Deep Learning Approaches for Hedging

by

Candidate Number: MFNY5

August 2025

Dissertation Supervisor: Dr. Codina Cotar

Dissertation submitted in part fulfilment of the

Degree of Master of Science in Data Science

Department of Statistical Science

University College London

[Word Count: 14,405]

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In the financial industry, Greek hedging based on the Black–Scholes model is widely applied to reduce risk exposures. However, empirical studies show that its assumptions, particularly frictionless markets and constant volatility, mismatch real market conditions, undermining cost efficiency and performance. This thesis investigates alternative hedging approaches from statistical and machine learning models, including GARCH, Heston model, neural networks, and Q-learning. Methodologies are first reviewed in the literature and then evaluated using Conditional Value-at-Risk (CVaR) on data simulated from a Heston model calibrated to SP 500 options. Hedging approaches with the GARCH model and the Heston model were excluded from the experiments due to their limitations on pricing and Greek calculation, and Q-learning was omitted for computational intractability. The experimental design varied liability types, hedging instruments, hedging frequency, proportional transaction costs, and CVaR confidence levels. Across all settings, neural network hedging consistently delivered greater and more robust CVaR reductions at lower transaction costs, while Greek hedging performed poorly for discontinuous payoffs. Overall, neural networks emerge as a robust alternative under realistic market conditions.

Keywords: Deep learning, Hedging, Derivatives

1. INTRODUCTION

In the financial market, financial institutions and investors routinely acquire financial liabilities in the business, trading and investment activities. With the market dynamics, the value of such financial liabilities can fluctuate unexpectedly, creating uncertainty and risk exposure to financial institutions and investors. Such uncertainties and risks can result in unforeseeable payoff from the financial liabilities. Therefore, financial institutions and investors must implement risk management to protect themselves from unfavourable and unpredictable outcomes. One of the practice in risk management is known as hedging, which involves entering an offsetting positions in certain financial instruments to reduce the risk from the financial liabilities partially or completely.

Among the various hedging approaches, **Greek hedging** is one of the most widely adapted practices in financial industries. The method involves assessing the risk sensitivities to some underlying risk factors that drive the fluctuation of the value of the financial liability and entering positions of financial instruments with offsetting risk sensitivities. It usually relies on a framework provided by the **Black-Scholes model** introduced by Black and Scholes in 1973. The model relies on restrictive assumptions: frictionless market, continuous trading, constant risk-free rate, constant volatility, and log-normal distribution of the equity price. However, empirical evidence consistently demonstrates the existence of market friction due to transaction cost and liquidity, discrete trading, stochastic volatility, and the distribution of the equity price deviating from log-normal. These discrepancies between theory and reality raise concerns about the robustness and performance of Greek hedging in realistic trading environments.

To address these limitations, this thesis aims to find **alternative hedging methodologies**, particularly with machine learning tools, that can account for more realistic market dynamics, especially with stochastic volatility, transaction cost, discrete trading, and market incompleteness. Recent advances in machine learning, particularly deep hedging framework advocated by Buehler and his fellow researchers in 2018, provide powerful tools for modelling high-dimensional, non-linear relationship and decision marking under uncertainty. Therefore, in addition to statistical models such as **the generalised autoregression conditional heteroskedasticity model (GARCH)** and the **Heston model**, machine learning models such as **neural network model** and **Q-learning model** are also elaborated and evaluated in the literature and computational simulation experiments.

For a comprehensive comparison of different methodologies in hedging, the computational simulation will incorporate some realistic market situation with prudential consideration on the balance between the effect on the risk and the computational complexity. To align with the aim of this thesis, the stochastic volatility, transaction cost, trading frequency, and availability of financial instruments are explicitly encoded in the simulation and testing environment. To construct and evaluate models for hedging purposes, a risk measure called **conditional value-at-risk (CVaR)** is reviewed with out-of-sample data. Furthermore, different financial liabilities with the **S&P 500 Index** as the underlying risk factor are examined as illustrative examples, in order to provide an insight on how the methodologies handle financial liabilities with different complexity. Ultimately, the objective of thesis is to determine whether machine learning models can deliver a practically viable and theoretically robust improvement over classical Greek hedging, being an alternative hedging approach.

2. LITERATURE REVIEW

The literature review begins by introducing fundamental financial concepts and common hedging practices in order to establish a baseline understanding of **(2.1) hedging**. It then discusses the **(2.2) Black–Scholes model**, outlining its assumptions, applications in delta and Greek hedging, and the limitations of these approaches in realistic market settings. Once the underlying shortcomings of classical hedging approach are identified, it turns to alternative statistical models, including the **(2.3) GARCH model** and the **(2.4) Heston model**, with emphasis on their strengths and weaknesses in capturing stochastic volatility. The review then proceeds to machine learning approaches, such as **(2.5) neural network model** and **(2.6) Q-learning model**, which are less familiar in the financial literature. These sections describe the model structures, fundamental mathematics, working principles, limitations, and their alignment with the hedging problem. After presenting the alternative approaches, the review introduces risk measures, particularly **(2.7) Conditional Value-at-Risk (CVaR)**, as both the optimisation objective and the performance metric for comparing hedging strategies in this thesis. Finally, the **(2.8) gradient descent methods**, which underpin the training of machine learning models, are explained to provide technical context for the computational methodology.

2.1 Hedging

This section provides an overview of hedging in financial markets. It begins with a basic understanding of risk and hedging. The discussion then shifts to the sources and natures of financial liabilities and risk exposures faced by financial institutions and the common financial instruments used in hedging, and finishes with the concept of the classic hedging approach delta and Greek hedging with an illustrative example.

Hedging is a practice of reducing or mitigating financial risk exposure by taking offset positions in some financial instruments that have the same or similar underlying instruments or correlation with the underlying risk factors [12]. In financial markets, risks are the uncertainty of certain events that can deviate the actual outcomes from the expected outcomes and usually associate with unfavourable outcomes, driven by some underlying causes or risk factors. In risk management, risks are usually grouped by the underlying risk factors, for example, market risk driven by market movements and operation risk driven by flawed or failed operation process [12].

Financial institutions regularly acquire financial liabilities and risk exposures as a result of their daily business activities, for example, selling financial instruments to their clients. It is essential for them to hedge their risk exposure beyond their presumed risk exposures to protect themselves from unfavourable price movements, interest rate changes, volatility shift, settlement failure or default, and any other market uncertainties. In reality, not all risks can be hedged perfectly due to the unique and/or complicated nature of some risks and market incompleteness, where offsetting instruments are not readily available, leaving certain risks unhedged. However, some financial institutions intentionally remain with some risks within acceptable levels, known as risk tolerance, because the cost of hedging those risks could be costly or, sometimes, they expect to profit from the risk exposures.

To effectively reduce or mitigate risks, financial institutions must formulate and apply hedging strategies to align their risk exposures with their risk tolerance. This thesis will focus on hedging strategies for market risk from derivatives trading, where hedging is constantly practised. The following subsections will cover what financial liabilities associated with market risk exposures financial institutions deal with, what common financial instruments can be used in hedging, and how the classic hedging approach, delta and Greek hedging, is conducted.

2.1.1 Financial Liability and Risk Exposure

The daily activities of financial institutions inherently involve the acquisition of financial risks. For example, borrowing and lending create exposures to interest rate and credit risk, and transactions in foreign currencies add foreign exchange risk. In financial markets, institutions also buy and sell financial instruments to profit from price fluctuations or bid–ask spreads (difference between prices to buy and to sell) . Such trading may involve homogenous products traded in exchange houses as well as derivatives and highly customised structured products of different asset classes traded over the counter outside exchange houses. These positions with uncertain future payoffs can generate potential liabilities to the financial institutions, so the risk exposure is also called financial liability. Consequently, financial institutions are typically exposed to a wide range of underlying instruments and risk factors simultaneously, rendering hedging a complex and challenging task. In this study, hedging strategies are evaluated against risk exposures of varying complexity to reflect these practical challenges.

2.1.2 Hedging Instruments

Financial institutions have access to a variety of financial instruments in financial markets for investment, speculation, and hedging. Practitioners should prudently assess the characters and underlying risk factors of risk exposure or liability and explore the possibility and availability of financial instruments in the market to determine which hedging instruments should be used. For a better picture of how hedging is practised, it is important to understand some common hedging instruments.

Cash and risk-free equivalence Theoretically, cash and risk-free equivalence does not fluctuate with market movement, i.e. zero risk sensitivities (Greeks, more details in later section) to most risk factors and cannot hedge market risks, except foreign exchange risk. However, they still play very important roles in the hedging process by funding hedging strategies and modelling the time value of money: A positive cash balance earns risk-free lending rates r_L and a negative cash balance costs risk-free borrowing rates r_B.

Underlying Instrument The underlying instrument is one of the most commonly used hedging instruments. It has a risk sensitivity to the underlying instrument itself (Delta) 1 and theoretically zero risk sensitivities to other risk factors (other Greeks) concerning derivatives, so it can provide a very direct and efficient hedge when only Delta is considered. In the financial market, stocks can pay a dividend as a redistribution of the cash reserve to the shareholder and is measured by the dividend yield q.

Forward A forward is a contract between two parties to buy or sell an underlying instrument at an agreed price, the forward price F, at a specific future time T [12]. At the initial time t, neither party has to pay or receive any cash exchange and at T, the payoff of a forward is $S_T - F$ with the price of underlying instrument S_T. If a financial institution has a financial liability with payoff S_T, then buying a forward can create a constant net payoff $-F$, removing the uncertainty.

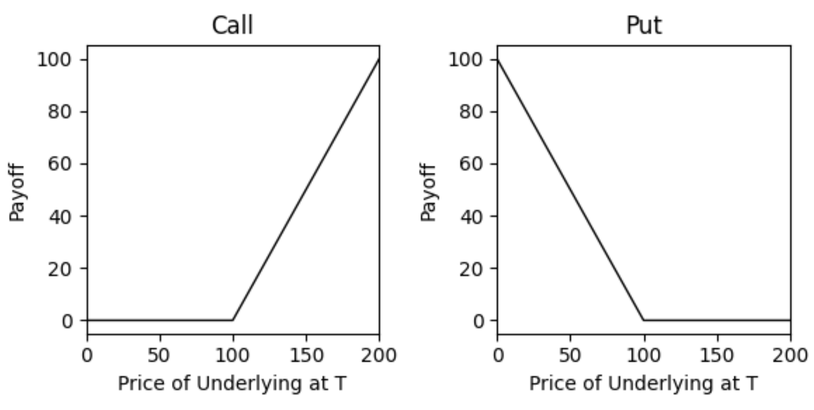
By comparing the payoff structure of a forwards and an underlying instrument as $S_T - S_0$, a constant difference $F - S_0$ is result. Therefore, practitioners can create a synthetic future by borrowing at risk-free rate to buy the underlying instrument to replicate the payoff of a future. When the underlying instrument is not easily available, for example, indices and non-deliverable currencies, forward can be used with lending instead to mimic the underlying instrument.

$$\text{Forward price } F_{t,T} = S_t e^{-(r-q)(T-t)} \tag{1}$$

$$\text{Payoff at time } T = S_T - F_{t,T} \tag{2}$$

European Call and Put An European call is an option that allows the option holder to buy the underlying instrument at an agreed price, the strike price K, at a specific future time T, and an European put is an option that allows the option holder to sell the underlying instrument at an agreed price, the strike price K, at a specific future time T [12]. Unlike forward, options give buyers rights instead of obligation, so an option can be not executed on the expiration date T when the payoff of execution the option is negative, making the payoff of an European call $max(0, S_T - K)$. Similarly, put is a right to sell so the payoff is $max(0, K - S_T)$.

Figure 2.1: Payoff of European call and put with strike price 100



With the payoff structure of call and put, by buying a call and selling a put both with strike price K and maturity T, the total payoff should be constant as $S_T - K$. According to the no-arbitrage theory that suggests that financial products that have identical payoff should have the same value, the value of buying a call ad selling a put at t should have the same value as buying $e^{-q(T-t)}$ share of the underlying instrument and borrowing the present value of K. This relationship is called call-put parity [12].

Definition 1. Call-Put Parity: At any $t \geq 0$, given the risk-free rate r, the prices of a call and a put with the same underlying instrument with price S_t and dividend yield q, the same strike price K and maturity T should satisfy the following equation [12]:

$$C_t(K) - P_t(K) = \underbrace{S_t e^{-q(T-t)}}_{e^{-q(T-t)} \text{ share of underlying instrument}} - \underbrace{K e^{-r(T-t)}}_{\text{present value of risk-free note face value K}} \tag{3}$$

$$\tag{4}$$

In addition to its simple payoff structure, European call and put have elegant closed-form solutions by the Black-Scholes model in **2.1.4** for calculation of their prices and risk sensitivities (Greeks), making them highly accessible and convenient in delta and Greek hedging.

2.1.3 Delta and Greek Hedging

One of the popular hedging strategies for market risk is called delta hedging, which focuses on the rate of change of the value V with respect to the price of the underlying instruments S, known as Delta Δ, and usually responds on a stochastic model called the Black-Scholes model. Delta hedging basically tunes the total delta of a portfolio to zero or a target level by adjusting positions of financial instruments. accordingly. Regarding derivatives and structured products non-linear to the price of the underlying instrument, delta hedging may not be enough to mitigate any non-linear risk exposure and risk exposures to other risk factors such as volatility change σ, time to expiration $\tau = T - t$, and interest rate r. Then an improved version, called Greek hedging, provides a more robust hedging by including more rates of change of the value V with respect to other risk factors in first- and higher-order, denoted by Greek letters as below.

First-order rate of change of value (risk sensitivity) with respect to [12]:

$$\begin{aligned}
&\text{Price of underlying instrument } S, \text{delta } \Delta = \frac{\partial V}{\partial S} \\
&\text{Volatility of underlying instrument } \sigma, \text{Vega } \mathcal{V} = \frac{\partial V}{\partial \sigma} \\
&\text{Time to expiration } \tau, \text{theta } \Theta = -\frac{\partial V}{\partial \tau} \\
&\text{Interest rate } r, \text{rho } \rho = -\frac{\partial V}{\partial r}
\end{aligned}$$

Second-order rate of change of value (risk sensitivity) with respect to:

$$\text{Price of underlying instrument } S, \text{gamma } \Gamma = \frac{\partial^2 V}{\partial S^2}$$

As the underlying risk factors such as the interest rate r, price S_t, volatility σ and dividend yield q of the underlying instrument are all subject to random movement, Greeks can be considered as risk measure or risk sensitivities [12]. To manage risks to different random underlying risk factors, financial institutions can take certain actions to reduce or increase the values of Greeks of a portfolio according to their risk tolerance and limit. Similarly to delta hedging, Greek hedging is a practice to reduce the value of Greeks to zero or target level and usually involves adding positions with opposite values of Greeks to cancel out the existing Greeks. Therefore, Greek hedging is a broader category of delta hedging, which solely involves reduction of one Greek of a portfolio.

This delta and Greek hedging process can be continuous throughout the life of a portfolio with Greeks varying over time. In delta hedging with non-zero Gamma, Delta continuously changes with respect to the price of its underlying instrument and that may require continuous hedging action, called rebalancing, to keep the Delta align with the risk limit. The hedging frequency can vary from multiple times within a day to once a month or even less frequently. With positive transaction costs in reality, frequent rebalancing can result in a high total transaction costs, but infrequent rebalancing can result in unwanted risk exposure, therefore, financial institution have to balance between the cost and the risk tolerance.

In the delta hedging process, for a portfolio with n instruments with positions $\mathbf{N_P} = \{N_i\}_{i=1}^{n} \in \mathbb{R}^n$ and m possible hedging instruments, the deltas per position of the instruments in the portfolio $\boldsymbol{\Delta_{P,t}} = \{\Delta_{i,t}\}_{i=1}^{n} \in \mathbb{R}^n$ and the delta of the portfolio $\Delta_{Portfolio,t}$ are first calculated as the sum of the deltas of all positions in the portfolio. Then the Deltas per position of the hedging instruments $\boldsymbol{\Delta_{H,t}} = \{\Delta_{i,t}^{Hedge}\}_{i=1}^{m} \in \mathbb{R}^m$ are also calculated. Finally, the positions of each hedging instruments $\mathbf{N_{H,t}} = \{N_{i,t}^{Hedge}\}_{i=1}^{m} \in \mathbb{R}^m$ are determined by equating the Delta of the hedged portfolio Δ_{Hedged} to either zero or the target level D^*, and finally the trades are executed accordingly to offset the Delta of the portfolio. The frequency of hedging can vary from intra-day rebalancing to only when the Delta of the hedged portfolio exceeds the limit. With a single equation and multiple hedging instruments, there could be infinite solutions for $\mathbf{N_{H,t}}$. In real trading situation, there could be more constraints like capital limit, liquidity, transaction cost, position limit, short selling restriction, counterparty risk limit, etc.

Delta hedging does not consider the other Greeks, so it could increase the other Greeks unknowingly. To prevent undesired risks, traders would in fact consider other Greeks of the portfolio as in Greek hedging. Similarly, the process with d Greeks will require calculation of the Greeks per position of instruments in the portfolio $\mathbf{G_{P,t}} = \{G_{i,j,t}\}_{i,j=1}^{n,d} \in \mathbb{R}^{n\times d}$ and hence the Greeks of the portfolio $\mathbf{G_{Portfolio}} = \{G_{j,t}^{Portfolio}\}_{j=1}^{d} \in \mathbb{R}^d$. With Greeks per position of the possible hedging instruments $\mathbf{G_{H,t}} = \{G_{i,j,t}^{Hedge}\}_{i,j=1}^{m,d} \in \mathbb{R}^{m\times d}$, the positions of the hedging instruments $\mathbf{N_{H,t}}$ are determined by equating the post-hedging values of Greeks to be either $\mathbf{0} = \{0\}_{j=1}^{d} \in \mathbb{R}^d$ or $\mathbf{G}^* = \{G_j^{Target}\}_{j=1}^{d} \in \mathbb{R}^d$.
Delta hedging [12]:

$$\Delta_{Portfolio,t} = \mathbf{N_P}^\top \boldsymbol{\Delta_{P,t}} = \sum_{i=1}^{n} N_i \Delta_{i,t} \tag{5}$$

$$\Delta_{Hedged,t} = \Delta_{Portfolio,t} + \mathbf{N_{H,t}}^\top \boldsymbol{\Delta_{H,t}} = \Delta_{Portfolio,t} + \sum_{i=1}^{m} N_{i,t}^{Hedge} \Delta_{i,t}^{Hedge} = 0 \text{ or } D^* \tag{6}$$

Greek Hedging [12]:

$$\mathbf{G_{Portfolio,t}} = \mathbf{N_p}^\top \mathbf{G_{P,t}} = \left\{ \sum_{i=1}^{n} N_i G_{i,j,t} \right\}_{j=1}^{d} \tag{7}$$

$$\mathbf{G_{Hedged,t}} = \mathbf{G_{Portfolio,t}} + \mathbf{N_{H,t}}^\top \mathbf{G_{H,t}} = \left\{ \sum_{i=1}^{n} N_i G_{i,j,t} + \sum_{i=1}^{m} N_i^{Hedge} G_{i,j,t}^{Hedge} \right\}_{j=1}^{d} = \mathbf{0} \text{ or } \mathbf{G}^* \tag{8}$$

Example 1: A portfolio has one share of instrument A and one share of instrument B with the same underlying instrument S. Currently, A has Delta 0.55 and Gamma 0.03, and B has Delta 0.45 and Gamma 0.03. S has price £100, Delta 1 and Gamma 0 as $\frac{\partial S}{\partial S} = 1$ and $\frac{\partial^2 S}{\partial S^2} = 0$ and another instrument C with the underlying instrument S has price £4, Delta 0.2 and Gamma 0.02. The target Delta is 0 and either Gamma or the total value of the hedging instruments should be 0.

Delta hedging with additional constraint:

$$\begin{aligned}
\Delta_{Portfolio} &= 1 \times 0.55 + 1 \times 0.45 = 1 \\
\Delta_{Hedged} &= 1 + (n_S + 0.2 n_C) = 0 \Rightarrow n_S + 0.2 n_C = -1 \\
V_{Hedge} &= 100 n_S + 4 n_C = 0 \\
\Rightarrow n_S &= 0.25, \quad n_C = 6.25
\end{aligned}$$

Greek hedging:

$$\begin{aligned}
G_{Portfolio} &= [1 \quad 1] \begin{bmatrix} 0.55 & 0.03 \\ 0.45 & 0.03 \end{bmatrix} = [1 \quad 0.06] \\
G_{Hedged} &= \begin{bmatrix} 1 & 0.06 \end{bmatrix} + [n_S \quad n_C] \begin{bmatrix} 1 & 0 \\ 0.2 & 0.02 \end{bmatrix} = [0 \quad 0] \\
\Rightarrow n_S &= -0.4, \quad n_C = -3
\end{aligned}$$

2.2 Black-Scholes Model

After going through the background of delta and Greek hedging, this section focuses their supporting model, the Black-Scholes model, and covers its assumption, basic mathematics, closed-form solution for price and Greeks of European options, and eventually its limitation to model the real financial market.

In order to measure the Greeks, it is essential to know how the value of a financial instrument is determined by the risk factors. In 1973, Fischer Black and Myron Scholes published a paper called "The Pricing of Options and Corporate Liabilities" and defined a stochastic model, the Black-Scholes model, to provide a simple and elegant solution for European option pricing with Itô's Lemma [3], [12]. Since then, the Black-Scholes model has become the foundational framework for Greek hedging and modern quantitative finance.

This model is based on a set of core assumptions: (a) a stochastic price S_t following a geometric Brownian motion resulting in a log-normal distribution; (b) constant risk-free rate r in risk-neutral measure, constant dividend yield q and constant volatility σ; (c) in a market setting with continuous and frictionless trading and financing at risk-free rate, without arbitrage, and without default risk. The following mathematical definitions are important to establish a basic understanding of the Black-Scholes model.

Definition 2. Brownian Motion: A continuous process $\{W_t, t \geq 0\}$ is a standard Brownian motion if and only if: (a) $W_0 = 0$, (b) $dW_t = W_{t+dt} - W_t \sim \text{Normal}(0, dt)$, and (c) for all $0 \leq p < q \leq r < s, W_q - W_p$ is independent of $W_s - W_r$ [12].

Definition 3. Generalised Brownian Motion: A process $\{X_t, t \geq 0\}$ is a generalised Brownian motion with drift rate a and volatilty rate $b > 0$ satisfies the following stochastic differential equation (SDE) [12]:

$$dX_t = adt + bdW_t \Rightarrow X_t \sim \text{Normal}(at, b^2t) \tag{9}$$

Definition 4. Geometric Brownian Motion (GBM): A process $\{X_t, t \geq 0\}$ is a geometric Brownian motion with drift rate μX_t and volatility rate $\sigma X_t > 0$ satisfies the following SDE [12]:

$$dX_t = \mu X_t dt + \sigma X_t dW_t \tag{10}$$

Definition 5. Itô's Lemma: A process $\{X_t, t \geq 0\}$ is an Itô's process with drift rate $a(X_t, t)$ and volatility rate $b(X_t, t) > 0$ as functions of X_t and t satisfies the following SDE [12]:

$$dX_t = a(X_t, t)dt + b(X_t, t)dW_t \tag{11}$$

Then, for a function $G(X_t, t)$, G is also a Itô's process with SDE:

$$dG = \left[\frac{\partial G}{\partial X_t} a(X_t, t) + \frac{\partial G}{\partial t} + \frac{\partial^2 G}{\partial X_t^2} b^2(X_t, t)\right] dt + \frac{\partial G}{\partial X_t} b(X_t, t) dW_t \tag{12}$$

Definition 6. Black-Scholes Model: Under the assumption of frictionless and continuous trading and financing without arbitrage in the market, an asset with price following the below geometric Brownian motion SDE, with a constant risk-free rate r, constant dividend yield q and constant volatility σ, should have a log-normal distribution [3], [12].

$$\text{SDE: } dS_t = (r - q)S_t\, dt + \sigma S_t\, dW_t^S \tag{13}$$

$$\Rightarrow \ln \frac{S_t}{S_0} \sim \text{Normal}\left[(r - q - \frac{\sigma^2}{2})t, \sigma^2 t\right] \tag{14}$$

$$S_t = S_0 \exp\left\{(r - q - \frac{\sigma^2}{2})t + \sigma W_t\right\}, \text{ where } W_t \sim \text{Normal}(0, t) \tag{15}$$

Proof: For $G(S_t, t) = \ln S_t$, by Itô's Lemma, the SDE of G:

$$dG = d\ln(S_t) = \left(r - q + \frac{\sigma^2}{2}\right) dt + \sigma dW_t \tag{16}$$

since

$$\frac{\partial G}{\partial S_t} = \frac{1}{S_t}, \frac{\partial^2 G}{\partial S_t^2} = -\frac{1}{S_t^2}, \frac{\partial G}{\partial t} = 0$$

Under the risk-neutral measure, with further application of Itô's Lemma, the model yields an elegant closed-form solution for the prices and the Greeks of European calls and puts with strike price K and maturity T as below [3], [12]. For clarity of exposition, and since the emphasis here is on the hedging applications of the model, the full derivation is omitted.

The price of European call at time t:

$$C_t(S_t, K, T) = S_t e^{-q(T-t)} N(d_1) - K e^{-r(T-t)} N(d_2) \tag{17}$$

and the price of European put at time t:

$$P_t(S_t, K, T) = K e^{-r(T-t)} N(-d_2) - S_t e^{-q(T-t)} N(-d_1) \tag{18}$$

where

$$d_1 = \frac{\ln(S_t/K) + (r - q + \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}}, \quad d_2 = d_1 - \sigma\sqrt{(T-t)}$$

and $N(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

2.2.1 Greeks

With a well-defined option pricing solution, the Black-Scholes model can also provide closed-form solutions with calculus for the Greeks, the risk sensitivities of the price of derivatives to changes in one or more underlying risk factors. In **2.1.3**, these Greeks of a financial instrument can be calculated as first-order and second-order partial derivatives of its value with respect to different risk factors [12]. With the option pricing formula of the Black-Scholes model, the Greeks can be readily calculated with the formulae in the following table. Therefore, with well-defined closed-form solutions for the Greeks, the Black-Scholes model provides a direct and comprehensible guideline on delta and Greek hedging [12].

Table 2.1: Black-Scholes Greeks for European Call and Put Options

Greek	**Call Option**	**Put Option**
Delta (Δ_t)	$e^{-q\tau}N(d_1)$	$-e^{-q\tau}N(-d_1)$
Gamma (Γ_t)	$e^{-q\tau}N'(d_1)/S_0\sigma\sqrt{\tau}$	
Vega ($\mathcal{V}_t$)	$S_0e^{-q\tau}N'(d_1)\sqrt{\tau}$	
Theta (Θ_t)	$-\frac{S_0\sigma e^{-q\tau}N'(d_1)}{2\sqrt{\tau}} + qS_0e^{-q\tau}N(d_1)$ $-rKe^{-r\tau)}N(d_2)$	$-\frac{S_0\sigma e^{-q\tau}N'(d_1)}{2\sqrt{\tau}} - qS_0e^{-q\tau}N(-d_1)$ $+rKe^{-r\tau}N(-d_2)$
Rho (ρ_t)	$K\tau e^{-r\tau}N(d_2)$	$-K\tau e^{-r\tau}N(-d_2)$

2.2.2 Limitations

Although the Black-Scholes model has been widely used as the fundamental framework for Greek hedging and modern quantitative finance, it has limitations to suit the real situation. The model is based on a set of core assumptions which can be systematically violated in the real market, raising concerns on the efficiency and performance of the hedging.

First, the actual distribution of the price of an underlying instrument can deviate from the log-normal distribution in the assumptions; for example, the log-return of S&P 500 from 2010 to 2025 in Figure 2.1 shows a leptokurtosis pattern with a concentrated central density and slightly fatter tails compared to the normal distribution in (14). Second, there are empirical studies and market observations showing that volatilities are stochastic and not constant, i.e., heteroskedasticity, and Figure 2.2 illustrates how 1-month and 3-month rolling volatilities fluctuate and cluster over time. The volatility fluctuation can respond to new information and news, investor behaviour change, business cycle, market crush, etc. Lastly, continuous and frictionless trading and financing is, in fact, the most mismatched assumption. Although throughout the day there are high over-the-counter (OTC) activities without the limitation of trading hour window of exchange houses, this does not completely omit friction and discontinuous trading. The OTC transactions are usually for derivatives and limited for stocks and other exchange-traded financial instruments, and the liquidity could be low due to low trading activity outside exchange trading hour, i.e. financial instruments cannot be easily and quickly bought or sold in flexible quantity at stable prices. Moreover, transactions always associate regulation restrictions and transaction costs like tax, exchange and clearing house charges, brokerage fee, and bid-ask spread in a real trading environment. These violations of assumptions of the Black-Scholes model could result in discrepancies between the actual option price and the Greeks and the calculation by the Black-Scholes model. Therefore, the delta and Greek hedging based on the Black-Scholes model may not have desirable and efficient performance.

Figure 2.2: Daily log-return $\ln(S_t/S_0)$ of S&P 500 Index with normal distribution PDF

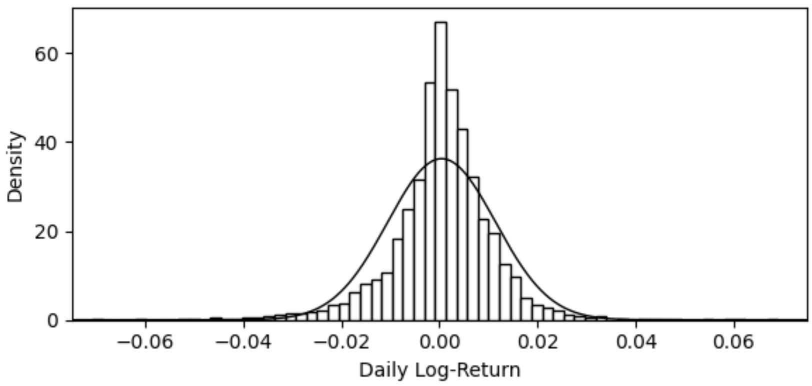
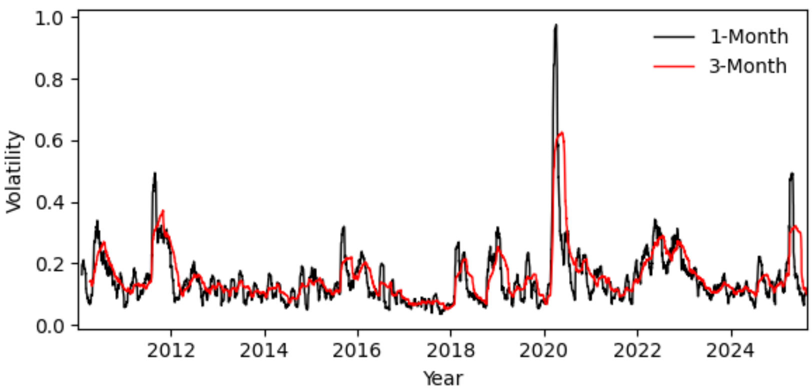


Figure 2.3: Rolling volatility of S&P 500 Index



Due to the above limitation of the Black-Scholes model, this thesis navigates the alternative models for stochastic volatility such as GARCH model and the Heston model and alternative approaches for hedging such as neural network and Q-learning. Each strategy is tested in a more realistic market setting with stochastic volatility, frictions including transaction costs and market incompleteness, and their performance will be evaluated and compared.

2.3 GARCH Model

To capture the stochastic volatility, an alternative class of discrete-time models, the Autoregressive Conditional Heteroskedasticity (ARCH) model by Engle (1982) [7] and its generalisation GARCH by Bollerslev (1986) [4], aims to capture such time-varying volatility.

Definition 7. GARCH Model: Let $r_t = \ln(S_t/S_{t-1})$ denote the log-return of the asset price S_t . The conditional variance of returns σ_t^2 evolves as [4]:

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sigma_t z_t, \quad z_t \sim \text{i.i.d. } N(0,1), \tag{19}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{20}$$

where

$$\omega > 0, \quad \alpha_i \geq 0, \quad \beta_j \geq 0, \quad \sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \beta_j < 1.$$

Example, GARCH(1,1):

$$\sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2, \quad \omega > 0,\ \alpha, \beta \geq 0,\ \alpha + \beta < 1. \tag{21}$$

The GARCH (p, q) model suggests that stochastic volatility σ_t^2 depends both on the p lagged squared return (ARCH term) and the q lagged variance (GARCH term) [4]. This structure enables formulation of volatility clustering, in which large shocks tend to be followed by high volatility, and small shocks by low volatility. As a result, the model provides a powerful description of stochastic volatility in financial returns.

GARCH can be fitted with historical data to capture the stochastic nature of volatility directly, and model returns with conditional heteroskedasticity, filling the gap of assumption violation of the Black-Scholes model. In practice, GARCH models are widely applied to model time series data to study volatility for risk management.

2.3.1 Limitations

GARCH model can model stochastic volatility efficiently; however, the application of the GARCH model to the hedging strategy for derivative trading can be limited. As a discrete-time model, the GARCH model has an inherent mismatch with the time-continuous stochastic calculus in option pricing. This could complicate the no-arbitrage pricing of options, and hence the hedging process. Moreover, the model does not provide a closed-form solution for option pricing. Instead, option pricing often requires computationally intensive Monte Carlo simulation. With the already computationally intensive hedging algorithms with neural network and Q learner reinforcement learning, the use of GARCH model can significantly increase the computational complexity and time to have a conclusive result.

2.4 Heston Model

Besides the GARCH model, there are more models that can formulate stochastic volatility. Heston model by Heston (1993) introduces a mean-reversion process of stochastic volatility on top of the framework on Black-Scholes model [11]. The model consists of two stochastic processes: the price of the underlying instrument as a geometric Brownian motion and the variance as a mean reversion process with a rate of mean reversion $\kappa > 0$, long-term mean $\theta > 0$ and volatility of the process $\sigma > 0$. The standard Brownian motion in these two processes dW_t^S and dW_t^v has correlation ρ [11].

Definition 8. Mean Reversion (Ornstein-Uhlenbeck) Process: A process $\{X_t, t \geq 0\}$ is a mean reversion process if it reverts towards a long-run mean θ with speed $\kappa > 0$ and has volatility rate $\sigma > 0$ that satisfies the following SDE [12]:

$$dX_t = \kappa(\theta - X_t)dt + \sigma dW_t, \tag{22}$$

Definition 9. Heston Model: Asset price S_t is a geometric Brownian mode with drift rate $\mu - q$ and its volatility rate v_t which is a mean reversion process with long-term mean $\theta > 0$, speed $\kappa > 0$ and volatility rate $\sigma > 0$. The standard Brownian motions in these two processes are correlated at $-1 \leq \rho \leq 1$ and satisfy the following SDEs [11]:

$$dS_t = (\mu - q)S_t\,dt + \sqrt{v_t}S_t dW_t^S \tag{23}$$

$$dv_t = \kappa(\theta - v_t)\,dt + \sigma\sqrt{v_t}dW_t^v \tag{24}$$

$$dW_t^S dW_t^v = \rho dt \tag{25}$$

2.4.1 European Option Pricing

In contrast to the GARCH model, the Heston model provides a semi-closed-form risk-neutral pricing solution for European calls and puts with strike price K and maturity T. Compared to the Black-Scholes model, the solution is more complicated with imaginary numbers and integration due to the complexity of the combination of stochastic price S_t and stochastic variance v_t [8], [11]:

$$\text{Call}(S_t, K, T) = S_t e^{-q\tau} P_1 - K e^{-r\tau} P_2 \tag{26}$$

$$\text{Put}(S_t, K, T) = K e^{-r\tau}(1 - P_2) - S_t e^{-q\tau}(1 - P_1) \tag{27}$$

For j = 1, 2

$$b_1 = \kappa - \rho\sigma, \quad b_2 = \kappa, \quad u_1 = 0.5, \quad u_2 = -0.5 \tag{28}$$

$$d_j(\phi) = \sqrt{(\rho\sigma\phi i - b_j)^2 - \sigma^2\phi(2u_j i - \phi)} \tag{29}$$

$$g_j(\phi) = \frac{b_j - \rho\sigma\phi i + d_j(\phi)}{b_j - \rho\sigma\phi i - d_j(\phi)} \tag{30}$$

$$C_j(\phi, T) = (r - q)\phi\tau i + \frac{\kappa\theta}{\sigma^2}\left[(b_j - \rho\sigma\phi i + d_j(\phi))\tau - 2\ln\left(\frac{1 - g_j(\phi)e^{d_j(\phi)\tau}}{1 - g_j(\phi)}\right)\right] \tag{31}$$

$$D_j(\phi, T) = \frac{b_j - \rho\sigma\phi i - d_j}{\sigma^2}\left(\frac{1 - e^{d_j(\phi)\tau}}{1 - g_j(\phi)e^{d_j(\phi)\tau}}\right) \tag{32}$$

$$f_j(\phi, S_t, v_t, T) = \exp\{C_j(\phi, T) + D_j(\phi, T)v_t + \phi i \ln S_t\} \tag{33}$$

$$P_j = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \text{Re}\left[\frac{e^{-\phi i \ln K} f_j(\phi, S_t, v_t, T))}{\phi i}\right] d\phi, \text{ for } j = 1, 2 \tag{34}$$

2.4.2 Parameters and Calibration

Unlike the Black-Scholes model, the Heston model involves latent parameters $\Theta = [v_t, \kappa, \theta, \sigma, \rho]$. To estimate those parameters, the least squared error estimation regarding the difference between market prices and the modelled prices of European options can be one of the common methods [17]. Now with the well-defined objective function to minimise, the complex mathematics in option pricing by the Heston model limits the possibility of an exact and analytical solution by calculus for the parameter estimation. Alternatively, some machine learning techniques, such as gradient descent methods, can be used to obtain an approximate solution. The gradient descent method and the calibration details of the Heston model will be covered in **2.8 Gradient Descent Methods** and methodology later.

$$\begin{aligned} \text{Loss}(\Theta) &= \sum_{i=1}^{N} \left(C_{\text{Heston}}(K_i, T_i; \Theta) - C_{\text{market},i}\right)^2 \\ \text{where } \Theta &= [v_t, \kappa, \theta, \sigma, \rho] \end{aligned} \tag{35}$$

As a machine learning model will require data of as many as possible different paths of price and volatility to learn how to tackle the underlying problem with a specified objective. Historical data can only provide a single realised paths and simulation by resampling from historical return to cannot account for correlated evolution of the stochastic price and stochastic volatility. A calibrated Heston model can simulate possible paths of stochastic price and stochastic volatility of the underlying instrument, besides calculating pricing of the European calls and puts, making the Heston model a suitable candidate to model the stochastic volatility in this thesis.

The simulated paths will be used as the training data to train the machine learning models, and the semi-closed-form option pricing will determine the costs of hedging instruments and hence the associated transaction costs in every time step of the simulated paths. More details related to the machine learning models will be discussed in the following sections **2.5**, **2.6** and **2.7**.

2.4.3 Limitations

While the Heston Model can improve upon the assumption of constant volatility of the Black-Scholes model by introducing stochastic volatility, it still has assumption on log-normal distribution of price of underlying instrument, continuous and frictionless trading and financing.

Second, compared to the Black-Scholes model, the Heston model requires calibration and has more complex calculations, making it less accessible than the Black-Scholes model. The model calibration and option pricing calculation are highly dependent on access to training data and computation capacity, making it a more costly alternative. The calibration requires a large size of data to result in a more robust set of parameters.

Lastly, the Heston model only provides a semi-closed-form solution for European calls and puts, but there are no semi-closed or closed-form solutions for the Greeks due to its mathematical complexity. Instead, approximation methods such as the finite difference are needed for the Greek calculation. Therefore, practitioners may still prefer the Black-Scholes model as a guideline for delta and Greek hedging.

In spite of the limitations of the Heston model, it models stochastic volatility for more realistic market simulation and yields a semi-closed-form solution to price in stochastic volatility in European calls and puts. As this paper focuses on seeking machine-learning-based alternatives to delta and Greek hedging, the contribution of Heston model to a more realistic data simulation in training should be sufficient.

2.5 Hedging based on Neural Network

This section introduces the structure, working principle, and limitations of neural network and different variants, and then elaborates how an appropriate neural structure can align with the hedging problem in the realistic market settings for the aim of this thesis for a possible alternative to Greek hedging with machine learning tools.

With the recent advances of machine learning, researchers have studied its possible applications in different aspects. In 2018, Buehler and his fellow researchers released a research paper "Deep Hedging" to discuss the possibility of using neural networks in hedging financial liability and risk exposure. They illustrated the idea with some semi-recurrent neural network models that input the current market information and the previous positions of hedging instruments in each time step to optimise a hedging strategy under a set of real-life constraints to determine the sequential hedging actions with a predefined objective function as either risk measure or utility (in **2.7**) [5]. With its comprehensiveness, this article acts as a core reference in the construction of the hedging strategy based on the recurrent neural network in this thesis.

Regarding the hedging problem of this thesis, a suitable neural network structure should input current market information such as the price and volatility of the underlying instrument and other risk factors, the time to expiration of the risk exposure or financial liabilities, and any appropriate and accessible market data, and then it should determine the sequential positions of the hedging instruments to minimise the risk [5]. As hedging involves sequential actions over time, the network should be able to handle dynamic sequential data and incorporate transactions and other constraint in the path-dependent decision to formulate the optimal decisions. To determine how to construct a suitable neural network model for this study, the working mechanism of neural networks and variants should be reviewed and compared.

2.5.1 Neural Network and Variants

Neural network $\mathcal{NN}_k(x)$ is a computational model inspired by the structure of the biological neural system to recognise patterns and process complicated data [6], [10]. A neural network model consists of one or multiple layers of interconnected neurones, where each neurone processes the input using weighted sum and non-linear activation function $A(\cdot)$ to generate output for the next layers [6], [10]. As the layer of neurones is mainly to transform the input data without interpretability, they are called hidden layers. In short, a neural network is an application of basis expansion of input features to significantly higher number of features in the hidden layer(s) combined with non-linear activation functions. This typical and basic neural network structure is also called a feedforward neural network to distinguish itself from other neural network variants.

Similar to classic regression models, neural networks require a large and representative dataset to achieve accuracy and robustness. Due to the substantially greater number of parameters inside the hidden layers, neural networks place an even higher demand on the size of the data. As a result, the computational complexity increases with the size of the data, the number of hidden layers and hidden features, and other factors, together called the hyperparameters, in the construction of a neural network. Therefore, balancing accuracy and computational complexity during the construction is an important task and is commonly done by experimenting with different set of hyperparameters and comparing the results.

In mathematics, with input $\mathbf{x} \in \mathbb{R}^{d_{input}}$ and output $\hat{\mathbf{y}} \in \mathbb{R}^{d_{output}}$for a neural network with k hidden layers which has its own number of neurones $d^{(k)}_{hidden}$, activation function $A^{(i)}(\cdot)$, weights $W^{(i)}$ and intercept $b^{(i)}$ for $i = 1, \dots, k$, has number of parameters N_p [6], [10]:

$$\mathbf{z}^{(1)} = A^{(1)}(\mathbf{x}\mathbf{W}^{(1)} + \mathbf{b}^{(1)}) \tag{36}$$

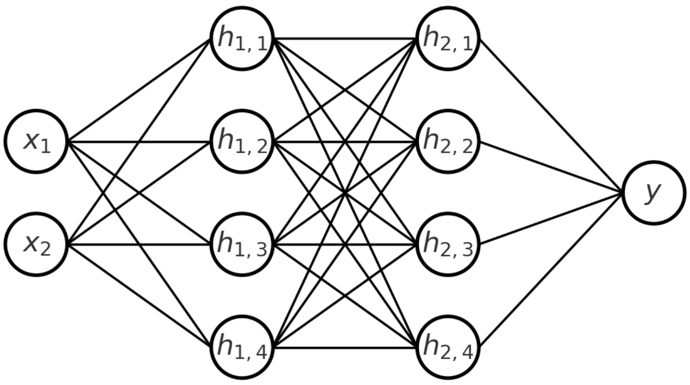
$$\mathbf{z}^{(k)} = A^{(k)}(\mathbf{z}^{(k-1)}\mathbf{W}^{(k-1)} + \mathbf{b}^{(k-1)}) \tag{37}$$

$$\hat{\mathbf{y}} = \mathbf{z}^{(k)}\mathbf{W}^{(k+1)} + \mathbf{b}^{(\mathbf{k+1})} \tag{38}$$

$$\Rightarrow \hat{\mathbf{y}} = \mathcal{N}\mathcal{N}_k(x) = A^{(k)}((\dots A^{(1)}(\mathbf{x}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\dots)\mathbf{W}^{(k)} + \mathbf{b}^{(k)})\mathbf{W}^{(k+1)} + \mathbf{b}^{(\mathbf{k+1})} \tag{39}$$

$$N_p = (d_{input} + 1) \times d^{(1)}_{hidden} + \sum_{j=2}^{k} \left[d^{(j-1)}_{hidden} + 1\right] \times d^{(j)}_{hidden} + \left[d^{(k)}_{hidden} + 1\right] \times d_{output} \tag{40}$$

Figure 2.4: Structure of neural network with 2 hidden layers of 4 neurones



Activation functions $A(\cdot)$ in the neural network mentioned before are non-linear functions to enable the model to handle intricate and non-linear dependencies that cannot be easily captured by classic regression models. The common activation functions are sigmoid, re-defined linear unit (ReLU), and hyperbolic tangent functions [6], [10].

Table 2.2: Common Activation Functions in Neural Networks

Activation Function	**Formula**	**Range**
Sigmoid Function	$\sigma(x) = \dfrac{1}{1+e^{-x}}$	$(0, 1)$
Rectified Linear Unit (ReLU) Function	$\text{ReLU}(x) = \max(0, x)$	$[0, \infty)$
Hyperbolic Tangent (tanh) Function	$\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$

Example 2: There are 2 features x_1, x_2 and they are fed into a feedforward neural network of 1 hidden layer of 4 neurones. The hidden layer will use the existing 2 features to generate 4 new hidden features z_1, z_2, z_3, z_4 by 4 linear models and an activation function $A(\cdot)$ which encodes non-linearity. Finally, a linear model will use these 4 hidden features to generate output as follows:

$$[z_1 \quad z_2 \quad z_3 \quad z_4] = [x_1 \quad z_2] \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} & w_{4,1} \\ w_{1,2} & w_{2,2} & w_{3,2} & w_{4,2} \end{bmatrix} + [b_1 \quad b_2 \quad b_3 \quad b_4]$$

$$\hat{y} = [A(z_1) \quad A(z_2) \quad A(z_3) \quad A(z_4)]\,[w_1 \quad w_2 \quad w_3 \quad w_4]^T + b$$

$$N_p = (2+1) \times 4 + (4+1) \times 1 = 17 > 3 \text{ for linear regression model}$$

In real application of neural networks, the number of input features d_{input}, the number of hidden layers k, and the number of neurones $d_{hidden}^{(k)}$ in each layer can be much larger, exponentially increasing the number of learnable parameters (weights and bias terms). By expanding the input features and introducing non-linearity, neural network could explore the possibility of some uninterpretable relationship between the input and the output.

In classic regression, parameters are typically estimated by maximising the likelihood in maximum likelihood estimation and by minimising the sum of squared error in ordinary least squares estimation. Instead of estimation in statistics, machine learning, call the process of adjusting the parameters to improve the performance **learning** and the process of feeding data to the network to learn **training** [6], [10]. To learn the weights and bias terms for each neurone, an objective function, sometimes called loss function and reward function, a metric that evaluates the performance between the target output and the model output, is typically optimised [6], [10]. With the layered structures and activation functions of neural networks, the optimisation is hardly done analytically for a closed solution. Instead, in machine learning, most optimisation is done by gradient descent methods in **2.8**, for example stochastic gradient descent [6], [10]. The training process of a typical neural network involves random initialisation of weights and multiple iterations of forward propagation to pass the data into the hidden layers to generate prediction and back-propagation to update the weights with the gradient descent method in the learning process [6], [10].

Regarding the hedging problem, the simple and static architecture of feedforward neural networks limits the ability and efficiency to handle dynamic sequential data or decision making. One possible solution would involve training multiple neural network models for different time steps, significantly increasing the number of parameters and the computational complexity. Hence, feedforward neural network is not suitable for this study.

Recurrent neural network (RNN) is a type of neural network designed to process and learn from sequential or time series data by its recurrent structure. RNNs pass a hidden state vector $\mathbf{h}_t$, which can be seen as the summarised multi-step memory capturing temporal dependencies, from the previous time step to the next time step, allowing the persistence of past information across time steps [6], [10]. Therefore, the next time step will process the current input and the previous hidden state.

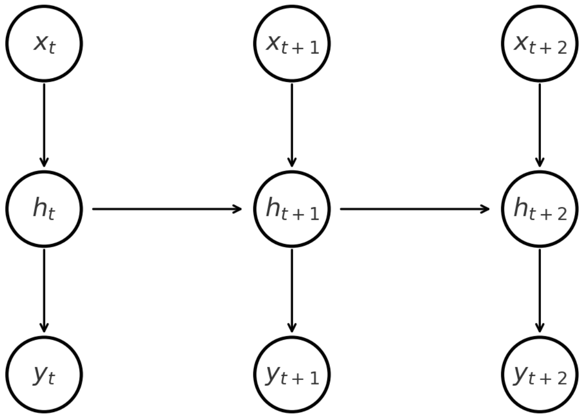
Each time step shares the same sets of weights and intercepts $\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{b}$ and c in the following mathematical expression of the RNN structure [6], [10]. This significantly reduces the number of learnable parameters and the training time in the neural network, compared to feedforward neural network. At the same time, this may increase the under-fitting risk and limit the flexibility of the model to handle non-recurrent relationship.

$$\mathbf{h}_t = A(\mathbf{x}_t\mathbf{W} + \mathbf{h}_{t-1}\mathbf{V} + \mathbf{b}) \in \mathbb{R}^m, \quad \mathbf{h}_0 = \mathbf{0} \in \mathbb{R}^m \tag{41}$$

$$\hat{\mathbf{y}}_\mathbf{t} = \mathbf{h}_t\mathbf{U} + \mathbf{c} \tag{42}$$

$$\Rightarrow \{\hat{\mathbf{y}}_\mathbf{t}\}_{t=1}^T = \mathcal{RNN}(x) = \{\mathbf{h}_t\mathbf{U} + \mathbf{c}\}_{t=1}^T \in \mathbb{R}^T \tag{43}$$

Figure 2.5: Structure of recurrent neural network



The hedging process must make sequential decisions based on current and previous market information and past decisions. RNNs could be a considerable candidate to support such a path-dependent problem. However, RNN has its limitations and complexity. In hedging, it is important to take the current position of hedging instruments as current output in consideration to determine the next action, but RNNs do not pass this information explicitly and only depends the abstract hidden state vector that may encode such information. In addition, as RNN passes the past information carried by the hidden state vector, with a large number of recurrent time steps, the long chain of multiplication with $\mathbf{V}$ and other mathematical operations can result in vanishing or exploding gradients, and hence failure of learning for long-term problems. With the wide range of maturities of financial liabilities and risk exposures, this problematic feature of RNNs can threaten the usefulness of the model.

Semi-RNNs has a mixed architecture of the feedforward neural network and RNN. Unlike a typical RNN, semi-RNN does not pass the hidden state from the previous step into the network to generate the output [5], [6]. Instead, the output from the previous step as a one-step memory is explicitly fed into the feedforward neural network with the data in the current step. It does not propagate the abstract and uninterpretable hidden state but the previous output to pass the past information into the network, making this architecture semi-recurrent as following:

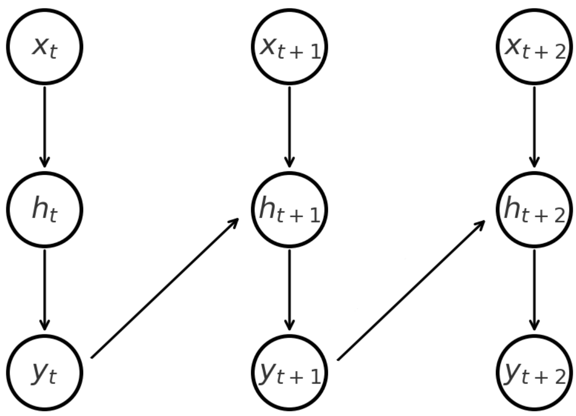
$$\mathbf{z}_t^{(1)} = A^{(1)}([\mathbf{x_t} \quad \hat{\mathbf{y}}_{\mathbf{t-1}}]\mathbf{W}^{(1)} + \mathbf{b}^{(1)}) \tag{44}$$

$$\mathbf{z}_t^{(k)} = A^{(k)}(\mathbf{z}_t^{(k-1)}\mathbf{W}^{(k-1)} + \mathbf{b}^{(k-1)}) \tag{45}$$

$$\hat{\mathbf{y}}_t = \mathbf{z}_t^{(k)}\mathbf{W}^{(k+1)} + \mathbf{b}^{(\mathbf{k+1})} \tag{46}$$

$$\{\hat{\mathbf{y}}_t\}_{t=1}^T = \mathcal{SRNN}(x) = \{\mathcal{NN}_\theta(\mathbf{x}_t, \hat{y}_{t-1})\}_{t=1}^T \in \mathbb{R}^T, \quad \mathbf{y_0} = 0 \tag{47}$$

Figure 2.6: Structure of semi-recurrent neural network



For the hedging problem, the structure of semi-RNN can input time-series data such as current market information and explicitly feed back the previous positions of the hedging instruments to provide the current position of the hedging instruments, accounting for the sequentiality of decision making in real situation. Semi-RNNs maintain the strength of processing sequential or time-series data and avoids the vanishing and exploding potential of hidden states. In addition, without hidden states and the parameters to generate them, the computational complexity is significantly lower. Overall, semi-RNNs can provide an ideally robust and simpler model aligning the hedging problem in this thesis.

2.5.2 Limitations of Neural Network

Although machine learning techniques such as neural networks have proven their success in handling complicated problems, there are certain shortcomings that cannot be neglected. With a large number of hidden parameters in the architecture of hidden layers, training neural network models is typically computationally intensive and costly, which requires computational resources and time [6], [10]. In order to train a well-performing model, a large size of high-quality data usually plays a key role, further increasing the computation workload and cost. On the other hand, if the data lack representativeness and diversity, the model may suffer overfitting problem just like classic regression models. Therefore, out-of-sample performance of the models should be reviewed.

By construction, neural networks learn to optimise a specified objective function. Their predictive power and generalisation are therefore strongest in the specialised context of the tasks aligned with that objective. If the task slightly deviated from the objective, the models probably do not perform satisfactorily. Even when trained models can deliver desirable performance, they often lack interpretability, making it difficult to explain the underlying relationships in an analytical manner. Therefore, users should be cautious about the fact that neural network models only perform tasks by design instead of providing an explanation of a problem.

Regarding the hedging problem, trained models are usually highly subject to the objective function and market setup such as characters of the risk exposure, set of hedging instruments, hedging frequency according, interest rate and transaction cost. If the setting is more specific, less information is needed for input, and the models can probably perform better in the specific task. In contrast, if the setting is more general, more information has to be provided as input, and the generalised models may not perform as good as specialised for certain tasks as the models may have to comprise certain performance to boost the overall performance.

2.6 Hedging based on Q-Learning

Besides neural networks, a researcher Halperin suggested another machine learning technique called Q-learning to hedging financial risk [9]. Q-learning is a reinforcement learning (RL) algorithm designed to solve sequential decision-making problems under uncertainty, which aligns the nature of the hedging problem. Instead of learning the parameters, it learns a policy as a rule that decide which action to take in each state to optimise the expected cumulative reward measure by a function called **Q-function** $Q(s, a)$ [19]. The training involves sequential evaluation of actions a from the action space $\mathcal{A}$ interacting with the current state s from the state space $\mathcal{S}/$, with a predefined reward function that calculates the reward of action a in state s, $R(s,a)$, and eventually formulate **an optimal policy** with optimal Q-function $Q^*(s,a)$ [19]. Therefore, the optimal action in each state is identified among all possible given actions. A continuous action space contains infinite actions to evaluate, making the process practically impossible. Hence, action space is set to be finite and discrete.

Definition 10. Q-function is the expected cumulative reward of action a in current state s under a policy π that decides the future actions in future states, with a discount factor $0 \leq \gamma \leq 1$ for the rewards R_t [19].

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \middle| S_0 = s, A_o = a, \pi\right] \tag{48}$$

Definition 11. Bellman Optimality Equation An equation in the dynamic programming for optimality that states the value $V^*(s)$ of current state s is the sum of immediate reward $R(s,a)$ and the expected value of future states s' from state space $\mathcal{S}$ with conditional probability $\mathbb{P}(s'|s,a)$ discounted by discount factor $0 \leq \gamma \leq 1$, assuming optimal action now a from action space $\mathcal{A}$ and going forward, as following [19]:

$$V^*(s) = \max_{a \in \mathcal{A}} \left[R(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a) V^*(s')\right] \tag{49}$$

Definition 12. Optimal Q-function: A Q-function that satisfies the Bellman optimality equation as following [19]:

$$Q^*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q^*(s',a') \middle| S_t = s, A_t = a\right] \tag{50}$$

The training process starts with initialisation of assigning random values or guessing values to each combination of states and actions, i.e. the number of Q-values N_Q is the multiple of the number of states N_s and the number of actions N_a. For a state and an action, a Q-value is calculated according to **Equation (50)** such as $R_{t+1} + \gamma \max_{a'} Q(s',a')$ and compared to the current value. Then the Q-value of this given state and action is updated by the difference factored by a learning rate $\eta > 0$ as following:

$$Q(s,a) \leftarrow Q(s,a) + \eta\left[r + \gamma \max_{a' \in \mathcal{A}} Q(s',a') - Q(s,a)\right] \tag{51}$$

After a large number of iterations of exploring possible combination, the Q-values should converge to the optimal Q-values for all combination of states and actions.

Regaring to the hedging problem, the state should include the market situation such as the price and volatility of the underlying instrument, the current positions of hedging instruments, and the time to expiration; the action should be different discrete trading decision of each hedging instruments; and the reward should be the negative of a risk measure at T so risk is minimised when reward is maximised [9]. Also, discount factor should be set to be 1 so the final risk measure is not discounted.

2.6.1 Limitation

In general financial and hedging problems, trading decisions are often continuous, therefore, the discrete action space may result in a suboptimal decision. To create an action space to approximate the truly optimal policy, it will require a higher number of finely discrete actions for each instrument. With multiple hedging instruments, the action space grows exponentially. Combining the size and dimension of the state space with price, volatility, positions, and time to expiration, the number of Q values $Q(s, a)$ becomes even greater, making training **computationally intensive and challenging**. To solve this problem, a technique called the deep Q-network (DQN) is introduced to use the state as input of a neural network to estimate the Q-value of all actions, significantly reducing the number of parameters to learn [15].

According to **Example 3**, despite the number of parameters to learn is significantly reduced, the computational intensity is still challenging when action space gets larger for higher-dimension decision making. In order to determine the feasibility of the Q-learning model in the hedging problem, a simple test case should be first reviewed by the computational time and performance on risk reduction with the given computational capacity and time restraint.

Example 3: A hedging problem involves 3 hedging instruments and 10 time steps. The action level for each hedging instrument is set to be (-1, 0, 1) as to sell 1 unit, to do nothing, and to buy 1 unit. The state space includes 100 possible price levels and the time to expiration. The number of parameters to learn with classic Q-learning and DQN with one hidden layer of 64 neurons should be:

$N_a = 3^3 = 27$

In classic Q-learning, $N_Q = N_a \times N_s = 27 \times (100 \times 30) = 81,000$

With DQN, $N_p = (d_{in} + 1) \times d_{hidden} + (d_{hidden} + 1) \times d_{output} = (2 + 1) \times 64 + (64 + 1) \times 27 = 1,947$

If the size of action levels for each hedging instrument is doubled to 6,

$N'_a = 6^3 = 216$

$N'_Q = N'_a \times N_s = 216 \times (100 \times 30) = 375,000$

$N'_p = (d_{in} + 1) \times d_{hidden} + (d_{hidden} + 1) \times d_{output} = (2 + 1) \times 64 + (64 + 1) \times 216 = 14,232$

,

2.7 Objective Function: Risk Measures

In the previous sections, training a neutral network model and Q-learner model involves adjusting the parameters of the linear models within the network and Q function by optimising a defined objective function. The objective function should capture the difference between the actual or target values and the predicted values. Despite the absence of definite actual or target values in the hedging problem, hedging itself has a well-defined target of minimising risk. Therefore, the loss function for hedging should be some risk measure based on the output of the neural network, and then the target value can simply be set to be the lower bound of the risk measure.

2.7.1 Coherent Risk Measure

In finance, there is a wide range of risk measures to capture uncertainty in financial markets. One of the typical measures for uncertainty and variation in statistics is standard deviation (volatility). As it captures the variation in both directions, reducing volatility means reducing the possibility and magnitude of both negative outcomes and positive outcomes, while risk management should focus on lowering the risk associated with negative outcomes. In 1999, Artzner and his fellow researchers purposed a set of properties for risk measures of financial risk that aligns financial and risk management principles and called such risk measures coherent risk measures [1], [14], acting as a widely adapted guideline for financial risk measures.

Definition 13. Coherent Risk Measure: a risk measure $\rho : \mathcal{X} \to \mathbb{R}$ for payoff of a financial position $X, X_1, X_2 \in \mathcal{X}$ is a coherent risk measure if it shows all the following properties [1], [14]:

(a) **Monotonicity**: A position with lower payoff in every state is riskier.
If $X_1 \leq X_2$, then $\rho(X_1) \geq \rho(X_2)$

(b) **Translation invariance**: Adding cash m reduces risk by c.
For all $m \in \mathbb{R}$, $\rho(X + m) = \rho(X) - m$

(c) **Positive homogeneity**: Scaling the positive by m scales the risk proportionally.
For all $\lambda \geq 0$, $\rho(\lambda X) = \lambda \rho(X)$

(d) **Subadditivity**: Diversification should not increase risk.
For all X_1, X_2, $\rho(X_1 + X_2) \leq \rho(X_1) + \rho(X_2)$

2.7.2 Conditional Value-at-risk (CVaR)

With the above criterion by Artzner, financial institutions and regulators can access the suitability of difference risk measure for financial risks. In 2010, the Basel Committee on Banking Supervision published BASEL III, a regulatory framework, which required regulated financial institutions to implement and monitor a coherent risk measure **conditional value-at-risk (CVaR)** [1], known as expected shortfall, of their assets [2]. Since then, CVaR has been widely used as a risk measure in the financial industry.

CVaR is the expected loss conditional on the loss exceeding the threshold called value-at-risk (VaR) [1], [3], [14]. VaR is the minimum loss that the loss of a financial position may not exceed with the probability of $0 \leq \alpha \leq 1$, in short, VaR can be considered as the α-th percentile of the loss [1], [3], [14]. As the probability α determines the value of VaR, VaR and CVaR always presented with the probability α as α% VaR and $\alpha\%$ CVaR. In addition to measuring the minimum loss, CVaR represents the expected value of loss greater than the minimum loss so it can capture the tail shape, in other words, the distribution of loss beyond the threshold VaR [1], [3], [14]. With its popularity in financial risk management and its ability to capture tail risk, this thesis uses CVaR of the loss at maturity T to be the objective function for the neural network and Q-learner models.

Definition 14. Value-at-risk (VaR): For $0 < \alpha < 1$, let the X be the payoff of a financial position, so $L = -X$ be its loss. Then VaR is calculated as [3], [14]:

$$\mathrm{VaR}_\alpha(L) = \inf\{\ell \in \mathbb{R} : \mathbb{P}(L \leq \ell) \geq \alpha\} \tag{52}$$

Definition 15. Conditional Value-at-risk (CVaR): For $0 < \alpha < 1$, let the X be the payoff of a financial position, so $L = -X$ be its loss. Then CVaR is calculated [3], [14], [16]:

$$\begin{aligned}
\mathrm{CVaR}_\alpha(L) &= \mathbb{E}[L | L \geq \mathrm{VaR}_\alpha(L)] && (53)\\
&= \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_u(L)du && (54)\\
&= \inf_{\eta\in\mathbb{R}} \left\{\eta + \frac{1}{1-\alpha}\mathbb{E}\Big[(L-\eta)_+\Big]\right\} \quad \text{(Rockafellar-Uryasev formulation)} && (55)
\end{aligned}$$

Proof for CVaR as a coherent risk measure [1], [14]:

1. **Monotonicity**: For loss $L_1 \leq L_2$,

$$\begin{aligned}\mathrm{CVaR}_\alpha(L_1) &= \inf_{\eta\in\mathbb{R}}\left\{\eta + \frac{1}{1-\alpha}\mathbb{E}\Big[(L_1-\eta)_+\Big]\right\}\\ &\leq \inf_{\eta\in\mathbb{R}}\left\{\eta + \frac{1}{1-\alpha}\mathbb{E}\Big[(L_2-\eta')_+\Big]\right\}\\ &= \mathrm{CVaR}_\alpha(L_2)\end{aligned}$$

2. **Translation invariance**: For adding cash $m > 0$ to a financial position,

$$\begin{aligned}\mathrm{CVaR}_\alpha(L-m) &= \inf_{\eta\in\mathbb{R}}\left\{\eta + \frac{1}{1-\alpha}\mathbb{E}\Big[(L-m-\eta)_+\Big]\right\}\\ &= \inf_{\eta'\in\mathbb{R}}\left\{\eta' + \frac{1}{1-\alpha}\mathbb{E}\Big[(L-\eta')_+\Big]\right\} - c \quad \text{for } \eta = \eta' - m\\ &= \mathrm{CVaR}_\alpha(L) - m < \mathrm{CVaR}_\alpha(L) \Rightarrow \text{Risk is reduced by adding cash}\end{aligned}$$

3. **Positive homogeneity**: For $\lambda \geq 0$,

$$\begin{aligned}\mathrm{CVaR}_\alpha(\lambda L) &= \inf_{\eta\in\mathbb{R}}\left\{\eta + \frac{1}{1-\alpha}\mathbb{E}\Big[(\lambda L-\eta)_+\Big]\right\}\\ &= \lambda \inf_{\eta'\in\mathbb{R}}\left\{\eta' + \frac{1}{1-\alpha}\mathbb{E}\Big[(L-\eta')_+\Big]\right\} \quad \text{for } \eta = \lambda\eta'\\ &= \lambda\mathrm{CVaR}_\alpha(L)\end{aligned}$$

4. **Subadditivity**: For loss L_1 & L_2,

$$\begin{aligned}\mathrm{CVaR}_\alpha(L_1+L_2) &= \inf_{\eta\in\mathbb{R}}\left\{\eta + \frac{1}{1-\alpha}\mathbb{E}\Big[(L_1+L_2-\eta)_+\Big]\right\}\\ &\leq \inf_{\eta_1\in\mathbb{R}}\left\{\eta_1 + \frac{1}{1-\alpha}\mathbb{E}\Big[(L_1-\eta_1)_+\Big]\right\} + \inf_{\eta_2\in\mathbb{R}}\left\{\eta_2 + \frac{1}{1-\alpha}\mathbb{E}\Big[(L_2-\eta_2)_+\Big]\right\}\\ &= \mathrm{CVaR}_\alpha(L_1) + \mathrm{CVaR}_\alpha(L_2)\end{aligned}$$

2.7.3 Calculation of CVaR

The main goal of this study is to compare the performance of traditional Greek hedging and machine-learning-based hedging algorithm in a more realistic setting with transaction costs and stochastic volatilities. Therefore, it is important to encode transaction costs and volatility in the CVaR calculation. As transaction costs occur whenever a trade is executed, continuous hedging activities will accumulate transaction costs affecting the intermediate cash balance and the overall profit and loss (PnL) at maturity T. In particular, the intermediate cash balance can generate interest income or expense over time. In addition to the stochastic price and volatility, the overall profit and loss must account for the path dependence of the hedging problem.

Therefore, at each time step, the cost of the hedging instruments H_t, the associated transaction cost C_t, the interest income or expense I_t, and the cash balance M_t must be carefully tracked. In order to keep the prices of hedging instruments consistent with the stochastic volatility setting, all options should be priced by the calibrated Heston model at each time step. At maturity T, the total payoff of the hedging instruments Π_T^{Hedge} and the payment of the risk exposure or liability $\Pi_T^{Liability}$ are calculated to determine the overall profit and loss [5]. Finally, with the overall profit and loss of each simulated path, the objective CVaR function can be calculated. For simplicity, a proportional transaction cost c is used.

$$\text{Change in position } \Delta N_{i,t}^{Hedge} = N_{i,t}^{Hedge} - N_{i,t-1}^{Hedge} \tag{56}$$

$$\text{Cost of hedging instruments } H_t = \sum_i P_{i,t}^{Hedge} \Delta N_{i,t}^{Hedge} \quad \text{where } P_{i,t}^{Hedge} \tag{57}$$

$$\text{Transaction cost } C_t = c \sum_i \left| P_{i,t}^{Hedge} \Delta N_{i,t}^{Hedge} \right| \tag{58}$$

$$\text{Interest income } I_t = M_{t-\Delta t} r \Delta t \tag{59}$$

$$\text{Cash balance } M_t = M_{t-\Delta t} - H_t - C_t + I_t \tag{60}$$

$$\text{Total payoff of hedging instruments } G_T = \sum_i \Pi_{i,T}^{Hedge} N_{i,T-\Delta t}^{Hedge} \tag{61}$$

$$\text{Final PnL } X = -L_T + M_{T-\Delta t} + \Pi_T + I_T \tag{62}$$

where $N_{i,t}^{Hedge}$, $P_{i,t}^{Hedge}$ and $\Pi_{i,t}^{Hedge}$ are position, price and payoff of hedging instrument i at t

With final profit and loss $\mathbf{X} = \{X_j\}_{j=0}^n$ and $\mathbf{L} = -\mathbf{X}$ for n simulated paths of price and volatility of the underlying instrument, for $\widehat{\mathrm{VaR}}_\alpha(\mathbf{L})$ as the α-quantile of $\{L_j\}_{j=0}^n$

$$\widehat{\mathrm{CVaR}}_\alpha(\mathbf{L}) = \widehat{\mathrm{VaR}}_\alpha(\mathbf{L}) + \frac{1}{n(1-\alpha)} \sum_{j=1}^{n} \left[L_j - \widehat{\mathrm{VaR}}_\alpha(\mathbf{L}) \right]_+ \tag{63}$$

2.8 Gradient Descent Methods

After the machine learning models are introduced in the previous section, this section will explain the underlying machine learning technique called **gradient descent methods** to train the model with the well-defined objective function. Even though the in-depth understanding of gradient descent methods are one of the essential knowledges to understand this thesis, it is helpful to glean how the parameters are updated in the optimisation of the objective function.

The gradient descent methods are to iteratively update the parameters $\theta^{(t)}$ **in the direction opposite** to the gradient of the objective function **to minimise** as the partial derivative $\frac{\partial Loss(\theta)}{\partial \theta}$ with learning rate γ_t in order to find the values of the parameter with zero or sufficiently small gradient for minimum loss [6], [10]. As a positive gradient means dependent variable *Loss* increases with the independent variable θ and a negative gradient means the dependent and independent variables move in different direction, the adjustment of the parameters should be in the opposite to reach a lower value of objective function [6], [10].

Parameter update by gradient descent [6], [10]:

$$\theta^{(t+1)} = \theta^{(t)} - \gamma_{t+1} \frac{\partial Loss(\theta^{(t)})}{\partial \theta} \tag{64}$$

This method requires computing the gradient with the entire data set with a long computational time and does not guarantee that the minimum found is global minimum, as local minima can have zero gradient when the loss function is not strongly convex with respect to the parameters.

To address this, an alternative, **Stochastic Gradient Descent (SGD)**, approximates the gradient by bootstrapping the dataset (using a random mini-batch) to shorten the computational time [6], [10]. Moreover, introduction of randomness can lower the chance that the parameters get stuck at the local minimum where update of parameters are limited by the very small gradients around the local minimum [6], [10].

Parameter update by stochastic gradient descent (SGD) [6], [10]:

$$\theta^{(t+1)} = \theta^{(t)} - \gamma_{t+1} \frac{\partial Loss_{\mathbf{B_t}}(\theta^{(t)})}{\partial \theta} \tag{65}$$

where $Loss_{\mathbf{B_t}}(\theta^{(t)})$ is the loss function calculated with only the data in mini-batch $\mathbf{B_t}$.

To improve the stability of the result from stochastic gradient descent, the variable adaptive moment estimation (ADAM) with momentum and adaptive learning rates is introduced by maintaining exponentially decaying averages of past gradients (first moment) and squared gradients (second moment) by decay factor β_1 and β_2, respectively. In addition to stability, ADAM provides a faster computation based on the adaptive learning rate adjusted by $\sqrt{\hat{v}_t}$.

Parameter update by Adaptive Moment Estimation (ADAM) [13]:

$$\theta^{(t+1)} = \theta^{(t)} - \gamma_{t+1} \frac{\hat{m}_t}{\sqrt{\hat{v}_t}} \tag{66}$$

where

$$g_t = \frac{\partial Loss_{\mathbf{B_t}}(\theta^{(t)})}{\partial \theta}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

β_1 and β_2 are typically set at 0.9 and 0.999.

Even though gradient descent methods can yield approximate solutions with gradients, the calculation of gradients can be challenging for optimisation problems with complex structure objective function with respect to the parameters. Fortunately, modern algorithms have adapted the chain rule from calculus to handle the mathematical complexity in gradient calculation, making gradient descent methods more practical and reliable. Therefore, complicated optimisation problems such as minimising the sum of squared error in calibration of the Heston model, minimising CVaR of loss in the training of semi-recurrent neural network model, and maximising reward function in the Q-learning model can be solved with gradient descent methods. For a stable and computationally efficient result, this study will implement ADAM.

3. METHODOLOGY

This section outlines the methodology for developing and evaluating Greek hedging and hedging strategies based on machine learning techniques such as neural network and Q-learning to minimise Conditional Value at Risk (CVaR). It begins with sources, specification, and processing of **(3.1) data** in this study. It continues with the experiment settings for realistic market situation with the fundamental **(3.2) market settings** and the **(3.3) Heston model** that was calibrated and used in the simulation of the price and volatility of S&P 500 Index. Then it describes the **(3.4) hedging** scenarios included in the experiment and how **(3.5) risk measure CVaR** was calculated to evaluate different approaches. Following all the experimental setup, the section will present the specification of **(3.6) Greek hedging**, **(3.7) neural network model** and **(3.8) Q-learning model** respectively, and finished with **(3.9) data analysis** and **(3.10) a summary of experiment**.

3.1 Data

3.1.1 Data Processing Software

The data processing, simulation, and machine learning algorithms of this study were carried out with original code in Python 3.12, mainly supported with the following packages: **Numpy** for data manipulation; **Pytorch** for data manipulation and optimisation; **TorchQuad** for integration in option pricing by the Heston model; **Scipy** for matrix manipulation; **yfinance** for historical financial data from Yahoo! Finance, and **Panda** for input and output data.

3.1.2 Historical Financial Data

S&P 500 Index is one of the well-known equity indexes and there is a large market for S&P 500 Index exchange-traded funds and derivatives of the SP 500 Index with high liquidity and accessibility. For simplicity, all risk exposure and hedging instruments have underlying instrument as S&P 500 Index and matching life time to the risk exposure. Historical data of the SP 500 Index, including the close price of the index and the data of its European calls and puts, including the last transaction price, maturity dates, strike prices, open interest (total number of active contracts of an option), and last trading date were collected on 15 August 2025. The options data were then filtered by (a) the last trading date less than a week ago to ensure that the last transaction prices could reflect the latest market, (b) open interest above 100 contrast to ensure the options were traded actively and representative, and (c) the time to expiration between 1 month and 15 months to match the time to expiration of the risk exposure in the test cases in **3.4**. After data cleaning, there were 3686 observations for the options. Historical data were used to estimate the risk-free rate r, the dividend yield d, and the drift rate mu of the index, and to calibrate the Heston model for the data simulation used in training the machine learning model and the evaluation of the performance of the hedging strategies.

3.1.3 Simulated Data from the Calibrated Heston Model

With the Heston model calibrated with the above historical data in **3.3** and an initial price S_0 scaled to 100 for simplicity, 100,000 90-day paths of price and volatility of §P 500 Index were simulated and the prices of all the hedging instruments in **3.4** were then calculated at every time step of each path. The data was stored as the master data for training the neural network model and Q-learning model. During each training, a batch of 2,000 paths was randomly drawn and divided into training data, validation data, and testing data according to the next subsection.

3.1.4 Data Split

In machine learning, it is critical to divide the data into 3 batches: training data, validation data, and testing data. The data split for the simulated data from the calibrated Heston model was set to 80-10-10. Training data were used to train the semi-recurrent neural network and Q-learner reinforcement learning algorithms [6], [10]. The validation data was used to check the performance of the trained model in each epoch. The validation loss, as the loss function calculated with the model under training and the validation data, acted as an intermediate out-of-sample performance of the model [6], [10]. Validation could terminate training early to save computational time and prevent overfitting when validation loss did not show a significant improvement of more than 0.0001 for more than 10 epochs. In addition, validation data helped tune hyperparameters such as the total epoch number, the learning rate, the batch size, the optimiser, and the number of hidden layers and neurones. After training and validation, the performance of the model was evaluated with other out-of-sample data, i.e., testing dat [6], [10].

3.2 Market Setting

From the literature review, some key market parameters are involved in multiple calculations such as pricing, Greek calculation, profit and loss, risk measure, and model design. In order to have a realistic and practical result from the hedging strategies, it is important to have a realistic market setting. For instance, risk-free rate r, dividend yield q, the drift rate of underlying instrument μ, and the proportional transaction cost c.

3.2.1 Risk-Free Rate r and Dividend Yield q

The risk-free rate is a theoretical rate of return of an investment without market risk, no credit risk, no liquidity risk, and no reinvestment risk. In short, it should be a constant rate without any risks. In real market situation, interest rates are subject to exchange rate risk, credit risk, liquidity risk, reinvestment risk, etc. Practitioners usually take the possibly lowest and safest interest rate in the markets as a proxy of the risk-free rate. It could be some overnight interbank interest rates measured by central banks or some central financial organisation, for example, the Sterling Overnight Index Average (SONIA) by the Bank of England and the Secured Overnight Financing Rate (SOFR) by the Federal Reserve Bank of New York. In spite of the low-risk nature of those benchmarks, they fluctuate over time due to macroeconomic factors like monetary policy, market stress, money supply and demand, etc, violating the definition of risk-free rate. Also, using a proxy risk-free rate means an additional stochastic model to simulate the market situation, increasing the complexity and the efficiency of the algorithms.

To achieve a reasonable balance between the fidelity of the model and the efficiency, the risk-free rate r was set to be constant. With the Call-Put Parity and market data for calls and puts of S&P 500 Index, implied continuous risk-free rate and the continuous dividend yield were estimated with the ADAM gradient decent method with loss function below. With historical data from Yahoo! Finance, the estimated risk-free rate $\hat{r}$ and the dividend yield $\hat{q}$ were 4.2874% and 1.4317%, respectively.

Call-put parity:

$$C_0(K_i, T_j) - P_0(K_i, T_j) = S_0 e^{-qT_j} - K_i e^{-rT_j} \Rightarrow \tag{67}$$

$$\text{Loss}(\hat{r}, \hat{q}) = \sum_{i,j} C_0(K_i, T_j) - P_0(K_i, T_j) - \left(S_0 e^{-\hat{q}T_j} - K_i e^{-\hat{r}T_j}\right) \tag{68}$$

3.2.2 Drift Rate of Underlying Instrument: S&P 500 Index

With historical data from Yahoo! Finance, the daily log-return of S&P 500 Index between 15 August 2020 and 15 Aug 2025 was calculated and then annualised with the basis of the count of trading days 252 days per year to be 13.0139%.

3.2.3 Proportional Transaction Cost c

As the primary incentive of this study, the proportional transaction cost c is critical to encode the friction of the financial market in the hedging problem. In order to investigate the performance of different hedging strategies in the presence of friction as transaction cost, three testing cases such as 0%, 0.01% and 0.1% were considered. The same proportional transaction cost was applied to the absolute value of the trading notional of each hedging instrument.

3.2.4 Self-Financing Setting

Besides the above parameters, the hedging strategy was set to be self-financing, in other words, there is no injection of capital at any time after the risk exposure or liability is acquired. Any cash surplus or deficit due to the trading of hedging instruments should be deposited or borrowed at a risk-free rate r. This constraint can ensure that all the position of hedging instruments are financed from the proceed from the trading activities with the hedging instruments only, including cash and risk-free equivalence, so no additional profit or loss due to abnormal positions of hedging instruments with injection or withdrawal of external capital, stabilising the CVaR calculation and hence the hedging models. For simplicity, the cash balance at $t = 0$ was set to be the cash proceeds from obtaining the risk exposure or liability; therefore, its price at $t = 0$. Then, the profit and loss in each simulation and the CVaR reflected the net profit and loss at $t = T$ as mentioned in **2.7**.

3.3 Price and Volatility Simulation based on Heston Model

3.3.1 Calibration

With the estimated risk-free interest rate $\hat{r} = 4.2874\%$, the estimated dividend yield $\hat{q} = 1.4317\%$ of the S&P 500 Index, the historical data in **3.1.1**, including the market price, strike price and the time to expiration of European calls and puts, a Heston model was optimised with the objective function (35) as the total sum of squared error between the observed market price and the modelled price of European call and put. Due to the complexity of semi-closed-form solution of pricing option by Heston Model. The optimisation problem did not have an easily computed estimation solution or gradients. The calibration employed a gradient descent methods adaptive moment estimation (ADAM) included in a Python package Pytorch. In addition, the scheduler, which decreases the learning rate when the improvement of the loss function is less than a threshold or the loss function is close to its minimum, was also used to improve the efficiency and performance of the trained model [18]. The calibrated result gave parameters: $\hat{\kappa} = 1.9216, \hat{\theta} = 0.0442, \hat{\sigma} = 0.2905, \hat{V_0} = 0.0157$ and $\hat{\rho} = -1$.

3.3.2 Data Generation

To simulate the stochastic price $S_{t+\Delta t}$ and volatility $v_{t+\Delta t}$ with estimated correlation $\hat{\rho}$ and other estimated parameters of the Heston model, for each time step, two sets of independent standard normal random variables $Z_{1,t}$ and $Z_{2,t}$ were first generated and then transformed into ΔW_t^s and ΔW_t^v as below:

$$
\begin{aligned}
\Delta t &= T/N \\
Z_{1,t}, Z_{2,t} &\sim Normal(0,1), \text{ independent} \\
\Delta W_t^s &= Z_{1,t}\sqrt{\Delta t} \\
\Delta W_t^v &= \left(\hat{\rho} Z_{1,t} + \sqrt{1-\hat{\rho}^2} Z_{2,t}\right)\sqrt{\Delta t}
\end{aligned}
$$

Then, we can calculate the prices and volatilities over time in each path.

$$S_{t+\Delta t} = S_t \exp\left\{(\hat{\mu} - q - \frac{v_t}{2})\Delta t + \sqrt{v_t}\Delta W_t^S\right\} \tag{69}$$

$$v_{t+\Delta t} = v_t + \hat{\kappa}(\hat{\theta} - v_t)\,\Delta t + \hat{\sigma}\sqrt{v_t}\,\Delta W_t^v \tag{70}$$

Before simulation, the computational time for different time horizons T and the number of steps N was assessed. The time should cover the lifetime of the risk exposure we aim to hedge, and the number of steps should match with hedging frequency. For flexibility in hedging frequency from daily to monthly, Δt is set to 1 day. Within the time limit of this study, the time horizon T and the number of time steps were eventually set to 90 days and 90 respectively, and then the number of paths simulated was 100,000 as a result of balancing between accuracy and computing time. As the scale of the price does not affect the actual comparison of the performance of different hedging approaches, the price S_0 is normalised to be 100.

3.3.3 Pricing of Derivatives

All the hedging instruments and financial instrument were priced with the calibrated Heston model to narrow down the gap between the experiment setting and real market situation.

3.4 Hedging

In this study, the three hedging strategies such as (1) Greek hedging, (2) neural-network-based hedging, and (3) Q-learning-based hedging were experimented and compared, with a benchmark of no hedging action. For simplicity, as mentioned at the beginning, the hedging process of this study focused on the market risk which is driven by market movements and used financial liabilities with underlying instrument as S&P 500 Index as an illustrative examples to evaluate the risk measure CVaR of each hedging strategy with the simulated price and volatilities by the calibrated Heston model in **3.3**.

3.4.1 Hedging Frequency

The hedging frequency determines the trading frequency and hence indirectly impacts the total transaction costs and final profit and loss. The continuous trading assumption in the Black-Scholes model and the Heston model in **2.2** and **2.3** is practically violated due to trading hour and cost inefficiency. Frequent rebalancing on hedging instrument can ensure the risk exposure is managed properly according to the latest market conditions, especially in volatile markets, however, this increases trading activities and hence the total transaction costs and over-hedging. On the other hand, infrequent hedging could result intermediate risk exposure exceeding the daily limit imposed by internal control and regulators and undesirable mark-to-market loss.

To address the effect of hedging frequency, different frequencies such as daily, every 2 day, weekly and monthly, were tested to check if any particular hedging strategy would behave better with certain hedging frequency.

3.4.2 Hedging Instruments

In reality, there are a vast number of choices of hedging instruments. Inclusion of all possible hedge instruments in this study is impractical and inefficient. The number of hedging instruments significantly affects the total computational time: the number of variables in Greek hedging, the number of weights in the hidden layers of the neural network, and the action space of Q-learner. Despite of the flexibility provided by larger number of hedging instruments, it could increase the computation complexity exponentially in machine learning and result in an unstable performance. Therefore, prudential selection of hedging instruments becomes critical for the efficiency of the models. It is possible to hedge risk exposure with derivatives with mismatching underlying instrument or time to expiration, especially when matching hedging instruments are not easily accessible. In this study, for simplicity, all hedging instruments had the time to expiration matching the risk exposure / liability, i.e. 90 days. The types of hedging instruments were the following:

Cash can reduce the CVaR value as one of the coherent risk measure properties. Inclusion of cash can also support the self-financing setting as a residual account to absorb gains and losses from all the trading activities and encode interest income or expense over the time horizon, providing the financing consistency and a path-dependent calculation of final profit and loss.

In reality, the interest rates for borrowing $r_B$and lending r_L can be different even if the credit risk is symmetrical, due to the administration fee and other fees in a frictional market, so having different risk-free rates for borrowing and lending does not contradict the concept of risk-free rate. Moreover, a constant risk-free rate exists only in the theoretical setting. For simplicity, a single constant value $\hat{r}$ estimated in **3.2.1** is used.

S&P 500 Index In this study, even though the underlying instrument is S&P 500 Index, an equity index that cannot be traded, fortunately, many exchange traded funds (ETFs) passively track this well-known index and have decent liquidity. Therefore, using the simulated price of S&P 500 Index as price of S&P 500 Index ETF can keep the study close to the reality.

European Call Despite the difference in payoff of European call and put, only one of them needs to be considered under the call-put parity with the presence of risk-free financing and the underlying instrument [12]. Therefore, only calls were included in the set of hedging instruments.

Although put options are no longer necessary due to Call-Put parity, it could be crucial to include call options with multiple strike prices because the Greeks of options with different strike price can have different values and behave differently with movement of the risk factors. The diversity of strike prices makes the hedging strategy more realistic and flexible, yet increases the computational time; therefore, only one out-of-money (OTM) ($K = 0.95S_0 < S_0$), one at-the-money (ATM) ($K = S_0$) and one in-the-money (ITM)($K = 1.05S_0 > S_0$) call options at time t were considered.

Binary Option: Digital Call An additional uncommon derivative, the binary option, was also included in the experiment as a hedging instrument to demonstrate market completeness and as a financial liability to mimic some financial liability with discontinuous payoff. A binary option gives a fixed payoff P when a certain condition is met and can usually be interpreted as a bet [12]. A typical condition is that the price of the underlying instrument S_T on the expiration date T is above or below the predetermined strike price K. In that case, such a binary option is called a digital option.

Price of digital call $C_{digital,t}(K,T)$ and put $P_{digital,t}(K,T)$ [12]:

$$C_{digital,t}(K,T) = e^{-r(T-t)}P_2, \quad P_{digital,t}(K,T) = e^{-r(T-t)}(1-P_2) \tag{71}$$

$$\text{Payoff}_{\text{digital call}} = \begin{cases} P, & \text{if } S_T \geq K \\ 0, & \text{if } S_T < K \end{cases}, \quad \text{Payoff}_{\text{digital put}} = \begin{cases} P, & \text{if } S_T < K \\ 0, & \text{if } S_T \geq K \end{cases} \tag{72}$$

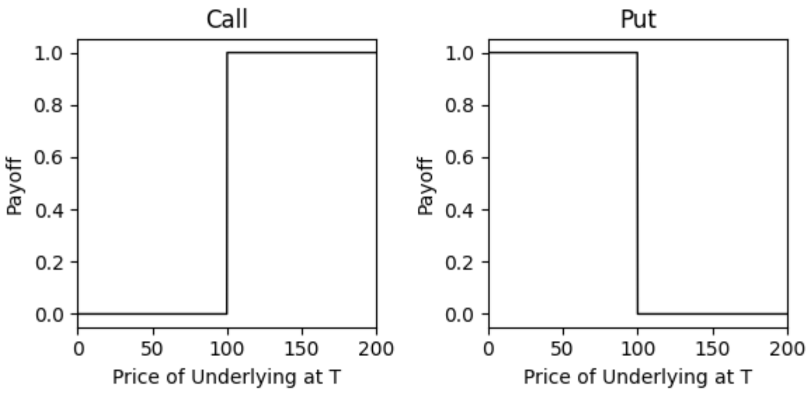


Figure 3.1: Payoff of digital call and put with strike price 100

Unlike standard European options, binary options exhibit a discontinuous, all-or-nothing payoff structure. This unique feature of payoff can effectively hedge jump risk, enriching the effectiveness of hedging strategy especially for non-linear, discontinuous exposure which cannot be replicated with standard call or put alone. In fact, some exotic options and structured products can have barrier embedded, creating non-linear, discontinuous risk exposure.

As the binary option buyer and the seller are in a zero-sum game, the syndicated binary put can be constructed with a short binary call with strike price K and a risk-free zero-coupon note of face value K with the same maturity or expiration date T. In this study, only only one out-of-money ($K = 0.95S_0 < S_0$), one at-the-money ($K = S_0$) and one in-the-money ($K = 1.05S_0 > S_0$) digital call options were considered.

To test how different hedging approaches with different degree of market completeness, 4 sets of hedging instruments were included: (a) only S&P 500 Index; (b) S&P 500 Index, and calls (OTM, ATM, ITM); (c) S&P 500 Index, ATM call, and ATM digital call; and (d) S&P 500 Index, calls (OTM, ATM, ITM), and digital calls (OTM, ATM, ITM).

3.4.3 Risk Exposure and Liabilities

To evaluate the performance of the hedging approaches across different complexity of financial liabilities, they were tested across different financial liabilities with underlying instrument S&P 500 Index, from as simple as an European call, to as complicated as a digital call and a basket of calls, puts and digital calls.

European Call In practice, European options are one of the most commonly traded derivatives in the financial market because of their simple structure and versatility in hedging and investment [12]. The risk exposure or liability from an European call or put can be easily offset by an opposite position of an European call or put with identical characteristics. Using machine learning approaches on such highly manageable risk exposure is definitely impractical, but this was necessary to test if the approaches could handle the basic liabilities before further investigation on more complex liabilities. Therefore, a 90-day at-the-money European call on S&P 500 Index was tested.

Digital Call The discontinuous payoff structure can be seen in many exotic derivatives and structure products, for example, barrier options where the option is only activated when the underlying price hit a barrier. The discontinuous structure cannot be easily hedged with the underlying instrument, European call or put, and standard exchanges typically do not list binary options, making the liquidity of binary options very limited for hedging purposes. To demonstrate how the machine learning approach responds to the discontinuous payoff with absence of binary options or other derivatives with discontinuous payoff, a 90-day at-the-money digital call was tested.

Basket/Portfolio of Options As mentioned in 2.1.1, financial institutions usually hold position in a wide range of financial derivatives, marking the hedging process more complicated. A basket that consists of one out-of-money ($K = 0.95S_0$) call, two short at-the-money ($K = S_0$) call, three in-the-money ($K = 1.05S_0$) call, four short in-the-money ($K = 0.95S_0$) put, five at-the-money ($K = S_0$) put, six short out-of-money ($K = 1.05S_0$) put, seven out-of-money ($K = 0.95S_0$) digital call, eight short at-the-money ($K = S_0$) digital call, and nine in-the-money ($K = 1.05S_0 > S_0$) digital call was considered as financial liability with complexity in Greek and discontinuous payoff:

$$
\begin{aligned}
\text{Basket} =& \text{Call}(0.95S_0) - 2\text{Call}(S_0) + 3\text{Call}(1.05S_0) \\
& - 4\text{Put}(0.95S_0) + 5\text{Put}(S_0) - 6\text{Put}(1.05S_0) \\
& + 7\text{Digital Call}(0.95S_0) - 8\text{Digital Call}(S_0) + 9\text{Digital Call}(1.05S_0)
\end{aligned}
$$

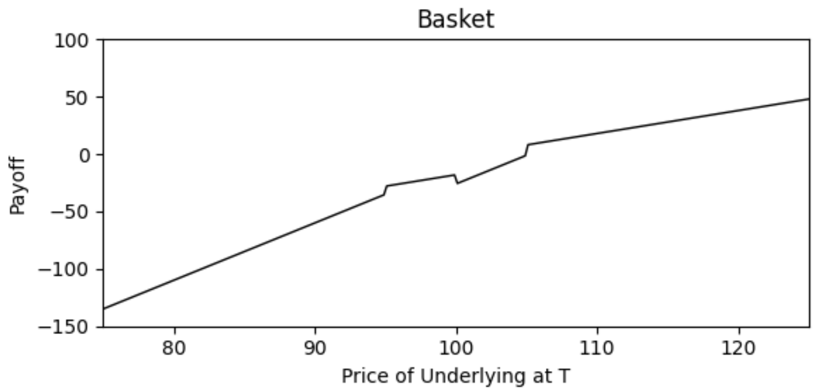


Figure 3.2: Payoff of the basket of options

3.5 Risk Measure CVaR

As detailed mentioned in **2.7.3**, the profit and loss of each simulated path with the hedging positions determined by different hedging approaches were calculated and then the corresponding α% CVaRs of the loss were estimated. As an objective function, the probability α was set at 0.95 and 0.99 to study the difference; while as a performance metric to evaluate the approaches, 90% CVaR, 95% CVaR and 99% CVaR were all reviewed. As a benchmark and for scaling the CVaRs of different financial liabilities to comparable magnitudes, the CVaRs of financial liabilities with no hedging action were also calculated.

3.6 Greek Hedging

The Greeks such as Delta Δ, Gamma Γ, Vega $\mathcal{V}$ and Theta Θ of the financial liabilities and hedging instruments were calculated by the Black-Scholes model to demonstrate how Greek hedging based on Black-Scholes performed. Where there was only S&P 500 Index as the hedging instrument, only Delta was considered; When there were only 3 hedging instruments, Theta was excluded; and when there were more than 4 hedging instruments, solution with the minimal total cost of changing the hedging positions was optimised by ADAM (see **2.8**).

3.7 Neural Network Model

3.7.1 Construction

The model was constructed based on the semi-recurrent neural network structure with one hidden layer of 64 neurones and activation function rectified linear unit (ReLU)function. The network had input dimension as 3 plus the number of hedging instruments for the price, volatility, time to expiration, and prior positions of each hedging instrument, and output dimension as the number of hedging instruments for the posterior position of each hedging instrument.

3.7.2 Training

The optimisation was to minimise the CVaR of the final loss to minimise with 1000 iterations of parameters update by ADAM (see **2.8**) with learning rate 0.001 adjusted by a scheduler that halved the learning rate every 200 iterations. Validation was also added to early terminate the training if the CVaR calculated with the intermediate model and the validation data did not improve more than 0.0001 for more than 10 iterations.

3.8 Q-Learning Model

To test the computational efficiency of the Q learning model, a sample run was taken with the basic discrete action level of $(-1, 0, 1)$ with 3 hedging instruments (27 actions in action space $\mathcal{A}$) for each of the 30 time steps with a reward function as negative of

CVaR of loss of a European call liability. The training took approximately 2 hours and the CVaR was significantly increased instead of being reduced compared to the case of no hedging action. To have a coherent comparison with other approaches, more diverse action levels, time steps, and hedging instruments must be included, making the total number of parameters exponentially greater. As a consequence, the Q-learning model for this study would require unpractical computational time ,even in a simpler setting. Therefore, the experiment **excluded the Q-learning model** to prevent an incomplete and misleading result.

3.9 Data Analysis

For each model, a random set of training data and a random set of testing data were drawn from the master data to train and evaluate the model and Greek hedging. Then the total transaction costs, 90%, 95%, and 99% CVaR for the approaches and for no hedging were measured. To adjust for the scale of different financial liabilities, total transaction costs were normalised by the value of the financial liability and CVaR for hedging approaches $\mathrm{CVaR}^{Hedging}$ were normalised by the CVaR of no hedging $\mathrm{CVaR}^{NoHedging}$, so the risk reductions (improvement) were calculated as $1 - \frac{\mathrm{CVaR}^{Hedging}}{\mathrm{CVaR}^{NoHedging}}$.

3.9.1 Descriptive Analysis

With a total of 288 combinations of the experiment variables and multiple runs, the result was first presented in descriptive statistics to check the patterns of performance of the hedging approaches. This included a frequency table of which hedging approaches reach the minimum CVaR compared to the others, mean risk reduction for different financial liabilities and a set of hedging instruments, mean total transaction costs for different proportional transaction costs and hedging frequencies, and the mean and standard deviation of risk reduction grouped by the experiment variables.

3.9.2 Statistical Inference

The resulting total transaction costs and CVaR were measured in pairs for the Greek hedging model and the neural network. To study the performance deference between these two hedging approaches, paired t-tests were performed to determine whether the neural network hedging approach can reduce more risk (CVaR) than Greek hedging. The CVaR difference between the hedging methodology and the absence of hedging, normalised by the CVaR of no hedging to eliminate the scaling by magnitude of the CVaR due to different financial liabilities, was defined as the percentage risk reduction $d = 1 - \frac{\text{CVaR}^{Hedged}}{\text{CVaR}^{NoHedge}}$. The following hypothesis test was conducted on the difference between the percentage risk reduction in **95% CVaR** of these two methodologies ($\delta_i = d_i^{NN} - d_i^{Greek}$).

24 one-tailed paired t-test were performed with the overall result and results grouped by experiment variables. The null hypothesis would be rejected **if the test statistic $\hat{t}$ is greater than the critical value** (or the p-value is less than the significance level). To control the overall error rate with multiple hypothesis tests, the Bonferroni adjusted p-value with the total significant level of 5% was used in the statistical inference, i.e., the significance level for each test = 5%/24 = 0.2083% . The paired t-test was formulated as follows:

$$H_0 : \bar{\delta} \leq 0 \text{ (Neural network model cannot reduce more CVaR than Greek hedging) vs}$$
$$H_1 : \bar{\delta} > 0 \text{ (Otherwise)}$$
$$\text{for } \bar{\delta} = \frac{\sum_{i=1}^{N} d_i^{NN} - d_i^{Greek}}{N}, \quad \hat{t} = \frac{\bar{\delta}}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\delta_i - \bar{\delta})^2} \Big/ \sqrt{N}} \sim t_{N-1} \text{ under } H_0 \tag{73}$$

3.10 Summary of Experiment

Table 3.1: Experiment design: fixed parameters and settings

Parameter	Specification
Underlying instrument	S&P 500 Index
Risk-free rate (r)	4.2874%
Drift rate (μ)	13.0139%
Dividend yield (q)	1.4317%
Stochastic model	Heston model for stochastic price and volatility
	($\hat{\kappa} = 1.9216, \hat{\theta} = 0.0442, \hat{\sigma} = 0.2905, \hat{v_0} = 0.0157$ & $\hat{\rho} = -1$)
Time horizon (T)	90 days
Number of steps	90 in simulation
Risk measure	Conditional value-at-risk (CVaR)
Hedging approach	(a) Greek hedging and (b) Semi-RNN model*

*Q-learning model was omitted due to the computational intensity.

Table 3.2: Experimental design: test cases (288 combination)

Parameter	Specification
Financial liability	European call option; Digital call option; Basket of options
Hedging frequency (Δt)	Daily (1 day); Every 2 days; Weekly (7 days); Monthly (30 days)
Transaction cost rate	0%; 0.01% (1 basis point); 0.10% (10 basis points)
α **for CVaR**	95%; 99%
Hedging instruments	(a) Underlying asset only
	(b) Underlying asset and European calls (OTM, ATM, ITM)
	(c) Underlying asset, ATM call, and ATM digital call
	(d) Underlying asset, European calls (OTM, ATM, ITM) and Digital calls (OTM, ATM, ITM)

4. RESULTS

4.1 Descriptive Statistics

The semi-recurrent network models were trained for each of the 288 test case combinations for 4 runs (1,152 models in total). The summarised statistics are following:

Table 4.1: Frequency of different hedging approaches with minimum CVaR

	No hedging	**Greek hedging**	**Neural network model**
90% CVaR	3 (0.26%)	214 (18.58%)	935 (81.16%)
95% CVaR	1 (0.09%)	179 (15.54%)	972 (84.38%)
99% CVaR	17 (1.48%)	167 (14.50%)	968 (84.03%)

Table 4.2: Mean risk reduction in CVaR for different hedging instrument sets and financial liabilities

	Call		**Digital call**		**Basket of options**	
Hedging instrument set	**Greek hedging**	**Neural network**	**Greek hedging**	**Neural network**	**Greek hedging**	**Neural network**
(a)	83.76%	83.40%	-15.88%	5.32%	82.39%	76.17%
(b)	109.75%	114.87%	-95.92%	36.91%	-97.86%	93.11%
(c)	106.96%	113.52%	1.26%	26.88%	39.05%	91.67%
(d)	36.31%	116.30%	-41.90%	46.03%	30.10%	93.95%

Table 4.3: Mean total transaction costs relative to the value of financial liability for different hedging frequencies and transaction cost rates

	Transaction cost rate = 0.01%			**Transaction cost rate = 0.10%**		
Hedging frequency	**Greek hedging**	**Neural network**	**Difference**	**Greek hedging**	**Neural network**	**Difference**
3	0.4696%	0.2202%	0.2494%	4.7633%	1.5112%	3.2521%
12	1.0023%	0.1460%	0.8563%	10.0561%	1.2829%	8.7732%
45	1.9373%	0.1738%	1.7636%	19.4824%	1.3978%	18.0846%
90	2.7214%	0.2055%	2.5159%	26.9326%	1.4665%	25.4661%

Table 4.4: Mean (standard deviation) percentage improvement in CVaR by Greek hedging and Neural network in overall and under different settings

	90% CVaR		**95% CVaR**		**99% CVaR**	
	Greek hedging	**Neural network**	**Greek hedging**	**Neural network**	**Greek hedging**	**Neural network**
Overall	26.93%	76.46%	19.83%	74.84%	6.59%	71.94%
	(66.69%)	(37.42%)	(71.49%)	(37.58%)	(83.40%)	(39.06%)
Financial liability						
Call	87.57%	108.91%	84.20%	107.02%	78.07%	104.34%
	(32.72%)	(18.31%)	(34.20%)	(17.39%)	(40.20%)	(17.52%)
Digital call (d.call)	-29.23%	31.04%	-38.11%	28.78%	-52.26%	24.06%
	(38.80%)	(22.66%)	(39.35%)	(22.64%)	(44.66%)	(24.53%)
Basket of options	22.46%	89.43%	13.42%	88.73%	-6.04%	87.40%
	(62.61%)	(8.09%)	(71.28%)	(8.43%)	(92.40%)	(9.26%)
Hedging instrument set						
(a) S&P 500	56.86%	56.49%	50.09%	54.96%	49.09%	51.31%
	(41.80%)	(35.55%)	(47.77%)	(35.76%)	(57.59%)	(37.55%)
(b) S&P 500 and	-17.23%	83.08%	-28.01%	81.63%	-51.12%	79.46%
calls	(93.59%)	(34.9%)	(100.11%)	(34.85%)	(118.28%)	(35.36%)
(c) S&P 500,	51.52%	79.22%	39.70%	77.36%	44.80%	74.05%
ATM call and d.call	(45.14%)	(39.06%)	(45.08%)	(39.74%)	(45.71%)	(42.16%)
(d) S&P 500,	16.57%	87.05%	8.17%	85.43%	-7.03%	82.92%
calls and d.calls	(41.48%)	(32.29%)	(44.17%)	(32.09%)	(49.06%)	(32.73%)
Hedging frequency						
3 (monthly)	36.06%	75.79%	28.78%	73.88%	12.87%	71.47%
	(59.93%)	(50.59%)	(64.58%)	(49.76%)	(78.83%)	(49.14%)
12 (weekly)	26.63%	74.21%	19.48%	73.00%	6.52%	70.40%
	(66.54%)	(36.22%)	(71.63%)	(36.59%)	(83.93%)	(37.83%)
45 (every 2 days)	23.52%	75.02%	16.73%	73.17%	4.42%	69.47%
	(69.25%)	(31.94%)	(74.02%)	(32.94%)	(85.83%)	(36.49%)
90 (daily)	21.51%	80.81%	14.35%	79.33%	2.54%	76.41%
	(69.97%)	(26.39%)	(74.82%)	(27.13%)	(84.90%)	(30.20%)
Transaction cost rate						
0.00%	31.24%	77.02%	23.98%	75.32%	11.13%	72.21%
	(65.25%)	(37.71%)	(70.39%)	(37.93%)	(82.07%)	(39.96%)
0.01%	31.32%	76.93%	24.46%	75.26%	12.11%	72.41%
	(64.53%)	(37.59%)	(68.96%)	(37.65%)	(79.40%)	(38.87%)
0.10%	18.23%	75.43%	11.07%	73.96%	-3.48%	71.19%
	(69.50%)	(37.02%)	(74.38%)	(37.24%)	(87.81%)	(38.42%)
α of training CVaR						
0.95	27.05%	77.85%	20.02%	75.95%	7.47%	72.53%
	(66.69%)	(36.36%)	(71.47%)	(36.88%)	(82.60%)	(38.71%)
0.99	26.81%	75.06%	19.65%	73.74%	5.71%	71.34%
	(66.74%)	(38.43%)	(71.57%)	(38.26%)	(84.26%)	(39.43%)

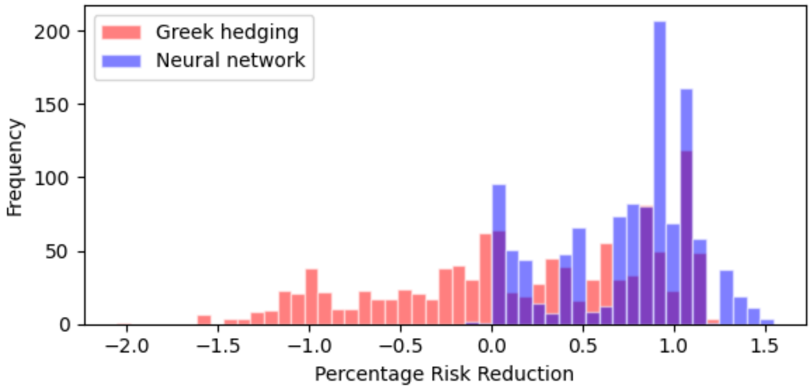


Figure 4.1: Distribution of percentage risk reduction by hedging approaches

4.2 Statistical Inference

With significance levels of 1% and 5% for 24 paired t-test, the Bonferroni adjusted p-values are 0.0417% and 0.2083%. The test statistic, its p-value, and critical values are presented as follows:

Table 4.5: test statistic and p-values for different settings

	test statistics	p-value	5% adjusted critical value	1% adjusted critical value
Overall	30.2100**	2.272e-28	2.8710	3.3503
Financial liabilities				
Call	11.8972**	2.72e−28	2.8826	3.3682
Digital call (d.call)	24.6384**	2.68e−81	2.8826	3.3682
Basket of options	19.6478**	3.03e−60	2.8826	3.3682
Hedging instrument set				
(a) S&P 500	5.8153**	8.06e−09	2.8884	3.3772
(b) S&P 500 and calls	22.6414**	4.06e−66	2.8884	3.3772
(c) S&P 500, ATM call and d.call	19.2175**	8.83e−54	2.8884	3.3772
(d) S&P 500, calls and d.calls	41.2922**	4.59e−123	2.8884	3.3772
Hedging frequency				
3 (monthly)	14.8004**	1.61e−37	2.8884	3.3772
12 (weekly)	14.6331**	6.57e−37	2.8884	3.3772
45 (every 2 days)	14.7971**	6.15e−37	2.8884	3.3772
90 (daily)	16.6254**	3.08e−44	2.8884	3.3772
Transaction cost rate				
0%	16.8641**	2.02e−48	2.8826	3.3682
0.01%	17.1833**	9.07e−50	2.8826	3.3682
0.10%	18.4318**	4.57e−55	2.8826	3.3682

* H_0 was rejected at the total significance level of 1%.
** H_0 was rejected at the total significance level of both 1% and 5%.

Table 4.6: Test statistics and p-values for different hedging sets and financial liabilities

	Call		Digital call		Basket of options	
Hedging set instrument set	Test statistic	p-value	Test statistic	p-value	Test statistic	p-value
(a)	-0.5891	0.7214	18.1353**	6.49e−33	-13.1207	1
(b)	6.2933**	4.76e−09	42.5610**	5.66e−64	55.5045**	1.67e−74
(c)	9.7441**	2.91e−16	10.8921**	1.04e−18	34.4132**	9.74e−56
(d)	23.3731**	1.78e−41	25.4753**	1.52e−44	28.7472**	5.82e−49

Note: 1% and 5% adjusted critical values are 3.4516 and 2.9363.
* H_0 was rejected at the total significance level of 1%.
** H_0 was rejected at the total significance level of both 1% and 5%.

5. DISCUSSION

5.1 Descriptive Analysis

Table 4.1 showed that neural network models could reach the minimum CVaR among Greek hedging and no hedging for more than 80% time of the 4 runs of 288 test cases (1152 cases) and Figure 4.1 shows that the neural network model could reduce CVaR with greater stability in most cases, on average more than 70% CVaR and with standard deviation of about 40%, while Greek hedging could increase CVaR in some cases with a mean around 20%. Although Greek hedging had overall poorer and less stable performance, it also had its strengths in certain aspects such as hedging the at-the-money call and the basket of options with only S&P 500 Index (see **Appendix**). Therefore, it is important to check how the hedging approaches performed in different settings.

To deepen the interpretation, the discussion now turns to results disaggregated by the factors of experiment setting:

Type of financial liability Unexpectedly, Greek hedging could not reduce the CVaR of an ATM European call as much as the neural network model, even when the Greeks were well defined, indicating that more investigation is needed. As expected, Greek hedging failed to hedge the ATM digital call with a remarkably negative reduction lower than -30% and had limited risk reduction for the basket including digital calls. This outcome aligns with the weakness of Greek hedging for the discontinuous payoff and not continuously differentiable value with respect to the price of the underlying instrument, in particular, the undefined Delta and Gamma at $S_0 = K$. At the same time, neural network models could reduce all risk for the call and most risk for the basket and had limited, yet better, performance for the digital call.

Set of hedging instruments With only S&P 500 Index, the two approaches had similar performance around 50%. With more hedging instruments available, the neural network model showed better performance, but Greek hedging had worse performance whenever digital calls were available due to its shortfall in handling binary options and other derivatives with discontinuous payoffs and non-differentiable values. Then it raises the question why Greek hedging failed with increased CVaR with set (b), which consists of S&P 500 Index and calls with well-defined Greeks.

Hedging frequency The performance of neural network model was rather stable with different hedging frequency as the objective function encoded the impact of transaction cost, so the model was trained to avoid increasing total transaction cost and hence frequent trading. However, Greek hedging was designed to keep Greeks at target values, so it rebalanced each time and increased total transaction costs, decreasing CVaR reduction with hedging frequency.

Proportional transaction cost As mentioned above, neural network models were trained to avoid increasing transaction costs, so the increase in proportional transaction cost did not affect their average performance. In contrast, Greek hedging had a significant decrease in performance when the proportional transaction cost increased significantly from 0.01% to 0.10%.

Probability α of training objective CVaR The different α did not show noticeable differences in performance. Beyond the isolated effect of each factor in the experiment setting, the interaction between factors provides further insight into when and why certain hedging approaches succeed or fail. According to Table 4.2, both approaches had comparable risk reductions for the ATM European call, except the noticeably poorer performance of the Greek hedging with the set (d) (S&P 500, calls, and digital calls), and the neural network had a greater risk reduction for the ATM digital option and the basket of options, except for the basket with the S&P 500 index only. The possible explanation for these two observations could be the complication of Greek hedging with possibly undefined Greeks from the multiple digital calls in (d) lowering its mean performance for call. and the similarity of the shape of payoff of the S&P 500 Index and the basket. Also, the set (b) had a remarkably negative reduction around -95% for the digital call and the basket, distorting the mean performance of Greek hedging with (b) mentioned earlier. In addition, neural network models managed to take advantage of the richer instrument hedging set to have more risk reduction, which Greek hedging could not. However, it is important to note that the ATM digital call liability could actually be completely offset by the ATM digital call in the hedging instrument set, but neither Greek hedging nor the neural network model could respond correctly.

Regarding total transaction costs, the costs were higher and increased with hedging frequency and the transaction cost rates for Greek hedging, as mentioned in **2.1.3**. In contrast, costs with neural network models did not increase with frequency and were multiplied about 8 times when the rate was multiplied by 10 times, demonstrating the outstanding ability of the neural network model to reduce transaction costs. When the transaction cost rate was higher, the impact of the transaction cost on the CVaR objective function increased, so the optimised models formulated decision to reduce the costs.

5.2 Statistical Inference

After descriptive analysis and discussion, to conclude the result, it is essential to test whether neural network models are statistically significantly better at reducing CVaR with statistical inference. In 3.9.2, the one-tailed paired t-test was purposed and 24 of them were taken with adjusted p-value. According to Tables 4.5 and 4.6, 22 of 24 hypothesis tests rejected the null hypothesis H_0 that neural network models cannot reduce more CVaR than Greek hedging with a p-value significantly lower than 10^{-8}, except for the 2 cases of hedging the ATM European call and the basket of options with only S&P 500 index. The first case has a small difference on average, 0. 36%, but could indicate that Greek hedging still has its value in some simple situation. The second case could be explained by the shape of the payoff of the basket of options. Although it includes 3 digital calls, they are relatively small with notional 7, 8, and 9, creating just small jumps in the payoff and limiting their impact on the final payoff. In this situation with the calls and puts options dominating in the basket, Greek hedging could provide an analytical solution to reduce risk and CVaR.

5.3 Limitations and Possible Improvements

5.3.1 Scope of Financial Liabilities

As the financial liability shapes the final payoff and hence the CVaR dominantly, the coverage of financial liabilities determines the representative of the study and comparison. This experiment only covered an at-the-money European call, an at-the-money digital call, and a basket of options; therefore, the interpretation of the result could limit to European call, European put under the call-put parity, digital call, digital put, and baskets of the former derivatives. In addition to the type of financial liabilities, the study only covered one underlying instrument S&P 500 Index and a 90-day time horizon. With the strong economic foundation of the US, the global interest in the US market and the diversification nature, S&P 500 Index is more liquid and less risky than most equity and indexes, so the extension of the result of this study is limited to liquid and less risky underlying equities similar to S&P 500 Index, for example, some value stocks in developed counties. To broaden the scope of this study, the types of financial liability, the time horizons, and the underlying instruments could be diversified in a further study.

5.3.2 Market Model

The simulation of price and volatility was based on a calibrated Heston model, which is highly subject to the model assumptions and the calibration process including the data selection and preprocessing. It is inevitable that there might be some discrepancy between the simulated situation and the real market. Consequently, the actual performance in the practical use of the neural network models may differ from the observation in this study. This limitation is not unique to the Heston model, as all models reply on their assumptions and approximation to simulate the reality, while historical data itself are limited in that the past behaviour does not guarantee future outcomes. In this study, the Heston model was prudently chosen as a practical compromise, as its stochastic price and stochastic volatility features provide a more realistic market representation than simpler alternatives like the Black-Scholes model, while remaining computationally tractable.

5.3.3 Transaction Costs and Financing Rate

Transaction cost was encoded by proportional transaction cost and financing, either borrowing and lending, was at a constant risk-free rate. In reality, transaction cost has more complicated structures, for example, bid-ask spread and non-linear costs schedules, and financing is charged at different rate for different direction, creditability, and time horizon. The study simplified them to lower the computational complexity. For an in-depth study on the neural network model on transaction cost and financing cost, a more realistic calculation should be included.

5.3.4 Simple Structure of Neural Network

The semi-recurrent network in this study only took three variables besides the position of hedging instruments and included only one hidden layer of 64 neurones. Even though the performance of neural network models has proven better than in this section, the performance may be boosted by including more input features and constructing the network with more hidden layers of more neurones if computational capacity allows.

5.3.5 Exclusion of Q-Learning Model

Due to the extensive computational time and the limit time frame of this study, the Q-learning model was omitted in this study, so this study could not provide a conclusion on how well Q-learning models could hedge risk. The use of GPU and parallel computing could allow the exploration on the model, but it could be costly to reach a conclusion. Nevertheless, this study could conclude that the Q-learning model is not a feasible alternative with limited computational capacity.

6. CONCLUSION

6.1 Best Performers if any

This thesis aimed to find an alternative hedging approach particularly with machine learning techniques among other statistical models, such as the GARCH model, Heston model, neural network model, and Q-learning, to provide a more cost efficient and more robust risk reduction in the realistic market conditions with frictions and stochastic volatility. The approaches were compared in literature and later by computational simulation experiments. With simulated paths of price and vocality of the S&P 500 Index by a calibrated Heston model, the relative reduction of risk measured by conditional value-at-risk (CVaR) by Greek hedging and semi-recurrent neural network models were examined across multiple financial liabilities, hedging instruments, hedging frequency, proportional transaction costs, and CVaR confidence levels.

The empirical results consistently demonstrated the superiority of the neural network approach over Greek hedging. Neural networks achieve statistically greater reductions in CVaR across the majority of settings, with higher robustness and lower transaction costs. In contrast, Greek hedging showed unstable and poorer performance and often failed to reduce risk for derivatives with discontinuous payoffs such as digital options. The increased hedging frequency led to higher transaction costs by Greek hedging, whereas the neural network was able to adapt to transaction costs and produce cost-efficient strategies. The results underline the ability of machine learning tools to capture complicated and structured dynamics of the hedging problem and account for the frictions and stochastic volatility in real market situation which classical approaches neglect.

Nevertheless, the study has several limitations. The simulated data were subject to the weaknesses of the Heston model and calibration process with data selection and preprocessing. The experiment was limited to three types of financial liabilities, one underlying instrument, one time horizon, and a constant risk-free rate. Frictions was only encoded as proportional transaction costs while it can come in other forms. The implemented neural network structure was simple with only two input features to cover market information and one hidden layer, while Q-learning was excluded due to its computational intensity. These limitations indicate the need for future research to extend the study to more diverse liabilities, richer cost structure, and other machines learning tools and to validate the result with real market data.

Above all, the thesis provides statistically strong evidence that machine learning, in particular semi-recurrent neural networks, offers a significant improvement over classical Greek hedging. By systematically outperforming Greek hedging across a wide range of realistic market settings and simple discontinuous financial liabilities, the neural network approach demonstrates the potential of machine learning to become an efficient and robust hedging alternative in financial markets.

REFERENCE

[1] Artzner, P., Delbaen, F., Eber, J.-M., Heath, D. (1999). Coherent Measures of Risk. Mathematical Finance, 9(3), 203–228. https://doi.org/10.1111/1467-9965.00068

[2] Basel Committee On Banking Supervision Revises Standards On Minimum Capital Requirements For Market Risk. (2016). Mondaq Business Briefing.

[3] Black, F., Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. Journal of Political Economy, 81(3), 637–654. http://www.jstor.org/stable/1831029

[4] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

[5] Buehler, H., Gonon, L., Teichmann, J., Wood, B. (2019). Deep hedging. Quantitative Finance, 19(8), 1271–1291. https://doi.org/10.1080/14697688.2019.1571683

[6] Caldwell, M. (2024). Lecture Notes for COMP0088: Introduction to Machine Learning. University College London. Unpublished lecture notes.

[7] Engle, R. F. (1982). "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of U.K. Inflation." Econometrica, 50(4), 987–1007.

[8] Fouque, J.-P., Papanicolaou, G., Sircar, R., Sølna, K. (2011). Multiscale Stochastic Volatility for Equity, Interest Rate, and Credit Derivatives / Jean-Pierre Fouque, George Papanicolaou, Ronnie Sircar, Knut Sølna. Cambridge University Press.

[9] Halperin, I. (2017). QLBS: Q-Learner in the Black-Scholes(-Merton) Worlds. https://doi.org/10.48550/arxiv.1712.04609

[10] Hastie, T., Tibshirani, R., Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction / Trevor Hastie, Robert Tibshirani, Jerome Friedman. (Second editon.). Springer Verlag.

[11] Heston, S. L. (1993). A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. The Review of Financial Studies, 6(2), 327–343. https://doi.org/10.1093/rfs/6.2.327

[12] Hull, J. C. (John C.). (2022). Options, futures, and other derivatives (Eleventh edition, global edition). Pearson.

[13] Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arxiv.1412.6980

[14] Maier, S. (2025). Lecture Notes for STAT0018: Stochastic Methods in Finance II. University College London. Unpublished lecture notes.

[15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature (London), 518(7540), 529–533. https://doi.org/10.1038/nature14236

[16] Rockafellar, R. T., Uryasev, S. (2000). Optimization of conditional value-at-risk. The Journal of Risk, 2(3), 21–41. https://doi.org/10.21314/JOR.2000.038

[17] Siddiqui, S. (2016). The Calibration of the Heston Model. University College London

[18] Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. Proceedings 2017 IEEE Winter Conference on Applications of Computer Vision: 24-31 March 2017, Santa Rosa, California, 464–472. https://doi.org/10.1109/WACV.2017.58

[19] Watkins, C. J. C. H., Dayan, P. (1992). Technical Note: Q-Learning. Machine Learning, 8(3–4), 279–292. https://doi.org/10.1023/A:1022676722315

APPENDIX A: DETAILED RESULTS

The mean CVaR of no hedging and CVaR reduction and relative transaction cost of Greek hedging and neural network models for the 288 combinations can be downloaded from:

https://github.com/neville-1201/Deep-learning-approach-for-hedging/blob/0c5095a29c69015298671b20f9d21afacf9ab8ac/Result.xlsx

APPENDIX B: PYTHON CODE

6.2 Semi-Recurrent Neural Network Structure

```
class DeepHedging_NN_d(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim, cap =  None,
    ↪ dtype=torch.float64, device='cpu'):
        super().__init__()
        self.input_dim = input_dim #number of input features of each
        ↪ observation: St, vt,
        self.hidden_dim = hidden_dim
        self.output_dim = output_dim #number of risky hedging instrument (No
        ↪ risk-free note)
        self.cap = cap
        dtype = dtype
        device = device
        self.ff_in = nn.Linear(input_dim + output_dim, hidden_dim, dtype =
        ↪ dtype, device = device)
        self.ff_mid = nn.ReLU()
        self.fc_out = nn.Linear(hidden_dim, output_dim, dtype = dtype, device
        ↪ = device)

    def forward(self, X):
        n_sim, n_step, _ = X.shape
        dtype = X.dtype
        device = X.device
        output_dim = self.fc_out.out_features
```

```
        hidden = torch.zeros(n_sim, self.hidden_dim, dtype = dtype, device =
         ↪  device)
        pos_hedge = torch.zeros(n_sim, output_dim, dtype = dtype, device =
         ↪  device)
        actions = []
        cap = None
        if getattr(self, "cap", None) is not None:
            cap = torch.as_tensor(self.cap, dtype = dtype, device = device)
        for tt in range(n_step - 1):
            X_t = X[:, tt, :]
            z_t = torch.cat([X_t, pos_hedge], dim = -1)
            hidden = self.ff_mid(self.ff_in(z_t))
            d_hedge = self.fc_out(hidden)
            if cap is not None:
                d_hedge = torch.clamp(d_hedge, -cap - pos_hedge, cap -
                 ↪  pos_hedge)
            pos_hedge = pos_hedge + d_hedge
            actions.append(d_hedge)
        return torch.stack(actions, dim = 1)
```

6.3 Calculation of Profit and Loss

```
def PnL(d_hedge, price_hedge, liability, transaction_cost = 0.001, TC_output =
 ↪  False):
    n_sim, n_step, n_hedge = d_hedge.shape
    T = liability.T
    r = liability.r
    dt = T/n_step
    cash_0 = liability.Price()
    cash = torch.ones(n_sim, device = price_hedge.device) * cash_0
    #cash = torch.zeros(n_sim, device = price_hedge.device)
    position = torch.zeros(n_sim, n_hedge, device = price_hedge.device)
    total_TC = torch.zeros(n_sim, device = price_hedge.device)
    total_interest = torch.zeros(n_sim, device = price_hedge.device)
    # from t=0 to t=T-dt
```

```
    for tt in range(n_step):
        price_hedge_t = price_hedge[:, tt, :]
        d_hedge_t = d_hedge[:, tt, :]
        position = position + d_hedge_t
        proceed_t = torch.sum(price_hedge_t * d_hedge_t, dim = -1)
        TC_t = torch.sum(torch.abs(d_hedge_t) * price_hedge_t, dim = -1) *
         ↪  transaction_cost
        total_TC = total_TC + TC_t
        interest_t = cash * (r * dt)
        total_interest = total_interest + interest_t
        cash = cash - proceed_t - TC_t + interest_t
    payoff_hedge = price_hedge[:, -1, :] #payoff at T of each hedging
     ↪  instruments
    value_hedge = torch.sum(payoff_hedge * position, dim = -1)
    payoff_liability = liability.Payoff(payoff_hedge[:, 0])
    PnL = value_hedge + cash * (1 + r * dt) - payoff_liability
    PnL_NH = - payoff_liability + cash_0 * (1 + r * T)
    if TC_output:
        return PnL, PnL_NH , total_TC
    else:
        return PnL, PnL_NH
```

6.4 Calculation of CVaR

```
def CVaR_PnL(PnL, alpha):
    alpha = float(torch.clamp(torch.tensor(alpha), 1e-8, 1 - 1e-8))
    loss = -PnL
    VaR = torch.quantile(loss.detach(), 1.0 - alpha)
    tail = F.relu(loss - VaR)
    CVaR = VaR + tail.mean() / alpha
    return CVaR
```

6.5 Training

```
def Training(model, liability, Loss, X, price_hedge, T, r, tc, alpha, n_epoch
↪ = 100, lr = 1e-3, batch_size = 512, val_portion = 1/9 , clip = 1.0,
↪ verbose = True, early_terminate = False, patience = 10, min_improve =
↪ 1e-4):
    device = next(model.parameters()).device
    dtype = next(model.parameters()).dtype
    n_sim = X.shape[0]
    ### Data Split ###
    X = X.to(device = device, dtype = dtype).detach()
    price_hedge = price_hedge.to(device = device, dtype = dtype).detach()
    index1 = torch.randperm(n_sim, device = device)
    n_Va = int(round(val_portion * n_sim))
    n_Tr = n_sim - n_Va
    Tr_index = index1[n_Va:]
    Va_index = index1[:n_Va]
    X_Tr = X[Tr_index]
    price_hedge_Tr = price_hedge[Tr_index]
    X_Va = X[Va_index]
    price_hedge_Va = price_hedge[Va_index]
    ### Optimisation setup ###
    optimiser  = torch.optim.Adam(model.parameters(), lr = lr, betas = (0.9,
    ↪ 0.999), eps = 1e-8)
    #optimiser = torch.optim.SGD(model.parameters(), lr = lr)
    scheduler = torch.optim.lr_scheduler.StepLR(optimiser, step_size =
    ↪ max(n_epoch//5, 1), gamma = 0.5)
    best_loss_Va = float("inf")
    wait = 0
    ### Training ###
    model.train()
    for epoch in tqdm(range(n_epoch), ncols = 100):
        index2 = torch.randperm(n_Tr, device = device)
        epoch_loss = 0.0
```

```
n_batch = 0
for start in range(0, n_Tr, batch_size):
    end = min(start + batch_size, n_Tr)
    batch_index =  index2[start:end]
    X_batch = X_Tr[batch_index]
    price_hedge_batch = price_hedge_Tr[batch_index]

    optimiser.zero_grad(set_to_none = True)

    d_hedge_batch = model(X_batch)

    loss = Loss(d_hedge_batch, price_hedge_batch, liability, alpha,
     ↪  transaction_cost = tc, TC_output = False)
    loss.backward()

    if clip is not None:
        nn.utils.clip_grad_norm_(model.parameters(), clip)
    optimiser.step()
    epoch_loss = epoch_loss + float(loss.item())
    n_batch = n_batch + 1
scheduler.step()
### Validation ###
model.eval()
with torch.no_grad():
    d_hedge_Va = model(X_Va)
    loss_Va = float(Loss(d_hedge_Va, price_hedge_Va, liability, alpha,
     ↪  transaction_cost = tc, TC_output = False).item())
if best_loss_Va - loss_Va > min_improve:
    best_loss_Va = loss_Va
    wait = 0
else:
    wait = wait + 1
if early_terminate and wait >= patience:
    if verbose:
```

```
            tqdm.write(f"###Early terminated without improvement greater
            ↪  than {min_improve} for {patience} epochs.")
            tqdm.write(f"Epoch {epoch + 1: 02d}/{n_epoch} | Training CVaR:
            ↪ {epoch_loss/n_batch: .6f} | Validation CVaR:
            ↪ {epoch_loss/n_batch: .6f} [Best: {best_loss_Va: .6f}] | LR:
            ↪ {cur_lr: .3g}")
        break
    if verbose:
        cur_lr = optimiser.param_groups[0]["lr"]
        if epoch % (n_epoch // 10) == 0 or epoch == n_epoch - 1:
            tqdm.write(f"Epoch {epoch + 1: 02d}/{n_epoch} | Training CVaR:
            ↪ {epoch_loss/n_batch: .6f} | Validation CVaR:
            ↪ {epoch_loss/n_batch: .6f} [Best: {best_loss_Va: .6f}] | LR:
            ↪ {cur_lr: .3g}")
    model.train()
```

The complete version of Python code for this thesis can be founded:

https://github.com/neville-1201/Deep-learning-approach-for-hedging