QF634 Group Project

Application of Artificial Neural Networks in Option Pricing and Hedging

Team members:

JERIEL WONG KWAN KIT (G2)

QU WENQIN (G2)

15 December 2022

Abstract

In recent studies, the non-parametric models such as artificial neural networks (ANN) achieve similar pricing performances than the parametric models such as Black-Scholes (BS) pricing formula. This report compares the ANN and the BS pricing formula, regarding both pricing and delta hedging strategy performance, and aims to explore if the positive results of the previous literature can be replicated using more recent real-world equity options data, specifically the European Volatility Index call options from January 2021 to December 2021.

The result shows that the performance of the ANN is close to that of the BS pricing formula in option pricing and delta hedging performance when using the recent real-world European call options data.

Introduction

Since the 1990s, neural networks have been studied as a non-parametric method for option pricing and hedging. Many studies focus the approximation performance by the ANN on either simulated or real datasets. Various inputs and outputs, performance measures, data splitting methods, regularization techniques and underlying assets have been used and discussed, and often the ANN are compared to a variety of benchmarks, the simplest one being the BS pricing formula.

The dependencies of the BS Model are typical examples of traditional parametric models. For example, the volatility has an essential role in the BS Model and therefore, it is important to compare different volatility estimation models to guarantee the lowest mispricing in the BS pricing prediction. Hence, an inaccurately specified BS formula could lead to pricing errors. Since changes in economic and market conditions may influence the option price, introducing a data-driven and non-parametric model such as the ANN could reduce the pricing error by learning the dynamics behind the option price and minimising the uncertainty in parametric assumptions.

A good pricing prediction is not sufficient to support the practical relevance of the non-parametric model. Since the BS is a no-arbitrage-based pricing formula, it is important to prove the ability of the model in replicating option through a dynamic hedging strategy, for instance, delta hedging.

The rest of the report is organised as follows:

- Literature Review summarises previous studies that concern the use of ANN for nonparametric pricing (and hedging) of options;
- Part I Option Pricing Method includes data preparation and model set-up for option pricing;
- Part I Option Pricing Results presents the pricing performance comparison;
- Part II Delta Hedging Method includes the approach to perform delta hedging;
- Part II Delta Hedging Results presents the delta hedging performance comparison;
- Conclusion discusses the results and suggests possible future work.

Literature review

Several previous literatures have compared the performances of the BS pricing formula and ANN based models in option pricing. The majority have illustrated that generally ANN models can predict option prices from real data with higher accuracy. Hutchinson et al. (1994) is one of the first papers that proposes to use the ANN method to pricing options in a non-parametric way and to challenge the use of parametric BS formula and introduce a meaningful methodology to evaluate the hedging performance over multiple periods. Subsequently, many researchers started exploring the ANN methods on option pricing and obtained similar positive results to the previous researchers.

Ruf and Wang (2020) provide a comprehensive review on the academic literature of the ANN based option pricing and hedging with a statistical perspective. As shown in the Table 1 below, they summarise a number of relevant literatures and compared several characteristics that describe how each paper approaches the pricing or hedging, including input features, outputs of the ANN, benchmark models, data splitting method between training and test datasets, and the underlying assets along with the time span of the data. It is observed that as an overall trend, more recent papers use more complex methods, which is in line with improved availability of computational resources.

Table 1. Characteristics that describe how literature approaches the pricing or hedging

Components	**Key Observations**

Input Features	- Using the underlying price, St and the option strike price, X separately
	- Only using their ratio, St/X (moneyness)
	  - Reduces the number of inputs and thus makes the training of the ANN easier
	  - Helps generalization and reduces overfitting
	- Time to maturity, volatility (historical, implied, GARCH generated, etc.), interest rate

Outputs of ANN	- The most common output is the option price
	  - Depending on whether underlying price and strike price are used separately or moneyness is used, the output can be the option price, or the option price divided by the strike price
	  - Hybrid ANN; validate to hedging errors; regulation techniques
	- Implied volatility - converted to option prices by the Black-Scholes formula
	- Sensitivity / Hedging ratio

Benchmark Models	- Single period: MAE, MAPE, MSE
	- Multiple period: mean absolute tracking error (MATE) and prediction error (PE)
	- Compare to a benchmark: parametric pricing model
	  - The most widely used benchmark is the BS pricing formula
	  - Stochastic volatility pricing models

Data Partition	- Chronologically - the early data constitutes the training set, and the late data constitutes the test set - Randomly - breaks the time structure and introduces information leakage between the training set and the test set

Underlying Assets	- Simulation data - free of noise and sometimes a close-to-optimal solution is available as a benchmark - Most other papers use either both simulation and real data or only real data (options on S&P500, FTSE100 and S&P100)
Regularisation Techniques	- As the advance of hardware allows for bigger ANNs to be built, regularization techniques have become more important as part of the ANN training. (e.g., L2, dropout, early stopping, etc.)

5-year S&P 500 future option data (January 1987 – December 1991) was used to study option pricing and hedging performance. Non-parametric ANN models were implemented using three different algorithms:

1. Radial Basis Function (RBF) with 4 multiquadric centers and an output sigmoid

2. Projection Pursuit Regression (PPR) with four projections

3. Multi-Layer-Perceptron (MLP) with one hidden layer of four neurons and the two

input variables.

Part I Option Pricing Method

Data Preparation

- Daily Options: OptionMetrics – Wharton Research Data Services
- Forward Prices: OptionMetrics – Wharton Research Data Services
- Interest Rates: Federal Reserve Economic Data (FRED)

The 3-month U.S. Treasury Bill was extracted from (FRED) and mapped to the option's traded month. The sole purpose for the 3-month U.S. Treasury Bill was to calculate the present value of the underlying price (VIX), as the team wants to ensure data vendor consistency for options data.

Model Build Up

With the intuition from Black Scholes Call Option Price = $C(S_t , K, t , r_t, s_t)$, this parametric model requires 5 inputs, as compared to the machine-learning model which only requires 2 inputs. Without the use of (r_t, s_t) parameters, no statistical distribution is required to be implied on the remaining 2 features, as such, the Artificial Neural Network = $C(S_t/K, t)$

The team used two 2 input features from OptionMetrics. Feature 1, Stock / Strike (S/X) was produced given the daily trading day's underlying price (VIX) which was reversed calculated using the monthly interest rate from FRED, to find the present value of the underlying price (VIX).

Feature 2, Time to expiration (T- t) was calculated using the Maturity Date and the current Trading Date to find the day count difference, and then converted to per unit of 1 year (252 trading days)

The team then instantize the Artificial Neural Network model from Tensorflow's Keras library, with the settings of 1 layer with 4 nodes and 1 output layer with 1 node.

Part I Option Pricing Results

Table 2 below presents the performance comparison results on the test dataset, between the pricing prediction of the BS model (y_BS/X) vs. the market (y_C/X) and the pricing prediction of the ANN (yhat_NN/X) vs. the market (y_C/X) cross different moneyness and time-to-maturity, considering Mean Square Root of Error (MSE), Root of Mean Sqaure of Error (RMSE) and Mean Absolute Error (MAE). The differences between performance measures of BS and ANN are generally small, showing a similar performance. Comparing the results, the ANN model performs relatively better for ATM, OTM, and Long Maturity cases.

Table 2. Pricing performance on the test dataset

	test	mse_BS	rmse_BS	mae_BS	mse_NN	rmse_NN	mae_NN	mse_diff (BS-NN)	rmse_diff (BS-NN)	mae_diff (BS-NN)
0	All	0.010340	0.101684	0.074990	0.012491	0.111763	0.049619	-0.002151	-0.010079	0.025371
1	ITM	0.014707	0.121273	0.095716	0.021010	0.144947	0.060215	-0.006303	-0.023674	0.035502
2	OTM	0.004931	0.070220	0.049104	0.000156	0.012482	0.008613	0.004775	0.057738	0.040491
3	ATM	0.013664	0.116894	0.095032	0.000246	0.015677	0.011453	0.013419	0.101217	0.083579
4	Short Maturity	0.000756	0.027494	0.016124	0.005939	0.077065	0.025775	-0.005183	-0.049571	-0.009651
5	Medium Maturity	0.002478	0.049778	0.036729	0.007921	0.088998	0.030840	-0.005443	-0.039220	0.005888
6	Long Maturity	0.023649	0.153781	0.144419	0.017481	0.132215	0.046418	0.006168	0.021566	0.098001

Figure 1 and 2 below are the graphical representation of the entire test dataset results where the predicted price (y_BS/X, yhat_NN/X) is plotted as a function of the actual price (y_C/X). Similar trend is observed when applying to the train datasets. Technically for the best results is a diagonal linear line, where the actual price equals the predicted price as unit increases. The variance of the prediction price of the BS model is relatively larger than that of the ANN model. However, while the ANN predicts well from unit 0 to 1, the predicted price converges to 1 after unit 1. We further investigate the dataset and observe that in Figure 3 the C/X distribution in the dataset, the majority of the observations have C/X < 1 in the data set. Since the ANN is a data driven model, the model would learn the features associated with the most common type of data in the data set. Therefore, It can be argued that the mispricing is due to data limitation.

Figure 1. Predicted price vs Actual price (BS)

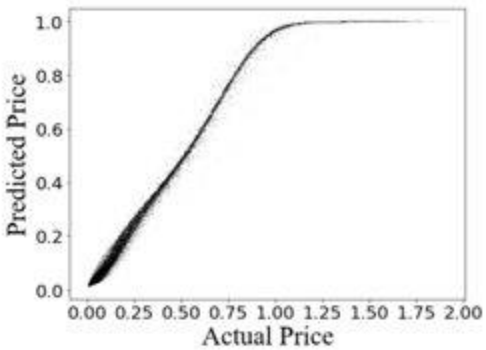


Figure 2. Predicted price vs Actual price (ANN)

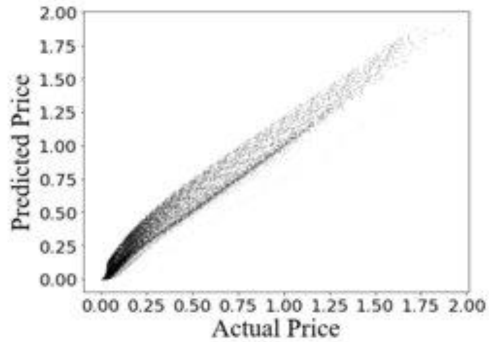
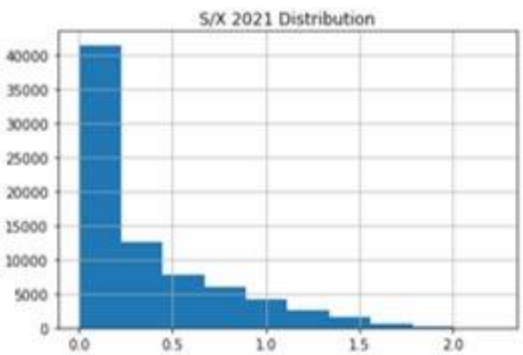


Figure 3. C/X distribution in the dataset



We further investigate the pricing performance by moneyness and maturity. Figure 4 represents the pricing error as a function of moneyness. The red horizontal line divides ITM options (above) from OTM options (below). For the BS model, the dispersion on the left side of the mean is bigger in magnitude. Therefore, the options tend to be more underpriced by the BS model than the ANN.

Figure 4. Moneyness vs. Pricing error

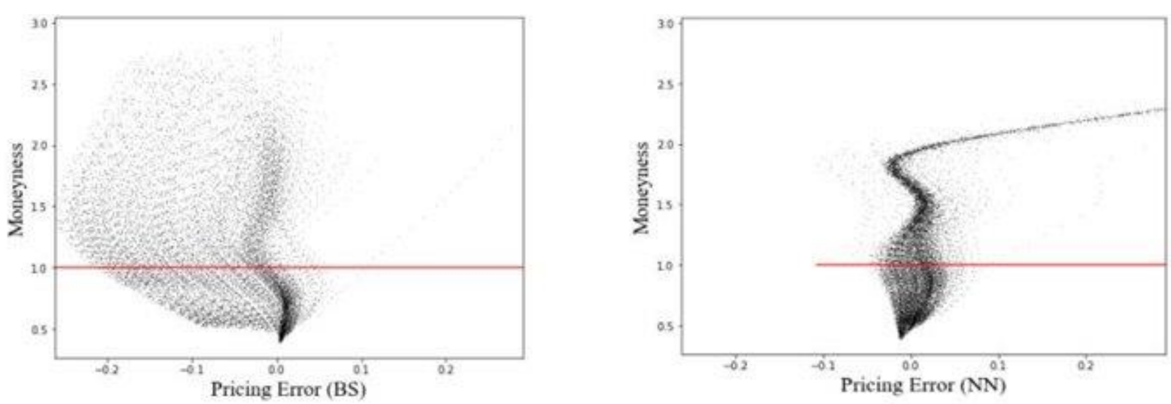


Figure 5 represents the ANN pricing errors as a function of time-to-maturity. It is observed that the ANN pricing error is centered at 0, whereas the BS pricing error has larger variance.

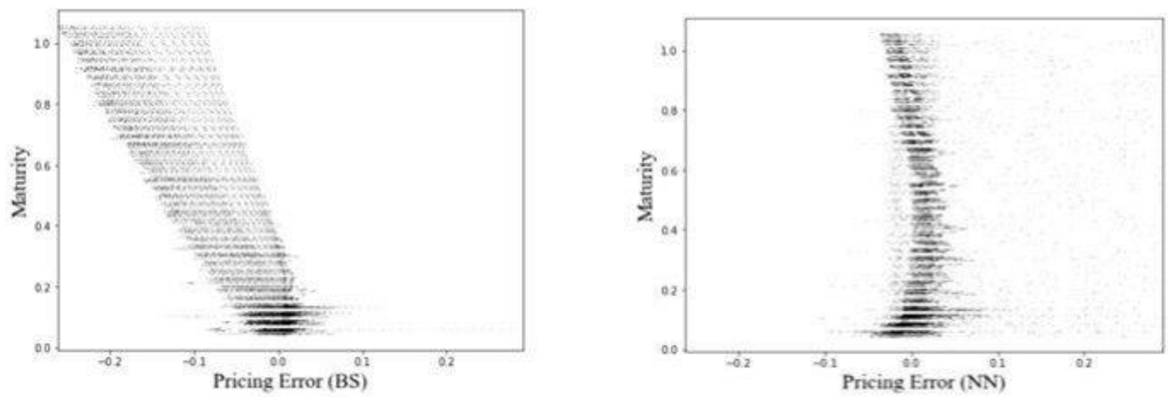
Long Maturity = T-t> $\frac{1}{12}$

Short Maturity = T-t< $\frac{1}{12}$

Medium Maturity = $\frac{1}{12}$ ≤T-t≤ $\frac{1}{12}$

For Long Maturity options, the ANN model outperforms the BS model as the dispersion of the BS of the left side of the mean is bigger in magnitude. In the Short Maturity case, the mispricing of the two models is similar.

Figure 5. Maturity vs. Pricing error



Part II Delta Hedging Method

To replicate the option through a delta hedging, we follow the method used by Hutchinson et al. (1994). The main idea of the strategy is to set up a replicating portfolio Vt that offsets the risk of an option position:

$$Vt = Vt(S) + Vt(B) + Vt(C)$$

where $Vt(S)$ is the value of the underlying asset position, $Vt(B)$ is the value of a bond position used to finance the position in the underlying asset, and $Vt(C)$ is the value of the option position held in the portfolio at time t.

The composition of the portfolio at time t = 0 is assumed to be:

$$V_0(C) = -C_{BSM,0}$$

$$V_0(S) = S_0 \Delta_{NN,0}$$

$$V_0(B) = -\big(V_0(S) + V_0(C)\big)$$

$$\Delta_{NN,0} = \frac{\partial C_{NN,0}}{\partial S_0}$$

where $C_{NN,0}$ is the price of the call option predicted by the ANN from which the delta of the call option $\Delta_{NN,0}$ can be computed.

The strategy consists of shorting one call option, longing for the underlying asset for $\Delta_{NN,0}$ number of shares at price S_0, and shorting for a bond to finance the rest of the long position in the underlying asset that is not financed with the sale of the call option.

The initial value of the replicating portfolio is 0 since the long position is financed entirely with riskless borrowing and the sale of the call option.

$$V_0 = V_0(S) + V_0(C) + V_0(B) = 0$$

Between the initialization of V at time t = 0 and the expiration at T, at all-time T-t, the spot and bond positions are rebalanced daily so that:

$$Vt(S) = St\Delta_{NN,t} \; where \; \Delta_{NN,t} = \frac{\partial C_{NN,t}}{\partial St}$$

$$Vt(B) =\leq e^{rT} V_{t-\tau}(B) - St\left(\Delta_{NN,t} - \Delta_{NN,t-\tau}\right) \; where \; \tau \; is \; defined \; to \; be \; 1 \; day \; in \; this \; paper$$

To calculate the delta of the neural network for the call option, after training using the ANN, we could derive the optimal weights on the input features and hence in principle are able to derive the partial derivative of the ANN predicted price with respect to the features. Using the Jacobian matrix, assuming that y is the output vector of f, the Jacobian matrix of f contains the partial derivatives of each element of the output y with respect to each element of the input x.

In this paper, more meaningful performance measures are also introduced. We adopt the same performance measures to analyse the hedging strategy performance.

1) The tracking error is defined by the value of the replicating portfolio at expiration date: $V_T = V_T(S) + V_T(B) + V_T(C)$
2) The present value of the expected absolute tracking error: $\varepsilon = e^{-rT} E[|V(T)|]$
3) The prediction error: $\eta = e^{-rT\sqrt{E^2[V(T)] + Var[V(T)]}}$

ε contains information regarding the accuracy of the option pricing formula, and for η, the information in the expected tracking error is combined with the information in the variance of the tracking error.

The team then bootstraps to resample observations from the dataset, options with the same options ID are grouped and the new resampled prices are construct in time order as the next step path. Through the extraction of built-in keras's gradient, the V(T) is further calculated to observe the V(T) in the long run, and it would be comparable to the V(T) of Black-Scholes.

Part II Delta Hedge Results

The hedging performance obtained from the bootstrapping consists of a comparison between the delta-hedge analysis for the BS model and the ANN model. This comparison is developed on the test set considering only the options contracts that have over 10 days of observations in the test set. Based on the error comparison shown in Table 3, the ANN model and the BS formula have similar performance.

Table 3. Hedging Performance

	num_of_options	epsilon_BS	nu_BS	epsilon_ANN	nu_ANN	tracking_error_ANN	tracking_error_BS
0	1878	1.578401	1.570971	1.648419	1.623731	2.817918	2.714089

A two-tailed paired t-test is performed to test for the null hypothesis H_0: two independent samples have identical expected. Based on the finds from the statistical test shown below, H_0 is not rejected which would mean the tracking error from both are not statistically significant different. Hence, similar hedging performance is obtained.

```
---------------------- All samples ---------------------------
Tabulate Table:
Sample                              t-statistic    p-value    Observations    % NN < BS
------------------------------    -------------  ---------  --------------  -----------
ITM (moneyness > 1.05)                  1.59967    0.10985             870         5.86
OTM (moneyness < 0.95)                  0.35006    0.72633             881        10.56
ATM (0.95 <= moneyness <= 1.05)         0.30712    0.759               127         8.66
All                                     1.38461    0.16626            1878         8.25
```

Conclusion

Using a simple neural network to train 2 input features can provide an option price like that of Black-Scholes theoretical option price, without implying any statistical distribution for any pseudo-stochastic parameters.

The performance metrics show that the tracking error is very similar to the Black-Scholes tracking error, i.e. the use of ANN option pricing for delta-hedging results are very similar to Black-Scholes tracking error

Final Thoughts

The team used bootstrapping, sampling underlying price from historical price points from the dataset, however, could have explore to simulate geometric Brownian motion paths instead, to enforce independent price paths, instead of conditional dependent price point.

The team's approach of extracting the partial derivative of ANN Option Price w.r.t to Underlying price was to use the TensorFlow's keras library method to extract gradient/coefficient from a Jacobian matrix, as the gradient value was passed into the rows and columns of the dataset.

A different approach could also be taken, where the weights of the 2 inputs could be used, as it would also mean the gradient/coefficient, and then apply directly to new simulated geometric Brownian price paths to calculate V(T)

The team could have explored a different asset class, as VIX option prices are usually priced with the GARCH model in the industry, instead of Black-Scholes

References

P. C. Andreou, C. Charalambous, and S. H. Martzoukos. Generalized parameter functions for option pricing. Journal of Banking & Finance, 34(3):633–646, 2010.

C. Boek, P. Lajbcygier, M. Palaniswami, and A. Flitman. A hybrid neural network approach to the pricing of options. In Proceedings of ICNN'95–International Conference on Neural Networks, volume 2, pages 813–817. IEEE, 1995.

R. Garcia and R. Genc¸ay. Option pricing with neural networks and a homogeneity hint. In Decision Technologies for Computational Finance, pages 195–205, 1998.

R. Garcia and R. Genc¸ay. Pricing and hedging derivative securities with neural networks and a homogeneity hint. Journal of Econometrics, 94(1-2):93–115, 2000.

E. Ghysels, V. Patilea, ´ E. Renault, and O. Torr`es. Nonparametric methods and option pricing. In D. Hand and S. Jacka, editors, Statistics in Finance, chapter 13, pages 261–282. John Wiley & Sons, 1998.

R. Gençay and R. Gibson. Model risk for European-style stock index options. IEEE Transactions on Neural Networks, 18(1):193–202, 2007.

J. M. Hutchinson, A. W. Lo, and T. Poggio. A nonparametric approach to pricing and hedging derivative securities via learning networks. The Journal of Finance, 49(3):851–889, 1994.

S. Liu, C. W. Oosterlee, and S. M. Bohte. Pricing options and computing implied volatilities using neural networks. Risks, 7(1):1–22, 2019b.

D. Ormoneit. A regularization approach to continuous learning with an application to financial derivatives pricing. Neural Networks, 12(10):1405–1412, 1999.

J. Ruf and W. Wang. Neural networks for option pricing and hedging: a literature review. SSRN 3486363, 2020.

J. Yao, Y. Li, and C. L. Tan. Option price forecasting using neural networks. Omega, 28(4):455–466, 2000.