Hedging Derivative Securities based on the Neural Network Coefficient Model

P. C. Ko[1], P. C. Lin[2], W. L. Chien[1], Y. S. Cheng[2]
[1]Department of Information Management
[2]Institute of Finance and Information
[1,2]National Kaohsiung University of Applied Sciences

Abstract

Investment in options has attracted much interest among investors for both speculative and hedging reasons in financial markets at present. Applying neural networks to forecast volatility in option pricing has increased in popularity in recent years since many studies have indicated that the conventional option pricing models are not sufficiently accurate. This article proposes a neural network coefficient (NNC) model to re-price option values to improve on the tracking error in the measurement of hedging capability. The NNC model uses the variables introduced by the Black-Scholes (BS) Model and applies the linear regression (LR) model to price option values. It is worth noting that each corresponding weight coefficient in LR is constructed by a complete neural network rather than by a scalar value. By capturing the nonlinear behaviors of option pricing, our proposed NNC model has lower tracking error and better hedging capability than the BS model. Besides, the experimental data are obtained from the Taiwan Stock Index Commodity (TIMEX) index options instead of artificially simulated data in order to avoid departing from reality.

1. Introduction

Option investment has become very popular among investors in financial markets for speculative and hedging purposes in recent years. In particular, derivatives can effectively reduce risk by enabling investors to fix a price for a future transaction now. The Black-Scholes (BS) [14] model constitutes the earliest option pricing methodology and was introduced in 1973. Although numerous pricing models have been studied, the BS model still exhibits systematic, significant and persistent bias [13]. The call option (C) and put option (P) values of European-style options are illustrated in Equations (1)-(4), where the related notations are listed in Table 1.

$$C = S \cdot \Phi(d_1) - K \cdot e^{-r_f T} \cdot \Phi(d_2) \tag{1}$$

$$P = K \cdot e^{-r_f T} \cdot \Phi(-d_2) - S \cdot \Phi(-d_1) \tag{2}$$

$$d_1 = \frac{\ln\frac{S}{K} + (r_f + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \tag{3}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{4}$$

Table 1. The Notations Used in the BS Model

Symbol	Description
S:	The current market price of the stock.
K:	The strike price of the option.
r_f:	The risk-free interest rate.
T:	The expiration date.
σ:	The volatility of the stock price.
Φ(·):	The cumulative distribution function for the standard normal distribution.

However, the hedging capability of option pricing in the BS model is generally not good enough because the real data in financial markets cannot fully conform to the assumptions introduced by the BS model.

The neural network-based regression technique is a nonparametric pricing model which has the distinct advantage of not relying on specific assumptions. Many studies apply neural networks to option pricing. Hutchinson, et al. adopt a nonparametric neural network method to estimate the pricing formula of derivative assets [4]. They use neural networks in the nonlinear regressions of some input variables on the observed market prices. Henrik uses multilayer perceptions to find a call option pricing formula and to map from the inputs to the bid and the ask prices of the options instead of assuming the mid-point [5]. Hamid and Iqbal use a back-propagation network to forecast the volatility of S&P 500 index futures prices. They compare forecasting volatility from neural networks with implied volatility using the Barone-Adesi and Whaley American futures options pricing models [7]. Yao, et al. use backpropagation neural networks to forecast the option prices of Nikkei 225 index futures and they outperform the BS model [10]. Ormoneit introduces the iterative extended Kalman filter as a neural network learning rule to effectively eliminate no-arbitrage pricing restrictions when pricing German stock index options [12].

The Neural network model also effectively improves the forecasting accuracy of option hedging and produces better hedging parameters.

The Bayesian regulation generates significantly smaller pricing and delta hedging errors than the baseline neural networks and the BS model [8]. Carverhill, et al. examine the best way to set up and train a neural network for option hedging [11]. Morelli, et al. [9] apply multi-layer perceptrons and radial basis functions to European and American options to evaluate the Greek letters for hedging strategy. The results show that neural networks are able to precisely predict the values of the options and the Greek letters. Gencay and Qi apply a Bayesian regularization to mitigate overfitting and improve generalization for hedging derivative securities with S&P 500 index call options.

Using the nonlinear optimization characteristics of traditional neural networks, the objective of this paper is to propose a neural network coefficient (NNC) model to re-price the call option value. This NNC model considers the same variables as those introduced by the BS model, and simply combines these variables linearly by applying the linear regression (LR) model. Each corresponding weighted coefficient in LR is determined by a complete neural network rather than just a scalar value. The results of our experiment show that our proposed model effectively improves the delta-hedging error and enhances hedging capability when compared with the BS model. In addition, all of the experimental data are from Taiwan Stock Index Commodity (TIMEX) index options instead of artificial simulated data in order to avoid departing from reality.

2. Neural Network Coefficient Model

A conventional M-Layer neural network is shown in Figure 1. Let $\mathbf{p}=[p_1\ p_2\ \dots\ p_R]^T$ denote the input signals. $\mathbf{w^m}$ represents the weight matrix for the m-th layer shown in Equation (5), where $w^m_{j,i}$ refers to the synaptic weight connecting the m-th layer of neuron j to the *(m-1)*-th layer of neuron i, and S^m refers to the number of neurons in layer m. $f^m(\cdot)$, b^m_j and $\mathbf{a^m}$ are the used activation function, the bias applied to neuron j and the output signals in layer m, respectively. Let $\mathbf{p}=\mathbf{a}^0$. $\mathbf{a^m}$ should be shown in Equation (6).

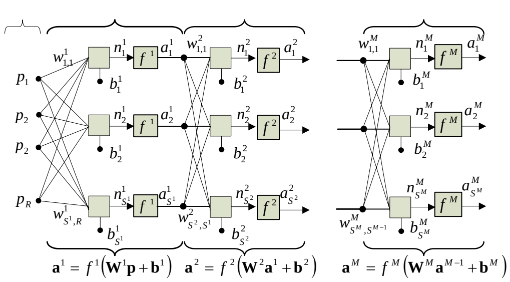

Figure 1. M-Layer Neural Network

$$\mathbf{w^m}=\begin{bmatrix} w^m_{1,1} & w^m_{1,2} & \cdots & w^m_{1,S^{m-1}} \\ w^m_{2,1} & w^m_{2,2} & \cdots & w^m_{2,S^{m-1}} \\ \vdots & \vdots & \vdots & \vdots \\ w^m_{S^m,1} & w^m_{S^m,2} & \cdots & w^m_{S^m,S^{m-1}} \end{bmatrix} \quad (5)$$

$$\mathbf{a}^{m+1} = f^{m+1}\left(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}\right) \quad \text{for } m = 0,1,2,\ldots,M\text{-}1 \quad (6)$$

The proposed NNC model is shown in Figure 2. This model is based on the linear regression method, and means that $C=\beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_Nx_N$, where C denotes the predicted value of the call option and $x_1, x_2, \ldots, x_N$ are the considered variables. It is worth noting that the weight coefficients $\beta_0,\beta_1,\ldots,\beta_N$ are composed of distinct neural networks. Assuming that the corresponding neural network for β_i is an M^i-layer network with input set $\{x_{i,1}, x_{i,2}, \ldots, x_{i,m_i}\}$, the total error energy (ξ) is estimated from the tracking error introduced by Hutchinson, et al. in 1994 [4].

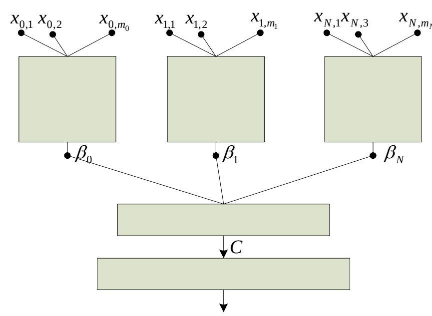

Figure 2 Neural Network Coefficient Model

Tracking Error

The use of the tracking error as the performance measure is based on the no-arbitrage phenomenon. Let $V(t)=V_S(t)+V_B(t)+V_C(t)$ be the value of the portfolio using our model, where $V_S(t)$, $V_B(t)$ and $V_C(t)$ are the values of stocks, bonds and held call options at time t. The expected value of such a hedged option portfolio at the expiration date should be exactly zero. Let α_{t_i} denote the composition of the portfolio at time t_i shown as follows:

$$\alpha_{t_i}=\begin{bmatrix} V_S(t_i) \\ V_C(t_i) \\ V_B(t_i) \end{bmatrix}=\begin{bmatrix} S(t_i)\Delta(t_i) \\ -C_M(t_i) \\ e^{r_f(t_i-t_{i-1})}V_B(t_{i-1})-S(t_i)(\Delta(t_i)-\Delta(t_{i-1})) \end{bmatrix} \quad (7)$$

where $\Delta(t_i)=\dfrac{\partial C_m(t_i)}{\partial S}$. $C_m(t_i)$ is defined as the predicted call option value based on Model m, where $m=BS$ or NNC. The initial case α_{t_0} is shown in Equation (7).

$$\alpha_{t_0} = \begin{bmatrix} V_S(t_0) \\ V_C(t_0) \\ V_B(t_0) \end{bmatrix} = \begin{bmatrix} S(t_0)\Delta(t_0) \\ -C_M(t_0) \\ -(V_S(t_0)+V_C(t_0)) \end{bmatrix} \quad (8)$$

The total error energy ξ_m is defined as follows:

$$\xi_m = e^{-r_f T} E(|V(T)|) \quad (9)$$

Let $\xi_m(q)$ refer to the instantaneous error energy at iteration q by using model m. Used in a manner similar to the Least-Mean-Square (LMS) algorithm, the back-propagation algorithm applies a correction $\Delta w_{j,i}^{n,m}$ to the synaptic weight $w_{j,i}^{n,m}$ denoted by $w_{j,i}^{m}$ in β_n, n=0,1,2,…,N.

$$\frac{\partial \xi_{NNC}(q)}{\partial w_{j,i}^{n,m}} = \frac{\partial \xi_{NNC}(q)}{\partial V(q)} \cdot \frac{\partial V(q)}{\partial C(q)} \cdot \frac{\partial C(q)}{\partial \beta_n(q)} \cdot \frac{\partial \beta_n(q)}{\partial w_{j,i}^{n,m}(q)} \quad (10)$$

where

$$\frac{\partial \xi_{NNC}(q)}{\partial V(q)} = \begin{cases} e^{-r_f T} & if\ V(q) > 0 \\ -e^{-r_f T} & if\ V(q) \le 0 \end{cases} \quad (11)$$

$$\frac{\partial V(q)}{\partial C(q)} = \frac{\partial V_S(q)}{\partial C(q)} + \frac{\partial V_B(q)}{\partial C(q)} + \frac{\partial V_C(q)}{\partial C(q)} = e^{r_f(t_i - t_{i-1})} \frac{\partial V_B(q)}{\partial C(q)} + S(t_i) \frac{\partial \Delta(t_{i-1})}{\partial C(q)} \quad (12)$$

$$\frac{\partial C(q)}{\partial \beta_n(q)} = \begin{cases} 1 & if\ n = 0 \\ x_n & if\ n = 1,2,...,N \end{cases} \quad (13)$$

The adjustment to $w_{j,i}^{n,m}$ is similar to the traditional back-propagation algorithm, because each β_n is defined by a complete neural network. So,

$$\frac{\partial \beta_n(q)}{\partial w_{j,i}^{n,m}} = \begin{cases} \varphi'(\cdot) a_j(q) & if\ m = M^n \\ \sum_k w_{k,j}^{n,m} \varphi'(\cdot) a_j(q) & if\ m \ne M^n \end{cases} \quad (14)$$

Finally, $\Delta w_{j,i}^{n,m}$ is illustrated in Equation (15).

$$\Delta w_{j,i}^{n,m} = -\eta \cdot \frac{\partial \xi_{NNC}(q)}{\partial w_{j,i}^{n,m}} \quad (15)$$

3. Results of Experiment

To avoid losing reality, our experimental data are obtained from Taiwan Stock Index Commodity (TIMEX) index options instead of using artificial simulated data. These data are European-style options. It means that they are exercised only on the option expiration date, and the payoff is determined by the Taiwan Stock Index on the option maturity date. There are 841 samples chosen from the TIMEX during 2004, which are further divided into training and testing sets with corresponding ratios of 80% and 20%.

The simulation programs are written in Borland C++ Builder 6.0 run on a Microsoft Windows XP professional platform. The considered variables set (x_1, x_2, x_3, x_4, x_5) used in our NNC model is (S, K, r_f, T, σ), which is introduced by the BS model illustrated in Table 1. The corresponding input set for β_i is shown in Table 2.

Table 2. The Corresponding Input Set for β_i

β_i	Input Set
β_0	(K, T)
β_1	(K, T, S, V)
β_2	(K, T, S)
β_3	(K, T+1, T+2, T+3)
β_4	(K, T+1, T+2, T+3)
β_5	(K, T, $\sigma_{t_{i-1}}, \sigma_{t_{i-2}}, \sigma_{t_{i-3}}$)

The configuration of each neural network is presented in Table 3 of our NNC model.

Table 3. The Configuration of Each Neural Network

Parameter	Value
No. of Hidden Layers	2
No. of Neurons in each Hidden Layer	6
Activation Function (Hidden Layer)	Sigmoid
Activation Function (Output Layer)	Linear
Learning Rate	0.01
No. of Epochs	1000

The derived absolute delta-hedging errors of the NNC and BS models are shown in Table 4 and Table 5, respectively. The total error energies are ξ_{NNC}=200.56 and ξ_{BS}==714. These results show that our proposed NNC model performs better than the BS model.

Table 4. Delta Hedge Error – NNC model (Year: 2004)

Strike price	Mar.	Apr.	May	Jun.	Sep.
5300	NA	NA	NA	200.5	124.4
5500	NA	NA	NA	289.2	188.3
5700	NA	NA	NA	198.5	167.6
5900	NA	NA	NA	295.6	127.1
6100	NA	NA	NA	187.7	170.9
6200	NA	288.3	NA	NA	NA
6300	NA	264.6	0.0	125.0	160.1
6400	NA	404.5	211.0	NA	NA
6500	NA	NA	219.7	61.4	211.8
6600	NA	347.5	161.1	NA	NA
6700	NA	128.4	173.3	251.9	219.4
6800	NA	127.6	271.7	NA	NA
6900	NA	133.8	248.4	206.5	188.7
7000	NA	231.2	243.2	NA	NA
7100	225.0	263.3	194.7	200.4	168.9
7200	NA	127.5	253.3	NA	NA
7300	367.0	211.2	243.9	251.4	NA

Table 5. Delta Hedge Error - BS Model (Year 2004)

Strike price	Mar.	Apr.	May	Jun.	Sep.
5300	NA	NA	NA	91.1	34.2
5500	NA	NA	NA	21.1	13.6
5700	NA	NA	NA	66.4	151.4
5900	NA	NA	NA	145.3	195.1
6100	NA	NA	NA	74.5	44.9

6200	NA	39.1	NA	NA	NA
6300	NA	73.7	0.0	51.2	150.4
6400	NA	62.4	218.9	NA	NA
6500	NA	NA	415.5	159.9	333.6
6600	NA	1912.6	2305.9	NA	NA
6700	NA	2727.1	1830.4	2017.4	1890.3
6800	NA	2813.4	974.5	NA	NA
6900	NA	1458.7	1382.0	1463.8	1648.3
7000	NA	206.0	513.6	NA	NA
7100	28.5	101.7	356.3	260.8	262.9
7200	NA	54.3	139.2	NA	NA
7300	0.7	127.2	126.9	176.1	NA

4. Conclusions

Generally speaking, financial data forecasting is always difficult because it is greatly influenced by economic, international and political events. The option pricing model tends to be more popular with financial institutions for both speculative and hedging purposes. The Black-Scholes model is a well-known option pricing model based on certain specific assumptions. However, past studies have shown that the Black-Scholes model cannot effectively capture the nonlinear behavior of option prices if real options data is applied. In this article, we therefore introduce a neural network coefficient (NNC) model to estimate option pricing to improve the tracking of error performance used to measure hedging capability. This model uses the variables included in the BS model and combines them with linear regression to evaluate the option value. To capture the nonlinear behavior of option prices with a high degree of accuracy, each coefficient in the linear regression model is produced by a neural network. The empirical results show that the derived average absolute delta-hedging errors of the NNC and BS models are ξ_{NNC}=200.56 and ξ_{BS}==714, respectively. It is obvious that our proposed NNC model performs better hedging capability than the BS model.

References

[1] N. Burgess and A-P. N. Refenes, "Modelling non-linear moving average processes using neural networks with error feedback: An application to implied volatility forecasting," Signal Processing, 74:89-99, 1999.

[2] A-P. N. Refenes and W. T. Holt, "Forecasting volatility with neural regression: A Contribution to model adequacy," IEEE Transactions on Neural Networks, 12:850-864, 2001.

[3] P. Lajbcygier, "Improving option pricing with the product constrained hybrid neural network," IEEE Transactions on Neural Networks, 15:465-476, 2004.

[4] J. M. Hutchinson, A. W. Lo and T. Poggio, "A nonparametric approach to pricing and hedging derivative securities via learning networks," Journal of Finance, 49:851-889, 1994.

[5] H. Amilon, "A neural network versus Black-Scholes: A comparison of pricing and hedging performances," Journal of forecasting, 22:317-335, 2003.

[6] J. Bennell and C. Sutcliffe, "Black-Scholes versus neural networks in pricing FTSE 100 options," Working Paper 00-156, University of Southampton - Department of Accounting and Management Science, 2003.

[7] S. A. Hamid and Z. Iqbal, "Using neural networks for forecasting volatility of S&P 500 index futures prices," Journal of Business Research, 57:1116-1125, 2004.

[8] R. Gencay and M. Qi, "Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging," IEEE Transactions on Neural Networks, 12:726-734, 2001.

[9] M. J. Morelli, G. Montagna, O. Nicrosini, M. Treccani, M. Farina and P. Amota, "Pricing financial derivatives with neural networks," 338:160-165, 2004.

[10] J. Yao, Y. Li and C. L. Tan, "Option price forecasting using neural networks," International Journal of Management Science, 28:455-466, 2000.

[11] A. Carverhill and T. H. F. Cheuk, "Alternative neural network approaches for option pricing and hedging," Working paper, Dec. 2003.

[12] D. Ormoneit, "A regularization approach to continuous learning with an application to financial derivatives pricing," Neural Networks, 12:1405-1412, 1999.

[13] G. Bakshi, C. Cao, and Z. Chen, "Empirical performance of alternative option pricing models," Journal of Finance, LII:2003-2049, 1997.

[14] F. Black and M. Scholes, "The pricing of options and corporate liabilities," Journal of Political Economy, 81:637-659, 1973.