Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220659

Research on the Applications of Neural Network Algorithms in Deep Hedging

Chenjia Jin[1]

[1]University of Liverpool, Liverpool, UK

sgcjin3@liverpool.ac.uk

Abstract. Under market completeness assumptions, hedging a portfolio of derivatives is straightforward. In view of friction, transaction costs, liquidity and other factors, a framework is presented to extend the pricing and hedging with the hedging strategy treated as a neural network. We study the deep hedging model under incomplete market constraints such as frictions, traction cost, permanent impacts on the market and illiquidity. We discuss the limitations of certain models concerning the applications in deep hedging with constraints. After which, we analyse the advantages of different models and their joint models and find that the hedging strategy is close to the Black-Scholes delta hedging strategy. An example is also given when training after designing two hedging models. The Black-Scholes delta hedging is indeed approximated by unsupervised learning.

Keywords: hedging, machine Learning, the Black-Scholes model, the Brownian motion

1. Introduction

Markets are suffered from imperfections especially when restrictions are observed during transactions or research is conducted based on its properties. In quantitative finance, pricing and hedging are fundamental topics in risk management and financial mathematics. Risk-neutral gives us an idea and a tractable solution under idealized markets where few factors are considered. When factors such as friction, liquidity and transaction costs are added to the model or the leading order needs to be explicitly decided, a model is needed which is precise and complex to emulate real-life trading situations with empirical data. However, even with recent models such as robust-hedging[1] and super-hedging[2], few solutions can be scaled independently and survive with a large portfolio. Combining the properties of deep learning and the techniques of mathematical models, the respective algorithms are model-free which is more suitable as a model when studying the markets.

On one hand, we can see from the evaluation by Shi et al.[3] of the FBSDE solver developed by Han et al.[4], the solution is able to be found with a supervised learning framework. They found that the FBSDE solver performed well under short time horizons whereas the deep hedging algorithm yielded a stable while reliable for both short and fairly long-time horizons. On the other hand, the deep hedging model pioneered by H. Buehler et al.[5] addresses the problem of the lack of efficient alternatives in complete market models. As mentioned by Shi et al.[3], the FBSDE solver's algorithm has a problem with the time horizon and discrete-time when scaling. Their approach is targeting the utility function, training and learning the optimal trading strategy directly with reinforcement learning algorithms[6].

DOI: 10.54254/2755-2721/2/20220659

As part of the examination of the deep hedging algorithm, a model is constructed which approximates the Black-Scholes model under continuous-time to approximate the deep hedging method under continuous time. Under this assumption, it indeed tends to the continuous-time Black-Scholes under rescaling of time and interpolation. Combined with Brownian motion [7], our model can be trained to see if it is suitable for hedging the call option. After modelling the profit and loss function as a function of the trading strategy and the change in price under a time sequence, we are able to match the trading strategy as a neural network in deep hedging. Considering that the deep hedging model works in discrete time, the model is capable of approximating the result with other models under certain constraints. At the end of this work, we have derived the Black-Scholes delta hedge by unsupervised learning. Compared with the delta hedge, we have shown that it is a good approximation which the model is approximate time normalization. By taking a smaller time interval, the hedging strategy tends to be a replication with a trained model.

This paper also discusses the data pre-processing procedures of a deep hedging model and examines the pricing and hedging with friction markets in discrete time under certain measures. Additionally, in order to better compare the result of the Black-Scholes delta hedge strategy with the deep hedging model, graphs are generated to visualize under various parameters.

2. Related works

Deep Hedging is a relevantly new topic emerging in recent years, it is an application of neural network algorithms in quantitative finance. There are many works [3, 5, 8] concerning applications of neural network algorithms in deep hedging. For example, deep generative methods such as Time Series Generative Adversarial Network which preserves the temporal dynamics property for time-series data, are used in commodity markets in which a Feedforward Neural Network named "deep hedger" is used to approximate an optimal policy [8]. Hence, the application of neural network algorithms in deep hedging is a fundamental element to be considered.

In light of the hedging strategy that experts are looking for, researchers Buehler et al. [5] submitted the idea that the hedging strategy can be represented by a neural network [5]. In this model, the inputs are current and past market data, and the optimal Profit and Loss of a hedger can be gained after training the network under certain measures. Although the approach is in principle model-free due to the properties of neural networks, the available historical price paths for training may not be enough to feed the general model and thus, a model used to generate the training data could be an effective way to provide sufficiently many data. Another problem under this assumption could be the impact on the result given different calibrated data. The result shows that when synthetic time series is used in pricing derivatives and derivation of corresponding hedging strategies, two different hedging strategies and policies could be derived from two different calibrated samples[9]. As a result, the model that produces the training data should be calibrated to market data or data can be otherwise generated from another market simulator.

As a remark of a concept in deep hedging, continuing from the proposition that the approach is model-free, we can include market frictions [3]. Considering the applications of deep reinforcement machine learning methods [6] as typical example models with the presence of market frictions, only generator, loss function and trading instruments are additionally needed [9] that depend on no market dynamics. And Shi et al. [3] have developed an algorithm that yields optimum even in high dimensions which primarily depends on the number of hedge instruments [5]. In the neural network, optimization is performed under some performance measures such as hedging error, risk measure or utility function with a set price path. This paper also examines the framework of pricing and hedging using convex risk measures with market frictions in discrete time.

Specific examples with certain conditions are examined in order to acquire a better understanding of some of the applications in deep hedging which is useful when generalizing to more general cases as well as used to improve and optimize models. This paper compares the deep hedging strategy with the corresponding Black-Scholes delta hedging strategy. Since deep hedging works in discrete time, a model can be constructed to approximate the continuous-time Black-Scholes model. This paper also compares the deep hedging model created by Buehler et al. [5] with other models and analyses the advantages of combined models.

3. Methods

The approach of deep hedging is a realization of the idea of treating hedging strategy as a neural network. As mentioned above, we input historical data or data from other market simulator and manage to optimise the *Profit and Loss* of the hedger under some performance measure with price paths which is shown below.

3.2. Model establishment

Here, we go through the classical Black-Scholes delta hedging strategy with the deep hedging approach. One advantage of using deep hedging is that, in terms of the imperfection of markets such as the friction of markets, illiquidity, permanent impact on the market and transaction cost, it is relatively convenient to include more information or calibrated data.

Since the deep hedging strategy is effective in discrete time, we set up the price process in a way that enables our corresponding model approximates to the Black-Scholes delta hedging model [10]. The price process is shown in Equation (1), where we have positive constants $\mu > 0, \sigma > 0$ and $S_0 > 0$, $\xi_1, \ldots, \xi_T$ are mutually independent random variables which is $N(0, \frac{1}{T})$ distributed.

$$S_t = S_0 \cdot e^{(\mu \frac{t}{T} + \sigma \sum_{i=1}^{t} \xi_i)} \tag{1}$$

As T tends to infinity, under rescaling and interpolation, the process S tends to continuous-time Black-Scholes price process as expected which is given below:

$$S_t^* = S_0 \cdot e^{\mu t + \sigma W_t} \tag{2}$$

In this formula, $W = (W_t)_{t \in [0,1]}$ is the standard Brownian motion. We set the values of the parameters μ =0.1, σ =0.5, T =100, S_0 =1.

Since the deep learning method is applied in this model, we set N = 100000 the number of independent samples of price path S for training purposes.

Our aim here is to hedge the call option which is expressed as:

$$(S_T - K)^+ \tag{3}$$

We are in search of an adapted and self-financing trading strategy for the underlying asset whose terminal value at time T is as close as possible to the payoff of the option. Denote every trading strategy that is found by its initial value $x \in \mathbf{R}$ and its position p_t for the intended asset. Because it is an adapted process, p_t has to be a function of the past prices $S_t, S_{t-1} \ldots S_0$ only. Assuming zero interest rate applies here, we can write out the final value of the strategy as

$$V_T = x + \sum_{t=1}^{T} p_{t-1}(S_t - S_{t-1}) \tag{4}$$

Hence, the Profit and Loss of the option hedger would therefore be:

$$Profit\ and\ Loss = V_T - (S_T - K)^+ \tag{5}$$

Since x and S_0 are fixed, we can treat Profit and Loss as a function of $p_0, p_1, ..., p_{T-1}$, also we define the difference in prices as $\Delta S = S_t - S_{t-1}, t \in \mathbf{Z} \geq 1$, then we can express the Profit and Loss as a function shown below,

Profit and Loss

$$(p_0, p_1, ...p_{T-1}; \Delta S_1, \Delta S_2, ..., \Delta S_T) = x + \sum_{t=1}^{T} p_{t-1}\Delta S_t - (S_0 + \sum_{t=1}^{T} \Delta S_t - K)^+ \quad (6)$$

Now, we represent the trading strategy p or π_t as a neural network with inputs available historical and current market data and outputs the corresponding hedge as below,

$$p_t = f_t(S_t, S_{t-1}, ..., S_0) \quad (7)$$

Here we can simplify our problem without a loss of generality, we search for a single network $f:[0,1] \times \mathbf{R} \to \mathbf{R}$ such that $\pi_t = p_t = f(\frac{t}{T}, S_t)$ with $t = 0,1,...,T-1$, where f_t is a neural network f for the time sequence t from 0 to $T-1$ and the price process S_t is a Markov process.

After reviewing the property of the given and output data, we can specify our function f as below:

$$f \in N_r \ (2, 100, 100, 100, 1;\ ReLU, ReLU, ReLU, Sigmoid) \quad (8)$$

where r = 4 gives a four activation functions neural network. Here we use the Sigmoid simply because the hedging position is 0 or 1 intuitively. This choice is not necessary, but it helps optimize the result.

3.2. Model optimization

We train our f with independently chosen samples $S^i = (S_t^i)_{t=0}^{T}, i = 0, ..., N-1$.
Our objective here is to find the quadratic hedging error, namely when the square of the Profit and Loss is empirically minimized. So, we define a loss function,

$$l((y_0, y_1, ..., y_{T-1}), (y_0, y_1, ..., y_{T-1})) := \text{Profit and Loss}(y_0, y_1, ..., y_{T-1}; y_0, y_1, ..., y_{T-1})^2 \quad (9)$$

We fix x as the respective Black-Scholes call price BS $(S_0, K, 1)$. Since the loss function l is customized, it needs to be implemented separately in Keras. Then, we train the network using Adam with minibatch of size 100 over 4 epochs.

In training, the i-th sample is

$$y_t = f(\frac{t}{T}, S_t^i),\ y_t = \Delta S_{t+1}^i \quad (10)$$

Therefore, we need to evaluate the outputs $f(\frac{t}{T}, S_t^i), t = 0,1,...,T-1$ for samples $i = 0,1,...,N-1$ at the same time. By supplying the features $(\frac{t}{T}, S_t^i)$ as a higher-dimensional array in Keras, we can get the result.

DOI: 10.54254/2755-2721/2/20220659

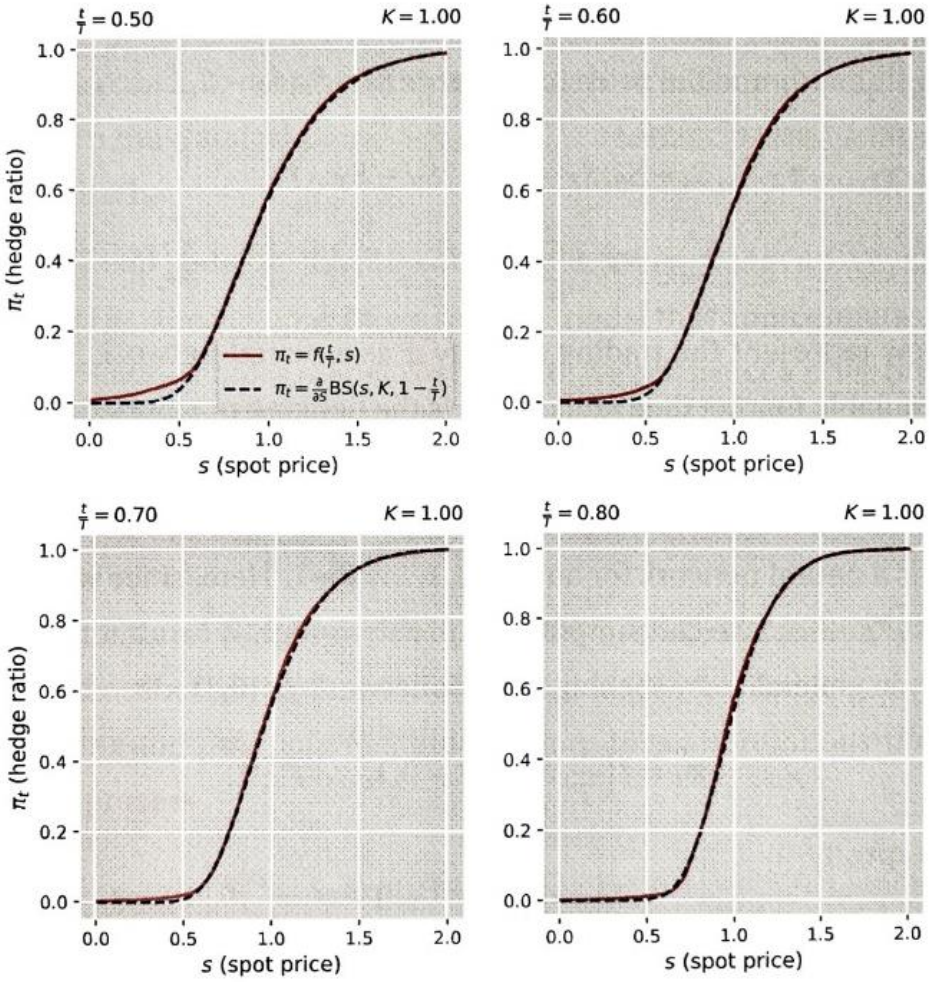

Figure 1. The Deep Hedging strategy and The Black Scholes hedging strategy.

Figure 1 shows a good approximation for enough data. We can see that for large T, the price process S approximates the Black-Scholes price process S^*, the hedging strategy $p_t = f(\frac{t}{T}, S_t)$ should be close to the Black-Scholes' delta hedging strategy.

$$p_t^* = \frac{\delta}{\delta S} BS(S_t, K, 1 - \frac{t}{T}) \tag{11}$$

This shows a perfect replication which yields zero Profit and Loss. It can also be seen from the graph for the trained network f.

4. Discussions

Having considered the approach of treating trading strategy as a neural network, we built a model around this concept which showed the application of deep learning in deep hedging. Unlike robust-hedging or super-hedging where few solutions can be independently scaled and considered with large portfolio, this model approximates from discrete-time to continuous-time with rescaling of time and interpolation as T tends to infinity. Since neural network algorithm is implemented in deep hedging, we can take advantage of the property of it which is that imperfections mentioned can be incorporated directly into the framework and that is hard to work with from the analytical point of view.

As mentioned previously, with the help of neural network algorithm, we are able to cope with the imperfection of the market, the model works well even with the emergence of friction, transaction cost, illiquidity or permanent impact on the market. The model showed a good approximation to the Black-Scholes delta hedge by deep learning which is learnt by the network itself through *Profit and Loss* optimisation. The model yields a trading strategy which considered imperfections of markets, the modification of the model would depend on the purposes of the model, thus different generators can be used, or the model can be set under convex transaction costs on trading rates. The approach is the same since the optimum solution is calculated with a trained learning-based algorithm. Then the outcome is compared with a benchmark, for example, the Black-Scholes hedge strategy, for further conclusions.

As for the pre-processing procedure of data used for a deep hedging model, we can perform a calibration to market data on a model which generates training data or use some other market simulator.

When considering the potential problem of overfitting, it can be treated with a regularisation method by introducing dropout. A dropout layer is added after each hidden layer with a rate $p \in [0,1]$ creating a new network. Although the training loss remains bigger than the model error variance of 0.1, the validation loss is close to the training loss.

Since we are considering the call option here, we assume the pricing hedging duality gap is preserved, otherwise, we may link it with strict local martingales which are related to trading restrictions and current market prices. This content would be beyond the scope of the paper.

5. Conclusions

We have discussed the application of neural network algorithm in deep hedging which is shown by a model that approximates the classic Black Scholes hedge strategy in the incomplete market. The neural network algorithm has such a wide application in deep hedging and potential in other fields. Optimised approaches of solutions would have varied impacts on the program running time which is also the case from the aspect of the numerical method or operational research. Deep learning will have more applications in the financial industry not limited to finding the optimal trading strategy in deep hedging. Multi-factors analysis and backpropagation with other classic neural network algorithms can also be implemented in the stock price forecast and so on. Deep hedging is a relatively new topic in the deep learning community, more correlations and applications would be found for joint subjects.

References

[1] Y. Dolinsky, H. M. Soner, Martingale optimal transport and robust hedging in continuous time, Proceedings of Probability Theory and Related Fields, Springer, 2014, pp. 391-427. DOI: https://doi.org/10.1007/s00440-013-0531-y

[2] M. Nutz, H.M. Soner, Superhedging and Dynamic risk Measures under Volatility Uncertainty, Proceedings of the Industrial and Applied Mathematics, vol. 50, SIAM Journal on Control and Optimization, 2012. DOI: https://doi.org/10.1137/100814925

[3] X. Shi, D. Xu, Z. Zhang, Deep Learning Algorithms for Hedging with Frictions, arXiv preprint, vol. 2112.04553, arXiv, 2021. DOI: https://doi.org/10.48550/arxiv.2111.01931.

[4] J. Han, A. Jentzen, W. E., Solving high-dimensional partial differential equations using deep learning, in: Proceedings of the National Academy Sciences, 2018, pp. 8505-8510. DOI: https://doi.org/10.1073/pnas.1718942115

[5] H. Buehler, L. Gonon, J. Teichmann, B. Wood, Deep hedging, Quantitative Finance, vol. 19, No.8, 2019, pp. 1271-1291. DOI: https://doi.org/10.1080/14697688.2019.1571683

[6] J. Cao, J. Chen, J. Hull, Z. Poulos, Deep Hedging of Derivatives Using Reinforcement Learning, in: The Journal of Financial Data Science Winter, arXiv, 2021. DOI: 10.3905/jfds.2020.1.052

[7] A. Löffler, L. Kruschwitz, The Brownian Motion A Rigorous but Gentle Introduction for Economists, Springer International Publishing, 2019. DOI: https://search-ebscohost-com.liverpool.idm.oclc.org/login.aspx?direct=true&db=cat00003a&AN=lvp.b5649694&site=eds-live&scope=site

[8] M. Kozyra, Deep learning approach to hedging, Oxford, 2018. DOI:https://www.maths.ox.ac.uk/system/files/media/michal_kozyra.pdf

[9] N. Boursin, C. Remlinger, J. Mikael, C. A. Hargreaves, Deep Generators on Commodity markets; application to Deep Hedging, Proceedings of the Quantitative Finance, arXiv, 2022. DOI: http://arxiv.org/abs/2205.13942X

[10] M. Mastinšek, Discrete-time delta hedging and the Black-Scholes model with transaction costs, Proceedings of the Mathematical Methods of Operations Research, Springer, 2006, pp. 227-236. DOI: https://doi.org/10.1007/s00186-006-0086-0