Deep Hedging of Basis Risk

Olatomiwa Ayooluwa Adewusi

A dissertation submitted to the Faculty of Commerce, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

October 12, 2022

MPhil in Mathematical Finance,
University of Cape Town.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Philosophy in the University of the Cape Town. It has not been submitted before for any degree or examination in any other University.

October 12, 2022

Abstract

Basis risk arises when the writer of a contingent claim cannot trade in the underlying asset and must use a correlated proxy asset to hedge the contingent claim. Suppose the proxy asset is not perfectly correlated to the underlying. In that case, there is a risk that the hedge portfolio does not precisely track the contingent claim, which may lead to significant losses at maturity. There are several existing approaches to hedging and pricing of contingent claims in the presence of basis risk. The existing approaches considered in this dissertation are based on the quadratic and exponential utility functions. This dissertation compares these current approaches to a new policy that parameterises the hedge parameters as a recurrent neural network at each rebalancing date. This new approach is called Deep Hedging, and under this approach, the hedge parameters are determined in a model agnostic way. This is achieved using Long Short-Term Memory networks written in TensorFlow. This allows one to make the hedge parameters at each time point a function of current market data and previous hedging decisions. Deep Hedging is Greek-free and more easily allows for the incorporation of other market frictions, like transaction costs, compared to existing approaches. Lastly, we can find optimal hedging strategies under coherent risk metrics, like expected shortfall, using the Deep Hedging approach and given a price. By fixing the volatility and correlation parameters, Deep Hedging produces results that are comparable to the best existing strategies, in both complete and incomplete market settings, across a variety of moneyness levels.

Acknowledgements

I want to thank my supervisor, Tom, for all the advice that led to this final document. I would also like to say how appreciative I am that he gave me such a stimulating topic. I have been interested in machine learning applications in finance for some time now, and Deep Hedging provided the perfect foray into the field. I would also like to thank my parents, Ola and Kike, for all the support they have given me over the years. Your guidance has made me into the man I am today, and for that, I am forever grateful.

All glory to God.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Basis Risk

Basis risk arises when the writer of a contingent claim can not trade in the underlying asset. Usually, the writer can not trade in the underlying because it is unsuitable for hedging — an illiquid asset — or it may be impractical to hedge — an index. In some cases, the underlying asset may even be unavailable for trade, and the writer can only observe the underlying price. Not being able to trade in the underlying poses a risk to the writer, as when she wants to hedge the contingent claim, she will have to use a correlated asset. Suppose the hedging asset is not perfectly correlated to the underlying. In that case, this opens the writer to the risk that the hedging portfolio does not precisely track the claim (Figlewski (1984)), possibly leading to significant losses at maturity. Moreover, Davis (2006) it has shown that even hedging assets nearly perfectly correlated to the underlying pose significant basis risk, demonstrating the need for basis risk to be carefully considered when hedging contingent claims. There are also wide-reaching applications of basis risk. These include hedging European options, pension products (Coughlan *et al.* (2011); Li and Hardy (2011)), and weather derivatives (Woodard and Garcia (2007)). Given the diversity of domains where basis risk exists, and the significant threat, it is reasonable to propose a method that minimises this risk.

1.2 Incomplete Markets and Existing Strategies

Since we cannot trade in the underlying asset, there is no self-financing replicating portfolio, and the contingent claim in question is not attainable. The absence of an underlying asset moves us from the realm of complete to incomplete markets. Therefore there is no unique, preference-independent risk-neutral measure to obtain prices, but infinitely many preference-dependent measures (McWalter (2006)). Moreover, hedging is no longer risk-free as basis risk introduces unhedgeable risk

into our model; this implies that, when one is hedging in incomplete markets, one needs to find a price and trading strategy that minimises the unhedgeable risk. Or given a price, one needs to minimise the unhedgeable risk (Davis (2006); McWalter (2006)). Given that we are in an incomplete market, several authors have proposed approaches to minimise basis risk and obtain prices when dealing with contingent claims that are European. However, the only methods we will consider in this dissertation are Hulley and McWalter's (2015) quadratic approach and Monoyios's (2004) exponential-utility based approach. Other approaches not considered include Zhang *et al.* (2017). The quadratic approach assumes that the writer of the option behaves according to a quadratic-utility function and relies on a minimal martingale and variance-optimal measure to derive the local-risk and mean-variance optimal strategy, respectively. These two strategies represent the two different attitudes to risk under this approach. Moreover, if one assumes particular market dynamics, these two measures coincide, and we can obtain numerically tractable solutions for these two strategies based on the familiar Black and Scholes (1973) formula. Monoyios's (2004) approach assumes that the writer of the option behaves according to an exponential-utility function and also obtains a tractable solution for finding prices and hedging strategies, using a distortion method based on perturbation expansions. However, for both of these strategies to be numerically tractable, strong assumptions are made about the dynamics of the underlying asset, and only a limited number of option payoffs can be priced and hedged. The Deep Hedging framework deals with these shortcomings by obtaining prices and hedging strategies in a model agnostic way.

1.3 Deep Hedging and Neural Networks

Deep Hedging was first proposed by, Buehler *et al.* (2019a) and it obtains optimal prices and hedging strategies using neural networks. The "optimality" of these prices and hedging strategies are based on the particular objective trying to be achieved. For example, one such goal could be to produce a price and set of hedging strategies such that the variance of the hedging error at maturity is minimised. Another aim could be given a price, produce a set of hedging strategies such that some coherent risk metric (like expected shortfall) is minimised at maturity. Recurrent neural networks are used over vanilla neural networks to determine the optimal hedging strategy; because recurrent neural networks allow for better predictions on sequential data (like a time series of share prices). Additionally, recurrent neural networks have a "memory" of the values at previous time steps making predictions more accurate. Unfortunately, standard recurrent neural networks

struggle to make predictions on long data sequences, so a more sophisticated type of recurrent neural network — Long Short-Term Memory networks — are adopted in this dissertation. Since effective hedging requires regular rebalancing, the ability to make accurate predictions on long sequences is crucial. Deep Hedging computes hedging strategies in a model agnostic way, so the Greeks have to be calculated at no stage. Consequently, the writer of the contingent claim can focus on modelling realistic market dynamics with more exotic option payoffs instead of deriving solutions for optimal hedging strategies under specific market assumptions. Additionally, since the Deep Hedging framework learns optimal hedging strategies using recurrent neural networks, we can easily make our hedging strategy a function of various trading signals and previous hedging decisions. Therefore, Deep Hedging creates an optimal system similar to how an actual trader would behave.

1.4 Limitations

Since the computational power available for this dissertation was limited, many simplifying assumptions had to be made. The most significant of these simplifying assumptions was to fix the volatility and correlation parameters during testing and training. This reduced the dimensionality of the Deep Hedging problem, and made the neural networks easier to train. Another simplification made was to work under the forward measure, as this allows one to ignore the impact of interest rates. This not only improved computational performance, but also made the notation used in the Deep Hedging framework compact. The last major limitation was the fact that this dissertation does not tackle pricing in the Deep Hedging framework, and only focussed on hedging.

1.5 Structure of Dissertation

The dissertation is divided into five chapters. The second chapter is a review of existing literature. The first half of the second chapter provides a high-level overview of two existing strategies to hedge basis risk. This overview is not rigorous, and the reader is directed to the source material for more details. The second half of the second chapter are the machine learning preliminaries. Here we provide and introduction to artificial neural networks, reinforcement learning and long short-term memory networks. An experienced reader can skim through this section. In Chapter 3 we introduce the Deep Hedging framework, and describe how it can be used to solve complete and incomplete hedging problems. In this chapter, we also describe the neural network architecture, as well as the data generation pro-

cess. Chapter 4 compares the results of two Deep Hedging strategies that use mean square error and expected shortfall as loss measures, to optimal strategies in complete and incomplete market settings. Here we find that the Deep Hedging strategy that uses mean square error as a loss measure, is a good approximation to the optimal complete and incomplete market strategies. Furthermore, when Deep Hedging uses expected shortfall as a loss measure, it outperforms McWalter's variance optimal strategy for hedging basis risk. The final chapter summarises the findings of this dissertation and discusses the applications of Deep Hedging to risk management. This dissertation should be accessible to a graduate student with a background in statistics and mathematical finance.

Chapter 2

Literature Review

2.1 Brief Overview of Existing Hedging Approaches

In the presence of basis risk, we cannot trade in the underlying, leading to an incomplete market. Furthermore, since we have an incomplete market, there is no unique martingale measure nor single preference-dependent price for contingent claims. To obtain a unique price for the contingent claim in question where standard arbitrage arguments fail, Davis (1997) states that the pricing (and subsequent hedging) decisions should be reformulated as a utility maxmisation problem. A benefit of using this approach is that basis risk will be minimized taking the investor's attitude to risk into account. This idea of reformulating pricing and hedging decisions as a utility maximisation problem is the foundation of the two existing approaches that are considered in this dissertation. The first approach is the exponential utility based approach initially proposed by Davis (2006) (a modification of Davis (2000)), with refinement and numerical computation done by Monoyios (2004, 2010), using a distortion method by Zariphopoulou (2001) based on perturbation expansions. The second approach that is considered is the quadratic approach of McWalter (2006); Hulley and McWalter (2015). This approach also considers the investor's level of risk aversion. Instead of assuming that an investor follows an exponential utility function, they assume that the investor behaves according to a quadratic utility function. Furthermore, they define a hedging cost process and two different risk processes, which are quadratic functions of the hedging cost process. By minimising these two respective risk processes, we get two different strategies for minimising basis risk.

2.2 Problem Set Up

Fix $[0, T]$ a finite time horizon and a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$ where $(\mathcal{F}_t)_{t\geq 0}$ is a filtration satisfying the usual conditions. Let W^1 and W^2 be

adapted independent Brownian motions. Define a bank account process B with dynamics $dB_t = rB_t$ where r is the risk-free rate of interest. Let Y be the underlying asset for the contingent claim, where the claim is specified as $h(Y_T)$. T is the maturity time for the claim and $h : \mathbb{R}_+ \to \mathbb{R}_+$ a Borel measurable function. Since we can only observe the price of Y but not trade it, let S be a tradeable proxy asset. The theory on which the optimal solution under the Quadratic approach is derived is based on discounted assets, so let $\bar{Y}_t = \frac{Y_t}{B_t}$ and $P_t = \frac{S_t}{B_t}$.
$Y_t, S_t, \bar{Y}_t$ and X_t can be modelled as geometric Brownian motions as follows:

$$\frac{dY_t}{Y_t} = \mu_y dt + \sigma_y dW_t^y, \tag{2.1}$$
$$\frac{dS_t}{S_t} = \mu_s dt + \sigma_s dW_t^1, \tag{2.2}$$
$$\frac{d\bar{Y}_t}{\bar{Y}_t} = (\mu_y - r)dt + \sigma_u dW_t^y, \tag{2.3}$$
$$\frac{dP_t}{P_t} = (\mu_s - r)dt + \sigma_s dW_t^1, \tag{2.4}$$

where $dW_t^u = \rho\, dW_t^1 + \sqrt{1-\rho^2}\, dW_t^2$, $\mu_s, \mu_y \in \mathbb{R}$ and $\sigma_s, \sigma_y \in \mathbb{R}_+$. Let the discounted European claim be $\bar{h}(\bar{Y}_T)$ where $\bar{h} : \mathbb{R}_+ \to \mathbb{R}_+$ is defined by $\bar{h}(P) = e^{-rT}h(e^{rT}P)$. The value that we wish to hedge is $\bar{h}(\bar{Y}_T)$. It is important to note that in this dissertation we are working under the forward measure, so the impact interest rates can be ignored.

2.3 Special Case: Perfect Correlation

Solving the stochastic differential equations in (2.1) and (2.2), we obtain,

$$\begin{aligned} Y_t &= Y_0 \exp\left[\left(\mu_y - \frac{1}{2}\sigma_y^2\right)t + \sigma_y W_t^y\right], \\ S_t &= S_0 \exp\left[\left(\mu_s - \frac{1}{2}\sigma_s^2\right)t + \sigma_s W_t^1\right]. \end{aligned} \tag{2.5}$$

If $\rho = 1$, we see that $W_t^y = W_t^1$. Therefore

$$Y_t = Y_0 \exp\left[\left(\mu_y - \frac{1}{2}\sigma_y^2\right) + \sigma_y W_t^1\right]. \tag{2.6}$$

Isolating W_t^1 in (2.5) and substituting into (2.6) we obtain

$$Y_t = Y_0 e^{\kappa t}\left(\frac{S_t}{S_0}\right)^{\frac{\sigma_y}{\sigma_s}}, \tag{2.7}$$

where $\kappa = \mu_y - \frac{\sigma_y}{\sigma_s}\mu_s + \frac{1}{2}\sigma_y\left(\sigma_s - \sigma_y\right)$. Davis (2000) then notes that due to the relation in (2.7), Y_t and S_t are both tradeable assets, and must therefore grow at the risk-free

rate under the risk neutral measure. This implies that Y_t and S_t have the same market price of risk

$$\frac{\mu_y - r}{\sigma_y} = \frac{\mu_s - r}{\sigma_s}.$$

Moreover, the perfect correlation makes it possible to perfectly delta hedge any call or put option, where Y_t is the underlying by simply holding

$$\frac{\partial \mathsf{BS}}{\partial y}(Y_t, t)\frac{\sigma_y Y_t}{\sigma_s S_t} \tag{2.8}$$

many shares of S_t, where $\mathsf{BS}(Y_t, t)$ is the Black-Scholes value of the option at time t. Expression (2.8) forms the basis for a naive strategy of hedging basis risk. However, Hulley and McWalter (2015) improve on the strategy shown in (2.8) by using a Capital Asset Pricing Model relation that incorporates the correlation between the proxy and underlying assets:

$$\frac{\partial \mathsf{BS}}{\partial y}(Y_t, t)\frac{\sigma_y Y_t}{\sigma_s S_t}\rho.$$

2.4 Exponential Utility-Based Approach

2.4.1 The Investor's Optimisation Problem

The investor's risk appetite can be described using a negative exponential as the utility function:

$$U(x) = -\exp(-\gamma x), \tag{2.9}$$

where $0 < \gamma < 1$ represents a constant level of risk aversion. For the rest of this dissertation, U will refer to (2.9). The investor is trying to maximise their expected utility of wealth at the maturity time of the option. This maximisation is achieved by trading a self-financing portfolio, X_t that consists of δ_t of the proxy asset at time $0 \leq t \leq T$, with the remainder invested in the bank account, which grows at the risk-free rate. Additionally, the investor incurs a liability equal to n units of $h(Y_T)$ at the option's maturity. Therefore, the investor's portfolio process is as follows

$$\begin{aligned} dX_t &= r\left(X_t - \delta_t S_t\right)dt + \delta_t dS_t \\ &= rX_t dt + \delta_t S_t\left((\mu_s - r)dt + \sigma_s dW_t^1\right), \\ &= rX_t dt + \pi_t\left((\mu_s - r)dt + \sigma_s dW_t^1\right), \end{aligned} \tag{2.10}$$

where $\pi_t := \delta_t S_t$, $0 \leq t \leq T$ is the amount of the investor's portfolio that is invested in the share. Monoyios (2004) notes that since (2.10) has no explicit dependence on S_t we can describe the market dynamics using (2.1) and (2.10).

Monoyios (2004) describes the investor's optimisation problem as follows: given an initial endowment, $X_t = x$, and share price for the underlying, $Y_t = y$, at time $0 \leq t \leq T$, the investor must find a trading strategy that $\pi := (\pi_t)_{0 \leq t \leq T} \in \mathcal{P}_0$ that achieves the following supremum

$$F^n(t,x,y) := \sup_{\pi \in \mathcal{P}_0} \mathbb{E}_{t,x,y} U\left(X_T + nh(Y_T)\right), \tag{2.11}$$

where $\mathbb{E}_{t,x,y}$ denotes the $\mathbb{P}$-expectation conditional on $X_t = x, Y_t = y$ and $\mathcal{P}_0$ is the class of all trading strategies. The n in equation (2.11) represents the number of option payoffs at maturity, with $n = 0$ and $n = -1$ being the most important cases. Note that since we are dealing with the exponential utility function, the optimisation problem in (2.11) is only meaningful if the option payoff, $nh(Y_T)$, is bounded from below (Davis (2006)). This means that under the current utility function we are unable to hedge short calls. For a similar specification of the investor's optimisation problem, see Davis (1997, 2000, 2006).

2.4.2 Trading Strategies

A trading strategy is defined as a progressively measurable process $(\pi_t)_{0 \leq t \leq T}$ satisfying

$$\int_0^T \pi_t^2 dt < \infty \quad \text{a.s.}$$

Davis (2006) notes that in order to preclude the existence of doubling strategies, restrictions have to be placed on $\mathcal{P}_0$

$$\begin{aligned}
\mathcal{P}_1 &= \{\pi \in \mathcal{P}_0 : X_t \geq c \in \mathbb{R} \text{ a.s.} \quad \forall t \in [0,T]\}, \\
\mathcal{U}^n &= \{U\left(X_T + nh(Y_T)\right) : \pi \in \mathcal{P}_1\}^c, \\
\mathcal{P} &= \{\pi \in \mathcal{P}_0 : U\left(X_T + nh(Y_T)\right) \in \mathcal{U}^n\},
\end{aligned}$$

where $\{\ldots\}^c$ denotes the closure in $L_1(\Omega, \mathcal{F}_T, \mathbb{P})$. Davis then raises the point that $\mathcal{P}_1$ is not large enough, as there are certain cases where in order to find the optimal trading strategy it is necessary for wealth not to be bounded from below. Furthermore, by enlarging the class of trading strategies from $\mathcal{P}_1$ to $\mathcal{P}$, we include some trading strategies for which wealth is not necessarily bounded from below (Monoyios (2004)).

2.4.3 Pricing and Hedging

Looking at (2.11), it is seen that there are two cases to consider. The first case is where $n = 0$ and the utility maximisation problem depends on the underlying

asset Y_t. We define $F(t,x) := F^0(t,x,y)$ to make this fact explicit, and let $\pi^0 = (\pi^0_t)_{0\leq t\leq T}$ be the optimal trading strategy that achieves the supremum in (2.11). The second case is where $n = -1$. This corresponds to the investor receiving $p(t,x,y)$ immediately in exchange for incurring liability of $h(Y_T)$ at maturity. By definition, the utility indifference ask price, $p^a(t,x,y)$ of the liability is the solution to

$$F(t,x) = F^{-1}(t, x + p^a(t,x,y), y).$$

At the indifference ask price, the investor derives the same expected utility from selling the option and receiving $p^a(t,x,y)$ versus doing nothing (Henderson and Hobson (2008)). Similarly, let the optimal trading strategy associated with $F^{-1}(t, x+p^a(t,x,y), y)$ be $\pi^{-1} = (\pi^{-1}_t)_{0\leq t\leq T}$. Given the fact that π^{-1} and π^0 represent the optimal strategy when the liability $h(Y_T)$ is and is not incurred respectively, Monoyios (2004) defines $\pi^h := \pi^{-1} - \pi^0$ as the trading strategy associated with the additional holding that must be taken when the liability is incurred at the asking price $p^a(t,x,y)$. This trading strategy reduces to the Black-Scholes delta-hedge in a complete market.

2.4.4 The Minimal Martingale Measure

Let $\mathcal{M}$ be the set of all $\mathbb{P}$-equivalent local martingale measures under which the discounted price process for the traded asset is a local martingale. Then the traded and underlying assets have the following price dynamics under $\mathbb{Q} \in \mathcal{M}$

$$\frac{dS_t}{S_t} = rdt + \sigma_s d\tilde{W}^1_t,$$
$$\frac{dY_t}{Y_t} = (\mu_y - \sigma_y(\rho\lambda + \epsilon g_t))dt + \sigma_y d\tilde{W}^y_t,$$

where $\lambda := \frac{\mu_s - r}{\sigma_s}$, g_t is a trading strategy $\mathbb{P}$-a.s, and

$$\tilde{W}^1_t := W^1_t + \lambda t,$$
$$\tilde{W}^2_t := W^2_t + \int_0^t g_s ds.$$

Note that $\tilde{W}^1_t$ and $\tilde{W}^2_t$ are independent $\mathbb{Q}$-Brownian motions and $\tilde{W}^y_t := \rho\tilde{W}^1_t + \epsilon\tilde{W}^2_t$, where $\epsilon = \sqrt{1-\rho^2}$. Additionally, the set $\mathcal{M}$ has a one-to-one correspondence with the set of processes g_t. The above leads to the following definition by Monoyios (2004)

Definition 2.1. The *minimal martingale measure* $\mathbb{Q}^0 \in \mathcal{M}$ corresponds to $g_t = 0, t \in [0,T]$.

2.4.5 Optimal Hedging Strategy

Monoyios' solution to the optimisation problem in (2.11) is based on the Hamilton-Jacobi-Bellman equation (HJB) equation of dynamic programming (Kalman (1963)). Using the HJB equation, we see that the value function $F^n(t,x,y)$ satisfies the following PDE

$$F_t^n(t,x,y) + rxF_x^n(t,x,y) + \mu_y y F_y^n(t,x,y) + \frac{1}{2}\sigma_y^2 y^2 F_{yy}^n(t,x,y) - \frac{\left[\lambda F_x^n(t,x,y) + \rho\sigma_y y F_{xy}^n(t,x,y)\right]^2}{2F_{xy}^n(t,x,y)} = 0,$$

with boundary condition at maturity being $F^n(T,x,y) = -e^{\gamma(x+nh(y))}$. Since we have an exponential utility function and using the distortion method of Zariphopoulou (2001), Monoyios obtains the following solution to the value function in (2.11) as

$$F^n(t,x,y) = -e^{\gamma\beta(t,T)x}\left[\mathbb{E}_{t,y}^0\left(e^{-\alpha(T-t)-\gamma\epsilon^2 nh(Y_T)}\right)\right]^{1/\epsilon^2}, \tag{2.12}$$

where $\beta(t,T) := e^{r(T-t)}$, $\alpha = \frac{1}{2}\lambda^2\epsilon^2 = \frac{1}{2}\left(\frac{\mu_s - r}{\sigma_s}\right)(1-\rho)^2$, $\mathbb{E}_{t,y}^0$ denotes the expectation with respect to the minimal martingale measure $\mathbb{Q}^0$, conditional on $Y_t = y$.

2.4.6 Closed-Form Solution

Theorems 1 and 2 in Monoyios (2004) give expressions for the utility indifference asking price and hedging strategy. However, by using the power series expansions in Theorem 3 and Corollary 1, Monoyios gets the following approximations for the utility indifference price ($p^a(t,y)$) and delta ($p_y^a(t,y)$) respectively

$$\begin{aligned} p^a(t,y) = \exp -r(T-t)\Big[&M_1 + \frac{1}{2!}\gamma\epsilon^2\left(M_2 - M_1^2\right) \\ &+ \frac{1}{3!}\gamma^2\epsilon^4\left(M_3 - 3M_1M_2 + 2M_1^3\right) \\ &+ \frac{1}{4!}\gamma^3\epsilon^6\left(M_4 - 3M_2^2 + 12M_1^2M_2 - 4M_1M_3 - 6M_1^4\right) + \mathrm{O}\left(\epsilon^8\right)\Big] \end{aligned} \tag{2.13}$$

and

$$\begin{aligned}\frac{\partial}{\partial y}p^a(t,y) = \exp(-r(T-t))\Big[&\partial M_1 + \frac{1}{2}\gamma\epsilon^2(\partial M_2 - 2M_1\partial M_1)\\ &+ \frac{1}{3!}\gamma^2\epsilon^4(\partial M_3 - 3M_2\partial M_1 - 3M_1\partial M_2 + 6M_1^2\partial M_1)\\ &+ \frac{1}{4!}\gamma^3\epsilon^6(\partial M_4 - 6M_2\partial M_2 + 12M_1^2\partial M_2 + 24M_1M_2\partial M_1\\ &- 4M_1\partial M_3 - 4M_3\partial M_1 - 24M_1^3\partial M_1) + \mathrm{O}\left(\epsilon^8\right)\Big] \qquad (2.14)\end{aligned}$$

where

$$\begin{aligned}M_1 &= K\Phi(-d_1+s) - ye^{(r-q)(T-t)}\Phi(-d_1),\\ M_2 &= K^2\Phi(-d_1+s) - 2Kye^{(r-q)(T-t)}\Phi(-d_1) + y^2e^{(2(r-q)+\sigma_y^2)(T-t)}\Phi(-d_1-s),\\ M_3 &= K^3\Phi(-d_1+s) - 3K^2ye^{(r-q)(T-t)}\Phi(-d_1) + 3Ky^2e^{(2(r-q)+\sigma_y^2)(T-t)}\Phi(-d_1-s)\\ &\quad - y^3e^{3(r-q+\sigma_y^2)(T-t)}\Phi(-d_1-2s),\\ M_4 &= K^4\Phi(-d_1+s) - 4K^3ye^{(r-q)(T-t)}\Phi(-d_1) + 6K^2y^2e^{(2(r-q)+\sigma_y^2)(T-t)}\Phi(-d_1-s)\\ &\quad - 4Ky^3e^{3(r-q+\sigma_y^2)(T-t)}\Phi(-d_1-2s) + y^4e^{2(2(r-q)+3\sigma_y^2)(T-t)}\Phi(-d_1-3s),\\ \partial M_1 &= -e^{(r-q)(T-t)}\Phi(-d_1),\\ \partial M_2 &= -2e^{(r-q)(T-t)}[K\Phi(-d_1) - ye^{(r-q+\sigma_y^2)(T-t)}\Phi(-d_1-s)],\\ \partial M_3 &= -3e^{(r-q)(T-t)}[K^2\Phi(-d_1) - 2Kye^{(r-q+\sigma_y^2)(T-t)}\Phi(-d_1-s)\\ &\quad + y^2e^{(2(r-q)+3\sigma_y^2)(T-t)}\Phi(-d_1-2s)],\\ \partial M_4 &= -4e^{(r-q)(T-t)}[K^3N(-d_1) - 3K^2ye^{(r-q+\sigma_y^2)(T-t)}\Phi(-d_1-s)\\ &\quad + 3Ky^2e^{(2(r-q)+3\sigma_y^2)(T-t)}\Phi(-d_1-2s) - y^3e^{3(r-q+2\sigma_y^2)(T-t)}\Phi(-d_1-3s)]\\ d_1 &= \frac{\log\frac{y}{K} + \left(r-q+\frac{\sigma_y^2}{2}\right)(T-t)}{s},\\ q &= r-(\mu_y - \sigma_y\rho\lambda),\\ s &= \sigma_y\sqrt{T-t},\end{aligned}$$

and $\Phi(.)$ is the cumulative distribution function of a standard normal random variable.

2.5 Quadratic Hedging Approach

Here we outline the derivation for the local-risk minimising and mean-variance optimal strategies, respectively. Only the salient points are covered with the reader

is directed to McWalter (2006) for the derivations in full detail, which is a summary of Föllmer and Sondermann (1985); Schweizer (1988, 1990); Föllmer and Schweizer (1991); Schweizer (1992, 1994); Heath *et al.* (2001).

2.5.1 Mathematical Preliminaries

Again, let $[0, T]$ be a finite time horizon and fix a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$ where $(\mathcal{F}_t)_{t\geq 0}$ is a filtration satisfying the usual conditions. Assume the existence of discounted stock price process P, a bank account process B and an $\mathcal{F}_T$ measurable contingent claim H, maturing at time T. P is a square-integrable martingale, and H is the quantity that we wish to hedge, which is assumed to have the property that $H \in L^2(\Omega, \mathcal{F}, \mathbb{P},)$. By the Galtchouk–Kunita–Watanabe (GKW hereafter) decomposition theorem (Proposition 2.75 of McWalter (2006)), H has the following unique representation:

$$H = H_0 + \int_0^T \xi_s^H dP_s + L_T^H, \tag{2.15}$$

where $H_0 = \mathbb{E}[H]$, $L^H \in \mathcal{M}_0^2$ is strongly orthogonal to P and $\int_0^{\cdot}(\xi_s^H)^2 \, d\langle \mathrm{P}\rangle_s$ is integrable. Concerning basis risk, this means that we can decompose a contingent claim H into a bank account component, a stock component and an unhedgeable component denoted by L^H

Definition 2.2. The *intrinsic value* of a contingent claim H is

$$\mathbb{E}[H|\mathcal{F}_t] = H_0 + \int_0^t \xi_s^H dP_s + L_t^H,$$

where ξ^H and L^H are defined the same way as in (2.15). The intrinsic value is the best estimate of H given current information.

Definition 2.3. A strategy (ξ, η) is called *feasible* if $\int_0^{\cdot}(\xi_s)^2 \, d\langle \mathrm{P}\rangle_s$ is integrable, the hedge portfolio $V(\xi, \eta)$ is continuous[1] with $V_t(\xi, \eta) \in L^2(\Omega, \mathcal{F}, \mathbb{P},) \; \forall t \in [0, T]$, η is the bank account and $V_T(\xi, \eta) = H$ a.s.

.

Definition 2.4. The *cost process* $C(\xi, \eta)$ associated with a feasible strategy (ξ, η) is defined by

$$C_t(\xi, \eta) := V_t(\xi, \eta) - G_t(\xi), \quad \forall t \in [0, T],$$

where $G_t(\xi) = \int_0^t \xi_s dP_s$ is the gains from trading the risky asset.

[1] McWalter considered general semimartingales, while we will only consider geometric Brownian motion.

We follow the notation iteMc06 in Definition 2.3, but $C_t(\xi, \eta)$ is a hedging cost process and unrelated to transaction costs. The cost process also gives us a way to model how the intrinsic value deviates from the portfolio value. Moreover, the cost process helps account for the changing hedging cost due to the hedging portfolio not replicating the claim exactly. Note that this hedging cost is constant in complete markets and is equal to the risk-neutral price of the claim. Moreover, using a feasible strategy and the cost process in Definition 2.3, we have

$$H = V_T(\xi, \eta) = C_T(\xi, \eta) + G_T(\xi).$$

Where ξ is chosen to minimise the risk associated with the cost process in some optimal way, the above shows that we can write the contingent claim as the sum of the hedging cost and gains from trading at maturity. With Definition 2.3, we can now define the risk process as follows:

Definition 2.5. The *risk process* $R(\xi, \eta)$ associated with a feasible strategy (η, ξ) is defined by

$$R_t(\xi, \eta) := \mathbb{E}[(C_T(\xi, \eta) - C_t(\xi, \eta))^2 | \mathcal{F}_t], \quad \forall t \in [0, T].$$

Now, when hedging a contingent claim, in the light of unhedgeable risk L^H, we have two choices. We can either continually add or subtract money to ensure that the hedge portfolio agrees with the intrinsic value or make any corrections at maturity. The latter is self-financing, while McWalter defines the former as mean self-financing. The self-financing choice is a variance-optimal strategy, and it minimises the unhedgeable component at maturity. Equivalently, this self-financing strategy minimises $R_0(\xi, \eta)$. The mean self-financing choice is local risk-minimising and minimises the unhedgeable component at every time. Equivalently, it minimises $R_t(\xi, \eta)$ $\forall t$.

2.5.2 Optimal Hedging Strategy

The previous section aimed to provide some intuition for how this approach works while omitting most technical details. We are now ready to tackle the basis risk problem. Note that this section follows Chapter 7 McWalter (2006) closely.

Definition 2.6. A *semimartingale* is a process P of the form,

$$P = P_0 + M + A \tag{2.16}$$

Where P_0 is finite and $\mathcal{F}_0$-measurable, $M \in \mathcal{M}_{0,\text{loc}}$ and A is a finite variation process null at zero.

Definition 2.7. Suppose P is a semimartingale with a finite variation component that is previsible. Then (2.16) is called the *canonical decomposition* of P

Definition 2.8. Let P be a semimartingale. P satisfies the *structure condition* if its canonical decomposition takes the form

$$P_t = P_0 + M_t + \int_0^t \alpha_s d\langle M\rangle_s$$

$\forall\, t \in [0,T]$, where $M \in \mathcal{M}^2_{0,\text{loc}}$ and $\int_0^{\cdot}(\alpha_s)^2\, d\langle P\rangle_s$ is locally integrable

Since our tradeable proxy asset P satisfies the structure condition, it has a canonical decomposition that takes the form.

$$P_t = P_0 + M_t + \int_0^t \alpha_s d\langle M\rangle_s,$$

with

$$M_t := \int_0^t \sigma P_s dW_s^1 \quad \text{and} \quad \alpha_t := \frac{\mu_s - r}{\sigma_s^2 P_t}.$$

Now we need to find the Föllmer-Schweizer (FS hereafter) Decomposition of the discounted claim, which can be written as

$$\bar{h}(\bar{Y}_T) := H = H_0 + \int_0^T \xi_s^H\, dP_s + L_T^H,$$

where ξ^H has the property that $\int_0^{\cdot}(\xi_s^H)^2\, dP$ is integrable and $L^H \in \mathcal{M}_0^2$ strongly orthogonal to M. Once we have the FS-decomposition, we can easily obtain the price and hedging parameters by inspection. Note that under the minimal martingale measure, the GKW and FS-decomposition coincide (Choulli *et al.* (2010)). We can then use α_t and M_t to construct a minimal martingale measure for P. After that, we use the Feynman-Kač Theorem to obtain a PDE representation of the claim. Consequently, we get the following expression for the discounted European claim:

$$\bar{h}(\bar{Y}_T) = F(0,\tilde{Y}_0) + \int_0^T \frac{\rho\sigma_y \tilde{Y}_l}{\sigma_s P_l}\frac{\partial F}{\partial p}(l,\tilde{Y}_l)dX_l + \int_0^T \sigma_y \tilde{Y}_l\sqrt{1-\rho^2}\frac{\partial F}{\partial p}(l,\tilde{Y}_l)dW_l^2,$$

where

$$F : [0,T]\times(0,\infty) \to \mathbb{R}_+ \quad \text{with} \quad F(t,x) := \mathbb{E}^{\hat{\mathbb{P}}}[\bar{h}(\bar{Y}_T)|\tilde{Y}_t = p],$$

$\hat{\mathbb{P}}$ is the minimal martingale measure and

$$\tilde{Y}_t = e^{-\kappa(T-t)}\bar{Y}_t \quad \text{where} \quad \kappa = \sigma_y\left(\rho\cdot\frac{\mu_s - r}{\sigma_s} - \frac{\mu_y - r}{\sigma_y}\right).$$

We can then compare the two equations for $\bar{h}(\tilde{Y}_T)$ and obtain the following FS-decomposition for the discounted claim:

$$H_0 = F(0, \tilde{Y}_0) = \mathbb{E}^{\hat{P}}\left[\bar{h}(\tilde{Y}_T)\right], \tag{2.17}$$

$$\xi_t^H = \frac{\rho\sigma_y \tilde{Y}_t}{\sigma_s P_t}\frac{\partial F}{\partial p}(t, \tilde{Y}_t), \tag{2.18}$$

$$L_t^H = \int_0^t \sigma_y \tilde{Y}_l \sqrt{1-\rho^2}\frac{\partial F}{\partial p}(l, \tilde{Y}_l) dW_l^2. \tag{2.19}$$

2.5.3 Closed-Form Solution

We can now use (2.17), (2.18) and (2.19) to obtain the local-risk minimising and mean-variance optimal strategy. As a result, we have the following proposition taken from Hulley and McWalter (2015). Note that in the following proposition, they use the Black-Scholes formula with continuous dividends, with κ being substituted in place of the dividend parameter, q.

Proposition 2.9. *Under the market assumptions of Section 2.2, the approximation price of the claim is given by*

$$v = V(0, Y_0) = \mathsf{BS}(t = 0, s = Y_0, q = \kappa, \sigma = \sigma_y).$$

The local risk-minimising strategy is given by

$$\hat{\xi}_t = \rho\frac{\sigma_y Y_t}{\sigma_s S_t}\frac{\partial V}{\partial s}(t, Y_t) = \rho\frac{\sigma_y Y_t}{\sigma_s S_t}\frac{\partial \mathsf{BS}}{\partial s}(t, Y_t, \kappa, \sigma_y), \tag{2.20}$$

and the mean-variance optimal strategy is given, in the feedback form, by

$$\begin{aligned}\tilde{\xi}_t &= \hat{\xi}_t + \frac{\mu_s - r}{\sigma_s^2 e^{-rt} S_t}\left(\hat{V}_t - v - G_t(\tilde{\xi})\right) \\ &= \hat{\xi}_t + \frac{\mu_s - r}{\sigma_s^2 e^{-rt} S_t}\left(\hat{V}_t - v - \int_0^t \tilde{\xi}_l d(e^{-rl} S_l)\right),\end{aligned} \tag{2.21}$$

with the intrinsic value given by

$$\hat{V}_t = e^{-rt} V(t, Y_t) = e^{-rt}\mathsf{BS}(t, Y_t, \kappa, \sigma_y), \tag{2.22}$$

for all $t \in [0, T]$.

2.6 Machine Learning Preliminaries

2.6.1 Introduction

Machine learning is a field that focuses on developing algorithms that improve with increasing amounts of data (Staudemeyer and Morris (2019)). This field can be

divided into three different areas: supervised learning, unsupervised learning and reinforcement learning. Supervised learning occurs when an algorithm is given a set of labelled data — input and target pairs — and using the input, the algorithm must produce an output that best matches the targets, according to some criterion. In unsupervised learning, an algorithm is given unlabelled data, and instead of matching some targets, the algorithm is trying to extract key features of the data set. Finally, reinforcement learning involves an agent interacting with an environment, trying to find a collection of optimal decisions or actions that maximises some reward (Sutton and Barto (2018)). Although reinforcement learning algorithms are the focus of this dissertation, an understanding of supervised learning techniques is necessary to understand how reinforcement learning algorithms work. This is because when one is trying to approximate optimal strategies in continuous state space (as is done in this dissertation), one needs a tool to approximate the value function (which is related to the rewards that the agent is trying to maximise). Supervised learning algorithms, specifically neural networks, are adequately suited to approximate the value function due to their universal function approximation properties as described in Hornik (1991).

2.6.2 Supervised Learning

Supervised learning can be divided into two sections: classification and regression. Classification is where the targets are a discrete set of categories, and in regression, targets are real numbers. An example of the former is classifying handwritten digits. An example of the latter is estimating the temperature of a city based on some environmental conditions. The task of finding optimal trading strategies and prices is a supervised regression problem. Any supervised regression algorithm can be broken up into two parts: prediction and training. Prediction is how the algorithm generates output based on its input, and training is how the algorithm uses this output and the targets to improve the parameters of the algorithm. As we shall see, prediction is nothing more than matrix multiplication, while training is nothing more than elementary calculus.

Artificial Neural Networks

Linear regression is the most straightforward supervised regression algorithm. Unfortunately, it assumes that the inputs are linearly related to the targets (Murphy (2012)). Artificial neural networks (ANNs) rectify this limitation by introducing non-linear activation functions, which model more complex relationships between the inputs and targets. The activation function mimics the behaviour of the neuron

in the brain (Higham and Higham (2019)) and, based on the value of the input, will determine if that particular input is active or inactive. The activity level determines how much those inputs contribute to the output and hence the training of the network. An ANN with two hidden layers makes predictions according to the following relation.

$$H^{(1)} = f^{(1)}(W^{(m_1,m_0)}X_{\text{train}} + b^{(1)}), \tag{2.23}$$

$$H^{(2)} = f^{(2)}(W^{(m_2,m_1)}H^{(1)} + b^{(2)}), \tag{2.24}$$

$$\text{Output} = f^{(3)}(W^{(m_3,m_2)}H^{(2)} + b^{(3)}), \tag{2.25}$$

where X_{train} is a matrix of inputs and $f^{(i)}$, $W^{(*)}$, $b^{(i)}$ are the activation functions, weight matrices and bias vectors at each layer. Note that the activation functions are identical throughout the neural network and X_{train} is also referred to as the training data. $H^{(1)}$ and $H^{(2)}$ are the outputs of the first and second hidden layers, respectively. An ANN with two or more hidden layers is regarded as a deep neural network. Common choices for activation functions include $\sigma(z) = \frac{1}{1+e^{-z}}$, $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ and $\text{ReLU}(z) = \max(z, 0)$. Looking at the sigmoid function, σ, we can get a better understanding of how the activation function operates on the input, as very large positive inputs to the sigmoid activation function are approximately mapped to one, while very small negative inputs are approximately mapped to zero. The dimensions of the matrices in the model are as follows. X_{train} is a $m_0 \times n$ matrix where m_0 is the number of input features, and n is the number of samples. For each weight matrix, the dimension of the rows is the number of nodes at the current layer. In contrast, the dimension of the columns is the number of nodes at the previous layer, where m_1 and m_2 are the numbers of nodes in the first and second hidden layers, respectively, and m_3 is the dimension of the output. The bias vectors $b^{(1)}, b^{(2)}$ and $b^{(3)}$ are column vectors of length m_1, m_2 and m_3 respectively. For example, if $m_0 = 3, m_1 = 5, m_2 = 4$ and $m_3 = 1$ then (2.23), (2.24) and (2.25), and Fig. 2.1 are equivalent representations for the same process. Fig. 2.1 looks at how each sample gets propagated through the ANN. The grey lines going from the nodes green to blue, blue to blue and blue to red represent the matrix multiplication contribution to the output of the next layer. In contrast, yellow nodes represent the contribution of the bias. This provides a different representation of how the hidden and output layers are a function of previous layers. It also shows a fully connected neural network (every node in the next layer is connected to every node in the previous layer). We can interpret the weight matrices at each of the intermediary steps as a linear transformation, with the overall effect being the map Output: $\mathbb{R}^{(m_0 \times n)} \to \mathbb{R}^{(m_3 \times n)}$. Another interpretation is that the weight matrices and bias vectors shift the scale and location of the activation function (Higham and

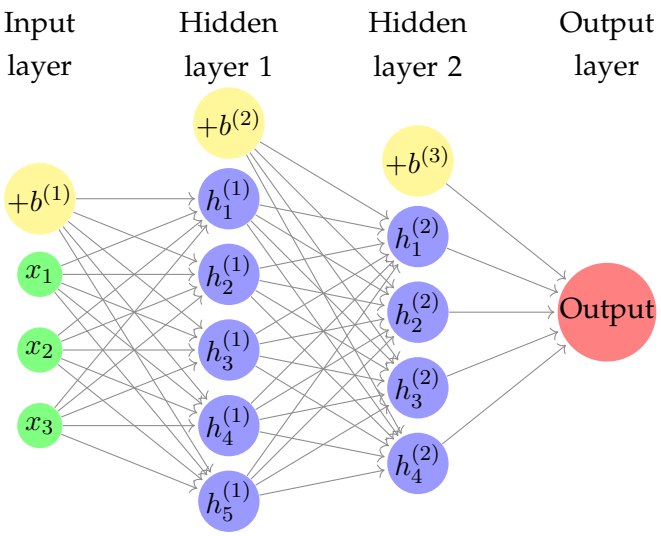

Fig. 2.1: Artificial Neural Network with Two Hidden Layers.

Higham (2019))

Now that we know how an ANN makes predictions, we can move on to how it improves those predictions through training. We can compare the output from the prediction step to the targets to determine how well our ANN is performing. We do this using a cost function.

$$\text{Cost}\left(W^{(1)}, W^{(2)}, W^{(3)}, b^{(1)}, b^{(2)}, b^{(3)}\right) = \frac{1}{n}\sum_{i=1}^{n} G\left(\text{Target}_i - \text{Output}_i\right), \tag{2.26}$$

where Target_i and Output_i represent the target and output value corresponding to the i^{th} sample. The function G represents the error between the target and the output. In supervised regression problems, a common choice for error function is the squared error function, $f(t, o) = (t - o)^2$. An important fact to note is that the weights and biases are the only variable parameters, so training this ANN is the same as updating these variable parameters until the cost function is minimized. Using notation from Higham and Higham (2019) we can write the inputs to (2.26) more compactly as p, where p is a vector containing all the weights and biases in the ANN. Given that we have a value for our cost function, Cost(p), how can we improve it? We answer this question using the gradient descent algorithm, which iteratively updates p until the cost function converges. Using a Taylor series expansion and ignoring higher order terms, the value of the cost function under a

small perturbation Δp is

$$\begin{aligned}\text{Cost}(p+\Delta p) &\approx \text{Cost}(p) + \sum_{k=1}^{m} \frac{\partial\, \text{Cost}(p)}{\partial p_k} \Delta p_k \\ &= \text{Cost}(p) + \nabla \text{Cost}(p)^T \Delta p, \end{aligned} \tag{2.27}$$

where m represents the cardinality of all the parameters in the neural network and $\nabla \text{Cost}(p)$ is the gradient, which is a vector of partial derivatives (Higham and Higham (2019)). Looking at (2.27), we see that in order to improve the cost function, we should step in the negative direction of the gradient. However, since (2.27) is just an approximation, this improvement is only valid for small perturbations, so the step we take, η should also be small. We now have a way to update the ANN parameters in order to improve the cost output

$$p \to p - \eta \nabla \text{Cost}(p),$$

where η is also known as the learning rate. The procedure for updating the neural network is known as gradient descent. The neural network parameters are updated until a certain number of passes through the training data (epochs) are concluded or sooner if some early stopping callback is implemented. To use this gradient descent, we require a way to calculate the gradient of the cost function. This is done through a technique called Backpropagation that was first described by Werbos (1974), and then popularised by Rumelhart *et al.* (1986). Backpropagation involves the repeated use of the chain rule to calculate the partial derivatives necessary to update the weights and biases of the network. The details of Backpropagation can be ignored as we will be using TensorFlow (Abadi *et al.* (2015)), an open-source machine learning library, to perform all machine learning computations. Now that we understand the fundamentals of neural networks, we can describe more sophisticated supervised regression algorithms.

Recurrent Neural Networks

From a theoretical standpoint, neural networks have the potential to be excellent function approximators, and the details of this will be formally described in the next section. However, the efficacy of the function approximation depends on the type of neural network used. In theory, we could use an ANN to approximate the optimal hedging strategy that spans a series of rebalancing dates. At each rebalancing date, we have an ANN similar to Fig 2.1, with the inputs being our market data and the output being the hedging strategy. However, if we have a reasonable number of rebalancing dates — 30 or more — this would introduce many parameters into the neural network, making it difficult to train.

Furthermore, approximating hedging strategies in this way assumes that all the inputs are independent of each other (Lipton *et al.* (2015)). This is fine if we approximate strategies for a Markov process like geometric Brownian motion but becomes unsuitable when this independence assumption no longer holds. Recurrent neural networks (RNNs) rectify these issues. They have a shared weight structure that significantly reduces the number of parameters needed in the model, and they can capture dependencies across time steps. Moreover, the outputs at each time step are a function of the input at the current time step and the output at the previous time step. This makes RNNs better suited for modelling the sequential data (paths of our price processes) we will be working with when trying to approximate optimal hedging strategies when compared to ANNs. The notation of Lipton *et al.* (2015) the output at each time step is the following.

$$h_t = \tanh(W^{\mathrm{h,x}} x_t + W^{\mathrm{h,h}} h_{t-1} + b_h), \tag{2.28}$$

where $W^{\mathrm{h,x}}$, $W^{\mathrm{h,h}}$ and b_h are identical across all time steps, which is why RNNs are shared weight models. Based on the activation function chosen, the output at each time step will be a value between -1 and 1. Since we are trying to approximate hedging strategies, these are reasonable values for the hedging values to take. Expression (2.28) is the simplest form a recurrent neural network can take. As we shall see, there is a superior RNN structure. To get the general representation of an RNN, we will call the connections, weight matrices, bias vector, and activation function collectively A so that Fig. 2.2 becomes an equivalent representation for (2.28). In Fig. 2.2. A is also called the memory cell. Training RNNs are very similar to training ANNs and the algorithm that achieves this is called Backpropagation through time (BPTT) and was first described by Werbos (1990). Unfortunately, vanilla RNNs suffer from vanishing and exploding gradients and not being able to learn long sequence dependencies (for more details, see Staudemeyer and Morris (2019) or Lipton *et al.* (2015)), which can lead to sub-optimal training. Therefore a superior memory cell needs to be introduced.

Long Short-Term Memory Networks

Long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber (1997)) are a type of RNN that overcomes the short-term memory issues associated with vanilla RNNs. LSTMs achieve this by using a memory cell with a number of gates that regulate the flow of information throughout the neural network, leading to improved memory retention. However, this increase in memory retention comes with an increased computational cost due to the higher number of parameters. These gates are similar to the neural network layers we have previously seen. Olah's

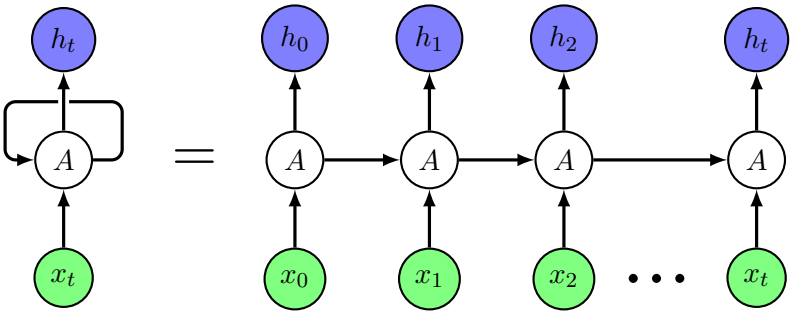

Fig. 2.2: Unrolled Recurrent Neural Network Structure.

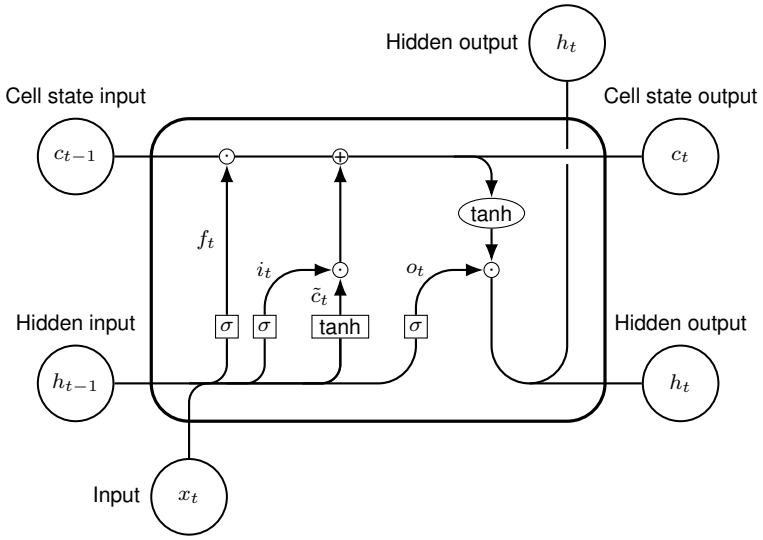

Fig. 2.3: Long Short-Term Memory Cell.

(2015) representation for the a LSTMs memory cell is found in Fig. 2.3. Looking at the internal components of the memory cell, note that

- Square or rectangular blocks represent layers of weights, biases and activation functions.
- Circular or oval nodes represent element-wise operations.
- Arrows represent a vector transfer.
- Lines joining together represent vector concatenation.
- Lines splitting apart represent vector duplication, with these different copies going to different parts of the network.

The most crucial part of an LSTM is the cell state, c_t, the horizontal line at the top of Fig. 2.3 (Olah (2015)). The cell state acts as a highway for information to be

shared across multiple time steps, with the flow of information being regulated by various gates. Each of these gates comprises of a sigmoid layer and an elementwise multiplication node. Since the sigmoid layer maps all inputs to values in $[0, 1]$, coupled with the elementwise multiplication, values can either be retained — corresponding to a sigmoid output approximately equal to one — or discarded — corresponding to a sigmoid output approximately equal to zero.

The first gate is called the forget gate. It uses the current input, x_t, and the hidden output from the previous layer, h_t, to determine how much information in the cell state gets discarded. This gate is labelled f_t in Fig. 2.2. and is calculated as follows

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right),$$

where $[\cdot]$ represents concatenation. The second gate is called the input gate, i_t. This gate determines what values in the cell state get updated. Coupled with a $\tanh$ neural network layer that determines new possible values for the cell state, $\tilde{c}_t$, we can obtain values for our current cell state. This can be described using the following equations

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x] + b_i), \\ \tilde{c}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \end{aligned} \tag{2.29}$$

where $\odot$ represents elementwise multiplication. Note that since i_t and f_t use sigmoid activation functions, all values in these gates are between zero and one. Therefore, the current cell is a combination of the previous cell state values that have been retained, as well as potential current cell state values that have been selected to be updated. Finally, the LSTM needs to determine what the new hidden output values, h_t are, and the output gate, o_t, helps us achieve this. The output gate is another sigmoid neural network layer, so the final hidden output is just a filtered version of the transformed cell state. Recall that the $\tanh$ function maps inputs to values in $[-1, 1]$. This can be described using the equations below

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\ h_t &= o_t \odot \tanh(c_t). \end{aligned}$$

2.6.3 Reinforcement Learning

Reformulating option hedging as a reinforcement learning problem, we see that the agent is the option trader. The environment $\mathcal{S}_t$ represents all observable market information at time t. We then define the state $s_t \in \mathcal{S}_t$ as all the relevant information

that the trader requires to perform hedging. In the case of basis risk, this information is the prices of the underlying and proxy assets, and the previous position, ξ_{t_i-1} in the proxy asset, i.e,

$$s_{t_i} := \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R} = \{(Y_{t_i}, S_{t_i}, \xi_{t_{i-1}}), i = 0, 1, \dots N\} \quad \text{where } \xi_{-1} := 0.$$

The condition $\xi_{-1} := 0$ is a liquidation constraint as our entire shareholdings are converted to cash at maturity. At each time t_i, this trader can perform an action, a_{t_i}. This set of actions, $\pi \in \mathcal{A}$, is called the policy in the reinforcement learning literature. All actions are assumed to be deterministic functions of the state, i.e., $a_t \equiv a^\pi_{t_i}(s_{t_i})$. These actions determine the number of shares traders hold in their replicating portfolio at initiation and subsequent rebalancing dates. The number of shares held in the replicating portfolio is defined as follows: $\xi^\pi_{t_i} = a^\pi_{t_i} + \xi^\pi_{t_{i-1}}$. The reward, $R^\pi_{t_i}$, is composed of positive and negative rewards. The positive rewards are the cash inflows from our replicating portfolio, while the negative rewards are cash outflows related to transaction costs and any liabilities incurred. This can be stated more formally as

$$R^\pi_{t_i} = \xi^\pi_{t_{i-1}} \cdot (S_{t_i} - S_{t_{i-1}}) - c \cdot |S_{t_i} \cdot (\xi^\pi_{t_i} - \xi^\pi_{t_{i-1}})|,$$

where c is the proportional transaction cost. Note that in the above formulation, interest rates have been ignored. This because we are working under the forward measure, leading to a simple expression for the rewards. In the special case of t_0 and t_N, the reward becomes $R^\pi_{t_0} = p_0$ and $R^\pi_{t_N} = -H_T$, where p_0 is the option price and $-H_T$ is the value of option liability at maturity. The total future reward is then given by

$$G^\pi_{t_i} = R^\pi_{t_i+1} + R^\pi_{t_i+2} + \ldots + R^\pi_{t_N}.$$

The value function, $v^\pi_{t_i}$, is simply the expected total future reward, and the goal of traders is to find a policy π^* that maximises their expected total future rewards. This expectation can be taken using a metric that minimises the variance of the rewards, like mean square error. Another approach could be to use a coherent risk metric, like expected shortfall, that considers the trader's risk appetite. The Deep Hedging approach solves a deep reinforcement learning problem. This means that Deep Hedging uses a (deep) supervised learning algorithm to perform reinforcement learning. Specifically, the supervised learning algorithm approximates the value function. Furthermore, gradient descent (ascent in this case) allows us to maximise the expected total reward by iterating through all policies that can be specified by a neural network. In the Deep Hedging case, the final (optimal) policy that leads to the maximum expected reward is the holdings, ξ_{t_i}, that traders will use to hedge their derivatives.

2.7 Brief Overview of Reinforcement Learning Techniques Applied to Hedging

We end this chapter by briefly surveying papers that implement reinforcement learning in the context of hedging derivatives and how they relate to the Deep Hedging framework implemented in Buehler *et al.* (2019a) or Buehler *et al.* (2019b). The first development is the QLBS algorithm. The theory and rationale behind the algorithm were outlined in Halperin (2020), and further numerical considerations were discussed in Halperin (2019). The QLBS algorithm uses Q-Learning, a reinforcement learning algorithm, to extend the dynamic option replication strategy described in the seminal Black and Scholes (1973) and Merton (1974) (BSM) papers to the scenario where the assumption of continuous rebalancing is ignored. The Q-Learning algorithm is described in detail in Watkins and Dayan (1992). Halperin's rationale for the QLBS algorithm is as follows.

If all the assumptions in the BSM model were true, derivatives would be redundant. This is because we would be able to perfectly replicate the payoff of the derivative by holding a portfolio of the underlying stock and cash in a bank account. This replicating portfolio is unique and instantaneously risk-free. Consequently, this portfolio would be independent of an individual investor's risk preferences. However, many of the assumptions in the BSM model are invalid in reality, which means that the individual investor's risk preference matters, as the replicating portfolio is not risk-free. Although there are many sources that introduce risk into the replicating portfolio, only the risk caused by being unable to continuously rebalance the replicating portfolio (mis-hedge risk) is considered in Halperin (2020). This is in contrast to Buehler *et al.* (2019a) that considers hedging in incomplete markets in general, and multiple sources of risk. Additionally, Halperin states that since the writer of the option can only rebalance their replicating portfolio at discrete times, their aim should be to minimise the risk-adjusted cost of hedging the option (slippage risk) by dynamic option replication. Over the lifetime of the derivative, the writer of the option is trying to find the set of sequential actions that minimise the slippage risk, and as we have seen in the previous section, this is a reinforcement learning problem. Halperin proceeds to use Q-Learning, a model-free reinforcement learning algorithm, to find the optimal risk-adjusted price and hedging strategy. However, even though Q-Learning is a model-free algorithm, it is not clear that the QLBS algorithm is model-free since its formulation relies heavily on the BSM model. The argument on how to expand the QLBS algorithm to a portfolio that contains multiple options is in Halperin (2017). Deep Hedging can also handle a portfolio of mutliple options, with a formulation that is less delicate

than the QLBS solution, with computational time that increases (mostly) linearly with an increase in portfolio size.

It is easier to see methods for hedging outlined in Buehler *et al.* (2019b) and Cao *et al.* (2021) as model-free as they both are completely agnostic to the underlying dynamics, and show this by creating optimal policies in a variety of conditions, which include stochastic volatility and in the presence of transaction costs. The main difference between the two approaches are the optimisation objectives and the reward structure. In Buehler *et al.* (2019b) the objective is to find the hedging strategy that minimises a convex risk measure like expected shortfall, while Cao *et al.* (2021) minimise a function that incorporates the average hedging cost and standard deviation of the hedging cost scaled by a constant. The reward structure under deep hedging is follows

$$(R_{t_i}^{\pi})^{DH} = \xi_{t_{i-1}}^{\pi} \cdot (S_{t_i} - S_{t_{i-1}}) - c \cdot |S_{t_i} \cdot (\xi_{t_i}^{\pi} - \xi_{t_{i-1}}^{\pi})|,$$

where the notation is identical to the notation introduced in Section 2.6.3. In the case of Cao *et al.* (2021), $(R_{t_i}^{\pi})^{DH}$ is also used as a reward structure, but they also use another structure,

$$(R_{t_i}^{\pi})^{C} = V_{t_i} - V_{t_{i-1}} + (R_{t_i}^{\pi})^{DH},$$

where V is the value of the derivative position. The structure $(R_{t_i}^{\pi})^{DH}$ is called the Cash Flow formulation, while $(R_{t_i}^{\pi})^{C}$ is called the Accounting P&L formulation. Although Cao *et al.* (2021) report improved hedging performance using $(R_{t_i}^{\pi})^{C}$, it requires a pricing model to calculate the V at each time point. However, the determination of the pricing model is an implicit part of the learning process if one uses $(R_{t_i}^{\pi})^{DH}$ as a reward structure. Therefore, this increase in performance can be attributed to the Accounting P&L formulation having more information than the Cash Flow formulation. Furthermore, in order to keep the learning process truly model-free, $(R_{t_i}^{\pi})^{DH}$ is preferred, and is the reward formulation used in this dissertation.

Chapter 3

Deep Hedging Approach

3.1 Framework

As we have seen, under the quadratic and exponential-utility based approaches to hedging basis risk, optimal strategies assume specific market dynamics and use model parameters when deriving these strategies. This can be problematic if the chosen model is too simple to reflect the current market situation or incorrect model parameters are used. Deep Hedging — first introduced in Buehler *et al.* (2019a) — overcomes this issue by approximating the optimal hedging strategies using neural networks. Since these neural networks are model agnostic, the optimal hedging strategies for more complex and realistic market dynamics can be approximated, which includes incomplete markets. This gives us another way to hedge basis risk, along with other market frictions like stochastic volatility or transaction costs. However, in this dissertation a number of simplifications have been made. Although pricing and hedging are closely linked, we only focus the hedging problem. This means that we take the option premium as a given. Another simplification made is the use of the forward measure, so that interest rates can be ignored. One of the benefits of working under the forward measure is that the notation becomes compact, and the calculation of the profit and loss is more computationally efficient due to vectorisation.[1] Further, the use of the forward measure leads to the implicit assumption of uncorrelated interest rates, which means that we don't have to assume constant interest rates. Lastly, the correlation and volatility parameters are fixed. This was done to reduce the dimensionality of the problem and make the neural networks easier to train. The use of neural networks is motivated by their universal approximation property, which was first described by Hornik (1991). Moreover, the results from Bolcskei *et al.* (2019) explain why neural networks are excellent at approximating hedging strategies. Quoting Buehler *et al.*

[1] This second point is essential for the Deep Hedging component of this dissertation as there was limited GPU capacity available for training.

(2019a), it is because

> *they quantify the minimum network connectivity needed to allow approximation of all elements in pre-specified classes of functions to within a prescribed error, which establishes a universal link between the connectivity of the approximating network and the complexity of the function class that is approximated.*

In Buehler *et al.* (2019a), two methods are used to obtain optimal hedging strategies. For the first method, the hedging strategy at each rebalancing time is encoded as a separate ANN, and in the second method, a vanilla RNN is used to approximate the optimal hedging strategy. Buehler *et al.* improved their previous architectures by using a stacked LSTM to obtain the optimal hedging strategies. The term "stacked" is used since the output from one LSTM is fed into another LSTM. This helps the neural network extract important features in the input, and a stacked LSTM is akin to a deep ANN with many hidden layers. Now that we can encode hedging strategies as neural networks, our modelling task is reduced to stipulating the market dynamics, tradeable assets, market frictions and a loss function. The loss function is not restricted to mean square error, but can be any convex risk measure. These risk measures allow us to capture a variety of risk preferences, like exponential utility or any other coherent risk metric. For more detail on pricing and hedging under convex risk measures see Buehler *et al.* (2019a). Neural networks are used, and not some other kind of approximating function, because the gradient descent has been extensively researched and optimised (Ruder (2016)), making it very efficient.

3.1.1 Hedging in Complete Markets

We take a short aside to explain hedging in a complete market. In a complete market where one can trade in the underlying asset, an option can be replicated precisely using a Delta-hedging strategy. This strategy ensures that the delta of the replicating portfolio, X, tracks the delta of the underlying option at every rebalancing date (Haugh (2016)). In this way, the writer of the option is shielded from changes in the underlying. In a perfect world, the writer could be completely protected from the changes in the underlying if the replicating portfolio is continuously rebalanced, without frictions. Since this is impractical, rebalancing is done on $N \in \mathbb{Z}^+$ rebalancing dates. The portfolio begins with an initial endowment of cash equal to the initial value of the contingent claim H_0 that we wish to hedge. Then on each of the following rebalancing dates, the portfolio is rebalanced in such a way that we are long ξ_{t_i} units of the stock S where ξ_{t_i} is the delta of the share at

time t_i, i.e.,

$$X_0 = H_0, \tag{3.1}$$

$$X_{t_i+1} = X_{t_i} + \xi_{t_i}\left(S_{t_{i+1}} - S_{t_i}\right), \tag{3.2}$$

$$\text{PnL} = X_T - H_T, \tag{3.3}$$

where $\Delta t = \frac{T}{N}$ and H_t is the value of the contingent claim at time t. The equations (3.1), (3.2) and (3.3) can be written succinctly as follows

$$\text{PnL} = H_0 + \sum_{i=0}^{N-1} \xi_{t_i}\left(S_{t_{i+1}} - S_{t_i}\right) - H_T. \tag{3.4}$$

3.1.2 Deep Hedging in Complete Markets

After introducing the Deep Hedging hedging framework, we are now able to explain what optimal strategies and prices are and how the Deep Hedging framework calculates them. As a first example, we calculate the optimal strategy for a complete market situation with geometric Brownian motion for the underlying dynamics. As we know, the optimal strategy should be the Delta strategy described in the previous section, with the optimal price being determined by the Black-Scholes formula. We can then compare these results to the Deep Hedging solution. The optimal strategy is the solution to the following optimisation problem

$$\min_{p_0 \in \mathbb{R}, \xi \in \Theta} \mathbb{E}\left[\zeta\left(p_0 + \int_0^T \xi_s dS_s - h(S_T)\right)\right], \tag{3.5}$$

where p_0 is the fair price that the writer charges at time $t = 0$, Θ contains all possible predictable strategies, $h(S_T)$ is the value contingent claim at maturity, H is a process that represents the writer's holdings in the underlying asset and ζ is some measure of loss. Since it is infeasible to optimise over this entire space (Cuchiero and Teichmann (2019)), we can specify a subset of this space $\tilde{\Theta}_t$ and $\tilde{\mathbb{R}}$, where $\tilde{\Theta}_t$ represents all predictable strategies that a neural network can represent for each time point at which the portfolio is rebalanced and $\tilde{\mathbb{R}}$ are all prices that can be represented using the neural network. Note that (3.5) is simplified when the price is determined by external factors (e.g. competitive considerations). The price is therefore fixed and removes a dimension of this optimisation problem. This is the case considered in this dissertation.

3.2 Deep Hedging Applied to Basis Risk

Once the efficacy of Deep Hedging applied to a complete market problem is determined, we can then use the Deep Hedging framework to solve the similar basis

risk optimisation problem

$$\min_{\xi\in\Theta} \mathbb{E}\Big(\zeta\Big[p_0 + \int_0^T \xi_s dS_s - h(Y_T)\Big]\Big), \tag{3.6}$$

where $h(Y_T)$ is the value of the contingent claim based on the underlying that is not available for trade and only observed. Since we are using S as a proxy to hedge Y, coupled with the fact that we want to replicate realistic dynamics, it is reasonable for ξ_t to be a function of current market information and the previous hedging decision.

$$\xi_t = g_t(S_t, Y_t, \xi_{t-1}),$$

where g_t is the neural network for time t. The problem in (3.6) can be represented as a deep reinforcement learning problem. The input data are the sample paths for Y_t and S_t, with the goal of the algorithm being to minimise the loss per sample (LPS)

$$\text{LPS} = \frac{\sum\limits_{\omega\in\Omega} \zeta\left(p_0 + \left(\sum_{i=1}^N \xi_{i-1}(\omega))(S_i(\omega) - S_{i-1}(\omega))\right) - h(Y_T)(\omega)\right)}{|\Omega|}, \tag{3.7}$$

where $|\Omega|$ represents the number of sample paths used to generate this particular loss value and N are the number of rebalancing dates. Subsets of the training data will be used to calculate the loss value. This method is called batch gradient descent and it is used to speed up training. In order for the results to be comparable to the existing strategies, the market dynamics described in Section 2.2 will be used for the path processes, which are as follows

$$Y_t = Y_{t-1} \exp\left[\left(\mu_y - \frac{\sigma_y^2}{2}\right)\frac{T}{N} + \sigma_y\sqrt{\frac{T}{N}}\left(\rho Z_i^1 + \sqrt{1-\rho^2}Z_i^2\right)\right], \tag{3.8}$$

$$S_t = S_{t-1} \exp\left[\left(\mu_s - \frac{\sigma_s^2}{2}\right)\frac{T}{N} + \sigma_s\sqrt{\frac{T}{N}}Z_i^1\right]. \tag{3.9}$$

3.2.1 Transaction Costs

Although we do not compute optimal trading strategies in presence of transaction costs in this dissertation, here is a brief interlude to discuss how it could be implemented. In the case of proportional transaction costs, the optimisation problem in (3.6) becomes

$$\min_{\xi\in\Theta} \mathbb{E}\Big[\zeta\Big(p_0 + \int_0^T \xi_s dS_s - h(Y_T) - \tilde{C}_T(H)\Big)\Big], \tag{3.10}$$

where $\tilde{C}_T(H)$ represents the total cost of trading at maturity. The loss per sample is now

$$\text{LPS} = \frac{\sum_{\omega\in\Omega} \zeta\left[p_0 + \left(\sum_{i=1}^{N} \xi_{i-1}(\omega))(S_i(\omega) - S_{i-1}(\omega))\right) - h(Y_T)(\omega) - \tilde{C}_T(H)(\omega)\right]}{|\Omega|}, \tag{3.11}$$

where

$$\tilde{C}_T(H)(\omega) = c\sum_{i=0}^{N} |S_i(\xi_i(\omega)) - \xi_{i-1}(\omega)| \quad c \in (0,1).$$

We adopt the notation of Buehler *et al.* (2019a) where $\xi_{-1} = 0$ and $\xi_N := 0$, where the latter implies that the writer's entire position is liquidated at maturity. For more detail on other market constraints that can be included in the Deep Hedging framework see Zhang and Huang (2021).

3.3 Neural Network Architecture

Figure 3.1 provides a high-level overview of the neural network architecture. Each type of node has a specific meaning.

- Green nodes represent inputs to the model,
- blue nodes represent neural network layers,
- yellow nodes represent intermediary outputs — which are inputs to other parts of the model,
- orange nodes represent operations
- and the red node is the final output of this model.

3.3.1 Data

There are three data sets used in this neural network: the training, validation and test set. The training set is the data set used to update the weights and biases of the neural network. The validation set is also used in training, but it only provides a stopping criterion for neural network training. At the end of each pass through data (an epoch), a loss value is computed according to (3.7) or (3.11) in the case of transaction costs. If the loss on the validation data does not improve after five consecutive epochs, or 50 epochs is reached — whichever comes first — then training is stopped. Due to increased variability in the basis risk test compared to the complete market test, these criteria will be adjusted to ten consecutive epochs or if 100

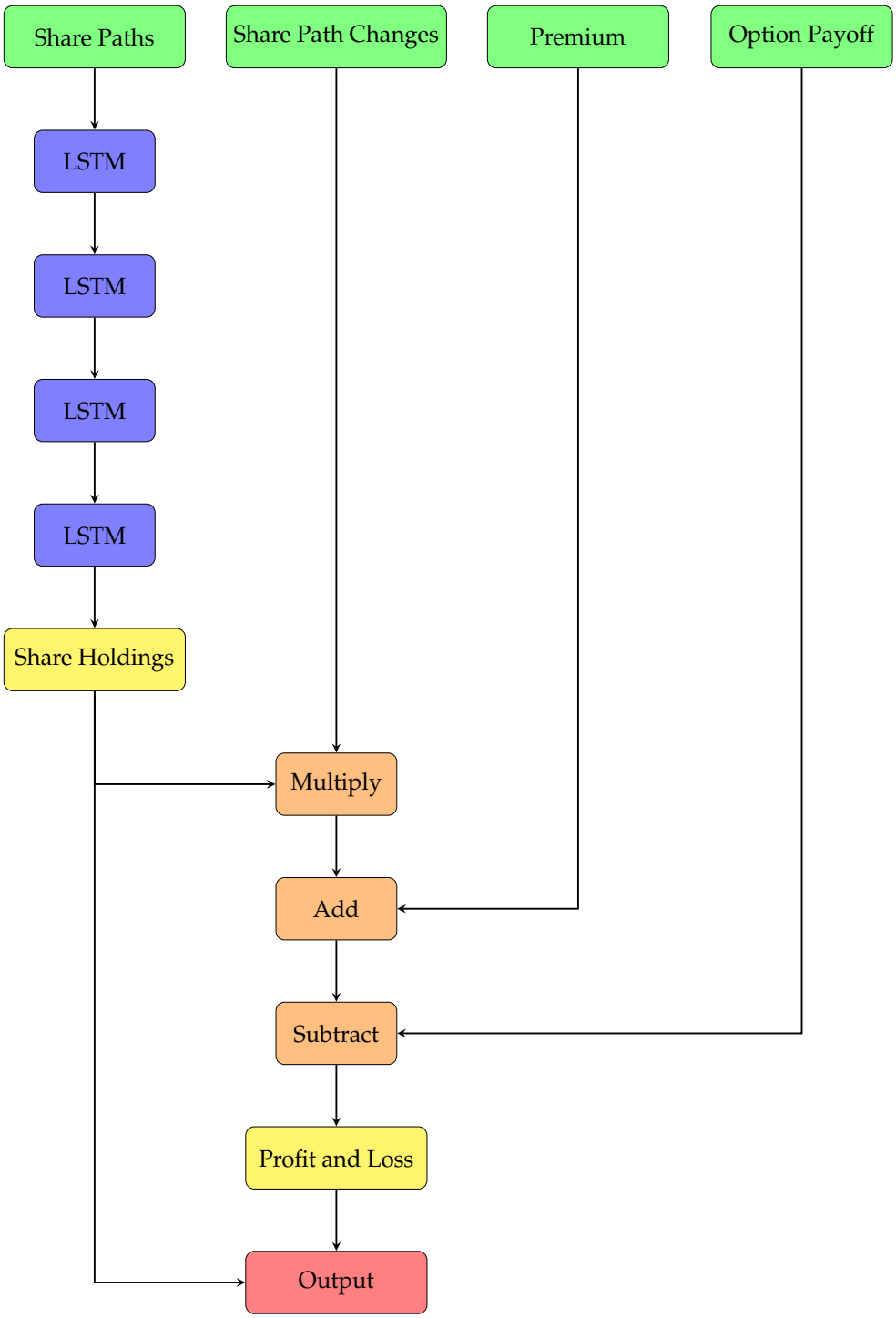

Fig. 3.1: Computational Graph of the Neural Network Architecture

epochs are reached. The benefit of having one data set to update the parameters and another to evaluate the performance of the neural network is that the neural network generalises well to data it has not seen before. The test set is entirely out of sample, and this is the data set that will be used to evaluate the performance compared to the existing strategies. All three data sets are three-dimensional matrices (tensors) with the exact dimensions: $(n, N+1, k)$. The first dimension is the number of samples, the second dimension is the number of rebalancing dates, and the last dimension is the number of features. The features are all available market data. These tensors can be generated by concatenating two three-dimensional tensors, with dynamics described by (3.8) and (3.9) and any other relevant market data. Note that since we want the neural network to perform well across a variety of moneyness levels, the strike of the option is included as one of the features.

3.3.2 Neural Network Layers

The neural network used in this dissertation is a stacked LSTM with $2N$ nodes in the first layer, $N+1$ nodes in the second layer, $N/2+1$ in the third layer and one node in the final layer. All activation functions used in the individual LSTMs are described in Section 2.6.4. This stacked LSTM is used to obtain the optimal shareholdings, akin to the "Deltas" in a complete market case.

3.3.3 Loss Measure

Two loss measures are used in this dissertation. The first on is mean square error, and leads to the interpretation that the hedging error at maturity is being minimised by neural network. The second loss measure used in this dissertation is expected shortfall as defined by Airouss *et al.* (2018). Expected shortfall leads to the interpretation that the neural network finds the shareholdings that minimise the average loss in the bottom $1-\alpha$ percentile of the profit and loss distribution, where $\alpha \in (0,1)$. The value α is known as the confidence level. Expected shortfall is also adopted in Buehler *et al.* (2019a), Buehler *et al.* (2019b) and Kim (2021). The optimiser used in this neural network is the Adam optimiser (Kingma and Ba (2015)), due to its performance and popularity within the deep learning literature.

3.4 Data Generation

3.4.1 Complete Market Test

For the underlying asset, an arbitrary training sample path is generated as follows

$$S_t = S_0 \prod_{i=1}^{t} \exp\left[\left(\mu_s - \frac{\sigma_s^2}{2}\right)\frac{T}{N} + \sigma_s\sqrt{\frac{T}{N}}Z_i^1\right], \tag{3.12}$$

where $t = 1, 2, \ldots, N$, and the parameters in Equation (3.12) are defined in Tab. 3.1. Note that for each sample path, S_0 and μ_s are independent samples from the uniform distribution as defined in Tab. 3.1, and Z_i^1 is a standard normal random variate. The training set is a tensor with shape $(n_{\text{train}}, N+1, 2)$. The parameter υ is the percentage of the training data that is used for validation. Recall that the second feature is the strike, $K := S_0 \times Moneyness$. Moneyness is also uniformly distributed. There are five test sets, one for each strike defined in Tab. 3.2. Each test set is a tensor with shape $(n_{\text{test}}, N+1, 2)$. Note for S_0 and K are fixed for each test set. The testing sample paths are also generated according to (3.12). The parameters σ_s, r, T and N are the same for testing and training. The option premium is determined by the Black-Scholes formula using the parameters in Tables 3.1 and 3.2.

S_0	Moneyness	μ_s	σ_s	r	T	N	n_{train}	υ
70-130	0.8-1.2	0.0-0.2	0.3	0	$\frac{30}{365}$	30	200000	0.2

Tab. 3.1: Complete Market Training Parameters.

S_0	K (DITM)	K (ITM)	K (ATM)	K (OTM)	K (DOTM)	μ_s	n_{test}
100	110	105	100	95	90	0.12	100000

Tab. 3.2: Complete Market Testing Parameters.

3.4.2 Incomplete Market Test

Similarly, for the underlying and proxy assets, an arbitrary training sample path is generated as follows

$$S_t = S_0 \prod_{i=1}^{t} \exp\left[\left(\mu_s - \frac{\sigma_s^2}{2}\right)\frac{T}{N} + \sigma_s\sqrt{\frac{T}{N}}Z_i^1\right], \tag{3.13}$$

$$Y_t = Y_0 \prod_{i=1}^{t} \exp\left[\left(\mu_y - \frac{\sigma_y^2}{2}\right)\frac{T}{N} + \sigma_y\sqrt{\frac{T}{N}}\left(\rho Z_i^1 + \sqrt{1-\rho^2}Z_i^2\right)\right], \tag{3.14}$$

where $t = 1, 2, \ldots, N$, and the parameters in Equations (3.13) and (3.14) are defined in Tab. 3.3. Similar to the complete market test, for each sample path, S_0 , Y_0 μ_s and μ_y are independent samples from the uniform distribution as defined in Tab. 3.3. Z_i^1 and Z_i^2 are independent standard normal random variates. The training set is a tensor with shape $(n_{\text{train}}, N + 1, 3)$, where the third feature K, is defined as before. The test sets are constructed in a similar way to the complete market test, with testing parameters being contained in Tab. 3.4. The parameters ρ, σ_s, σ_y r, T and N are the same for testing and training. The option premium used for all incomplete market tests is the approximation price defined in Proposition 2.9.

S_0	Y_0	Moneyness	μ_s	μ_y	ρ	σ_s	σ_y	r	T	N	n_{train}	υ
80-120	80-120	0.8-1.2	0.0-0.2	0.0-0.2	0.85	0.25	0.30	0	$\frac{30}{365}$	30	400000	0.3

Tab. 3.3: Incomplete Market Training Parameters.

S_0	K (DITM)	K (ITM)	K (ATM)	K (OTM)	K (DOTM)	μ_s	μ_y	n_{test}
100	110	105	100	95	90	0.10	0.12	100000

Tab. 3.4: Incomplete Market Testing Parameters.

Chapter 4

Discussion and Results

4.1 Deep Hedging compared to a Complete Market Solution

		Mean	Std	Median	Min	Max	Skew	Kurt	ES50
DITM	DH MS	0.0041	0.4415	0.0027	-4.7302	2.4542	-0.2666	3.7135	0.3070
	DH ES	0.0030	0.4451	0.0122	-4.5291	2.5186	-0.4204	4.3195	0.3052
	BS	-0.0006	0.4030	0.0142	-3.9732	2.2867	-0.7393	5.7173	0.2601
ITM	DH MS	0.0007	0.5180	0.0041	-3.6291	2.3841	-0.3204	2.2442	0.3757
	DH ES	0.0011	0.5216	0.0162	-3.7709	2.4397	-0.3871	2.5070	0.3737
	BS	-0.0020	0.5143	0.0115	-4.0517	2.3896	-0.4144	2.4266	0.3744
ATM	DH MS	0.0015	0.5436	0.0032	-3.7912	2.2360	-0.2310	1.4490	0.4087
	DH ES	0.0045	0.5490	0.0149	-3.9014	2.3160	-0.2875	1.6204	0.4068
	BS	-0.0002	0.5397	0.0084	-3.9487	2.2589	-0.2844	1.5905	0.4057
OTM	DH MS	0.0019	0.4562	0.0081	-3.6578	2.2673	-0.3220	2.4436	0.3287
	DH ES	0.0067	0.4603	0.0330	-3.5881	2.2285	-0.4760	2.6142	0.3239
	BS	0.0017	0.4532	0.0081	-3.8316	2.3036	-0.3427	2.5729	0.3260
DOTM	DH MS	-0.0000	0.2981	0.0057	-3.1425	1.8835	-0.7146	7.1325	0.1886
	DH ES	0.0039	0.3027	0.0348	-3.0423	1.8187	-1.3565	8.3343	0.1799
	BS	0.0004	0.2951	0.0091	-3.0800	1.9018	-0.7980	7.4618	0.1844

Tab. 4.1: Summary Statistics for the Complete Market Test.

As a first example we compare the performance of the Black-Scholes Delta-Hedging (BS) strategy to the Deep Hedge strategies, as the BS strategy is the benchmark to assess hedging performance in a complete market setting. Still, both the Deep Hedge strategies, using Mean Square Error and Expected Shortfall as loss measures (DH MS and DH ES), fair well against the BS strategy, although the BS strategy is still superior. Looking at the summary statistics in Tab. 4.1, the performance of the Deep Hedge strategies are similar to the BS strategies for strikes that

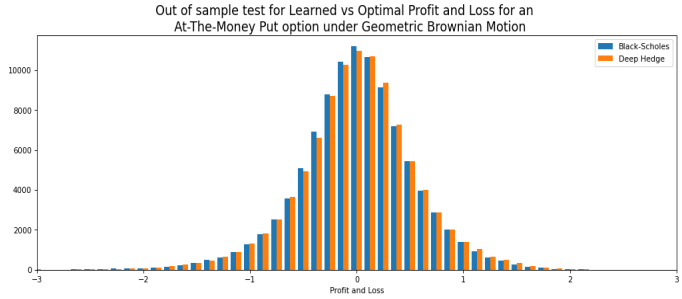

Fig. 4.1: PnL for an ATM Put Option in a Complete Market under Expected Shortfall.

are not deep in-the-money (DITM) or deep out-the-money (DOTM). For example, Fig. 4.1 and Fig. 4.4 show that the Deep Hedge replicates BS strategy for a strike that is at-the-money (ATM). Furthermore, looking at the Figures 4.2 and 4.3, even if the profit and loss distributions do not exactly match that of the BS strategy, the shareholdings or ”implied deltas” are very similar to Deltas of the BS strategy as seen in Figures 4.5 and 4.6. The difference between the DH MS and DH ES across all levels of moneyness is small. The results of the complete market test will not be discussed further as this is not the focus of this dissertation.

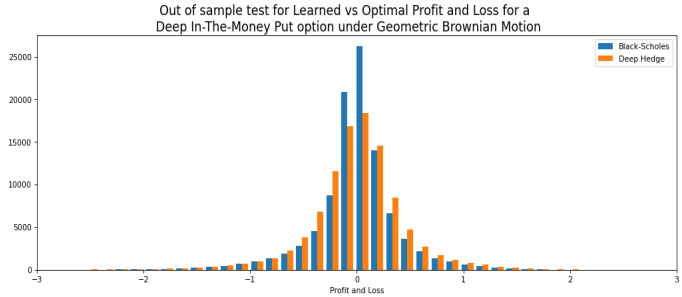

Fig. 4.2: PnL for a Deep ITM Put Option in a Complete Market under Expected Shortfall.

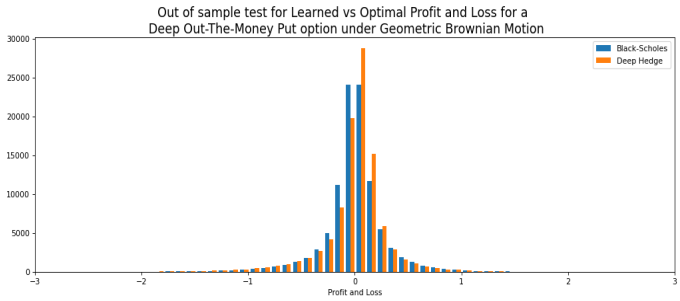

Fig. 4.3: PnL for a Deep OTM Put Option in a Complete Market under Expected Shortfall.

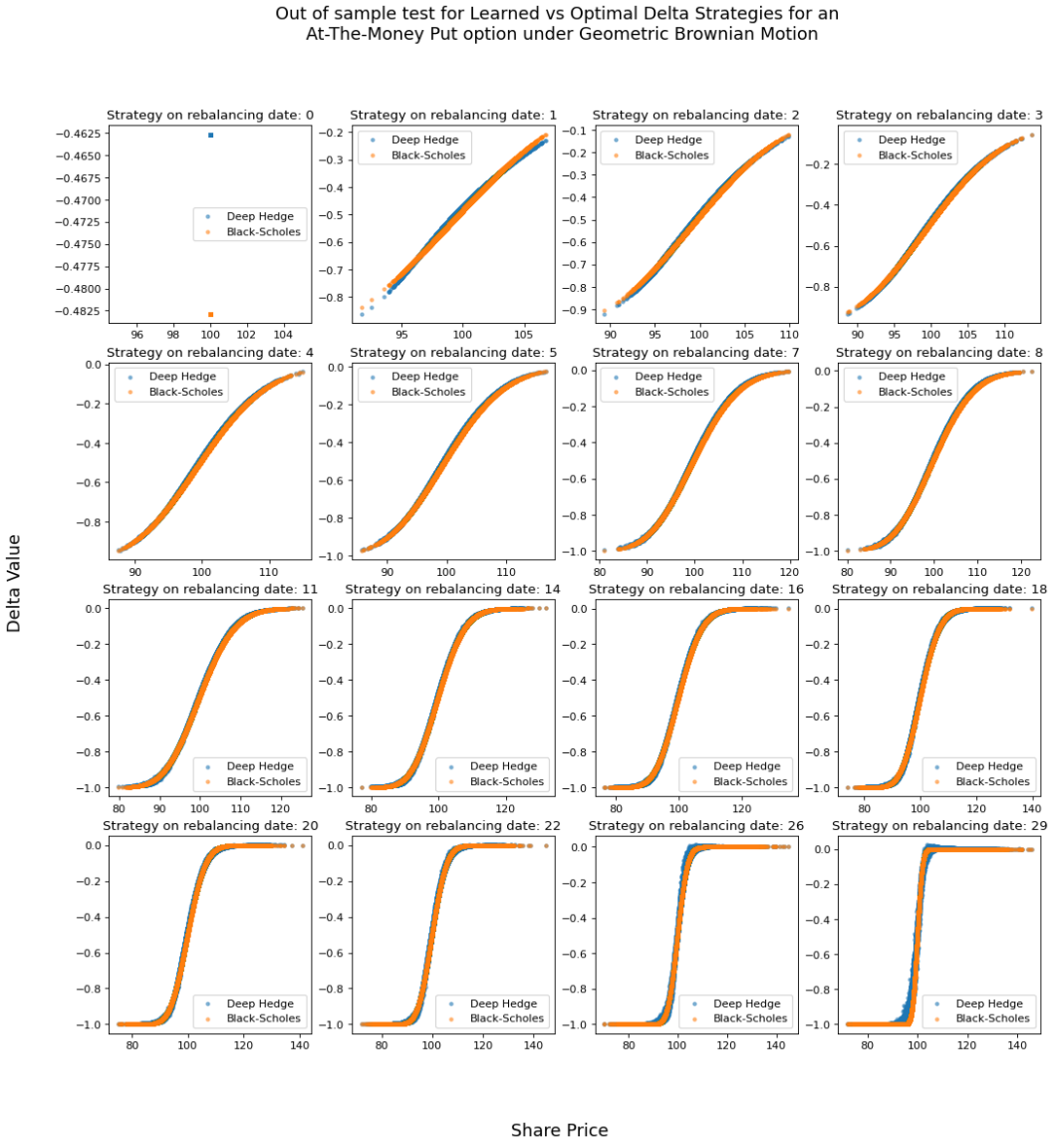

Fig. 4.4: Black-Scholes vs Deep Hedge Deltas for an ATM Put Option in a Complete Market under Expected Shortfall.

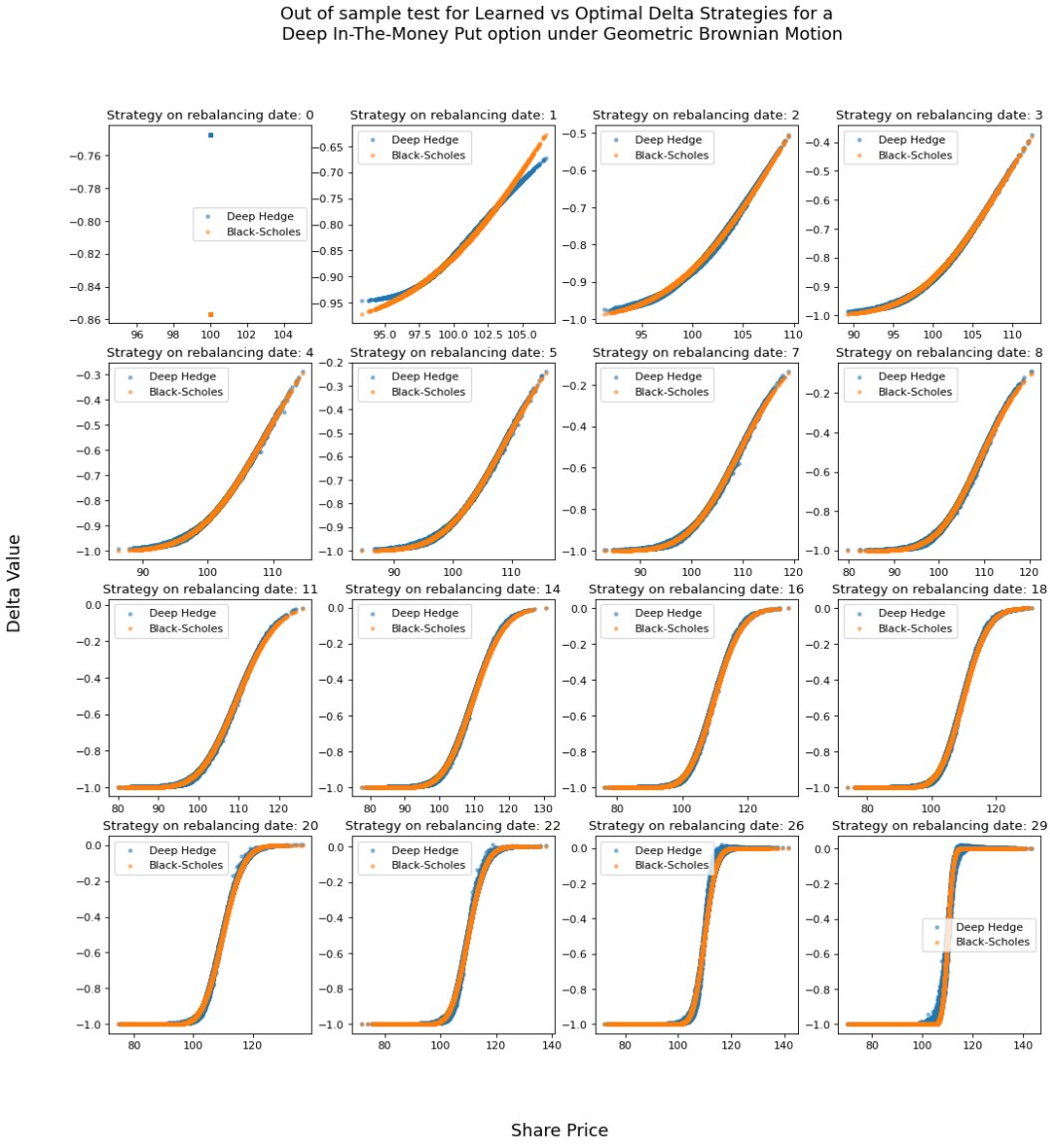

Fig. 4.5: Black-Scholes vs Deep Hedge Deltas for an ITM Put Option in a Complete Market under Expected Shortfall.

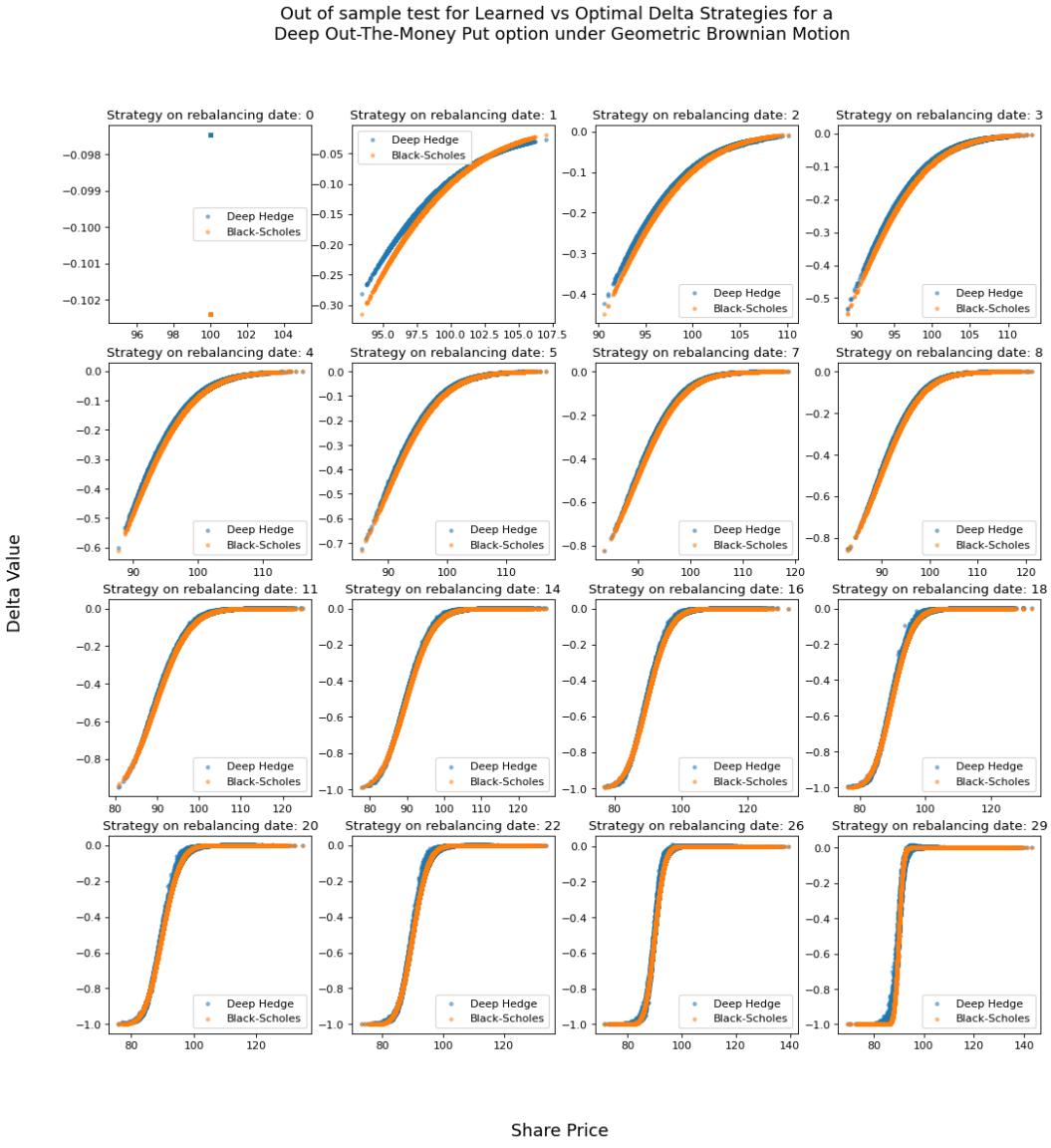

Fig. 4.6: Black-Scholes vs Deep Hedge Deltas for an OTM Put Option in a Complete Market under Expected Shortfall.

4.2 Deep Hedging compared to an Incomplete Market Solution

		Mean	Std	Median	Min	Max	Skew	Kurt	ES50
DITM	DH MS	0.0194	4.0961	0.0461	-17.3907	18.7046	-0.0615	0.0369	3.2396
	DH ES	0.1123	3.8293	-0.0491	-24.2511	19.1405	0.0089	0.6008	2.8853
	VO	0.0146	4.0756	-0.1178	-16.1607	21.6740	0.1933	0.2135	3.2051
ITM	DH MS	0.2664	3.4436	0.2517	-16.8993	16.6531	-0.0217	0.3173	2.4318
	DH ES	0.3227	3.3033	0.0592	-21.4988	16.8027	0.1240	1.1187	2.1825
	VO	0.2702	3.4203	0.1545	-15.3676	17.3001	0.1961	0.5376	2.3826
ATM	DH MS	0.4303	2.5022	0.3509	-14.1149	14.6271	0.0759	1.1565	1.4597
	DH ES	0.4631	2.4725	0.1569	-18.2783	16.8240	0.4308	2.6053	1.2625
	VO	0.4329	2.4963	0.3012	-13.2700	16.5495	0.2792	1.5431	1.4220
OTM	DH MS	0.3523	1.4793	0.2384	-10.2621	11.3441	0.2922	3.4124	0.6896
	DH ES	0.3707	1.4934	0.1778	-12.7890	13.0227	0.9446	7.5471	0.5093
	VO	0.3534	1.4797	0.2200	-9.8763	13.0938	0.5083	4.1699	0.6578
DOTM	DH MS	0.1708	0.6785	0.0905	-6.9774	8.6294	0.8826	11.6529	0.2570
	DH ES	0.1872	0.6702	0.1298	-9.0324	9.1312	1.7593	30.2846	0.1043
	VO	0.1734	0.6784	0.0871	-6.7074	10.1413	1.2925	14.7860	0.2215

Tab. 4.2: Summary Statistics for the Incomplete Market Test.

Looking at summary statistics for the various PnL distributions in Tab. 4.2, we see that the performance of the DH MS strategy is comparable to McWalter's Variance Optimal (VO) hedging strategy. Additionally, the difference between the two strategies is small, showing that if we Deep Hedge using a quadratic loss measure, it is good approximation for the VO strategy. The objective of DH MS strategy is to minimises the variance of hedging error at maturity. This objective has been achieved, as the standard deviation of the DH MS strategy is very close to the VO strategy, with the largest difference occurring with in-the-money (ITM) options, although this difference is small. Even though the variance of the two strategies is similar, the VO is still preferable due to its sligtly higher peaks. This can be confirmed visually by looking at Figures 4.7, 4.8, 4.9, 4.10 and 4.11, as well as the slightly higher kurtosis values for the VO strategy. Overall, the performance of the VO strategy is marginally better than the DH MS strategy, although the DH MS strategy is a good approximation.

When comparing the DH ES strategy to the VO and DH MS strategies, the DH ES strategy outperforms both of them. Even though the objective of the the DH ES strategy is to minimise the expected shortfall in the bottom 50% (ES50) of the PnL distribution, the DH ES strategy not only had the lowest ES50 values, it also

had a slightly better standard deviation on average to the other strategies, across all levels of moneyness. Moreover, the DH ES strategy has higher peaks and is more negatively skewed on in across all levels of moneyness. This can be confirmed by looking at the skewness and kurtosis values in Tab. 4.2. The fact that the DH ES strategy produces PnL distributions with lower values of skewness should is unsurprising. Both the VO and DH MS strategies are optimising quadratic measures, which means that they are indifferent to profits and losses. However, the optimisation regime of DH ES strategy is only based on losses, and ignores profits. Hence, the losses for the DH ES strategy should be lower on average when compared to the other two remaining strategies. This is indeed the case and can be confirmed visually by looking at Figures 4.12, 4.13, 4.14, 4.15 and 4.16. Even though the performance of the DH ES strategy is generally better than the other two strategies, the DH ES strategy is prone to large losses, especially with options that are DITM or DOTM.

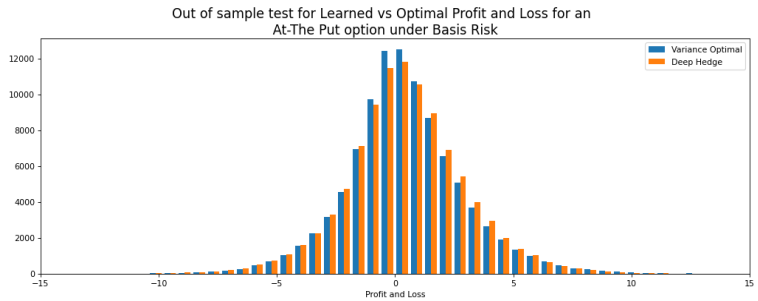

Fig. 4.7: PnL for an ATM Put Option in an Incomplete Market under Mean Square Error.

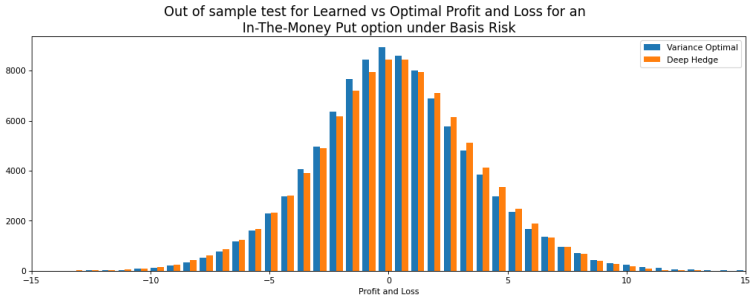

Fig. 4.8: PnL for an ITM Put Option in an Incomplete Market under Mean Square Error.

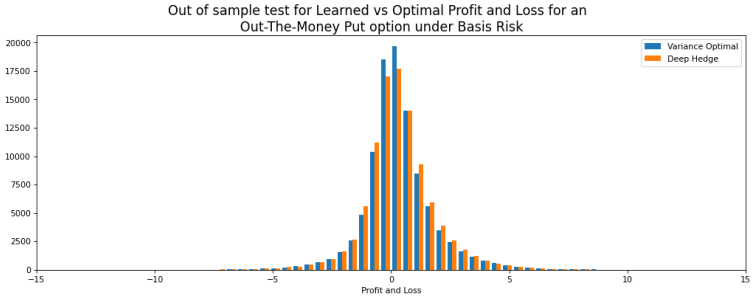

Fig. 4.9: PnL for an OTM Put Option in an Incomplete Market under Mean Square Error.

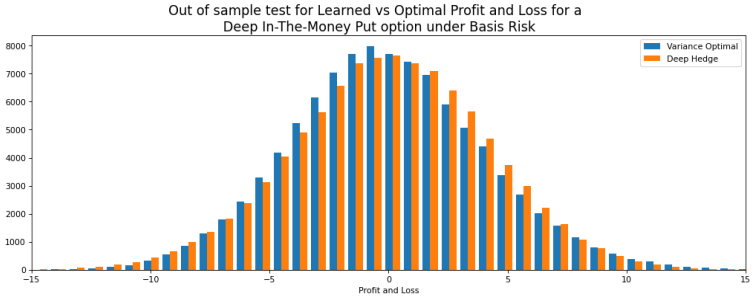

Fig. 4.10: PnL for a Deep ITM Put Option in an Incomplete Market under Mean Square Error.

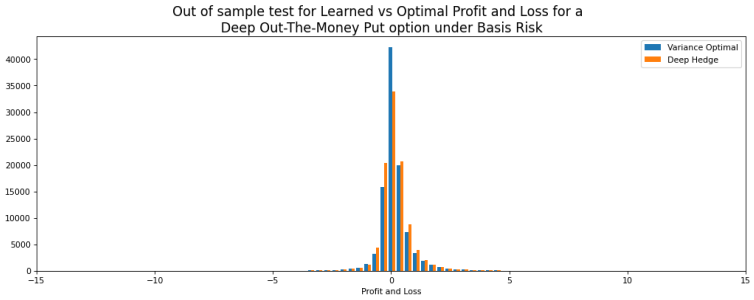

Fig. 4.11: PnL for a Deep OTM Put Option in an Incomplete Market under Mean Square Error.

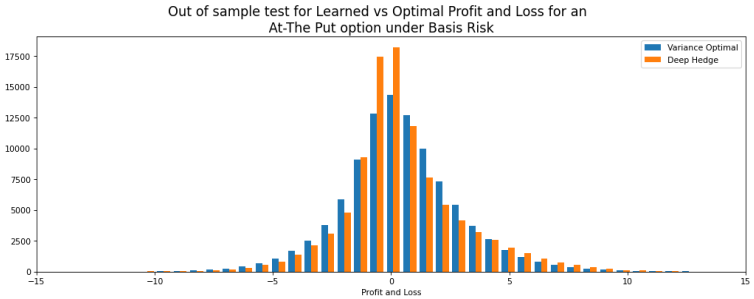

Fig. 4.12: PnL for an ATM Put Option in an Incomplete Market under Expected Shortfall.

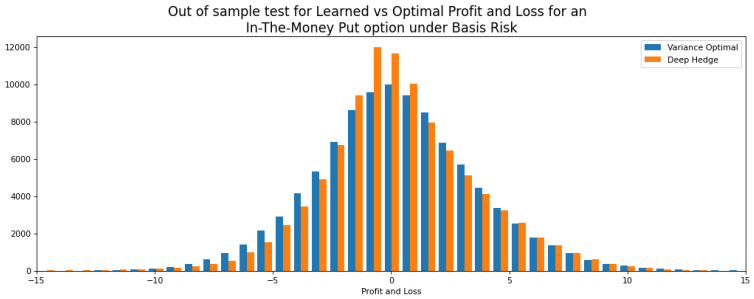

Fig. 4.13: PnL for an ITM Put Option in an Incomplete Market under Expected Shortfall.

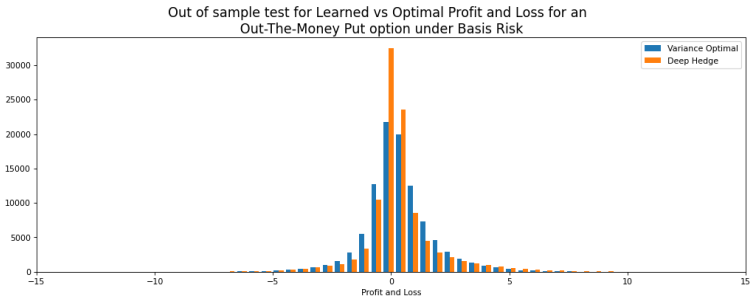

Fig. 4.14: PnL for an OTM Put Option in an Incomplete Market under Expected Shortfall.

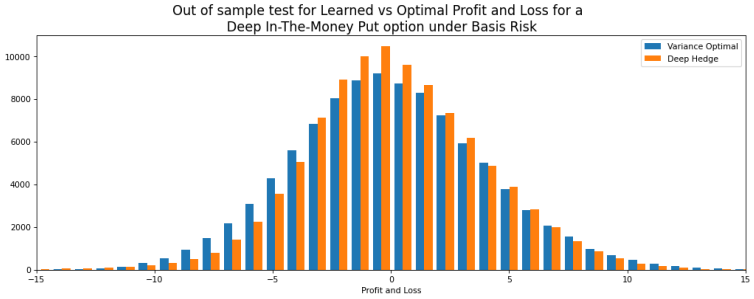

Fig. 4.15: PnL for a Deep ITM Put Option in an Incomplete Market under Expected Shortfall.

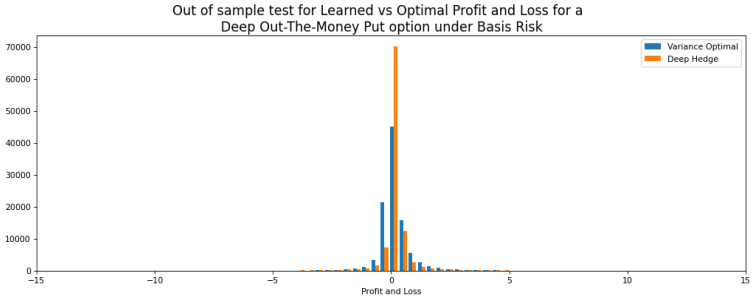

Fig. 4.16: PnL for a Deep OTM Put Option in an Incomplete Market under Expected Shortfall.

Chapter 5

Conclusion

There were a number of limitations in this dissertation. The most significant being the amount of computational power available. Deep reinforcement learning is a very computationally intensive process, and there was only limited amount of GPU access available. These limits on computing power meant that the correlation and volatility parameters had to be fixed in both the complete and incomplete market tests. If Deep Hedging were to be used in a production environment, the volatility and correlation should be inputs to the model. Moreover, these limits on computing power meant that there was a restriction on the amount training data and number of training epochs that could be used. This was unfortunate as this could have improved the performance of the neural networks used in this dissertation. Another limitation is the fact that this dissertation did not tackle pricing. We made the simplifying assumption that the writer is a price taker, and the price is set by external factors. Although the option premium is often set by competitive factors, pricing and hedging are closely related. Therefore, future research into the Deep Hedging Basis Risk should consider how one would integrate pricing into the Deep Hedging framework.

Despite these limitations, the results from this dissertation are promising. We showed that under certain limiting conditions, Deep Hedging is a good approximation for optimal hedging strategies in both a complete and incomplete market settings. Additionally, in the case of an incomplete market, we showed that by changing the loss measure from a quadratic measure to expected shortfall, we were able to outperform the McWalter's variance optimal strategy for hedging basis risk. Further research could investigate whether or not these results still hold if some of the limiting restrictions on this dissertation are removed.

These findings have important applications for risk management. Although the loss measures used in this dissertation were mean square error and expected shortfall, any coherent risk measure can be used. Furthermore, Deep Hedging is agnostic to the underlying data generating process, which easily allows for hedging

of basis risk under numerous market situations and market frictions, not just under the assumption of geometric Brownian motion without transaction costs, as is done in this dissertation. This increases the modelling flexibility available for the option writer, which ultimately leads to better risk management of basis risk.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
URL: *http://tensorflow.org/*

Airouss, M., Tahiri, M., Lahlou, A. and Hassouni, A. (2018). Advanced expected tail loss measurement and quantification for the Moroccan all shares index portfolio, *Mathematics* **6**(3): 38.

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

Bolcskei, H., Grohs, P., Kutyniok, G. and Petersen, P. (2019). Optimal approximation with sparsely connected deep neural networks, *SIAM Journal on Mathematics of Data Science* **1**(1): 8–45.

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

Buehler, H., Gonon, L., Teichmann, J., Wood, B., Mohan, B. and Kochems, J. (2019b). Deep hedging: hedging derivatives under generic market frictions using reinforcement learning, *Swiss Finance Institute Research Paper* .

Cao, J., Chen, J., Hull, J. and Poulos, Z. (2021). Deep hedging of derivatives using reinforcement learning, *The Journal of Financial Data Science* **3**(1): 10–27.

Choulli, T., Vandaele, N. and Vanmaele, M. (2010). The Föllmer–Schweizer decomposition: comparison and description, *Stochastic Processes and their Applications* **120**(6): 853–872.

Coughlan, G. D., Khalaf-Allah, M., Ye, Y., Kumar, S., Cairns, A. J. G., Blake, D. and Dowd, K. (2011). Longevity hedging 101, *North American actuarial journal* **15**(2): 150–176.

Cuchiero, C. and Teichmann, J. (2019). Lecture notes for machine learning in finance.

Davis, M. (1997). Option pricing in incomplete markets, *Mathematics of derivative securities* **15**: 216–226.

Davis, M. (2000). Option valuation and hedging with basis risk, *System Theory*, Springer, pp. 245–254.

Davis, M. (2006). *Optimal Hedging with Basis Risk*, From Stochastic Calculus to Mathematical Finance, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 169–187.

Figlewski, S. (1984). Hedging performance and basis risk in stock index futures, *The journal of finance* **39**(3): 657–669.

Föllmer, H. and Schweizer, M. (1991). Hedging of contingent claims under incomplete information, *Applied stochastic analysis* **5**(389-414): 19–31.

Föllmer, H. and Sondermann, D. (1985). Hedging of non-redundant contingent claims, *Technical report*, Sonderforschungsbereich 303, Bonn (Germany).

Halperin, I. (2017). Relative option pricing in the qlbs model (supplementary note), *Available at SSRN 3090608* .

Halperin, I. (2019). The qlbs q-learner goes nuqlear: fitted q iteration, inverse rl, and option portfolios, *Quantitative Finance* **19**(9): 1543–1553.

Halperin, I. (2020). Qlbs: Q-learner in the black-scholes (-merton) worlds, *The Journal of Derivatives* **28**(1): 99–122.

Haugh, M. (2016). Lecture notes for quantitative risk management.

Heath, D., Platen, E. and Schweizer, M. (2001). Numerical comparison of local risk-minimisation and mean-variance hedging, *in* E. Jouini, J. Cvitanić and M. Musiela (eds), *Option pricing, interest rates and risk management*, Cambridge University Press, Cambridge, United Kingdom, pp. 509–537.

Henderson, V. and Hobson, D. (2008). Utility indifference pricing: an overview, *in* R. Carmona (ed.), *Indifference pricing: theory and applications*, Princeton University Press, Princeton, United States of America, pp. 44–73.

Higham, C. F. and Higham, D. J. (2019). Deep learning: An introduction for applied mathematicians, *SIAM Review* **61**(4): 860–891.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural computation* **9**(8): 1735–1780.

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

Hulley, H. and McWalter, T. A. (2015). Quadratic hedging of basis risk, *Journal of Risk and Financial Management* **8**(1): 83–102.

Kalman, R. (1963). The theory of optimal control and the calculus of variations, *in* R. Bellman (ed.), *Mathematical optimization techniques*, University of California Press, Los Angeles, United States of America, pp. 309–331.

Kim, H. (2021). Deep hedging, generative adversarial networks, and beyond, *arXiv preprint arXiv:2103.03913* .

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization, *International Conference on Learning Representations (ICLR)*.

Li, J. S.-H. and Hardy, M. R. (2011). Measuring basis risk in longevity hedges, *North American actuarial journal* **15**(2): 177–200.

Lipton, Z. C., Berkowitz, J. and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning, *arXiv preprint arXiv:1506.00019* .

McWalter, T. A. (2006). *Quadratic criteria for optimal martingale measures in incomplete markets*, M.Sc. dissertation, University of Witswatersrand, Witswatersrand, South Africa.

Merton, R. C. (1974). On the pricing of corporate debt: The risk structure of interest rates, *The Journal of finance* **29**(2): 449–470.

Monoyios, M. (2004). Performance of utility-based strategies for hedging basis risk, *Quantitative finance* **4**(3): 245–255.

Monoyios, M. (2010). Utility-based valuation and hedging of basis risk with partial information, *Applied Mathematical Finance* **17**(6): 519–551.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*, MIT press.

Olah, C. (2015). Understanding LSTM networks.
URL: *http://colah.github.io/posts/2015-08-Understanding-LSTMs/*

Ruder, S. (2016). An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747* .

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *Nature* **323**(6088): 533–536.

Schweizer, M. (1988). *Hedging of options in a general semimartingale model*, PhD thesis, ETH Zurich.

Schweizer, M. (1990). Risk-minimality and orthogonality of martingales, *Stochastics: An International Journal of Probability and Stochastic Processes* **30**(2): 123–131.

Schweizer, M. (1992). Mean-variance hedging for general claims, *The annals of applied probability* **2**(1): 171–179.

Schweizer, M. (1994). Approximating random variables by stochastic integrals, *The Annals of Probability* **22**(3): 1536–1575.

Staudemeyer, R. C. and Morris, E. R. (2019). Understanding LSTM - a tutorial into long short-term memory recurrent neural networks, *arXiv preprint arXiv:1909.09586* .

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*, MIT press.

Watkins, C. J. and Dayan, P. (1992). Q-learning, *Machine learning* **8**(3): 279–292.

Werbos, P. J. (1974). *Beyond regression : new tools for prediction and analysis in the behavioral sciences*, PhD thesis, Harvard University, Cambridge, United States of America.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it, *Proceedings of the IEEE* **78**(10): 1550–1560.

Woodard, J. D. and Garcia, P. (2007). Basis risk and weather hedging effectiveness, *Management of Climate Risks in Agriculture*, EAAE Seminar Paper.

Zariphopoulou, T. (2001). A solution approach to valuation with unhedgeable risks, *Finance and stochastics* **5**(1): 61–82.

Zhang, J. and Huang, W. (2021). Option hedging using LSTM-RNN: an empirical analysis, *Quantitative Finance* **21**(10): 1–20.

Zhang, J., Tan, K. S. and Weng, C. (2017). Optimal hedging with basis risk under mean–variance criterion, *Insurance: Mathematics and Economics* **75**: 1–15.

Appendix A

Appendix

The code for the dissertation can be found here,here, here and here

A.1 Graphs Showing Loss versus Epochs during Training

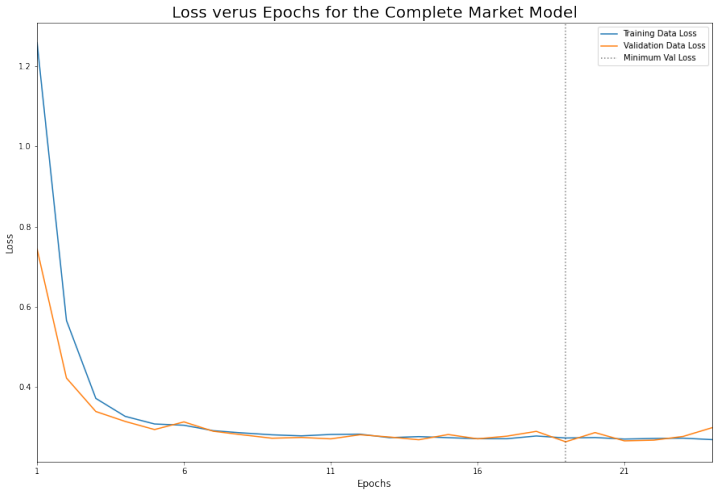

Fig. A.1: Loss versus Epochs Graph for the Complete Market Test under Expected Shortfall.

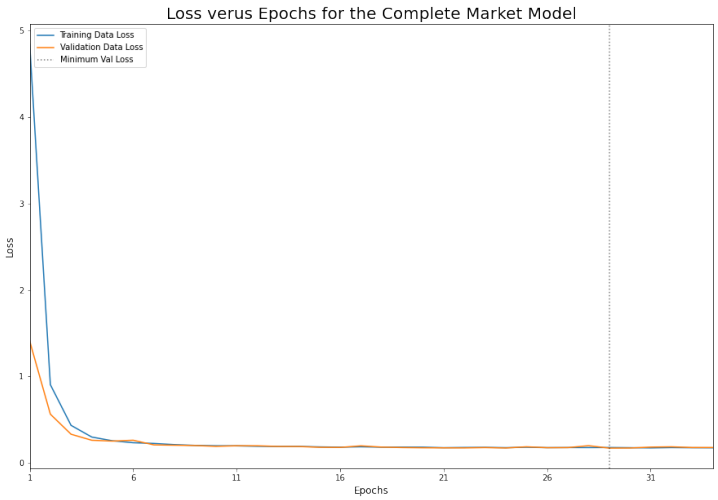

Fig. A.2: Loss versus Epochs Graph for the Complete Market Test under Mean Square Error.

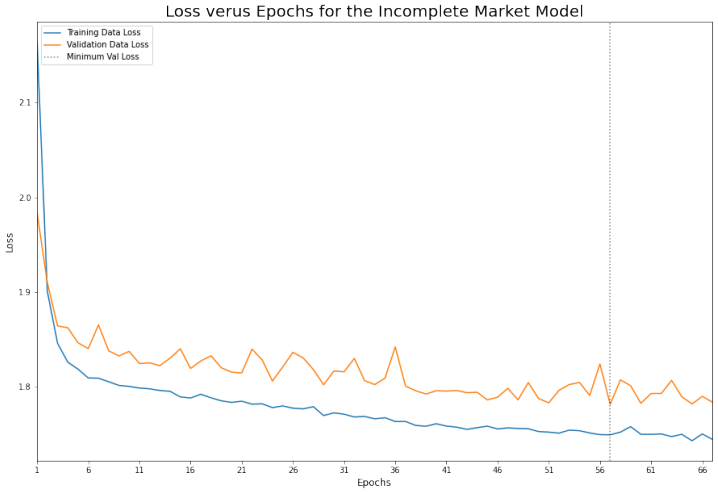

Fig. A.3: Loss versus Epochs Graph for the Incomplete Market Test under Expected Shortfall.

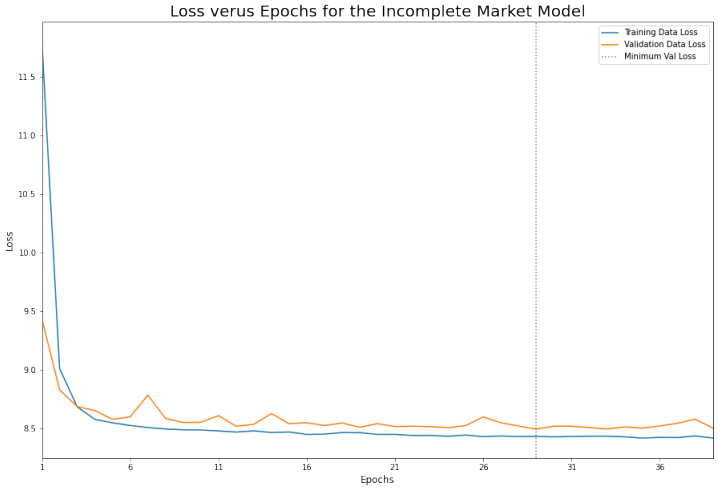

Fig. A.4: Loss versus Epochs Graph for the Incomplete Market Test under Mean Square Error.