Deep Hedging Fixed Strike Asian Options

Lauri Teerimäki
Yonsei Univesity, Seoul, South Korea

ABSTRACT

Controlling ones risk in the financial markets is something traders have been trying to master for decades. In 1973, Black & Scholes realized that two risky positions taken together, could essentially eliminate risk itself, which resulted in the creation of the Black-Scholes model, one of the most important concepts in financial theory. However, the model makes some heavy assumptions, such as no trading costs and continuous rebalancing, which make the model suboptimal in real use. This paper uses deep hedging, a method based on neural networks, for hedging arithmetic average Asian fixed strike options. Arithmetic average Asian fixed strike options are path dependent options that do not have a closed form equation, and hence do not have a delta. A pseudo-delta will be estimated by the neural network model.

1 Introduction

This paper aims to hedge arithmetic average Asian fixed strike options using deep neural networks. The Black-Scholes model is one of the most important and widely used models in modern finance, but it makes some heavy assumptions that make it suboptimal for pricing options in real life. Essentially the model assumes perfect markets where there are no transaction costs. This leaves it for the traders to take into consideration the market imperfections. Deep hedging uses neural networks to estimate a pseudo-delta and to minimize the cost of hedging. In this paper, a semi-recurrent neural network is built and is tested for hedging arithmetic average Asian fixed strike options.

Asian options are path dependent options where the options price relies on the average price of the underlying asset between a certain period, which creates some interesting implications. First and foremost, the options are highly path dependent, which make them more difficult to evaluate correctly. They also have a lower volatility than their traditional counterparts, which make them intriguing for hedging purposes and being less exposed to market manipulation.

2 Theoretical framework

2.1 Black – Scholes model

The Black-Scholes model is one of the quintessential models in financial theory. The model is used for calculating the theoretical value of options. Since almost all corporate liabilities can be regarded as combinations of options, the model is also applicable to corporate liabilities such as common stocks, warrants and corporate bonds (Black & Scholes 1973).

The following Stochastic Differential Equation can be used to describe asset prices:

$$dS=\sigma SdZ+(\mu-\delta)Sdt \tag{1}$$

Where S is the asset value, σ is the volatility, μ is the drift or expected return, δ is the continuous dividend yield of the asset, dS represents incremental changes in asset value, dZ is a Weiner process and dt represents incremental changes in time. Lakhlani (2013) derives the Black-Scholes model in his research paper by using Ito's Lemma.

By applying Ito's Lemma for δ = 0, we can notice that S solves the aforementioned equation. Then V(S,t) solves the following:

$$dV=\sigma SV_s dX+\left(\mu SV_s+\frac{1}{2}\sigma^2 S^2 V_{SS}+V_t\right)dt \tag{2}$$

By constructing a portfolio ∏, with one long option V(S,t) and being short a Δ fraction of the underlying asset

$$\Pi=V-\Delta\cdot S\Rightarrow d\Pi=dV-\Delta dS \tag*{(3), (4)}$$

and substituting equations 1 and 2 into 4, and after some algebraic manipulation, the following formula can be derived:

$$d\Pi=\sigma S(V_S-\Delta)dX+\left(\mu SV_S+\frac{1}{2}\sigma^2 S^2 V_{SS}+V_t-\mu\Delta S\right)dt$$

for $\Delta=V_S$,we get

$$d\Pi=\left(\mu SV_S+\frac{1}{2}\sigma^2 S^2 V_{SS}+V_t\right)dt$$

By constructing the portfolio this way, the portfolio is completely independent of the randomness from the underlying asset. One of the assumptions in the Black-Scholes model is a no riskless arbitrage market. Assuming there are no riskless arbitrage opportunities and by additionally investing and divesting at a risk-free-rate, the equation above turns into

$$\Pi=\frac{1}{r}\left(\mu SV_S+\frac{1}{2}\sigma^2 S^2 V_{SS}+V_t\right) \tag{5}$$

Now, substituting $\Delta= V_s$ and (5) in (3) the Black-Scholes Equation (BSE) can be shown as

$$v_t+\frac{1}{2}\sigma^2 S^2 V_{SS}+rSV_s-rV=0 \tag{6}$$

In the derivation above, the Black-Scholes model makes several different assumptions. Black and Scholes assume that the underlying asset follows geometric Brownian motion with constant volatility, that there are no dividends and that there is a constant risk-free rate, which market participants can use to borrow or lend. Additionally, they assume that:

1. The number of outstanding stocks is constant
2. The price of the stock is log-normally distributed
3. No transaction costs

The payoff of a European call option at the time of maturity is:

C = max(S-K, 0)

For a strike price, *K > 0, max(0,0-K) = 0.* The underlying asset growing without a bound, the payoff will be *max(0, S-K) = S*. Hence, the boundary conditions are:

$$C(0,t)=0$$

$$C(S\rightarrow\infty,t)=S$$

Using the boundary conditions in BSE, the equation for a non-dividend paying European call option can be derived as:

$$C=S_0 N(d_1)-K\,\mathrm{e}^{-rT}N(d_2) \tag{6}$$

The payoff for a European put option can be expressed with:

$$P=Ke^{-rT}N(-d2)-S_0 N(-d_1) \tag{7}$$

where S_0 is the initial value of the stock price, K is the strike price, T is the maturity, *r* is the risk-free interest rate and N() is the cumulative standard normal distribution function. Furthermore, d_1 and d_2 are calculated as

$$d_1=\frac{\ln\left(\frac{S_0}{K}\right)+\left(r+\frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

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

where d_1 can be interpreted as the option delta and d_2 as the risk neutral probability of the option being exercised.

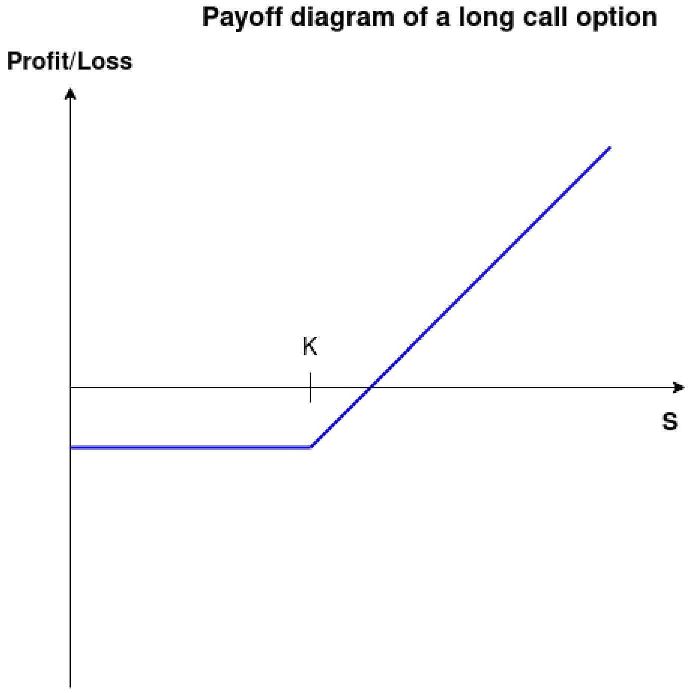
2.2 Asian options

Asian options are options where the underlying variable is the average price over a fixed period of time. The payoff of the option therefore depends on the average price of the asset under a certain time, instead of the price when the option expires. There are several different kinds of Asian options. The key differentiating characteristics between them are when they can be exercised, is the option fixed- or floating strike, and the way the average is calculated. This paper will solely focus on European-style Asian fixed strike options. Both averaging methods are discussed in the paper.

In European options, the payoff of the option at the date of maturity is the difference in the current spot price of the underlying asset and the strike price, whereas the payoff in Asian options is determined by the difference of the strike price and the average value of the underlying asset over a fixed time interval (Zhang 2009). This feature leads to certain interesting implications. By counting the difference between the strike price and the average value of the underlying asset, the payoff becomes less volatile compared to more traditional options contracts, such as European options. The reduced volatility also reduces the risk of market manipulation of the underlying instrument at maturity, as manipulating the price of the underlying asset near the maturity date would be less impactful than it would be with options that only use the spot price when the contract is exercised (Rogers & Shi 1995).

In general, the lower volatility makes Asian options less expensive than their counterparts. Asian options can be useful in delta hedging, as the delta converges to 0 as the contract approaches maturity.

Figure 1: Payoff diagram of a long call option



2.2.1 Asian fixed strike option payoffs

The payoff diagram in figure 1 works both for Asian options as well as it’s traditional counterparts. Recall that the payoff of a European call option can be expressed with the following formula:

$$C = max(S_T - K, 0)$$

The payoff of an Asian fixed strike long call option can be expressed as

$$C = max(\bar{S}_T - K, 0)$$

where $\bar{S}_T$ is the average price of the underlying asset under a fixed time period at the date of the maturity. The following section will explain the payoffs in greater detail.

The payoffs for the arithmetic average Asian options are calculated as

$$C = max(\frac{1}{T}\int_0^T S(t)\,dt - K, 0) \quad \text{or} \quad P = max(K - \frac{1}{T}\int_0^T S(t)\,dt, 0) \qquad (8), (9)$$

The continuous geometric average Asian option with:

$$C=max\left(e^{(\frac{1}{T}\int_0^T \log(s)dt)}-K,0\right) \quad \text{or} \quad P=max\left(K-e^{(\frac{1}{T}\int_0^T \log(s)dt)},0\right) \qquad (11), (12)$$

The discrete arithmetic average Asian option with

$$C=max\left(\frac{1}{m}\sum_{i=0}^{m} S(t_i)-K,0\right) \quad \text{or} \quad P=max\left(K-\frac{1}{m}\sum_{i=0}^{m} S(t_i),0\right) \qquad (13), (14)$$

The disrcrete geometric average Asian option with

$$C=max\left(e^{\frac{1}{T}\sum_0^m \log(S(t_i)\text{¿})}-K,0\right) \quad \text{or} \quad P=max\left(K-e^{\frac{1}{T}\sum_0^m \log(S(t_i)\text{¿})},0\right) \qquad (15), (16)$$

2.2.2 Pricing Asian fixed strike options

We assume that the price of the underlying asset follows a log-normal distribution continuous in time. The product of log-normally distributed values is also log-normal, while the sum is not. Hence, it is possible to use the derive the Black-Scholes model to a closed form analytic solution. The arithmetic averaging method doesn’t follow a log-normal distribution, which is why there are no closed form equations to price Asian options using arithmetic averaging (Zhang. H, 2009).

2.2.2.1 Geometric closed form solution

Following the closed form equation by Kemna and Vorst (1990) where the geometric averaging pricing solution alters the volatility and cost of carry term, the closed form equation for an Asian call and put option can be given as:

$$C=S_t e^a N(d_1)-Ke^c N(d_2) \quad \text{or} \quad P=Ke^c N(-d_2)-S_t e^a N(-d_1) \qquad (17), (18)$$

Where N() is the cumulative standard normal distribution function

$$a=b(b-r)(T-T_0)$$
$$b=\frac{1}{2}\left(r-D-\frac{\sigma^2}{6}\right)$$
$$c=-r(T-T_0)$$

where r is the risk free rate, T is the maturity date and T_0 is the beginning date.

$$d_1=\frac{\ln\frac{(S_0)}{K}+\left(b-\frac{1}{2}\left(\frac{\sigma}{\sqrt{3}}\right)^2\right)T}{\frac{\sigma}{\sqrt{3}}\sqrt{T}} \quad , \quad d_2=d_1-\frac{\sigma}{\sqrt{3}}\sqrt{T}$$

2.2.2.2 Pricing an arithmetic Asian option

As it was stated previously, the sum of log-normally distributed values is not log-normal and therefore there are no closed form solutions for pricing Asian options with arithmetic averaging. There are several different methods for trying to calculate the price of arithmetic Asian options. Hull and White (1993) applied binomial option pricing method by Cox, Ross & Rubinstein (1979) both for Eurasian and Amerasian options. Boyle (1977) was the first person to apply Monte Carlo simulations to financial derivatives. Monte Carlo methods have a fascinating history behind them, being used to simulate the behaviour concerning neural diffusion in fissionable material, among other things. The Monte Carlo method can be used not only for solving deterministic problems, deterministic problems can be solved by the Monte Carlo method, given it has the same formal expression as some stochastic process (Galda 2008). Monte Carlo simulation is highly robust and works well for solving multi-dimensional or path-deterministic problems, which is why it is also applicable to pricing Asian options. Monte Carlo simulation heavily relies on the law of large numbers, which in this context implies how obtaining a large number of results from trials, and taking the average value of them, should be close to the expected value and tends to become more accurate the more trials are performed. This can be mathematically expressed as:

$$\lim_{n \to \infty} \sum_{i=1}^{n} \frac{X_i}{n} = \bar{X}$$

Unfortunately, using Monte Carlo simulations can be highly time consuming due to the large amount of estimations required, which becomes especially devious if the estimations are complex to calculate. This paper will use Monte Carlo methods to price Asian fixed strike options, as well as use a control variate method in order to reduce the variance and get quicker results.

2.3 Monte Carlo Simulation

The Monte Carlo simulation will go as follows:

- Sample a random price path for S following Geometric Brownian Motion, under a risk neutral probability
- Calculate the payoff for the derivative per simulation
- Repeat n times to gain a sufficient sample
- Compute the mean of the derivative payoffs
- In order to gain the estimate of the value of the derivative, discount the mean with the risk-free rate

geometric Brownian motion can be expressed as

$$S_t = S_0 e^{(r - \frac{\sigma^2}{2})t + \sigma W_t}$$

Figure 2: Asset price simulations following GBM, low volatility case (std = 0.2)

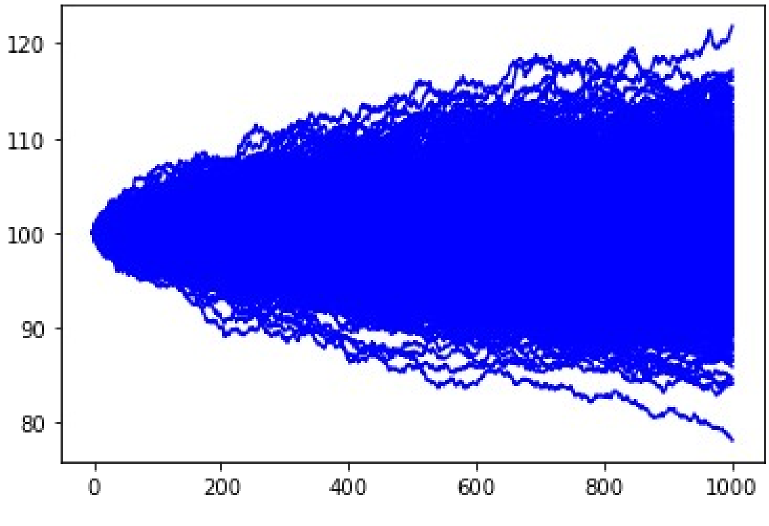
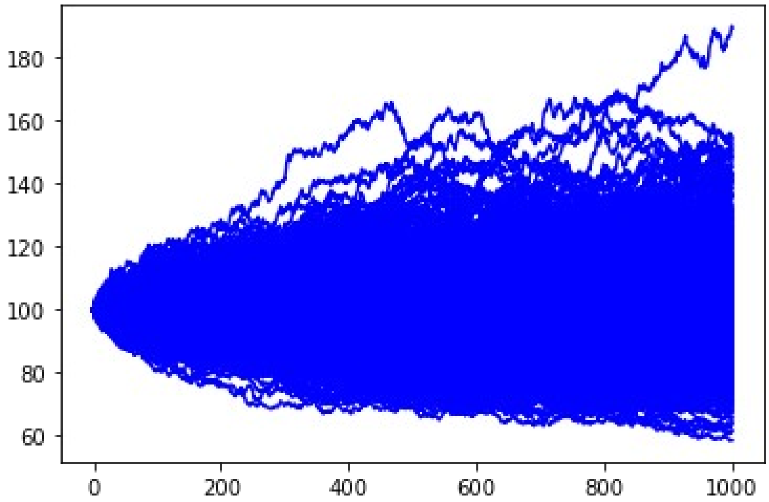


Figure 3: Asset price simulations following GBM, high volatility case (std = 0.6)



First, the payoff of an Asian arithmetic fixed strike call option is calculated with equation 13, with S_0 and K both being 100, r = 0, σ = 0.2, time to maturity 30/365 and 30 simulated steps, the option price began stabilizing after 1000 simulations, with a standard deviation of 1.93. Running 100 000 simulations yielded a call option price of roughly 1.313, with a standard deviation of 1.957. This leaves a lot of room for improvement, which is why a control variate method was applied to the arithmetic option pricing.

Following the research paper by Kemna & Vorst, a control variate variance reduction method was constructed for the Monte Carlo simulations. If the desired simulation quantity is $\theta = E(X)$, and there is another random variable Y, with a known expectation of $\mu_Y = E(Y)$, there is an unbiased estimator of θ, $Z = X - c(Y - \mu_Y)$, by linearity of expectation. It can be proved that

$$V(Z)=V(X)+c^2V(Y)+2cCov(X,Y)$$

is minimized when $c=-\frac{Cov(X,Y)^2}{V(Y)}$. Here, Y is the control variate of X. To reduce variance, a *Y* with a high correlation to *X* is chosen. In this case, *X* is the Asian option with arithmetic sampling, while *Y* is the Asian option with geometric sampling. Provided that *Y* is a satisfactory approximation of *X*, the standard deviation should reduce drastically.

The control variate is calculated with the following equations:

$$X_j=e^{-rT}[max(\frac{1}{N}\sum_{i=0}^{N}S_i-K,0)]$$

$$Y_j=e^{-rT}[max((\prod_{i=0}^{N}S_i)^{\frac{1}{N}})-K,0]$$

$$c=-\frac{\sum_{i=0}^{N}(X_i-\bar{X})(Y_i-\bar{Y})}{\sum_{i=0}^{N}(X_i-\bar{X})^2}$$

$$d=\frac{(\ln(\frac{S_0}{K})+(r+\frac{\sigma^2}{6})\frac{T}{2})}{\sigma\sqrt{\frac{T}{3}}}$$

$$\mu\gamma=e^{-rT}(e^{\varphi}S_0N(d)-KN(d-\sigma\sqrt{\frac{T}{3}}))$$

$$\varphi=\frac{T}{2}(r-\frac{\sigma^2}{6})$$

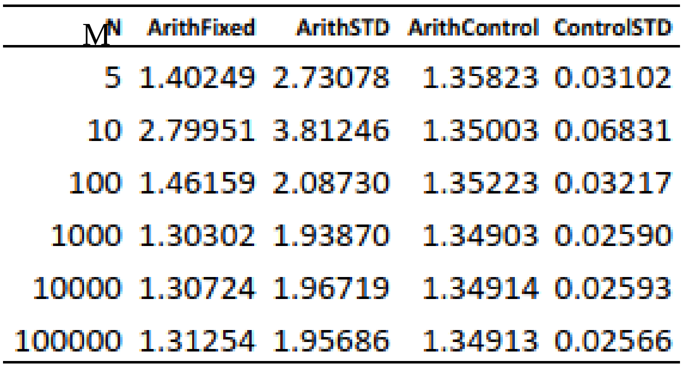
$$Z_j=X_j+c(Y_j-\mu\gamma) \qquad (19)$$

Lastly, the option price using the control variate is calculated simply as

$$C_v=\frac{1}{N}\sum_{j=1}^{N}Z_j$$

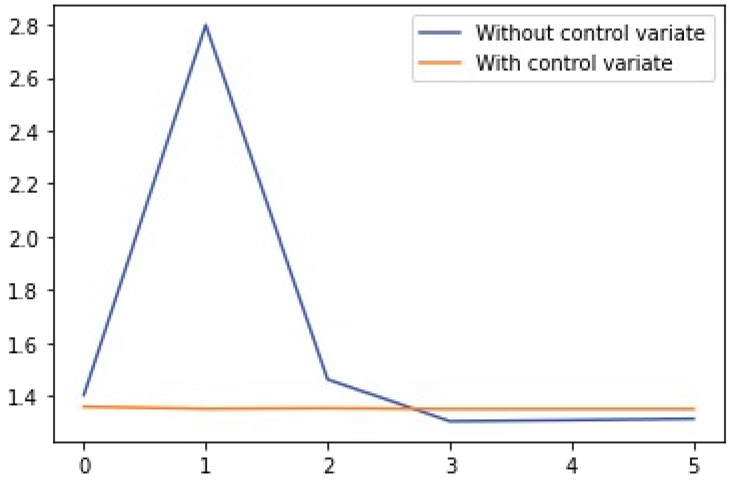
To illustrate the efficiency of the control variate method, a test was conducted where the prices of an Asian fixed strike call option were calculated using both the arithmetic average, as well as the control variate method. Monte Carlo simulations with 30 steps in each simulation were conducted with an increasing amount of simulations to see at which point the price stabilizes. The results can be seen in table 1, where N is the amount of simulations run.

Table 1: Comparison between vanilla arithmetic call option and control variate call option



MN	ArithFixed	ArithSTD	ArithControl	ControlSTD
5	1.40249	2.73078	1.35823	0.03102
10	2.79951	3.81246	1.35003	0.06831
100	1.46159	2.08730	1.35223	0.03217
1000	1.30302	1.93870	1.34903	0.02590
10000	1.30724	1.96719	1.34914	0.02593
100000	1.31254	1.95686	1.34913	0.02566

Figure 4: Price estimations of vanilla arithmetic call option and control variate call option



By looking at figure 4 and table 1, it is evident that the control variate method is highly efficient. The standard deviation is instantly almost 99% smaller, and the price is stable at around 1.5 almost instantly. Hence, when performing the deep hedging, the control variate method will be used for estimating the price of an arithmetic Asian fixed strike call option.

2.4 Delta hedging

Traditionally traders use “Greek letters” to hedge the risks associated with derivatives. Delta (*N(d$_1$))*, which is the partial derivative of the value of a transaction with respect to the underlying asset price, is the most often used of the Greek letters. Usually traders try to maintain a delta-neutral position. The delta of the option changes as time progresses and thus the position must be rebalanced at certain intervals. In optimal conditions the position would be rebalanced continuously, which could lead to the cost of hedging the option equal its theoretical price. In practice, transaction costs and other trading costs, as well as liquidity constraints imply that the strategy should be modified. In addition to these trading costs, realistically the positions are rebalanced periodically, instead of being balanced continuously (Cao, Hull, Zissis & Poulos 2019).

The delta of an option is the sensitivity of the option price to a change in the price of the underlying asset. Using the Black Scholes model option pricing formula C, delta can be expressed as

$$\Delta = \frac{\partial C}{\partial S} = N\left(\frac{\ln\left(\frac{S}{K}\right)+\left(r+\sigma^2/2\right)T}{\sigma\sqrt{T}}\right)$$

The fact that there are no closed form equations for arithmetic Asian options, and technically no delta for them, makes Asian options particularly interesting subjects for deep hedging, where the hedging costs are attempted to be minimized by other factors the Greeks.

2.5 Neural networks

Feedforward neural networks are the quintessential deep learning models. The goal of a feedforward network is to approximate a function *f**. For instance, for a classifier, *y* = *f*(**x**)* maps an input ***x*** to a category *y*. A feedforward neural network (*Neural network*) defines a mapping ***y*** = *f(**x**;**θ**)* and learn the value of the parameters ***θ*** that result in the best function approximation. These neural network models are called feedforward as there are no feedback connections in which the model outputs are fed back into itself (Goodfellow et al. 2016).

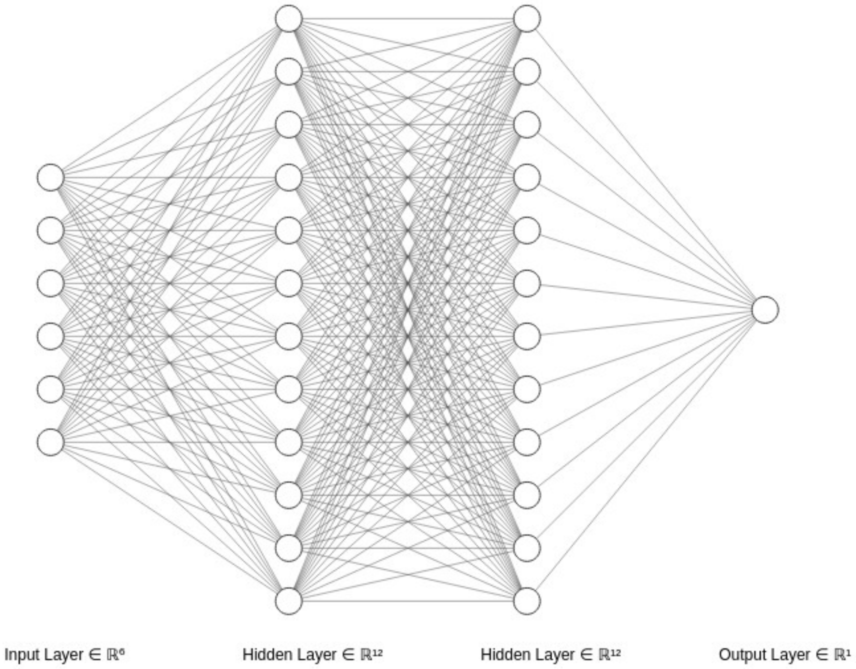


Figure 5: An example of a feedforward neural networks structure

Neural networks are called networks because they are typically a combination of multiple different functions. In figure two, there are four different functions, $f^{(1),}$, $f^{(2),}$ $f^{(3)}$, $f^{(4)}$ which are respectively the input layer $\mathbb{R}^6$, the two hidden layers $\mathbb{R}_i^{12}$ and the output layer $\mathbb{R}^1$. These are connected to a chain to form $f(x)=f^{(4)}(f^{(3)}(f^{(2)}(f^{(1)}(x))))$.

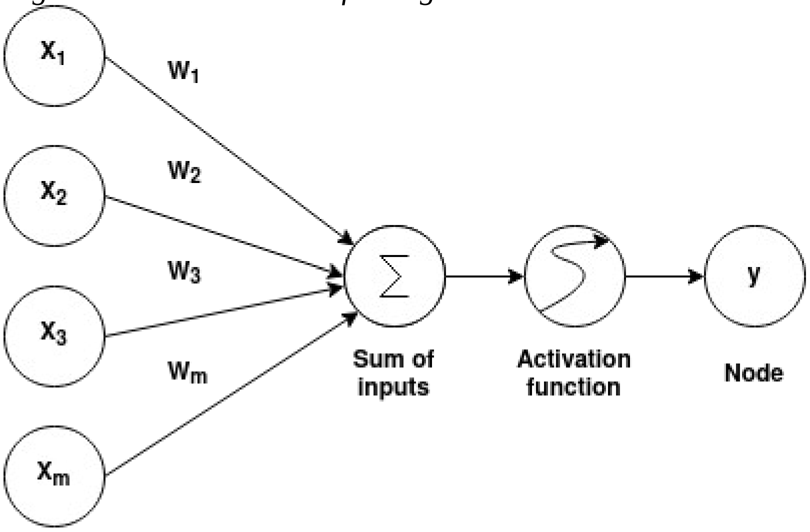
Each individual data point, apart from the input data, is called a node or a neuron in the network. A neuron is a computational unit that has m amounts of weighted input connections, and can be seen as a linear regression model as follows

$$y=\sum_{i=1}^{m} w_i x_i+b$$

where *b* is the bias, w is the weight and x is the input data.

The most appealing property of neural networks is the ability to adapt their behaviour as the characteristics of the system changes. In order for neural networks to be more accurate, the input of a node is run through an activation function before the neuron gets its final value. Without an activation function, the output signal would merely be a simple linear function which is just a polynomial degree of one. The most commonly used activation functions are non-linear functions, which are used to induce non-linearities into the network (Sharma & Sharma 2020).

Figure 6: An illustration of a single node in the network



Lastly, a feedforward neural network can be defined as follows:

Let L, N_0, N_1,…, $N_L \in \mathbb{N}$ *with L≥2, let* $\sigma : \mathbb{R} \Rightarrow \mathbb{R}$ *and for any l = 1,…,L, let* $W_l : \mathbb{R}^{N_{l-1}} \Rightarrow \mathbb{R}^{N_l}$ *an affine function. A function* $F : \mathbb{R}^{N_0} \Rightarrow \mathbb{R}^{N_L}$ *defined as*

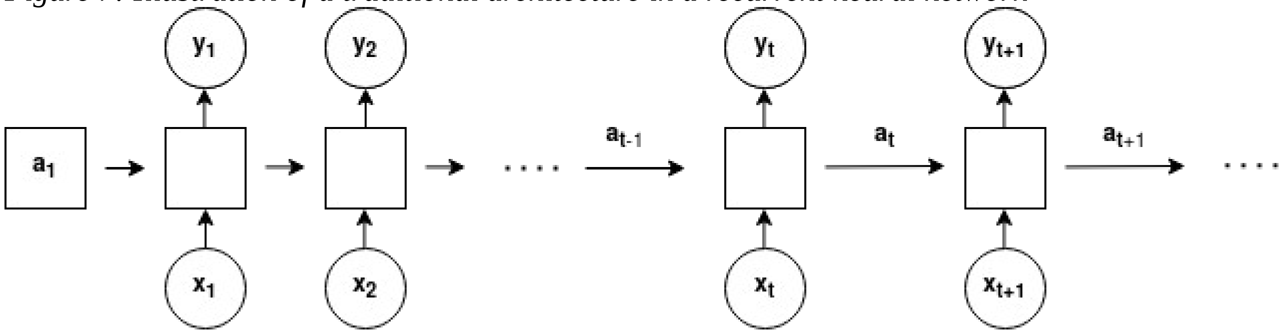
$F(x) = W_L \cdot F_{L-1} \cdot \ldots \cdot F_1$ *with* $F_l = \sigma \cdot W_l$ *for* $l = 1, \ldots, L-1$

In the definition above, σ is the activation function, *L* denotes the number of layers, N_1,...N_{L-1} denote the dimensions of the hidden layers and N_0, N_L are the input and output layers, respectively (Buehler et al. 2019).

2.5.1 Recurrent Neural Networks

One shortcoming in feedforward neural networks is that are unable to take the past into consideration. Oftentimes however, it would be useful for a model to be able to take the past in consideration. Recurrent neural networks also include feedback connections, which enable them to use past data (Goodfellow, Bengio & Courville 2016).

Figure 7: Illustration of a traditional architecture in a recurrent neural network



The hidden layer activation *a* and the output *y* at time t can be expressed as follows

$$a_t = \phi(W_{aa} a_{t-1} + W_{ax} X_t + b_a)$$

where *W* terms denote weight matrices which connect the inputs to the hidden layer *a* correspondingly, *b* denotes bias vectors and ϕ is the hidden layer function. Given the hidden sequences, the output of the sequence can be expressed as

$$\hat{y} = b_y + \sum_{i=1}^{m} W_{h^i y} h_t^i$$

$$y_t = \gamma(\hat{y}_t)$$

where γ is the output layer function. Hence, the complete network defines a function, parametrised by the weight matrices from input histories $x_{1:t}$ to the output vectors y_t (Graves 2013).

2.6 Deep hedging

In the absence of trading constraints, such as trading costs, the Black-Scholes model can yield and maintain a perfect hedge. It is evident that such an assumption does not reflect real markets. There have been several different strategies that attempt improving the Black-Scholes model under the presence of transaction costs, such as in the study of Leland (1985), where the researcher extended the Black-Scholes model by replicating a portfolio to a discretely rebalanced portfolio. Deep hedging (Buehler et al. 2019) is a ground-breaking framework, which especially in the presence of trading costs and other constraints has been shown to often outperform the traditional hedging strategies, thus drastically reducing the cost of hedging. Deep hedging represents a hedging strategy using deep learning models. In the research paper of Buehler et al. the authors used a deep semi-recurrent neural network model and demonstrated that the deep hedging model could be feasibly used to hedge a European option under exponential utility.

3 Deep heding for an Asian fixed strike call option

Following the steps of Buehler et al. 2019, a neural network model is constructed for essentially creating a pseudo-delta, which the neural network model creates and that is not related to the traditional Greeks. Additionally, the model attempts to minimize the cost of hedging.

The chosen network structure used in the research paper is inspired by the study of Beuhler et al. 2019. In order to calculate each pseudo-delta at a given step in a simulation, the following network architecture was built:

Table 2: Deep hedging neural network architecture

Input	*input*
Layer 1	*Dense(32, activation = ‘relu)*
Layer 2	*Dropout(0.5)*
Layer3	*Dense(32, activation = ‘relu’)*
Layer 4	*Dense(32, activation = ‘leaky_relu’)*
Output	*Dense(1)*

The model used Adam optimizer and a mean-squared-error loss function. After the pseudo-delta for each step in the simulation is calculated, the model tries to maintain a delta-neutral position, and ultimately minimize the cost of hedging. This can be described as:

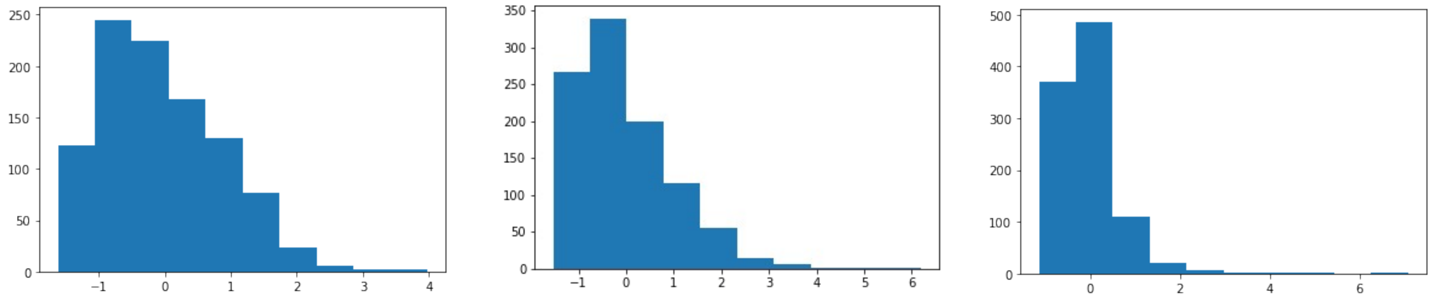
$$[\sum_{i=0}^{n} \Delta_i (S_i - S_{i+1})] + max(A_T - K, 0) \approx C \tag{20}$$

where *Δ* is the pseudo-delta estimated by the neural network model, A_t is the arithmetic average of the price sequence *S*, and *C* is the price of an Asian fixed strike call option calculated with the control variate method (*see equation 19*). For the scenarios tested out, S_0 is 100, time to maturity is 30/365, and the risk-free interest rate is 0. Each simulation that is run consists of 30 steps.

3.1 Results

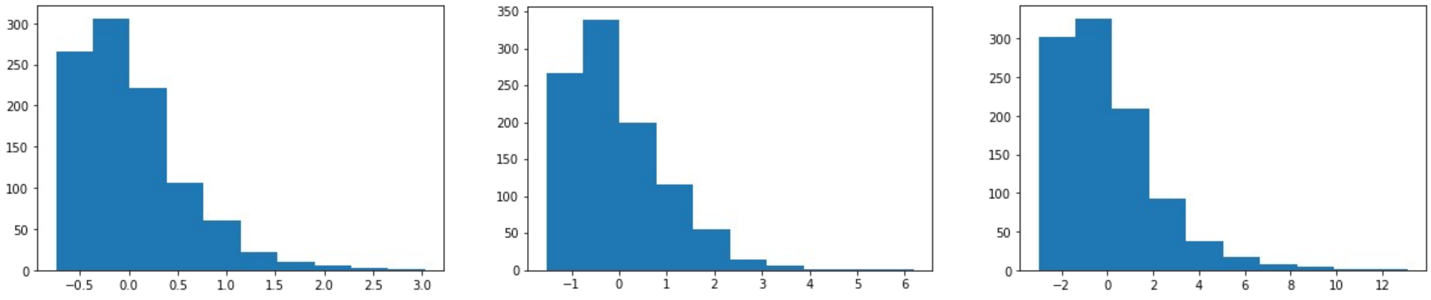
The performance of the model was tried out in different conditions. The validation results from different scenarios are displayed below.

Figure 8: Validation results with different strike prices



a) K = 97 *b) K = 100* *c) K = 103*

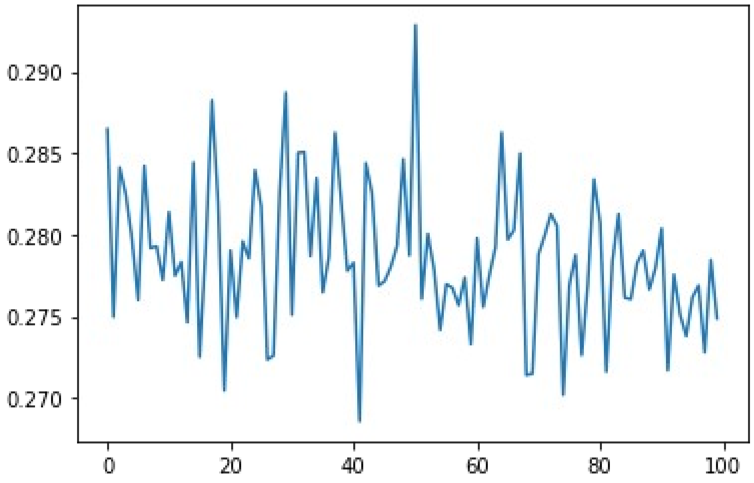
Figure 9: Validation results with different volatilities



a) σ = 0.1 *a) σ = 0.2* *a) σ = 0.4*

From the results above, it can be seen that the adapts well into different scenarios. The most challenging situations for the model seem to be higher volatility situations, which is natural, since that's the expectation for all hedging models. However, it is noteworthy that the hedging results could probably be made significantly better. By further analysing the model performance, it can be seen from figure 8 that the mean-squared-error of the model is around 0.27, regardless of the amount of epochs. This could be due to slight inaccuracies in the Monte Carlo pricing method, even when using the control variate.

Figure 8: MSE loss function at different epochs



4 Conclusions

This research paper discussed deep hedging methods with a focus on Asian fixed strike options. Different pricing methods for Asian options were introduced and analysed, whereafter the fundamental concepts of neural networks were reviewed. The empirical part of the paper aimed to hedge Asian options by creating a pseudo-delta and trying to minimize the profit and loss function for hedging. The neural networks model does a good job at estimating the pseudo-delta, but further improvements could be made to increase the accuracy of the model. Perhaps a binomial option pricing method done in a similar fashion as in the research paper of Hull & White (1993), would lead to a more accurate option price, thus improving the accuracy of the model (*recall chart 20*). Alternatively, more focus could be laid on optimizing the neural network model by hyperparameter tuning and building pipelines for creating various models. Recurrent neural networks could be an interesting alternative, as they take the past actions into consideration.

References

P. Glasserman, Monte carlo methods in financial engineering, Applications of mathematics : stochastic modelling and applied probability, Springer, 2004.

Goodfellow, I., Bengio, Y. and Courville, A., Deep Learning, 2016
(MIT Press). Available online at: http://www.deeplearningbook.
Org.

Lakhlani, V. B. (2013). Pricing and Hedging Asian Options.

Zhang, H. (2009). *Pricing Asian Options using Monte Carlo Methods*.

Joy, C., Boyle, P. P., & Tan, K. S. (1996). Quasi-Monte Carlo methods in numerical finance. *Management science*, *42*(6), 926-938.

Galda, G. (2008). Variance Reduction for Asian Options.

Cox, J. C., Ross, S. A., Rubinstein, M., “Option Pricing: A Simplified Approach”, Journal of Financial Economics (1979)

Vajjha, K. (2017). Module 3 Exam Solutions.

Hull, J. C., & White, A. D. (1993). Efficient procedures for valuing European and American path-dependent options. *The Journal of Derivatives*, *1*(1), 21-31.

Black, F., & Scholes, M. (2019). The pricing of options and corporate liabilities. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Volume 1: Foundations of CCA and Equity Valuation* (pp. 3-21).

Rogers, L. C. G., & Shi, Z. (1995). The value of an Asian option. *Journal of Applied Probability*, *32*(4), 1077-1088.

Kemna, A. G., & Vorst, A. C. (1990). A pricing method for options based on average asset values. *Journal of Banking & Finance*, *14*(1), 113-129.

Boyle, P. P. (1977). Options: A monte carlo approach. *Journal of financial economics*, *4*(3), 323-338.

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

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, *6*(12), 310-316.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Leland, H. E. (1985). Option pricing and replication with transactions costs. *The journal of finance, 40*(5), 1283-1301.