DIPLOMARBEIT

Deep Hedging

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Finanz-und Versicherungsmathematik

eingereicht von

Ana Grm B.Sc.

Matrikelnummer: 01528183

ausgeführt am Institut für Stochastik und Wirschaftsmathematik
der Fakultät für Mathematik und Geoinformation der Technischen Universität Wien

Betreuung
Betreuer: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Gerhold

Wien, 15.12.2023

(Unterschrift Verfasserin) (Unterschrift Betreuer)

Abstract

This thesis presents an innovative approach developed by Bühler et al. ([1]), for hedging a portfolio of derivatives in the presence of market frictions such as transaction costs, market impact, liquidity constraints or risk limits. Recognizing the limitations of traditional risk management models like Black-Scholes, which rely on assumptions of complete, frictionless markets, this thesis introduces a hedging methodology based on deep reinforcement learning. Unlike conventional models, this approach does not depend on predefined market dynamics or the computation of Greeks. Instead, it employs neural networks to model hedging strategies, adapting to various market conditions including those with market frictions. This represents a significant shift from traditional derivative trading methods, focusing on practical market dynamics rather than theoretical constructs.

Moreover, the thesis explores the subdiffusive Black-Scholes model, providing a more comprehensive view of market dynamics and the valuation of assets. This model in particularly reflects market behaviors not adequately represented by the traditional Black-Scholes model, such as long-range correlations, heavy-tailed distributions, and periods of constant asset prices – characteristics especially prevalent in emerging markets.

The practical application of these methodologies is demonstrated through Python-based implementations, showcasing the deep hedging method and the subdiffusive Black-Scholes model. The effectiveness and accuracy of these models are highlighted, along with a comparative analysis between the two approaches.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 15.12.2023

Ana Grm

Contents

List of Figures

1 Introduction

1.1 Reinforcement Learning and Risk Management

Risk management has always been an integral part of every financial institution. The whole process of identifying, analyzing the risk in financial assets has nowadays become even more important, because with the rapid globalization, new, more complex risks have emerged. In order to prevent huge losses that are caused through that risks, a company has to adapt different techniques to manage the risk. Traditionally, risks in derivatives trading have been assessed using the computation of Greeks within the framework of classical stochastic theory, relying on a frictionless and complete market model, known as risk-neutral pricing. However, in practise complete markets do not exist. In real world trading in any financial instrument is typically influenced by transaction costs, enduring effects on market prices, and limitations related to the liquidity of the asset. Traditional models like the Black-Scholes model, which assume complete markets, may not be fully applicable or accurate in these scenarios. This has led to the development of more sophisticated models and approaches, such as those incorporating deep reinforcement learning to better manage risks in these more complex and realistic market environments.

This thesis introduces a novel hedging approach, that was established by Buehler [1], which utilizes reinforcement learning where hedging strategies are modelled as neural networks. This model is applicable across various markets including those with complex market frictions, because it does not depend on a chosen market dynamics and does not require the calculation of Greeks. Instead of calculating Greeks, the focus shifts to modeling realistic market dynamics and evaluating the actual performance of the hedging strategy in out-of-sample scenarios. This aspect marks a significant departure from traditional derivative trading approaches, emphasizing practical market dynamics over theoretical models.

Furthermore, this thesis delves into the subdiffusive Black-Scholes model, providing an enhanced view of market dynamics and asset valuation. This is particularly crucial for comprehending subdiffusive market behaviors, which are not effectively represented by the traditional Black-Scholes model. Black Scholes Model is a well known and mostly used model for pricing of European Options. Even though it is a very good model it doesn't take into account market properties like: long range correlation, heavy tailed and skewed marginal distributions, lack of scale invariance and periods of constant values, ([12]). Exactly the last point we would like to observe and take into consideration in our model. Therefore, we present the so called subdiffusive Black Scholes model in which the prices are modelled by subdiffusive geometric Brownian motion. Subdiffusive characteristics describe price staleness effect that is particularly shown in emerging markets where the number of participants and the number of transactions are low and so the price processes

stay constant or have very low fluctuations,([11]).

Lastly, we delve into practical applications by showcasing some examples of the deep hedging method and subdiffusive Black Scholes method, implemented using Python programming. These examples serve to highlight the theoretical aspects of both methods and to also demonstrate the effectiveness and accuracy of the deep hedging model. Furthermore, we perform a comparative analysis between the deep hedging approach and the subdiffusive Black Scholes model.

This thesis is structured as following:

- Chapter 2: in this chapter, we establish the mathematical foundation for pricing and hedging with the deep hedging model. We focus on describing a model in an incomplete market, one that encompasses various market frictions. We optimize hedging of a portfolio of derivatives under convex risk measures.
- Chapter 3: Here, we explore the application of neural networks in the financial hedging. We provide a basic knowledge on the training process of neural network and explain why hedging strategies can be modeled by neural networks.
- Chapter 4: We delve into the mathematical concepts of subdiffusion and apply them to the Black Scholes model in order to better reflect the realities of market behavior.
- Chapter 5: The final chapter provides a practical application of the concepts discussed. We present an example in Python, demonstrating how the deep hedging model can be applied in a real-world setting. Following this, we conduct a comparative analysis, contrasting the results obtained from the deep hedging model with those derived from the subdiffusive Black Scholes model. This comparison is aimed at highlighting the practical differences between these two approaches in modeling financial market dynamics and in developing hedging strategies

.

2 Incomplete Market Model

In this chapter, we lay out a theoretical foundation for both pricing and hedging in the context of discrete-time financial markets that are subject to various frictions. This is achieved through the application of convex risk measures, a mathematical approach that enables assessment of risk under these market conditions. This framework not only provides a structured method for evaluating financial instruments in markets with imperfections but also offers strategies for hedging against potential risks. By employing convex risk measures, we can address the complexities and challenges posed by discrete-time markets, where market dynamics evolve in distinct time intervals and frictions such as transaction costs, liquidity constraints, and market impacts play a significant role.

2.1 Mathematical Finance Basics

First, we will provide some basic definitions from mathematical finance based on [28], that will be needed for better understanding of our optimization problem.

Throughout this section, we consider a discrete time market model in which $d+1$ assets are priced at times $k = 0, ..., T$.

Definition 2.1.1: A filtration $\mathbb{F} = (\mathcal{F}_k)_{k \in T}$ of $\mathcal{F}$ is defined as an increasing family of sub-σ-algebras of $\mathcal{F}$, meaning $\mathcal{F}_s \subset \mathcal{F}_k \subset \mathcal{F}$ for all $s < k$ in T. The triple $\Omega, \mathcal{F}, \mathbb{F}$ is called measurable space. We assume that $\mathcal{F}_0 = \{0, \Omega\}$ and $\mathcal{F}_T = \mathcal{F}$.

Let

$$(\overline{S_k}) = (S_k^0, ..., S_k^d) \quad \text{for} \quad k = 0, ..., n$$

denote mid prices given by an $\mathbb{R}^{d+1}$-valued $\mathbb{F}$-adapted stochastic process where S_k^i represents the price of an i-th asset held in the portfolio at time k. The method used by the deep hedging algorithm presented here does not require an equivalent martingale measure under which S is martingale.

Definition 2.1.2: A stochastic process $Y = (Y_k)_{k=0,...,T}$ is called adapted with respect to the filtration $(\mathcal{F}_t)_{k=0,...,T}$ if each Y_k is $\mathcal{F}_k$-measurable. A stochastic process $Z = (Z_k)_{k=1,...,T}$ is called predictable with respect to $(\mathcal{F}_k)_{k=0,...,T}$ each Z_k is $F_{k-1}-$ measurable.

Definition 2.1.3: Trading strategy is a predictable $\mathbb{R}^d$-valued stochastic process

$$(\overline{\delta_k}) = (\delta_k^0, ..., \delta_k^d) \qquad \text{for} \quad k = 1, ..., T$$

where δ_k^i represents the amount of i-th asset held in the portfolio at time k.

Definition 2.1.4: A strategy $\overline{\delta}$ is self-financing (or replicating) if the relation

$$\overline{\delta}_k \overline{S}_k = \overline{\delta}_{k+1} \ \overline{S}_k$$

holds for every $k = 1, ..., T-1$.
Note that:

$$\overline{\delta}_k \overline{S}_k := \sum_{i=0}^{d} \delta_k^i S_k^i \qquad \text{for} \quad k = 1, \ldots, T-1$$

In other words, the self-financing strategy means, that at time t_k the amount of wealth at our disposal is $\overline{\delta_k S_k}$ and we decide to re-balance our portfolio with the following quantity $\overline{\delta}_{k+1}$ in such way that the overall value of the portfolio does not get changed. The changes in the portfolio value are solely resulting from the accumulated gains and losses arising from fluctuations in the asset price

$$\overline{\delta}_{k+1} \ \overline{S}_{k+1} - \overline{\delta}_k \ \overline{S}_k = \overline{\delta}_{k+1}(\overline{S}_{k+1} - \overline{S}_k) \tag{2.1}$$

In fact, $\overline{\delta}$ is self-financing if and only if 2.1 holds for $k = 1, ..., T-1$. It follows through summation over 2.1 that

$$\overline{\delta}_k \ \overline{S}_k = \overline{\delta}_1 \ \overline{S}_0 + \sum_{j=1}^{k} \overline{\delta}_k (\overline{S}_k - \overline{S}_{k-1}) \quad \text{for} \quad k = 1, ..., T.$$

Here, the constant $\overline{\delta}_1 \ \overline{S}_0$ can be interpreted as the initial investment for the purchase of the portfolio $\overline{\delta}_1$.
Let us now assume that S_0^0 is the numeraire, then the discounted price process is

$$X_k^i := \frac{S_k^i}{S_k^0} \quad k = 0, ..., T, \quad i = 0, ..., d, \quad X_k^0 = 1.$$

Definition 2.1.5: The (discounted) value process $V = (V_k)_{t=0,...,n}$associated with a trading strategy $\overline{\delta}$ is given by

$$V_0 := \overline{\delta}_1 \overline{X}_0 \quad \text{and} \quad V_k := \overline{\delta}_k \overline{X}_k \quad \text{for} \quad k = 1, ..., T.$$

V_k can be interpreted as the portfolio value at the end of the k-th trading period expressed in units of the numéraire asset

$$V_k = \overline{\delta}_k \overline{X}_k = \frac{\overline{\delta}_k \overline{S}_k}{S_k^0}.$$

Definition 2.1.6: A self-financing trading strategy is called an arbitrage opportunity if its value process V satisfies

(1) $V_0 \geq 0$

(2) $V_n \geq 0 \quad \mathbb{P} - a.s.$

(3) $\mathbb{P}[V_n > 0] > 0$

Theorem 2.1.7 (Fundamental Theorem of Asset Pricing): The market model is arbitrage-free if and only if the set $\mathcal{P}$ of all equivalent martingale measures is non-empty. In this case, there exists a $\mathbb{P}^* \in \mathcal{P}$ with bounded density $d\mathbb{P}^*/d\mathbb{P}$.

Definition 2.1.8: A non-negative random variable Z on $(\Omega, \mathcal{F}_T, \mathbb{P})$ is called a European contingent claim. A European contingent claim Z is called a derivative of the underlying assets $S^0, S^1, ..., S^d$ if Z is measurable with respect to the σ-algebra generated by the price process $(\overline{S}_k)_{k=0,...,n}$.

Definition 2.1.9: We assume that the market is arbitrage free. Contingent claim Z is attainable if there exists a self-financing strategy $\overline{\delta}$ such that

$$Z = \overline{\delta_T}\,\overline{S_T} \quad \mathbb{P} - a.s. \tag{2.2}$$

Such a trading strategy $\overline{\delta}$ is called a replicating strategy for Z.
The discounted value of a contingent claim Z when using the numéraire S_0 is given by

$$H := \frac{Z}{S_T^0}.$$

Clearly it follows from 2.2

$$H = \overline{\delta}_T \overline{X}_T = V_T = V_0 + \sum_{k=1}^{T} \delta_k (X_k - X_{k-1})$$

Theorem 2.1.10: Any attainable discounted claim H is integrable with respect to each equivalent martingale measure, i.e.,

$$\mathbb{E}_{\mathbb{P}^*}[H] < \infty \quad \forall \mathbb{P}^* \in \mathcal{P}.$$

Moreover, for each $\mathbb{P}^* \in \mathcal{P}$ the value process of any replicating strategy satisfies

$$V_k = \mathbb{E}_{\mathbb{P}^*}[H|\mathcal{F}_k] \quad \mathbb{P} - \text{a.s.} \quad \text{for} \quad k = 0, ..., T. \tag{2.3}$$

In particular, V is a non-negative $\mathbb{P}^*$-martingale.

Remark 2.1.11: When applied to an attainable contingent claim Z prior to discounting, upper Theorem states that

$$\overline{\delta}_k \overline{S}_k = S_k^0 \mathbb{E}_{\mathbb{P}^*}[\frac{Z}{S_T^0}|\mathcal{F}_k] \quad k = 0, ..., T \quad \mathbb{P} - \text{a.s.} \quad \forall \mathbb{P}^* \in \mathcal{P}$$

for every replicating strategy $\overline{\delta}$. In particular, the initial investment which is needed for a replication of Z is given by

$$\overline{\delta}_1 \overline{S}_0 = S_0^0 \mathbb{E}_{\mathbb{P}^*}[\frac{Z}{S_T^0}].$$

Let us now turn to the problem of pricing a contingent claim. Consider first a discounted claim H which is attainable. Then the (discounted) initial investment

$$\overline{\delta}_1 \overline{X}_0 = V_0 = \mathbb{E}_{\mathbb{P}^*}[H].$$

needed for the replication of H can be interpreted as the unique (discounted) "fair price" of H. In fact, a different price for H would create an arbitrage opportunity.

Definition 2.1.12: A real number π^H is called an arbitrage-free price of a discounted claim H, if there exists an adapted stochastic process X^{d+1} such that

(1) $X_0^{d+1} = \pi^H$

(2) $X_k^{d+1} \geq 0 \quad k = 1, ..., T-1$

(3) $X_T^{d+1} = H.$

and such that the enlarged market model with price process $(X^0, X^1, ..., X^{d+1})$ is arbitrage-free. The set of all arbitrage-free prices of H is denoted by $\Pi(H)$.

Thus, an arbitrage-free price π^H of a discounted claim H is by definition a price at which H can be traded at time 0 without introducing arbitrage opportunities into the market model: If H is sold for π^H , then neither buyer nor seller can find an investment strategy which both eliminates all the risk and yields an opportunity to make a positive profit.

Theorem 2.1.13: Let H be a discounted claim.

(a) If H is attainable, then the set $\Pi(H)$ of arbitrage-free prices for H consists of the single element V_0, where V is the value process of any replicating strategy for H.

(b) If H is not attainable, then $\inf \Pi(H) < \sup \Pi(H)$ and $\Pi(H) = (\inf \Pi(H), \sup \Pi(H))$.

Theorem 2.1.14 (2nd FTAP): An arbitrage-free market model is complete if and only if there exists exactly one equivalent martingale measure.

Lemma 2.1.15: An arbitrage free market is complete if and only if every contingent claim is attainable.

2.2 Incomplete Markets

In this section we will define the setting of the incomplete market that will be needed for our deep hedging strategy. Let us define the model in a discrete time financial market with finite time horizon T with trading times

$$0 = t_0 < t_1 < ... < t_n = T.$$

Let $(\Omega, \mathcal{F})$ denote a measurable space where $\Omega = \{w_1, ..., w_N\}$ is a finite probability space with a probability measure $\mathbb{P}$ such that $\mathbb{P}\big[\{w_i\}\big] > 0$ for all i.
Let $I_k \in \mathbb{R}^r$ represent all important information such as mid prices, risk limits, news and trading signals available at time t_k. The process $I = (I_k)_{k=0,...,n}$ generates a filtration $\mathbb{F} = (\mathcal{F}_k)_{k=0,...,n}$, which means that $\mathcal{F}_k$ models the information available at time t_k, which is essential in mathematical finance, because trading strategies can't use future information, so that situations like insider trading can be avoided.

We define a market with d hedging instruments and let

$$(\overline{S_k}) = (S_k^1, ..., S_k^d) \quad \text{for} \quad k = 0, ..., n$$

denote mid prices given by an $\mathbb{R}^d$-valued $\mathbb{F}$-adapted stochastic process.
S_k^i represents the price of an i-th asset held in the portfolio at time t_k. The method used by the deep hedging algorithm presented here does not require an equivalent martingale measure under which S is martingale.
A contingent claim Z is $\mathcal{F}_T$-measurable random variable and in our case represents a portfolio of liquid and OTC derivatives.

Furthermore, a predictable $\mathbb{R}^d$-valued stochastic process

$$(\overline{\delta_k}) = (\delta_k^1, ..., \delta_k^d) \qquad \text{for} \quad k = 0, ..., n-1$$

represents trading strategy where δ_k^i denotes the amount of i-th asset held in the portfolio at time t_k. We fix $\delta_{-1} = \delta_n =: 0$.
$\mathcal{H}^u$ represents the set of all possible trading strategies without any constraints. However, in real-world scenarios, there are constraints like liquidity, asset availability, or specific trading restrictions. These constraints are represented by $\mathcal{H}_k$, which is a set of allowed strategies at time t_k. The constraints can vary over time, so $\mathcal{H}_k$ is given as a image of a continuous function

$$H_k : \mathbb{R}^{d(k+1)} \to \mathbb{R}^d \tag{2.4}$$

i.e. $\mathcal{H}_k := H_k(\mathbb{R}^{d(k+1)})$. We assume that the trading strategy δ_k is restricted to a set $\mathcal{H}_k$.
Furthermore, for an unconstrained trading strategy $\delta^u \in \mathcal{H}^u$ we inductively define with:

$$(H \circ \delta^u)_k = H_k((H \circ \delta^u)_0, ..., (H \circ \delta^u)_{k-1}, (H \circ \delta^u)_k)$$

its constrained projection into $\mathcal{H}_k$ as
Now, we can define $\mathcal{H} := (H \circ \mathcal{H}^u) \subset \mathcal{H}^u$ which is the set of all constrained trading strategies.
The value of the portfolio at maturity is given by

$$-Z + p_0 + \overline{\delta_T}\,\overline{S_T}$$

where

$$\overline{\delta_T}\,\overline{S_T} = \sum_{k=0}^{n-1} \overline{\delta}_k(\overline{S}_{k+1} - \overline{S}_k)$$

and p_0 is the fair price. Furthermore, we would also like to take into consideration the costs that occur through trading activity. We assume that the costs $c_k(n)$ arise while buying a position $n \in \mathbb{R}^d$ in $\overline{S_k}$ at time t_k. The total cost of trading a strategy $\delta = (\overline{\delta}_k)_{k=0,\dots,n-1}$ up to maturity

$$C_T(\delta) = \sum_{k=0}^{n} c_k(\overline{\delta}_k - \overline{\delta}_{k-1}). \tag{2.5}$$

We set $c_k(0) = 0$ and we assume that c_k is upper-semi continuous. This property is needed for the proof of the proposition 3.3.5. Recall, the upper semi-continuity:

Definition 2.2.1, [30]: Let $\mathbf{X}$ be a topological space, $x \in \mathbf{X}$ and $f : \mathbf{X} \to \mathbb{R}$ be a real valued function. The function f is called upper semi-continuous at a point x_0 if for every real $y > f(x_0)$ there exists a neighborhood U of x_0 such that $f(x) < y$ for all $x \in U$. Equivalently, f is upper semi-continuous at x_0 if and only if

$$\limsup_{x \to x_0} f(x) \geq f(x_0).$$

Definition 2.2.2, [30]: A function $f : \mathbf{X} \to \mathbb{R}$ is called upper semi-continuous if it is upper semi-continuous in every point of its domain.

The value of the portfolio at maturity can now be rewritten to

$$PL_T := -Z + p_0 + \overline{\delta_T}\,\overline{S_T} - C_T(\delta).$$

Lastly, for simplifications we set the risk free rate $r = 0$.

2.2.1 Hedging using convex risk measures

In a complete market with no frictions exists for every contingent claim Z unique replicating strategy δ with fair price p_0 so that

$$-Z + p_0 + \overline{\delta_T}\,\overline{S_T} = 0.$$

But since we are working in a market with frictions, this does not hold and we face the following problems, [2]:
1. The uniqueness of the price of the derivative: the arbitrage price should be independent of the estimate of the probability measure and should only depend on the prices of the underlying assets. In other words, the price described in the equation 2.1 is in the incomplete market not unique anymore but rather depends on the choice of the equivalent martingale measure.
2. Hedging techniques: in an incomplete market not every derivative can be replicated, therefore an appropriate hedging strategy that lowers the risk of the portfolio needs to be considered. In order to do so, we must specify an optimality criterion which will define the minimal price of any financial position and give us the optimal hedge. This minimal

price can also be interpreted as the minimal amount of capital we need to make this position acceptable.

As optimaltity criterion we will take convex risk measure.
Our positions in the portfolio will be modeled as real-valued measurable functions X on a measurable space $(\Omega, \mathcal{F})$ of possible scenarios. $\mathcal{X}$ will denote the linear space of all measurable functions.
Definition 2.2.1.1: A functional $\rho : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ with $\rho(0) \in \mathbb{R}$ is called a risk measure if it is:

1. Monotone decreasing

   If $X_1, X_2 \in \mathcal{X}$ and $X_1 \leq X_2$, then $\rho(X_1) \geq \rho(X_2)$.
   Meaning, that a position which is valued more, needs less capital that will make it acceptable.

2. Translation invariant or cash-invariant

   If $a \in \mathbb{R}$ and $X \in \mathcal{X}$, then $\rho(X + a) = \rho(X) - a$.
   Adding cash to the position reduces the need for more cash by as much. We can interpret $\rho(X)$ as the minimal capital needed to make the position X acceptable.

Definition 2.2.1.2, [31]: A risk measure $\rho : \mathcal{X} \to \mathbb{R}$ is called convex if it fulfils the following property:

$$\rho(\lambda X + (1-\lambda)Y) \leq \lambda\rho(X) + (1-\lambda)\rho(Y) \qquad \forall X, Y \in \mathcal{X}, \lambda \in [0,1].$$

Through this property it is assured that a diversification of a portfolio does not increase risk.

Definition 2.2.1.3: A convex risk measure is called coherent if it is:

1. positively homogeneous:

$$\rho(\lambda X) = \lambda\rho(X) \qquad \forall X \in \mathcal{X}, \lambda \geq 0$$

2. subadditive:

$$\rho(X + Y) \leq \rho(X) + \rho(Y) \qquad \forall X, Y \in \mathcal{X}.$$

Any coherent risk measure is also normalized, meaning $\rho(0) = 0$.

Definition 2.2.1.4: Let C be a convex subset of $\mathbb{R}^n$. The function $f : C \to \mathbb{R}$ is convex if

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y) \quad \forall x, y \in C, \quad \forall \alpha \in [0,1].$$

Now we have all of the infrastructure needed to build a convex risk measure. Therefore we define the following optimization problem:

$$\pi(X) := \inf_{\delta \in \mathcal{H}} \rho(X + \overline{\delta_T}\,\overline{S_T} - C_T(\delta))$$

where $\rho : \mathcal{X} \to \mathbb{R}$ is a convex risk measure, $C_T(\delta)$ are trading costs as defined in 2.5 and $X \in \mathcal{X}$.
The optimal trading strategy has been defined as a minimizer of $\pi(X)$.

Proposition 2.2.1.5: π is monotone decreasing and cash-invariant. If $C_T(\cdot)$ and $\mathcal{H}$ are convex, then π is a convex risk measure.
Proof: see [1].
To remind again, $\pi(-Z)$ is the minimal price, meaning, the minimal amount of capital the trader needs to charge to make the liability $-Z$ acceptable. But this minimal price does not take into consideration the possibility that having no liabilities in the portfolio could actually increase value. For that reason, we specify the indifference price $p(Z)$ which is the amount of capital the trader would charge to be indifferent between the position $-Z$ and not taking any action. Through cash invariance we get

$$\pi(-Z + p_0) = \pi(-Z) - p_0 = \pi(0)$$

we can set $p_0 := p(Z)$ where

$$p(Z) := \pi(-Z) - \pi(0). \tag{2.6}$$

In a market without transaction costs $p(Z)$ is the price of a replicating portfolio.

2.2.2 Utility indifference pricing

In complete markets, where all risks can be hedged, traditional pricing methods like risk-neutral valuation work well. But in incomplete markets, not all risks can be hedged away. This is where utility indifference pricing becomes important.It provides a way to value derivatives and other instruments by considering the investor's attitude towards risk and the impact of adding or removing an instrument from their portfolio. In other words, utility indifference pricing is a method to price financial instruments by determining the price at which an investor is indifferent, in terms of their utility, to holding or not holding the instrument.
We consider the exponential utility function:

$$U(x) = -\exp(-\lambda x), \qquad x \in \mathbb{R}$$

where $\lambda > 0$ is the risk aversion parameter. The indifference price $q(Z) \in \mathbb{R}$ of Z is:

$$\sup_{\delta \in \mathcal{H}} \mathbb{E}[U(q(Z) - Z + \overline{\delta_T}\ \overline{S_T} + C_T(\delta))] = \sup_{\delta \in \mathcal{H}} \mathbb{E}[U(\overline{\delta_T}\ \overline{S_T} + C_T(\delta))]. \tag{2.7}$$

The utility indifference price is the smallest price the investor is willing to accept in order to sell Z.

Lemma 2.2.1: Let ρ define an entropic risk measure:

$$\rho(X) = \frac{1}{\lambda} \log \mathbb{E}[\exp(-\lambda X)]$$

and let $q(Z)$ be defined as in 2.7 and let $p(Z)$ be defined as in 2.6. Then $p(Z) = q(Z)$.
Proof: see [1].
This lemma proves that in the above presented framework, the exponential utility indifference pricing is also included.

3 Reinforcement learning

The primary aim of this chapter, and indeed the overarching goal of our paper, is to present an innovative approach, developed in the paper from Buehler et al.[1], to portfolio hedging, employing deep reinforcement learning to address challenges posed by market frictions like transaction costs, liquidity constraints, and varying risk limits. This method stands out because it utilizes neural networks for making trading decisions, a significant shift from conventional hedging strategies. These neural networks are not limited to using just the price data of hedging instruments. They also integrate additional data types, such as trading signals, relevant news analytics, and historical hedging decisions. This broad data integration mirrors the multifaceted information a human trader might consider, aligning the approach more closely with real-world trading scenarios.

The goal of this chapter and of the whole paper is to approximate an optimal hedging strategy using neural networks. We will describe the deep learning process and provide a mathematical description of it and lastly approximate a hedging strategy using neural networks.

3.1 Introduction to Reinforcement Learning

Machine learning can be divided into three main domains: supervised learning, unsupervisend learning and reinforcement learning. The latter will be the main focus of our paper.

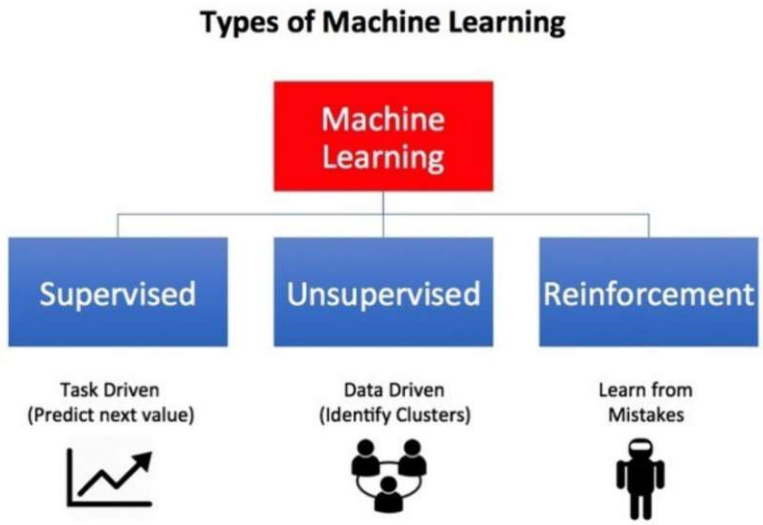

Figure 3.1: Types of Machine Learning([3])

Reinforcement learning is a type of machine learning that enables an agent to learn in a complex environment through trials and errors, meaning in case of doing a desired action, the agent gets positive reinforcement and in case of doing an undesired action, the agent gets negative feedback, [3].

Let $\mathcal{A}$ define a set of actions, $\mathcal{S}$ the set of environments' states and $\mathcal{R}$ the set of rewards. The environment is in this case a physical world that the agent interacts with and S_t represents the situation of the agent at timepoint t. At every timestep $t = 0, 1, 2, 3, \ldots$ the agent decides to take an action $A_t \in \mathcal{A}$ based on the state $S_t \in \mathcal{S}$ he is currently in, [4]. As a consequence of its action, the agent receives a reward $R_{t+1} \in \mathcal{R}$ and moves to state S_{t+1}.

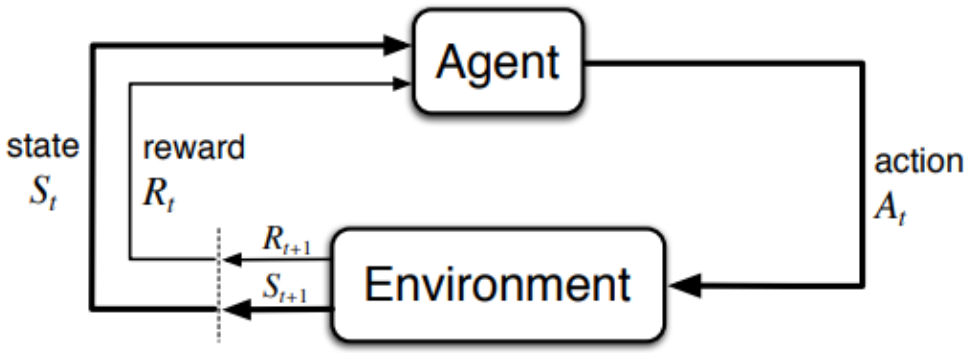

Figure 3.2: Representation of the learning process ([4])

Definition 3.1.1: A state-transition probability function $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ is specified as, [4]:

$$p(s'|s, a) = \mathbb{P}[S_t = s'|S_{t-1} = s, A_{t-1} = a]. \tag{3.1}$$

p therefore describes the probability of transitioning to state s' from state s at time t under action a.

Definition 3.1.2: A reward-probability function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as,[4]:

$$r(s, a, s') = \mathbb{E}[R_{t-1}|S_{t-1} = s, A_{t-1} = a, S_t = s]. \tag{3.2}$$

r is the immediate reward after transitioning from state s to state s'.
The goal of reinforcement learning is for the agent to learn a policy π,

$$\pi : \mathcal{A} \times \mathcal{S} \to [0, 1] \tag{3.3}$$

$$\pi(s, a) = \mathbb{P}[A_t = a|S_t = s] \tag{3.4}$$

that maximizes his expected cumulative rewards, [5].

In case of the high dimensional states, the deep learning comes into play which transforms the set of inputs to the set of outputs using deep neural networks. With deep reinforcement learning the agent makes decisions based on the unstructured data without manual engineering the space of states $\mathcal{S}$, [6]. To recall, our goal is to find an optimal hedging strategy with deep neural networks and in that case we can interpret market information as states, trading strategies as actions and portfolio value as a reward.

3.2 Deep Neural Networks

In this section we will first go through some basics of neural networks and then apply this architecture to our hedging problem. We will illustrate how neural networks can be utilized to model and predict market dynamics, enabling the formulation of effective hedging strategies that can adapt to complex, real-world financial scenarios. A mathematical description of a neural network involves several key components: the architecture of the network (layers and neurons), the activation functions, the process of forward propagation (how data moves through the network), and backpropagation (how the network learns).

Neural Network Architecture: A neural network consists of an input layer which is the first layer of the network, where each node represents one feature of the input data, one or more hidden layers that process the inputs from the previous layer, and an output layer which is the final layer that produces the output of the network. Each layer is made up of units or neurons. Neurons in one layer are connected to neurons in the next layer through weights. Figure 3.5 depicts a neural network with one hidden layer. A feed forward neural network is a neural network where input data are fed into each layer into forward direction, meaning in each layer the values are calculated and passed on forward to the next layer.

Simply speaking, the goal of a neural network is to approximate some function f with the input $\mathbf{x}$ and output $\mathbf{y}$. A feedforward network defines a mapping $\mathbf{y} = f(\mathbf{x}; \mathbf{w})$ and learns the value of the parameters w that result in the best function approximation. To train the neural network would then mean finding the weights $\mathbf{w}$, such that $f(\mathbf{x}, \mathbf{w}) = y$. For that lets first recall that for very simple linear regression the following equation stands $y = f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}$. In the following steps we will present how the model is trained to find the optimal weights.

Data enters the network at the input layer, then travels through each hidden layer, and finally exits via the output layer. At each layer, the output of a neuron is calculated as a weighted sum of its inputs, followed by an application of an activation function. This process is called forward propagation. The output of a j-th neuron in the i-th layer is calculated as a weighted sum of its inputs, plus a bias term b_j

$$z_{ij} = \sum_{i=1}^{n} w_{ij}x_i + b_j \quad \text{for} \quad j = 1, ..., \text{number of neurons}$$

where x_i are the inputs, w_{ij} are the weights and b_j are biases. The bias allows the neuron to shift the activation function to the left or right, which can be critical for successful

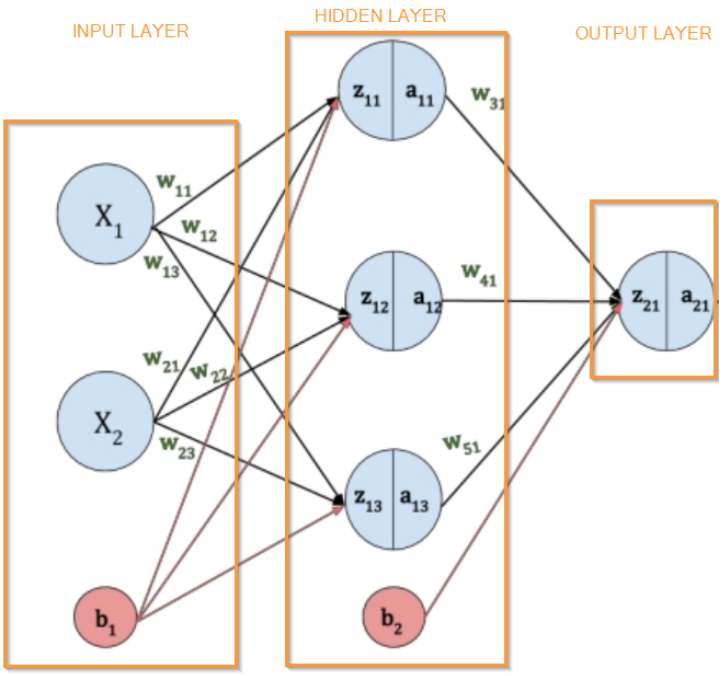

Figure 3.3: Neural network with one hidden layer, ([7])

learning. Linear models possess a significant flaw which is that they are limited to linear functions, which restricts their ability to capture interactions between any two input variables. Consequently, these models are unable to comprehend or represent the complexities that arise from the interactions of different input factors. To enable linear models to represent nonlinear functions of the input x, we can adapt the approach by applying the linear model to a transformed version of the input, denoted as $\sigma(x)$, where σ is a nonlinear function, for example sigmoid, RELU or leaky RELU, [26]. This alteration allows the model to handle more complex relationships by first transforming the input through a nonlinear mapping before applying the linear model. Finally, the result of the weighted sum and bias is passed through an activation function. This function introduces non-linearity, allowing the neuron to learn more complex patterns. The final output of the j-th neuron in the i-th layer is

$$a_{ij} = \sigma(z_{ij})$$

where σ is the activation function and z_{ij} is the output neuron. This result is passed on to the next layer. The final layer produces the output of the network, which is the network's prediction based on the input data.

For example, let us take a look into the calculation of weights in the neural network presented in the figure 3.5. Let $\mathbf{x} \in \mathbb{R}^2$ define a vector of our input values. Let $\mathbf{W} \in \mathbb{R}^3 \times \mathbb{R}^2$ represent weights which gives us the information on how much does the given input of a neuron affects its output and $b_1, b_2 \in \mathbb{R}$ biases. Let z_{ij} denote the value of j-th

neuron in the i-th layer. The calculation is therefore, the following:

$$\begin{bmatrix} z_{11} \\ z_{12} \\ z_{13} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b1 \\ b1 \end{bmatrix}$$

In the next step we input $\mathbf{z}$ into activation function σ that can be defined arbitrary by the user. Let a_{ij} denote the value of the activation function in the i-th layer of the j-th neuron:

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} = \begin{bmatrix} \sigma(z_{11}) \\ \sigma(z_{12)} \\ \sigma(z_{13}) \end{bmatrix}$$

The output value a_{21} is then calculated through:

$$z_{21} = \begin{bmatrix} w_{21} & w_{22} & w_{23} \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} + b_2$$

Then the activation function is applied $a_{21} = \sigma(z_{21})$. In that way the network calculates the output value. In the next step loss function (like mean squared error for regression or cross-entropy for classification) can be calculated to evaluate how far the network's predictions are from the actual target values.

The training aims to minimize a loss function that measures the difference between the network output and the actual target values. Our goal is, that the difference between the target value and output value predicted by the neural network is minimal. In order to reduce this difference an optimization algorithm like gradient descent or ADAM in connection with backpropagation can be used. Backpropagation is a systematic method that involves calculating and distributing errors back through the network, enabling the adjustment of weights and biases to minimize the loss. In this case, the gradient descent of the loss function is calculated for each layer, starting from the output layer and moving backward through the hidden layers to the input layer. At each layer, the gradient tells how much the weights and biases need to be adjusted to reduce the error. This is done by calculating partial derivation of the loss function with respect to each parameter.

After computing the gradients, the weights and biases of the network are updated, typically using an optimization algorithm like Gradient Descent or ADAM. Gradient Descent optimization is done with a learning rate η that controls how much does the weights and biases change, [8]:

$$w_{ij} = w_{ij} - \eta \frac{\partial C}{\partial w_{ij}}$$

$$b_i = b_i - \eta \frac{\partial C}{\partial b_i}.$$

This weight adjustment is needed because we want to get a predicted value that would as little as possible differentiate from our target. Lastly, after the update of weights, the output is calculated again with forward propagation. This whole process is repeated over

many iterations (epochs), each time using either the entire dataset or batches of data (stochastic or mini-batch gradient descent).

The learning process continues until the network's performance converges, i.e., the loss stops decreasing significantly, or a predetermined number of epochs is reached. Thus, the learning rate η has to be chosen, so that the loss converges to the global minimum. Starting with a too low learning rate would result in excessively slow convergence, whereas initiating with an excessively high rate could lead to overshooting the minima, as seen in the figure 3.5.

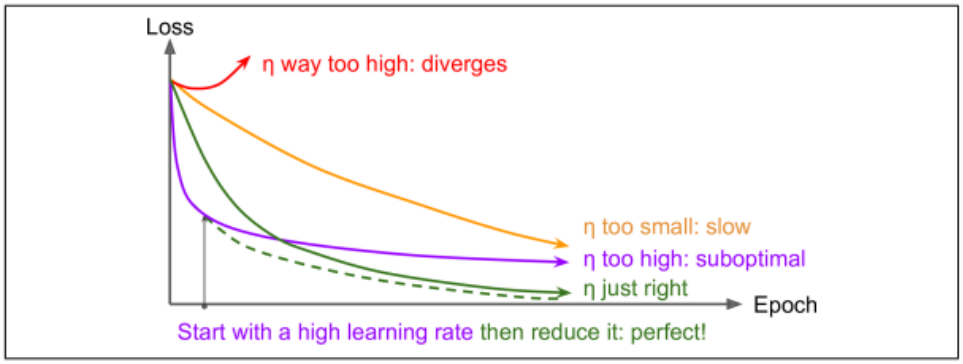

Figure 3.4: Learning curves of different learning rates η, ([27])

Gradient descent is working well for loss functions that are convex, but in other cases the global minimum may never be reached, [27]. An improvement of the Gradient Descent is ADAM-Adaptive Moment Estimation. This algorithm is updating the learning rate for each parameter based on the calculation of exponential moving average of first and second gradient descent. This approach makes Adam very efficient, especially for large datasets and parameters with complex relationships. That was also one of the reasons why we used ADAM in our deep hedging model presented in the chapter 5.

Besides the optimization method and learning rate, the choice of activation function can also impact the network's ability to learn complex patterns and converge during training. Certain activation functions, like the sigmoid or tanh, are susceptible to the vanishing gradient problem, where gradients become very small and the network stops learning effectively. Newer functions like ReLU or leaky ReLU are less prone to this issue, making the network's training more efficient, [27]. In our deep hedging model presented in the chapter 5 we have used the leaky ReLU activation function:

$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ x\alpha & \text{for } x < 0 \end{cases}$$

with $\alpha = 0.01$.

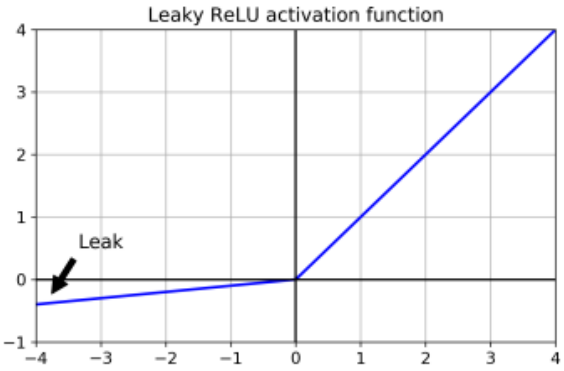

Figure 3.5: Leaky ReLU activation function ([27])

Finally, with a solid understanding of neural networks and deep learning established, we can proceed now to a mathematical definition of the neural network.

Definition 3.2.1 (Feed forward neural network, [1]): Let $L \in \mathbb{N}_{\geq 2}$ denote the number of layers, $d_0 \in \mathbb{N}$ define dimension of the input layer, $d_L \in \mathbb{N}$ denote dimension of the output layer and $d_1, \ldots, d_{L-1} \in \mathbb{N}$ define dimension of the hidden layers. Let $\sigma : \mathbb{R} \to \mathbb{R}$ define the activation function. Lastly, let $W_i : \mathbb{R}^{N_{i-1}} \to \mathbb{R}^{N_i}$ for any $i = 1, \ldots, L$ denote an affine function. A feed forward neural network is a function $F : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ defined as

$$F(x) = W_L \circ F_{L-1} \circ \cdots \circ F_1$$
$$\text{with } F_i = \sigma \circ W_i \qquad \text{for } i = 1, \ldots, L-1.$$

For any $i = 1, \ldots, L$ the affine function W is structured as

$$W_i(x) = A^i x + b^i$$

for some $A^i \in \mathbb{R}^{d_i \times d_{i-1}}$ and $b^i \in \mathbb{R}^{d_i}$. For any $j = 1, \ldots, d_i$ and $k = 1, \ldots, d_{i-1}$ the number A^i_{jk} can be interpreted as the weight connecting the node j of the $i-1$ layer to the node k of the i-th layer.

We have now defined the neural network architecture, so the next question that poses itself is why did we chose neural network to approximate hedging strategies and not some other parametric family of functions? The Universal Approximation Theorem is a fundamental theorem in the field of neural networks which states that a feedforward neural network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of $C(\mathbb{R}^{d_0})$ to any desired degree of accuracy, provided the activation function of the neurons is a non-constant, bounded, and continuous function. In our setting, the prices of hedging instruments show some dependencies, meaning we have a lower dimensional training data set which, proved by Bölcskei ([10]), can also be efficiently approximated by neural network.

3.3 Deep Hedging Strategy

Our goal is to find an optimal hedging strategy in an infinite dimensional space. First, we define $\mathcal{NN}_{\infty,d_0,d_1}$ as the set of neural networks $F : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$.

Definition 3.3.1, [9]: For $1 \leq p < \infty$ we write

$$\| f \|_{p,\mu} = \left[\int_{\mathbb{R}^k} |f(x)|^p d\mu(x) \right]^{1/p}$$

so that $\rho_{p,\mu}(f,g) = \| f - g \|_{p,\mu}$.$L^p(\mu)$ is the space of all functions f such that $\| f \|_{p,\mu} < \infty$. A subset S of $L^p(\mu)$ is dense in $L^p(\mu)$ if for arbitrary $f \in L^p(\mu)$ and $\epsilon > 0$ there is a function $g \in S$ such that $\rho_{p,\mu}(f,g) < \epsilon$.

Definition 3.3.2,[9]: $C(X)$ is the space of all continuous functions on a compact set X in $\mathbb{R}^k$. A subset S of $C(X)$ is dense in $C(X)$ if for arbitrary $f \in C(X)$ and $\epsilon > 0$ there is a function $g \in S$ such that $\sup_{x \in X} |f(x) - g(x)| < \epsilon$.

Theorem 3.3.3, [1]: Suppose σ is bounded and non-constant. The following statements hold:

1. For any finite measure μ on $(\mathbb{R}^{d_0}, \mathcal{B}(\mathbb{R}^{d_0}))$ and $1 < p < \infty$, the set $\mathcal{NN}_{\infty,d_0,1}$ is dense in $L^p(\mathbb{R}^{d_0}, \mu)$.

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

Note that $L^p(\mathbb{R}^{d_0}, \mu)$ is the space of all functions on $\mathbb{R}^{d_0}$ such that $\int_{\mathbb{R}^{d_0}} |f(x)|^p d\mu(x) < \epsilon$. In simpler terms, this theorem asserts that such neural networks are capable of approximating any continuous function, no matter how complex it is, given enough neurons in the hidden layer and a suitable activation function.

Using this theorem we now show that the set of unconstrained trading strategies can approximate well the constraint set of trading strategies.

Let $\{\mathcal{NN}_{M,d_0,d_1}\}$ with $M \in \mathbb{N}$ denote a sequence of subsets of $\mathcal{NN}_{\infty,d_0,d_1}$ and let our activation function be bounded and non constant. $\{\mathcal{NN}_{M,d_0,d_1}\}$ has the following properties:

- $\mathcal{NN}_{M,d_0,d_1} \subset \mathcal{NN}_{M+1,d_0,d_1}$ for all $M \in \mathbb{N}$
- $\bigcup_{M \in \mathbb{N}} \mathcal{NN}_{M,d_0,d_1} = \mathcal{NN}_{\infty,d_0,d_1}$
- for any $M \in \mathbb{N}$ one has $\mathcal{NN}_{M,d_0,d_1} = \{F^\theta : \theta \in \Theta_{M,d_0,d_1}\}$ with $\Theta_{M,d_0,d_1} \subset \mathbb{R}^q$ for some $q \in \mathbb{N}$. The function F^θ is a neural network function parametrized by θ where θ denotes the weights and biases of the layers in the network.

Lemma 3.3.4: The constrained problem

$$\pi(X) = \inf_{\delta \in \mathcal{H}} \rho(X + \overline{\delta_T}\ \overline{S_T} - C_T) \tag{3.5}$$

can be modified to the unconstrained problem:

$$\pi(X) = \inf_{\delta \in \mathcal{H}^u} \rho(X + (H \circ \delta \cdot S)_T - C_T(H \circ \delta))$$

where H is defined as in 2.4. Recall that the information accessible in our market at time t_k is represented by the observed maximal feature set $I_0, ..., I_k$ where I_k has values in $\mathbb{R}^k$. Consequently, our trading strategies should be based on this information, as well as our prior position in our tradable assets. So the set of unconstrained trading strategies can be defined as

$$\begin{aligned} \mathcal{H}_M &= \{(\delta_k)_{k=0,\dots,n-1} \in \mathcal{H}^u : \delta_k = F_k(I_0, \dots, I_k, \delta_{k-1}), F_k \in \mathcal{NN}_{M,r(k+1)+d,d}\} \\ &= \{(\delta_k^\theta)_{k=0,\dots,n-1} \in \mathcal{H}^u : \delta_k^\theta = F^{\theta_k}(I_0, \dots, I_k, \delta_{k-1}), \theta_k \in \Theta_{M,r(k+1)+d,d}\} \end{aligned} \tag{3.6}$$

We replace the set of unconstrained trading strategies $\mathcal{H}^u$ in 3.3 by $\mathcal{H}_M \subset \mathcal{H}^u$. Then the optimization problem is

$$\begin{aligned} \pi_M(X) &= \inf_{\theta \in \mathcal{H}_M} \rho(X + (H \circ \delta \cdot S)_T - C_T(H \circ \delta)) \\ &= \inf_{\theta \in \Theta_M} \rho(X + (H \circ \delta^\theta \cdot S)_T - C_T(H \circ \delta^\theta)) \end{aligned} \tag{3.7}$$

where $\Theta_M = \prod_{k=0}^{n-1} \Theta_{M,r(k+1)+d,d)}$.

Therefore, the challenge of identifying an optimal hedging strategy, originally an infinite-dimensional problem, is transformed into the more manageable task of determining the optimal parameters for our neural network, which is a finite-dimensional constraint problem. Now we can show, that the neural network price $\pi^M(-Z) - \pi^M(0)$ converges towards exact price $p(Z)$.

Proposition 3.3.5: Define $\mathcal{H}_M$ as in 3.6 and π^M as in 3.7. Then for any $X \in \mathcal{X}$,

$$\lim_{M \to \infty} \pi^M(X) = \pi(X) \tag{3.8}$$

Proof: see [1].

Having established a theoretical foundation for employing hedging strategies created from neural networks through Proposition 4.9, let us take a look into an example on how to find the optimal parameter numerically. The goal is to find an optimal or near-optimal parameter $\theta \in \Theta_M$. As an example, let us take a entropic risk measure. If we insert it into 3.7 we get:

$$\pi^M(-Z) = \frac{1}{\lambda} \log \inf_{\theta \in \Theta_M} J(\theta)$$

where

$$J(\theta) := \mathbb{E}[\exp(-\lambda(-Z + (\delta^\theta \cdot S)_T - C_T(\delta^\theta))].$$

A close to optimal $\theta \in \Theta_M$ can be found numerically through gradient descent algorithm and backpropagation.

4 Subdiffusive Model

The classic Black-Scholes model assumes that asset prices follow a geometric Brownian motion, meaning they have a constant volatility and drift rate. This model has been foundational in the development of modern financial theory, particularly for pricing European-style options.

However, the classic Black-Scholes model operates under certain assumptions that may not always hold true in real financial markets. One of these assumptions is the nature of the asset price movement, which in the standard model is typically modeled as a diffusion process with continuous paths. The subdiffusive Black-Scholes model modifies this assumption. It incorporates the concept of subdiffusion, a type of stochastic process where the rate of diffusion is slower than in a standard Brownian motion. This can be more representative of certain market conditions where volatility is not constant and can vary over time, often observed in markets exhibiting periods of low liquidity or high uncertainty. This can lead to more accurate pricing of financial derivatives in markets that are not perfectly efficient or when the underlying assets exhibit non-standard movements.

In this chapter, first some basic mathematical concepts regarding subdiffusion will be presented in order to get the foundational understanding necessary to derive the formula for subdiffusive Black Scholes.

4.1 Mathematical Preliminaries

For the understanding of subdiffusion some knowledge in stochastic calculus is needed. Therefore, we will present some basic concepts in stochastic theory that will be used later on in the subdiffusive Black Scholes model.

4.1.1 Stochastic Fundamentals

Definition 4.1.1.1,[19]: Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $\mathcal{F}_t$. $W_t = W_t(\omega)$ is a one-dimensional Brownian motion with respect to $\mathcal{F}_t$ and the probability measure $\mathbb{P}$, started at 0, if
(1) W_t is $\mathcal{F}_t$ measurable for each $t \geq 0$.
(2) $W_0 = 0$, a.s.
(3) $W_t - W_s$ is a normal random variable with mean 0 and variance $t - s$ whenever $s < t$.
(4) $W_t - W_s$ is independent of $\mathcal{F}_s$ whenever $s < t$.
(5) W_t has continuous paths

Definition 4.1.1.2:A stochastic process $X = (X_t)_{t\in T}$ is called cadlag if almost all paths of X are right continuous for which the left-side limit exists.

Definition 4.1.1.3: A cadlag, adapted, real valued stochastic process $(U_t)_{t\geq 0}$ on a filtered probability space $(\Omega, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$ with $U_0 = 0$ a.s. is said to be a Lévy process if it satisfies the following conditions, [14]:
(1) U has independent increments, i.e. $U_t - U_s$ is independent of $\mathcal{F}_s$ for any $0 \leq s \leq t$.
(2) U has stationary increments, i.e. for any $0 \leq s, t$ the distribution of $U_{t+s} - U_t$ does not depend on t.
(3) U is stochastically continuous, i.e. for every $0 \leq t$ and $\epsilon \geq 0$:

$$\lim_{s\to t} \mathbb{P}(|U_t - U_s| > \epsilon) = 0.$$

Examples of Levy process are Brownian motion and compound Poisson process.

Definition 4.1.1.4: A Levy process $\{U(t)\}_{t\geq 0}$ with nonnegative increments is called a subordinator if and only if its cumulant function can be written as, ([24]):

$$\psi(u) = \lambda u + \int_0^\infty (e^{ux} - 1)\nu(dx)$$

where $\lambda \geq 0$ is the drift and $\nu(dx)$ is the Levy measure that satisfies

$$\nu(-\infty, 0) = 0, \qquad \int_0^\infty (x \wedge 1)\nu(dx) < \infty.$$

Definition 4.1.1.5: Given a subordinator $\{U(t)\}$, the first-passage time process defined as

$$S(t) = \inf\{\tau > 0 : U(\tau) > t\}$$

is called the inverse subordinator.

Definition: A stable process is a real-valued Levy process $\{U(t)\}_{t\geq 0}$ with initial value $U(0) = 0$ that satisfies the self-similarity property, [25]

$$\frac{U(t)}{t^{1/\alpha}} \stackrel{d}{=} U(1) \quad \forall t > 0.$$

The parameter α is called the exponent of the process.

Continuous-Time Random Walk
Unlike in the random walk model where the time interval between steps remain constant, in the Continuous-Time Random Walk (CTRW) model the time interval between steps and the steps size are both random variables. In this model, the walker starts at the origin (point zero) at the initial time T_0, which is set to 0. The walker remains stationary until time T_1, at which point a jump occurs. The size of this jump, R_1, can be either positive or negative. After this jump, the walker waits again until time T_2 to make the

next jump of size R_2, and this pattern continues. Thus, the time intervals $\tau_i = T_i - T_{i-1}$ $i = 1, 2, ...$ which are called waiting times and the sizes of these jumps R_i are both treated as independent and identically distributed variables, [23]. The time of the nth jump in this sequence is denoted as T_n, [23]:

$$T_n = T_0 + \sum_{i=1}^{n} \tau_i \quad \text{for} \quad n = 1, 2, ... \quad \text{and} \quad T_0 = 0$$

The position of the walker at time t is given by

$$W(t) = \sum_{i=1}^{N(t)} R_i$$

where $N(t)$ is the counting process $N_t = \max\{n \in \mathbb{N} : \sum_{i=1}^{n} T_i \leq t\}$.

4.1.2 Stochastic Calculus

A diffusion process is a fundamental concept in physics and mathematics, referring to the way particles spread out over time. It describes the random movement of particles from regions of higher concentration to regions of lower concentration, eventually leading to an equilibrium distribution, [17]. Brownian motion is a key example of a diffusion process. In mathematical finance, diffusion refers to a type of stochastic process that models the evolution of financial variables, such as asset prices, interest rates, or indices, over time under the influence of both deterministic and random factors. Financial diffusion processes are often modeled using stochastic differential equations (SDEs). To analyze and solve SDEs in finance, Itô calculus is used.

Definition 4.1.2.1 (Ito Diffusion): Let $(\Omega, \mathbb{F}, \mathbb{P})$ denote a probability space with a m-dimensional Brownian motion $W_t = (W_t^1, ..., W_t^m)^T$ with respect to the filtration $\mathcal{F}$, which is right-continuous and complete. To go further in detail we need the stochastic differential equation, defined as

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t \quad t \in [0, T]$$

with Borel-measurable functions $b(\cdot, \cdot) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$ and $\sigma(\cdot, \cdot) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^{n \times m}$ for $t \in [0, T]$. Furthermore, the drift coefficient and the diffusion coefficient satisfy a uniform Lipschitz condition: $\exists K > 0 \;\; \forall X, Y \in \mathbb{R}^n$

$$\| b(t, X) - b(t, Y) \| + \| \sigma(t, X) - \sigma(t, Y) \| \leq K \| X - Y \|$$

and the linear growth condition: $\exists K > 0 \;\; \forall X \in \mathbb{R}^n$:

$$\| b(t, X) \| + \| \sigma(t, X) \| \leq K(1 + \| X \|).$$

These conditions imply the existence of a unique strong solution X_t which is called Ito diffusion. The corresponding matrix defined as

$$a(t, x) = \sigma(t, x)\sigma(t, x)^T$$

is called diffusion matrix of X_t.

Definition 4.1.2.2 (Time-homogeneous Ito Diffusion): A time-homogeneous Ito diffusion is a stochastic process $X_t(\omega) = X(t,\omega) : \mathbb{R}^+ \times \Omega \to \mathbb{R}^n$, which is a solution of the stochastic differential equation

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t \quad t \in [0,T],$$

where the coefficients $b(x), \sigma(x)$ fulfill the Lipschitz condition and the linear growth condition, in which case they are simplified to

$$\| b(X) - b(Y) \| + \| \sigma(X) - \sigma(Y) \| \leq K \| X - Y \| \quad X, Y \in \mathbb{R}^n.$$

Definition 4.1.2.3 (Markov Property for time-homogeneous Ito Diffusion): Let X_t be a time-homogeneous Ito diffusion, $f : \mathbb{R}^n \to \mathbb{R}$ be a bounded, Borel-measurable function, $\tau < \infty$ a stopping time and $t, s \geq 0$. Then it holds that

1. Markov property: $\mathbb{E}[f(X_{\tau+s})|\mathcal{F}_t] = \mathbb{E}_{X_t}[f(X_s)]$
2. Strong Markov property: $\mathbb{E}[f(X_{\tau+s})|\mathcal{F}_\tau] = \mathbb{E}_{X_\tau}[f(X_s)]//$

This means that the behavior of the process in the future depends only on the present and not on the past.

4.1.3 Fractional Calculus

Fractional derivatives

We consider $f(x) = x^k, k \in \mathbb{N}$, then the nth derivative of $f(x)$ is:

$$\begin{aligned}\frac{d^n f}{dx^n} = D^n x^k &= k(k-1)(k-2)...(k-n-1)x^{k-n} \\ &= \frac{k!}{(k-n)!}x^{k-n} \\ &= \frac{\Gamma(k+1)}{\Gamma(k-n+1)}x^{k-n}\end{aligned}$$

where $\Gamma(\cdot)$ denotes the gamma function. We can replace n with some arbitrary number α and get:

$$\frac{d^\alpha f}{dx^\alpha} = D^\alpha x^k = \frac{\Gamma(k+1)}{\Gamma(k-\alpha+1)}x^{k-\alpha} \quad \alpha > 0$$

Riemann-Liouville Fractional Derivative is defined as, [23]:

$${}_bD_x^{-\alpha} f(x) = {}_bI_x^\alpha f(x) = \frac{1}{\Gamma(\alpha)}\int_b^x f(u)(x-u)^{\alpha-1}du$$

and for $0 < \alpha < 1$ we get:

$$\begin{aligned}{}_bD_x^{-\alpha} f(x) &= \frac{d}{dt}{}_bI_x^{1-\alpha} f(x) \\ &= \frac{1}{\Gamma(1-\alpha)}\int_b^x f(u)(x-u)^{-\alpha}du\end{aligned} \tag{4.1}$$

Fractional Differential Equations of the Riemann-Liouville type is defined as, [23]:

$$D^{\alpha} f(x) = u(x, f(x)) \quad \alpha > 0$$

with initial conditions:

$$D^{\alpha-k} f(0) = b_k \quad k = 1, 2, ..., n-1$$
$$I^{n-\alpha} f(0) = b_n$$

4.2 Fractional Diffusion

A distinctive feature of Brownian motion is that its path is continuous but lacks smoothness; as a result, it is not differentiable at any point, [23]. This continuous nature implies that a particle undergoing Brownian motion is unable to make instantaneous jumps from one position to another. Anomalous or fractional diffusion refers to a type of diffusion process that deviates from the classical Brownian motion observed in normal diffusion. In normal diffusion, the variance is linearly proportional to time, meaning the variance of the process after time t is proportional to t, reflecting the fundamental property that the spread of the particle's path grows linearly with time. This linear growth in variance is a key feature distinguishing Brownian motion from other types of stochastic processes where variance might evolve differently over time. In contrast, in anomalous diffusion, the variance follows a non-linear time dependency, that is described by a power law

$$\mathbb{E}[X(t)^2] \sim ct^{\alpha} \quad \text{as} \quad t \to \infty \tag{4.2}$$

where K is the so-called generalized diffusion coefficient, [20].
Anomalous diffusions are classified into three main groups:, [20]:

- $0 < \alpha < 1$: subdiffusion
- $\alpha = 1$: Brownian motion
- $\alpha > 1$: superdiffusion

In subdiffusion, particles experience significantly longer travel times compared to classical diffusion, primarily because they frequently pause between steps, leading to a slower diffusion rate. On the other hand, superdiffusion is characterized by particles spreading more rapidly than in regular diffusion. This phenomenon is often observed in biological systems, whereas subdiffusion is typically found in transport processes, [23]. The focus of this paper centers on the analysis of subdiffusion.

The universal formula for subdiffusion is the so called fractional Fokker Planck equation that was derived from the continuous time random walk (CTRW) model. In mathematics, a continuous-time random walk (CTRW) is a generalization of a random walk where the wandering particle waits for a random time between jumps, see the definition in 4.1.1. It is a stochastic jump process with arbitrary distributions of jump lengths and waiting times. The subdiffusive nature of this process is directly linked to the heavy-tailed distributions of the waiting times in the CTRW model. This heavy-tailed distribution means there's a

higher probability of encountering long waiting times compared to what is expected in a normal distribution. Each time the particle jumps, it becomes temporarily immobilized, waiting a comparatively long time before its next movement.

These prolonged waiting periods are what slow down the diffusion, leading to the incorporation of the fractional derivative in equation 4.3. In physics this particle transportation occur under the influence of an external force field (drift in finance) $F(x) \in \mathcal{C}^1([0,\infty))$. To simulate subdiffusion when an external force is present, a generalized version of the fractional Fokker–Planck equation is used, [13]:

$$\frac{\partial w(x,t)}{\partial t} = {}_0D_t^{1-\alpha}\Big[\frac{-\partial}{\partial x}F(x) + K\frac{\partial^2}{\partial x^2}\Big]w(x,t) \tag{4.3}$$

with the initial condition $w(x,0) = \delta(x)$.

In principle, one can investigate subdiffusion processes by using two methods. The first deterministic one is the fractional Fokker-Planck equation (FFPE) that we have presented above, and the second one is based on the subordinated Langevin equation. In this paper we focus on the latter approach, a stochastic subordination method, since it permits to apply Monte Carlo methods, [11].

4.2.1 Stochastic Representation of Subdiffusion

The subdiffusive effects are modeled through the so called inverse α-stable subordinator (see also 4.1.1, 4.1.1,4.1.1) which is defined as:

$$S_\alpha(t) = \inf\{\tau > 0 : U_\alpha(\tau) > t\}$$

where $\{U_\alpha(\tau)\}_{\tau\geq 0}$ is the α-stable subordinator with Laplace transform:

$$\mathbb{E}[e^{-uU_\alpha(\tau)}] = e^{-\tau u^\alpha}, 0 < \alpha < 1.$$

Using $1/\alpha$ self- similarity of $\{U_\alpha(\tau)\}_{\tau\geq 0}$ we obtain, [13]:

$$\begin{aligned}\mathbb{P}[S_\alpha(t) \leq \tau] &= \mathbb{P}[U_\alpha(\tau) \geq t] \\ &= \mathbb{P}[U_\alpha(1)\tau^{1/\alpha} \geq t] \\ &= \mathbb{P}\Big[\Big(\frac{t}{U_\alpha(1)}\Big)^\alpha \leq \tau\Big].\end{aligned} \tag{4.4}$$

Therefore, the distribution of $S_\alpha(t)$ is equal to the distribution of the random variable $\left(\frac{t}{U_\alpha(1)}\right)^\alpha$ which is an important characteristic that we will use later on. Since $U_\alpha(t)$ is a pure-jump process with cadlag (right continuous with left limits) trajectories, the sample paths of $U_\alpha(t)$ are continuous but singular in relation to the Lebesgue measure. Moreover, each jump in $U_\alpha(t)$ aligns with the stationary period of its inverse. Importantly, these extended stationary periods in $S_\alpha(t)$, marked by heavy tails, are suggesting a subdiffusive dynamics. They symbolize prolonged waiting periods during which the particle remains stationary.

Furthermore, a subdiffusion process $Z_\alpha(t)$ can be generated from standard diffusion process by subordination, [11]

$$Z_\alpha(t) = Z(S_\alpha(t)), \quad t \geq 0$$

where $S_\alpha(t)$ is independent from the diffusion process $Z(\tau)$. More precisely, $Z(\tau)$ denotes here the diffusion i.e., corresponding Itô's process given by some stochastic differential equation w.r.t. the Brownian motion $B(t)$.

Since the inverse subordinator $S_\alpha(t)$ is non-Markovian (see Defintion 4.1.2.3), this property is inherited also by the subdiffusion $Z_\alpha(t)$.

4.3 Subdiffusive Black Scholes Model

The subdiffusive Black-Scholes model, an extension of the classical Black-Scholes model, incorporates the element of subdiffusion to better capture the complex behaviors observed in financial markets. Subdiffusion, characterized by slower-than-normal diffusive processes, aligns more closely with the actual movement of asset prices, especially in markets experiencing low volatility or other anomalies not accounted for in traditional models. By integrating this model, we were able to generate sample paths that reflect a more exact view of market movements, thereby providing a robust foundation for our hedging strategy.
In this section we will present a generalization of the Black Scholes model that takes into consideration those subdiffusive characteristics of a financial market.

According to Black Scholes Model asset prices follow a geometric Brownian motion $Z(t)$ that satisfies the following stochastic differential equation, [15]:

$$dZ(t) = \mu Z(t)dt + \sigma Z(t)dB(t) \tag{4.5}$$

where $B(t)$ denotes the Brownian motion, $\mu \in \mathbb{R}$ is the drift and $\sigma > 0$ is the volatility. For an arbitrary initial value Z_0 the above SDE has the analytic solution:

$$Z(t) = Z_0 \exp((\mu - \frac{\sigma^2}{2})t + \sigma B(t)).$$

Since we want to also consider periods of constant prices, the prices are modelled by the subordinated geometric Brownian motion, defined as, [11]:

$$Z_\alpha(t) = Z(S_\alpha(t)) = Z_\alpha(0) \exp((\mu - \frac{\sigma^2}{2})S_\alpha(t) + \sigma B(S_\alpha(t))). \tag{4.6}$$

$S_\alpha(t)$ is assumed to be independent from $B(t)$. This equation 4.6 can also be represented as Langvin type stochastic differential equation of the form, [11]:

$$dZ_\alpha(t) = \mu Z_\alpha(t)dS_\alpha(t) + \sigma Z_\alpha(t)dB(S_\alpha(t))$$

Simulation of the trajectories of the subdiffusive geometric Brownian motion can be performed using the following approximation $\{S_{\alpha,\delta}\}_{t\geq 0}$ of the $\{S_\alpha\}_{t\geq 0}$

$$S_{\alpha,\delta}(t) = (\min\{n \in \mathbb{N} : U_\alpha(\delta n) > t\} - 1)\delta$$

where $\delta > 0$ is the step length and $U_\alpha(\tau)$ is the α-stable subordinator, [11]. The generation of $U_\alpha(\delta n)$ is done by the Euler method of summing up the increments of the subordinator $U_\alpha(\tau)$

$$\begin{aligned} U_\alpha(0) &= 0 \\ U_\alpha(\delta n) &= U_\alpha(\delta(n-1)) + \delta^{1/\alpha}\xi_n \end{aligned} \tag{4.7}$$

where $\xi_n, n \in \mathbb{N}$ are the i.i.d. totally skewed positive α-stable random variables that are generated like this:

$$\xi_n = \frac{\sin(\alpha(V + c_1))}{(\cos(V))^{1/\alpha}} \left(\frac{\cos(V - \alpha(V + c_1))}{W} \right)^{(1-\alpha)/\alpha}$$

where $c_1 = \pi/2$, V is uniformly distributed on $(-\pi/2, \pi/2)$ random variable and W has the exponential distribution with mean one.

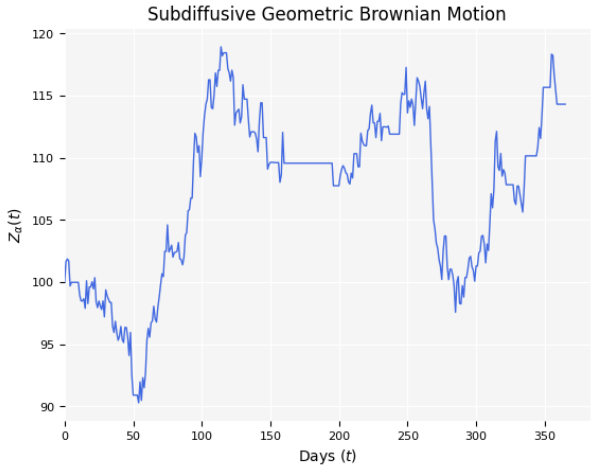

Figure 4.1: Trajectories of the subdiffusive geometric Brownian motion with parameters $S_0 = 100, \sigma = 0.2, \mu = 0, \alpha = 0.9, T = 1$ and $\alpha = 0.8$

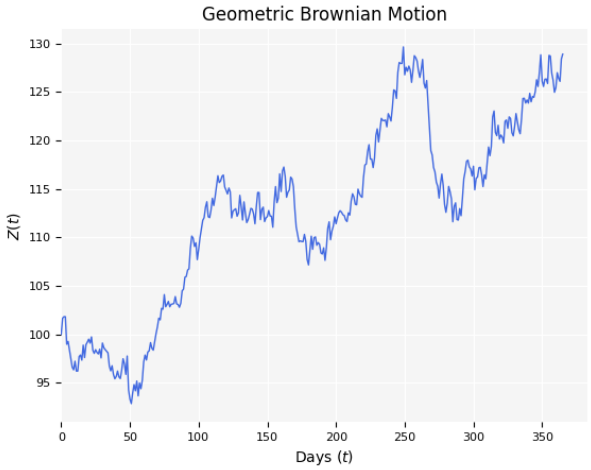

Figure 4.2: Trajectories of the geometric Brownian motion with parameters $S_0 = 100, \sigma = 0.2, \mu = 0, \alpha = 0.9$ and $T = 1$.

The comparison of trajectories between the traditional geometric Brownian motion (GBM) 4.2 and its subdiffusive alternative 4.1provides insightful contrasts. In the classical GBM, the movement is continuous and smooth. However, in the subdiffusive version of GBM,

there is a notable presence of constant, unchanging periods within the trajectory. In our case, these stationary phases represent periods of constant price. Both $Z(t)$ and $Z_\alpha(t)$, representing the classic and subdiffusive processes respectively, share the same spatial characteristics, implying that their movements in space follow similar patterns. Despite this spatial similarity, their temporal behavior sets them apart significantly. This difference in the time aspect is controlled by the subordinator S_α that introduces flat periods in the trajectory. These flat periods, where there is an apparent lack of movement or change, are a defining feature of the subdiffusive GBM.

In order to define a Black Scholes formula with subdiffusive dynamics, we first need to set the conditions and requirements of the market model. Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a probability space. Recall, the filtration $(\mathcal{F}(t))_{t\in[0,T]}$ contains all the information regarding the assets available at the time t. Let $(Z_\alpha(t))_{t\in[0,T]}$ denote the prices of the financial assets. One of the first requirements of the asset pricing rules is that the market model is arbitrage free.

The first fundamental theorem of asset pricing (see Theorem 2.1.7) states that the market model on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is arbitrage free if and only if there exists a risk neutral probability measure $\mathbb{Q}$ equivalent to $\mathbb{P}$ under which $(Z_\alpha(t))_{t\in[0,T]}$ is a martingale with respect to $\mathbb{Q}$. This implies, we have to define that risk neutral measure $\mathbb{Q}$ under which all financial assets have the same expected rate of return.

Theorem 4.3.2: Let the probability measure $\mathbb{Q}$ be defined as, [12]:

$$\mathbb{Q}(A) = \int_A \exp(-\gamma B(S_\alpha(t)) - \frac{\gamma^2}{2} S_\alpha(t)) d\mathbb{P} \tag{4.8}$$

where $\gamma = \dfrac{\gamma + \frac{\sigma^2}{2}}{\sigma}$ and $A \in \mathcal{F}$.

Using the risk neutral measure $\mathbb{Q}$ as defined above we have now achieved that by the first fundamental theorem of asset pricing, our market model is arbitrage free, [12]. The second FTAP (see Theorem 2.1.14) states that the market is complete if and only if there exists one equivalent martingale measure $\mathbb{Q}$. But in this case that is does not hold, since $\mathbb{Q}$ defined in 4.8 is not unique [12] so therefore our subdiffusive market is incomplete.

In the next part we will derive the Black Scholes formula for subdiffusive characteristics, but before let us take a look into the classical Black Scholes formula for pricing European options.

European call option is a contract between two parties, a buyer and a seller. A buyer of the option has a right but not an obligation to buy the underlying Z at maturity T for a specified price K, called strike price. The payoff is

$$C = \max\{(Z(T) - K), 0\}$$

where $Z(T)$ is the spot price of the underlying asset at maturity. On the other hand, European put option is a contract between a buyer and a seller, where the buyer of the

option has a right, but not an obligation to sell the underlying Z at maturity T for a strike price K. The value of the put option at maturity is

$$P = \max\{(K - Z(T)), 0\}.$$

The fair price of the European call option calculated by the Black Scholes model is:

$$C_{BS}(Z_0, K, T, \sigma, r) = \phi(d_1)Z_0 - \phi(d_2)Ke^{-r(T-t)} \tag{4.9}$$

where

$$d_1 = \frac{log(\frac{Z_0}{K}) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$$

and

$$d_2 = d_1 - \sigma\sqrt{T-t}.$$

ϕ denotes the standard normal cumulative distribution defined as:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt.$$

The fair price of a European put option is given by the put-call parity:

$$P_{BS}(Z_0, K, T, \sigma, r) = C_{BS}(Z_0, K, T, \sigma, r) - Z_0 + Ke^{-r(T-t)}. \tag{4.10}$$

Black Scholes Delta is the first derivative of the option price with respect to the underlying price :

$$\frac{\partial C_{BS}}{\partial S} = \phi(d_1). \tag{4.11}$$

Delta will be needed for our numerical experiments in the 5th Chapter.

Let us derive the subdiffusive Black Scholes formula for the price of European options.

Theorem 4.3.3, [12]: Let the martingale measure $\mathbb{Q}$ be given by 4.8 and let assume that the asset prices follow a subdiffusive geometric Brownian motion $Z_\alpha(t)$. Then, the price of an European call option is defined as:

$$\begin{aligned} C_{BS}^{sub}(Z_0, K, T, \sigma, r, \alpha) &= \mathbb{E}[C_{BS}(Z_0, K, S_\alpha(T), \sigma, r)] \\ &= \frac{1}{T^\alpha} \int_0^\infty C_{BS}(Z_0, K, x, \sigma, r) g_\alpha\Big(\frac{x}{T^\alpha}\Big) dx \end{aligned} \tag{4.12}$$

and respectively the price of an European put option:

$$\begin{aligned} P_{BS}^{sub}(Z_0, K, T, \sigma, r, \alpha) &= \mathbb{E}[P_{BS}(Z_0, K, S_\alpha(T), \sigma, r)] \\ &= \frac{1}{T^\alpha} \int_0^\infty P_{BS}(Z_0, K, x, \sigma, r) g_\alpha\Big(\frac{x}{T^\alpha}\Big) dx \end{aligned} \tag{4.13}$$

where C_{BS} and P_{BS} are given by 4.9, 4.10 respectively. Here, $g_\alpha(z)$ is a probability density function of $S_\alpha(1)$ and can be expressed as Fox-H-function, [12]:

$$g_\alpha(z) = H_{11}^{10}\Big(z|_{(0,1)}^{(1-\alpha,\alpha)}\Big)$$

which can numerically be calculated for only some values of α, for example for $\alpha = 1/2$, [12], but it is more convenient to approximate the expected values in the above expressions 4.12, 4.13 using Monte Carlo methods by simulating trajectories of the inverse α-stable subordinator $S_\alpha(t)$, [12]. This can be done by generating random samples from the distribution of the inverse α-stable subordinator and using them to estimate the desired expected values, [12]. With the help of the property

$$S_\alpha(t) \sim \Big(\frac{t}{U_\alpha(1)}\Big)^\alpha$$

mentioned in 4.4, we can express the estimator of the expected value defined in 4.12, 4.13 as, [11]:

$$\hat{C}_{BS}^{sub}(Z_0, K, T, \sigma, r, \alpha) = \frac{1}{n}\sum_{i=1}^{n} C_{BS}(Z_0, K, S_i^\alpha(T), \sigma, r)$$

and

$$\hat{P}_{BS}^{sub}(Z_0, K, T, \sigma, r, \alpha) = \frac{1}{n}\sum_{i=1}^{n} P_{BS}(Z_0, K, S_i^\alpha(T), \sigma, r)$$

where $S_i^\alpha(T) \sim \Big(\frac{T}{U_i^\alpha(1)}\Big)^\alpha$ and $U_i^\alpha(1)$ are independent and identically distributed (i.i.d.), totally skewed positive α-stable random variables (therefore $S_i^\alpha(T)$ are also the i.i.d.). The realizations of $U_i^\alpha(1)$ are generated in the following way, [11]:

$$U_i^\alpha(1) = \frac{\sin(\alpha(V+c_1))}{(\cos(V))^{1/\alpha}}\Big(\frac{\cos(V-\alpha(V+c_1))}{W}\Big)^{(1-\alpha)/\alpha}$$

where $c_1 = \pi/2$, V is uniformly distributed on $(-\pi/2, \pi/2)$ random variable and W has the exponential distribution with mean one.

Using this approximation, we have calculated a subdiffusive Black Scholes price of European option for different values of α ranging from 0.1 to 1 and with parameters $\sigma = S_0 = 1, K = 2, T = 5$. Those price paths are presented in the figure 4.3 where $\alpha = 1$ is following the classical Black Scholes price. It is clearly visible in the figure 4.3, for small exercise times T the classical BS price is underestimated, while for larger T it is overestimated in comparison to subdiffusive BS price. Figure 4.4 is more precisely presenting the fact, that if α is set to 1 in the subdiffusive model, then the price exactly follows the classical BS model.

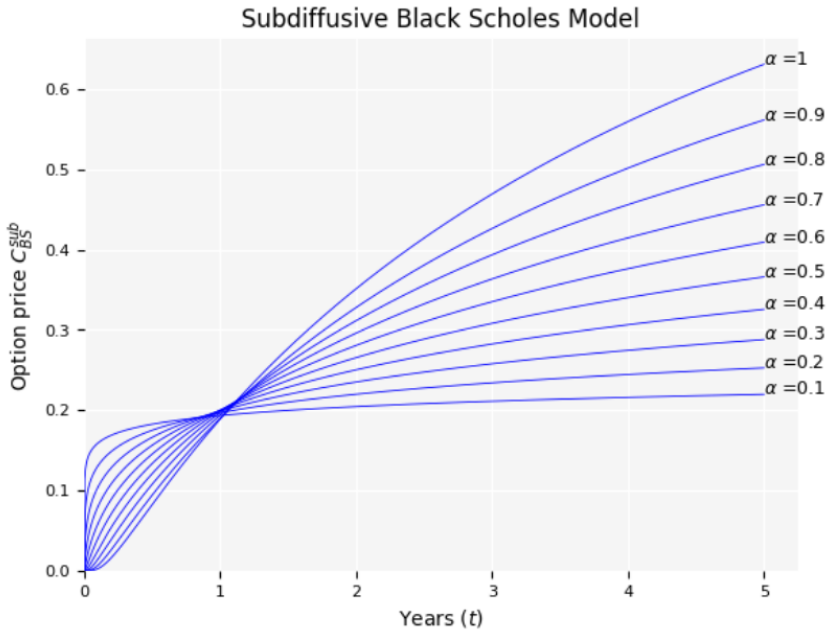

Figure 4.3: Subdiffusive Black Scholes prices with parameters $\sigma = S_0 = 1, K = 2, T = 5$ years and for different values of α

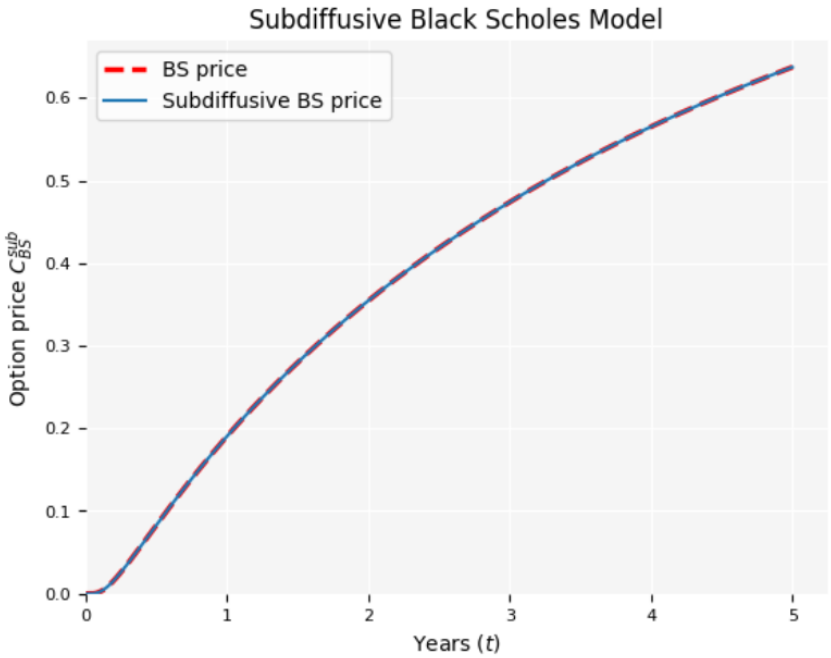

Figure 4.4: Classical Black Scholes price compared to subdiffusive Black Scholes price with $\sigma = S_0 = 1, K = 2, T = 5$ and $\alpha = 1$

In the next example we have chosen a time horizon of 30 days and parameters $S_0 = K = 1, \sigma = 1, T = 30/365$. In the figure 4.5 the effect of the subdiffusive parameter α is

clearly visible, with lower α values resulting in a steeper initial increase in option price relative to higher α values. The impact of α is evident even over this shorter time frame.

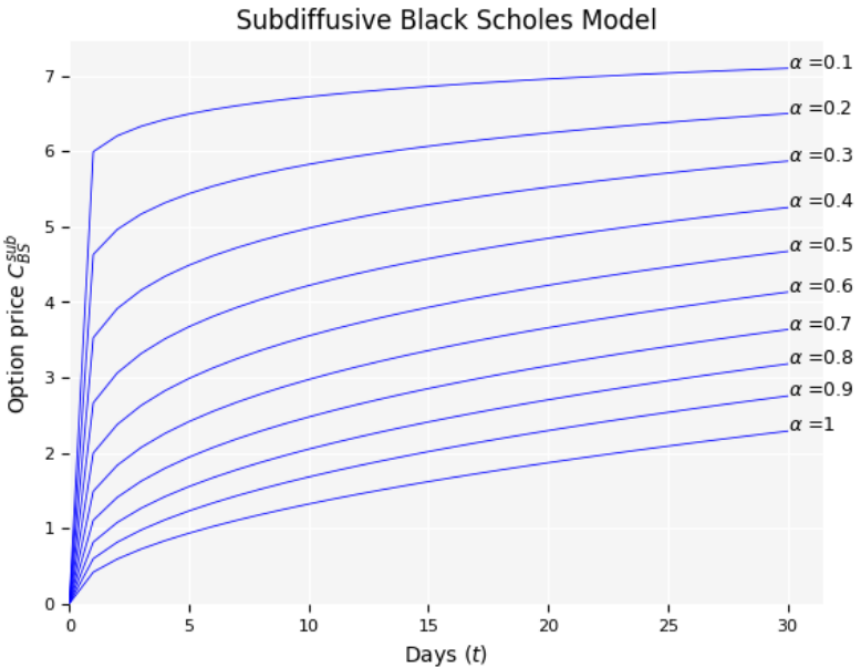

Figure 4.5: Prices of European call option using subdiffusive Black Scholes model with different values of α.

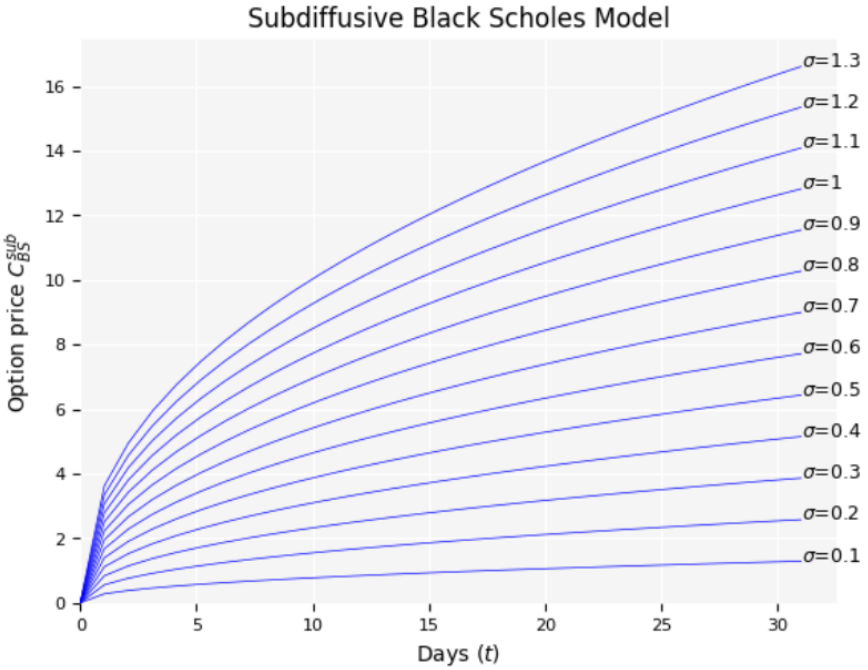

Figure 4.6: Prices of European call option using subdiffusive Black Scholes model with different values of σ.

Different values of α are not the only thing that is influencing the option price. Choosing the appropriate volatility to model the option price, also in the subdiffusive model, is of a big importance. There is a clear distinction between each curve in the figure 4.6 , showing a progressive increase in option price with the increase in volatility. The difference between the curves suggests that even small changes in volatility can significantly affect option pricing. That being said, to fit the subdiffusive BS model to the real data, the calibration of the parameters α, σ and μ is very important.

5 Numerical experiments

After introducing the optimal hedging problem in Section 2 and discussing the numerical approximation of its solution using neural networks in Section 3, our focus now shifts to conducting numerical experiments. These experiments aim to demonstrate the practical viability of our approach to hedging financial instruments using a neural network-based methodology. We start by presenting the details of our modeling choices.

5.1 Setting

Time horizon: We have used a time interval of 30 days with daily re-balancing, so $T = 30/365$. This period is broken down into discrete trading dates, each denoted as $t_k = k/365$ for $k = 0, ..., 30$.

Subdiffusive parameters: Central to our analysis was the generation of sample paths for asset prices, a task for which we employed a subdiffusive Black-Scholes model. This choice was pivotal in creating a realistic, yet computationally manageable, framework for simulating market dynamics. We have used this model as a benchmark, meaning we compared the results of our deep hedging model with the prices generated by the subdiffusive Black-Scholes model. We consider a European call option with payoff $K = (S_T^1 - K)^+$ and strike $K = S_0 = 100$. We have chosen a volatility $\sigma = 0.2$, risk free rate r and dividend yield q are set to zero and for the subdiffusive parameter we took $\alpha = 0.9$. In our simulation we won't consider the trading costs, so $C_T(\delta) = 0$.

Neural Network Design: As we discussed in the previous chapter, $\delta_k \in \mathbb{R}^d$ is representing the number of units of each asset that the investor is holding at time t_k and is modelled by semi-recurrent neural network. This network is composed of a feedforward architecture with two hidden layers:

$$\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}^\theta)$$

where F^{θ_k} is a feed forward neural network with two hidden layers so $L = 3$. The network's input is a function of the sequence of asset prices up to the current trading date, providing a comprehensive view of market dynamics, but in order to compare the neural network strategy to our benchmark, the network input is chosen as $I_k = \log(\overline{S}_k)$. Furthermore, the dimension of the input layer being $N_0 = 2d$ and dimension of the inner layers, $N_1 = N_2 = d + 15$ and lastly $N_3 = d$ with $d = 1$. The activation function across

the network is the leaky ReLU function, [16]:

$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ x\alpha & \text{for } x < 0 \end{cases}$$

where x is the input of the nonlinear activation f and α is a coefficient controlling the slope of the negative part.
Lastly, the risk measure will be the entropic risk measure, i.e.

$$\rho(X) = \frac{1}{\lambda} \log \mathbb{E}[\exp(-\lambda X)]$$

with risk-aversion parameter $\lambda = 1$. Each layer's weights and biases are dynamically adjusted and optimized to refine the model. These parameters are crucial in determining the network's performance and are finely tuned during the model's training phase. The model was implemented in Python using TensorFlow for building and training the neural networks. For training process Adam algorithm with learning rate of 0.01 and a batch size of 256 has been used but we did not apply batch normalization technique. The weights were initialized using the He uniform initializer which initializes the weights with values drawn from a uniform distribution within a range, which is determined based on the number of input units in the weight tensor [16]. The He uniform initializer was chosen because it is particularly suited for layers with ReLU (Rectified Linear Unit) activations, [16].It's designed to keep the scale of the gradients roughly the same in all layers, [16]. In deep networks, this initializer helps in mitigating the problem of vanishing gradients, leading to more effective training, [16].

Simulation of stock prices: We have simulated random paths of stock prices using the subdiffusive geometric Brownian motion described in the section 2. The algorithm has generated 120 000 paths that we have split into training and testing data. On the figure 5.1. those straight lines which that are significant for a subdiffusive model and represent the times of constant prices, are clearly visible.

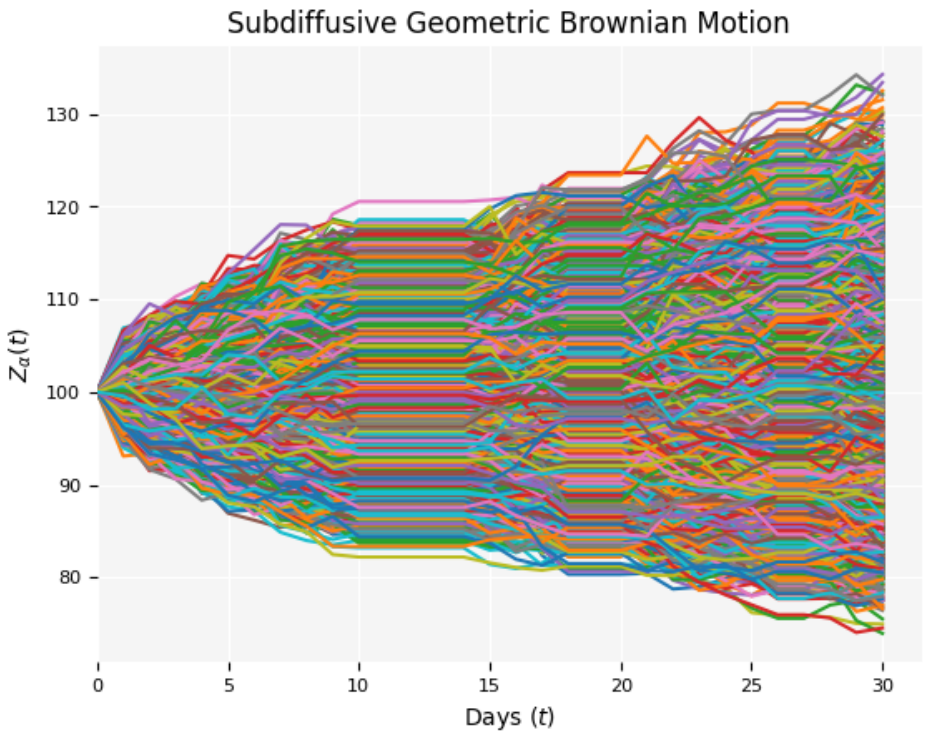

Figure 5.1: Simulation of stock prices driven by the subdiffusive geometric Brownian Motion

5.2 Results

After running the model we obtain the following prices presented in the figure 5.2. Note that we have calculated the subdiffusive Black Scholes price based on the approximation scheme defined in the section 4.3. Subdiffusive Black Scholes price is expectedly higher as classical Black Scholes price which we have seen also in the figure 4.1. As you can see that there is not much difference between the price calculated using simple neural network structure and the price calculated using reccurent neural network.

Black Scholes Price (Classical Model)	Black Scholes Price (Subdiffusive Model)	Deep Hedging Price (Simple Network)	Deep Hedging Price (Reccurent Network)	Risk Neutral Price
2.29	2.59	2.85	2.87	2.95

Figure 5.2: Option prices at maturity $T = 30/365$ based on the following parameters: $S_0 = K = 100, \sigma = 0.2, r = q = 0, \alpha = 0.9$

The risk-neutral price of an option is the expected discounted value of the option's payoff under the risk-neutral probability measure. In our case the subdiffusive geometric Brownian Motion is generating a bit higher future stock prices than the classical geometric Brownian motion. This variance accounts for the higher risk-neutral price observed.

The basis for this higher risk-neutral price lies in the simulated stock prices at maturity, which, as depicted in Figure 5.1, vary between 75 and 140.

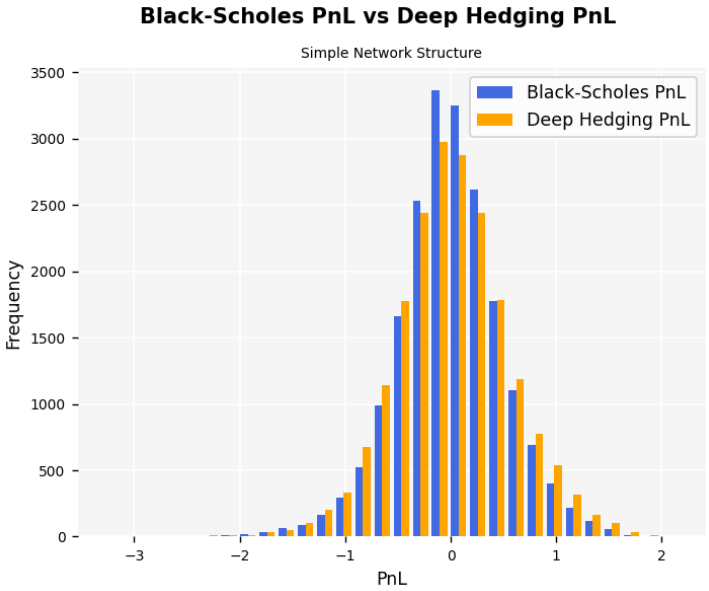

Figure 5.3: Histogram of the terminal hedging error evaluated on the deep hedging model and on the subdiffusive Black Scholes model

Figure 5.2.showcases a comparison of the profit and loss at maturity, expressed as $p - Z + (\delta \cdot S)_T$ that was calculated once for deep hedging and once for the subdiffusive Black Scholes model with the subdiffusive Black Scholes price p being charged on both. As one can see, hedging performance of both strategies δ^H and δ^δ is very similar from which we could imply that the deep hedging model is able to approximate the subdiffusive Black Scholes model. The histogram in the figure further highlights several noteworthy observations.

The Profit and Loss (PnL) distributions for both strategies are centered around zero, indicating that, on average, the strategies are balanced and tend to break even. The distribution of the Black-Scholes PnL appears to have a slightly wider spread than the Deep Hedging PnL, indicating more significant losses and gains. The Deep Hedging strategy seems to have a higher peak and a narrower distribution, suggesting that it more consistently achieves a PnL close to zero, which could be interpreted as a more risk-averse strategy or one that is better at minimizing the cost of hedging. Overall, the histogram suggests that the Deep Hedging model might be offering a more stable and less volatile hedging performance compared to the traditional Black-Scholes model.

Another important proof of the efficiency of deep hedging model is the comparison of subdiffusive Black Scholes delta with deep hedging delta. In this case delta has been calculated for three different time intervals, $t = 1, 15, 29$ and as illustrated in the figure 5.3. deep hedging delta is almost exact to the subdiffusive Black Scholes delta which again proves that deep hedging model is able to effectively approximate subdiffusive Black Scholes model.

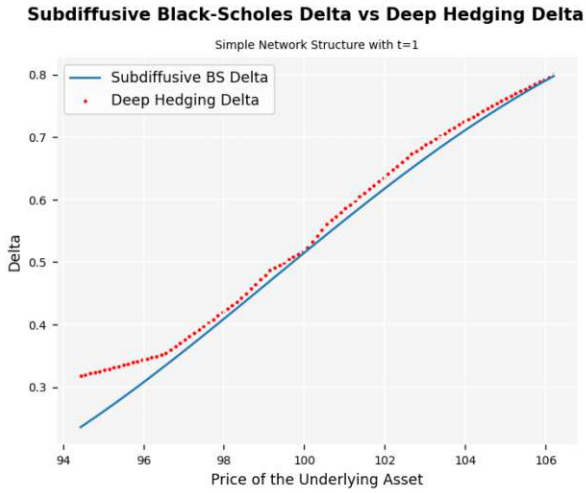

Figure 5.4: Comparison of subdiffusive Black Scholes delta and deep hedging delta for time interval $t = 1$ and $\alpha = 0.9$.

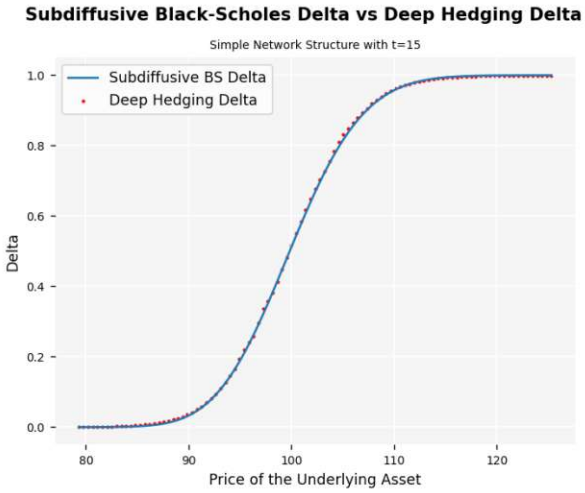

Figure 5.5: Comparison of subdiffusive Black Scholes delta and deep hedging delta for time interval $t = 15$ and $\alpha = 0.9$.

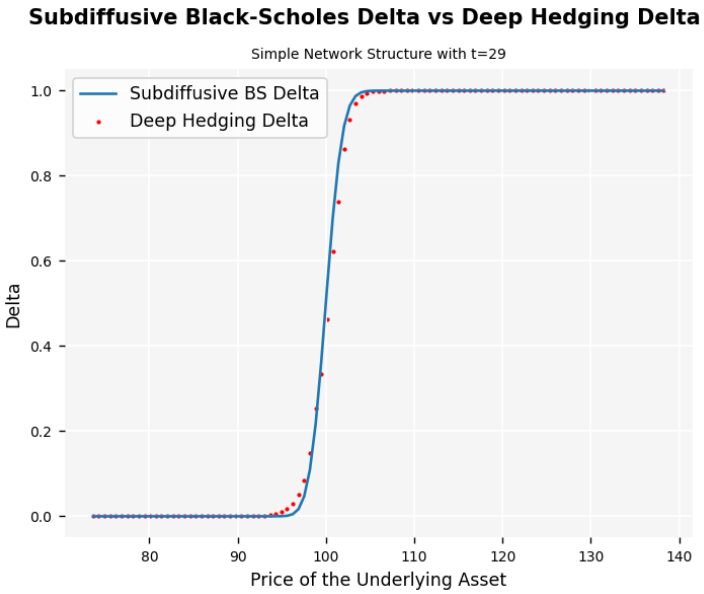

Figure 5.6: Comparison of subdiffusive Black Scholes delta and deep hedging delta for time interval $t = 29$ and $\alpha = 0.9$.

PnL distribution of two different network models: a Simple Network and a Recurrent Network is compared in the figure 5.7. Both distributions are centered around 0, which might indicate that neither model has a significant bias towards profit or loss. They appear normally distributed, suggesting that the results of both models on the trading days are symmetrical around the mean. Simple Network PnL: The blue bars represent the frequency of the PnL results for the Simple Network. It appears to have a narrower distribution, suggesting less variance in the outcomes compared to the Recurrent Network. This could mean that the Simple Network has a more consistent performance, but not necessarily a better one. Recurrent Network PnL: The orange bars represent the Recurrent Network. This network has a wider distribution, indicating a higher variance in outcomes. It seems to have more frequent extreme results (both profits and losses), which might suggest that the Recurrent Network takes on higher risk, potentially leading to higher rewards but also higher losses. Overlap: There is significant overlap in the middle of the distribution, but the tails (especially for losses of around -2 to -3 PnL and profits of around 1 to 2 PnL) show some divergence. The Recurrent Network appears to have a slightly longer tail on the profit side, which might mean it is capable of achieving higher profits on some days compared to the Simple Network.

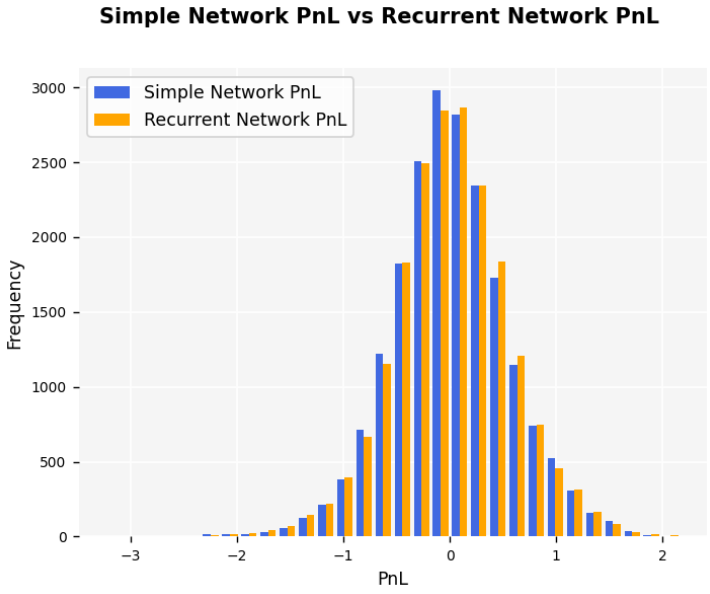

Figure 5.7: Comparison of PnL from the simple network model and from the recurrent network model

Furthermore, we ran the model three times and calculated the delta and as seen in the figure 5.6. the Deep Hedging Deltas for all three runs closely follow the subdiffusive Black-Scholes Delta, indicating that the deep learning model is able to approximate the Delta of the theoretical model quite well. There is a slight divergence in the behavior of the Deep Hedging Delta and the Subdiffusive Black-Scholes Delta, especially at the lower and higher ends of the underlying asset price spectrum. The consistency across the three runs of the Deep Hedging model suggests that the model's performance is stable and not highly sensitive to different initializations or data samples, at least within the range of

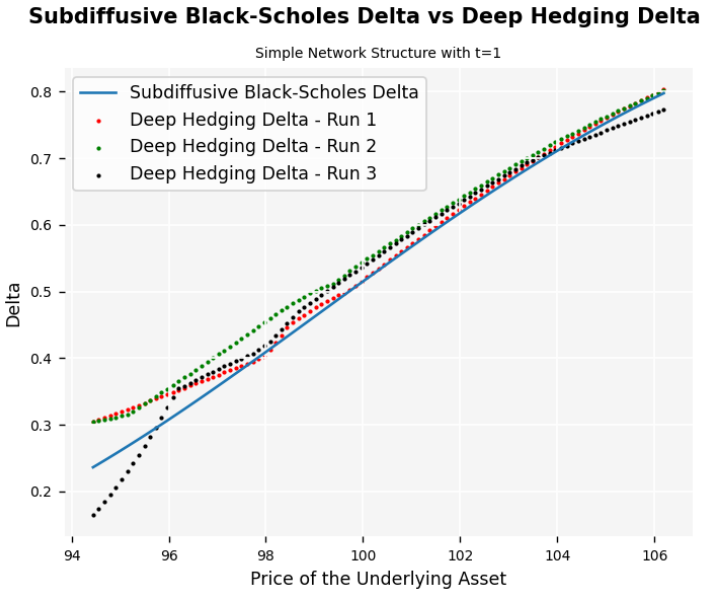

Figure 5.8: Comparison of delta values between the subdiffusive Black Scholes and deep hedging models for the time interval at $t = 1$.

this experiment.
Lastly, we run our model again but this time incorporating the assumption that the actual volatility is a multiple of the simulated volatility. We introduce a new parameter called $\zeta = 1.1$ which adjusts the real volatility to $\sigma_{real} = \sigma * \zeta$. We simulate stock prices that follow subdiffusive Brownian Motion and compute option price based on our trained deep hedging model. The results, as depicted in the figure 5.9, show a PnL distribution that aligns with that of the subdiffusive Black Scholes model, affirming the model's accuracy. The peak of the PnL distribution of both models is close to 0, but the tallest bar is slightly in the negative PnL region, suggesting that the most common outcome was a small loss. The fact that the tallest bars are on the negative side for both strategies may suggest that under the current settings, and within the simulated environment, losses are slightly more common than gains. However, it doesn't necessarily indicate poor performance. The goal of hedging is often to reduce risk, which might mean accepting small, frequent losses to avoid large, infrequent losses. Still, both distributions are centered around zero, which is typical for hedging strategies aimed at risk minimization rather than profit maximization. The deltas calculated by both the Subdiffusive Black-Scholes model and the Deep Hedging model are almost identical across the range of prices, as illustrated in the figure 5.10. This close alignment suggests that, for this range of underlying asset prices and at this particular time step, both the traditional model and the more complex Deep Hedging model yield very similar hedging strategies.The similar delta values indicate that, in terms of delta hedging, the simple network structure used in Deep Hedging does not significantly diverge from the Subdiffusive Black-Scholes model under the conditions tested.

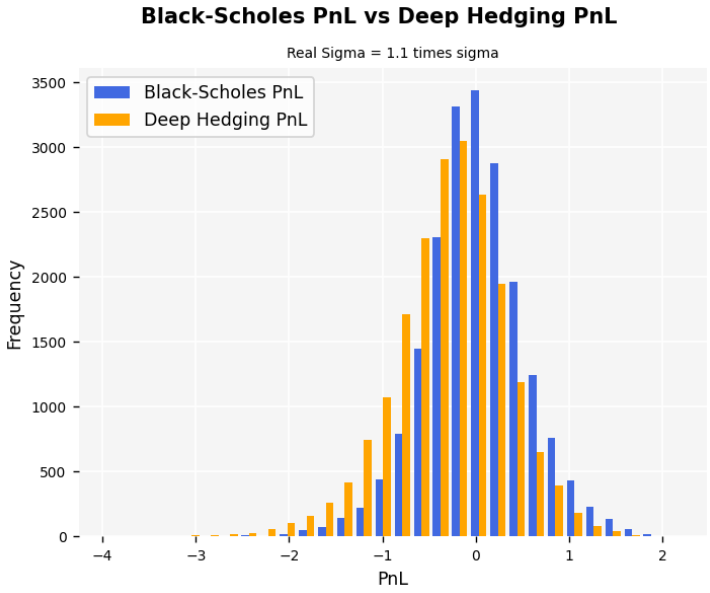

Figure 5.9: Comparison of PnL distribution of subdiffusvie Black Scholes model and deep hedging model for parameters:
$T = 30/365, S_0 = K = 100, r = q = 0, \sigma_{real} = \sigma * \zeta, \zeta = 1.1, \alpha = 0.9$.

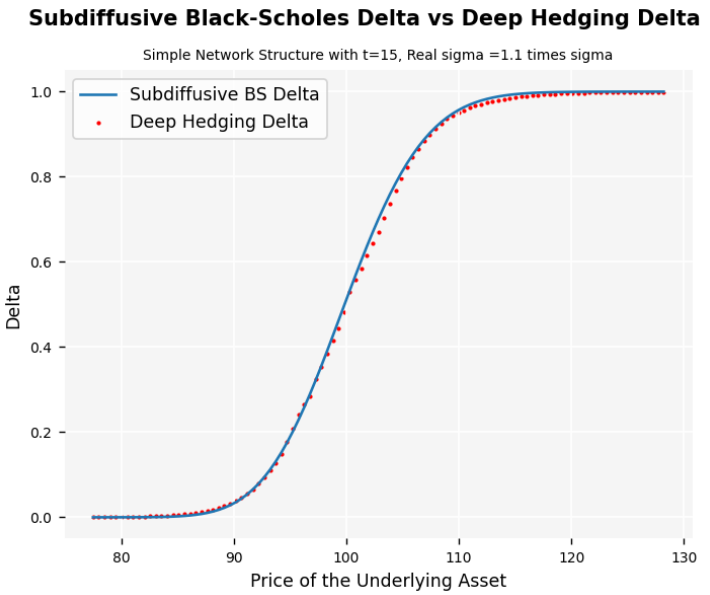

Figure 5.10: Comparison of subdiffusive Black Scholes delta and deep hedging delta for time interval $t = 15$.

6 Conclusion

This thesis considers the problem of hedging a portfolio in incomplete markets. As a solution to that it presents the deep hedging model implemented by Buehler, [1]. It shows that this approach does not depend on specific market dynamics and does not calculate Greeks based on the typical Black Scholes formula, meaning, it is independent from any financial models. As optimality criteria a convex risk measure such as entropic risk measure was taken.

Another core topic of this thesis was also subdiffusive Black Scholes model which is used in incomplete markets where the prices have constant periods. We show that the price was driven by the subdiffusive geometric Brownian motion and we provide an approximation scheme on the simulation of those price paths that was introduced in [12]. We also present the subdiffusive Black Scholes formula and provide a numerically possible estimation.

Lastly, after lying out the fundamentals and theoretical points of neural networks and subdiffusive Black Scholes model, we demonstrate this theory on numerical examples performed in Python. The goal was to calculate the option's price of a short call option. Our neural network was tasked with determining the optimal quantity of financial instruments to hold at each trading interval, based on the input of asset prices over time. The model was trained based on the sample paths for asset prices, which were generated through the subdiffusive Black-Scholes model. The neural network, designed with feed-forward architecture and ReLU activation functions, was fine-tuned through a training process involving the Adam optimization algorithm. This choice was pivotal in creating a realistic, yet computationally manageable, framework for simulating market dynamics. We compared the results of our deep hedging model with the prices generated by the subdiffusive Black-Scholes model. This comparison was crucial in assessing the efficacy of the deep hedging method. In our analysis, we observed that the deep hedging model, with its capacity to learn and adapt from the input data, was able to approximate the subdiffusive Black-Scholes price well.

7 Bibliography

[1] Hans Bühler and Lukas Gonon and Josef Teichmann and Ben Wood. *Deep Hedging.* arxiv, 2018, https://arxiv.org/abs/1802.03042.

[2] Andrea Pascucci. *PDE and Martingale Methods in Option Pricing.* Springer-Verlag, Italia, 2011.

[3] *Reinforcement Learning 101.* https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292. Accessed November, 2023.

[4] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* Cambridge, MA : The MIT Press, 2018.

[5] *Reinforcement Learning.* Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[6] *Deep Reinforcement Learning.* Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[7] *Mathematics behind the Neural Network.* `https://studymachinelearning.com/mathematics-behind-the-neural-network/`. Accessed November, 2023.

[8] *Math behind Neural Networks.* https://towardsdatascience.com/introduction-to-math-behind-neural-networks-e8b60dbbdeba. Accessed November, 2023.

[9] Kurt Hornik. *Approximation Capabilities of Multilayer Feedforward Networks.* Neural Networks, Volume 4, Issue 2, 1991.

[10] Helmut Bölcskei and Philipp Grohs and Gitta Kutyniok and Philipp Peterse. *Optimal Approximation with Sparsely Connected Deep Neural Networks.* arXiv, 2018, https://arxiv.org/abs/1705.01714.

[11] Sebastian Orzeł and Aleksander Weron. *Calibration of the Subdiffusive Black–Scholes Model.* Acta Physica Polonica B. 4150, 2010.

[12] Marcin Magdziarz. *Black-Scholes Formula in Subdiffusive Regime.* Journal of Statistical Physics, 136. 553-564. 10.1007/s10955-009-9791-4, 2009.

[13] Marcin Magdziarz. *Stochastic representation of subdiffusion processes with time-dependent drift.* Stochastic Processes and their Applications, Volume 119, Issue 10, 2009.

[14] Antonis Papapantoleon. *An Introduction to Levy Processes with Application in Finance.* arXiv, 2008, https://arxiv.org/abs/0804.0482.

[15] Geometric Brownian motion. Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[16] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.* arXiv, 2015, https://arxiv.org/abs/1502.01852.

[17] Diffusion. Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[18] Brownian Motion. Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[19] Richard F. Bass. Stochastic Processes. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511997044, 2011.

[20] Anomalous Diffusion. Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[21] Fokker–Planck Equation. Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[22] R. Metzler, J. Klafter.The random walk's guide to anomalous diffusion: a fractional dynamics approach. Phys. Rep., 339 (2000), pp. 1-77.

[23] Oliver C. Ibe. Markov Processes for Stochastic Modeling. Elsevier Inc., 2013.

[24] T.Rheinländer. *Hedging Derivatives.* World Scientific, 2011.

[25] Paul Embrechtsy and Makoto Maejimaz. *An introduction to the theory of selfsimilar stochastic processes.* International Journal of Modern Physics B 2000 14:12n13, 1399-1420.

[26] Steven P. Lalley. *Levy Processes, Stable Processes, and Subordinators.* `https://galton.uchicago.edu/~lalley/Courses/385/LevyProcesses.pdf`. Accessed November, 2023.

[27] Goodfellow-et-al-2016. *Deep Learning.* MIT Press, 2016.

[28] Aurelien Geron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, Sebastopol, CA, 6 edition, 2017.

[29] Hans Föllmer and Alexander Schied. *Stochastic Finance: An Introduction in Discrete Time.* Walter de Gruyter GmbH & Co. KG, 2011.

[30] Semi-continuity. Wikipedia, The Free Encyclopedia. Accessed November, 2023.

[31] Dimitri P. Bertsekas. *Convex Optimization Theory.* Athena Scientific, 2009.