CAN NEURAL NETWORKS LEARN THE BLACK-SCHOLES MODEL?: A SIMPLIFIED APPROACH

Shaikh A. Hamid
Southern New Hampshire University
School of Business

Abraham Habib
Boston University
School of Management

Working Paper No. 2005-01

School Of Business, Southern New Hampshire University, 2500 N. River Road, Manchester, NH 03106

CAN NEURAL NETWORKS LEARN THE BLACK-SCHOLES MODEL?: A SIMPLIFIED APPROACH

Abstract

Neural networks have been shown to learn complex relationships. It would be interesting to see if the networks can be trained to learn the nonlinear relationship underlying Black-Scholes type models. Interesting hypothetical questions that can be raised are: If option pricing model had not been developed, could a technique like neural networks have learnt the nonlinear form of the Black-Scholes type model to yield the fair value of an option? Could the networks have learnt to produce efficient implied volatility estimates? Our results from a simplified neural networks approach are rather encouraging, but more for volatility outputs than for call prices.

Key words: Black-Scholes model, Black model, neural networks, implied volatilities.

CAN NEURAL NETWORKS LEARN THE BLACK-SCHOLES MODEL?: A SIMPLIFIED APPROACH

One of the fundamental problems that had long challenged finance theorists and practitioners was the valuation of risky assets that grant certain options to their holders. This quest had formally started with Bachelier (1900) and was sustained over the subsequent seventy years by an impressive array of financial economists. The Black-Scholes (1973) options pricing model is the capstone of this long quest.

An interesting hypothetical question that can be posed is: If Fisher Black and Myron Scholes, or Robert Merton (1973), or some of the other acclaimed option pricing theorists after them had not been able to develop a Black-Scholes type option pricing model, could a technique like neural networks have learnt the nonlinear form of the model to yield the fair value of an option?[1]

The Black-Scholes model and its variants postulate that option price is a function of five variables: the value of the underlying asset, the standard deviation of its expected returns, the exercise price of the option, the time until the maturity of the option, and the interest rate on default-free bond. Values of four of the five variables are observable. The market has to form expectation about the standard deviation of the returns expected from the underlying asset over the life of the option. The relationship between option price and the five variables is a complex nonlinear one. But given that neural networks have been shown to learn complex relationships, it would be interesting to see if neural networks can be trained to learn the nonlinear relationship underlying the Black-Scholes type models.

The use of neural networks in finance is a recent phenomenon. They have been used to predict the economy, pick stocks, construct portfolios, spot insider dealings, assess bond risks, recognize financial distress, detect credit card fraud, improve real estate appraisal, and

identify good credit or insurance candidates. They have been found to perform well in a number of applications in which linear models fail to perform well. Specially, when it comes to forecasting financial market variables characterized by nonstationarity, neural networks incorporating nonlinear regression models have a distinct edge.

Recently, a number of studies have looked at the ability of the networks to learn Black-Scholes type model.[2] This study, was done prior to nearly all of them, and gets encouraging results with rather straightforward, simplified approach compared to the sophisticated approaches adopted by other authors. The simplicity will have appeal for those who consider the technology to be an esoteric "black-box" enveloped in mystique. It would remove some of the mist and spur greater research by finance academics in the many unresolved issues relating to its applications in economics and finance.[3]

WHAT IS A NEURAL NETWORK?

A neural network is a computational technique inspired by studies of the brain and nervous system. It is an information-processing system designed to mimic the ability of the human brain to comprehend relationships and patterns. The program imitates the structure and operations of the three dimensional lattice of network among brain cells (or neurons, and hence the term "neural"). The network learns by example. It learns by gradually fine-tuning the connections between electronic neurons in its system. The learning process of the network can be likened to the way a child learns to recognize patterns, shapes and sounds, and discerns among them. For example, the child has to be exposed to a number of examples of a particular type of tree for her to be able to recognize that type of tree later on. In addition, the child has to be exposed to different types of trees for her to be able to differentiate among trees.

A neural network has (a) processing elements, (b) connections between the elements, (c) weights associated with the connections, (d) activation function.

A neural network can have a large number of processing elements called *neurons* or *nodes* or *cells* or *units* in which information processing takes place. In its simplest form, these neurons are arranged in two layers. The first layer is the input layer -- it takes in inputs to be processed. There will be as many neurons in the input layer as there are input categories. The second layer is the output layer. It will have as many neurons as there are output categories. To process complex problems, the network can have one or more intermediate layers called hidden layers -- so called, because they are essentially hidden from the access of users. The optimal size for the number of layers and neurons per layer is a matter of experimentation. But one hidden layer can essentially map any nonlinear function. The optimal number of neurons in the hidden layer will probably be from one-half to two times the number of neurons in the input layer.

The neurons have connection links through which signals are passed. The number of layers together with the pattern of connections between the neurons determines the architecture of the network. A network may be fully connected, or partially connected. In a fully connected network, each neuron in the input layer is connected to each neuron in the next layer; each layer in the second layer is connected to each neuron in the third layer (this would be the output layer in a three-layered network). In a partially connected network, each neuron in the input layer is connected to one neuron in the second layer, and so on. Neurons in the same layer are not connected to each other either in a fully connected or partially connected network.

Each connection link goes with a weight that multiplies the signal transmitted. The network has a method of determining the weights on the connections. This is known as the training, or learning algorithm. The weights help the net to solve a problem.

Each neuron has an activation (also called "transfer" or "squashing") function which is applied to the input to determine the output from that neuron. The inputs that a neuron receives from other neurons are summed up and then passed through its transfer function to get the output from that neuron. This function tries to find a relationship between the input variables on one hand and the desired output on the other. The most commonly used transfer functions are the sigmoid[4], variations of sigmoid[5], the hyperbolic tangent[6], and the Gaussian[7]. Thus, a transfer function describes the behavior of a neuron in a neural network. For nonlinear modeling, a transfer function should be nonlinear and continuously differentiable.[8]

EXPERIMENTAL DESIGN

We use a backpropagation neural network system to explore how well it learns the nonlinear form underlying the Black-Scholes type models. We generate two types of outputs. Firstly, we find how well the network learns to yield call option price given data on four of the five variables on which the call price depends in option pricing models: price of underlying asset, its volatility, time to maturity and riskfree rate. Then we find the volatility of the underlying asset given call option price as one of the inputs along with the other four dependent variables. We compare network yielded call price (NCP) with market call price (MCP). We compare volatility implied by the network (NIV) with implied volatility that we found using Black (1976) commodity option pricing model (OIV). To do the two types of comparisons, we do standard t-tests and find mean absolute errors (MAE) and root mean squared errors (RMSE). Our data period is from March 1983 to June 1995.

We use the Black (1976) model to extract volatilities implied in the prices of call options on S&P 500 Index futures contracts. The Black model is modified to get implied

standard deviation (ISD) per trading days as opposed to calendar days which is the basis for most studies. The modified Black model for a call option, C, is given by:

$$C = e^{-rT_c} * [F*N(d_1) - K*N(d_2)] \quad ....(1)$$

where:

F = futures price;

K = strike price;

r = per calendar day risk-free interest rate;

T_c = calendar days to maturity of options;

d_1 = ln(F/K) + ½ $T_t \sigma^2$)/($\sigma \sqrt{T_t}$);

d_2 = d_1 - $\sigma \sqrt{T_t}$;

T_t = trading days until maturity of options;

σ = standard deviation of percentage changes in the futures price;

N() = standard cumulative normal probability function.

The Newton-Raphson search technique yields the standard deviation implied by a call option. If the futures and options markets are efficient and the model is properly specified, then the standard deviation implied by the market price of the call option will represent the market consensus forecast of expected volatility of the futures contract over the remaining life of the option.

We use data on nearest to maturity[9] S&P 500 Index futures and options on futures contract classes from March 1983 to June 1995. This period gives us 49 futures and futures options series spread over 49 quarters (12 1/4 years). From the data of each series we find network yielded call price (NCP), network implied volatilities (NIV), and implied

volatilities using Black model (OIV) on 9 days. NCPs are compared to market call prices (MCPs). NIVs are compared to OIVs.

NCPs, NIV and OIVs found on three of the days form longer horizons (57, 56 and 55 days to maturity of a series), three are for intermediate horizons (37, 36 and 35 days to maturity), and three are for shorter horizons (17, 16 and 15 days to maturity). (Futures options on the nearest futures series typically has about sixty to sixty-three days.)

On each of these nine days of a series, we use data on all call options on the nearest futures contracts traded on the day. Using the Black model, we extract implied volatilities for all futures options traded on the nine days in the life of a series. For the 49 series of options and underlying futures contracts, we get 4,131 observations involving 441 days (49 series x 9 days per series). This gave us plenty of observations for training the neural network and obtaining forecasts.

Using S&P 500 Index futures options contracts rather than S&P 500 Index options contracts has certain advantages. Feinstein (1989b) mentions the following advantages:

- The S&P 500 index futures and its options contracts are heavily traded in nearby pits at the Chicago Mercantile Exchange. This creates condition for simultaneous trading. Since most of the trades take place towards the end of the day, the settlement prices reported in the newspapers virtually represent actual simultaneous prices for the two contracts.

- As Kawaller, Koch and Koch (1988) mention, stock index reflects the effect of last traded prices of stocks comprising the index. This may not reflect the effect of recent trades. But in valuing stock index futures, all information about the included stocks is available. The index itself may not reflect all available information.

- In pricing options on index futures, we do not need to subtract expected dividends from stock prices as in pricing index options. The futures contract incorporates market assessment on payout over the life of the futures contract.

Randolph et al. (1990) point out that the use of futures contracts rather than the spot index help in bypassing the variance estimation problems caused by autocorrelation in the index returns.

We generate call prices from neural network. We use a three-layered backpropagation network. We have five input variables to generate call price. So the network has 5 neurons in the first layer, five in the hidden layer, and one neuron for the single output that we want. We create a file with data on the following five variables: trading days to maturity, calendar days to maturity, futures price, strike price, and interest rate. (Recall that in finding implied volatilities using the Black model we modify it to get implied volatilities per trading days as opposed to calendar days; so for consistency we include data for both of them as inputs in the neural network.) The file has data on market call price on the last column. The network trains by using data on the five input variables, producing an output, comparing with the data in the last column (the desired call price). We do not use values on volatility as one of the inputs though in options pricing models it is one of the inputs in finding call price. It would be interesting to see how close to the market call price the network generates output when the network is not fed the volatility parameter. As a next step we want to see what improvement is achieved by the network when implied volatility values from the Black model are fed along with the values of other variables mentioned above.

The output we get from the network termed network implied call price (NCP). Before we can get output from the network it is trained with 102 sets of observations out of 4,131 observations in all. It is fed with the first set of data on the five input categories; it

produces an output, compares with the desired call price (C_l) -- the market call price that goes with the five values, finds the error and feeds the magnitude of the error backwards. As the error is fed back from the output layer to the second and the first layers through the various neurons, the weights associated with each neuron is reset. Then the initial values are again fed through the network. A new output is produced. It is compared with the desired call price C_l, the new error is calculated and fed back to the network and weights again reset. This feedforward and backpropagation of errors is done 500 times on each set of data. Then the network is trained on the second data set 500 times. In this way the network is trained on 102 data sets one after another -- in each case, 500 times. Then the network is set to yield call price based on 103rd data set. In this data set the strike price is just-out-of-the-money (futures price minus strike price is closest to zero but negative)[10]. This forms our first observation. In this case, the futures and corresponding futures options contracts are 57 days to maturity.

Then the network trains with 1st data set to 107th data set and produces the second call price observation based on input data for 108th data set (again, in this data set, the strike price is just-out-of-the-money). This gives an observation with 56 days to maturity of futures and futures options contracts. Then the network is trained with 1st to 113th data set and then it produces call price observation with 114th data set in which case the strike price is again just-out-of-the-money. This serves as an observation for 55 days to maturity. In this way we get call price observations on 37, 36, 35, 17, 16 and 15 days to maturity of futures and futures options contracts. In each case we use the data set in which the strike price is just-out-of-the-money. We get a total of 47 call price observations for each of the nine maturity horizons. We use 102 data sets covering 2 futures and futures options contract classes. We have data on a total of 49 contract classes.

We compare the network implied call prices (NCP) on 57 days to maturity with market call price (MCP) on those days. Similarly, we compare NCPs for the other horizons with corresponding MCPs. The results of the tests of the differences in the means of market call prices (MCP) and network implied call prices for various days to maturity of the options are presented in Table 1.

We next create a file with data on the following six input variables: trading days to maturity, calendar days to maturity, futures price, strike price, market call price and interest rate. This time the network is trained with data on these six variables to produce the futures price volatility implied by these variables. We call these network implied volatilities (NIV). In this case, we specify six neurons in the first layer, six in the hidden layer, and one in the output layer for the single output we want. The network is trained with 102 data sets and volatility output is produced with 103rd data set as done for call price outputs. The other outputs are produced as in the case of call price outputs. Then we compare the network implied volatilities (NIV) on 57 days to maturity with option-model implied volatilities (OIV) on those days. Similarly, we compare NIVs for the other horizons with corresponding OIVs. The results of the tests of the differences in the means of option-model implied volatilities and network implied volatilities on various days to maturity of the options are presented in Table 2. Mean absolute errors and root mean squared errors of network implied volatilities compared to option-model implied volatilities are also presented.

ANALYSIS OF RESULTS

Table 1 shows the means and standard deviations of market call prices and network implied call prices on the nine selected days to maturity. The table shows the results of t-tests for differences in the means. The test is based on assumption of two independent samples when sigmas are unknown (paired test). Network implied call prices are in all cases lower than market call prices. But there is no significant difference in the two types of prices for seven of the nine maturity classes: 57, 56, 55, 37, 36, 35 and 17 days to maturity. The differences are significant for the 16-day maturity class (p-value = 0.07) and the 15-day maturity class (p value = 0.00). That implies, when it comes to shorter horizons, the network fails to generate output close to market call price. It may be that the market gets choppier as future options contracts approach maturity which creates noise that the network is not able to filter out. It may be because the network does not have enough information to generate efficient output. Recall that the input variables based on which the network learns to produce efficient output do not include futures price volatility. We intend to add volatility to the list of inputs and then train the network to generate call price. But given that the network can generate call prices not significantly different from market call prices without the advantage of volatility data is a testament to efficiency of neural networks.

The results of tests of differences in the means of option model implied volatilities and neural network implied volatilities for selected days to maturities are given in Table 2. The outputs from NN this time for all maturity classes are not significantly different from Black model implied volatilities. Test values are the highest for the 37-day maturity class: p-value = 0.18, mean absolute error = 0.0017, and root mean squared error = 0.0046. Network implied volatilities have higher means than Black model implied volatilities for all 9 maturity classes. This is consistent with outcomes for call prices from the network. The means of call prices are lower than market call prices for all maturity classes.

CONCLUSION

Neural networks have proved to be promising in various applications in finance. Results from this study show that the networks can have promising application in option pricing – in finding option price as well as implied volatility.

Notes

[1]This issue has been addressed by Hutchinson et al. In a 1995 paper. This author came up with this issue in 1993 during doctoral dissertation work using neural networks to forecast S&P 500 Index futures price volatility. Hutchinson et al. (1995) use simulated data. This study uses transactions data covering 12 1/4 years.

[2] See for example, Lajbcygier and Connor (1997), Geigle and Aronson (1999), Hanke (1999), Zapart (2003).

[3] See Hamid (2004) for a primer on NN and a list of many unresolved issues in which NNs can be applied.

[4]Function with S-shaped curve; the standard sigmoid function (Y) has the form: $Y = \{1/[1 + \exp(-x)]\}$.

[5]In one variation, the tails of the sigmoid function less rapidly approach 0 and 1 than the standard sigmoid. In another, the function roughly falls in between the first two.

[6]A hyperbolic tangent is S-shaped like a sigmoid, but is extended to the left and to the right.

[7]A Gaussian transfer function looks like a bell-shaped curve which is normally distributed. Under this function, maximal output from a neuron is achieved with low levels of inhibition or activation, and higher levels of activation or inhibition suppress the neuron's response.

[8]A function whose derivative exists at every point is continuous.

[9]Contract class that will mature in the nearest maturity month. The maturity months are March, June, September and December.

[10] Researchers find that the Black model implied volatilities from just-out-of-the-money options produce reasonably efficient forecasts of volatility given its parsimonious data needs. Feinstein (1989a) shows that only implied volatilities from at-and-near-the-money options should be unbiased estimates of rational forecasts of future volatility. It has been found that at-the-money options are almost linear in sigma at all maturities (so that there is no Jensen's inequality type effects), and the implied volatilities derived from inverting a volatility model that assumes nonstochasticity should accurately reflect average volatility expected over the remaining life of such options. In the next part, we use implied volatilities from Black model using just-out-of-the-money options to compare against volatility from neural network. So, here we use just-out-of-the-money options.

Table 1

Results of tests of differences in the means of market call prices and neural
network implied call prices for options on S&P 500 Index futures
(nearest contract from March 1983 to June 1993) for various
days to maturity of options.

Days to maturity	Means of		Std. Dev. of		t-test	
	Market Call Price	Neural Implied Call Price	Market Call Price	Neural Implied Call Price	t-stat.	p-value
57	7.71	7.54	3.15	3.05	0.27	0.39
56	7.83	7.65	3.17	3.06	0.27	0.40
55	7.94	7.73	3.06	2.96	0.34	0.37
37	6.54	6.43	2.81	3.16	0.18	0.43
36	6.67	6.35	3.78	3.20	0.45	0.33
35	6.30	6.13	3.24	3.03	0.26	0.40
17	4.50	3.94	2.16	3.40	0.95	0.17
16	4.45	3.78	1.94	2.49	1.48*	0.07
15	4.47	3.45	2.20	1.60	2.57*	0.00

Note: Differences are significant at 7% and 0% levels, respectively.

Table 2

Results of tests of differences in the means of option model implied volatilities (OIV) and neutral network implied volatilities (NIV) of options on S&P 500 Index futures on various days to maturities; no significant differences are found; also, mean absolute errors (MAE) and root mean squared errors (RMSE) of NIV compared to OIV are shown.

Days to maturity	Means of		Std. Dev. of		t-test		MAE	RMSE
	OIV	NIV	OIV	NIV	t-stat.	p-value		
57	0.0093	0.0094	0.0021	0.0029	-0.23	0.41	0.00126	0.00183
56	0.0095	0.0096	0.0026	0.0028	-0.08	0.47	0.00109	0.00198
55	0.0093	0.0096	0.0024	0.0027	-0.43	0.33	0.00087	0.00121
37	0.0094	0.0102	0.0042	0.0056	-0.91	0.18	0.00177	0.00466
36	0.0099	0.1019	0.0044	0.0053	-0.27	0.40	0.00105	0.00150
35	0.0095	0.0100	0.0036	0.0046	-0.59	0.28	0.00116	0.00171
17	0.0088	0.0091	0.0019	0.0029	-0.72	0.24	0.00130	0.00207
16	0.0088	0.0091	0.0020	0.0028	-0.46	0.32	0.00098	0.00146
15	0.0090	0.0091	0.0037	0.0030	-0.15	0.44	0.00108	0.00222

References

Bachelier , L. (1900) Theorie da la speculation." Annales de l'Ecole Normale Superieure. Trans. By A. J. Boness in The Random Chracter of Stock Market Prices, ed. P.H. Cootner, Cambridge, Mass.: MIT Press, 1967.

Black, F., and M. S. Scholes (1973) The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81, (May-June), 637-654.

Black, F. (1976) The Pricing of Commodity Contracts. *Journal of Financial Economics*, 3, (March), 167-179.

Feinstein, S. (1989a) The Black-Scholes Formula is Nearly Linear for At-The-Money Options; Therefore Implied Volatilities From At-The-Money Options are Virtually Unbiased. Unpublished Working Paper, (December).

Feinstein, S. (1989b) Forecasting Stock-Market Volatility Using Option Index Futures. *Economic Review*, Federal Reserve Bank of Atlanta, (May/June), 12-31.

Geigle, D.S. and J.E. Aronson (1999) An artificial neural network approach to the valuation of options and forecasting of volatility. *Journal of Computational Intelligence in Finance* 7:6 (November/December).

Hamid, S. A. (2004) Primer on Using Neural Networks for Forecasting Market Variables. Working Paper No. 2004-03, The Center for Financial Studies, Southern New Hampshire University, www.snhu.edu/cfs.

Hanke, M. (1999) Option pricing using neural networks vs. Black/Scholes: An empirical comparison of two fundamentally different option pricing methods. *Journal of Computational Intelligence in Finance* 5:1 (January), 26-34.

Hutchinson, J. M., A. Lo, and T. Poggio (1995). A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks. *Journal of Finance*, 49, 851-889.

Kawaller, I. G., P. D. Koch, and T.W. Koch (1988) The Relationship Between the S&P 500 Index and S&P 500 Index Futures Prices. *Federal Reserve Bank of Atlanta Economic Review*, 73, (May-June), 2-10.

Lajbcygier, P.R. and Connor, J.T. (1997) Improved option pricing using artificial neural networks and bootstrap methods. *International Journal of Neural Systems* 8:4, 457-471.

Merton, R. C. (1973) Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science*, 4, (Spring), 141-183.

Randolph, W. L., B. L. Rubin and E. M. Cross (1990) The Response of Implied Standard Deviations to Changing Market Conditions. Advances in Futures and Options Research, JAI Press Inc., Vol. 4, 265-280.

Zapart, C.A. (2003) Beyond Black-Scholes: A neural networks –based approach to options pricing. *International Journal of Theoretical and Applied Finance* 6:5, 469-489.