May 2019

Model-Independent Estimation of Optimal Hedging Strategies with Deep Neural Networks

Tobias Michael Furtwaengler
University of Wisconsin-Milwaukee

MODEL-INDEPENDENT ESTIMATION OF OPTIMAL HEDGING STRATEGIES WITH DEEP NEURAL NETWORKS

by

Tobias Michael Furtwaengler

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

in MATHEMATICS

at

The University of Wisconsin-Milwaukee

May 2019

ABSTRACT

MODEL-INDEPENDENT ESTIMATION OF OPTIMAL HEDGING STRATEGIES WITH DEEP NEURAL NETWORKS

by

Tobias Michael Furtwaengler

The University of Wisconsin-Milwaukee, 2019
Under the Supervision of Professor Chao Zhu

Inspired by the recent paper Buehler et al. (2018), this thesis aims to investigate the optimal hedging and pricing of financial derivatives with neural networks. We utilize the concept of convex risk measures to define optimal hedging strategies without strong assumptions on the underlying market dynamics. Furthermore, the setting allows the incorporation of market frictions and thus the determination of optimal hedging strategies and prices even in incomplete markets. We then use the approximation capabilities of neural networks to find close-to optimal estimates for these strategies. We will elaborate on the theoretical foundations of this approach and carry out implementations and a detailed analysis of the method with simulated market data. Our experiments show that the neural network-based algorithm is a powerful tool for the model-independent pricing of any financial derivative and the estimation of optimal hedging strategies for these instruments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The pricing of financial derivatives and closely related to that the hedging of these instruments is still one of the most important topics in financial mathematics. For an agent on the financial markets, it is essential to determine appropriate prices for the derivatives he is trading. On the one hand these prices must be competitive in the market, but on the other hand, reflect the real risk the agent has to bear by entering the contract. After the trade, the agent (at least if we consider a financial institution that has to follow the regulations of the supervisor authorities) has to guarantee continual risk management up to the maturity or liquidation of the contract to prevent unexpected losses and thereby ensure the financial stability of the institution.

Although much progress in the field of option valuation has been made since the 1970s, there are still many unsolved problems, and the research continues. One of the most common approaches for pricing and the implementation of risk management for financial derivatives are stochastic financial models. These models assume that a particular stochastic dynamic drives the underlying assets of the derivatives. Most of the popular models assume the market to be complete, i.e., for all contingent claims, there exists a trading strategy on the underlying assets that replicates the payoff perfectly. That allows the market participants to determine fair prices by risk-neutral valuation either with closed-form solutions or with a Monte Carlo method. Probably the most popular among those models is the so-called Black-Scholes-Merton model from 1977 [6]. The model assumes the underlying stock price to follow a geometric Brownian motion and offers a closed-form solution for the price of European options. Furthermore, the model offers analytical expressions for the so-called *Greeks*, the partial derivatives of the option value with respect to the underlying asset as well as to the different parameters of the model. These sensitivities then allow dynamic risk management by implementing a Greek-based hedging strategy.

Even though it is one of the most used and most taught models, the Black-Scholes-Merton model exhibits some severe shortcomings. The model assumes the volatility of the underlying asset price process to be constant. Observations of real market data, however, show that the volatility exhibits a nonconstant behavior and varies with time and the level of the price itself. This phenomenon is mainly referenced as the *volatility smile*. One approach to take nonconstant volatility into account is the local volatility Model of Dupire [10] that expands the Black-Scholes-Merton model by replacing the constant volatility with a deterministic function that depends on time and the level of the underlying price process. Another way to model these characteristics are the so-called stochastic volatility models. Arguably, the most popular of these models is the Heston model. Here the volatility itself is modeled by a stochastic differential equation. More precisely, it is assumed to follow a Cox-Ingersoll-Ross process [12]. Although the mentioned models may capture the non-constant behavior of the volatility to some extent, they still require other assumptions that do not hold in reality. Most importantly they all assume the possibility of unrestricted trading in time and amount of the hedging instruments as well as the complete absence of trading costs for all transactions that are required for the implementation of the hedging strategy. In reality, a trader will only adjust his strategy at discrete time steps. Moreover, he has to take into account trading costs, market impact, limitations in liquidity and limited capacity of risk or capital to find an optimal hedging strategy. There are a variety of different approaches to include market frictions into option pricing models, see for example [19]. However, all of them are restricted to strong assumptions and need concepts as the expected utility pricing. Moreover, as shown in [20], there is no nontrivial hedging strategy for options in the Black-Scholes-Merton setting if transaction costs are included. Therefore, these methods are of limited value in practice, and despite their shortcomings, stochastic models are still widely used, mainly due to the lack of effective alternatives.

Motivated by these findings, the recent paper [7] introduces a new approach to determine optimal hedging strategies that no longer depend on an underlying stochastic model. The authors use artificial neural networks to model a trading strategy that is capable of replicating an arbitrary payoff in any market scenario. Neural networks are mathematical structures that are utilized for the approximation of mathematical relations without knowledge of the specific characteristics of the actual function. The networks can be trained on given data and then later make predictions based on new data. Initially used for picture and voice recognition, it turned out that neural networks show surprisingly good performances in many different applications. Modern machine learning and optimization techniques allow efficient training of these structures and make them applicable to large data sets. Out of this reason, they offer a well-suited framework for the problem of determining optimal hedging strategies. The input of the networks might not only consist of market data but also on other information and previous trading decisions. The approach offers several advantages to traditional stochastic financial models: At first, the method is independent on any assumption about the underlying price process. Moreover, one can easily include any market friction, for example, transaction costs and trading constraints. Finally, this approach offers the possibility to find optimal hedging strategies for a variety of different risk measures. The versatile applications may be of interest for financial institutions that face strict capital requirements by the supervisory authorities that often depend on a particular risk measure.

The first part of this thesis will summarize the framework that was introduced [7] and elaborate on the theoretical background where it is necessary for the understanding of their ideas. Thereby the focus will be set on those parts of the work that are relevant for the second part of this thesis. Here we will present the results and analysis of our numerical experiments. We want to address several questions. At first, we are interested in finding out if the neural network based approach is capable of learning a model hedge on an idealized market that is assumed to follow the Black-Scholes-Merton model without any market frictions. We then extend the analysis by incorporating trading costs into the model to assess if the method can be used to make reasonable hedging decisions under more realistic market conditions. Furthermore, we may use this setting to analyze the influence of hedging costs on the risk indifference price of a derivative. Finally, we also want to assess the performance of the approach on a market that is driven by a more complex stochastic model. Therefore, we consider the stochastic volatility model of Heston. In this setting, an optimal hedging strategy has to consider more than one risk factor and therefore trade in different assets.

2 Discrete capital market

2.1 Setting

This work aims to analyze the performance of deep neural networks for the estimation of optimal hedging strategies. For this purpose, we will consider the following idealized financial market that will later also allow us to compare the performance to a conventional model-based hedging strategy. For the setup of this market, we will use a similar setting as in [7]. We consider a market with finite time horizon T and discrete trading dates $0 = t_0 < t_1 < ... < t_n = T$. Furthermore, we have a finite and filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ where $\Omega = \{\omega_1, ..., \omega_N\}$ and $\mathbb{P}(\{\omega_i\}) > 0$ for $i = 1, ..., N$ and $\mathcal{X} := \{X : \Omega \to \mathbb{R}\}$ is the set of all real-valued random variables over Ω. The filtration $\mathbb{F}$ is generated by an $\mathbb{R}^m$-valued process $I = (I_k)_{k=0,\dots,n}$, i.e., $\mathcal{F}_k = \sigma(I_0, ..., I_k)$ and $\mathbb{F} = (\mathcal{F}_k)_{k=0,\dots,n}$. In this setting I_k can be interpreted as the new market information an agent gets at time t_k and the sub-σ-algebra $\mathcal{F}_k$ represents all information that is available for the agent up to time t_k. By this definition, every $\mathcal{F}_k$-measurable random variable can then be written as a function of $I_0, ..., I_k$. In this work the information process I will only consist of prices of liquid financial instruments that are observable at the market. In general, it could also include additional information, for example, news, financial reports or analyst's opinions. Furthermore, there is a set of d assets on the market that can be traded by an agent for hedging purposes. The prices of these assets are given by a $\mathbb{R}^d$-valued and $\mathbb{F}$-adapted stochastic process $S = (S_k)_{k=0,\dots,n}$. These tradable assets can be primary assets, for example, the underlying stock of an equity option but also other derivatives with observable market prices. Finally, we have a $\mathcal{F}_T$-measurable random variable Z that represents the payoff of a derivative and thus the liability of the writing agent. The agent must hedge the payoff in order to minimize the risk he takes by entering the contract.

In order to evaluate and compare the performance of our neural network based strategies, we build up a setting that is as simple as possible. We assume that the only payoff of our contingent claim Z occurs at maturity T as it is the case for European options. These limitations can be easily extended to more complex derivatives by assuming that possible intermediate payoffs are accrued with a risk-free rate r until maturity T. Moreover, we will not consider derivatives with true optionality, such as American options, because of the complex modeling of the corresponding payoffs.

2.2 Hedging in a discrete capital market

To minimize the risk of the payoff Z, the agent may trade in the available assets on the market. This strategy can be represented by a $\mathbb{R}^d$-valued and $\mathbb{F}$-adapted stochastic process $\delta = (\delta_k)_{k=0,\dots,n-1}$ with $\delta_k = (\delta_k^1, \dots, \delta_k^d)$. Note that since Ω is finite, $\delta_k \colon \Omega \to \mathbb{R}^d$ is bounded. This definition can be understood in the following way: The agent holds δ_k^i shares of the i-th asset at time t_k. Furthermore we set $\delta_{-1} = \delta_n := 0$, i.e., before time $t = 0$ the agent does not hold any assets at all and liquidates all his positions at time T. We further assume that the strategy is self-financing, i.e., there are no additional cash injections during the trading period. The set of all such hedging strategies is denoted by $\mathcal{H}$. For this work this definition is sufficient. As emphasized in [7], in reality, an agent could be subjected to trading constraints, for example, limitations in risk (often measured in the form of Greeks) or restrictions in the liquidity of the hedging instruments. Restrictions like that can be easily included in the introduced framework. To see that, assume that δ_k is restricted to a set $\mathcal{H}_k^c$. Then there exists a continuous and $\mathcal{F}_k$-measurable mapping $H_k \colon \mathbb{R}^{d(k+1)} \to \mathbb{R}^d$, so that $H_k(0) = 0$. The set of restricted strategies at time t_k can then be represented as $\mathcal{H}_k^c := H_k(\mathbb{R}^{d(k+1)})$. Accordingly, we might define an constrained strategy δ^c by defining a set of all restricted hedging strategies $\mathcal{H}^c := (H \circ \mathcal{H}) \subset \mathcal{H}$ where H is iteratively defined with $(H \circ \delta)_k := H_k((H \circ \delta)_0, \dots, (H \circ \delta)_{k-1}, \delta_k)$.

Since the considered hedging strategy is self-financed, it might be necessary to inject an additional amount of cash p_0 into the portfolio that consists of the liability and the hedging strategy at the beginning of the trading period. After the maturity of the liability at time T the agent's wealth is then given by $-Z + p_0 + (\delta \cdot S)_T$, with

$$(\delta \cdot S)_T := \sum_{k=0}^{n} \delta_k \cdot (S_{k+1} - S_k)$$

A crucial benefit of the here presented framework to conventional model-based approaches is that we can easily incorporate market frictions such as trading costs. In this work we will consider two different kinds of transaction costs c_k that occur for all transactions at time t_k:

(i) Fixed transaction costs: $c_k(\delta_k - \delta_{k-1}) := \sum_{i=1}^{d} c_k^i 1_{|\delta_k^i - \delta_{k+1}^i| \geq \epsilon}$
where $c_k^i > 0$ for $i = 1, .., d$ and $\epsilon > 0$ is a fixed number.

(ii) Proportional transaction costs: $c_k(\delta_k - \delta_{k-1}) := \sum_{i=1}^{d} c_k^i S_k^i |\delta_k^i - \delta_{k+1}^i|$
where $c_k^i > 0$ for $i = 1, .., d$.

The total cost of the hedging strategy δ is then given by:

$$C_T(\delta) := \sum_{k=0}^{n} c_k(\delta_k - \delta_{k-1}).$$

Note that in both cases the cost function c_k is upper semi-continuous and normalized to $c_k(0) = 0$. The value of the agent's portfolio at maturity Z, also called *profit and loss*, is accordingly given by:

$$PnL_T(Z, p_0, \delta) := -Z + p_0 + (\delta \cdot S)_T - C_T(\delta).$$

3 Convex risk measures

Conventional financial models are mostly based on the assumption of a complete market with continuous-time and unconstrained trading and no transaction costs. In this scenario there exists for every contingent claim Z a unique replicating strategy δ and a fair price p_0 such that at maturity T it holds: $-Z + p_0 + (\delta \cdot S)_T = 0,\ \mathbb{P} - a.s.$ In the setting introduced in Section 2, this does not longer hold since we included transaction costs and restricted the possible trading times to a discrete set. An agent in this market, therefore, has to specify a different criterion to determine an optimal minimal price he has to charge for the derivative Z. This price is the minimal amount of cash the agent has to add to his portfolio such that the overall position is acceptable. Out of this reason, the price should be based on the risk that the agent has to take by selling the liability. We therefore assume a position Z to be acceptable if it holds that $\rho(Z) \leq 0$ for a given risk measure $\rho\colon \mathcal{X} \to \mathbb{R}$. We assume the risk measure to be normalized, i.e., $\rho(0) = 0$. In this work we will focus on convex risk measures, that cover most of the used risk measures in practice. In the following part we consider a random variable X that represent a long position in one or more assets, i.e. we can represent a short position or a liability as $-X$. For the definition of convex risk measures we follow [11].

Definition 3.1. Convex risk measure. Let $X, X_1, X_2 \in \mathcal{X}$. A risk measure $\rho\colon \mathcal{X} \to \mathbb{R}$ is called convex if it is:

(i) Monotonic: $X_1 > X_2 \Rightarrow \rho(X_1) \leq \rho(X_2)$., i.e., a more favorable position requires less additional cash to make it acceptable.

(ii) Convex: $\rho(\lambda X_1 + (1-\lambda)X_2) \leq \lambda\rho(X_1) + (1-\lambda)\rho(X_2)$ for all $\lambda \in [0,1]$, i.e., the risk of a diversified portfolio is always less or equal the sum of risks of the individual positions.

(iii) Cash-Invariant: $\rho(X + c) = \rho(X) - c$ for $c \in \mathbb{R}$, i.e., if we add cash to a position, the risk is reduced by this amount. Specifically, it holds that $\rho(X + \rho(X)) = 0$, i.e. $\rho(X)$ is the least amount of cash the agent has to add to his position to make it acceptable.

The entropic risk measure, defined by: $\rho_\lambda(X) := \frac{1}{\lambda}\ln(\mathbb{E}[e^{-\lambda X}])$ for $X \in \mathcal{X}$ and risk aversion parameter $\lambda > 0$ is one example for a convex risk measure. Another risk measure of this class is the *Conditional Value at Risk* that will be discussed in detail in Section 3.3. Contrarily, the popular risk measure *Value at Risk*, which is a specific quantile of the distribution of the random variable $X \in \mathcal{X}$, is in general not a convex risk measure.

3.1 Hedging and pricing under convex risk measures

We now consider an agent who wants to hedge the payoff X that occurs at time T by trading in S according to the strategy δ. The agent then is interested in finding an optimal strategy in order to minimize his risk. This can be expressed as the following optimization problem:

$$\pi(X) := \inf_{\delta \in \mathcal{H}} \rho(X + (\delta \cdot S)_T - C_T(\delta)) \tag{1}$$

where $\rho\colon \mathcal{X} \to \mathbb{R}$ is a convex risk measure and $X \in \mathcal{X}$. As shown in [7], it holds that π is again a convex risk measure under some assumptions.

Theorem 3.2. Let π be as in (1). Then π is monotone decreasing and cash-invariant. If moreover C_T is convex and $\mathcal{H}$ is a convex set, i.e. for all $\delta_1, delta_2 \in \mathcal{H}$ and $\lambda \in [0,1]$ it holds $\lambda\delta_1 + (1-\lambda)\delta_2 \in \mathcal{H}$, then π is convex and therefore a convex risk measure.

Proof. At first, we note that the monotonicity and cash-invariance of π follow directly from its definition and the respective properties of ρ. For the convexity, let $\lambda \in [0,1]$ and set $\gamma := 1-\lambda$. Note that for all $\delta \in \mathcal{H}$ it holds: $(\delta \cdot S)_T = \lambda(\delta \cdot S)_T + \gamma(\delta \cdot S)_T$. Further assume that $X_1, X_2 \in \mathcal{X}$. By definition we can find for all $\epsilon > 0$ strategies $\delta_1, \delta_2 \in \mathcal{H}$ such that

$$\begin{aligned}
&\pi(X_1) > \rho(X_1 + (\delta_1 \cdot S)_T - C_T(\delta_1)) - \frac{\epsilon}{2} \quad \text{and} \\
&\pi(X_2) > \rho(X_2 + (\delta_2 \cdot S)_T - C_T(\delta_2)) - \frac{\epsilon}{2}.
\end{aligned}$$

Adding the two inequalities and setting $\delta := \lambda\delta_1 + \gamma\delta_2$ leads to:

$$\begin{aligned}
\lambda\pi(X_1) + \gamma\pi(X_2) &> \lambda\rho(X_1 + (\delta_1 \cdot S)_T - C_T(\delta_1)) + \gamma\rho(X_2 + (\delta_2 \cdot S)_T - C_T(\delta_2)) - \epsilon \\
&\geq \rho(\lambda[X_1 + (\delta_1 \cdot S)_T - C_T(\delta_1)] + \gamma[X_2 + (\delta_2 \cdot S)_T - C_T(\delta_2)]) - \epsilon \\
&= \rho([\lambda X_1 + \gamma X_2] + ([\lambda\delta_1 + \gamma\delta_2] \cdot S)_T - [\lambda C_T(\delta_1) + \gamma C_T(\delta_2)]) - \epsilon \\
&\geq \rho((\lambda X_1 + \gamma X_2) + ([\lambda\delta_1 + \gamma\delta_2] \cdot S)_T - C_T(\lambda\delta_1 + \gamma\delta_2)) - \epsilon \\
&\geq \inf_{\delta \in \mathcal{H}} \rho([\lambda X_1 + \gamma X_2] + (\delta \cdot S)_T - C_T(\delta)) - \epsilon \\
&= \pi(\lambda X_1 + \gamma X_2) - \epsilon
\end{aligned}$$

where we used sequentially the convexity of ρ, the convexity of C_T combined with the monotonicity of ρ, the convexity of $\mathcal{H}$ and finally the definition of π. Since $\epsilon > 0$ is arbitrary we can conclude:

$$\lambda\pi(X_1) + \gamma\pi(X_2) \geq \pi(\lambda X_1 + \gamma X_2)$$

□

An optimal hedging strategy with respect to the risk measure ρ can now be defined as a minimizer $\delta^* \in \mathcal{H}$ of (1). Given a liability $-Z$, where $Z \in \mathcal{X}$, the value $\pi(-Z)$ can then be interpreted as the minimal amount of cash an agent has to add to his portfolio in order to make it acceptable given that he is hedging optimally according to the strategy δ^*. As emphasised in [7] it could be possible that trading in some of the hedging instruments is expected to generate positive returns under $\mathbb{P}$ (for example by incorporating trading signals). Because of these considerations we normalize π to determine the minimal acceptable price for the payoff Z.

Definition 3.3. Risk indifference price. The risk indifference price $p(Z)$ for a payoff Z is defined by

$$p(Z) := \pi(-Z) - \pi(0) \tag{2}$$

According to the definition, $p(Z)$ is the solution p_0 of the equation $\pi(-Z + p_0) = \pi(0)$. This means that the agent is indifferent between selling the liability Z for the price p_0 or not doing so.

As highlighted in [7], this definition of the risk indifference price coincides to the price of a replicating portfolio, i.e., the fair price of a liability in the setting of a complete market (or more general of a market without transaction costs and trading constraints).

Lemma 3.4. Assume $C_T \equiv 0$. If a liability Z is replicable, i.e $\exists\ \tilde{\delta} \in \mathcal{H}$ and $p_0 \in \mathbb{R}$ such that $Z = p_0 + (\tilde{\delta} \cdot S)_T$, then it holds that $p(Z) = p_0$.

Proof. For an arbitrary $\delta \in \mathcal{H}$ it holds that

$$\rho(-Z + (\delta \cdot S)_T - C_T(\delta)) = \rho(([\delta - \tilde{\delta}] \cdot S)_T) + p_0$$

by substituting $p_0 + (\tilde{\delta} \cdot S)_T$ for Z and using the cash-invariance of ρ. The identity follows by taking the infimum over all $\delta \in \mathcal{H}$ on both sides of the equation:

$$\pi(-Z) = p_0 + \inf_{\delta \in \mathcal{H}} \rho(([\delta - \tilde{\delta}] \cdot S)_T) = p_0 + \pi(0).$$

using the fact that $\mathcal{H} - \tilde{\delta} = \mathcal{H}$. And thus $p_0 = \pi(-Z) - \pi(0) = p(Z)$. $\square$

For practical applications of the methodology presented above, the following observation in [7] is worth mentioning:

Remark 3.5. Assume that an agent is acting on a market where the price p_0 for a payoff Z is exogenously given. For a given loss function $\ell \colon \mathbb{R} \to [0, \infty)$ an optimal hedging strategy δ^* can then be defined as the minimizer of the following optimization problem:

$$\inf_{\delta \in \mathcal{H}} \mathbb{E}[\ell(-Z + p_0 + (\delta \cdot S)_T - C_T(\delta))]$$

An example for this approach would be an agent that wants to trade derivatives on the market at competitive prices without taking into account risk management.

In order to use deep neural networks to estimate optimal solutions for the presented optimization problem (1) we now introduce the concept of optimized certainty equivalents as introduced in [4]. The optimized certainty equivalent of a random variable $X \in \mathcal{X}$ is defined for a given utility function, i.e. continuous, non-decreasing and concave $u \colon \mathbb{R} \to \mathbb{R}$, as $OCE_u(X) = \sup_{w\in\mathbb{R}}\{w + \mathbb{E}[u(X-w)]\}$. The OCE can be interpreted as an optimal allocation of X between present consumption w and the expected utility of future consumption $E[u(X-w)]$. Furthermore, it can be shown that any OCE induces a risk measure ρ by setting $\rho(X) := -OCE_u(X)$. We note that for all utility functions $u \colon \mathbb{R} \to \mathbb{R}$ we can define a loss function $\ell \colon \mathbb{R} \to \mathbb{R}$ by setting $l(x) := -u(-x)$. This motivates the definition of a class of risk measure that are induced by an OCE.

3.2 OCE risk measures

Definition 3.6. OCE risk measure. Let $\ell \colon \mathbb{R} \to \mathbb{R}$ be a *loss function*, i.e, ℓ is continuous, non-decreasing and convex. For $X \in \mathcal{X}$ we might now define a risk measure ρ with:

$$\rho(X) := \inf_{w\in\mathbb{R}}\{w + \mathbb{E}[\ell(-X-w)]\} \tag{3}$$

It will turn out that this representation will later be helpful to build up a theoretical foundation for the use of neural networks for the estimation of optimal hedging strategies.

Theorem 3.7. An OCE risk measure as defined in (3) is a convex risk measure.

Proof. Let $X, X_1, X_2 \in \mathcal{X}$.

(i) Monotonicity: Suppose $X_1 \leq X_2$. Since ℓ is non-decreasing, it holds that $\mathbb{E}[\ell(-X_1-w)] \geq \mathbb{E}[\ell(-X_2-w)]\ \forall\, w \in \mathbb{R}$ and thus $\rho(X_1) \geq \rho(X_2)$.

(ii) Cash-invariance: $\rho(X+m) = \inf_{w\in\mathbb{R}}\{(w+m) - m + \mathbb{E}[\ell(-X-(w+m))]\} = \inf_{w\in\mathbb{R}}\{w + \mathbb{E}[\ell(-X-w)]\} - m = \rho(X) - m \quad \forall\, m \in \mathbb{R}$

(iii) Convexity: Let $\lambda \in [0,1]$ and set $\gamma = 1 - \lambda$. It holds that:

$$\begin{aligned}
\rho(\lambda X_1 + \gamma X_2) &= \inf_{w \in \mathbb{R}} \{w + \mathbb{E}[\ell(-\lambda X_1 - \gamma X_2 - w)]\} \\
&= \inf_{w_1 \in \mathbb{R}, w_2 \in \mathbb{R}} \{\lambda w_1 + \gamma w_2 + \mathbb{E}[\ell(\lambda(-X - w_1) + \gamma(-X_2 - w_2))]\} \\
&\leq \inf_{w_1 \in \mathbb{R}} \inf_{w_2 \in \mathbb{R}} \{\lambda(w_1 + E[\ell(-X_1 - w_1)]) + \gamma(w_2 + \mathbb{E}[\ell(-X_2 - w_2)])\} \\
&= \lambda \rho(X_1) + \gamma \rho(X_2)
\end{aligned}$$

since ℓ is assumed to be convex.

□

For a risk measure ρ that is induced by an OCE the following theorem holds:

Theorem 3.8. Assume S is a martingale under $\mathbb{P}$, i.e. it holds: $\mathbb{E}[|S_k|] < \infty$, S_k is $\mathcal{F}_k$-measurable and $\mathbb{E}[S_{k+1}|\mathcal{F}_k] = S_k$ for $k = 0, ..., n$. Moreover, let ρ be defined as in (3), π as in (1) and p as in (2). Then the following hold:

(i) $\pi(0) = \rho(0)$

(ii) $p(Z) \geq \mathbb{E}[Z] \quad \forall\, Z \in \mathcal{X}$

Proof. At first we note that $\pi(0) \leq \rho(0)$ for any risk measure ρ since $0 \in \mathcal{H}$ and $C_T(0) = 0$. To show that the converse inequality holds, we first note that since S is a martingale, we can use the tower property to observe:

$$\mathbb{E}[(\delta \cdot S)_T] = \sum_{k=0}^{n-1} \mathbb{E}[\delta_k \mathbb{E}[S_{k+1} - S_k|\mathcal{F}_k]] = \mathbb{E}[\delta_k(S_k - S_k)] = 0 \quad \forall\, \delta \in \mathcal{H}$$

We can now use this to show the following:

$$\begin{aligned}
\pi(-Z) &= \inf_{w \in \mathbb{R}} \inf_{\delta \in \mathcal{H}} \{w + \mathbb{E}[\ell(Z - (\delta \cdot S)_T + C_T(\delta) - w)]\} \\
&\geq \inf_{w \in \mathbb{R}} \inf_{\delta \in \mathcal{H}} \{w + \ell(\mathbb{E}[Z - (\delta \cdot S)_T + C_T(\delta) - w])\} \qquad (4) \\
&\geq \inf_{w \in \mathbb{R}} \{w + \ell(\mathbb{E}[Z] - w)\} = \rho(-\mathbb{E}[Z]) = \mathbb{E}[Z] + \rho(0)
\end{aligned}$$

where we used the convexity of ℓ and Jensen's inequality for the first inequality and $C_T(\delta) \geq 0$ for all $\delta \in \mathcal{H}$ as well as the fact that ℓ is a non-decreasing function for the second inequality. For $Z = 0$ we now get $\pi(0) = \rho(0)$ and thus (i). For (ii) we can use the identity in (4) and the definition of the risk indifference price (2) as well as (i) to observe: $p(Z) = \pi(-Z) - \pi(0) \geq \mathbb{E}[Z] + \rho(0) - \pi(0) = \mathbb{E}[Z] + \rho(0) - \rho(0) = \mathbb{E}[Z]$. □

3.3 Conditional Value at Risk

In this work we want to focus on estimating optimal hedging strategies with respect to one particular *OCE* risk measure, namely the *Conditional Value at Risk* (*CVaR*), also known as *Expected Shortfall* (*ES*) or *Average Value at Risk* (*AVaR*). While for a long time the relatively simple and easily interpretable *Value at Risk* (*VaR*) was the most used risk measure in the financial industry, *CVaR* is becoming more popular among market participants in recent years. Also, new regulations of the supervisor authorities often express capital requirements with respect to the *CVaR*. The increasing popularity of this risk measure is based on some important advantages it has to offer compared to the *VaR*. At first, the *CVaR* is a convex risk measure by definition whereas that need not to be true for the *VaR*. This can result in an increased risk in terms of *VaR* after combining positions into a portfolio and is therefore not consistent with the intuitive concept of diversification. Moreover, the *VaR* does not give any information about the risk of losses that exceed the risk value. These shortcomings can be avoided by using the *CVaR*. We define the risk measure for a particular risk level $\alpha \in [0, 1)$ following (3) and using $l(x) = \frac{1}{1-\alpha}x^+$ where $x^+ := \max(x, 0)$ with:

$$\rho_\alpha(X) = CVaR_\alpha(X) = \inf_{w\in\mathbb{R}} \{w + \frac{1}{1-\alpha}\mathbb{E}[(-X - w)^+]\} \tag{5}$$

As shown in [21] we can also use the following representation for (5):

$$CVaR_\alpha(X) = \frac{1}{1-\alpha}\int_{1-\alpha}^{1} VaR_\gamma(X)d\gamma,$$

where the *Value at Risk* for the risk level $\alpha \in [0, 1)$ is defined by:

$$VaR_\alpha(X) := -\inf\{x \in \mathbb{R} : \mathbb{P}(X \leq x) > 1 - \alpha\}.$$

Furthermore, if the distribution of X is continuous, the $CVaR$ coincides with the so called *Tail Conditional Expectation* (TCE) (see [1]) which is defined by :

$$TCE_\alpha(X) := -\mathbb{E}[X| - X \geq VaR_\alpha(X)]. \tag{6}$$

As proposed in [1], the natural estimator of the TCE for a sample $(X_1, ..., X_N)$ of the random variable $X \in \mathcal{X}$, is given with

$$T\hat{C}E_\alpha(X) := -\frac{\sum_{m=1}^{N} X_m 1_{\{X_m \leq X_{\lfloor N(1-\alpha)\rfloor:N}\}}}{\sum_{m=1}^{N} 1_{\{X_m \leq X_{\lfloor N(1-\alpha)\rfloor:N}\}}}, \tag{7}$$

where the order statistic of the sample is denoted by $X_{1:N} \leq ... \leq X_{N:N}$ and $\lfloor x \rfloor := \max\{n \in \mathbb{N} : n \leq x\}$.

4 Neural networks

In this section we want to introduce the so called *feedforward neural networks* as one particular class of artifical neural networks. These neural networks will later be used for the estimation of optimal hedging strategies according to the optimization problem (1) in Section 3.1.

4.1 Feedforward neural networks

Definition 4.1. Feedforward neural network. Let $L, N_0, N_1, ..., N_L \in \mathbb{N}$, $\sigma\colon \mathbb{R} \to \mathbb{R}$, matrices $A^l \in \mathbb{R}^{N_l \times N_{l-1}}$ and the vectors $b^l \in \mathbb{R}^{N_l}$ for $l = 1, ..., L$. A feedforward neural network is defined as the function $F : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ by

$$F(x) = F_L \circ F_{L-1} \circ ... \circ F_1 \quad \text{for } x \in \mathbb{R}^{N_0} \tag{8}$$

where $F_l(x) = \sigma(A^l x + b^l)$ for $x \in \mathbb{R}^{N_{l-1}}$, $l = 1, ..., L-1$ and $F_L(x) = A^L x + b^L$ for $x \in \mathbb{R}^{N_{L-1}}$.

We call σ the activation function and apply it componentwise. L denotes the number of layers, $N_1, ..., N_{L-1}$ the dimensions of the hidden layers and N_0, N_L respectively the dimensions of the input and output layers. Furthermore, the number A^l_{ij} represents the weight of the edge connecting node i of layer $l-1$ and node j of layer l for $i = 1, ..., N_l$ and for $j = 1, ..., N_{l-1}$. The number b^l_i represents the bias that is added at node i of layer l for $i = 1, ..., N_l$. Weights and biases are the parameters of our network that will later be trained on a given set of training data. We denote a specific neural network with F^θ where $\theta \in \mathbb{R}^q$ for some $q \in \mathbb{N}$ is the parameter vector that contains all parameters of the network.

We define $\mathcal{NN}^\sigma_{\infty,d_0,d_1}$ as the set of all feedforward neural networks mapping from $\mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$ with a fixed activation function σ. We will now reference the so called *Universal Approximation Theorem* of Hornik [14] that characterizes feedforward networks with one hidden layer as universal function approximators. The theorem will later justify the use of feedforward neural networks for the estimation of optimal hedging strategies.

Theorem 4.2. Universal Approximation Theorem. Assume the activation function σ to be bounded and non-constant. Then the following holds:

(i) The set $\mathcal{NN}^{\sigma}_{\infty,d_0,1}$ is dense in $L^p(\mathbb{R}^{d_0},\mu)$ for any finite measure μ on $(\mathbb{R}^{d_0},\mathcal{B}(\mathbb{R}^{d_0}))$ and $1 \leq p \leq \infty$.

(ii) If in addition $\sigma \in C(\mathbb{R})$, then $\mathcal{NN}^{\sigma}_{\infty,d_0,1}$ is dense in $C(\mathbb{R}^{d_0})$ for the topology of uniform convergence on compact sets.

As shown in [15], this result remains true for the set $\mathcal{NN}^{\sigma}_{\infty,d_0,d_1}$ with $d_1 > 1$, i.e for all neural networks with an arbitrary number of hidden layers and output layer dimension d_1.

For the following part we consider a sequence of subsets of $\mathcal{NN}^{\sigma}_{\infty,d_0,d_1}$, that is denoted by $\{\mathcal{NN}^{\sigma}_{M,d_0,d_1}\}_{M\in\mathbb{N}}$ and satisfies the following properties:

(i) $\mathcal{NN}^{\sigma}_{M,d_0,d_1} \subset \mathcal{NN}^{\sigma}_{M+1,d_0,d_1} \quad \forall\ M \in \mathbb{N}$

(ii) $\bigcup_{M\in\mathbb{N}} \mathcal{NN}^{\sigma}_{M,d_0,d_1} = \mathcal{NN}^{\sigma}_{\infty,d_0,d_1}$

(iii) For all $M \in \mathbb{N}$ we have the following representation: $\mathcal{NN}^{\sigma}_{M,d_0,d_1} = \{F^{\theta} : \theta \in \Theta_{M,d_0,d_1}\}$ where $\Theta_{M,d_0,d_1} \subset \mathbb{R}^{q(M)}$ and $q(M) \in \mathbb{N}$.

As discussed in [7] there are two classes of networks we could think of. At first $\mathcal{NN}^{\sigma}_{M,d_0,d_1}$ could represent all networks with an arbitrary number of layers but at most M non-zero weights. The second class represents the set of networks with a fixed architecture. That means the number of layers $L^{(M)}$, as well as the input and output dimension of each layer, are fixed. For the sequence $\{\mathcal{NN}^{\sigma}_{\infty,d_0,d_1}\}_{M\in\mathbb{N}}$, the number of layers is specified by the non-decreasing sequence $\{L^{(M)}\}_{M\in\mathbb{N}}$ and the dimension of the layers by d_0, d_1 for input and output layer and by the non-decreasing sequences $\{N_1^{(M)}\}_{M\in\mathbb{N}}, ..., \{N_{L^{(M)}-1}^{(M)}\}_{M\in\mathbb{N}}$ for the hidden layers. For both cases, the networks are completely parameterized by the matrices A^l and the vectors b^l for $l = 1, ..., L$.

4.2 Estimation of optimal hedging strategies with feedforward neural networks.

On base of the presented results of neural networks as universal function approximators we may introduce a setup to use neural networks for the estimation of a solution of the optimization problem (1) in Section 2.2. Here we again follow mainly the ideas presented in [7].

Recall that all available market information up to time t_k is described by $\mathcal{F}_k = \sigma(I_0, ..., I_k)$. The strategy for the specific time step, δ_k, can therefore only be based on this information as well as on the previous position δ_{k-1}. Thus $\delta_k = f_k(I_0, ..., I_k, \delta_{k-1})$ for some function $f\colon \mathbb{R}^{m(k+1)+d} \to \mathbb{R}^d$. We may now use the approximation capabilities of neural networks to approximate the function f_k with the neural network F_k and define the following set of strategies:

$$\begin{aligned}\mathcal{H}_M &= \{(\delta_k)_{k=0,...,n-1} \in \mathcal{H} : \delta_k = F_k(I_0, ..., I_k, \delta_{k-1}), F_k \in \mathcal{NN}_{M,m(k+1)+d,d}\} \\ &= \{(\delta_k)_{k=0,...,n-1} \in \mathcal{H} : \delta_k = F^{\theta_k}(I_0, ..., I_k, \delta_{k-1}), \theta_k \in \Theta_{M,m(k+1)+d,d}\}\end{aligned} \tag{9}$$

Since $\mathcal{H}_M \subset \mathcal{H}$ we can now set up the following optimization problem:

$$\begin{aligned}\pi^M(X) &:= \inf_{\delta\in\mathcal{H}_M} \rho(X + (\delta \cdot S)_T - C_T(\delta)) \\ &= \inf_{\theta\in\Theta_M} \rho(X + (\delta^\theta \cdot S)_T - C_T(\delta^\theta))\end{aligned} \tag{10}$$

where $\Theta_M = \prod_{k=0}^{n-1} \Theta_{M,m(k+1)+d,d}$. This is now a finite-dimensional problem of finding the optimal parameters for the neural network instead of the infinite-dimensional problem of finding an optimal strategy $\delta \in \mathcal{H}$.

Remark 4.3. As proposed in [7] we will refer to the setting (9) as a *semi-recurrent* network structure. Here, recurrence means that the output of the neural network F_k is part of the input of the neural network F_{k+1}. The setting would be fully recurrent if we only had one neural network to determine δ_k for each time step t_k, i.e. $\theta_k = \theta_0$ for $k = 1, ..., n-1$.

Remark 4.4. If S is a *Markov process* with respect to the probability measure $\mathbb{P}$ and the filtration $\mathbb{F}$, $Z = g(S_T)$ for some function $g\colon \mathbb{R}^d \to \mathbb{R}$ and with simplistic market frictions as the transaction costs defined in Section 2.2, we may expect that the optimal strategy simplifies to $\delta_k = f_k(I_k, \delta_{k-1})$ for some $f_k\colon \mathbb{R}^{m+d} \to \mathbb{R}^d$. Indeed, we could confirm this presumption with our numerical experiments, so that we refer in Section 5.4 to a simplified network structure of the form $F^{\theta_k}(I_k, \delta_{k-1})$.

The following theorem in [7] shows, that any strategy $\delta \in \mathcal{H}$ can be approximated arbitrary well by strategies $\delta^M \in \mathcal{H}_M$. Thus, the estimate of the risk indifference price for a payoff Z of the neural network $p^M(Z) = \pi^M(-Z) - \pi^M(0)$ converges to the true risk indifference price $p(Z)$.

Theorem 4.5. For $\mathcal{H}_M$ defined as in (9) and π^M as in (10), it holds:

$$\lim_{M\to\infty} \pi^M(X) = \pi(X) \quad \forall\, X \in \mathcal{X}.$$

Proof. In order to simplify notation we may rewrite (9) as

$$\mathcal{H}_M = \{(\delta_k)_{k=0,\dots,n-1} \in \mathcal{H} : \delta_k = F_k(I_0, ..., I_k), F_k \in \mathcal{NN}_{M,m(k+1)+d,d}\} \tag{11}$$

since δ_{k-1} is by itself a function of $I_0, ..., I_{k-1}$ and thus redundant as an argument of F_k. It holds that $\pi^M(X) \geq \pi^{M+1}(X) \geq \pi(X)$ since $\mathcal{H}_M \subset \mathcal{H}_{M+1} \subset \mathcal{H}$ for all $M \in \mathbb{N}$. Thus, for convergence it is sufficient to show that for all $\epsilon > 0$ there exists a $M \in \mathbb{N}$ such that $\pi^M(X) \leq \pi(X) + \epsilon$. By the definition of π, there exists $\delta \in \mathcal{H}$ such that:

$$\rho(X + (\delta \cdot S)_T - C_T(\delta)) \leq \pi(X) + \frac{\epsilon}{2} \tag{12}$$

Since δ_k is $\mathcal{F}_k$-measurable, there exists a measurable function $f_k\colon \mathbb{R}^{m(k+1)} \to \mathbb{R}^d$ such that $\delta_k = f_k(I_0, ..., I_k)$ for $k = 0, ..., n-1$. Since δ_k is bounded, $\delta_k \in L^1(\Omega, \mathbb{P})$ and hence $f_k^i \in L^1(\mathbb{R}^{m(k+1)}, \mu)$ for $i = 1, ..., d$, where μ is the law of $(I_0, ..., I_k)$ under $\mathbb{P}$. We may now apply Theorem 4.2 to find a sequence $F_{k,n}^i \in \mathcal{NN}_{\infty,m(k+1),1}$ such that $F_{k,n}^i$ converges to f_k^i in $L^1(\mathbb{R}^{m(k+1)}, \mu)$, and thus $F_{k,n}^i(I_0, ..., I_k)$ converges to $f_k^i(I_0, ..., I_k)$ in $L^1(\Omega, \mathbb{P})$ as

$n \to \infty$. We may choose a suitable subsequence so that the convergence also holds $\mathbb{P}$-a.s. simultaneously for $i = 1, ..., d$ and $k = 0, ..., n-1$ (see for example Theorems 4.1.4 and 4.2.3 in [8]). Denote the sequence by $\delta_k^{n_j} := F_{k,n_j}(I_0, ..., I_k)$. Since $\mathbb{P}(\{\omega\}) > 0$ for all $\omega \in \Omega$ it holds:

$$\lim_{j\to\infty} \delta_k^{n_j}(\omega) = \delta_k(\omega) \quad \forall\, \omega \in \Omega \tag{13}$$

Since Ω is finite, the function $\rho\colon \mathbb{R}^N \to \mathbb{R}$ is convex and in particular continuous. It holds that:

$$\begin{aligned}
&\liminf_{j\to\infty} \rho(X + (\delta^{n_j} \cdot S)_T - C_T(\delta^{n_j})) \\
&\leq \rho(X + (\delta \cdot S)_T - \limsup_{j\to\infty} C_T(\delta^{n_j})) \\
&\leq \rho(X + (\delta \cdot S)_T - C_T(\delta))
\end{aligned}$$

where we used the continuity of ρ, the fact that δ_k is bounded and (13) in the first step and the upper semi-continuity of c_k for a fixed $\omega \in \Omega$ as well as the monotonicity of ρ for the second inequality. Hence, for all $\epsilon > 0$ we can find $j \in \mathbb{N}$ such that:

$$\rho(X + (\delta^{n_j} \cdot S)_T - C_T(\delta^{n_j})) \leq \rho(X + (\delta \cdot S)_T - C_T(\delta)) + \frac{\epsilon}{2} \tag{14}$$

Combining (12) and (14), it holds that for all $\epsilon > 0$ there exists a $j \in \mathbb{N}$ such that:

$$\rho(X + (\delta^{n_j} \cdot S)_T - C_T(\delta^{n_j})) \leq \pi(X) + \epsilon \tag{15}$$

Finally, since for all $j \in \mathbb{N}$ there exists $M \in \mathbb{N}$ such that $\delta^{n_j} \in \mathcal{H}_M$ we obtain that $\pi^M(X) \leq \pi(X) + \epsilon$ by (15) and (10). $\square$

4.3 Numerical estimation of optimal hedging strategies for OCE-risk measures

In Section 4.1 we saw that (feedforward) neural networks are universal function approximators and can therefore be used for the estimation of optimal hedging strategies. As shown in [7], a (close-to) optimal parameter vector $\theta \in \Theta_M$ can be estimated efficiently for unconstrained strategies that are based on an OCE risk measure. To see that, we first set $\Theta = \mathbb{R} \times \Theta_M$. We then define for $\bar{\theta} = (w, \theta) \in \Theta$ the objective function $J \colon \Theta \to \mathbb{R}$ with:

$$J(\bar{\theta}) := w + \mathbb{E}[\ell(Z - (\delta^\theta \cdot S)_T + C_T(\delta^\theta) - w)]. \tag{16}$$

We may now rewrite the optimization problem (10) for the liability $-Z$ by:

$$\pi^M(-Z) = \inf_{\theta \in \Theta_M} \inf_{w \in \mathbb{R}} \{w + \mathbb{E}[\ell(Z - (\delta^\theta \cdot S)_T + C_T(\delta^\theta) - w)]\} = \inf_{\bar{\theta} \in \Theta} J(\bar{\theta}) \tag{17}$$

Remark 4.6. The definition of the objective function as in (16) puts our setting in the context of the so called *reinforcement learning*. In that framework an agent is taking actions in an environment in order to maximize some form of reward or minimize a cost which corresponds to the objective function in our setup. Reinforcement learning is besides of supervised and unsupervised learning, considered as one of the three paradigms in machine learning, see for example [5].

The optimization problem (17) is now solvable with deep learning optimization techniques since the objective function J satisfies the following two characteristics:

(i) The gradient of the objective function J decomposes into a sum over the samples, i.e. it holds that $\nabla_{\bar{\theta}} J(\bar{\theta}) = \sum_{i=1}^N \nabla_{\bar{\theta}} J(\bar{\theta}, \omega_i) \quad$ for $i = 1, ..., N$ where $J(\bar{\theta}, \omega) := [w + \ell(Z(\omega) - (\delta^\theta \cdot S)_T(\omega) + C_T(\delta^\theta)(\omega) - w)]\mathbb{P}(\{\omega\}) \quad \forall\ \omega \in \Omega$.

(ii) The gradient $\nabla_{\bar{\theta}} J(\bar{\theta}, \omega)$ can be calculated efficiently $\forall\ \omega \in \Omega$ using the so called *back-propagation* algorithm.

In the following, we assume the loss function ℓ, the cost function c as well as the activation function σ to be continuously differentiable. The calculation of the gradient $\nabla_{\bar{\theta}} J(\bar{\theta}, \omega)$ with the backpropagation algorithm is explained in the following part.

We define $\delta^{\theta} = F^{\theta}(I)$ where $F^{\theta}(I) := (F^{\theta_0}(I_0), ..., F^{\theta_{n-1}}(I_0, ..., I_{n-1}))$ and $I = \{I_0, ..., I_n\}$. We apply the chain rule to find the gradient of $J(\bar{\theta}, \omega)$ by calculating the gradient of F^{θ} which can be further decomposed in the gradients of F^{θ_k} for $k = 0, ..., n-1$. Note that F^{θ_k} is according to Definition 4.1 parameterized by matrices A^l and vectors b^l for $l = 1, ..., L$. We now want to find the partial derivatives of F^{θ_k} with respect to a single parameter $\tilde{\theta} \in \{A^l_{i,j}, b^l_i\}$. The algorithm, as for example described in [5], than works as follows: We start with the so called forward pass by setting $x^0 := (I_0(\omega), ..., I_k(\omega))$ and then iteratively set $x^l := F_l(x^{l-1})$ for $l = 1, ..., L-1$ and $x^L = A^L x^{L-1} + b^L$ (compare to (4)). For the backward pass we then set $D^L := A^L$ and iteratively $D^l := D^{l+1} \cdot dF_l^{\theta_k}(x^{l-1})$ for $l = L-1, ..., 1$, where $dF_l^{\theta_k}(x^{l-1}) = diag[\sigma'(A^l x^{l-1} + b^l)] \cdot A^l$. By once again applying the chain rule we may now get the partial derivatives of F^{θ_k} with respect to the parameter $\tilde{\theta}$ with:

$$\begin{aligned} \delta_{A^l_{i,j}} F^{\theta_k}(I(\omega)) &= D^{l+1}_i \sigma'((A^l x^{l-1} + b^l)_i) x^{l-1}_j \\ \delta_{b^l_i} F^{\theta_k}(I(\omega)) &= D^{l+1}_i \sigma'((A^l x^{l-1} + b^l)_i). \end{aligned} \tag{18}$$

With the gradient $\nabla_{\bar{\theta}} J(\bar{\theta})$ we can apply a gradient descent algorithm to solve the optimization problem (10). Therefore we start with an initial guess $\bar{\theta}^{(0)}$ and then iteratively set:

$$\bar{\theta}^{(j+1)} = \bar{\theta}^{(j)} - \eta_j \nabla_{\bar{\theta}} J(\bar{\theta}^{(j)}), \tag{19}$$

where $\eta_j > 0$ is the learning rate or step size. Following this approach and under suitable assumptions on the objective function J and the sequence $\{\eta_j\}_{j \in \mathbb{N}}$ for the learning rate the parameter vector $\theta^{(j)}$ converges at least to a local minimum of J as $j \to \infty$. Although in the theory of machine learning there are still no results that would guarantee the convergence to the global minimum, there are techniques that can be used to avoid getting stuck in a local minimum. One of these is the so-called minibatch approach. The idea is, to perform the gradient descent step (19) not on base of the whole sample space Ω but only on a randomly chosen subset, the so-called minibatch $\Omega_{batch}^{(j)} = \{\omega_1^{(j)}, ..., \omega_{N_{batch}}^{(j)}\}$ with sample size $N_{batch} \ll N$. We may now define

$$J_j(\bar{\theta}^{(j)}) = w + \sum_{i=1}^{N_{batch}} \ell(Z(\omega_i^{(j)}) - (\delta^\theta \cdot S)_T(\omega_i^{(j)}) + C_T(\delta^\theta)(\omega_i^{(j)}) - w)\frac{N}{N_{batch}}\mathbb{P}(\{\omega_i^{(j)}\}).$$

Instead of using the whole sample space we now calculate the gradient only for the minibatch. The update (19) of the parameter vector θ than becomes:

$$\bar{\theta}^{(j+1)} = \bar{\theta}^{(j)} - \eta_j \nabla_{\bar{\theta}} J_j(\bar{\theta}^{(j)})$$

Following this approach we can not only avoid getting stuck at a local minimum since the subset $\Omega_{batch}^{(j)}$ is chosen randomly for every iteration j. Moreover, in this way the computation of the gradient becomes more efficient since only based on the smaller minibatch instead of the whole sample space. To find a reasonable termination condition for the algorithm, one could for example consider the absolute change in the cost function after each iteration $|J_j(\bar{\theta}^{(j)}) - J_{j-1}(\bar{\theta}^{(j-1)})|$ and then terminate the algorithm when its value falls below a predefined threshold.

5 Numerical experiments and analysis

In this section, we want to test the previous findings empirically. Instead of using real market data we want to base the analysis on simulated data using stochastic financial models. This idealized approach will allow us to assess and compare the performance of the neural network based method compared to the conventional model-based methods for the estimation of optimal hedging strategies.

5.1 General setting, methodology, and implementation

For our numerical experiments, we want to use the following setting. The agent sells a payoff Z at time $t = 0$ with a maturity of 30 trading days, i.e., we set $T = 30/365$. Furthermore we assume the agent to adjust the hedging strategy on a daily basis so that we have $n = 30$ trading dates $t_k = k/365$ for $k = 0, ..., n$. We can then estimate the hedging strategy at time t_k, i.e., the position of the agent in the hedging instruments, following the approach introduced in Section 4 with $\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}^\theta)$. We set $I_k = S_k$, i.e., the only information the agent is using to determine his hedging strategy are the market prices of the hedging instruments. For the architecture of the neural network F^{θ_k} we set $L = 2, N_0 = 2d, N_1 = N_2 = 24$ and $N_3 = d$, i.e. we have a feedforward neural network with two hidden layers. All networks F^{θ_k} are parameterized by weight matrices A^l and bias vectors b^l which will be optimized according to the optimization problem (10). All parameters are initialized randomly before we start the optimization algorithm. For the activation function σ we chose $\sigma(x) = max(x, 0)$. Although by this choice, σ is not continuously differentiable, modern machine learning libraries are still capable of computing the gradients efficiently similar to the method that was described in Section 4.3. It turned out that this choice of the activation function shows better results for many applications than continuously differentiable functions. Moreover, we apply the technique of so-called batch normalization: At each node of the neural network the input batch is normalized by subtracting its mean and dividing by its standard deviation before applying the activation function. The application of this technique turned out to be necessary in order to deal with the problem of vanishing gradients and thus to achieve reliable results. The phenomenon of vanishing gradients shows up especially for deep neural networks, i.e., networks with a large number of hidden layers. By applying the chain rule multiple times, the gradient may become very small, and the optimization algorithm then fails to converge since there is no significant update of the weights in the front layers, see for example [5]. As shown in [17] the application of batch normalization not only improves the performance of neural networks in general but also allows larger learning rates, i.e., a faster training of the network to a given data set.

In order to assess the performance of the neural network estimated hedging strategy we use the following methodology: At first we generate a data set with sample size $N = 600,000$ and define the underlying finite probability space with $\Omega = \{\omega_1, ..., \omega_N\}$. We simulate sample trajectories $S(\omega_m)$ for $m = 1, ..., N$ of the price process of the available hedging instruments according to the underlying stochastic model. We further assume that the agent sells a path-dependent derivative of the hedging instruments with payoff Z, i.e., there exists an $\mathcal{F}_T$-measurable function $g\colon \mathbb{R}^{d(n+1)} \to \mathbb{R}$, such that $Z = g(S_0, ..., S_T)$. Accordingly we get samples for the payoff Z by evaluating g on the generated trajectories of S. We then split the data into a training sample and a sample for evaluation with sample sizes $N_{train} = 500,000$ and $N_{test} = 100,000$ by setting $\Omega_{train} := \{\omega_1, ..., \omega_{N_{train}}\}$ and $\Omega_{test} = \{\tilde{\omega}_1, ..., \tilde{\omega}_{N_{test}}\} := \{\omega_{N_{train}+1}, ..., \omega_N\}$. To be consistent with the notation in Section 4.3 we set the assigned probability weights respectively to be $P(\{\omega_i\}) = \frac{1}{N_{train}}$ for $i = 1, ..., N_{train}$ and $P(\{\tilde{\omega}_i\}) = \frac{1}{N_{test}}$ for $i = 1, ..., N_{test}$ such that each $(\Omega_{train}, \mathbb{P})$ and $(\Omega_{test}, \mathbb{P})$ define a (finite) probability space. We then use the training data to train the neural network as explained in Section 4.3, i.e., we are solving the optimization problem (11). Therefore, we use the Adam optimization algorithm, that was introduced in [18], with a step size of $\eta = 0.005$. The algorithm is a more sophisticated version of the stochastic gradient descent algorithm as described in Section 4.3. The size of the minibatches is chosen as $N_{batch} = 250$. We do not use a particular termination condition for the optimization algorithm but instead fix the number of iterations to 10^5. Since we set $\rho(X) = CVaR(X)$ we can rewrite the cost function (16) for a given risk level α into

$$J(\theta) = CVaR_\alpha(Z - (\delta^\theta \cdot S)_T + C_T(\delta^\theta))$$

For our experiments, we will set $\alpha = 0.5$ to compare the approach with a model implied strategy and $\alpha = 0.99$, which is a common choice for the risk level in practice. Although we assume the underlying probability space Ω to be finite, we use representation (6) and its natural estimator (7) for the optimization algorithm. This choice is justified since all underlying random variables in our setting have continuous distributions. By optimizing (10), we eventually get a (close-to) optimal parameterization θ for the neural network F^θ that minimizes $J(\theta)$ and thus a (close-to) optimal hedging strategy $\delta_k^\theta(\omega_i) = F^\theta(I(\omega_i), \delta_{k-1}^\theta(\omega_i))$ for each sample trajectory. We then may analyse the sample $-Z(\omega_i) + (\delta^\theta \cdot S)_T(\omega_i) - C_T(\delta^\theta)(\omega_i)$ for $i = 1, ..., N_{train}$. The sample mean is denoted by q_0^θ and represents the average hedging result and by applying (7) on the sample, we get an estimate p_0^θ for the risk indifference price $p(Z)$. We may also use the training data to estimate the risk neutral price q_0 of the underlying model for the payoff Z by estimating the mean of the sample $(Z(\omega_1), ..., Z(\omega_{N_{train}}))$. Subsequently we use the test data set as the input for the trained neural network to get a optimal hedging strategy $\delta^\theta(\tilde{\omega}_i)$ for each trajectory of the price process $S(\tilde{\omega}_i)$ for $i = 1, ..., N_{test}$. We may also compute a model-implied hedge for each trajectory of the price process S at time t_k, denoted with $\delta_k^H(\tilde{\omega}_i)$. In this way we can evaluate the out-of-sample performance of the neural network approach and compare it to the performance of the model hedge by evaluating the sample of the realized profits for both methods, which is given with:

$$PnL_T(Z, p_0, \delta)(\tilde{\omega}_i) := -Z(\tilde{\omega}_i) + p_0 + (\delta \cdot S)_T(\tilde{\omega}_i) - C_T(\delta)(\tilde{\omega}_i) \quad \text{for } i = 1, ..., N_{test} \quad (20)$$

Here p_0 represents any price that may have been charged for the payoff Z, e.g. the risk indifference price $p(Z)$ or the risk neutral price $q(Z) = q_0$. We will evaluate several statistics for the sample in order to assess the performance of our approach.

The sample trajectories of the price process S are generated according to two different

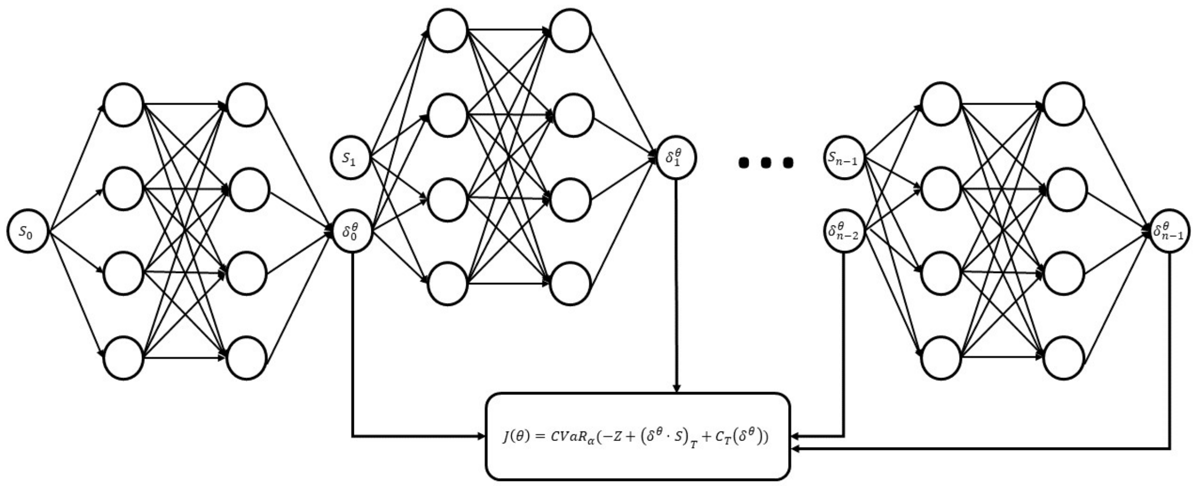

Figure 1: Exemplary illustration of the utilized neural network model

market models. At first, we consider a simple Black-Scholes-Merton model, i.e., the stock price itself is the only risk factor, and the agent will accordingly only trade in the stock itself to hedge the derivative. Since the Black-Scholes-Merton model does not only offer closed-form solutions for the price of European options but also for the Greeks, we use this setting to compare the performance of the neural network based approach with a conventional model hedge. Subsequently, we assume the price process S to follow a Heston model. Here we will have a second risk-factor since the volatility by itself is also assumed to follow a stochastic dynamic. In this setting, we can then assess if the neural network based approach is also capable of making reasonable hedging decisions if there is more than risk factor to hedge and therefore the agent has to trade two different assets. For simplicity reasons we directly consider both models under a risk-neutral measure $\mathbb{Q}$, i.e., the hedging strategy will be based on market anticipations of future prices. Furthermore, we also assume the risk-free to be zero in both models, i.e., $r = 0$. The two models will be described in more detail in Sections 5.2 and 5.3, respectively. All of the algorithms are implemented in Python. To build and train the neural networks we use Tensorflow, the machine learning library of Google. Furthermore, we used an implementation of the Heston pricer in Mathwork's Matlab to verify the risk-neutral prices in the setup of the Heston model, which we estimated with a Monte Carlo method.

5.2 Black-Scholes-Merton model

In the Black-Scholes-Merton (BSM) model the price process is assumed to follow the following dynamic:

$$dS_t = \sqrt{V} S_t dB_t \quad \text{for} \;\; t > 0 \text{ and } S_0 = s_0 \tag{21}$$

where B is a one-dimensional Browanian motion under $\mathbb{Q}$ and $V > 0$ constant. For a typical equity market scenario we choose $V = 0.04$ and $s_0 = 100$. By applying Itô's formula to $f(S_t) = \ln(S_t)$ we can find the solution for (21) with:

$$S_t = s_0 \exp\left(-\frac{V}{2}t + \sqrt{V}B_t\right) \tag{22}$$

The risk neutral price at time t for a payoff $Z = g(S_0, ..., S_t)$ that occurs at time T can then be expressed by:

$$q_t = \mathbb{E}_{\mathbb{Q}}[g(S_0, ..., S_T)|\mathcal{F}_t]$$

For an European style option there is an analytic solution for this expectation as shown in [6]. For example the risk neutral price at time t of an European Call option with strike K is given by choosing $g(S_T) = (S_T - K)^+$ with

$$q_t = \mathbb{E}_{\mathbb{Q}}[(S_T - K)^+|\mathcal{F}_t] = S_t\Phi(d_1(t, S_t)) - K\Phi(d_2(t, S_t)) \tag{23}$$

where $d_1(t, s) = \frac{\ln(\frac{s}{K}) + \frac{V}{2}(T-t)}{V\sqrt{(T-t)}}$, $d_2(t, s) = d_1 - V\sqrt{(T-t)}$ and $\Phi(\cdot)$ is the cumulative distribution function of a standard normally distributed random variable.

Since the price process S satisfy the Markov property it holds that $q_t = u(t, S_t)$ for some $u : [0, T] \times [0, \infty) \to \mathbb{R}$. We may now denote the value of a portfolio of an agent selling the liability Z with maturity T at time t with Π_t. Since we assume $r = 0$ the agent only has to trade the underlying stock, the amount of shares he holds at time t is denoted by δ_t and thus it holds that $\Pi_t = -u(t, S_t) + q_0 + \delta_t S_t$. Since in the Black-Scholes-Merton framework the market is assumed to be complete, we can determine δ_t such that $\Pi_T = 0$ under the assumption that continuous time trading is possible. The dynamic of the portfolio value is accordingly given by: $d\Pi_t = -du(t, S_t) + \delta_t dS_t$. Under the assumption that $u \in C^{1,2}$, we may apply Itô's formula to $u(t, S_t)$ to obtain

$$du(t, S_t) = \frac{\partial u}{\partial s} dS_t + \frac{\partial u}{\partial t} dt + \frac{1}{2} \frac{\partial^2 u}{\partial s^2} dS_t dS_t = \frac{\partial u}{\partial s} dS_t + \left[\frac{\partial u}{\partial t} dt + \frac{V S_t^2}{2} \frac{\partial^2 u}{\partial s^2} \right] dt$$

Since the agent wants to be neutral to changes in the stock price, we set $\delta_t = \frac{\partial}{\partial s} u(t, S_t)$. With an no-arbitrage argument we can further conclude that $d\Pi_t = r dt = 0$ for all $t \in [0, T]$ and thus the following representation holds:

$$g(S_T) = q_0 + \int_0^T \delta_t dS_t.$$

We will choose δ_t, i.e. the partial derivative of the price of the option with respect to the underlying stock price as our model hedge δ_t^H at time t and it can be easily shown that for an European call option it holds $\delta_t^H = \Phi(d_1(t, S_t))$ and thus $0 < \delta_t^H < 1$, see for example [16].

For our numerical experiments we need to discretize the above described model. To generate the sample trajectories for the price process S we use solution (22) of the SDE (21). We can then simulate a sample trajectory of the price process S with:

$$\ln(S_k(\omega_i)) = \ln(S_{k-1}(\omega_i)) - \frac{V}{2} \Delta(k) + \sqrt{V \Delta(k)} X_k(\omega_i)$$

where $S_0(\omega_i) = s_0$, $\Delta(k) = t_k - t_{k-1}$ and $X_k(\omega_i) \overset{\text{iid}}{\sim} N(0, 1)$ for $k = 1, ..., n$.

5.3 Heston Model

The Heston model is an extension of the Black-Scholes-Merton model. Here the variance of the stock price process is no longer assumed to be constant. Instead it is driven by a Cox-Ingersoll-Ross (CIR) process. The model is specified by the following differential equations:

$$\begin{aligned} dS_t^1 &= \sqrt{V_t} S_t^1 dB_t \quad \text{ for } \; t > 0 \text{ and } S_0^1 = s_0 \\ dV_t &= \kappa(\nu - V_t)dt + \sigma\sqrt{V_t} dW_t \quad \text{ for } \; t > 0 \text{ and } V_0 = v_0 \end{aligned} \tag{24}$$

where B and W are again one-dimensional Brownian motions under $\mathbb{Q}$ with correlation $\rho \in [-1, 1]$ and κ, ν, σ, v_0 and s_0 are positive constants. In order to ensure that $V_t > 0$ for all $t \in [0, T]$, it has to hold that $2\kappa\nu > \sigma^2$. This is often referred to as the Feller condition, see for example [2]. To simulate a typical equity market scenario we choose $\rho = -0.7, \kappa = 1, \nu = 0.04, \sigma = 0.25, v_0 = 0.04$ and $s_0 = 100$. In this market there are two risk factors and it is not sufficient to trade the underlying stock in order to optimally hedge a derivative on S^1. One also has to hedge the second risk factor, namely the volatility risk. The variance V is not tradeable directly, but only by derivatives on V. Therefore, we assume that there is a variance swap S^2 with maturity T traded on the market. Its price is given by (see for example [9]):

$$S_t^2 := \mathbb{E}_{\mathbb{Q}} \left[\int_0^T V_s ds \mid \mathcal{F}_t \right], \quad t \in [0, T] \tag{25}$$

The 2-dimensional process $S = (S^1, S^2)$ describes the prices of the two tradeable assets on the market.

Lemma 5.1. The price of a variance swap as defined in (25) is given by:

$$S_t^2 = \int_0^t V_s ds + L(t, V_t) \quad \text{with } L(t, V_t) := \nu(T - t) + \frac{(1 - e^{-\kappa(T-t)})}{\kappa}(V_t - \nu) \tag{26}$$

Proof. To solve (25) we start by solving the second SDE in (24):

$$
\begin{aligned}
& dV_t = \kappa(\nu - V_t)dt + \sigma\sqrt{V_t}dW_t \\
\Leftrightarrow\quad & e^{\kappa t}(dV_t + \kappa V_t dt) = e^{\kappa t}(\kappa\nu dt + \sigma\sqrt{V_t}dW_t) \\
\Leftrightarrow\quad & d(e^{\kappa t}V_t) = e^{\kappa t}\kappa\nu dt + e^{\kappa t}\sigma\sqrt{V_t}dW_t \\
\Leftrightarrow\quad & e^{\kappa t}V_t = v_0 + \int_0^t e^{\kappa s}\kappa\nu ds + \sigma\int_0^t e^{\kappa s}\sqrt{V_s}dW_s \\
\Leftrightarrow\quad & e^{\kappa t}V_t = v_0 + \nu(e^{\kappa t} - 1) + \sigma\int_0^t e^{\kappa s}\sqrt{V_s}dW_s \\
\Leftrightarrow\quad & V_t = \nu + (v_0 - \nu)e^{-\kappa t} + \sigma e^{-\kappa t}\int_0^t e^{\kappa s}\sqrt{V_s}dW_s
\end{aligned}
$$

We can now use the explicit form for V_t to solve (25) by:

$$
\begin{aligned}
S_t^2 =& \mathbb{E}_{\mathbb{Q}}\left[\int_0^T V_s ds \mid \mathcal{F}_t\right] = \mathbb{E}_{\mathbb{Q}}\left[\int_0^t V_s ds + \int_t^T V_s ds \mid \mathcal{F}_t\right] \\
=& \int_0^t V_s ds + \mathbb{E}_{\mathbb{Q}}\left[\int_t^T \left(\nu + (v_0 - \nu)e^{-\kappa t} + \sigma e^{-\kappa s}\int_0^s e^{\kappa r}\sqrt{V_r}dW_r\right) ds \mid \mathcal{F}_t\right] \\
=& \int_0^t V_s ds + \nu(T - t) + \frac{e^{-\kappa t} - e^{-\kappa T}}{\kappa}(v_0 - \nu) \\
& + \mathbb{E}_{\mathbb{Q}}\left[\int_t^T \left(\sigma e^{-\kappa s}\int_0^s e^{\kappa r}\sqrt{V_r}dW_r\right) ds \mid \mathcal{F}_t\right].
\end{aligned}
\tag{27}
$$

For the second part of the sum we can change the order of integration to obtain:

$$
\begin{aligned}
& \mathbb{E}_{\mathbb{Q}}\left[\int_t^T \left(\sigma e^{-\kappa s}\int_0^s e^{\kappa r}\sqrt{V_r}\ dW_r\right) ds \mid \mathcal{F}_t\right] \\
=& \mathbb{E}_{\mathbb{Q}}\left[\int_0^t \int_r^T \sigma e^{-\kappa s} e^{\kappa r}\sqrt{V_r}\ ds\ dW_r + \int_t^T \int_r^T \sigma e^{-\kappa s} e^{\kappa r}\sqrt{V_r}\ ds dW_r \mid \mathcal{F}_t\right] \\
=& \mathbb{E}_{\mathbb{Q}}\left[\int_0^t \frac{e^{-\kappa t} - e^{-\kappa T}}{\kappa}\sigma e^{\kappa r}\sqrt{V_r}\ dW_r + \int_t^T \frac{e^{-\kappa t} - e^{-\kappa T}}{\kappa}\sigma e^{\kappa r}\sqrt{V_r}\ dW_r \mid \mathcal{F}_t\right] \\
=& \int_0^t \frac{e^{-\kappa t} - e^{-\kappa T}}{\kappa}\sigma e^{\kappa r}\sqrt{V_r}\ dW_r + \mathbb{E}_{\mathbb{Q}}\left[\int_t^T \frac{e^{-\kappa t} - e^{-\kappa T}}{\kappa}\sigma e^{\kappa r}\sqrt{V_r}\ dW_r \mid \mathcal{F}_t\right] \\
=& \frac{e^{-\kappa t}(1 - e^{-\kappa(T-t)})}{\kappa}\sigma\int_0^t e^{\kappa r}\sqrt{V_r}\ dW_r
\end{aligned}
$$

Substituting this back into (27) gives us:

$$
\begin{aligned}
S_t^2 &= \int_0^t V_s ds + \nu(T-t) + \frac{e^{-\kappa t}(1-e^{-\kappa(T-t)})}{\kappa}(v_0 - \nu) \\
&\quad + \frac{e^{-\kappa t}(1-e^{-\kappa(T-t)})}{\kappa}\sigma \int_0^t e^{\kappa r}\sqrt{V_r}\, dW_r \\
&= \int_0^t V_s ds + \nu(T-t) + \frac{(1-e^{-\kappa(T-t)})}{\kappa}\left((v_0-\nu)e^{-\kappa t} + \sigma e^{-\kappa t}\int_0^t e^{\kappa r}\sqrt{V_r}\, dW_r\right) \\
&= \int_0^t V_s ds + \nu(T-t) + \frac{(1-e^{-\kappa(T-t)})}{\kappa}(V_t - \nu)
\end{aligned}
$$

□

We consider again an European style option on S^1, i.e., the payoff at time T is given by $g(S_T^1)$ for some $g\colon \mathbb{R} \to \mathbb{R}$. The risk neutral price at time t is given by $q_t := \mathbb{E}_{\mathbb{Q}}[g(S_T^1)|\mathcal{F}_t]$. Since the process (S^1, V) satisfy the Markov property, there exists a function $u\colon [0,T] \times [0,\infty)^2 \to \mathbb{R}$ such that $q_t = u(t, S_t^1, V_t)$. In order to show that the payoff of the option $g(S_T^1)$ is replicable by trading in $S = (S^1, S^2)$, we define similarly to Section 5.2 a portfolio that consists of a short position in the derivative, δ_t^1 shares of the underlying stock and δ_t^2 shares of the variance swap. The portfolio value at time t is $\Pi_t = -u(t, S_t, V_t) + q_0 + \delta_t^1 S_t^1 + \delta_t^2 S_t^2$. The dynamic of the portfolio value is hence given with:

$$
d\Pi_t = -du(t, S_t, V_t) + \delta_t^1 dS_t^1 + \delta_t^2 dS_t^2. \tag{28}
$$

At first, we observe from (26) and by applying Itô's formula that

$$
\begin{aligned}
dS_t^2 =& V_t dt + dL(t, V_t) \\
=& V_t dt + \frac{\partial L}{\partial t}dt + \frac{\partial L}{\partial v}dV_t + \frac{1}{2}\frac{\partial^2 L}{\partial v^2}\sigma^2 V_t dt = \frac{\partial L}{\partial v}dV_t + \left[V_t + \frac{\partial L}{\partial t}\right] dt.
\end{aligned} \tag{29}
$$

Substituting (29) into (28) leads to:

$$
d\Pi_t = -du(t, S_t^1, V_t) + \delta_t^1 dS_t^1 + \delta_t^2 \frac{\partial L}{\partial v} dV_t + \delta_t^2 \left[V_t + \frac{\partial L}{\partial t}\right] dt. \tag{30}
$$

Next we assume $u \in C^{1,2,2}$ and apply Itô's formula to obtain:

$$\begin{aligned} du(t, S_t^1, V_t) =& \frac{\partial u}{\partial s} dS_t^1 + \frac{\partial u}{\partial v} dV_t + \frac{\partial u}{\partial t} dt \\ &+ \frac{1}{2} \frac{\partial^2 u}{\partial s^2} dS_t^1 dS_t^1 + \frac{1}{2} \frac{\partial^2 u}{\partial v^2} dV_t dV_t + \frac{1}{2} \frac{\partial^2 u}{\partial s \partial v} dS_t^1 dV_t \\ =& \frac{\partial u}{\partial s} dS_t^1 + \frac{\partial u}{\partial v} dV_t \\ &+ \frac{1}{2} \left[\frac{\partial u}{\partial t} dt + \frac{\partial^2 u}{\partial s^2} V_t (S_t^1)^2 + \frac{\partial^2 u}{\partial v^2} \sigma^2 V_t + \frac{\partial^2 u}{\partial s \partial v} \rho \sigma S_t^1 V_t \right] dt \end{aligned} \tag{31}$$

Since the agent wants to be neutral with respect to both risk factors we match the coefficients in (30) and (31) and accordingly set $\delta_t^1 = \frac{\partial}{\partial s} u(t, S_t^1, V_t)$ and $\delta_t^2 = \frac{\partial u(t, S_t^1, V_t)/\partial v}{\partial L(t, V_t)/\partial v}$. We may now use again a no-arbitrage argument to conclude that $d\Pi_t = r dt = 0$ for all $t \in [0, T]$. Therefore, under the assumption that continuous time trading was possible the payoff of the option at time T is replicable and the following representation holds:

$$g(S_T^1) = q_0 + \int_0^T \delta_t^1 dS_t^1 + \int_0^T \delta_t^2 dS_t^2.$$

For the numerical experiments we use an Euler discretization of (24) as for example proposed in [3]. To increase the accuracy of the simulation we use the following approach: For the generation of the sample trajectories of the stock price and the variance process we choose a finer discretization $0 = t_0, ..., t_{\tilde{n}} = T$ of the time interval $[0, T]$ with $\tilde{n} = 10n$ and then afterwards choose the appropriate subset such that it matches the trading dates t_k for $k = 0, ..., n = 30$. We simulate a sample trajectory of the stock price process S^1 and the variance process V with:

$$\begin{aligned} \ln(S_k^1(\omega_i)) =& \ln(S_{k-1}^1(\omega_i)) - \frac{V_{k-1}(\omega_i)}{2} \Delta(k) + \sqrt{V_{k-1}(\omega_i) \Delta(k)} X_k(\omega_i) \\ V_k(\omega_i) =& V_{k-1}(\omega_i) + \kappa(\nu - V_{k-1}(\omega_i)) \Delta(k) + \sigma \sqrt{V_{k-1}(\omega_i) \Delta(k)} Y_k(\omega_i) \end{aligned}$$

where $S_0^1(\omega_i) = s_0$, $V_0(\omega_i) = v_0$, $\Delta(k) = t_k - t_{k-1}$, $X_k(\omega_i) \overset{\text{iid}}{\sim} N(0,1)$ and $Y_k(\omega_i) := \rho X_k(\omega_i) + \sqrt{1 - \rho^2}\ \tilde{Y}_k(\omega_i)$ with $\tilde{Y}_k(\omega_i) \overset{\text{iid}}{\sim} N(0,1)$ and $\tilde{Y}_k, X_k$ independent for all $k = 0, ..., \tilde{n}$.

Finally, we can simulate the price process of the variance swap S^2 by using the following discretization of (25):

$$S_k^2(\omega_i) = \sum_{j=0}^{k} (V_j(\omega_i) \cdot \Delta(j)) + L(t_k, V_k(\omega_i)) \quad \text{for } k = 0, ..., \tilde{n} \tag{32}$$

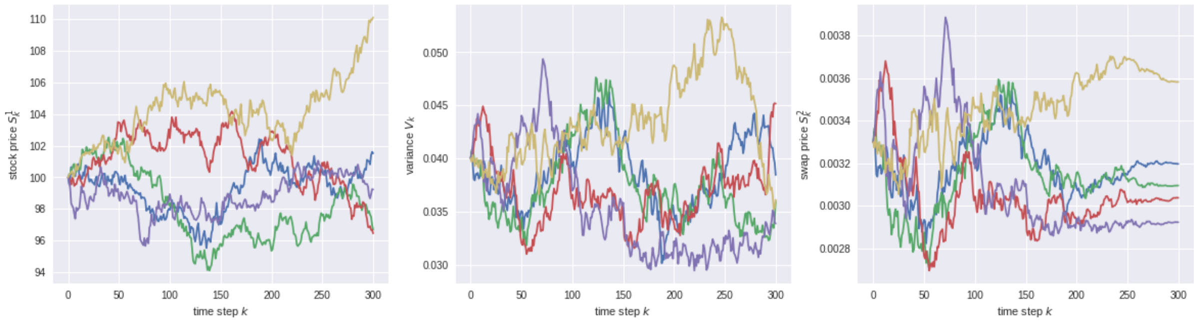

Figure 2: Illustration of 5 different trajectories of the stock price process S^1, the variance process V and the variance swap price process S^2

Remark 5.2. In order to illustrate the capability of the discussed approach of estimating optimal hedging strategies for any kind of derivatives we also want to apply our algorithm to more complex payoffs. For that purpose we consider additionally the following derivatives:

(i) Asian style call option with the payoff function:

$$g(S_1^1, ..., S_T^1) = (\bar{S} - K)^+ \quad \text{with } \bar{S} := \frac{1}{n} \sum_{k=1}^{n} S_k^1$$

(ii) Russian style call option with the payoff function:

$$g(S_1^1, ..., S_T^1) = (\hat{S} - K)^+ \quad \text{with } \hat{S} := \max_{k=1,...,n} S_k^1$$

5.4 Results and analysis

In this section, we want to present, analyze and interpret the results of the different numerical experiments we carried out in order to assess the performance of the introduced neural network-based approach for the estimation of optimal hedging strategies.

At first, we consider a European call option with strike $K = 100$ under the Black-Scholes-Merton model in the previously described setting. The risk-neutral price of this option is given according to (23) with $q_0 = 2.29$. We get an average hedging result for the neural network based strategy of $q_0^\theta = 2.29$. Therefore, q_0^θ can also be used as a valid approximation for q_0 in this setting. For the risk indifference price we get $p_0^\theta = 2.59$ for $\alpha = 0.5$ and $p_0^\theta = 3.42$ for $\alpha = 0.99$. Note that these prices are consistent with Theorem 3.8 (ii). The out-of-sample statistics for the PnL_T as defined in (20) and the corresponding values for the model strategy are presented in Table 1. In order to make the results comparable, we assume that for all strategies the risk-neutral price q_0 was charged. As one can see the performance of the model and the neural network based approach for $\alpha = 0.5$ are very similar. While the average PnL_T is close to 0 for both methods, the model strategy shows slightly better results for standard deviation, skewness and realized $CVaR$. If we compare the strategies themselves, i.e we consider δ_k^H and δ_k^θ for a fixed time t_k (Figure 4) or one particular sample trajectory (Figure 5), we see that the neural network based strategy is almost perfectly replicating the model hedge. Although we did not implement trading constraints, the neural network based strategy still stays between 0 and 1, with some exceptions at the end of the trading period. These results indicate that our approach can find a good approximation for the optimization problem (1). Considering the neural network based strategy for $\alpha = 0.99$, we can see one of the advantages of our approach. While the average PnL_T is still close to 0, the method can reduce the realized $CVaR_{0.99}$ significantly at the expanse of a higher standard deviation. While we saw left-skewed distributions of the realized profits for the model and $CVaR_{0.5}$ based strategies we now have a right-skewed distribution (see Figure 3). This finding is consistent to the fact, that the strategy aims to reduce the weight of the left tail. Considering δ_k^θ for a fixed time t_k directly, we find that the strategy now deviates stronger from the model strategy as for $\alpha = 0.5$. In general, the strategy acts more conservative compared to the $CVaR_{0.5}$-strategy. It holds a higher amount of shares for out-of-the-money options. If the option is in-the-money, it holds a slightly smaller position in the stock, which reverses as soon as the option is going deep-in-the-money and the strategy may even hold more than one share of the stock, i.e., $\delta_k^\theta > 1$. Considering a particular sample trajectory (Figure 5) we see that the strategy is tracking the model hedge closely as long as the option stays at-the-money but is holding a higher number of shares than the model hedge as the option falls out-of-the-money.

Method	Mean	SD	Skewness	real $CVaR_{0.5}$	real $CVaR_{0.99}$
Model	-0.0001	0.3600	-0.3086	0.2701	1.2557
$CVaR_{0.5}$	-0.0019	0.3959	-0.3424	0.3035	1.3341
$CVaR_{0.99}$	-0.0009	0.4640	0.3692	0.3658	1.1445

Table 1: Statistics of realized profits - Black-Scholes-Merton model

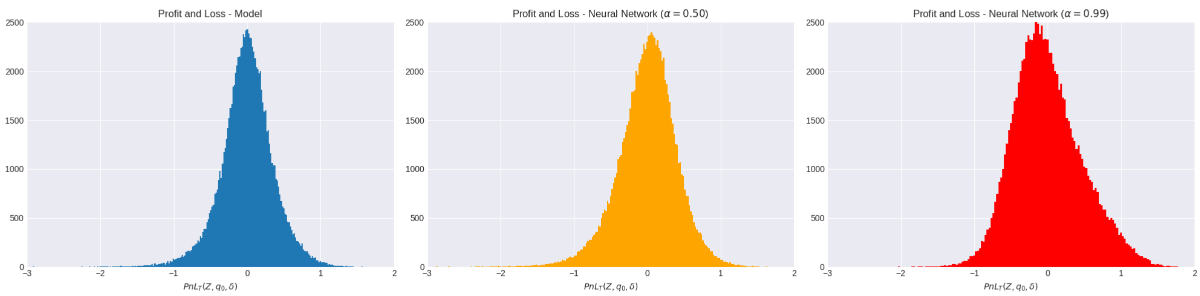

Figure 3: Profit and Loss - Black-Scholes-Merton model

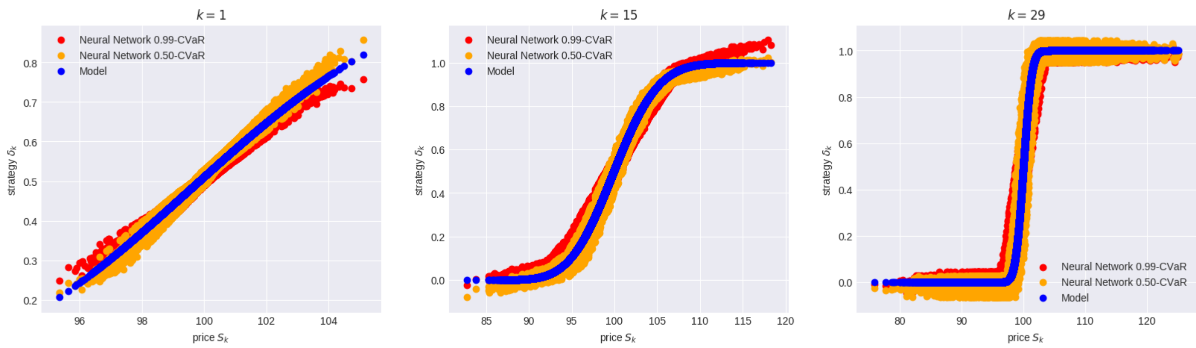

Figure 4: Hedging strategies at fixed timepoint t_k - Black-Scholes-Merton model

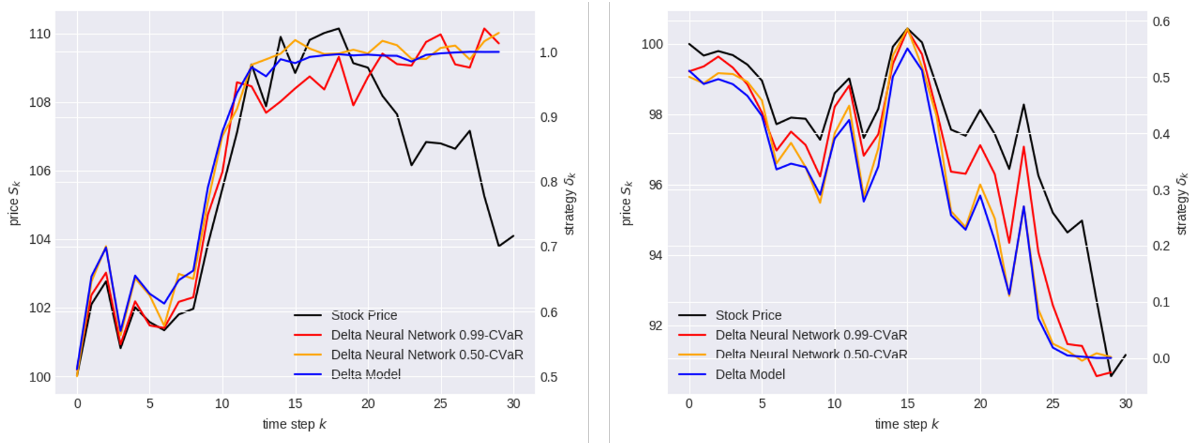

Figure 5: Strategy trajectories for two different scenarios - Black-Scholes-Merton model

We also want to highlight that in this setting, i.e. without any transaction costs, we expect that we can simplify the structure of the neural network by replacing the recurrent architecture $\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}^\theta)$ with $\delta_k^\theta = F^{\theta_k}(I_k)$ based on the considerations in Section 5.2. Indeed, we could confirm with our numerical experiments that we get similar results with the simplified network architecture. This does not hold if we include trading costs into our model. We consider proportional transaction costs as defined in Section 2.2 with constant $c_k = c$ for $k = 0, ..., n-1$. Thus, the total costs of a strategy δ are given by $C_T(\delta) = \sum_{k=0}^{n} cS_k|\delta_k - \delta_{k-1}|$. We choose very small proportional transaction costs and set $c = 2^{-8}$. In this setting our approach can again prove its power. We consider the model based strategy and the strategies based on the two different network architectures. We estimate the following prices with the recurrent neural networks: $q_0^\theta = 2.94$ and $p_0^\theta = 3.39$ for $\alpha = 0.5$ and $q_0^\theta = 3.17$ and $p_0^\theta = 4.44$ for $\alpha = 0.99$. For further analysis, we again assume that the risk-neutral price q_0 was charged. By considering the statistics in Table 2, we see that the model strategy causes the highest costs and consequently also the highest loss on average. Additionally, the distribution of the profits is strongly left-skewed and thereby it exhibits extreme values for the realized $CVaR$. The performance of the neural network based strategies is significantly better considering the mean hedging error as well as the realized $CVaR$. Both strategies manage to achieve a right-skewed distribution for the profits and can thereby reduce the weight on the left tails, see also Figure 6. We further note that the strategies based on the recurrent network structure exhibits better statistics than the one based on the simpler architecture. Hence, for a setting with incorporated market frictions such as trading costs, we indeed need additional information about previous hedging decisions to find an optimal strategy. In Figure 7 we can see the distribution of the realized costs of the model-based and the recurrent neural network-based strategy with $\alpha = 0.5$. While the distribution for the model strategy has its mode around a total cost of $C_T(\delta) = 1.1$, the neural network based strategy has its mode at $C_T(\delta) = 0.8$ and additionally manages to hedge a significant number of scenarios with lower costs (local maximum at $C_T(\delta) = 0.4$). If we consider the two strategies δ_k^θ and δ_k^H for a fixed time t_k, we find that the neural network, in general, reduces its positions compared to the setting without transaction costs (see Figure 8). We note that while the strategy of the simple network appears to be a functional of the stock price this is not true for the recurrent structure, i.e. the additional input of the previous position affects the trading decision. Moreover, if we consider a single trajectory of the strategies, we note that in particular the recurrent neural network based strategy reduces the turnover in order to avoid hedging costs while still tracking the model strategy to some extent. Based on these promising results we now want to investigate the relationship between the proportional cost c and the risk indifference price for $\alpha = 0.99$, similar to [7], where we can find an analysis with respect to the utility indifference price. Therefore, we consider different proportional costs $c_{(i)} = 2^{-(i+4)}$ for $i = 1, ..., 6$ and denote the corresponding risk indifference price as defined in (2) with $p_{c_{(i)}} = p_{c_{(i)}}(Z)$. As illustrated in Figure 10 we find by considering the pairs $(\ln(c_{(i)}), \ln(p_{c_{(i)}} - p_0))$ the following linear relationship: $\ln(p_c - p_0) = 0.40 \ln(c) + 2.19$. This indicates that at least for small c we have the following asymptotic behavior for the risk indifference price: $p_c - p_0 = O(c^{2/5})$. We also tested our approach in a market with fixed transaction costs as defined in Section 2.2 and got comparable results. In general, we can conclude that our approach can make reasonable hedging decisions in a market with simplistic transaction costs.

Next, we want to assess the neural network based approach in the Heston model as defined in Section 5.3. We again consider a European call option with strike $K = 100$. We estimate the risk-neutral price of the option with a Monte Carlo simulation and

Method	Mean	SD	Skewness	real $CVaR_{0.5}$	real $CVaR_{0.99}$	Aver. cost
Model	-1.0709	0.5015	-0.7173	1.4595	2.8148	1.0704
$CVaR_{0.50}$ (recurrent)	-0.6506	0.6464	-1.5997	1.1076	3.6368	0.6512
$CVaR_{0.50}$ (simple)	-0.7801	0.6423	-1.4433	1.2378	3.6631	0.7791
$CVaR_{0.99}$ (recurrent)	-0.8857	0.5319	0.3340	1.3101	2.1487	0.8880
$CVaR_{0.99}$ (simple)	-0.9571	0.5411	0.3320	1.3895	2.2149	0.9582

Table 2: Statistics of realized profits - Black-Scholes-Merton model with proportional transaction costs

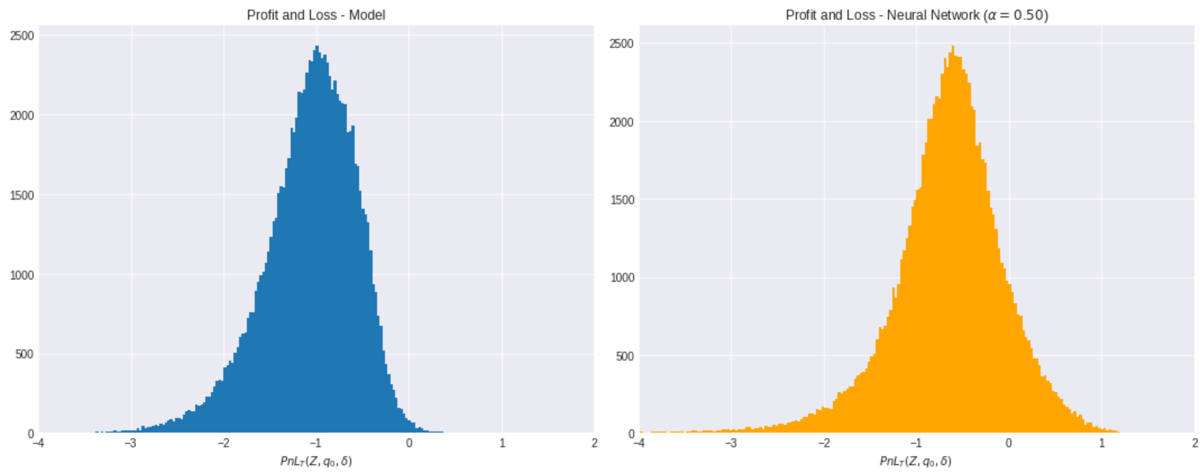

Figure 6: Profit and Loss - Black-Scholes-Merton model with proportional transaction costs

get $q_0 = 2.27$. With the neural network based algorithm we get the following prices: $q_0^\theta = 2.27$, $p_0^\theta = 2.58$ for $\alpha = 0.5$ and $p_0^\theta = 3.52$ for $\alpha = 0.99$. If we compare the different statistics (Table 3) and the distribution of the realized profits (Figure 11) we see similar results as for the Black-Scholes-Merton model. While the average hedging results are for both risk levels close to 0, the distribution for $\alpha = 0.50$ exhibits a smaller value for standard deviation, is left-skewed and minimizes the realized $CVaR_{0.5}$. In contrast, we see for $\alpha = 0.99$ a right-skewed distribution of the profits that minimizes the realized $CVaR_{0.99}$ at the cost of a higher standard deviation. Considering δ_k^1 for a fixed time t_k (Figure 12) or for one particular sample trajectory (Figure 14), we also get a comparable picture as for the Black-Scholes-Merton model. While the $CVaR_{0.99}$-strategy holds slightly smaller

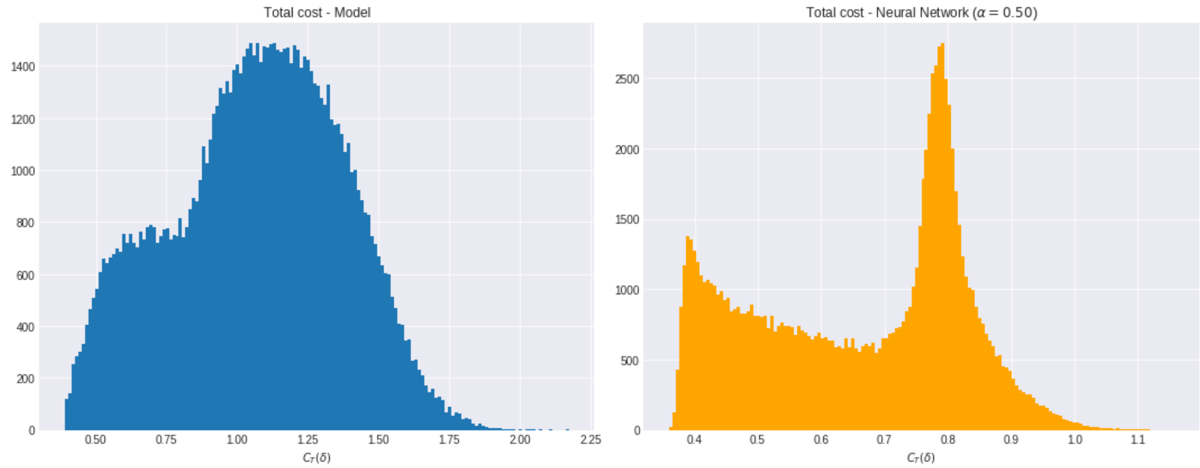

Figure 7: Realized total costs of the hedging strategies - Black-Scholes-Merton model with proportional transaction costs

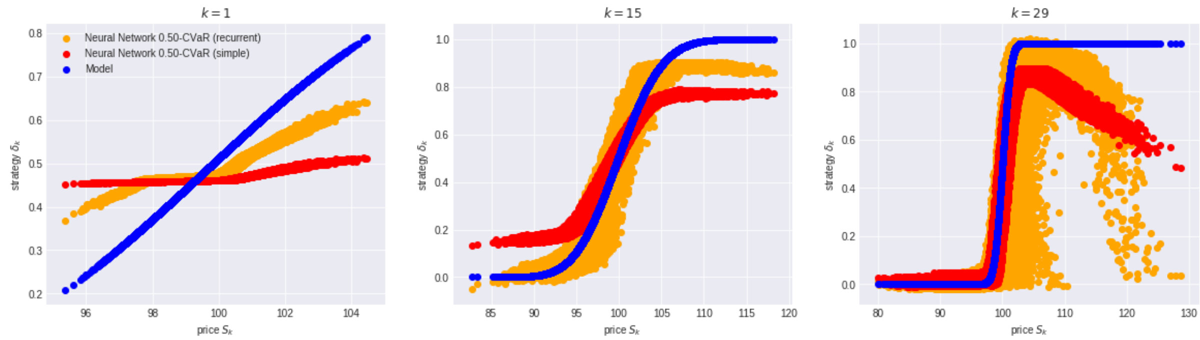

Figure 8: Hedging strategies at fixed time step t_k - Black-Scholes-Merton model with proportional transaction costs

positions for in-the-money options, it acts more conservative for out-of-the-money options and holds larger positions in the stock. If we consider the position in the variance swap δ_k^2 it is difficult to retrace the trading decisions of the neural network, especially since the volatility risk for a call option is minimal compared to the risk of the price dynamics of the underlying asset. In general, the strategy only holds an insignificant position in the swap, the amount of shares does not appear to be correlated with the price process of the swap S^2, but with the stock price S^1, see Figure 13. As shown in Figure 14 the strategy reduces its position in the variance swap over the trading period, what is consistent with the proportional relationship between volatility risk and the time to maturity of the option. Finally, we want to demonstrate the capability of our approach by applying it to more complex payoffs. Therefore, we use the in Remark 5.2 defined payoffs. For the Asian style call option with strike $K = 100$ we get the risk-neutral price $q_0 = 1.35$ and

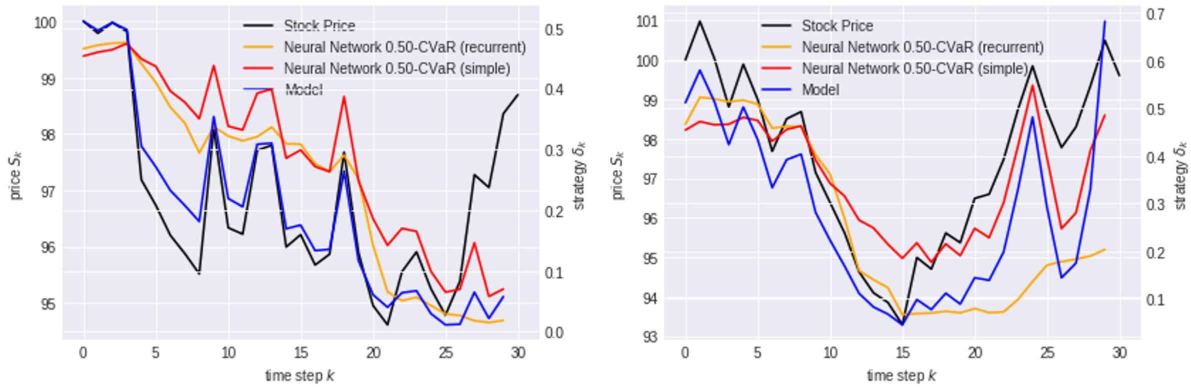

Figure 9: Strategy trajectories for two different scenarios - Black-Scholes-Merton model with proportional transaction costs

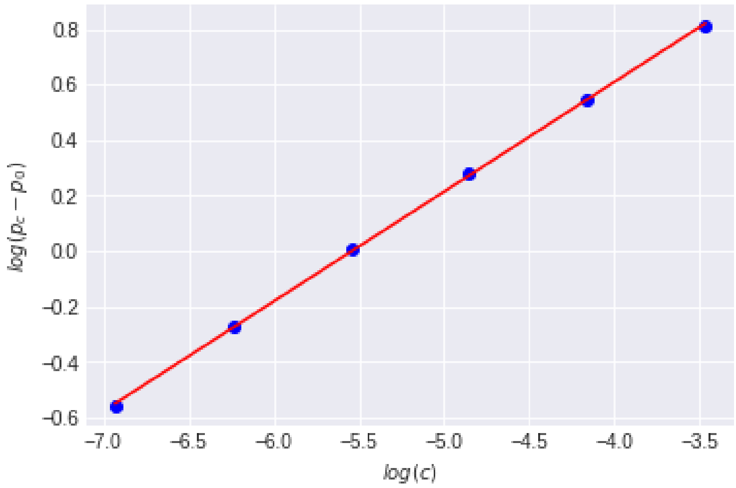

Figure 10: Asymptotics of the risk indifference price for $\alpha = 0.99$ - Black-Scholes-Merton model with proportional transaction costs

the prices implied by the neural network are given by $q_0^\theta = 1.35$, $p_0^\theta = 1.57$ for $\alpha = 0.5$ and $p_0^\theta = 2.25$ for $\alpha = 0.99$. For the Russian style call option with strike $K = 100$ we get the risk-neutral price $q_0 = 3.90$ and the prices implied by the neural network are given by $q_0^\theta = 3.90$, $p_0^\theta = 4.41$ for $\alpha = 0.5$ and $p_0^\theta = 5.99$ for $\alpha = 0.99$. The hedging results are for both payoffs similar to the previously discussed vanilla option. Thus, the approach shows good performance also in the hedging for more complex payoffs structures. Moreover, we find that the strategy holds larger positions in the variance swap for the Russian option than for the Asian option. This reflects the higher volatility risk that the first mentioned payoff exhibits and indicates, that our approach appears to be able of dealing with a second risk factor and making reasonable hedging decisions also in more a complex market.

Method	Mean	SD	Skewness	real $CVaR_{0.5}$	real $CVaR_{0.99}$
$CVaR_{0.5}$	-0.0019	0.3959	-0.3424	0.3035	1.3341
$CVaR_{0.99}$	-0.0036	0.5141	0.3346	0.4122	1.2561

Table 3: Statistics of realized profits - Heston model

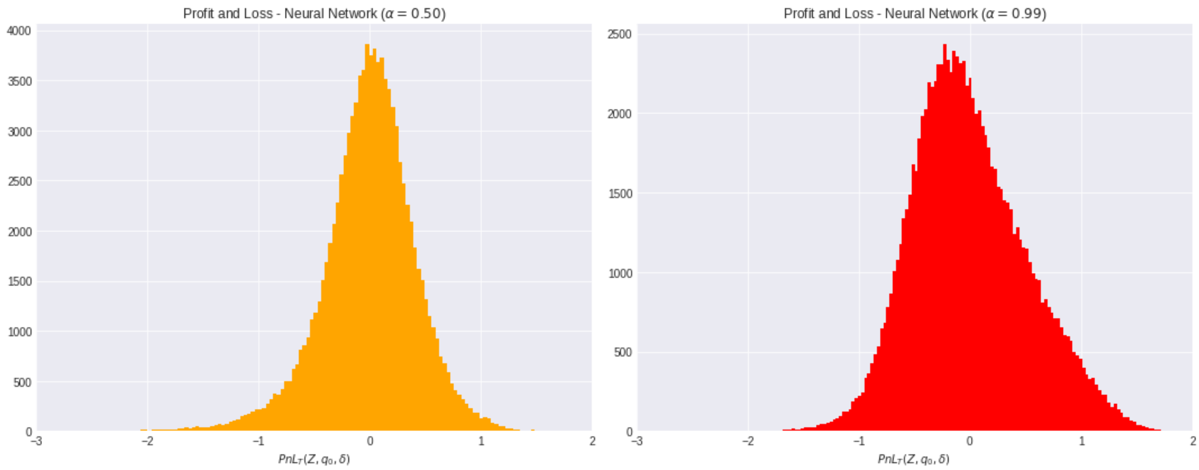

Figure 11: Profit and Loss - Heston model

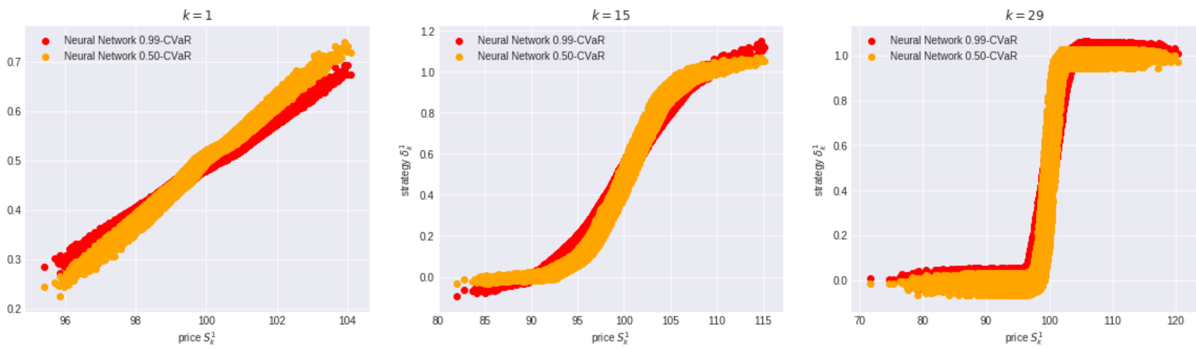

Figure 12: Hedging strategies δ_k^1 at fixed timepoint t_k - Heston model

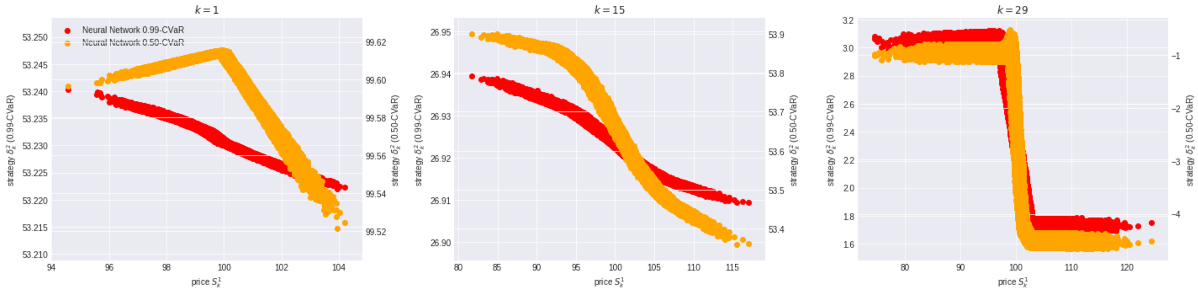

Figure 13: Hedging strategies δ_k^2 at fixed timepoint t_k - Heston model

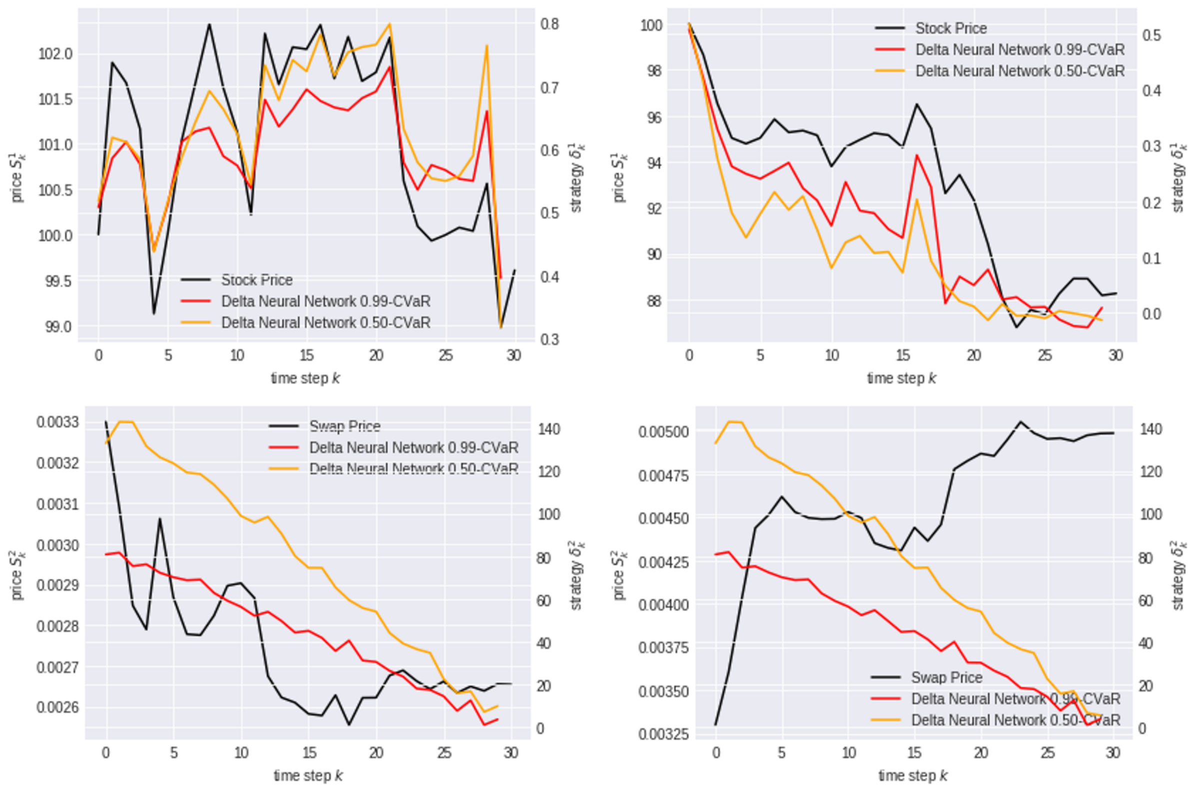

Figure 14: Strategy trajectories for two different scenarios - Heston model

6 Conclusion and open problems

The goal of this thesis was to assess the performance of hedging strategies that were determined using neural networks. The theoretical foundations that justify the approach were firstly introduced in [7]. We saw that we can utilize the concept of convex risk measures to define optimal hedging strategies that do not rely on any assumption of the underlying market dynamics. This broad setting also enables us to incorporate market frictions such as transaction costs into our model. Furthermore, the approximation capabilities of neural networks allow finding close-to optimal estimates for these strategies. For our numerical experiments, we chose two different synthetic markets and also included proportional and fixed transaction costs. We got promising results in the experiments that demonstrate how powerful the developed algorithm for hedging and pricing of derivatives is. Not only is the method able to learn a model implied strategy without any information about the underlying dynamics, but also outperforms this model hedge if we are aiming to reduce the risk for particular risk measures as the $CVaR$. We also demonstrated that the algorithm is able to make reasonable hedging decisions in an incomplete market with transaction costs. Here the approach showed significantly better results than the conventional model strategy. Building upon these findings, we also explained how the algorithm could be used to determine valid prices for derivatives in an incomplete market. A problem that is still considered to be very challenging. Finally, we applied the algorithm on a market with two risk factors. Again, the neural network based approach showed its capacity to make valid decisions in an environment where an optimal strategy has to trade in two different assets.

Although our implementation could already demonstrate the astonishing performance of the approach, our experiments were carried out in a constraint setting and the algorithm is still subdued to severe limitations. For example, the implementation is restricted to a particular payoff. A neural network trained on a specific payoff fails in the pricing or hedging of a different payoff. For example, even a slight change of the strike K for a European call option leads to unusable results. It is an interesting question if there are possibilities to generalize our approach in a way that allows using a neural network for a variety of different payoffs. Moreover, although we could prove in Section 4, that a feedforward neural network is theoretically able to approximate an optimal hedging strategy arbitrary well, there are no criteria that can guarantee the ϵ-optimality of an estimated solution. We only could assess the performance of our approach by comparing it to a model implied strategy, which justified the assumption that the algorithm is capable of finding close-to optimal estimates. However, it also turned out that the architecture (number of layers, size of layers) of the neural network has a significant effect on the resulting approximations. In this work, we did not try to optimize our network at all, although this is one of the most discussed problems in the field of machine learning and there are already many promising approaches to improve the efficiency and performance of neural networks.

During the implementation phase for this work, we also tested our method with a different network architecture. We used a so-called *Long short-term memory* (LSTM) network, that was introduced in [13]. This structure is a particular network in the class of the fully recurrent networks (see Section 4), but exhibits a more sophisticated structure than the feedforward networks discussed in this thesis. LSTM networks became very popular in recent years and are one of the main reasons for the tremendous progress achieved in the fields of machine learning and artificial intelligence. LSTM networks are in particular well-suited for the processing of sequential data, such as time series, and thus it appears reasonable to utilize these structures for the estimation of hedging strategies. It turned out that the LSTM network based algorithm was indeed able to find even better estimates for the optimal hedging problems considered in this work than the ones presented in Section 5.4. This finding motivates further research of the different neural networks with the focus on their application in financial mathematics. Finally, future work could aim on applying the presented algorithm on real market data to see if and how it could be used in practice. Here the challenge to overcome is to find data sets with a sufficient size to train the neural networks efficiently. Nevertheless, the promising results we could present in this work, motivate the further study of the approach.

7 References

[1] Acerbi, C., and Tasche, D. On the coherence of Expected Shortfall. *Journal of Banking and Finance 26*, 7 (2002), 1487–1503.

[2] Albrecher, H., Mayer, P. A., Schoutens, W., and Tistaert, J. The little Heston trap. *Wilmott*, 1 (2007), 83–92.

[3] Andersen, L. Efficient simulation of the Heston stochastic volatility model. SSRN, 2007. https://papers.ssrn.com/sol3/papers.cfm?abstract$_i d = 946405$.

[4] Ben-Tal, A., and Teboulle, M. An old-new concept of convex risk measures: the optimized certainty equivalent. *Mathematical Finance 17*, 3 (2007), 449–476.

[5] Bengio, Y., Goodfellow, I., and Courville, A. *Deep Learning.* MIT Press, 2016.

[6] Black, F., and Scholes, M. The pricing of options and corporate liabilities. *The Journal of Political Economy 81*, 3 (1973), 637–654.

[7] Buehler, H., Gonon, L., Teichmann, J., and Wood, B. Deep hedging. arxiv, 2018. https://arxiv.org/pdf/1802.03042.pdf, Accessed 2019/02/25.

[8] Chung, K. L. *A Course in Probability Theory.* Academic Press, 2001.

[9] Demeterfi, K., Derman, E., Kamal, M., and Zou, J. More than you ever wanted to know about volatility swaps. *Goldman Sachs - Quantitative Strategies Research Notes* (1999).

[10] Dupire, B. Pricing with a smile. *Risk 7* (1994), 18–20.

[11] Foellmer, H., and Schied, A. *Convex measures of risk and trading constraints.* Humboldt-Universitaet zu Berlin, Wirtschaftswissenschaftliche Fakultaet, 2001.

[12] Heston, S. L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies 6*, 2 (1993), 327–343.

[13] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780.

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

[15] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approxiators. *Neural Networks 2* (1989), 359–366.

[16] Hull, J. C. *Options and other Derivatives.* Prentice Hall, 2012.

[17] Ioffe, S., and Szegedy, C. Bactch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning* (2015), 448–456.

[18] Kingma, D. P., and Ba, J. Adam: a method for stochastic optimization. *Proceedings of the International Conference on Learning Representations (ICLR)* (2015).

[19] Panas, V. G., Davis, M. H. A., and Zariphopoulou, T. European option pricing with transaction costs. *SIAM Journal on Control and Optimization 31*, 2 (1993), 470–493.

[20] Sonder, H. M., Shreve, S. E., and Cvitanic, J. There is no nontrivial hedging portfolio for option pricing with transaction costs. *The Annals of Applied Probability 5*, 2 (1995), 327–355.

[21] Uryasev, S., and Rockafellar, R. Optimization of Conditional Value-at-Risk. *Journal of risk 2* (2000), 21–42.