PERGAMON

Omega 28 (2000) 455–466

omega
The International Journal of Management Science

www.elsevier.com/locate/dsw

Option price forecasting using neural networks

Jingtao Yao, Yili Li, Chew Lim Tan*

School of Computing, National University of Singapore, Singapore 119260, Singapore

Received 1 August 1998; accepted 1 September 1999

Abstract

In this research, forecasting of the option prices of Nikkei 225 index futures is carried out using backpropagation neural networks. Different results in terms of accuracy are achieved by grouping the data differently. The results suggest that for volatile markets a neural network option pricing model outperforms the traditional Black–Scholes model. However, the Black–Scholes model is still good for pricing at-the-money options. In using the neural network model, data partition according to moneyness should be applied. Those who prefer less risk and less returns may use the traditional Black–Scholes model results while those who prefer high risk and high return may choose to use the neural network model results. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Neural networks; Forecasting; Option pricing; Black–Scholes model

1. Introduction

There has been a tremendous increase in the awareness and activities of derivative securities in recent years. Option markets are among the top most popular shares of financial institutions. To explore the markets well and improve investment yields, pricing of options has attracted researchers' attention for years. Modeling or predicting option prices is very important for practitioners. Numerous pricing models have been created and studied. The Black–Scholes model [2,16] is the most widely used model which is based on certain assumptions such as geometric Brownian motion of stock price movements, that the option is exercised at the time of maturity (i.e. European option), constant interest rate, continuous trading without dividends and tax applied to the stocks, and that the market is frictionless. Although research results show that the Black–Scholes model does outperform other option pricing models except in the case of deep-in-the-money and deep-out-of-the-money, the fact that real data often violate most of the model's assumptions brings it under suspicion. Violations may be found in the following situations: (1) instead of random walk description, fractal may be a better hypothesis for the market movement [20]. Akgiray [1] has also rejected the assumption of constant variance of stocks. (2) In fact, American option dominates the real markets. People want more choices when the market changes, such as when it is near the time to pay dividends. (3) Dividends are common practice especially for stocks.

There are many attempts to modify the Black–Scholes model with the view to avoiding the above violations. The following are some examples of such attempts: pure jump model [5] and mixed diffusion jump model [17] based on continuous constraints; square root constant elasticity of variance diffusion model [5], displaced diffusion model [25] and compound option diffusion model [8] based on the con-

* Corresponding author. Tel.: +65-874-2900; fax: +65-779-4580.

E-mail address: tancl@comp.nus.edu.sg (C.L. Tan).

PII: S0305-0483(99)00066-3

stant volatility assumption; Merton's stochastic interest rate extensions [16]; stochastic volatility models [11]. Most recent attempts include regime switching model [18] and implied binomial tree [6,26].

All these attempts call for a more accurate and realistic option pricing model with less or even no assumptions. Neural networks are an emerging and challenging computational technology and they offer a new avenue to explore the dynamics of a variety of financial applications. Neural networks are simulated networks with interconnected simple processing neurons which aim to mimic the function of the brain's central nervous system. Neural networks are good at inputs and outputs relationship modeling even for noisy data. Many studies on the application of neural networks and related techniques for financial modeling are documented [7,10,29,31] on topics such as economic forecasting [9], stocks [22], foreign exchange rate forecasting [32]. Although neural network forecasting models are not always superior to traditional models [3,14], studies on option pricing using neural networks have attracted researchers and practitioners.

Moreover, the Black–Scholes model and its derivatives are based on their ability to capture the dynamic of stock prices, while the neural network model with little assumption is a data driven approach. It uncovers the relationship of option prices and its related information such as stock price, strike price, etc. from historical data.

Hutchinson et al. [12] apply a neural network pricing approach to the pricing and delta-hedging of S&P 500 futures options from 1987 to 1991. Their results show that learning networks can recover the Black–Scholes formula from a 2-year training set of daily option prices, and that the resulting network formulas can be used successfully to both price and delta-hedge options out-of-sample. They also compare the neural network model with four popular methods: ordinary least squares, radial basis function networks, multilayer perceptron networks and projection pursuit. They suggest that although not a substitute for the more traditional arbitrage-based pricing formulas, network-pricing formulas may be more accurate and computationally more efficient when the underlying asset's price dynamics are unknown.

Malliaris et al. [15] test on the S&P 100 for 1992 calendar year. A neural network has been developed by them to forecast future volatility, using past volatilities and other options market factors. Up to 13 inputs are explored in their research. The accurate forecasts of future volatility allow a trader to establish the proper strategic position in anticipation of changes in market trends. The outputs of their neural network model are compared with the implied volatility calculated based on the Black–Scholes model. They claim that the neural network volatility approach is superior

Table 1
A summary of previous neural network models on option pricing

Model	Hutchinson et al. [12]	Lajbcygier et al. [13]	Malliaris et al. [15]	Qi et al. [21]	Present Work
Input	S/X, $T-t$	S/X, $T-t$, r, σ	13 variables[a]	S, r, T, X, V	S, X, T
Hidden	4	4, 10, 20	5	5	2, 3
Output	C/X	C/X	σ	C	C
Activation	sigmoid	logistic	?	sigmoid	hyperbolic tangent
Measure	R^2	R^2, NRMSE, MAPE	MAD, MSE	ASE, MAE, R^2	NMSE
Market	SP500	SPI	SP100	SP500	Nikkei 225
Time	87–91	January 92–December 94	January 92–December 92	December 94–January 95	January 95–December 95
No. of Data	6000+	3308	253(?)	1107	17790

[a] Variable including (history, middle, distant) volatilities, (put, call, market) price and change, T and sum of some of them.

to the historical and implied volatility approach. According to their test results, the neural network forecasts are very accurate estimates of volatility preferred by traders.

Qi et al. [21] also apply a multilayer feedforward neural network to price S&P 500 index calls spanning December 1994 through January 1995. An extra open interest rate, V, which is found to be an important factor in pricing options is fed to the neural network to predict option prices. They claim that both the in- and out-of-sample accuracies are far better for the artificial neural network than for the Black–Scholes formula, and thus conclude that the neural network is a good alternative to the traditional Black–Scholes formula when its underlying assumptions are violated. After analyzing the weights of their neural network, they also confirmed that the lower the X (strike price), the higher the S (stock price) or the longer the T (time to maturity), the higher the C (call price).

Lajbcygier et al. [13] work on options of Australian All Ordinaries Share Price Index on futures. They use a two input model (S/X, T) and a four input model (S/X, T, r, σ) (r is interest rate, σ volatility) to compare with the two input model used by Hutchinson et al. [12]. The input σ, the volatility, is estimated by the historical approach in which 60 daily data are used. They claim that the four input model outperforms the two input model and the Black–Scholes model. Although the Black–Scholes model fits very well for Australian data in term of high coefficient of determination R^2 (0.9815), their neural network model works well for reduced data region (i.e. near the money and maturity).

Previous work on neural network applications on option pricing as discussed above is summarized in Table 1 together with the summary for our present work to be discussed later.

Neural network models are context sensitive. However, Callen et al. [3] claim that neural network models are not necessarily superior to linear time series models even when the data are financial, seasonal and nonlinear. Thus, a comprehensive study needs to be conducted for different markets and for different neural network models before claiming that the neural network model is superior to the Black–Scholes model.

We will present a study on neural network forecast for options on Nikkei 225 index futures in 1995. The present work is also an attempt to emulate the Black–Scholes formula using neural networks but with a different approach in several aspects. For instance, we note that Malliaris et al. [15] use a neural network to forecast the volatility σ while Lajbcygier et al. [13] estimate σ to serve as an input to their neural network. In the present work, we will incorporate σ into the neural network. It is as if that neural network model not only emulates the Black–Scholes formula but also provides σ in the process. Furthermore, we will work on an Asian option market to examine the applicability of our model. Other differences in our approach will be discussed in Section 4.

The organization of this paper is as follows, Section 2 gives a general overview of the Black–Scholes option pricing model. Section 3 describes basic concepts of neural networks. Our pricing model and experiment are presented in Section 4. Then a section on the results and discussion will follow. Finally we conclude this paper (Section 6) with suggestion for further research.

2. Option pricing and Black–Scholes model

This section will briefly introduce the Black–Scholes model to help readers to understand the concept of option pricing. A future is the obligation to transact an asset at a specified time and price in the future. An option is a contract that gives a party the right (but not the obligation) to buy or sell an asset for a specified time at a specified price. Options allow people to bet on the future events and to reduce the financial risk. There are two kinds of options: American option and European option. The former may be exercised any time before its expiration date while the later can only be exercised on its expiration date. The formulas derived by Fischer Black and Myron Scholes [2] in the early 70s are the most important model for pricing options. The model is based on some assumptions such as geometric Brownian motion of stock prices movement, continuous trading with no dividends and taxes applied to the stocks, and that the market is frictionless. Based on *random walk hypotheses* [19] of stock movement, the volatility of stocks is estimated based on historical data. The pair of formulas for call and put, respectively, in Eq. (1), constitute the Black–Scholes model.

$$
\begin{aligned}
c &= SN(d_1) - Xe^{-rT}N(d2) \\
p &= Xe^{-rT}N(d1) - SN(d_2)
\end{aligned}
\tag{1}
$$

where $d_1 = (\ln(S/X) + (r + (\sigma^2/2)T)/\sigma\sqrt{T}$; $d_2 = (\ln(S/X) + (r - (\sigma^2/2)T))/\sigma\sqrt{T} = d_1 - \sigma\sqrt{T}$; c the call price; p the put price; S the current underlying asset price; X the exercise price; T the time to maturity (in years); σ the volatility of the underlying asset; r the short-term risk free interest rate and $N(\,)$ the cumulative probability function.

The key, and unknown, parameter of this model is σ. Basically the Black–Scholes model says that the option price, no matter it is a call or put, is a function of asset price, exercise price, time-to-maturity, volatility of asset price and risk free interest rate. Eq. (2) is a simple form of the Black–Scholes function (BS).

$$\text{Option price} = BS(S, X, T, \sigma, r). \tag{2}$$

All those variables except for the volatility are easily obtainable from the market. σ is the only unknown factor in the formulas. σ is often assumed unchanged when forecasting option prices. Research has been conducted for the volatility calculation that leads to the derivatives of the Black–Scholes model. The forecasting accuracy is based on the σ computation. Basically, computing or estimating σ falls into two approaches: historical and implied volatility approaches. The historical approach is much simpler than the other one. Volatility for tomorrow or the next time period is estimated by the historical volatility. The annualized standard deviation of historical daily returns is defined as the historical volatility. Considering there are 253 trading days per year,

$$\sigma = s\sqrt{253}, \tag{3}$$

where s is the standard deviation of the daily returns $\log(S_t/S_{t-1})$. 30, 45 and 60 days are used as the historical data to represent nearby, middle and distance volatility [15]. The unreality of this approach is that it assumes that the market will not change. This assumption only holds if the volatility is stable. The implied volatility is to use the Black–Scholes formula in reverse. A call price can be considered as the true market price as it reflects the opinions of trading participants on future returns. The volatility which is implied thus can be computed out from the Black–Scholes formula. It is a more commonly used method to estimate volatility since it looks more on the future.

Is the Black–Scholes model the best for option pricing? Studies by investigators have led to the following general conclusions [23]:

1. The Black–Scholes model is extremely good for pricing at-the-money options, especially when time to expiration exceeds two months.
2. For deep in-the-money and out-of-the-money options, significant deviations between market prices and model prices can occur.
3. Short-term (less than one month) options are often mispriced.
4. Options on extremely low and extremely high volatility stocks are often mispriced.

Derivatives of the Black–Scholes model include those of Merton [16], Cox and Ross [5], Merton [17], Geske [8], Rubinstein [25], Hull and White [11], Naik [18], Derman and Kani [6] and Rubinstein [26].

3. Neural network as a forecasting tool

A neural network is a collection of interconnected simple processing elements. Every connection in a neural network has a weight attached to it. There are countless learning methods for neural networks. However, they can be classified into two groups, namely supervised and unsupervised method. Supervised learning requires historical data with examples of both dependent and independent variables to train the network. The known answers are worked as a teacher to correct the behavior of the training network. Hopfield, Boltzman, Adaline, Backpropagation, are some of the well known supervised learning methods. It is commonly used to build prediction, classification and time series models. Unsupervised learning method creates its own model to interpret the data without known answers. Adaptive resonance theory, Kohonen self-organizing map counterpropagation network are some of the popularly used unsupervised learning approaches. They are often used for clustering data. The backpropagation algorithm [27] has emerged as one of the most widely used learning procedures for multilayer networks. A typical backpropagation neural network usually has an input layer, some hidden layers and an output layer. The units in the network are connected in a feedforward manner, from the input layer to the output layer. The weights of connections have been given initial values. The error between the predicted output value and the actual value is backpropagated through the network for the updating of the weights. This is a supervised learning procedure that attempts to minimize the error between the desired and the predicted outputs. The output value for a unit j is given by the following function:

$$O_j = G\left(\sum_{i=1}^{m} w_{ij} x_i - \theta_j\right), \tag{4}$$

where x_i is the output value of the ith unit in a preceding layer, w_{ij} is the weight on the connection from the ith unit, θ_j is the threshold, and m is the number of units in the preceding layer. The function $G(\,)$ is a sigmoid hyperbolic tangent function:

$$G(z) = \tanh(z) = \frac{1 - e^{-z}}{1 + e^{-z}} \tag{5}$$

$G(\,)$ is a commonly used activation function for business applications in backpropagation networks.

Fig. 1 shows a one-hidden-layer neural network which can be used as our option pricing model.

In theory, neural networks can simulate any kind of data pattern provided it is trained sufficiently [30]. When applying a neural network as a forecasting tool, we first need to train the network. In the training pro-

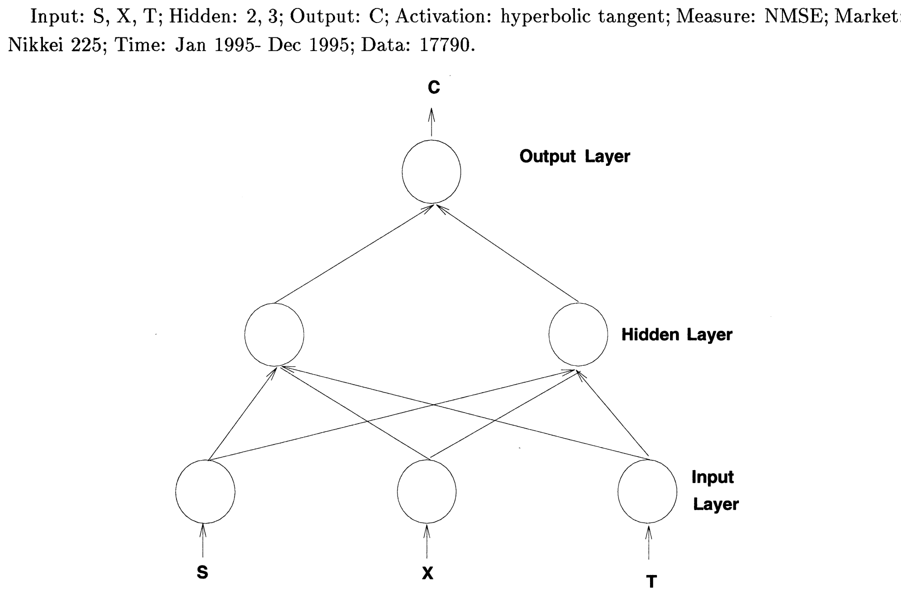

Fig. 1. A one-hidden-layer neural network used in this study.

cedure, the neural network will learn from experiences based on our hypotheses. For example, we have a huge set of data consisting of three columns, say, a, b and c. The neural network can be trained by training data patterns, i.e. (a, b, c). Here the inputs are a and b while the output is c. The hypothesis for this training is that we assume that c depends on a and b. With enough data patterns, a neural network model which represents the relationship between the inputs, a and b and the output, c, can be built. Thus this network can be used as a forecasting tool of c based on a and b. Similarly the neural network in Fig. 1 is to forecast C based on X, S and T.

4. Research methodology and aims

Neural networks have the ability to model nonlinear patterns and learn from the historical data. Can neural networks be used as a simulator of the Black–Scholes model or as an option price forecasting tool? One of the major issues of option pricing is to estimate the implied volatility using models like the Black–Scholes model, for instance. People forecast expected market returns based on the volatility of stocks. Research by Malliaris et al. [15] is mainly based on the forecasting of future market volatility. Can neural networks do the same or even better for option price forecasting comparing with the Black–Scholes model? Hutchinson et al. [12] state that neural networks could although it may not be a substitute for traditional pricing formulas. An aim of this research is to find out whether the neural network forecasting model can be used as an option price forecasting tool. Another aim is to find out whether the neural network model can overcome the drawbacks of the Black–Scholes model. As the BS function of Eq. (2) holds, we could use S, X, T, σ, r as the inputs and Option Price as the output to build up a neural network. Unfortunately, there is no such σ column in any data sets. In fact, estimating σ is a key issue in the Black–Scholes model. Hence, instead of trying to estimate σ as an input like what Lajbcygier et al. [13] did, we submit that σ can be built into a black box, namely the neural network, such that it carries σ into the modeling process of the Black–Scholes formula.

Thus instead of a backpropagation network with the pattern (S, X, T, σ, r, Option Price), the model building procedure becomes a mapping function that seeks for (S, X, T, r, Option Price) pattern. The first pattern is in the aim of modeling option prices based on S, X, T, σ and r. The only unknown parameter in this pattern is σ. If we assume that the volatility is kept unchanged for training, validation and forecasting time period (a constant volatility) or it is a function of other parameters (a dynamic volatility), e.g. our neural network, then σ can be omitted in the pattern. In other words, the neural network model is a model that by-passes the volatility estimation to forecast the real option prices. An argument for the above approach is that the volatility obtained from the neural network forecast bears no economic relation to the Black–Scholes formulas. The volatility here refers to our uncertainty about future asset price movements. More precisely, it is a measure of our uncertainty about the proportional asset price changes. So the standard deviation of the return is estimated as historical volatility. Although we cannot directly revert Eq. (1) so that σ is expressed as a function of S, X, r and c or p, there are some procedures used to calculate implied volatility in practice. If the Black–Scholes model is correct, then the historical volatility is the correct estimator for σ. There is no need to use neural network as a option pricing model as it is just a simulator of Black–Scholes model. However, if the Black–Scholes model is not correct, then a neural network forecast of volatility may not be the right quantity for option pricing purposes as both are based on wrong assumption. Our model uses the second pattern to model option prices. The volatility, σ is not directly reflected in the neural network model. Even though it may eventually be captured by the neural network. We still cannot explain it clearly due to the drawbacks of neural networks. A neural network is often regarded as a black box and it lacks insight. Due to its nonlinear activation functions, it is generally not possible to extract simple rules that describe how a network makes its prediction.

Moreover, as we know that r may remain relatively stable compared with the stock or option prices, it will confuse the network learning if we feed it to the neural network [7]. The neural network does not have such an ability to learn from relatively constant r. Indeed we have included r in our preliminary tests, but found the results to be rather unsatisfactory and in most cases we could not get a converged network. Our results showed that inclusion of r gave a higher error than when r was not present. The training errors of most of models with an additional input r were all above 0.9. For instance, the training error and testing error of a model with inclusion of r for sequenced whole data set were 0.913422 and 0.924432, respectively, as compared to 0.042494 and 0.102806 for a model without r as shown in row 1 of Table 4. Thus the riskless interest rate r is not used in our neural network model. This may differ from Qi's [21] result that the open interest is an important factor in pricing options. In his model, two interest rates were used and thus they did not confuse the neural networks. In the experiment of Hutchinson et al. [12] and Lajbcygier et al. [13] they used a S/X ratio because option price is homogeneous with degree one in S and X. This technique is used to reduce the number of inputs for a larger network [7]. It may contribute more information on the relation of (S, X) pair in one input. In our experiment, we use the quotients of stock prices to strike prices or moneyness [4] to partition the data. This approach will speed up neural networks training and allow us to build more stable models.

4.1. Data

The data used in this research are the transaction data of Nikkei 225 stock index options traded in Singapore International Monetary Exchange(SIMEX). We studied 17,790 call option price data points from 4 January 1995 to 29 December 1995. Only traded prices were used. Bid and ask prices were not included in this study. The columns of data are arranged as follows: trading day, date of expiration, strike price, option price. Another file which includes the stock index for the same period is also used. First we calculate the time-to-maturity according to the trading and expiration dates. This will be an important factor for option pricing as shown in Eq. (2). There can be dozens of prices appearing on a single day. To narrow the data range for a better training, moneyness is used to partition the data. The moneyness [4] is defined as the quotient of stock price and strike price. Options are referred to as in-the-money, at-the-money and out-of-the-money in the standard option pricing terminology. A call option is in-the-money when $S > X$, at-the-money when $S = X$ and out-of-the-money when $S < X$. Even in the absence of transaction costs, an at-the-money option is hard to find. So we name those options at-the-money when $S \approx X$. We use α to indicate the extent of $S \approx X$ to define the at-the-money data set. Table 2 shows the number of entries in each data set based on the choice of α. To balance the number of each set, $\alpha = 6\%$ is used in our experiment.

To further reduce the number of each set, the extreme moneyness is used. On every trading day, for the same maturation day one of the biggest quotients is chosen as out-of-the-money option while the smallest for in-the-money. The nearest to 1 is set as the at-the-money price except for cases when the difference between stock price and strike price exceeds 250 which is 1.5% of the average price. The total number of pat-

Table 2
Data partition according to moneyness(stock price/strike price), $\alpha = 6\%$ is adopted in this paper

α (%)	Out ($S/X \le 1-\alpha$)	At ($1-\alpha < S/X \le 1+\alpha$)	In ($S/X > 1+\alpha$)
2	11036	4762	1992
4	9899	4016	3875
5	9372	3667	4751
6	8772	3324	3875
10	6424	2288	9078

terns can be found in Table 3. These are the data sets we will explore for our neural network models. There are three groups of data sets. In the first group, op95c refers to the whole data we have, while op95c-s-in, op95c-s-at, and op95c-s-out belong to the second group. They are partitioned based on $\alpha = 6\%$. The last group, *in*, *at* and *out* data sets, is obtained by the extreme moneyness rule.

4.2. Neural network model

A backpropagation neural network [27], the most popular network in business applications [31], is used in this study. In order to receive a more even distribution over input space, a sigmoid hyperbolic tangent function is used as the activation function. We use one hidden layer for each neural network model. The neural network is built with three inputs in the input layer and one output in the output layer. The i–h–o stands for a neural network with i neurons in the input layer, h neurons in the hidden layer and o neurons in the output layer. Various architectures are experimented. The procedure we follow is the number of nodes in the hidden layer being experimented are in the order of $n2$, $n2 \pm 1$, $n2 \pm 2$, …, where $n2$ stands for half of the input number. The minimum number is 1 and the maximum number is the number of input, n, plus 1. Hence, a set of the best architectures based on the normalized mean squared error (NMSE) is chosen for use in forecasting. NMSE is defined as follows,

$$\mathrm{NMSE} = \frac{\sum_k (x_k - \hat{x}_k)^2}{\sum_k (x_k - \bar{x}_k)^2}. \tag{6}$$

The out-of-sample testing data were then used to measure the performance. The data were first preprocessed by normalizing them within the scale of -1 to 1. After training and forecasting, the neural network outputs need to be scaled back to their normal values. The general partition rule for training, validation and testing set is 70, 20 and 10%, respectively, according to the authors' experience.

The characteristics of our model are summarized in the last column of Table 1, as a comparison with the other existing works shown in the same table.

Table 3
Entries of each data set. All: number of patterns in the data set; Train, Validation, Test: number of patterns in training, validation and testing data set; Max, Min: maximum and minimum values of option price column in the data set

Name	All	Train	Validation	Test	Max	Min
op95c	17790	12490	3500	1800	6300.0	5.0
op95c-s-in:	3324	2334	660	330	7000.0	980.0
op95c-s-at:	5694	3984	1140	570	2000.0	5.0
op95c-s-out:	8772	6142	1750	880	1140.0	5.0
in	1433	1181	114	138	7000.0	270.0
at	1542	1281	116	145	1900.0	10.0
out	1773	1478	154	141	1400.0	5.0

5. Results and discussion

Several neural network models are studied in this paper. First, we start our experiment for the whole data set, namely the op95c model. The partition of data is according to the time sequence, e.g. the earlier part of data is used for training, the later part of the data for validation while the last part for testing or forecasting. Second, we mix the data up and sort them by time-to-maturity. Thus the shorter time-to-maturity data is used for training and longer time-to-maturity for testing. This is not fair since we force the neural network to forecast on certain time-to-maturity periods which the network never has a chance to learn. As expected, the results are very bad and hence we will not elaborate it here. As another alternative, we randomly select 10% of data from the whole data set as the testing set. Then we select another 20% as the validation set. The remaining part is used for training. We name this model op95c-r.

From the results shown in Table 4, we note that even though we can get reasonable convergence for randomly selected data, it performs much worse for the out of sample testing data. Specifically, even op95c-r model's training error for in sample data is improved by 63% comparing with op95c model, its testing error is degraded by nearly 400%. In order not to overfit the neural network, we do not aim for very low training errors. This will enhance the generalization ability of the model. Indeed the underlying rules (relation between inputs and outputs), if any, are different for different time periods, namely, training, validation and testing periods. Thus, when the data is not ordered by time sequence some of the information, such as the trend of σ, captured by the neural network will be different as some information is time sensitive. Random ordering of data thus renders the training meaningless. The final adopted training procedure is to partition data according to time.

In the Black–Scholes model, volatility is estimated based on the historical data such as the standard deviation of daily returns for a certain period. In our neural network model, the volatility is not fed in. This is based on the consideration that the daily asset figures are fed to the network and thus the daily returns, the fundamental factor of volatility, can be easily captured by the network. In stead of partition

Table 4
Neural network results of different models for option

File	Model	Training Err.	Testing Err.	Data type
op95c	3–2–1	0.042494	0.102806	sequence whole set
op95c-r	3–2–1	0.026822	0.442644	random whole set
op95c-r-out	3–5–1	0.304299	0.107606	random in set
op95c-r-at	3–6–1	0.561202	0.914014	random in set
op95c-r-in	3–5–1	0.094655	0.204800	random in set
op95c-s-out	3–3–1	0.326512	0.321604	sequence in set
op95c-s-at	3–4–1	0.230019	0.814014	sequence in set
op95c-s-in	3–6–1	0.062432	0.268731	sequence in set
out	3–3–1	0.001601	0.019461	sequence extreme
at	3–2–1	0.013107	0.019876	sequence extreme
in	3–2–1	0.019102	0.015196	sequence extreme
out-r	3–3–1	0.003472	0.474915	random extreme
at-r	3–4–1	0.016882	0.564408	random extreme
in-r	3–6–1	0.070034	0.215196	random extreme
out-r-d	4–2–1	0.080245	0.070445	random extreme with date
at-r-d	4–5–1	0.563329	0.339621	random extreme with date
in-r-d	4–3–1	0.329909	0.050324	random extreme with date

data in transaction order, we randomly partition training, validation and testing data in the second experiment. As the result is much better for data ordered in time sequence than in random order, it implies that time and volatility which is time-dependent can be captured by the neural network. In reality, there is no such data on volatility available in the market. When modeling option prices in other conventional methods, researchers try to capture the market movement based on the so-called historical or implied volatility. Our neural network modeling, on the other hand, captures σ in its black box. Thus σ is an important factor in both cases. However, it is encapsulated in our neural network model. but it is estimated separately in the conventional methods. This is evident from the observation that when the data is randomly ordered, we actually disturb the trend of volatility and thus the option prices are not forecast well.

The strike price of an option is only the agreement on price of the future. When the contract expires, the market price should be the same as the strike price. If the market price of an underlying asset is below the strike price, the option holder will not exercise the option because exercising option would be unprofitable or out-of-the-money. If the market price turns out to be above the exercise price exercising option would be profitable or in-the-money. To compare the stock price with the strike price will make sense in our case. The quotient of stock prices to strike prices or moneyness is used to partition the data set as the Black–Scholes model is known to misprice out-of-the-money and in-the-money options [23]. The aim of this partition is to find out which set the neural network will model better. Third, we partition the data according to its moneyness into three subsets. Quotients of stock prices to strike prices less than 0.94, between 0.94 and 1.06 and greater than 1.06 are used as the rules to partition the data. Referring to Table 2 for the number of each set, we name them as op95-s-in, op95-s-at and op95-s-out, respectively, while randomly chosen models for op95-r-in, op95-r-at and op-r-out. The results shown in Table 4 indicate that at-the-money would be the worst model in terms of fitness. Since the Black–Scholes model works well for at-the-money, this would imply that even the neural network model cannot replace entirely the traditional Black–Scholes model.

Table 5
Neural network results compared with Black–Scholes model in terms of NMSE

Data set	NN model	NN testing Err.	Black–Scholes Err.
in-the-money	3–3–1	0.019461	0.022042
at-the-money	3–2–1	0.019876	0.018293
out-of-the-money	3–2–1	0.015196	0.021932

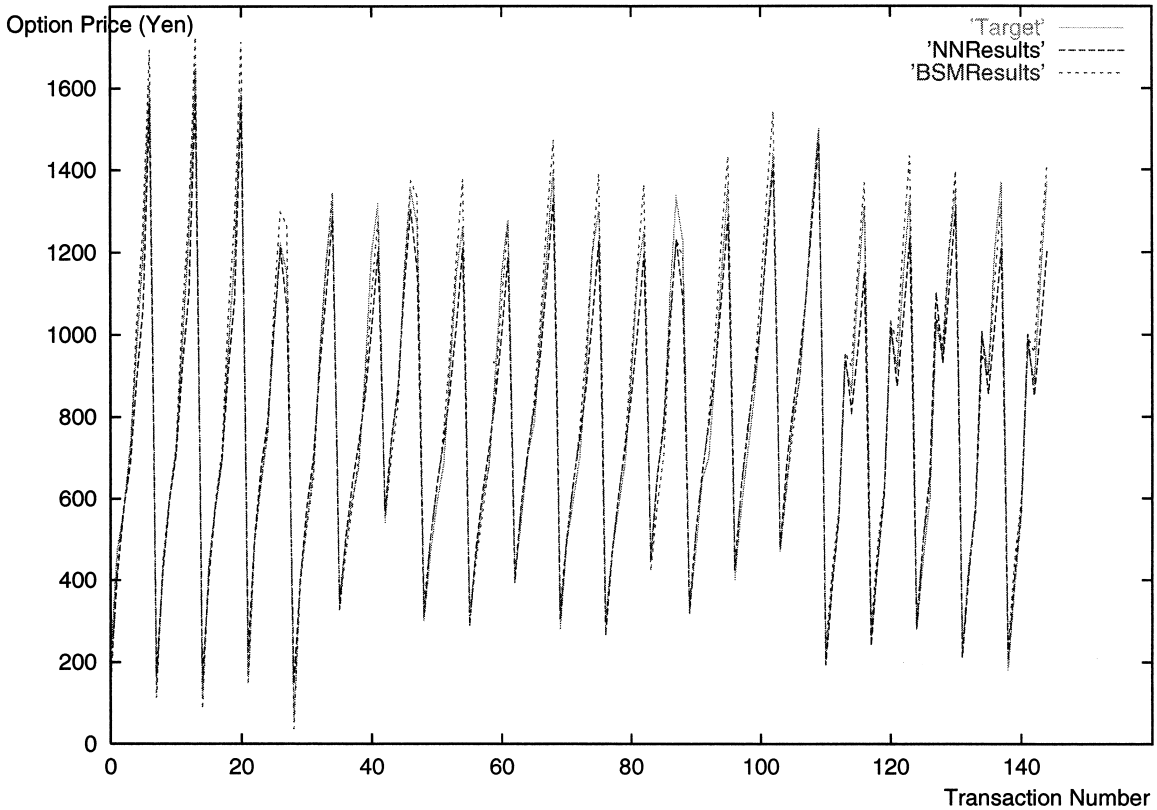

Fig. 2. A 1-month forecasting comparison of neural network and the Black–Scholes results on at-the-money call option prices in December 1995.

Both can be complementary to each other, as one model's weakness may be remedied by the other model.

Fourth, we use the extreme moneyness to partition the data set as mentioned in Section 3. The names of models are *in*, *at* and *out* accordingly. Last, we use the same number of the above three data sets but randomly choose each subset from the data set obtained in the fourth experiment to examine the time effect of this model. The names of the three sets are in-r, at-r and out-r respectively. Table 4 shows a superior performance of the sequential model over the random model, indicating that the time sequence is a very important factor for option pricing. In view of this, we add a time factor to the last random model. The three random data sets are changed by giving trading dates to them and denoted as in-r-d, in-at-d and out-at-d, respectively. Table 4 shows that adding trading dates does improve the forecasting result.

The above comparisons are among the neural network outputs. We next compare our best model with the Black–Scholes model in Table 5. The results suggest that the neural network's results are better than the Black–Scholes model's for in-the-money and out-of-the-money data. On the other hand, the Black–Scholes model does a very good job for at-the-money data. Fig. 2 shows a one month forecasting results. Fig. 3 is a close look for the first 50 transactions (first 10 calendar days).

6. Conclusion

Economic data forecasting is always and will remain difficult because such data are greatly influenced by economical, political, international and even natural shocks. Neural networks have the ability to model nonlinear patterns and learn from the historical data. They can be used in option pricing or option price forecasting by feeding sufficient known factors. Instead of studying the US market, Japan Nikkei 225 futures are modeled by neural networks in our work. Our

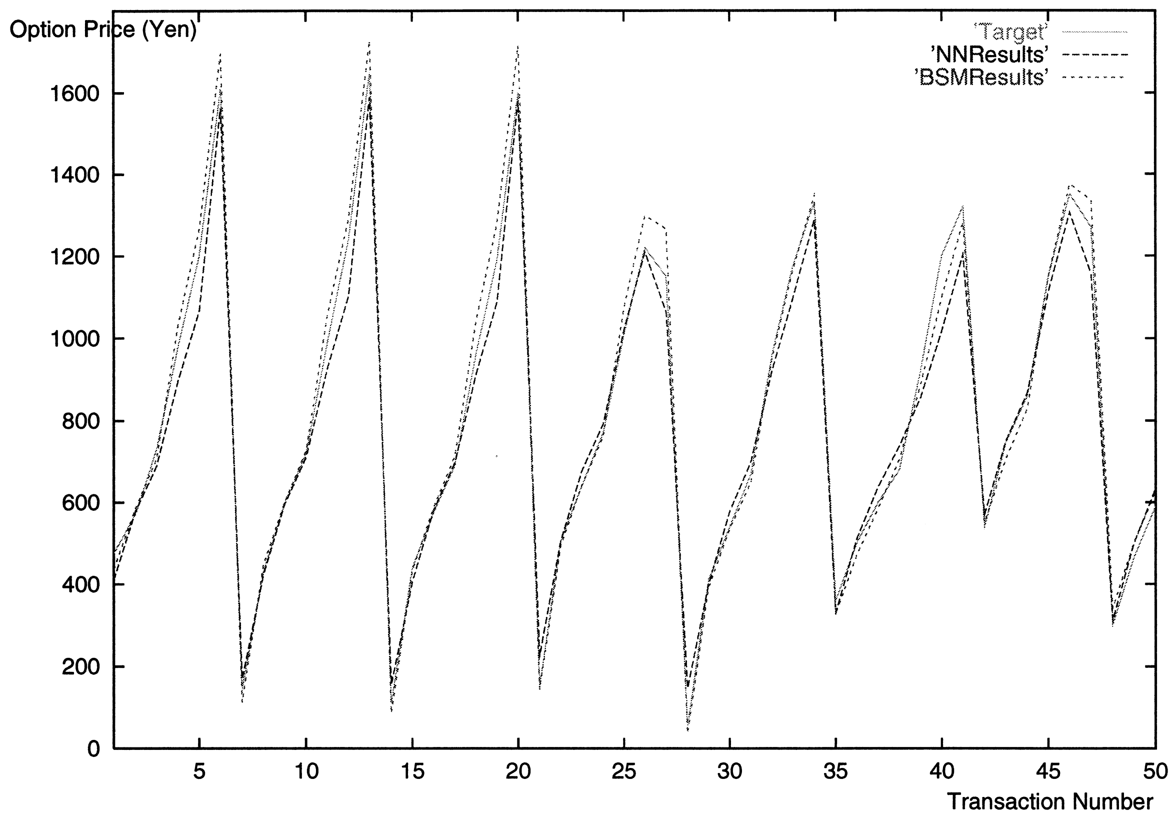

Fig. 3. A close look for Fig. 2. The option prices of the first 50 transactions in December 1995.

results suggest that our neural network model can beat the Black–Scholes model in volatile markets. The Black–Scholes model works when its assumption holds. The Black–Scholes model is like a theoretical model that works for pure data but the markets are more volatile than what the theory expects. As White [30] suggests that the relationship between neural networks and statistical approaches for time series forecasting are complementary, we will also suggest that the relationship between the Black–Scholes model and neural network models is complementary in nature. In a perfect market, the Black–Scholes model should be used as the first choice. On the other hand, the neural network model can be used for American option pricing for which the traditional Black–Scholes model cannot be used. Furthermore, the neural network model should be used for volatile markets which violate the Black–Scholes' assumption of constant σ. In using the neural network model, time should be included as an input either by time sequencing the data or by having the trading date as an extra input. To partition the data according to moneyness helps model building. Sorting data by its known characteristics enhances the convergent abilities of the neural network and thus a better forecasting results can be achieved.

At the current stage it may be too early to claim that the neural network forecasting ability for option prices is generally better than the conventional model, but there are several advantages of the neural network approach. Instead of reducing the parameters estimating bias, e.g. for volatility σ, the neural network approach builds models directly through learning of known factors.

To make the neural network model a more efficient forecasting tool, a portfolio model which consists of a set of neural network models is under exploration. The decision of trading will not only be based on just one network output. The portfolio model will direct the trading based on the majority of network outputs or weighted outputs. We hope that it can outperform the Black–Scholes model for all data sets.

Neural network models are noninterpretable models. The knowledge is encoded in the weights associated with the connections of the neural networks. Demystifying a neural network is desirable for many reasons

[24]. Rules extraction [28] of neural network models will also be studied to make the trading decision more transparent.

The partition of data set according to moneyness will benefit the practitioners. As three models are built up and experimented based on three data sets, those who prefer less risk and less returns may use the traditional Black–Scholes model's results as the model works best for at-the-money options. On the other hand, those who prefer high risk and high returns may choose to use the neural network model's results, in view of its better performance for in-the-money and out-of-the-money options. A decision support system based on the returns and risk will be explored in our future research. Although σ can be encapsulated in neural network to a certain level, the results do not suggest that σ can be thrown away. As a neural network may not guarantee a full coverage of σ, further research can introduce more inputs such as time series data of X, S or even history volatilities. The comparison of neural network models with other Black–Scholes derivatives will also be conducted to analyze if neural network models can really overcome some of the drawbacks of the Black–Scholes assumption.

Acknowledgements

The authors are grateful to three anonymous referees for their insightful comments enabled us to make significant improvements. We would like to thank Dr. Hong Yu for invaluable discussions.

References

[1] Akgiray V. Conditional heteroskedasticity in time series of stock returns: evidence and forecasts. J Bus 1989;61:55–80.
[2] Black F, Scholes M. The pricing of options and corporate liabilities. J Polit Econ 1973;81:637–59.
[3] Callen JL, Kwan CCY, Yip PCY, Yuan YF. Neural network forecasting of quarterly accounting earnings. Int J Forecasting 1996;4(12):475–82.
[4] Chang CW, Chang JSK, Lim K-G. Information-time option pricing: theory and empirical evidence. J Financ Econ 1998;48(2):211–42.
[5] Cox JC, Ross SA. The valuation of options for alternative stochastic processes. J Financ Econ 1976;3:145–60.
[6] Derman E, Kani I. Riding on a smile. Risk 1994;7:32–9.
[7] Gately E. Neural networks for financial forecasting. Jone Wiley & Sons, 1996.
[8] Geske R. The valuation of compound options. J Financ Econ 1979;7:63–81.
[9] Hansen JV, Nelson RD. Neural networks and traditional time series methods: a synergistic combination in state economic forecasts. IEEE Trans Neural Networks 1997;8(4):863–73.
[10] Hill T, O'Connor M, Remus W. Neural network models for time series forecasts. Manage Sci 1996;42:1082–92.
[11] Hull J, White A. The price of options on assets with stochastic volatilities. J Financ 1987;42:281–300.
[12] Hutchinson JM, Lo A, Poggio T. A nonparametric approach to pricing and hedging derivative securities via learning networks. J Financ 1994;49:851–89.
[13] Lajbcygier P, Boek C, Palaniswami M, Flitman A. Neural network pricing of all ordinaries SPI options on futures. In: Refenes A-PN, Abu-Mostafa Y, Moody J, Weigend A, editors. Neural networks in financial engineering, Proceedings of 3rd International Conference On Neural Networks in the Capital Markets, Oct 1995, London. World Scientific, 1996. p. 64–77.
[14] Makridakis S, Wheelwright SC, Hyndman RJ. Forecasting: methods and applications. John Wiley, 1998.
[15] Malliaris M, et al. Using neural networks to forecast the S&P 100 implied volatility. Neurocomputing 1996;10(2):183–95.
[16] Merton RC. Theory of rational option pricing. Bell J Econ 1973;4(1):141–83.
[17] Merton RC. Option pricing when underlying stock returns are discontinuous. J Financ Econ 1976;3:125–44.
[18] Naik V. Option valuation and hedging strategies with jump in volatility of asset returns. J Financ 1993;48:1969–84.
[19] Peters EE. Chaos and order in the capital markets: a new view of cycles, prices, and market volatility. John Wiley & Sons, 1991.
[20] Peters EE. Fractal market analysis: applying chaos theory to investment and economics. John Wiley & Sons, 1994.
[21] Qi M, Maddala GS. Option pricing using artificial neural networks: the case of S&P 500 index call options. In: Refenes A-PN, Abu-Mostafa Y, Moody J, Weigend A, editors. Neural networks in financial engineering, Proceedings of 3rd International Conference On Neural Networks in the Capital Markets, Oct 1995, London. World Scientific, 1996. p. 78–91.
[22] Refenes AN, Zapranis A, Francis G. Stock performance modeling using neural networks: a comparative study with regression models. Neural Network 1994;5:961–70.
[23] Ritchken P. Options: theory, strategy, and applications. Glenview, IL: Scott, Foresman, 1987.
[24] Roadknight CM, Balls GR, Mills GE, Palmer-Vrown D. Modeling complex environmental data. In: IEEE Transactions on Neural Networks, vols. 8 and 4, 1997. p. 852–62.
[25] Rubinstein M. Displaced diffusion option pricing. J Financ 1983;38:213–7.
[26] Rubinstein M. Implied binomial trees. J Financ 1994;49:771–818.
[27] Rumelhart DE, McClelland JL. Parallel distributed processing: explorations in the micro-structure of cognition, vol. 1. The MIT Press, 1986. p. 318–62.
[28] Setiono R. Extracting rules from neural networks by pruning and hidden-unit splitting. Neural Comput 1997;9(1):205–25.
[29] Weigend AS, Gershenfeld NA, editors. Time series prediction: forecasting the future and understanding the

past: Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, held in Santa Fe, NM, May 14–17, 1992. Addison-Wesley, 1994.
[30] White H. Learning in artificial neural networks: a statistical perspective. Neural Comput 1989;1:425–64.
[31] Wong BK. Bibliography of neural network business applications research: 1988–September 1994. Expert Syst 1995;12(3):253–62.
[32] Yao J, Li Y, Tan CL. Forecasting the exchange rates of CHF vs. USD using neural networks. J Comput Intell Financ 1997;5(2):7–13.