Using neural networks to forecast the S&P 100 implied volatility

Article *in* Neurocomputing · March 1996

DOI: 10.1016/0925-2312(95)00019-4 · Source: DBLP

2 authors:

Mary Malliaris
Loyola University Chicago

Linda M. Salchenberger
Marquette University

ELSEVIER

Neurocomputing 10 (1996) 183–195

NEUROCOMPUTING

Using neural networks to forecast the S&P 100 implied volatility

Mary Malliaris, Linda Salchenberger *

Management Science Department, Loyola University Chicago, 820 N. Michigan Ave., Chicago, IL 60611, USA

Received 27 September 1993; accepted 19 January 1995

Abstract

The implied volatility, calculated using the Black–Scholes model, is currently the most popular method of estimating volatility and is considered by traders to be a significant factor in signalling price movements in the underlying market. Thus, the ability to develop accurate forecasts of future volatility allows a trader to establish the proper strategic position in anticipation of changes in market trends. A neural network has been developed to forecast future volatility, using past volatilities and other options market factors. The performance of this network demonstrates its value as a predictive tool.

Keywords: Neural networks; Financial AI applications; Forecasting; Volatility

1. Introduction

The desire to forecast volatility of financial markets has motivated a large body of research during the past decade [10]. Volatility is a measure of price movement often used to ascertain risk and to signal large moves in the underlying markets [3]. Relationships between volatility and numerous other variables have been studied in an attempt to understand the underlying process so that accurate predictions may be made [4,7,11,14,18]. The predictability of market volatility is important for accurate valuation of stocks to determine expected market return [17]. Hull and White [12] showed that profits can be earned by trading on market information based on changes in volatility. Prediction of volatility is critical in designing optimal dynamic hedging strategies for options and futures [1,9]. Options traders use

* Corresponding author. Email: lsalche@orion.it.luc.edu

SSDI 0925-2312(95)00019-4

estimates of volatility to predict closing prices, to determine the optimal position to take early in the day. Recent examples of options traders capitalizing on a strategy based on a prediction of high volatility occurred in August, 1993, in the silver and grains markets [3].

The purpose of this research is to present a neural network which accurately forecasts the volatility most often used by traders, specifically, implied volatility calculated with the Black-Scholes formula. Neural networks, which have been shown to effectively model nonlinear relationships, prove to be a useful approach in predicting nearby options volatility in all cases tested and can thus be used to develop reliable forecasts.

The paper is organized as follows. In Section 2, we present the concepts of implied and historical volatility and show how each is computed. This is followed by a general discussion of neural networks as prediction models. The data set and methodology used to develop the predictions for each model are presented in Section 4. Development of the neural network models for volatility prediction is detailed in Section 5. Section 6 provides the results of the networks and compares the predictions to the implied volatility estimates. A discussion of the results is presented in Section 7, along with suggestions for future research.

2. Calculating historical and implied volatilities

In their seminal work on pricing options, Black and Scholes [2] assumed that the price of the underlying asset follows an Itô process

$$\mathrm{d}S/S = \mu \ \mathrm{d}t + \sigma \ \mathrm{d}Z \tag{1}$$

where dS/S denotes the rate of return, μ is the instantaneous expected rate of return, σ is the expected instantaneous volatility and Z is a standardized Wiener process, or dZ is a continuous-time random walk. The Black–Scholes option pricing formula for calculating the equilibrium price of call options is

$$C = S \cdot N(d_1) - Xe^{-rT} \cdot N(d_2) \tag{2}$$

where C is the market price to be charged for the option, N is the cumulative normal distribution, T is the number of days remaining until expiration of the option expressed as a fraction of a year, S is the price of the underlying asset, r is the risk-free interest rate prevailing at period t, X is the exercise price of the option and d_1 and d_2 are given by

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(r + \frac{\sigma^2}{2}\right) \cdot T}{\sigma\sqrt{T}} \tag{3}$$

and

$$\mathrm{d}_2 = \mathrm{d}_1 - \sigma\sqrt{\mathrm{T}} \tag{4}$$

where σ^2 is the variance rate of return for the underlying asset. For any time interval $[0, t]$ of length t, the return on the underlying asset is normally distributed with variance $\sigma^2 t$.

Their formula expresses the call price C, as a function of five inputs

$$C = C(S, X, T, \sigma, r) \tag{5}$$

Observe that the μ of Eq. (1) does not appear in (5). The mathematical derivation of the call option pricing formula as shown in [13] or [15] shows that arbitrage requires that the per unit of risk excess returns between two appropriately designed portfolios must be equal. Making the necessary substitutions in this arbitrage relationship, the term containing μ drops out. With μ now out of the picture and with four of the five remaining variables directly observable, an estimate of the asset's volatility σ in (5) becomes the focal point of attention for both theorists and traders.

There are two main approaches to estimating and predicting the nonconstant σ: the historical approach and the implied volatility approach. The historical approach, based on the statistical definition of volatility, is the simplest because tomorrow's volatility σ_{t+1} is an estimate obtained from a sample, of a given size, of past prices of the underlying asset. Suppose that the sample size is n and let

$$S_{t-n+1}, \ldots, S_{t-1}, S_t$$

denote daily historical prices for the underlying asset. To get an estimate for σ_{t+1}, first compute daily returns, r_{t-i}, $i = 0, \ldots, n-2$, where

$$r_{t-i} = \ln(S_{t-i}) - \ln(S_{t-i-1}).$$

For a sample of n historical prices, we obtain $(n-1)$ rates of daily return. The annualized standard deviation of these rates of return is defined as the volatility and called historical volatility and is used as an estimate of σ_{t+1}. The nearby historical volatility uses 30 days of data, the middle historical volatility uses 45, and the distant historical volatility has 60 daily prices.

An obvious problem with the historical approach is that it assumes that future volatility will not change and that history will exactly repeat itself. Markets, however, are forward looking and numerous illustrations can be presented to show that historical volatility does not always anticipate future volatility. A better estimate, the one most used by traders to price options, comes from the Black-Scholes option pricing model itself [5].

Simply stated, supporters of implied volatility claim that tomorrow's volatility σ_{t+1} can only be estimated during trading tomorrow, i.e. in real time. As option prices are being formed by supply and demand considerations, each trader assesses the asset's volatility prior to making his or her bid or ask prices and, accepting the consensus price of a call as a true market price reflecting the corporate opinions of the trading participants, one solves the Black–Scholes model for the volatility that yields the observed call price. When volatility is calculated in this way, it is called the implied volatility, with the adjective 'implied' referring to the volatility estimate obtained from the Black-Scholes pricing formula. Unlike historical volatility, which uses past returns, the implied volatility is forward-looking to the stock's future returns from now to the time of the expiration of the option. This implied volatility technique has become the standard method of estimating volatility at the moment of trading.

3. Neural networks for prediction

Neural networks are an information processing technology which model mathematical relationships between inputs and outputs. Based on the architecture of the human brain, a set of processing elements or neurons (nodes) are interconnected and organized in layers. These layers of nodes can be structured hierarchically, consisting of an input layer, an output layer, and middle (hidden) layers. Each connection between neurons has a numerical weight associated with it which models the influence of an input cell on an output cell. Positive weights indicate reinforcement; negative weights are associated with inhibition. Connection weights are 'learned' by the network through a training process, as examples from a training set are presented repeatedly to the network. Each processing element has an activation level, specified by continuous or discrete values. If the neuron is in the input layer, its activation level is determined in response to input signals it receives from the environment. For cells in the middle or output layers, the activation level is computed as a function of the activation levels on the cells connected to it and the associated connection weights. This function is called the transfer function or activation function and may be a linear discriminant function, i.e. a positive signal is output if the value of this function exceeds a threshold level, and 0 otherwise. It may also be a continuous, nondecreasing function. Feedforward networks map inputs into outputs with signals flowing in one direction only, from the input layer to the output layer.

While there are dozens of network paradigms, the backpropagation network has frequently been applied to classification, prediction, and pattern recognition problems. Financial applications of neural networks include underwriting [6], bond-rating [8], predicting thrift institute failure [20], and estimating options prices [16]. The term backpropagation technically refers to the method used to train the network, although it is commonly used to characterize the network architecture. In this learning algorithm, mean squared error and gradient descent are employed to determine a set of weights for the trained network. At each iteration, current weights are updated by minimizing the mean squared differences between the actual response of the system to a given example and the desired response. The nonlinear response functions generate gradients of the error function with respect to the weights and the chain rule is used to determine the appropriate weight changes which propagate back through the layers of the network. For more details of this method, see [19]. Currently, a number of variations on this method exist which overcome some of its limitations.

Nonlinear, multilayer, feedforward networks differ from traditional modelling techniques in several ways. Relationships between inputs and outputs are learned during a training process in which the network is repeatedly presented with historical examples. Neural networks possess the ability to approximate arbitrary mappings with no apriori assumptions about the nature of the underlying model required. Also, no assumptions about the distributions of the variables are required and the variables may be highly correlated.

4. Data and methodology

Data have been collected for the most successful options market: the S&P 100 (OEX), traded at the Chicago Board Options Exchange. Daily closing call and put prices and the associated exercise prices closest to at-the-money, S&P 100 Index prices, call volume, put volume, call open interest and put open interest were collected from the Wall Street Journal for the calendar year 1992. The data is organized into trading periods, with each period beginning on the first trading day after the third Friday of a month and ending on the third Friday of the following month. Data from January 1, 1992 through June 19, 1992 was used to develop training sets for the neural network forecasts made for trading periods beginning on June 22, 1992.

The historical volatility used an Index price sample of size 30 and was computed for each trading day in 1992. We used the Black–Scholes model to calculate implied volatilities for the closest at-the-money call for three contracts: those expiring in the current month, those expiring one month away, and those expiring two months away (nearby, middle, and distant, respectively). Thus, we have approximately 250 observations in each series of volatilities for use in our study.

The historical and implied volatility for the nearby contract for June 22 through December 30, 1992 are shown in Fig. 1. As can be observed, the historical estimate

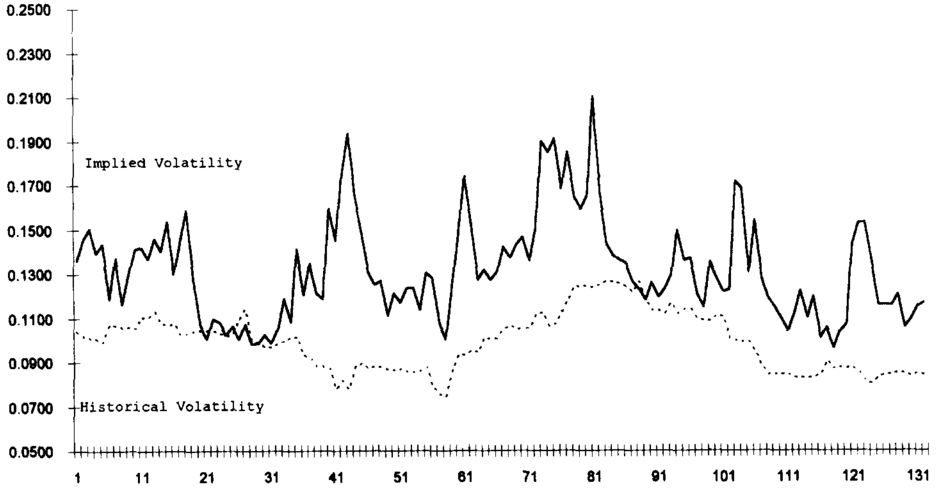


Fig. 1. Graph of implied and historical volatilities.

significantly underestimates the volatility used by most traders, i.e. the implied volatility. Since the historical volatility is an average based on returns from 30 preceding days, it is not surprising that the estimate smoothes out the peaks, giving a value for each day which is less variable, and thus less sensitive to daily market fluctuations. The implied volatility for any given day uses only trading information from that day, not a previous time period, to generate a value.

The difference in the historical and implied volatilities can be seen by calculating the MAD (mean absolute deviation) and MSE (mean squared error) over this period. The MAD and MSE from June 22 through Dec. 30 were 0.0331 and 0.0016. The proportion of times which the historical volatility correctly matches an increase or decrease in implied volatility is 0.4439, i.e. a little less than half of the time.

A neural network was developed to forecast implied volatility and tested on seven independent out-of-sample forecasts for the trading periods between June 22 through December 30, 1992. For each of the seven periods, MAD, MSE and the number of correct directions of the forecast were calculated.

5. Development of the neural networks

To develop a neural network which is capable of generalizing a relationship between inputs and outputs, the training set selected must contain a sufficient number of examples which are representative of the process which is being modelled. Therefore, the neural network models developed to predict volatility were trained with data sets from January 1 through June 19 and used to make predictions for seven separate trading cycles beginning with June 22 and ending December 30. All prior data was used when predicting the volatility for the next trading period. Predicting the volatility for the next period is a rather rigorous test of the forecasting capabilities of the network.

5.1 Selection of input variables

The selection of the input variables is a modelling decision and one which can greatly effect network performance. While neural nets can approximate a wide range of functions, training time can be reduced if the data is preprocessed to reflect known relationships. This relieves the network of the task of mapping simple arithmetic functions during training so it can devote more time to discovering higher order relationships. There is no well-defined theory to assist with the selection of input variables and heuristic methods are employed. One approach is to include all the variables in the network and perform an analysis of the connection weights or a sensitivity analysis to determine which may be eliminated without reducing predictive accuracy. An alternative is to begin with a small number of variables and add new variables which improve network performance. In this research, the latter approach was used and variables were selected using financial theory, sensitivity analysis, and correlation analysis. Table 1 is a list of all the variables derived from the data set and tested.

Table 1
Definition of input variables

VOLt-3	volatility, lag 3
VOLt-2	volatility, lag 2
VOLt-1	volatility, lag 1
VOLt	current volatility
CLOSECH	change in daily closing price
DAYS	days to expiration
VOLMID	middle volatility
VOLDIST	distant volatility
CSTR + MKTN	current call exercise price plus market price
CSTR + MKTM	middle market call exercise price plus market price
PSTR + MKTN	current put exercise price plus market price
PSTR + MKTM	current put exercise price plus market price
CHPUTOPEN	change in put open interest

All variables reflect closest to at-the-money prices for nearby markets, where contracts expire in the current month and middle markets, where contracts expire one month away.

A training set consisting of observations from January 1 to November 20 was used to perform these experiments. An out-of-sample test set consisted of the observations from November 23 to December 30. The first networks tested input variables representing volatility, lagged from 3 to 7 periods. A lag period of four resulted in the lowest mean squared error (MSE). The MSE for networks developed with 4, 5, and 6 lags is shown in Fig. 2. Financial variables were developed next, to maximize information content in the input variables. Variables were tested and if their inclusion reduced network error, they were added to the network. While the closing price itself did not improve performance, a variable which

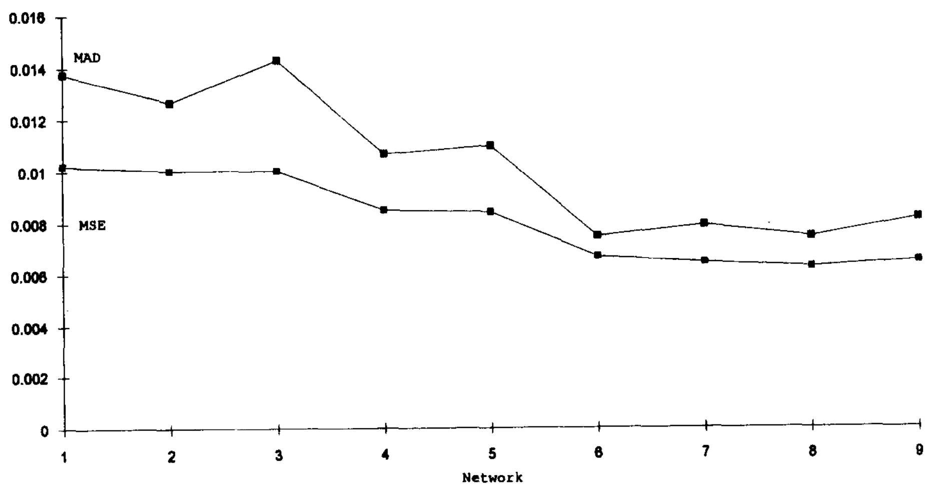


Fig. 2. Comparison of networks.

Table 2
Neural networks and input variables

Network	# of middle nodes	Input variables
1	3	volatility, 4 lags
2	4	volatility, 5 lags
3	4	volatility, 6 lags
4	4	VOLt-3, VOLt-2, VOLt-1, VOLt, CLOSECH, DAYS
5	6	VOLt-3, VOLt-2, VOLt-1, VOLt, CLOSECH, DAYS, VOLMID, VOLDIST
6	7	VOLt-3, VOLt-2, VOLt-1, VOLt, CLOSECH, DAYS, CSTR + MKTN, CSTR + MKTM, PSTR + MKTN, PSTR + MKTM
7	5	VOLt-3, VOLt-2, VOLt-1, VOLt, CLOSECH, DAYS, CSTR + MKTN, CSTR + MKTM, PSTR + MKTN, PSTR + MKTM, CHPUTOPEN, VOLMID, VOLDIST
8	7	VOLt-3, VOLt-2, VOLt-1, VOLt, CLOSECH, DAYS, CSTR + MKTN, CSTR + MKTM, PSTR + MKTN, PSTR + MKTM, CHPUTOPEN, VOLMID, VOLDIST
9	9	VOLt-3, VOLt-2, VOLt-1, VOLt, CLOSECH, DAYS, CSTR + MKTN, CSTR + MKTM, PSTR + MKTN, PSTR + MKTM, CHPUTOPEN, VOLMID, VOLDIST

computed daily changes in the closing price did. Next, the change in closing price and days to expiration were tested with the volatility lags as network #4 (see Table 2). Adding two new variables, middle and distant volatility in Network #5 reduced MSE. Further tests showed that the sum of current call exercise price and market price for puts and calls also reduced MSE. The variables included in each model, and MSE and MAD for each network is found in Table 2 and Fig. 2.

A variety of other financial variables were tested and not included in the final network. In all, 13 variables were included: change in closing price, days to expiration, change in open put volume, the sum of the at-the-money strike price and market price of the option for both calls and puts for the current trading period and the next trading period, daily closing volatility for the current period, daily closing volatility for the next trading period, and four lagged volatility variables. As shown in Fig. 2, Network 8, with 7 middle layer nodes, proved to be the best when tested and compared to networks with 5 and 9 middle layer nodes.

The backpropagation network developed to predict volatility has 13 input nodes representing the independent variables used for prediction, one middle layer consisting of 7 middle nodes, and an output node representing the volatility. The cumulative Delta Rule for training was selected, with an epoch size of 16, and decreasing learning rate initially set at 0.9 and an increasing momentum, initially set at 0.2. The networks were trained using NeuralWorks Professional II™ software from NeuralWare.

A variation on the backpropagation algorithm called fast-backpropagation was used to improve performance by reducing the number of iterations needed to achieve convergence. Essentially, an error is added to the activation value prior to the update of the weights. In traditional backpropagation, the gradient descent rule is used to update weights to decrease network error using

$$\Delta w_{ji}^{s} = -\alpha e_j^{s} x_i^{(s-1)}$$

In fast backprop, this becomes

$$\Delta w_{ji}^{s} = -\alpha e_j^{s}\{x_i^{(s-1)} + e_i^{(s-1)}\}$$

where w_{ji}^{s} is the weight connecting the ith neuron in layer $(s-1)$ to the jth neuron in layer s, α is the learning coefficent, $e_i^{(s-1)}$ is the local error at neuron i in layer $(s-1)$. This learning rule was proposed by Tariq Samad [21].

6. Results

6.1 Network estimates of future volatility

We evaluated the performance of the neural network by measuring MAD, MSE, and the number of times the direction of the volatility (up or down) was correctly predicted. These results are shown in Fig. 3 and Table 3, where comparisons are made between the volatility forecasted by the network and tomorrow's implied volatility. The overall MAD for the entire period was 0.0116 and the MSE was 0.0001. The overall proportion of correct direction predictions was 0.794. The correlation between the neural network forecast and the future implied volatility was 0.8535 with a significance level of 0.0001.

The errors from the forecasts made for June 22 through Dec 30 are shown in Fig. 4. The greatest errors occur close to the expiration of a period. All but two of

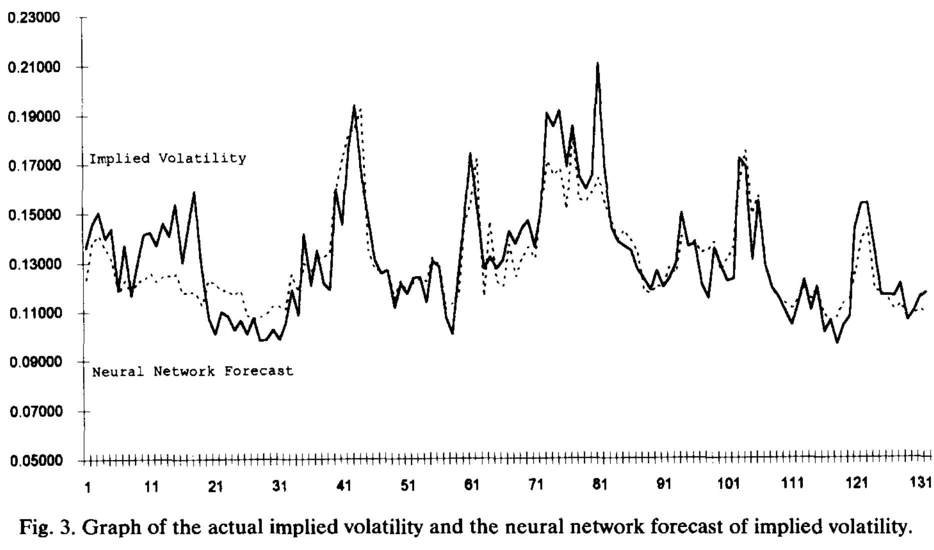


Fig. 3. Graph of the actual implied volatility and the neural network forecast of implied volatility.

Table 3
Neural network and implied volatilities

Dates of forecast	MAD	MSE	Proportion of same directions
Jun 22–Jul 19	0.0148	0.0003	16/19 = 0.842
Jul 20–Aug 21	0.0107	0.0002	16/25 = 0.640
Aug 24–Sep 18	0.0056	0.0001	13/18 = 0.722
Sep 21–Oct 16	0.0127	0.0003	19/20 = 0.950
Oct 19–Nov 20	0.0059	0.0001	20/25 = 0.800
Nov 23–Dec 18	0.0068	0.0001	15/18 = 0.833
Dec 21–Dec 30	0.0039	0.0000	5/6 = 0.833

the 132 errors have an absolute value less than 0.03 and all but 10 are less than 0.02 in absolute value.

6.2 Variable analysis

Forecasts of volatility have been reported for the final networks which included 13 financial variables. Table 4 shows each variable and the percentage change in the volatility prediction for a 10% and 50% change in each input, respectively. A positive change indicates the volatility prediction would be higher; a negative change indicates the predicted value would be reduced. These percentages may be used to evaluate the relative impact of each predictor variable. Not surprisingly, DAYS (days to expiration), CLOSECH (change in closing price), and CSTR + MKTN (the sum of the current call exercise price and market price) were the most significant for predicting volatility. CSTR + MKTN should be significant since, this

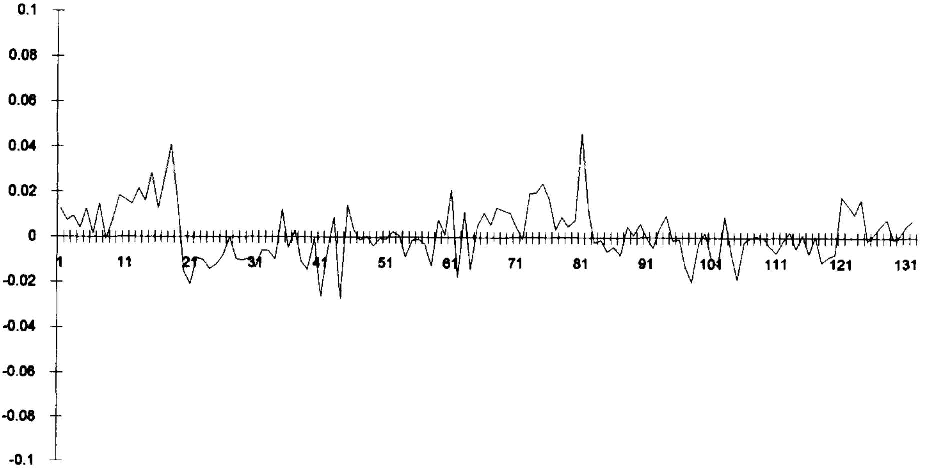


Fig. 4. Graph of errors (implied volatility–network forecast); end of trading periods occurred on days 20, 45, 63, 83, 108, 126, and 132.

Table 4
Percent change in volatility prediction

	10%	50%
VOLt-3	13.13	22.22
VOLt-2	2.07	2.85
VOLt-1	1.98	6.98
VOLt	4.00	6.21
CLOSECH	−11.91	−19.10
DAYS	−18.58	−24.20
VOLMID	36.57	49.28
VOLDIST	−5.21	−11.23
CSTR+MKTN	26.38	35.55
CSTR+MKTM	6.53	10.13
PSTR+MKTN	9.00	15.62
PSTR+MKTM	8.94	12.99
CHPUTOPEN	7.08	15.74

sum approaches the value of the underlying asset as the days to expiration approach 0. An interesting result is the relatively significant impact of VOLMID, the volatility of the related contract which expires in the next trading period. An explanation is that, as the nearby contract expires, traders begin to focus attention on the middle contracts, which will eventually replace the expired contracts.

7. Discussion

The results of this study of neural networks for forecasting volatility are encouraging. Because historical estimates are traditionally poor predictors, traders have been forced to rely on formulas like the Black-Scholes which can be solved implicitly for the real-time volatility. However, these models can only provide real-time estimates to the traders. Furthermore, they fail to incorporate knowledge of the history of volatility. The neural network model, on the other hand, employs both short-term historical data and contemporaneous variables to forecast future implied volatility, enabling the trader to take a position when the market opens which will provide a strategic advantage. For example, high implied volatility often indicates the market is about to consolidate while low volatility often signals that the market is preparing for a breakout.

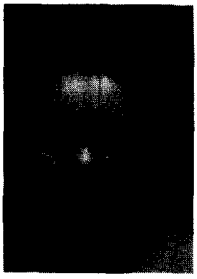
The neural network approach has two advantages which make it more useable as a forecasting tool. First, daily predictions can be made using data from previous trading cycles, thus providing a trading advantage. Secondly, in the cases we tested, the network forecasts were very accurate estimates of the volatility preferred by traders.

There are several ways to extend this research. Improvement may be possible through experimentation with other variables and network architectures. Radial basis networks offer another avenue of investigation, since these have been used for financial forecasting. In this research, we have predicted nearby volatility.

However, networks for predicting middle and distant volatility may be developed, as well, using different variables and different network architectures.

References

[1] R. Baillie and R. Myers, Modeling commodity price distributions and estimating the optimal futures hedge, *J. Applied Econometrics* 6 (1991) 109–124.
[2] F. Black and M. Scholes, The pricing of options and corporate liabilities, *J. Political Economy* 81 (1973) 637–654.
[3] D. Caplan and J. Stein, Expressed markets, implied filters, *Futures* (Nov., 1993) 54–56.
[4] J.Y. Choi and K. Shastri, Bid-ask spreads and volatility estimates: the implications for option pricing, *J. Banking and Finance* 13 (1989) 207–219.
[5] S. Choi and M.E. Wohar, Implied volatility in options markets and conditional heteroscedasticity in stock markets, *Financial Rev.* 27 (4) (1992) 503–530.
[6] E. Collins, S. Ghosh and C. Scofield, An application of a multiple neural-network learning system to emulation of mortgage underwriting judgments, *Proc. IEEE Int. Conf. on Neural Networks* (1988) 459–466.
[7] D.A. Dubofsky, Volatility increases subsequent to NYSE and amex stock splits, *J. Finance* XLVI (1) (1991) 421–431.
[8] S. Dutta and S. Shekhar, Bond-rating: a non-conservative application of neural networks, *Proc. IEEE Int. Conf. on Neural Networks* (1988) 142–150.
[9] R. Engle, C. Hong, A. Kane and J. Noh, Arbitrage valuation of variance forecasts with simulated options, Discussion Paper 92-19, Department of Economics, University of California, San Diego, 1991.
[10] R.F. Engle and M. Rothschild, Statistical models for financial volatility, *J. Econometrics* 52 (1992) 1–4.
[11] R.A. Haugen, E. Talmor and W. Torous, The effect of volatility changes on the level of stock prices and subsequent expected returns, *J. Finance* XLVI (3) (1991) 985–1007.
[12] J. Hull and A. White, The pricing of options on assets with stochastic volatilities, *J. Finance* 42 (1987) 281–300.
[13] C.F. Lee, J.E. Finnerty and D.H. Wort, *Security Analysis and Portfolio Management* (Scott, Foresman and Company, Glenview, IL, 1990).
[14] L.J. Lockwood and S.C. Linn, An examination of stock market return volatility during overnight and intraday periods, 1964–1989, *J. Finance* XLV (2) (1990) 591–601.
[15] A.G. Malliaris, *Stochastic Methods in Economics and Finance* (North-Holland, Amsterdam, 1982).
[16] M.E. Malliaris and L. Salchenberger, A neural network model for estimating option prices, *J. Applied Intelligence* 3 (1993) 193–206.
[17] R.C. Merton, On estimating the expected return on the market: an exploratory investigation, *J. Financial Economics* 8 (1980) 323–361.
[18] L.J. Merville and D.R. Pieptea, Stock-price volatility, mean-reverting diffusion, and noise, *J. Financial Economics* 24 (1989) 193–214.
[19] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing* (MIT Press, Cambridge, MA, 1986).
[20] L. Salchenberger, E.M. Cinar and N. Lash, Neural networks: a new tool for predicting thrift failures, *Decision Sci.* 23 (4) (1992) 899–916.
[21] T. Samad, Backpropagation is significantly faster if expected value of source unit is used for update, *INNS Conf. Abstracts*, 1988.



University of Chicago. Her research interests include artificial intelligence, neural networks, applications in economics and finance, and database applications. Malliaris holds a B.S. in mathematics from Louisiana State University, an M.S. in mathematics from Northwestern University, and a Ph.D. in research methodology from Loyola University. She has articles in Applied Intelligence and The Review of Quantitative Finance and Accounting.

Linda M. Salchenberger is an Associate Professor of Management Science at Loyola University Chicago. She received her Ph.D. from Northwestern University in Managerial Economics and Decision Sciences. Her research interests include applications of artificial intelligence, particularly neural networks and expert systems, and she has published articles in Applied Intelligence, Decision Sciences, and Information and Management.