* or *replication error* is

$$\tilde{Z}_T = \tilde{X} - \tilde{V}_0 - \sum_{i=1}^{d} (H^i \cdot \tilde{S}^i)_T. \tag{16}$$

Since any claim will carry intrinsic risk, any hedging strategy will fail to perfectly replicate the payoff of the claim at maturity. Therefore, the task of pricing and hedging is reduced to minimizing a replication error. The question of which loss function to choose for evaluation of the error in Equation (16) was addressed in Buehler et al. (2019) in their development of the deep hedging algorithm.

5 Artificial Neural Networks

This section provides the theoretical analysis of *artificial neural networks (ANN)*. Here we show sufficient theoretical results for the application of ANN´s to the stated objective of the thesis. In light of the intuition gained from the theory introduced in Section 4, we can provide a more precise

definition of the problem addressed in this thesis; Our aim is to construct a data driven model that can both approximate the conditional distribution of asset prices over time and statistically replicate the payoff of options appearing in the market. In particular *recurrent neural networks (RNN)* are utilized to model sequential data and *generative adversarial networks (GAN)* to approximate the distribution of a bounded random process. To that end, this section provides the necessary theory from machine learning in order to justify our method and implement the stated objective of the thesis.

Outline. In Subsection 5.1, we introduce the reader to the most simple type of neural networks, namely that of a *feed forward neural network (FNN)/multilayered perceptron*. We start by providing the definition of a feed forward network through the forward pass of information by concatenation of affine functions. Furthermore, we state the pivotal result of *universal approximation* for feed forward nets, which provides the theoretical justification for the usage of artificial neural networks to approximate functions. Lastly, we conclude the subsection by a short discussion on the process of *training* a neural network through gradient descent and backpropagation of error.

The notion of artificial neural networks can be extended to dynamical systems, thereby defining so called *recurrent neural networks (RNN)* which will be the subject of Subsection 5.2. Combining the result of universal approximation for RNN as proved by Schäfer & Zimmermann (2006) and backpropagation of error RNN´s constitutes natural candidates for the modelling of sequential data. Lastly we provide a toy example of how one can apply recurrent neural nets to model the dependence between two random processes, the aim is to provide the reader with some intuition of their practical applicability.

The concluding Subsection 5.3 defines *generative adversarial networks (GAN)* as tools for modelling distributions of random variables through so called *adversarial learning*. Furthermore, we provide the necessary theoretical results needed for the justification of using GAN´s to estimate the probability distribution of a random variable from given samples. Lastly we provide a slight extension of GAN´s to handle random processes, since we are interested in generating financial time series of prices.

5.1 Feed Forward Neural Networks

Artificial neural networks (ANN) is essentially a class of statistical algorithms, where the computational procedure is inspired by our current conception of biological neural nets and how they learn from sensory data. Biological neural networks, like our brain, learn from interacting with elements of the environment to collect sensory data which we then directly tie to some action to learn how the elements of the environment responds. For example, humans learn how to open bottles by attempting to open the bottle. After enough "training" the "prediction" will finally converge towards an action that is effective in opening the bottle. That is, conceptually, biological neural networks aim to maximise some sort of reward function associated to some action which corresponds to firing different neurons. By choosing an action which maximises the reward, biological neural networks *learns* how to perform a task. Artificial neural networks are conceptually no different and so called *feed forward* neural networks (FNN) formalises this very simple concept. A feed forward net uses interconnected layers of units, called neurons, where the data is being passed from the input layer to the output layer, which represents the predictions. Then, the neural network updates the connections between neurons such that it minimizes the difference between the predicted value and the actual real world value. The following definition is from Buehler et al. (2019).

Definition 5.1. Let $L, N_0, N_1, \cdots, N_L \in \mathbb{N}$ and $\sigma : \mathbb{R} \longrightarrow \mathbb{R}$ be differentiable and let $\left\{(\mathbb{R}^{N_\ell}.+,\cdot)\right\}_{\ell=1,\ldots,L}$ be a finite countable sequence of $\mathbb{R}$-vector spaces. For any $\ell = 1, \ldots L$ let

$$\begin{aligned} W_\ell : \mathbb{R}^{N_{\ell-1}} &\longrightarrow \mathbb{R}^{N_\ell} \\ x &\longmapsto W_\ell(x) := A^\ell \otimes x + b^\ell \end{aligned}$$

be an affine function where $\otimes$ denotes the *Kronecker* product operator/ matrix-vector multiplication and $A^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b^\ell \in \mathbb{R}^{N_\ell}$. The map F is called a *feed forward neural network* (FNN) if

$$\begin{aligned} F : \mathbb{R}^{N_0} &\longrightarrow \mathbb{R}^{N_L} \\ x &\longmapsto F(X) := (W_L \circ F_{L-1} \circ \ldots \circ W_2 \circ F_1)(x) \end{aligned}$$

where $F_\ell = \sigma \circ W_\ell$ for $\ell = 1, \ldots, L-1$.

Terminology. Below we summarise and provide some terminology and intuition for the definition of a feed forward neural network.

i . $L \in \mathbb{N}$ is the number of layers of the network where $1, \ldots, L-1$ are the hidden layers.

ii . $(N_\ell)_{\ell=1\ldots,L-1}$ is a sequence such that $N_\ell \in \mathbb{N}$ denotes the number of neurons for layer ℓ.

iii . (N_0, N_L) denotes the dimensions of the input and output layers respectively.

iv . A^ℓ and b^ℓ are called the *weights* and *biases* for layer ℓ such that $A^\ell_{ij} \in \mathbb{R}$ denotes the weights connected to the map from neuron i of layer $\ell - 1$ to neuron j in layer ℓ.

v . σ is called the *activation* function.

vi . F_ℓ is the activations at layer ℓ.

Hence according to the definition, a feed forward neural network is nothing more than a concatenation of affine functions, weighted by differentiable functions σ. We illustrate the *architecture* also called the *computational graph* of a feed forward neural network in Figure 2.

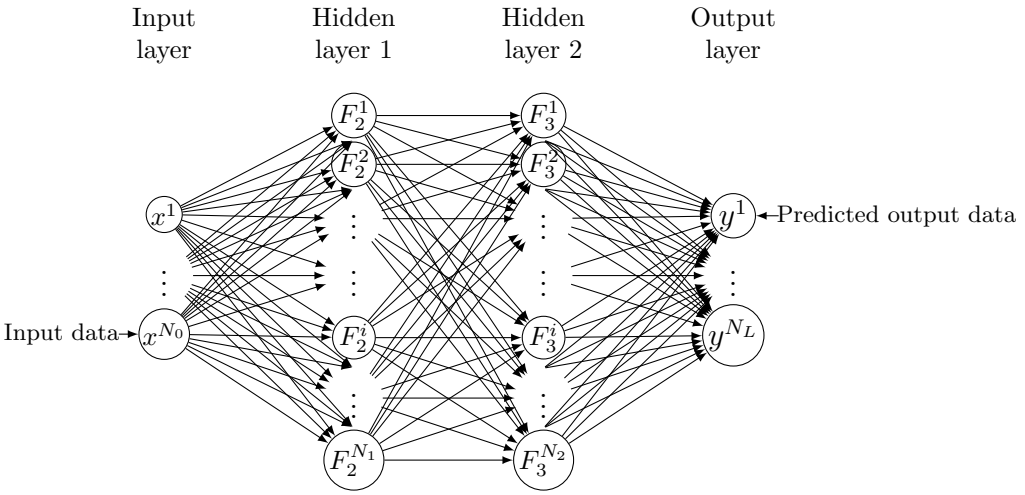

Figure 2: Deep feed forward neural network with 2 hidden layers. The computational graph illustrates a sequence of mathematical operations that are being performed on the object. Each arrow/connection represents the weights and bias being contributed to the rightward neuron. Each activation unit F is a composition of affine functions weighted by a differentiable function σ, see Definition 5.1.

Terminology. A neural network is called *deep*, if it has more than one hidden layer.

As one can see in Figure 2, a deep feed forward neural network has quite a lot of parameters. The result of this is that neural networks are very flexible, therefore they have the potential to approximate functions. As it turns out, deep feed forward neural networks are *universal approximators* under some conditions, meaning that some functions can be approximated arbitrarily well by a deep feed forward network. The original result was proved by Hornik (1991) for continuous real vector valued functions defined on a compact subset of $\mathbb{R}^d$. Note that the thesis does not provide a definition of compactness therefore the reader is referred to Rudin et al. (1964) for definition. Many have extended the so called *universal approximation theorem* in Hornik (1991), to bounded width, arbitrary depth and lebesgue intetrable functions, see Definition A.15. Before stating the extended theorem in Kidger & Lyons (2020), a definition of the space of deep feed forward neural networks is required.

Definition 5.2. Let $\sigma : \mathbb{R} \longrightarrow \mathbb{R}$ be an activation function, $n, m, k \in \mathbb{N}$. Then let $\mathcal{NN}^{\sigma}_{n,m,k}$ represent the class of functions $\mathbb{R}^n \xrightarrow{f} \mathbb{R}^m$,for $f \in \mathcal{NN}^{\sigma}_{n,m,k}$, described by feed forward neural networks with n neurons in the input layer, m neurons in the output layer and an arbitrary number of hidden layers, each with k neuron with activation function σ. Every neuron in the output layer has the identity function as activation.

Theorem 6 (Universal Approximation)**.** *Let $\sigma : \mathbb{R} \longrightarrow \mathbb{R}$ be any non-affine continuous function which is continuously differentiable at at-least one point, with non-zero derivative at that point. Let $K \subseteq \mathbb{R}^n$ be compact. Then $\mathcal{NN}^{\sigma}_{n,m,n+m+2}$ is dense in $\mathcal{C}^0(K, \mathbb{R}^m)$ with respect to the uniform norm.*

Proof. See Kidger & Lyons (2020). □

Intuition. What this essentially means is that any function that is continuously differentiable at some point $x \in K$ can be approximated arbitrarily well by a feed forward neural network, i.e.

$$\forall\, \varepsilon > 0, f \in \mathcal{C}^0(K, \mathbb{R}^m)\ \exists\ F \in \mathcal{NN}^{\sigma}_{n,m,n+m+2} : \sup_{x \in K} |f(x) - F(x)| < \varepsilon.$$

This theorem provides us with a solid mathematical foundation for statistical modelling with deep feed forward neural nets, as long as the random variables that we are modelling maps into some compact subset of $\mathbb{R}^m$.

In order to discuss the concept of *learning* for a neural network, one first needs a so called *loss function* for the network. Consider the universal approximation Theorem 6, the evaluation of the length between the observations, $f(x)$ and the neural network output $F(x)$ is evaluated by the uniform norm, also called supremum norm on a function space. For example, consider the set of continuous functions defined on a compact real subset that maps into $\mathbb{R}^m$, inheriting addition and multiplication point-wise on the target. Then the map

$$\begin{aligned} \|\cdot\|_\infty : \mathcal{C}^0(K, \mathbb{R}^m) &\longrightarrow \mathbb{R} \\ f &\longmapsto \|f\|_\infty := \sup_{x \in K} |f(x)| \end{aligned}$$

is called a *uniform norm* and satisfies the axioms required, see e.g. Rudin et al. (1964). The uniform norm is a prototypical example of a *loss function* that measures the distance between predictions and observations. Hence, the objective of *training* a neural network is to choose a optimal set of parameters that minimizes a given loss function. Learning is achieved through so called *(stochastic) gradient descent* and *backpropagation of error*. Gradient descent describes the simple traditional concept of minimizing a function. In particular, we want to evaluate the gradient with respect to model parameters and step down the gradient until the function reaches local minimum. Hence, the *gradient update step* is simply

$$\theta^{j+1} = \theta^j - \lambda \nabla_\theta \mathcal{J}(\theta^j)$$

and backpropagation is the method used to compute the gradient, for a arbitrary loss function $\mathcal{J}$. For further details see e.g. Goodfellow et al. (2016) ch. 6 and in particular page 213 for a the general backpropagation algorithm.

5.2 Recurrent Neural Networks

In this Subsection, the reader is provided with some intuition as to how the concept of artificial neural network extends to sequential data. We start by introducing the notion of recurrent neural networks through the study of dynamical systems as maps between the hidden states of the network. In particular, we follow the intuition provided in Goodfellow et al. (2016) and characterise the forward pass of a prototypical recurrent neural network as the composition of state transition maps. We then collect the results by a definition and lastly provide an example of how recurrent neural networks can be applied to study dependence structures between random processes.

Recurrent neural networks are able to learn sequential data by incorporating the context of previous known information and states. RNN´s can simply be seen as feed forward neural networks with cyclical connections. These recurrent connections allows the output in each node of the network to be dependent on its previous activation. Consider the dynamical system

$$s^{(t)} = f(s^{(t-1)}; \theta) \tag{17}$$

where $t \in \mathbb{N}$ is time and $s : \mathbb{N} \longrightarrow \mathbb{R}$ is called the *state* of the system. Furthermore, f is called the *state-transition map*, parameterized by an arbitrary parameter θ. Because the state of the system is only a function of the previous state, Equation (17) can be rewritten as

$$s^{(t)} = (f \circ f \circ \ldots \circ f)(s^{(1)}; \theta). \tag{18}$$

The expansion in Equation (18) highlights two important factors; First, the state of the system is a consecutive application of the state transition map from the initial state. Second, the parameter vector is the same for each state, so the state transition map is the same for each state. The representation in Equation (18) is called *unrolling* of the dynamical system and provides a simple characterisation of recurrent neural networks, for further details on unrolling see e.g. Goodfellow et al. (2016). Consider now the additional dependence of the state on some signal $x : \mathbb{N} \longrightarrow \mathbb{R}$, then one obtains the dynamical system

$$s^{(t)} = f(s^{(t-1)}, x^{(t)}; \theta). \tag{19}$$

If one lets s represent the *hidden* states of a neural network one can view Equation (19) as a recurrent neural network with no output layer Goodfellow et al. (2016). Furthermore, if one allows for cyclical connections between states, the hidden state in a arbitrary layer of an RNN is defined by the functional mapping

$$(h_{t-1}^{\ell}, x_t) \longmapsto h_t^{\ell} = f(h_{t-1}^{\ell}, x_t; \theta)$$

where $(h_t^{\ell})_{t\in\mathbb{N}}^{\ell=1,\ldots L_{N-1}}$, $h_t^{\ell} \in \mathbb{R}$ is the hidden state at time t and layer ℓ of the network. We now collect this result and provide the set of all recurrent neural networks of a fixed sequence length.

Definition 5.3. Let $\mathcal{T} \subset \mathbb{N}$ be a finite index set. Furthermore, let $x : \mathcal{T} \longrightarrow \mathbb{R}^d$ and $y : \mathcal{T} \longrightarrow \mathbb{R}^m$ be sequences. A prototypical (sequence to sequence) recurrent neural network is the functional mapping $x \longmapsto \hat{y}$ with a forward pass defined by the system

$$\begin{aligned} a_t^1 &= A^1 h_{t-1}^1 + b^1 + U^1 x_t \\ a_t^{\ell} &= A^{\ell} h_{t-1}^{\ell} + b^{\ell} \\ h_t^{\ell} &= \alpha(a_t^{\ell}) \\ h_t^{L_N} &= c^{L_{N-1}} + B h_{t-1}^{L_N} \\ \hat{y}_t &= \beta(h_t^{L_N}) \end{aligned} \tag{20}$$

where $\alpha, \beta : \mathbb{R} \longrightarrow \mathbb{R}$ are activation functions, $\ell = 2, \ldots, L_{N-1}$ denotes the layer and $t \in \mathcal{T}$ the sequential order. Fix a loss function $\mathcal{J} : \ell^{\infty}(\mathcal{T}) \times \ell^{\infty}(\mathcal{T}) \longrightarrow \mathbb{R}$ to evaluate the error of the predictions based on labelled examples y, where $\ell^{\infty}(\mathcal{T})$ are the bounded sequences with domain $\mathcal{T}$. The parameter space Θ is the collection of components for the respective matrices in Equation (20). The set of recurrent neural network is defined by its forward pass on bounded sequences

$$\mathcal{RNN}_{m,d,\mathcal{T}} := \left\{ F : \ell^{\infty}(\mathcal{T}) \times \Theta \longrightarrow \ell^{\infty}(\mathcal{T}) |\, F_t = (\beta \circ h_t^{L_N} \circ \ldots \circ \alpha \circ a_t^1)(x_t; \theta), F \in \mathcal{C}^0(\theta) \right\}.$$

Training of recurrent neural networks is no different than that of feed forward neural networks covered in Subsection 5.1. One seeks to find a optimal parameter $\theta \in \Theta$ of a given recurrent neural network by minimizing the loss function $\mathcal{J}$ trough gradient descent. In particular, the gradient $\nabla_\theta \mathcal{J}$ is computed by the backpropagation algorithm, which in essence recursively (starting from loss) applies the chain-rule for derivatives. For further details on backpropagation and

recurrent neural networks in general, see Goodfellow et al. (2016) page 213 for the general backpropagation algorithm. As we discussed in Subsection 5.1, the concept of *learning* for a artificial neural network relies on the theoretical results of universal approximation. Luckily, RNN´s also have this property, see Schäfer & Zimmermann (2006) for proof.

In practice, other types of recurrent neural network architectures are applied in order to be trained more efficiently and/or have memorizing properties. Arguably, the two most popular architectures are recurrent neural networks with *long-short-term-memory (LSTM)* layers and *gated-recurrent-units (GRU)*. The reader is referred to Goodfellow et al. (2016) for further details. We now conclude the subsection by providing a theoretical example of how to apply recurrent neural networks to learn the dependence structure between two processes.

Application. Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P})$ be a probability space and take $\mathcal{T} \subset \mathbb{N}$ to be a finite set. Consider two bounded random processes $Y, X \in L^\infty(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P})$ and suppose $Y = f(X)$ for some deterministic function f. Suppose that, theoretically, the parameters θ are learnable by backpropagation. Using the universal approximation results for recurrent neural networks in Schäfer & Zimmermann (2006), one can train a recurrent neural network to learn the dependence structure between X and Y using labelled data. That is, one can approximate the functional mapping $X \longmapsto f(X)$, using labelled examples of Y by a recurrent neural network. For example, consider the forward pass

$$\begin{aligned} h_t &= f_h(h_{t-1}, X_t(\omega); \theta) \\ \hat{Y}_t &= f_o(h_t) \qquad \forall\, t \in \mathcal{T}. \end{aligned}$$

where $f_h : \mathbb{R} \times \mathbb{R}^d \longleftarrow \mathbb{R}$ is a state transition map and $f_o : \mathbb{R} \longrightarrow \mathbb{R}$ is some output function, producing the predictions of the RNN. Now choose a loss function $\mathcal{J} : L^\infty(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P}) \times L^\infty(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{P}) \longrightarrow \mathbb{R}$ encoding the distance between $\hat{Y}(\Omega \times \mathcal{T})$ and $Y(\Omega \times \mathcal{T})$ and minimize $\mathcal{J}(\hat{Y}, Y; \theta)$ by gradient descent and backpropagation. Suppose a arbitrarily small error is reached $\varepsilon = \mathcal{J}(\hat{Y}_t(\omega), Y_t(\omega); \hat{\theta}) \underset{\mathbb{P}-a.s.}{>} 0$, then $\hat{Y}$ converges almost surely to Y, as measured by the distance function $\mathcal{J}$ and thereby implicitly learned the dependence structure given by $f(X)$.

5.3 Generative Adversarial Networks

In this Subsection we develop the theoretical technology of distributional approximation for random variables and processes, using machine learning. Recall that, embedded in the objective of the thesis, is the model independent asset simulation to extend the deep hedging algorithm from Buehler et al. (2019). Hence the thesis objective requires the ability of reconstructing probability distributions of asset prices using deep learning, which is the subject of this subsection.

Generative adversarial networks (GAN) are a class of deep learning techniques which aim to model high dimensional distributions of random variables and as we shall see later, random processes. The technique was proposed by Goodfellow et al. (2014) and can be characterised by so called *adversarial* learning, which is a game theoretic approach to learning. Conceptually, one can think of a prototypical GAN as two separate neural networks, one is called a *generator* $\mathcal{G}$ and the other is called a *discriminator* $\mathcal{D}$ trained in the same loop with adversarial objectives. The objective of the generator is to generate samples that "fools" the discriminator and the discriminator objective is to detect weather or not the generated samples are real or fake. Assuming the GAN converges, the loss of the discriminator grows until the discriminator can only produce random guesses of weather or not the generated samples are real or fake, constituting

aa nash-equilibrium. If this is the case, we have learned to sample from the, potentially high dimensional, marginal distribution of a random variable. In the case of convergence, the generator *implicitly* learns the distribution of the data through its interaction with the discriminator. The discriminator has access to both real and fake samples, without knowing *a priori* which are real and which are fake, but is trained on real samples Creswell et al. (2018).

In the following construction we fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and consider bounded random variables $X, Z \in L^\infty(\Omega, \mathcal{F}, \mathbb{P})$. Furthermore, we take $(\mathbb{R}^d, \mathcal{B})$ to be a Borel measurable space, see Definition A.5 and let $X_*\mathbb{P}$ be the distribution measure of the $(\mathcal{F}, \mathcal{B})$-measurable random variable $X : \Omega \longrightarrow \mathbb{R}^d$. See Definitions A.9 and A.10 for further details regarding distributions and random variables.

We start by providing the definition of a *generator*.

Definition 5.4 (Generator)**.** Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $Z \in L^\infty(\Omega, \mathcal{F}, \mathbb{P})$. Then the map

$$\begin{aligned} \mathcal{G} : L^\infty \times \Theta^g &\longrightarrow L^\infty \\ (Z, \theta^g) &\longmapsto \mathcal{G}(Z, \theta^g) \end{aligned}$$

is called a *generator* if $\mathcal{G}$ is a artificial neural network, differentiable w.r.t θ^g.

Remark. The reason we restrict the model to bounded random variables is that it should be computable on a Turing machine, which cannot assign values to infinity. Furthermore, the generator $\mathcal{G}$ is defined to act on $\omega \in \Omega$ *linearly* in the first slot

$$\mathcal{G}(Z, \theta^g)(\omega) := \mathcal{G}(Z(\omega), \theta^g) \in \mathbb{R}^d$$

Definition 5.5. The map

$$\begin{aligned} X_*\mathbb{Q} : \Theta^g \times \mathcal{B} &\longrightarrow [0, 1] \\ (\theta, B) &\longmapsto X_*\mathbb{Q}_{\theta^g}(B) \end{aligned}$$

is referred to as the *Model distribution measure* of the random variable and is presumed to satisfy the axioms in Definition A.6 for a fixed $\theta^g \in \Theta^g$.

In accordance with, Roth et al. (2017), the essential idea of generative adversarial networks is to pair a θ^g parameterized generator that produces the distribution $X_*\mathbb{Q}_{\theta^g}$ for $X(\omega) = \mathcal{G}(Z(\omega), \theta^g)$ with a *discriminator* which aims to distinguish between $X_*\mathbb{P}$ and $X_*\mathbb{Q}_{\theta^g}$, whereas the generator tries to make $X_*\mathbb{Q}_{\theta^g} \sim X_*\mathbb{P}$.

Definition 5.6 (Discriminator Roth et al. (2017))**.** A *discriminator* is a class of objective functions $\hat{\mathcal{F}}$ that measures the dissimilarity of pairs of distributions.

Finally, one can define a generative adversarial network (GAN).

Definition 5.7 (Generative Adversarial Network)**.** Let X be a random variable and $\mathcal{G}$ be a generator producing the model distribution $X_*\mathbb{Q}_{\theta^g}$ from samples of a bounded random variable Z. Furthermore, let $\hat{\mathcal{F}}$ be a discriminator. Then a *generative adversarial network* is the pair $(\mathcal{G}, \mathcal{D})$ with the objective function

$$\inf_{\theta^g \in \Theta^g} \left[\sup_{F \in \hat{\mathcal{F}}} F(X_*\mathbb{P}, X_*\mathbb{Q}_{\theta^g}) \right]. \tag{21}$$

where the loss is thus defined by

$$\ell(X_*\mathbb{Q}_{\theta^g}; \hat{\mathcal{F}}) := \sup_{F \in \hat{\mathcal{F}}} F(X_*\mathbb{P}, X_*\mathbb{Q}_{\theta^g})$$

and $F \in \hat{\mathcal{F}}$ is represented as the integrals

$$F(X_*\mathbb{P}, X_*\mathbb{Q}_{\theta^g}) = \mathbb{E}_{\mathbb{P}}[\log \mathcal{D}(X)] + \mathbb{E}_{\mathbb{Q}_{\theta^g}}\left[\log(1 - \mathcal{D}(\mathcal{G}(Z, \theta^q)))\right]. \tag{22}$$

In order to justify the objective function in Equation (21), one first need a general measure of divergence for distributions.

Definition 5.8 (f-divergence)**.** Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Furthermore let $\mathbb{Q} : \mathcal{F} \longrightarrow [0, 1]$ be a probability measure such that $\mathbb{P}$ is absolutely continuous with respect to $\mathbb{Q}$. Let also $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a convex function satisfying $f(1) = 0$. Then the *f-divergence* between $\mathbb{P}$ and $\mathbb{Q}$ is the real number

$$D_f(\mathbb{P}||\mathbb{Q}) := \mathbb{E}_{\mathbb{Q}}\left(f \circ \frac{d\mathbb{P}}{d\mathbb{Q}}\right)$$

where $\frac{d\mathbb{P}}{d\mathbb{Q}}$ is the *Radon-Nikodym derivative* of $\mathbb{P}$ with respect to $\mathbb{Q}$, see the Radon-Nikodym Theorem 9 for further details.

Using the empirical results collected in Nowozin et al. (2016), one can infer that any $f-$divergence measure can be used for training a generative adversarial network. However, no theoretical results have yet been obtained for proving that generative adversarial networks actually minimizes $f-$divergences. However, the empirical observations in Nowozin et al. (2016) indicates that this is the case. We collect this theoretically important result in the form of a empirical observation.

Empirical Observation 1 (Nowozin et al. (2016))**.** *Generative neural samplers minimizes the $f-$divergence between the real distribution and the model distribution.*

The above result provides a empirical observation indicating the theoretical justification for using GAN´s to learn the distribution of a random variable. However, in this thesis, we aim to approximate conditional distributions for random processes and hence need to extend the definition. Conceptually, only countable sequences of random variables needs to be considered in order to extend the definition.

Definition 5.9 (Generator)**.** Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a filtered probability space and let $Z \in L^\infty(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a $\mathbb{F}$-adapted random process. In particular, let $\mathbb{F} = \mathbb{F}^Z$ be the generated filtration and $\mathcal{T} \subset \mathbb{N}_0$ to be countably finite. Then the map

$$\begin{aligned} \mathcal{G} : L^\infty(\Omega \times \mathcal{T}) \times \Theta^g &\longrightarrow L^\infty(\Omega \times \mathcal{T}) \\ (Z, \theta^g) &\longmapsto \mathcal{G}(Z, \theta^g) \end{aligned}$$

is called a *generator* if $\mathcal{G}$ is a artificial neural network, differentiable w.r.t θ^g.

Remark. As in Definition 5.4, let $\mathcal{G}$ act linearly in the first slot on $\omega \in \Omega$ and $t \in \mathcal{T}$ to obtain function values of generated processes

$$(\omega, t) \longmapsto \mathcal{G}(Z, \theta^g)(\omega, t) := \mathcal{G}(Z_t(\omega), \theta^g) \in \mathbb{R}^d.$$

In a similar fashion one could extend the definition of a discriminator as a class of functions minimizing the dissimilarity for pairs of *joint* distributions of a random process under different probability measures. Naturally, we can reuse the objective in Equation (21) to characterize

a generative adversarial network for random processes and therefore recover the entire definition.

In practice however learning the joint distribution directly is a difficult objective function in the standard GAN framework. Therefore, for performance reasons, the consideration of alternative computational graphs is warranted. In this thesis the *TimeGAN* algorithm proposed by Yoon et al. (2019) is used to generate financial time-series, which utilises auxiliary maps to reconstruct the joint distribution of a random process from its lower dimensional latent space representation.

6 Methodology

In this section we utilise all previous developed theory in order to justify the proposed extension of the deep hedging model. To that end, we use generative adversarial networks, see Subsection 5.3, to approximate the (conditional) $\mathbb{P}$-distribution of market prices S, through the *TimeGAN* algorithm proposed by Yoon et al. (2019).

6.1 Problem Formulation & Proposed Approach

In this Subsection we provide a formal statement for the problem we seek to address and detail the proposed solution. Before proceeding, the model conditions and environment has to be stated.

Conditions 2. *Let $(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a discrete time filtered probability space and consider the market price process $S : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^{d+1}$ being a bounded discrete time random process. We fix S^0 as the numeraire asset, which for simplicity, is assumed to be deterministic and $S^0 = 1$ for all $(\omega, t) \in \Omega \times \mathcal{T}$. Furthermore, let $\mathbb{F} = \mathbb{F}^W$ for $W : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^d$ denote a Brownian filtration and motion respectively. We consider S in a arbitrage, incomplete and discrete market setting, hence the market provides the equivalent martingale measure $\mathbb{Q} \in \mathcal{M}$, see Definition 4.6 for further details. Furthermore, let $T \in \mathcal{T}$ and $K \in \mathbb{R}$ be a fixed maturity and strike respectively. Assume further that there exists a European option $X \in L^\infty(\Omega, \mathcal{F}_T, \mathbb{P})$ with maturity and strike at T, K and that X is a claim on S^1 that is liquidly traded.*

Recall from Section 5 and in particular Subsection 5.3 that a generative adversarial network is the pair $(\mathcal{G}, \mathcal{D})$ in which

$$\begin{aligned} \mathcal{G} : L^\infty(\Omega \times \mathcal{T}) \times \Theta^{\mathcal{G}} &\longrightarrow L^\infty(\Omega \times \mathcal{T}) \\ (W, \theta^{\mathcal{G}}) &\longmapsto \mathcal{G}(W, \theta^{\mathcal{G}}) \end{aligned}$$

is a generator that is defined linearly on $\Omega \times \mathcal{T}$ in the first slot. GANs have been empirically observed to be minimizers of the $f-$divergence, as per Empirical Observation 1, defined as

$$D_f(\hat{\mathbb{P}} || \mathbb{P}) := \mathbb{E}_{\mathbb{P}} \left(f \circ \frac{d\hat{\mathbb{P}}}{d\mathbb{P}} \right).$$

In addition, let $\mathcal{D}$ be a discriminator, represented as a neural network

$$\begin{aligned} \mathcal{D} : L^\infty(\Omega \times \mathcal{T}) \times \Theta^{\mathcal{D}} &\longrightarrow [0, 1] \\ (\tilde{S}, \theta^{\mathcal{D}}) &\longmapsto \mathcal{D}(\tilde{S}, \theta^{\mathcal{D}}) \end{aligned}$$

which is also linear in the first argument on $\Omega \times \mathcal{T}$. Let the generative adversarial network (GAN) be be trained with the adversarial objective

$$\inf_{\theta^{\mathcal{G}} \in \Theta^{\mathcal{G}}} \sup_{\theta^{\mathcal{D}} \in \Theta^{\mathcal{D}}} F(S_* \hat{\mathbb{P}}, S_* \mathbb{P}) \tag{23}$$

where $\hat{\mathbb{P}}$ is a model probability measure, inducing a model distribution $S_* \hat{\mathbb{P}}$ as per Definition 5.5. F measures the dissimilarity between the model distribution and the real distribution $S_* \hat{\mathbb{P}}, S_* \mathbb{P}$ and is defined by

$$F(S_* \hat{\mathbb{P}}, S_* \mathbb{P}) := \mathbb{E}_{\mathbb{P}} \left(\sum_{t \in \mathcal{T}} \log \mathcal{D}(\tilde{S})_t \right) + \mathbb{E}_{\hat{\mathbb{P}}} \left(\sum_{t \in \mathcal{T}} \log(1 - (\mathcal{D} \circ \mathcal{G})(W)_t) \right).$$

Assuming convergence of the generative adversarial network (GAN), the random process $\hat{S} = \mathcal{G}(W, \theta^{\mathcal{G}})$ converges to the approximated $\mathbb{P}$-dynamics of the process S. However the adversarial objective in Equation (23) might be unattainable for complex dependency structures. Therefore, in practice, one often consider variations on the GAN algorithm, see e.g. Wiese et al. (2020) and Ni et al. (2020). To that end, we use the *TimeGAN* model proposed by Yoon et al. (2019), for a detailed description see Appendix B.

Recall from Section 4 that arbitrage can be characterised in terms of the existence of a self-financing portfolio $H : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^{d+1}$ with wealth/value process $V : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}$ satisfying

$$V_T = V_0 + \sum_{i=1}^{d} (H^i \cdot S^i)_T$$

since $S^0 = 1$ and thus the normalized/discounted wealth process $\tilde{V} = V$ since H^0 is uniquely given, see Section 4 for further details. Furthermore, $(H^i \cdot S^i)_T$ is the discrete stochastic integral

$$(H^i \cdot S^i)_T := \sum_{k=1}^{n-1} H^i_k (S^i_{k+1} - S^i_k), \qquad i = 1, \dots, d, \; T = n$$

from Definition 3.12. Recall further that H is called a *hedging/replication* strategy for a given claim $X \in L^\infty(\Omega, \mathcal{F}_T, \mathbb{P})$ if the terminal replication condition is satisfied

$$X \underset{\mathbb{P}-a.s.}{=} V_T + Z_T \tag{24}$$

where $Z : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}$ is a martingale that is orthogonal to S and thus V in the Hilbert space $(L^2(\Omega \times \mathcal{T}, \mathcal{F}, \mathbb{F}, \mathbb{P}), +, \cdot, \langle \cdot, \cdot \rangle_2)$. Hence, Z represents a unhedgeable/intrinsic risk associated to the claim X. Recall further that V is a martingale under $\mathbb{Q}$ as per Theorem 2. Hence the arbitrage condition restricts self-financing portfolio wealth processes V, satisfying the terminal replication constraint in Equation (24) to the be bounded by the interval

$$V_t = \Pi_t \in \left[\inf_{\mathbb{Q} \in \mathcal{M}} \mathbb{E}_{\mathbb{Q}} \left(X | \mathcal{F}_t \right), \sup_{\mathbb{Q} \in \mathcal{M}} \mathbb{E}_{\mathbb{Q}} \left(X | \mathcal{F}_t \right) \right]. \tag{25}$$

where $\mathcal{M}$ is the set of martingale measures, see Definition 4.6, Equation (7) and Delbaen & Schachermayer (2006) Theorem 2.4.1. Note that in a complete market the interval in Equation (25) reduces to the singleton set given by the general pricing formula in Theorem 3.

Using the terminal replication constraint in Equation (24), one can easily characterise the terminal profit-loss of the a hedged position as

$$Z_T \underset{\mathbb{P}-a.s.}{=} X - V_0 - \sum_{i=1}^{d} (H^i \cdot S^i)_T. \tag{26}$$

Similarly to Schweizer (1995), Buehler et al. (2019) now formulates the hedging problem by the constrained optimization

$$\Psi(X) = \inf_{H \in \mathcal{H}} \rho \left(X - V_0 - \sum_{i=1}^{d} (H^i \cdot S^i)_T \right) \tag{27}$$

where $\mathcal{H}$ is the set of constrained hedging strategies for X

$$\mathcal{H} := \left\{ H : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^d \,\middle|\, X \underset{\mathbb{P}-a.s.}{=} V_T + Z_T \right\}$$

and ρ is a *convex risk measure*, see Definition 1 in Buehler et al. (2019) for details. However, as previously discussed, quadratic loss functions as defined in Föllmer & Schweizer (1991) are also reasonable to consider. In essence, the optimization aims to minimize the length of the replication error vector, to that end, let S, X and Π be square integrable and define the L^2 norm as the map

$$\begin{aligned} \|\cdot\|_2 : L^2 &\longrightarrow \mathbb{R} \\ Y &\longmapsto \|Y\|_2 := \langle Y, Y \rangle^{\frac{1}{2}} \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is the (sesquilinear) inner product on L^2. Then since Z is orthogonal to S, i.e. $\langle Z, S \rangle = 0$ the Hilbert space optimization

$$\Psi(X) = \inf_{H \in \mathcal{H}} \left\| X - V_0 - \sum_{i=1}^{d} (H^i \cdot S^i)_T \right\|_2$$

arises naturally if one lets $V_0 = \Pi_0$ where Π represents the current market price of the claim X. Furthermore, as discussed above, the TimeGAN architecture proposed by Yoon et al. (2019) can be used to approximate the random process S and thus approximates the payoff X by a claim $\hat{X} \in L^2(\Omega \times \mathcal{T}, \mathcal{F}_T, \hat{\mathbb{P}})$ on $\hat{S}^1$ where $\hat{\mathbb{P}}$ is the approximated measure induced by the generative adversarial network and $\hat{S} := \mathcal{G}(W, \theta)$. Since this thesis focuses on the pricing/hedging of call options, $\hat{X}$ is defined as

$$\hat{X} := \left(\hat{S}^1_T - K \right)_+ .$$

Thus the L^2 minimization in Equation (6.1) needs to be updated to explicitly take into account the objective of the thesis

$$\hat{\Psi}(X) = \inf_{H \in \mathcal{H}} \left\| \hat{X} - \Pi_0 - \sum_{i=1}^{d} (H^i \cdot \hat{S}^i)_T \right\|_2 .$$

Using the theory developed in Buehler et al. (2019), the formulation of unconstrained hedging strategies as outputs of neural networks allows for the explicit application of the universal approximation Theorem 6. However as discussed in Subsection 5.2, Schäfer & Zimmermann

(2006) have shown RNN´s to also have this property. Hence the set of recurrent neural network strategies can be defined as the set

$$\begin{aligned}\mathcal{H}_M &= \left\{ H : \Omega \times \mathcal{T} \longrightarrow \mathbb{R}^d \,\middle|\, H_t(\omega) = \underset{F \in \mathcal{RNN}^{\Sigma}_{M,d,d,\mathcal{T}}}{F(H_{t-1}(\omega), \hat{S}_t(\omega))} \right\} \\ &= \left\{ H : \Omega \times \mathcal{T} \times \Theta_M \longrightarrow \mathbb{R}^d \,\middle|\, H_t^\theta(\omega) = F^\theta(H_{t-1}(\omega), \hat{S}_t(\omega)) \right\}\end{aligned} \tag{28}$$

where M denotes the depth of the network, d the input and output dimensions, Θ_M the set of weights and biases for the recurrent neural network and Σ denotes a set of activation functions.Hence, using Lemma 1 in Buehler et al. (2019) one can reduce the infinite dimensional problem of finding optimal hedging strategies to the finite dimensional problem of finding optimal parameters for a recurrent neural network since

$$\begin{aligned}\hat{\Psi}^M(X) &= \inf_{H \in \mathcal{H}_M} \left\| \hat{X} - \Pi_0 - \sum_{i=1}^{d} (H^i \cdot S^i)_T \right\|_2 \\ &= \inf_{\theta \in \Theta_M} \left\| \hat{X} - \Pi_0 - \sum_{i=1}^{d} (H_\theta^i \cdot \hat{S}^i)_T \right\|_2\end{aligned} \tag{29}$$

for a fixed GAN parameterisation. However, in practice, since the gradient outside of the objective in Equation (29) vanishes, one may consider regularizing the distance from a target strategy according to Buehler et al. (2019)

$$\hat{\Psi}^M(X) = \inf_{\theta \in \Theta_M} \left\| \hat{X} - \Pi_0 - \sum_{i=1}^{d} (H_\theta^i \cdot \hat{S}^i)_T \right\|_2 + \gamma \left\| H_\theta - \delta \right\|_2 .$$

Note that the theory is developed for convex risk measures and we do not mathematically prove that $\Psi^M \to \Psi$ as $M \to \infty$. However since we can confirm convergence using experiments in Section 7, convergence is assumed.

7 Experimental Results

In this Section we provide some experimental results in a controlled environment using simulation. In particular, we consider the approximation of asset price dynamics in the famous *Black-Scholes* model introduced in Black & Scholes (1973) as an illustrative example. This means that we simulate sample paths of a *geometric Brownian motion (GBM)*, which are then feed to the TimeGAN algorithm to approximate its conditional distributions. Then, using the TimeGAN approximation of the $\mathbb{P}$-dynamics of the GBM and the Black-Scholes prices of options, we use the deep hedging algorithm to approximate hedging strategies.

Setting & Implementation. As indicated above, we consider the performance of the approach in a controlled experiment using the Black-Scholes model in addition to the constraints set by Conditions 2. Hence, a two asset market is defined and described in their $\mathbb{P}$-dynamics by the processes

$$\begin{aligned}S_t^0 &= e^{rt} \\ S_t^1 &= S_0^1 e^{\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W_t}\end{aligned} \tag{30}$$

which can be equivalently expressed in its instantaneous return form

$$\begin{aligned} \frac{dS_t^0}{S_t} &= rdt \\ \frac{dS_t^1}{S_t^1} &= \mu dt + \sigma dW_t \end{aligned} \tag{31}$$

using Itô's lemma, see e.g. Protter (2005) for further details regarding stochastic calculus. See Figure 3 for visualisation of sample path for a geometric Brownian motion. In practice however,

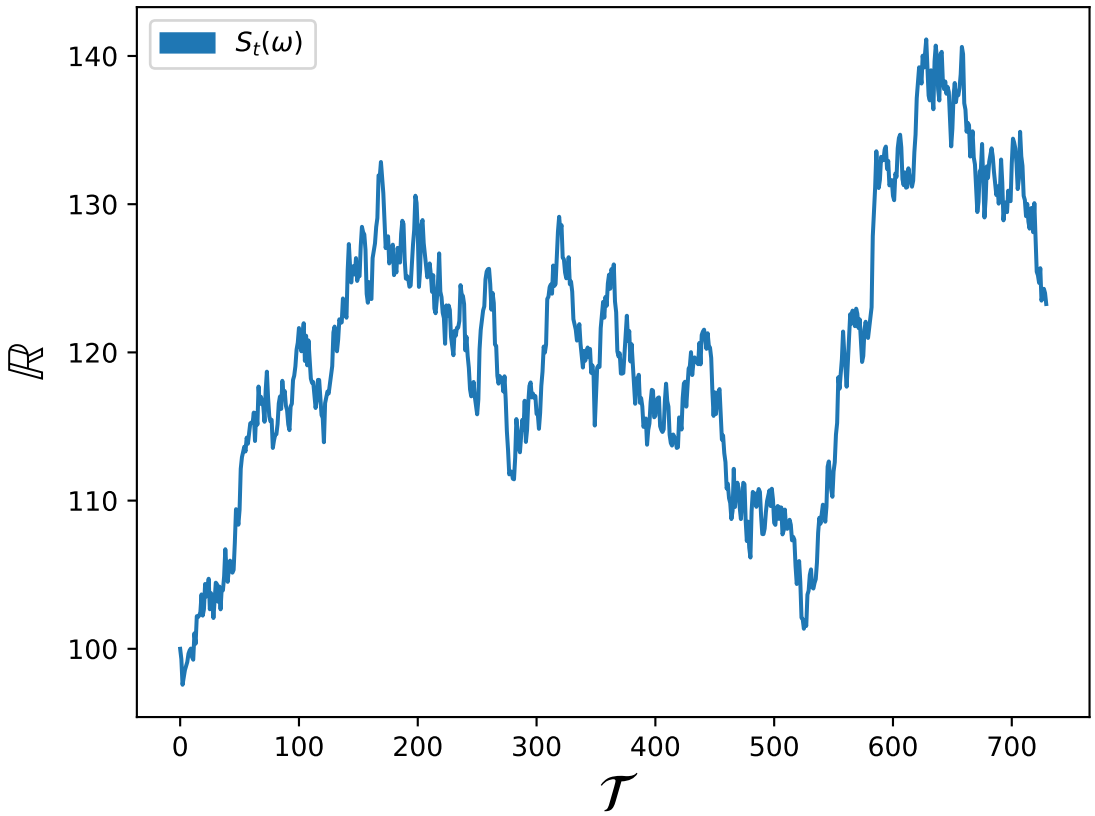

Figure 3: Function value of geometric Brownian motion for fixed $\omega \in \Omega$. In the illustration we use drift parameter $\mu = 0.1$ and volatility $\sigma = 0.2$ over two years of daily price quotes, i.e. 730 days. The discretization scheme used is the so called *Euler-Maruyama* scheme.

we actually consider a *Euler-Maruyama* discretization of Equation (30) because our market setting constitutes daily close prices. Following the theory developed earlier in Section 4 the price system in Equation (30) is normalized to units of the numeraire, here represented as a risk

free credit in a dollar note, to obtain discounted prices

$$\begin{aligned}\tilde{S}_t^0 &= 1 \\ \tilde{S}_t^1 &= S_0^1 \exp\left(\left(\mu - r - \frac{1}{2}\sigma^2\right)t + \sigma W_t\right)\end{aligned} \tag{32}$$

see e.g. Delbaen & Schachermayer (2006) for further details.

Notation. From now on let $S^0 = B$ and $S^1 = S$ in order to adhere to the notation developed in Delbaen & Schachermayer (2006) and Björk (2009) and to prevent clutter. Also, $\nu = \mu - r$ is let to denote the excess return on S.

Note that the process $\tilde{S}$ does not satisfy the martingale inequality in Definition 3.5 under $\mathbb{P}$. The market is complete since filtration $\mathbb{F}$ is only generated by a one dimensional Brownian motion and three, including the European call option, traded assets. Assuming the market is free of arbitrage then, by the second theorem of asset pricing 5, there exists a unique equivalent martingale measure $\mathbb{Q}$ for each choice of numeraire S^0 such that S is a martingale under $\mathbb{Q}$. Using the Girsanov Theorem, see e.g. Musiela & Rutkowski (2005), one can then deduce that the $\mathbb{Q}$-dynamics of S is given by

$$\begin{aligned}S_t &= S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)t + \sigma W_t^{\mathbb{Q}}\right) \\ W_t^{\mathbb{Q}} &= \frac{r-\mu}{\sigma}t + W_t\end{aligned} \tag{33}$$

where $W^{\mathbb{Q}}$ is a Brownian motion under $\mathbb{Q}$. Note that expressing S in Equation (33) in units of the numeraire asset yields

$$\tilde{S}_t := \frac{S_t^1}{S_t^0} = S_0 \exp\left(\left(-\frac{1}{2}\sigma^2\right)t + \sigma W_t^{\mathbb{Q}}\right)$$

which is a martingale under $\mathbb{Q}$ since $\mathbb{E}_{\mathbb{Q}}(S_T|\mathcal{F}_0^w) = S_0 e^{rT}$. Most notably, by applying Corollary 1 on a call option $X = (S_T - K)_+$ one can solve for the famous *Black-Scholes pricing formula*.

Theorem 7 (Black-Scholes Pricing Formula)**.** *Let* Π *be the price process of a European call option* X*. Then the Black-Scholes price of* X *is given by*

$$\Pi_0^{BS} = S_0\Phi\left(\frac{\log\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}\right) - Ke^{-rT}\Phi\left(\frac{\log\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}\right). \tag{34}$$

where Φ *is the distribution function of a standard normal random variable.*

Proof. See Delbaen & Schachermayer (2006) or for an alternative proof using partial differential equations Black & Scholes (1973). □

Our objective in this controlled simulation environment is to statistically estimate the $\mathbb{P}$-dynamics, i.e. Equation (32), using generative adversarial networks, then hedging and pricing the call option in Equation (34) for a discrete grid of strikes and maturities using deep hedging. Hence, the following metrics can be used in order to evaluate model performance:

- **Distribution approximation**. We evaluate the distributional performance of asset prices by comparing the model distribution of the underlying asset to the empirical distribution.

Fortunately, in the Black-Scholes model,using Equation (31) one can easily deduce that $\frac{dS}{S}$ is a normally distributed and stationary process with expected value and variance given by

$$\begin{aligned} \mathbb{E}_{\mathbb{P}}\left(\frac{dS_t}{S_t}\right) &= \mu dt \\ \text{Var}\left(\frac{dS_t}{S_t}\right) &= \sigma dt \end{aligned} \tag{35}$$

for all t. Therefore we test the divergence of the model distribution $\hat{\mathbb{P}}$ from the target $\mathbb{P}$-distribution given by $\mathcal{N}(\mu dt, \sigma dt)$ through the *Kolmogorov-Smirnov test* where $\mu = 0.1$ and $\sigma = 0.2$ represents drift and volatility of S respectively.

- **Hedging.** In the Black-Scholes model, hedging has analytical techniques, the proportion of stock to hold is given by the derivative of the option with respect to the stock price, i.e. $H^1 = \partial_S \Pi$ and $H^j = 0$ for all $j > 1$. See e.g. Björk (2009) ch. 9 for further details. Conveniently, H has a analytical solution given by

$$H^1 = \partial_S \Pi_t = \Phi\left(\frac{\log\left(\frac{S_t}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}}\right) \tag{36}$$

for a fixed maturity T. Therefore we compare the estimated hedging strategies from the optimization in Equation (29) with the so called *delta hedge* in Equation (36).

The return distribution approximation of the TimeGAN algorithm applied to Geometric Brownian motion is illustrated in Figure 4 and an example of generated sample path is also visualised in Figure 5. Furthermore, the descriptive statistics are presented in Table 1.

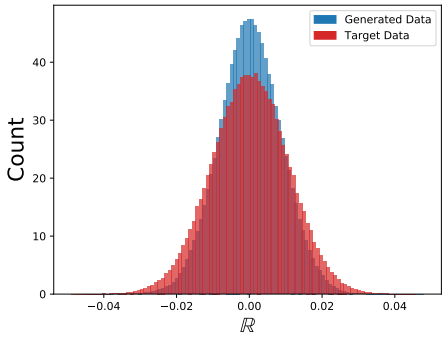

Figure 4: Histogram for returns of generated data $\mathcal{G}(W, \theta)$ obtained by iterating Equation (37) over 30 000 iterations with batch size 128 on a Nvidia© V100 16GB GPU using Tensorflow with CuDNN kernels. Target data S is given by S^1 in Equation (31) on 10 000 samples of 31 daily returns.

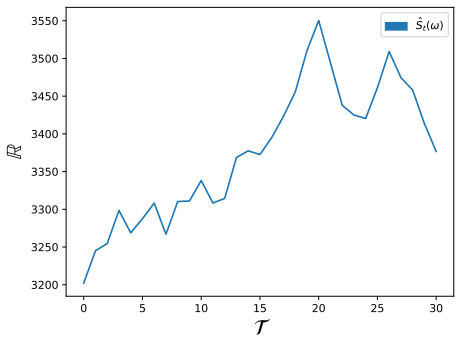

Figure 5: Sample path realization of generated random process $\hat{S}_t(\omega) := \mathcal{G}(W(\omega), \theta)_t$ where W is a Brownian motion as per definition 3.2. Furthermore, the initial condition is set to $S_0 = 3200$ and $\mathcal{T}$ is a index set over 31 days with daily prices. Original sample size is 3000 daily prices.

Data Type	K-S statistic	P-value	Min, Max	Mean	Std.dev	Skewness	Kurtosis
Target	·	·	$(-0.0528,\ 0.0649)$	0.0003	0.0001	-0.0077	0.0845
Model	0.500	0.000	$(-0.0502,\ 0.0466)$	0.0002	0.0001	-0.0313	0.0136

Table 1: Descriptive statistics for return distribution. Null hypothesis for Kolmogorov-Smirnov (K-S) test represents equivalence of distributions. Generated data obtained by calculating returns on the generator function values $\mathcal{G}(W_t(\omega), \theta)$ and target data from Euler discretization of $\frac{dS}{S}$ in Equation (31). Statistics are obtained for 10 000 sample path realizations of 31 daily prices.

Hence, although one can see in Figure 4 that the univariate $\mathbb{P}$-distribution of a geometric Brownian motion is rather well approximated, the Kolmogorov-Smirnov test concludes that the target data and generated return data is from different distributions. However, this is likely due to the large sample size chosen to conduct the test. If we decrease the sample size to 90 days of returns, the Kolmogorov-Smirnov test statistic is reduced to 0.0.161 and a p-value of 0.211 which is at least non-zero. However, the null hypothesis is still not rejected. In order to get a more direct overview on the distributional approximation of the price S, we consider a countable set of 200 function values of the random process S and its approximation $\hat{S}$. The results are shown in Figure 6, which illustrate similar results to those obtained in e.g. Buehler et al. (2020).

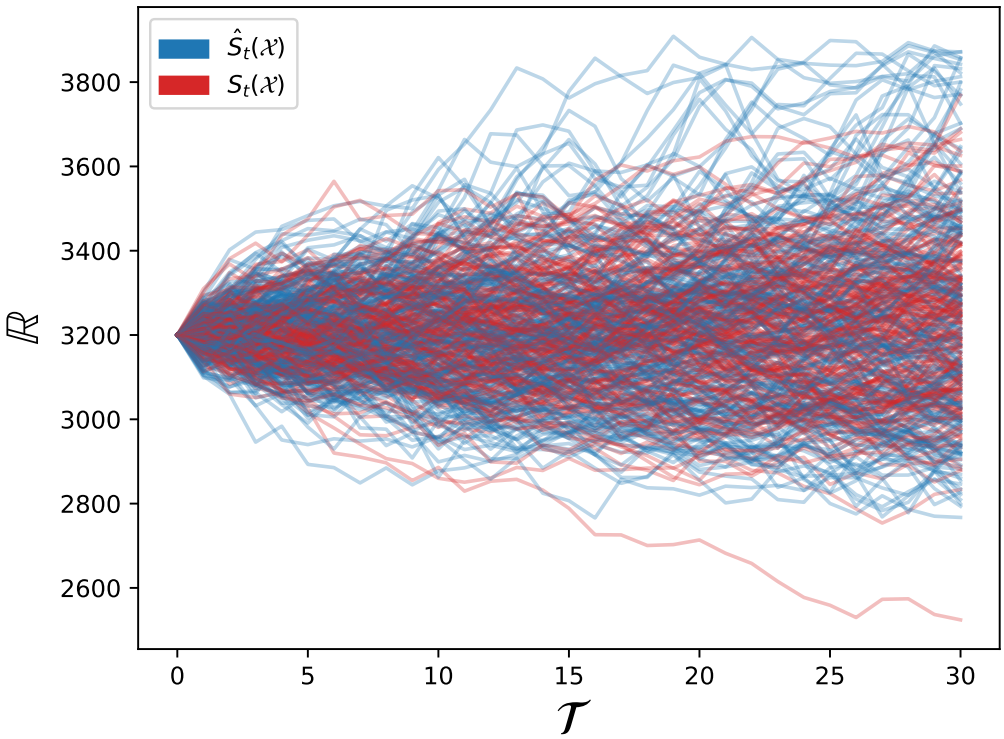

Figure 6: Simulated paths from $\hat{S}(\mathcal{X}) := \mathcal{G}(W(\mathcal{X}), \theta)$ (Blue) and $S(\mathcal{X})$ (Red) given by Equation (30).$\mathcal{X} \subset \Omega$ is a countable subset of 200 samples.

From Figure 6, we can see that marginal probability distributions, i.e. $\mathbb{P}(\omega \in \Omega : S_t(\omega) \leq s)$ seems rather well approximated. Although our implementation of the TimeGAN algorithm is unable to replicate the thickness of the tails for the target distribution. In Fu et al. (2019), they are able to more closely fit the tails of the distribution. The fat-tailed/power-law nature of returns is also used as a key evaluation metric used in Buehler et al. (2020), hence the ability for our proposed model to generalize to real financial time-series may be impaired by this stylised fact.

Since the random process S is a geometric Brownian motion given by Equation 32, it is Markovian and therefore its returns exhibit zero autocorrelation. We visualize the results in Figure 7 from which one can see that the TimeGAN approximation also preserves low autocorrelation. However, note that the autocorrelation is non-zero for the first time-step.

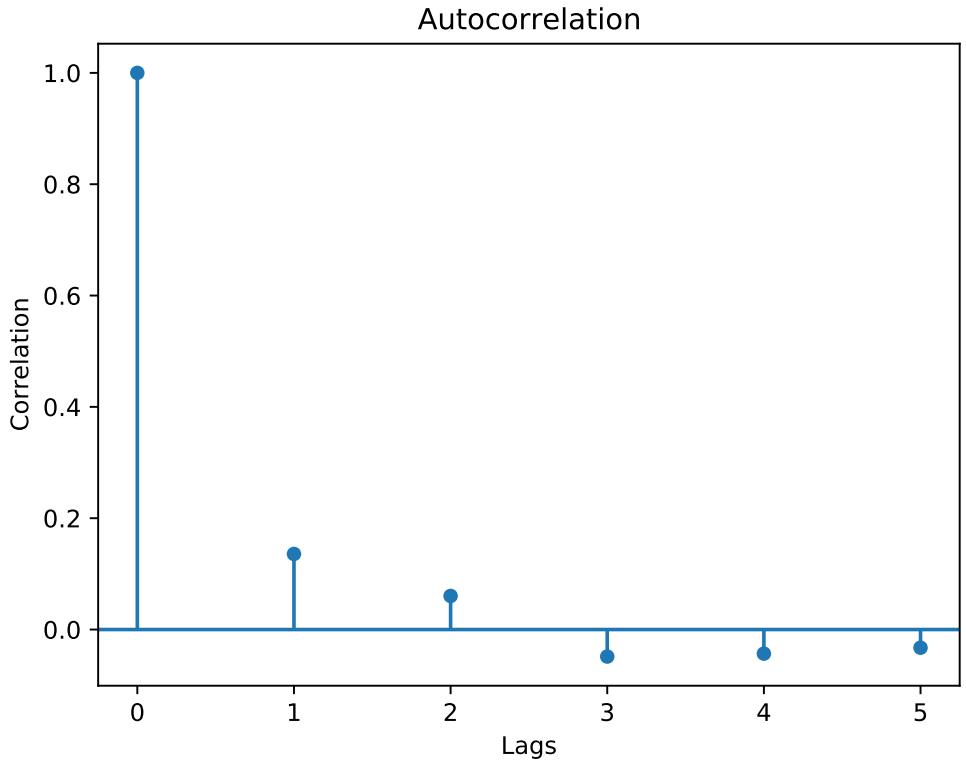

Figure 7: Autocorrelation plot for $\hat{S}(\mathcal{X})$. $\mathcal{X} \subset \Omega$ is a countable set of 10 000 sample paths. Geometric Brownian motion is a Markovian process hence zero autocorrelation beyond first step so S is omitted from the figure.

This is one property that, upon reflection, could have been more clearly incorporated in the architecture to suit the application of financial time-series generation. If one would have utilised *signature transforms*, one could potentially eliminate this unwanted dependence structure since the time-series would be encoded in very few coefficients. Furthermore, empirical research into asset returns also confirm that real financial time-series exhibit zero autocorrelation and hence

constitutes a desirable property to replicate in order to accurately generalize to real financial time-series.

We now consider the performance of the deep hedging algorithm. Firstly, the approximation of the target δ-strategy is investigated when trained on sample paths of the true geometric Brownian motion S. When the approximation capabilities and minimisation the L^2 loss given by Equation 29 is established, we extend to sample paths from the generator which approximates the $\mathbb{P}-$dynamics of the random process S. In Figure 8, we show the resulting profit-loss-distribution associated to the optimization in Equation 29 and Figure 9 displays sample path of the neural network hedge, trained on sample paths of S, along with the benchmark δ-hedge.

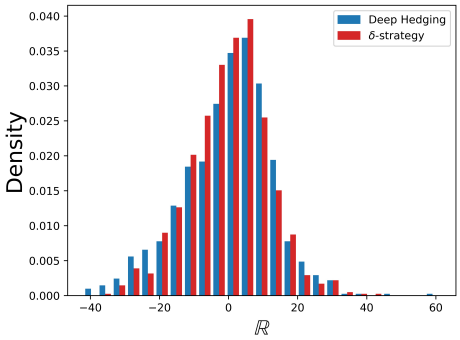

Figure 8: Profit and loss histogram of Z in Equation (26) of deep hedging algorithm H_θ in the set (28) obtained by Equation (29) and target δ-hedge computed by (36). V_0 in Equation (26) is set to current Black-Scholes of option $\Pi_0 = 74.4$. The model is trained on 200 000 sample paths of S given by Equation (32) and results are evaluated on 1 000 sample paths (out of sample) for at-the-money European call option with $K = 3200$. Training conducted on Nvidia© V100-GPU using Tensorflow.

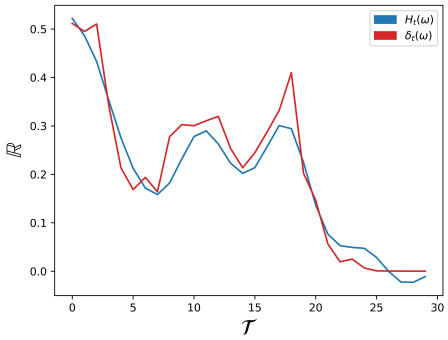

Figure 9: Sample path realization of hedging strategies. Deep Hedge H (blue) obtained by training on sample paths of S in Equation (32) and δ-hedge (red) computed by Equation (36). $\mathcal{T}$ is set to 31 days and $T = 31/365$. The option that is being hedged is at-the-money (ATM) call option with $K = 3200$. Data is obtained out-of-sample from trained model with 200 000 sample paths of geometric Brownian motion, with 4 hidden layers, 40 neurons in each layer and batch size of 128.

From the results presented in Figures 8 and 9, one can see that training the deep hedging model on sample paths of the true $\mathbb{P}$-dynamics of S approximates the target profit and loss distribution. Note that a replication error is always going to be present in any discrete-time implementation. Hence even though the theoretical hedge can completely hedge all risk in continuous time, the terminal profit loss distribution for the hedged portfolio will have non-zero variance. If one would allow more frequent hedge re-balancing, the profit loss distribution will converge towards zero.

As one can see in Figure 8, the target δ-strategy, given by Equation (36), is learned by the hedging algorithm which can be further illustrated by the function values in Figure 9. Hence one can empirically confirm that L^2 loss in Equation (29) is also a feasible loss function in the context of deep hedging. Therefore, the intuition of characterising optimal hedging policies as

constrained minimisation of the arbitrage vector length is confirmed empirically.

We now consider the model performance of the joint algorithm, i.e training the deep hedging model on the approximated $\mathbb{P}$-dynamics given by the generator function values $\mathcal{G}(W_t(\omega), \theta^{\mathcal{G}})$ where W is a standard Brownian motion, see Definition 3.2 for details. The pnl-distribution of the hedged portfolio is illustrated in Figure 10 and a sample path of the hedge in Figure 11. The relevant metrics are presented in Table 2.

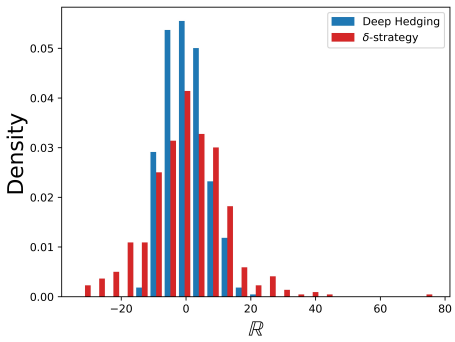

Figure 10: Profit and loss histogram of deep hedging trained on generator sample paths. The figure illustrates the density of Z in Equation (26) for deep hedging algorithm H_θ in the set (28) obtained by Equation (29). The target δ-hedge is computed by (36). V_0 in Equation (26) is set to Black-Scholes price of option given by Theorem 7 with $r = 0$ and $\sigma = 0.2$, resulting in $V_0 = \Pi_0 = 74.4$. The model is trained on 200 000 generated sample paths From $\hat{S} = \mathcal{G}(W, \theta^{\mathcal{G}})$ and results are evaluated on 1 000 sample paths (out of sample) for at-the-money European call option with $K = 3200$. Training conducted on V100-GPU using Tensorflow.

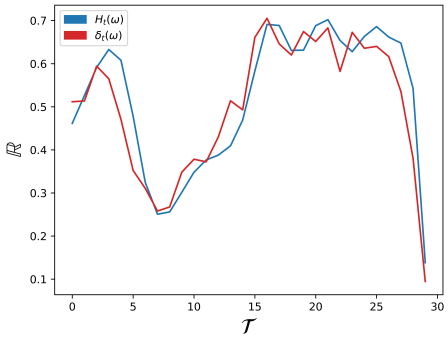

Figure 11: Sample path realization of hedging strategies. Deep Hedge H (blue) obtained by training on sample paths of the TimeGAN generator $\hat{S} = \mathcal{G}(W, \theta^{\mathcal{G}})$, see Appendix B. δ-hedge (red) computed by Equation (36) on S. $\mathcal{T}$ is set to 31 days and $T = 31/365$. The option that is being hedged is at-the-money (ATM) call option with $K = 3200$. Data is obtained out-of-sample from trained model with 200 000 sample paths from generator. Hedging network has 4 hidden LSTM layers, 40 neurons in each layer and batch size of 128.

Model Type	Π_0	Hedging Loss	$\|H - \delta\|_2$	Mean	Std.dev
δ-strategy	69.00	14.375	0.000	0.000	12.285
Deep Hedge	73.94	6.9851	0.0558	0.000	5.646

Table 2: Statistics on testing results of hedging using the joint model. $\Pi_0 := X - V_T$ for both strategies. Hedging loss is defined by the L^2 loss in Equation (29) evaluated on 1 000 samples of the generator (out-of-sample). Note that distributions has zero mean since they are de-meaned.

From Figures 10 to 11 and Table 2 one can see that the deep hedging model seemingly outperforms the δ-hedge even though their sample paths are very similar. However, note that one calculates the δ-strategy using the Black-Scholes model, on sample paths from the generator.

Therefore, since the GAN approximation is imperfect, especially in the tails, the $\delta-$strategy is naturally slightly biased. The GAN model will assign a larger probability to the option remaining in the money than what the normal distribution implies and therefore the δ-strategy will be under sensitive to lower probability moves. This likely causes the δ-strategy induced price to be lower than the Black-Scholes price of $\Pi_0 = 74.4$ as can be seen in Table 2. Furthermore, the TimeGAN induced distributional divergence may also be the root cause of the tighter profit-loss distribution, see Figure 10 and Table 2. However, due to the time constraints of a thesis project, we are not able to further investigate weather or not this is the case. In Figure 11, we can see that deep hedging strategy is very similar to the δ-hedge which is further confirmed by their L^2 distance presented in Table 2.

The code for the model herein presented is available at the projects GitHub repository. For further information regarding training time and hyper-parameter settings used, the reader is referred to the projects Weights And Biases pages.

As is indicated by the results, our combined model is seemingly able to outperform the δ-strategy, which could be due to the approximation error caused by the TimeGAN algorithm. However, we are unable to produce results on real market data to a satisfactory degree. We believe that this is at least in part attributable to the architecture of the TimeGAN algorithm, since it uses a multitude of recurrent neural networks and the CuDNN kernels for recurrent neural networks is somewhat constrained such that recurrent dropout regularisation is unavailable. However, we believe that the main reason as to why the model is unable to generalise is due to the fact that random process approximation is a highly specific task. Therefore, in our view, a different architecture should be considered for future research and applications. Preferably one that uses signature transforms and is tailor-made to random process approximation in the context of financial modelling such as Ni et al. (2020) or Kidger et al. (2019). More suggestions are provided in the conclusion.

8 Conclusions & Suggestions for Future Research

This thesis has investigated both the theoretical aspects and implementation of a derivative pricing model, exclusively using machine learning. We combine the deep hedging model presented by Buehler et al. (2019) with the TimeGAN algorithm in Yoon et al. (2019) in order to approximate the $\mathbb{P}$-dynamics of a given asset price process, trained on a reasonable number of samples. The deep hedging algorithm is trained on samples from the generator in order to arrive at a pricing and hedging policy.

Our results indicates that the model is successful in approximating both pricing/hedging and simulation in the context of the Black-Scholes model. Meaning that we have successfully implemented both component algorithms in a controlled simulation environment. However, we are unable to generalise beyond a one dimensional geometric Brownian motion. Generative adversarial networks are notoriously difficult to train because of the adversarial objective function. The objective in a regular neural network is to minimize some loss function by choice of parameters. However, in GAN's the objective is to reach a Nash-equilibrium and have a stable loss associated to the generator and discriminator. In essence, one want the generator to achieve a accuracy of "fooling the discriminator" of around 50% and a classification accuracy for the discriminator of 50% Salimans et al. (2016). We have found that if the generator loss is larger than 50%, then the generator likely fools the discriminator by generating samples that only exploits the discrim-

inator miss-classification, which has also been found in numerous other GAN architectures, see e.g. Salimans et al. (2016). In the case of the TimeGAN algorithm we believe that this is likely due to the many networks involved in the calculation of the total loss. Therefore we believe that in order to achieve stable training, the GAN architecture has to be suited to random process approximation. In our view, architectures similar to Ni et al. (2020) would likely be better since it explicitly uses signatures, which accelerates training and yet still preserves the essential structures of the original dataset. Furthermore, we also believe that some of the training instability could be addressed by placing larger emphasis on pre-processing in general. Furthermore, we also found that the TimeGAN approximation of the geometric Brownian motion yields non-zero auto-correlated returns. This would likely also be addressed by signature transforms and could perhaps be resolved through a larger amount of training iterations.

Even though we can visually identify that the distribution is approximated for a geometric Brownian motion, the training procedure was highly sensitive to hyper-parameter optimization. For example, the convergence of the TimeGAN algorithm was highly sensitive to changes in learning rate, batch size and the number of neurons in the hidden layers of the network. We achieved convergence of the algorithm only whilst using a depth of 3 layers in each component network, learning rate of 0.0005, batch size of 128 and 30 000 training iterations on a V100 GPU. Hence, upon reflection, we would likely choose a different architecture for the generative adversarial component of the joint method.

The deep hedging model converge to its benchmarked δ-hedging strategy when trained on sample paths of 1-dimensional geometric Brownian motion. Furthermore, replacing the convex risk measure in Equation (27), by the L^2 norm still approximates the target δ-hedging strategy as can be seen in Figures 8 and 9. However when training the deep hedging network on samples from the generator $\mathcal{G}$, the loss density becomes tighter than the delta hedging strategy, see Figures 10 to 11 and Table 2. The δ-strategy is computed by the Black-Scholes model and represents the in-the-money probability, see Equation (36). As discussed in Section 7, the δ-strategy is computed on sample paths of the generator which has tighter return distribution than that of the target $\mathbb{P}$-dynamics of the geometric Brownian motion. Therefore, the δ-strategy will naturally associate higher probability to tail events than the trained neural network strategy would. This causes a bias in the δ-strategy, which is likely the cause of the over-performance of the deep hedging strategy. Furthermore, this is also likely the cause of the miss-pricing illustrated in Table 2 for the δ-strategy. Therefore, it might be the case that the inability of the TimeGAN algorithm to approximate the tails of the target distribution, see Figure 4, causes a divergence in hedging performance from the benchmark strategy. However, it could also be a result of implementation error. Due to the limited time constraints placed on a thesis project, we have not been able to investigate this result any further.

Since we are unable to approximate the distribution of real asset prices to a satisfactory degree, a lot of research is still required for the interplay between the deep hedging algorithm and neural samplers. There is still no guarantee that the distribution implied by the neural sampler preserves no-arbitrage, especially the martingale condition in Theorem 2. One possible avenue of how to address this could be to utilise the deep hedging algorithm combined with a neural sampler (GAN, RBM, VAE), to better understand the connection between the physical measure $\mathbb{P}$ and the pricing measure $\mathbb{Q}$. For example, the theoretical extension of the deep hedging to cover the entire option chain is rather trivial, since arbitrage will be defined point-wise on the term-structure. If one is able to simultaneously price and hedge all options on a fixed underlying asset and then backpropagate the pricing error to the generator, then one could potentially

calibrate the generator to market prices. Thereby implicitly learning the pricing distribution of the underlying asset.

In conclusion, even though our proposed model, i.e. the combination of the TimeGAN algorithm with the deep hedging model, shows good performance in approximating the Black-Scholes model in terms of simulation, pricing and hedging, we are unable to generalise to real market data. In the future, we would consider replacing the TimeGAN algorithm with models similar to those in Ni et al. (2020) or Buehler et al. (2020) and Wiese et al. (2020), since larger emphasis is placed on pre-processing the data.

We hope that this thesis could provide a good introduction to the interplay between generative adversarial networks and the general field of arbitrage theory. Furthermore, given the lack of accurate approximations for real market data, we also hope that this thesis may highlight some implementation related difficulties using the TimeGAN and its impact on hedging strategies.

Bibliography

Billingsley, P. (2008), *Probability and measure*, John Wiley & Sons.

Björk, T. (2009), *Arbitrage theory in continuous time*, Oxford university press.

Black, F. & Scholes, M. (1973), 'The pricing of options and corporate liabilities', *Journal of political economy* **81**(3), 637–654.

Bogachev, V. I. (2007), *Measure theory*, Vol. 1, Springer Science & Business Media.

Buehler, H., Gonon, L., Teichmann, J. & Wood, B. (2019), 'Deep hedging', *Quantitative Finance* **19**(8), 1271–1291.

Buehler, H., Horvath, B., Lyons, T., Perez Arribas, I. & Wood, B. (2020), 'A data-driven market simulator for small data environments', *Available at SSRN 3632431* .

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. & Bharath, A. A. (2018), 'Generative adversarial networks: An overview', *IEEE Signal Processing Magazine* **35**(1), 53–65.

Delbaen, F. & Schachermayer, W. (2006), *The mathematics of arbitrage*, Springer Science & Business Media.

Föllmer, H. & Leukert, P. (1999), 'Quantile hedging', *Finance and Stochastics* **3**(3), 251–273.

Föllmer, H. & Schied, A. (2011), *Stochastic finance: an introduction in discrete time*, Walter de Gruyter.

Föllmer, H. & Schweizer, M. (1991), 'Hedging of contingent claims', *Applied stochastic analysis* **5**, 389.

Fu, R., Chen, J., Zeng, S., Zhuang, Y. & Sudjianto, A. (2019), 'Time series simulation by conditional generative adversarial net', *arXiv preprint arXiv:1904.11419* .

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, *in* 'Advances in neural information processing systems', pp. 2672–2680.

Hornik, K. (1991), 'Approximation capabilities of multilayer feedforward networks', *Neural networks* **4**(2), 251–257.

Jacod, J. & Protter, P. (2012), *Probability essentials*, Springer Science & Business Media.

Jacod, J. & Shiryaev, A. N. (1998), 'Local martingales and the fundamental asset pricing theorems in the discrete-time case', *Finance and stochastics* **2**(3), 259–273.

Kidger, P., Bonnier, P., Arribas, I. P., Salvi, C. & Lyons, T. (2019), Deep signature transforms, *in* 'Advances in Neural Information Processing Systems', pp. 3105–3115.

Kidger, P. & Lyons, T. (2020), Universal approximation with deep narrow networks, *in* 'Conference on Learning Theory', pp. 2306–2327.

Kondratyev, A. & Schwarz, C. (2019), 'The market generator', *Available at SSRN 3384948* .

Musiela, M. & Rutkowski, M. (2005), *Martingale methods in financial modelling*, Springer Finance.

Ni, H., Szpruch, L., Wiese, M., Liao, S. & Xiao, B. (2020), 'Conditional sig-wasserstein gans for time series generation', *arXiv preprint arXiv:2006.05421* .

Nowozin, S., Cseke, B. & Tomioka, R. (2016), f-gan: Training generative neural samplers using variational divergence minimization, *in* 'Advances in neural information processing systems', pp. 271–279.

Protter, P. E. (2005), *Stochastic integration and differential equations*, Springer.

Roth, K., Lucchi, A., Nowozin, S. & Hofmann, T. (2017), Stabilizing training of generative adversarial networks through regularization, *in* 'Advances in neural information processing systems', pp. 2018–2028.

Rudin, W. (2006), *Real and complex analysis*, Tata McGraw-hill education.

Rudin, W. et al. (1964), *Principles of mathematical analysis*, Vol. 3, McGraw-hill New York.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. & Chen, X. (2016), 'Improved techniques for training gans', *arXiv preprint arXiv:1606.03498* .

Schäfer, A. M. & Zimmermann, H. G. (2006), Recurrent neural networks are universal approximators, *in* 'International Conference on Artificial Neural Networks', Springer, pp. 632–640.

Schweizer, M. (1995), 'Variance-optimal hedging in discrete time', *Mathematics of Operations Research* **20**(1), 1–32.

Treil, S. (2016), 'Linear algebra done wrong'.

Vittori, E., Trapletti, M. & Restelli, M. (2020), 'Option hedging with risk averse reinforcement learning', *arXiv preprint arXiv:2010.12245* .

Wiese, M., Knobloch, R., Korn, R. & Kretschmer, P. (2020), 'Quant gans: Deep generation of financial time series', *Quantitative Finance* pp. 1–22.

Yoon, J., Jarrett, D. & van der Schaar, M. (2019), Time-series generative adversarial networks, *in* 'Advances in Neural Information Processing Systems', pp. 5508–5518.

A Mathematical Prerequisites

This Appendix serves the purpose of introducing the reader that is unfamiliar with probability theory to the necessary concepts utilised in the thesis. Therefore, this appendix contains all notation not covered in the main part of the thesis. Hence, the reader is referred here in case of unclearity regarding notation utilised in the above sections.

Outline. In Appendix A.1, we aim to provide a mathematical introduction to probability spaces. In particular, we define topological spaces and vector spaces, both of which will be needed for future constructions. Furthermore, in Appendix A.2, probability theory from a measure of theoretical point of view is considered, since it will be required throughout the thesis. Appendix A.2 contains several sub-appendices. We start by constructing the relevant space for general probability theory. In Appendix A.2.2, the theory developed in Appendix A.2.1 is used to provide a proper definition of *random variables* as measurable functions on the previously constructed probability space. We also define the *expectation operator* as the Lebesgue integral over the domain of a random variable.

A.1 Algebraic Structures

Algebraic structures embed information about a set and provide operations necessary for the definition of the object of study. Arguably the simplest structure is that of a *topology*

Definition A.1. Let M be a non-empty set. A *topology* is a collection of subsets $\mathcal{O} \subset \mathcal{P}(M)$ such that

- $\emptyset, M \in \mathcal{O}$.
- For any sequence $(A_n)_{n\in\mathbb{N}}$ such that $A_n \in \mathcal{O}\ \forall n \in \mathbb{N}$ it is true that
$$\bigcup_{n\in\mathbb{N}} A_n \in \mathcal{O}.$$
- Let $N \in \mathbb{N}$ be finite, then for any finite sequence $(A_n)_{n=1,\dots,N}$ such that $A_n \in \mathcal{O}\ n \leq N$ it is true that
$$\bigcap_{n\leq N} A_n \in \mathcal{O}.$$

For further reference on basic topology, see e.g. Rudin et al. (1964) ch 2.

Terminology. The elements of a topology are called the *open sets* and $(M, \mathcal{O})$ a *topological space*.

For completeness *vector spaces* are also defined. Throughout the thesis, the L^p spaces of random variables and processes (to be defined) are used for construction and can be shown to adhere to the vector space definition. Hence the following definition is absolutely central to the formal theory herein developed.

Definition A.2. Let $(K, +, \cdot)$ be a field. A *K-vector space* is a triple $(V, \oplus, \odot)$ where V is a non-empty set and two maps
$$\oplus : V \times V \longrightarrow V$$
$$\otimes : K \times V \longrightarrow V$$
satisfying the axioms: $C^{\oplus} A^{\oplus} N^{\oplus} I^{\oplus}$ $A^{(\oplus,\otimes)} D^{(\oplus,\otimes)} D^{(\oplus,\otimes)} U^{\otimes}$. We call the elements of a vector space *vectors*.

Any introductory linear algebra textbook can be consulted for the clarification of the vector space axioms, see e.g. Treil (2016) pp 1-2.

Notation. From now on, unless it is not clear from context which maps are meant, we drop the circle notation.

A.2 Probability Theory

In this Appendix we try to provide a acceptable foundation to readers that are uninitiated in probability theory. Therefore, naturally, this Appendix will also develop notation, that is utilised in the thesis.

A.2.1 Probability Spaces

Fundamentally, what is a probability? Intuitively a probability is notion of relative volume. The Formal definition of a probability require restricting to which sets/events probabilities can be defined on.

Definition A.3. Let Ω be a non-empty set. A collection of subsets $\mathcal{F} \subset \mathcal{P}(\Omega)$ is called a *σ-algebra* if:

- $\emptyset, \Omega \in \mathcal{F}$.
- $\forall A \in \mathcal{F} : \Omega \setminus A \in \mathcal{F}$.
- For all countable sequences $(A_n)_{n\in\mathbb{N}}$ such that $A_n \in \mathcal{F},\ \forall n \in \mathbb{N}$ it is true that

$$\bigcup_{n\in\mathbb{N}} A_n \in \mathcal{F}.$$

Terminology. The elements of a σ-algebra are called the *measurable sets* and $(\Omega, \mathcal{F})$ a *measurable space*.

The most frequently used σ-algebra in probability theory is the so called *Borel* σ-algebra. However, before defining the Borel σ-algebra, we need the notion of generated σ-algebras.

Definition A.4. Let M be a set and $A \subset M$. Then the collection

$$\mathcal{E} = \sigma(A) := \bigcap_{\substack{\sigma:\sigma-\text{algebra} \\ A\subset\sigma}} \sigma$$

is a σ-algebra. Furthermore, $\mathcal{E}$ is called the σ-algebra *generated* by A which is called the *generating set*. Note that $\mathcal{E}$ constitutes the *smallest* σ-algebra containing A.

Proof. See e.g. Jacod & Protter (2012). □

Definition A.5. Let $\left(\mathbb{R}, \mathcal{O}\big|_{\mathbb{R}}\right)$ be the *standard* topological space over $\mathbb{R}$. Then the *Borel* σ-algebra is defined as

$$\mathcal{B} := \sigma(\mathcal{O}\big|_{\mathbb{R}}).$$

Theorem 8. *Let $(\mathbb{R}, \mathcal{B})$ be a Borel measurable space over $\mathbb{R}$. Then*

$$\mathcal{B} = \sigma(\{(-\infty, x] : x \in \mathbb{R})\})$$

Proof. See e.g. Rudin (2006) page 12. □

Sufficient definitions have now been made in order to construct the notion of a *measure*

Definition A.6. Let $(\Omega, \mathcal{F})$ be a measurable space. A *measure* is a map

$$\mu : \mathcal{F} \longrightarrow \mathbb{R}$$

satisfying the following conditions:

i For all $A \in \mathcal{F}:\ \mu(A) \geq 0$.

ii $\mu(\emptyset) = 0$.

iii For any countable sequence $(A_n)_{n\in\mathbb{N}}$ of measurable and *pairwise disjoint* sets, it is true that

$$\mu\left(\bigcup_{n\in\mathbb{N}} A_n\right) = \sum_{n\in\mathbb{N}} \mu(A_n).$$

Terminology. We call the triple $(\Omega, \mathcal{F}, \mu)$ a *measure space*.

One general measure that is used in the definition of probability densities is the so called *Lebesgue* measure.

Definition A.7. Let $(\mathbb{R}^d, \mathcal{F})$ be a measurable space. The *Lebesgue* measure on $\mathbb{R}^d$ is the map

$$\lambda^d : \mathcal{F} \longrightarrow \mathbb{R}$$
$$E \longmapsto \lambda^d(E) := \prod_{i=1}^{d} (b_i - a_i)$$

for every closed interval E with $b_i \geq a_i$ for all i.

Now, the definition of a *probability* measure is very simple.

Definition A.8. Let $(\Omega, \mathcal{F})$ be a measurable space. A *probability* measure is a measure such that

$$\mathbb{P} : \mathcal{F} \longrightarrow [0, 1]$$

and we call the triple $(\Omega, \mathcal{F}, \mathbb{P})$ a *probability space*.

A.2.2 Random Variables & Integration

One of the most fundamental concepts in probability theory is that of a random variable and the expectation operator.

Definition A.9. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a measurable space and $(E, \mathcal{E})$ a measurable space. Then a map $X : \Omega \longrightarrow E$ is called a *random variable* if

$$\forall B \in \mathcal{E}: \qquad X^{-1}(B) \in \mathcal{F}.$$

where $X^{-1}(B) = \{\omega \in \Omega : X(\omega) \in B\}$ denotes the *pre-image* of B under the map X.

Terminology. In the above definition we say that the random variable X is $(\mathcal{F}, \mathcal{E})$-measurable. If the choice of σ-algebra in the target is immaterial, the σ-algebra in the target on which X is measurable is linguistically ignored.

One way to *inherit* a probability measure from a probability space is through what is called the *push-forward* method, which is vital in the construction of the Lebesgue integral.

Definition A.10. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(M, \mathcal{O})$ a topological space that induces a measurable space $(M, \sigma(\mathcal{O}))$. Furthermore, let $X : \Omega \longrightarrow M$ be a $(\mathcal{F}, \sigma(\mathcal{O}))$-measurable random variable. Then the *push-forward* probability measure $X_*\mathbb{P}$ is a probability measure on $(M, \sigma(\mathcal{O}))$ and is defined by

$$\begin{aligned} X_*\mathbb{P} : \sigma(\mathcal{O}) &\longrightarrow [0,1] \\ B &\longmapsto (X_*\mathbb{P})(B) := \mathbb{P}(X^{-1}(B)) = (\mathbb{P} \circ X^{-1})(B) \end{aligned}$$

for any $B \in \sigma(\mathcal{O})$

Remark. Let $M = \mathbb{R}$ and choose $\mathcal{O} = \mathcal{O}\big|_{\mathbb{R}}$ as the standard topology on $\mathbb{R}$, then the topology induces the Borel σ-algebra $\sigma(\mathcal{O}_{S|\mathbb{R}}) = \mathcal{B}$. Furthermore, by Theorem 8

$$\begin{aligned} (X_*\mathbb{P})(B) &= \mathbb{P}(\{\omega \in \Omega : X(\omega) \in B\}) \\ &= \mathbb{P}(\{\omega \in \Omega : X(\omega) \in (-\infty, x]\}) \\ &= \mathbb{P}(\{\omega \in \Omega : X(\omega) \leq x]\}) \end{aligned}$$

which provides the characterization of the *probability distribution* of a real valued random variable, as a measure inherited from the probability space over the target, making $(\mathbb{R}, \mathcal{B}, X_*\mathbb{P})$ a probability space in its own right.

Definition A.11. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Furthermore, let X and Y be two random variables defined on Ω. Then X is equal to Y *almost surely* if

$$X \underset{a.s.}{=} Y : \Longleftrightarrow \exists\, \mathcal{N} \in \mathcal{F} : X(\bar{\omega}) = Y(\bar{\omega}), \qquad \forall \bar{\omega} \in \Omega \setminus \mathcal{N}$$

where $\mathbb{P}(\mathcal{N}) = 0$. Furthermore, equal almost surely constitutes an equivalent relation. The definition works just as well when Y is a constant.

Notation. If it is unclear from context which probability measure is meant we explicitly denote the equivalence relation by $\underset{\mathbb{P}-a.s.}{=}$.

In order to develop a notion of integration of non-negative measurable functions / random variables on probability spaces, we need to define set functions.

Definition A.12. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(\mathbb{R}^+, \mathcal{B}^+)$ be a Borel measurable space. Then a measurable function $S : \Omega \longrightarrow \mathbb{R}^+$ is called a simple function if its image is finite

$$S(\Omega) = \{S_1, \ldots, S_N\}$$

for some $N \in \mathbb{N}$.

Remark.

$$S = \sum_{z \in S(\Omega)} z \cdot 1_{S^{-1}(\{z\})} \in \mathcal{F}, \qquad \forall i \leq N$$

where $1_{\{A\}}$ is an indicator function for an arbitrary set A. Note that $S(\Omega)$ is a countable set and S is finite and therefore do not need to worry about convergence.

We can now give the definition of integration on probability spaces.

Definition A.13. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(\bar{\mathbb{R}}_0^+, \mathcal{E})$ a measurable space over the *extended* non-negative real numbers. Then the *Lebesgue* integral over the domain of a non-negative real random variable $X : \Omega \longrightarrow \bar{\mathbb{R}}_0^+$ is a real number

$$\int_\Omega X d\mathbb{P} := \sup_{0 \leq Y \leq X} \sum_{y \in Y(\Omega)} y \cdot \mathbb{P}(Y^{-1}(\{y\})) = \sup_{0 \leq Y \leq X} \sum_{y \in Y(\Omega)} y \cdot (Y_* \mathbb{P})(\{y\})$$

where $Y : \Omega \longrightarrow \bar{\mathbb{R}}_0^+$ is a simple random variable as in Definition A.12 that lie below the graph of X.

Definition A.14. A random variable X is called *integrable* if

i X is measurable.

ii $\int_\Omega |X| d\mathbb{P} < \infty$.

Remark. Let $X : \Omega \longrightarrow \bar{\mathbb{R}}$ then $\int_\Omega |X| d\mathbb{P} < \infty$ if and only if

$$\int_\Omega X^+ d\mathbb{P} < \infty, \qquad \int_\Omega X^- d\mathbb{P} < \infty$$

where $X = X^+ - X^-$, $X^+ := \max(X, 0)$ and $X^- := \max(-X, 0)$.

Definition A.15. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(\mathbb{R}, \mathcal{E})$ be a measurable space over the real numbers. The function space of *p-integrable* real valued random variables is the set

$$\mathcal{L}^p(\Omega, \mathcal{F}, \mathbb{P}) := \left\{ X : \Omega \longrightarrow \mathbb{R} \,\middle|\, X \ (\mathcal{F}, \mathcal{E})\text{-measurable}, \ \int_\Omega |X| d\mathbb{P} < \infty \right\}$$

for some $p < \infty$ and any σ-algebra on the target.

The definition of the *expectation operator* is now very simple.

Definition A.16. Let $X \in L^0(\Omega, \mathcal{F}, \mathbb{P})$ be a random variable. The *expectation operator* is the *linear* map

$$\begin{aligned} \mathbb{E} : L^0 &\longrightarrow \mathbb{R} \\ X &\longmapsto \mathbb{E}(X) := \int_\Omega X d\mathbb{P}. \end{aligned}$$

Before being able too highlight the abstract definition of a expectation to the standard intrductory definition, we have to state a theorem from measure theory.

Theorem 9 (Radon - Nikodym)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Suppose that*

$$\exists\, \nu >> \mathbb{P} : \Longleftrightarrow \ \mathbb{P}(A) = 0 \implies \nu(A) = 0, \qquad \textit{for some } A \in \mathcal{F}.$$

Then

$$\begin{aligned} &\exists\, \mathcal{F}\textit{-measurable random variable } Y : \\ \nu(A) = &\int_A Y d\mathbb{P} \qquad \forall\, A \in \mathcal{F}. \end{aligned}$$

Proof. See e.g. Billingsley (2008) □

Notation. Y is called the *Radon-Nikodym* derivative and denote it by $\frac{d\nu}{d\mathbb{P}}$. Note that this theorem holds for general measures and not only probability measures.

Definition A.17. Let $X : \Omega \longrightarrow \mathbb{R}^n$ be a random variable. Furthermore, let λ^n be the Lebesgue measure on $\mathbb{R}^n$ as in Definition A.7, and suppose that $\lambda^n >> X_*\mathbb{P}$. Then the *probability density* X exists and is defined as the Radon-Nikodym derivative

$$f_X := \frac{dX_*\mathbb{P}}{d\lambda^n}.$$

For random variables with values in the integers, we let the reference measure ν be the counting measure.

Theorem 10 (Change of Variables)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ and $(E, \mathcal{E})$ be a probability space and measurable space respectively. Let $X : \Omega \longrightarrow E$ be a random variable. Another random variable Y defined on E is integrable with respect to the push-forward measure $X_*\mathbb{P}$ if and only if the composition $Y \circ X$ is integrable with respect to $\mathbb{P}$. In that case, the integrals coincide*

$$\int_\Omega Y dX_*\mathbb{P} = \int_E Y \circ X d\mathbb{P}.$$

Proof. See e.g. Bogachev (2007) □

We now give some concrete examples on applications of Radon - Nikodym derivatives and Theorem 9.

Examples. Let the target σ-algebra in Definition A.17 be the Borel σ-algebra $\mathcal{B}$ on $\mathbb{R}$.

i Then the distribution can be characterized as

$$(X_*\mathbb{P})(B) = \int_{X^{-1}(B)} dX_*\mathbb{P}, \quad \text{for any } B \in \mathcal{B}.$$

$$f_X := \frac{dX_*\mathbb{P}}{d\lambda^n} \text{ implies } dX_*\mathbb{P} = f_X d\lambda^n.$$

$$\begin{aligned}\text{Hence } \int_{X^{-1}(B)} dX_*\mathbb{P} &= \int_{X^{-1}(B)} f_X d\lambda^n \\ &= \int_{X^{-1}(B_1)} \int_{X^{-1}(B_2)} \dots \int_{X^{-1}(B_n)} f_X(x_1, \dots, x_n) dx_1 dx_2 \dots dx_n \\ &= \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \dots \int_{-\infty}^{x_n} f_X(y_1, \dots, y_n) dy_1 dx_2 \dots dy_n.\end{aligned}$$

which is the usual characterization of a probability density function for a $\mathbb{R}^n$ valued random variable,

ii Suppose $X : \Omega \longrightarrow \mathbb{R}$ and $\lambda >> X_*\mathbb{P}$, then by Theorem 9

$$\begin{aligned}\mathbb{E}(X) &= \int_\Omega X d\mathbb{P} = \int_{\mathbb{R}} x d(X_*\mathbb{P}(x)) \\ &= \int_{\mathbb{R}} x f_X(x) d\lambda = \int_{-\infty}^{\infty} x f_X(x) dx\end{aligned}$$

Notice that we go from the abstract Lebesgue integral to the Riemann integral. For conditions on which functions are Riemann integrable, see e.g. Rudin et al. (1964).

We are now able to properly discuss *conditional expectations*.

Definition A.18. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Furthermore, let $\mathcal{G} \subset \mathcal{F}$ be a σ-algebra. Then the conditional expectation of a $\mathcal{F}$-measurable random variable given the *information* $\mathcal{G}$ is a random variable $\mathbb{E}(X|\mathcal{G})$ such that

i . $\mathbb{E}(X|\mathcal{G}) \in L^1(\Omega, \mathcal{G}, \mathbb{P})$.

ii . For all $G \in \mathcal{G}$ it is true that

$$\int_G \mathbb{E}(X|\mathcal{G}) d\mathbb{P} = \int_\Omega X d\mathbb{P}$$

Remarks. The definition is quite abstract and only states which conditions must be satisfied. Hence we proceed by the following remarks

i . Since $\mathcal{G} \subseteq \mathcal{F}$, X is not necessarily $\mathcal{G}$-measurable. Which means that integrals like

$$\int_G X d\mathbb{P}\big|_{\mathcal{G}}, \quad G \in \mathcal{G}$$

where $\mathbb{P}\big|_{\mathcal{G}}$ is the restriction of $\mathbb{P}$ to $\mathcal{G}$, do not exist unless X is $\mathcal{G}$-measurable.

ii . The existence of $\mathbb{E}(X|\mathcal{G})$ can be established by noting that $\forall\, A \in \mathcal{F}$ it is true that

$$\mu^X : A \longmapsto \int_A X d\mathbb{P}$$

is a finite measure on $(\Omega, \mathcal{F})$ such that $\mu << \mathbb{P}$. Let $g : \mathcal{G} \longrightarrow \mathcal{F}$ be an inclusion map, then

$$\mu^X \circ g = \mu^X\big|_{\mathcal{G}}$$

and $\mathbb{P} \circ g = \mathbb{P}\big|_{\mathcal{G}}$. Furthermore, $\mu^X \circ g << \mathbb{P} \circ g$, hence their Radon-Nikodym derivative is defined and the conditional expectation exists in the form

$$\mathbb{E}(X|\mathcal{G}) = \frac{d\mu^X\big|_{\mathcal{G}}}{d\mathbb{P}\big|_{\mathcal{G}}}.$$

iii . Suppose Y is also random variable with values in the measurable space $(E, \mathcal{E})$. Then the conditional expectation of X given Y is defined as

$$\mathbb{E}(X|Y) := \mathbb{E}\left\{X|\sigma(Y^{-1}(\mathcal{E}))\right\} = \mathbb{E}\left\{X|Y^{-1}(\mathcal{E})\right\}.$$

B TimeGAN Algorithm

The particular architecture we consider for approximating distributions in this thesis is the *TimeGAN* algorithm introduced by Yoon et al. (2019). In practice, the convergence of a "prototypical GAN" as per Definition 5.7, can be difficult to achieve with respect to sequential data according to Yoon et al. (2019). The TimeGAN attempts to address this by introducing auxiliary components; One *embedding* function $\mathcal{E}$, *recovery* function $\mathcal{R}$, a generator $\mathcal{G}$ and finally a discriminator $\mathcal{D}$. The first two components are maps into and out of a lower dimensional *latent space representation* of the state space of $d+1$-dimensional market process S respectively. The

adversarial components $(\mathcal{G}, \mathcal{D})$, then aim to map some $d+1$-dimensional noise process Z into the latent space, on which the discriminator acts. For our applications, it makes sense to use a *Brownian motion* W, see Definition 3.2, to represent Z.

Formally, let $(\mathcal{V}, +, \cdot)$ be a $\mathbb{R}$-vector space of dimension $m < d$. Furthermore, let $\mathcal{E}$ and $\mathcal{R}$ be maps

$$\begin{aligned} \mathcal{E} : L^\infty(\Omega \times \mathcal{T}) \times \Theta^{\mathcal{E}} &\longrightarrow \mathcal{V} \\ \mathcal{R} : \mathcal{V} \times \Theta^{\mathcal{R}} &\longrightarrow L^\infty(\Omega \times \mathcal{T}) \end{aligned}$$

represented as outputs of two recurrent neural networks, as defined in Definition 5.3. In particular $\mathcal{E}$ and $\mathcal{R}$ adhere to the representation

$$\begin{aligned} \mathcal{E}(S, \theta^{\mathcal{E}})_t &= f_{\mathcal{E}}(\mathcal{E}_{t-1}, S_t; \theta^{\mathcal{E}}) \\ \mathcal{R}(\tilde{S}, \theta^{\mathcal{R}})_t &= f_{\mathcal{R}}(\mathcal{R}_{t-1}, \tilde{S}_t; \theta^{\mathcal{R}}) \end{aligned}$$

for state transition functions $f_{\mathcal{E}}$ and $f_{\mathcal{R}}$ and each $t \in \mathcal{T}$. It is evident that the objective of the recovery function is to reconstruct the random process from its latent space representation, i.e. serve as a approximation of the inverse embedding function $\mathcal{E}^{-1}$. To that end, Yoon et al. (2019) defines the *reconstruction loss*

$$\mathcal{L}_R := \sum_{t \in \mathcal{T}} \| S_t - (\mathcal{R} \circ \mathcal{E})(S)_t \|_2$$

where $\mathcal{R} \circ \mathcal{E}(S)$ is the *reconstructed process* and the norm $\|\cdot\|_2$ is the L^2 norm on the probability space $(\Omega, \mathcal{F}_t^W, \mathbb{P})$ defined by

$$\|S_t\|_2 = \left(\int_\Omega |S|^2 d\mathbb{P} \right)^{\frac{1}{2}}$$

for all $t \in \mathcal{T}$. Note that $\mathcal{T}$ is a countable and finite set, hence convergence is naturally satisfied. The *generator* and *discriminator* are now defined as a map and a class of functions

$$\begin{aligned} \mathcal{G} : L^\infty(\Omega \times \mathcal{T}) \times \theta^{\mathcal{G}} &\longrightarrow \mathcal{V} \\ \mathcal{D} : \mathcal{V} &\longrightarrow [0, 1]. \end{aligned}$$

Furthermore, in order to train the adversarial component one can define a *unsupervised loss* $\mathcal{L}_U$, equivalent to Equation (22), by

$$\mathcal{L}_U := \mathbb{E}_{\mathbb{P}} \left(\sum_{t \in \mathcal{T}} (\log \circ \mathcal{D})(\tilde{S})_t \right) + \mathbb{E}_{\tilde{\mathbb{P}}_\theta} \left(\sum_{t \in \mathcal{T}} (\log \circ \mathcal{D} \circ \mathcal{G})(1 - W)_t \right)$$

where W is a d-dimensional Brownian motion as per Definition 3.2 and $\tilde{S} \in \mathcal{V}$. However, Yoon et al. (2019) argues that $\mathcal{L}_U$ may be insufficient to recover the marginal conditional distributions of the random process S and to address this they consider the additional *supervised* loss $\mathcal{L}_S$ to further discipline learning

$$\mathcal{L}_S := \sum_{t \in \mathcal{T}} \left\| \mathcal{E}(S, \theta^{\mathcal{E}})_t - \mathcal{G}(W, \theta^{\mathcal{G}})_t \right\|_2$$

where again $\|\cdot\|_2$ is the L^2 norm on $(\Omega, \mathcal{F}_t^W, \mathbb{P})$. The training is then performed in two steps

$$\begin{aligned} &\text{step 1: } \inf_{(\theta^{\mathcal{E}}, \theta^{\mathcal{R}}) \in \Theta^{\mathcal{E}} \times \Theta^{\mathcal{R}}} \{ \lambda \mathcal{L}_S + \mathcal{L}_R \} \\ &\text{step 2: } \inf_{\theta^{\mathcal{G}} \in \Theta^{\mathcal{G}}} \{ \eta \mathcal{L}_S + \sup_{\theta^{\mathcal{D}} \in \Theta^{\mathcal{D}}} \mathcal{L}_U \} \end{aligned} \tag{37}$$

for hyperparameters $\lambda, \eta \in \mathbb{R}^+$, where step 1 is the training of the embedder and recovery networks and step 2 is the adversarial training. Note that step 2 in Equation (37) is very similar to the prototypical adversarial objective in Equation (21), with the only difference being an additional supervised loss component. Both steps in Equation (37) is achieved by gradient descent and backpropagation. The optimizations in Equation (37) enables the TimeGAN algorithm to encode the random process, generate latent space representations of the process and recover such latent space representations into bounded random processes.

Intuitively, one can think of the TimeGAN algorithm as optimizing a latent space representation and noise embedding maps such that the mapped noise is "dense" in the latent space representation of the random process. This makes hyper-plane separation between real and fake samples infeasible and thus high probability classification is unattainable. The objective is then to learn to reconstruct the distribution of the original process by learning the inverse embedding function.