A NOVEL METHOD FOR ESTIMATING THE INVERSE FUNCTION OF BLACK-SCHOLES OPTION PRICING MODEL USING ARTIFICIAL NEURAL NETWORKS

Authors
Hamed Shafiee Hasanabadi, Rene V. Mayorga

Faculty of Engineering and Applied Science

University of Regina

May 15, 2014

1. ABSTRACT

Black-Scholes (BS) model is a well-known model for pricing options. Option is a derivative financial instrument which gives its owner the right of buying the underlying asset at a pre specified date for a pre specified price. BS model calculates the option price using 5 input variables and parameters including current underlying price, strike price, time to maturity, interest rate and the volatility of the underlying asset price. Having a good estimation of volatility as a risk measure is also very important to calculate a fair option price. Simulating direct function of BS model with Artificial Neural Networks having 5 pre mentioned inputs is straight forward. On the other hand, simulating the inverse function of BS model to have an estimation of the comprising variables such as implied volatility based on the option price is tricky. ANNs cannot propagate error appropriately where the number of the outputs is more than the number of the inputs, so the network doesn't work well. Because of this fact, simulating the inverse function of BS model using the ANNs is not efficient. In order to improve the simulation power of ANNs for simulating the functions which have more output variables than input variables, a new method is introduced in this study. This method improves ANNs performance in estimating inverse function of BS model. The results prove that the suggested method outperforms the current simulation method. Both Feed Forward Back Propagation Network and Radial Basis Function Network which have been used in this study show good ability in this regard. Radial Basis Function Network performs better than the Feed Forward Back Propagation Network.

2. Introduction

There is a growing interest in the application of Artificial Neural Networks (ANNs) in different scopes of economic and finance area. Recognizing the internal patterns of financial models or data series, forecasting the future financial values, and financial models simulations are the most well known ANNs applications in this area (Hutchinson et al., 1994; Donaldson and Kamstra 1996a,b ; Zhang and Berardi, 2001; Kanas, 2001; Kanas and Yannopoulos1, 2001; Maasoumi and Racine,2002; Shively, 2003; Jasic and Wood, 2004; Rapach and Wohar, 2005; Shachmurove, 2005; Wang, 2009; Yu et al., 2009, Shafiee et al., 2012). Artificial Neural Networks (ANNs) can be used for modeling the non-linear relationships between some inputs and some outputs of a model. They are kind of an information processing technology which has been developed based on the construction of the human brain's neurons. They are organized in some layers that each layer includes a set of interconnected processing elements (nodes). They work together parallels to transform the inputs to the outputs of the model (Malliaris & Salchenberger, 1996). In case of complex system with complex relationships, which cannot be modeled using other models such as linear models, ANNs are appropriate (Hamid & Iqbal, 2004).

A call option is a derivative financial instrument. It gives its owner the right of buying the underlying asset at a pre specified date for a pre specified price. If option can be exercised just at a pre specified time, it called European call option. There are some models for pricing European call options. One of these models is Black-Scholes model. The Black-Sholes model is the most renowned model for pricing options (Black and Scholes, 1973). According to this model there are 5 factors which affect the option price including current underlying asset price, the strike price, the time to expiration, the volatility of the underlying asset's price (standard deviation of its expected returns) and the risk-free interest rate.

Pricing the derivatives, especially options has been a field of study in the ANNs applications in recent years (Wang, 2009; Ko, 2009; Andreou et al., 2008; Morelli et al., 2004; Xua et al., 2004; Bennell and Sutcliffe, 2004; Montagna et al., 2003; Yao et al., 2000; Garcia and Gencay, 2000; Geigle and Aronson, 1999; Malliaris and Salchenberger, 1996,1993; Hutchinson et al., 1994). In these studies, different models have been introduced for the option valuation using the Artificial Neural Networks. The results of these studies show that the Artificial Neural Networks outperform the other valuation models such as Black-Scholes model, regression models or path integral approach and etc (Montagna et al 2003). In the study by the Amilon, it is proved that the Artificial Neural Network outperforms the Black-Scholes model in pricing the daily Swedish stock index call options because of questionable assumptions of the Black-Scholes model (Amilon, 2003). Saxena showed that a hybrid model of ANNs outperforms the Black-Scholes model in pricing the options traded at National Stock Exchange of India Ltd (Saxena, 2008). What makes the Artificial Neural Network an appropriate tool for simulating some models such as the option pricing model is its ability in simulating non-linear relationships among the data without being restricted on the assumptions that the parametric approaches such as Black-Scholes model rely on them(Bennell and Sutcliffe, 2004).

In all of the mentioned studies, the option price as the output of an option pricing model is studied but none of them work on the inverse functions of the option pricing models. Sometimes, comprising parameters of the option pricing model including volatility should be estimated. Volatility, as an estimation of the market risk is an important input to many financial models. Volatility which is estimated according to the inverse function of Black-Scholes model is known as the implied volatility. Implied volatility is the value which the market gives to the volatility or the market's estimation of volatility for the life period of the option to specify price of the option. Given the price of a traded option, it is possible to determine the volatility forecast over the lifetime of the option implied by the option's price based on the Black-Scholes model using analytical or numerical methods (Li, 2008). It is also possible to do this using ANNs as well. Estimating the implied volatility solely having the other five variables of the Black-Scholes model using ANNs is possible even simple. Having a precise estimation of more variables which comprise the option price is not a simple job. The focus of this study is to improve the simulation power of the inverse function of Black-Scholes model using Artificial Neural Networks (ANNs) for estimating input variables of Black-Scholes model such as implied volatility, time to maturity, moneyness and interest rate based on the option price.

Simulating the direct model of Black-Scholes using ANNs is simple and straight forward because the system has 5 inputs and one output. ANNs are very powerful in function approximation problem when the number of input variables is more than the number of output variables, but for the inverse function where the number of the inputs is less than the number of the outputs the simulation error of ANNs increases. Eventually, the regular method of using ANNs cannot simulate these type of functions properly. One reason is that there are various combination of values for the input variables which generates the same value for a special set of the output variable. When the Artificial Neural Network is going to simulate the inverse function, there are different combinations of values for the output variables which are related to that special value of the input variable to the inverse function. In conclusion, the Artificial Neural Network has numerous set of related values for a special input set so it may generate the output which is very noisy comparing to the desired output. Simulating the inverse function of

Black-Scholes model enables us to have an estimation of current underlying price, strike price, time to maturity, interest rate and implied volatility based mainly on the option price.

In this study, the literature of the inverse functions estimation using the ANNs is extended. Feed Forward Back Propagation Network and Radial Basis Function Network are employed to model the inverse function of the Black-Scholes option pricing model.

In order to measure the performance of the suggested method for improving the simulating power of the inverse function of the Black-Scholes model using the ANNs, the system is simulated with two methods, one time with the current method of applying ANNs on the inverse function and one time using the suggested method. Then the results are compared with each other. shortly, the inverse function of the Black-Scholes option pricing model was simulated one time with regular ANN model and then with the suggested method.

In addition to compare the performance of regular ANN model with the suggested method, the performance of different classes of Artificial Neural Networks including Feed Forward Back Propagation Network and Radial Basis Function Network have been compared.

The rest of this study is organized as followings. At the Section 6.2, the data of the study which will be used for the simulations will be presented and analyzed. Section 6.3, is going to explain the Black-Scholes model briefly. Section 6.4, discusses the methodology including a detailed explanation about how to develop the inverse model. It also introduces the suggested method. The evaluation measures are discussed in section 6.5. Section 6.6, provides the inverse models estimations and the out-of sample simulation results. It also discusses the results and compares the models based on the results captured in previous section. The section 6.7, as the final section of this chapter provides the conclusion of this study.

3. Data

The simulation data for testing and training the models are generated randomly. In order to provide the option pricing data for the models, random numbers are generated for input variables of the models according to their logical technical ranges and specifications of each variable as followings.

1-The moneyness, ($^{X}/_{S_0}$ Strike price/Current underlying asset price), m ϵ [1 1.2]

2-The volatility of the stock price, σ ϵ [20%   40%]

3-The risk-free interest rate, r ϵ [1%   10%]

4-The time to expiration, T ϵ [1 6] month

A maximum value of 1.2 is considered for the upper limit of moneyness and the current price is considered equal to 1 for all the data points as explained before. The strike price will be determined using two values, moneyness and current price. Normally, the range of volatility is between 20% and 40% for the normal stocks, so this range is considered for generating random values for this variable (Hull, 2002). The rate of interest is generated in a range of 1% to 10%. The maturity time is generated in a range of 1 to 6 month.

Random values are generated in their appropriate ranges using the uniform random generator function. The Call option price, C, will be calculated analytically using Black-Scholes option pricing model and generated data using equations 6.2, 6.3 and 6.4 which are based on the Black-Scholes formula. 1000 series of data are generated for each of the training and testing data sets.

4. Black-Scholes option pricing Model

The Black-Scholes (BS) model is one of the most well-known option pricing models in finance. Black, Scholes, and Merton introduced the BS model for pricing the options in 1970 and the model absorbed lots of attentions in the field of option pricing. An option is a derivative financial instrument which can be considered as a contract between two parties to sell or buy an asset (underlying asset) at a pre specified price during a specified period of time or at a pre specified time. A call option is an instrument which offers the right of trading an asset, buying, or selling that, at a pre specified time for a pre specified price. The trading price is called strike price and the trading time is called as the expiration date, the exercise date, or the maturity. European options are options which have restricted time to maturity. It means that they can be exercised at some points of time only (Black and Scholes, 1973).

If the expiration time is shown with T and the current time with t where $t < T$, the current price of the underlying asset, $S(t)$, is known, but the price of the underlying asset at maturity of the option, $S(T)$, is not known. According to the Black and Scholes, $S(T)$ can be considered as a random variable and using some mathematical model, the dynamics of the price $S(t)$ can be modeled as a function of time; Let $f(S(t), t)$ be the price of the option at time t, or in short $f(S, t)$. Under suitable assumptions, the value of the option can be formulated as Eq. 6.1(Black and Scholes, 1973; Wilmott, 2007; Hull, 2002):

$$\frac{\partial f}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} + rS\frac{\partial f}{\partial S} - rf = 0 \qquad \text{(Eq. 6.1)}$$

The current value f (S_0, 0), where S_0= S(t=0), is the option price. The equation can be simplified and solved analytically if the distribution of the underlying asset assumed to be normal distribution. Assuming this, the European call option price can be calculated based on the Black-

Scholes formula according to equation 6.2, 6.3, and 6.4(Black and Scholes, 1973; Wilmott, 2007; Hull, 2002):

$$c = S_0N(d_1) - Xe^{-rT}N(d_2) \quad \text{(Eq. 6.2)}$$

Where

$$d_1 = \frac{\ln\left(S_0/X\right)+\left(r+\sigma^2/2\right)T}{\sigma\sqrt{T}} \quad \text{(Eq. 6.3)}$$

$$d_2 = \frac{\ln\left(S_0/X\right)+\left(r-\sigma^2/2\right)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \quad \text{(Eq. 6.4)}$$

N(x) is the cumulative of the probability function. This function is the standardized normal variable. In Other words, if the variable has the standard normal distribution, N(x) is the probability which the variable has the values less than the value of x. The variable C is the European call option price, X is the strike price, T is the time to expiration, σ is the volatility of the underlying asset price and r is the risk-free interest rate (Black and Scholes, 1973; Wilmott, 2007; Hull, 2002).

Based on the BS model, five factors which affect option price are the current underlying asset price, strike price, the time to expiration, the volatility of the underlying asset price and the risk-free interest rate. In this study, the inverse function of the Black-Scholes model is going to be simulated to estimate current underlying asset price, S, strike price, X, the time to expiration or maturity, T, the volatility of the underlying asset price, σ, and the risk-free interest rate, r mainly based on the European Call Option Price, C, using ANNs. The model has 5 independent variables and one dependent variable so the data sets should include these six variables and they will be fed into the networks according to each network's design. In order to make the model independent of any underlying asset, the moneyness which is defined as the ratio of Strike price to the current price of the underlying asset (Equation 6.5) has been used. This ratio makes the model free of any specific asset and makes the model general because it uses the ratio of strike price to the current price which is independent of any specific price range of the underlying assets (Merton, 1973, Black and Scholes, 1973; Wilmott, 2007; Hull, 2002).

$$m = {}^{X}/_{S_0} \quad \text{Eq. 6.5}$$

In some other studies, it has been shown that this variable has all the information which is in both underlying asset price and strike price. In one study, Anders et al. used both moneyness (${}^{X}/_{S_0}$) and current underlying price (s_0) as the inputs to his Artificial Neural Network. He found out that dropping (s_0) doesn't change the results (Anders et al., 1998). In another study, Garcia and Gencay showed that the Artificial Neural Networks which have moneyness (${}^{X}/_{S_0}$) as their input are superior to those which have current underlying price (s_0) and strike price (x) separately (Garcia and Gençay, 1998, 2000). According to these findings, the moneyness is used instead of current price of the underlying asset and Strike price.

5. Developing inverse models

The forward function of the BS model uses the current underlying asset price, Strike price, volatility of the underlying asset price, rate of interest and time to maturity as its inputs to calculate option price as its output. The strike price and underlying asset price are replaced with moneyness. The schematic view of the BS direct function is as figure 6.1. Suppose a system which has n input variables and m output variables as figure 6.2.

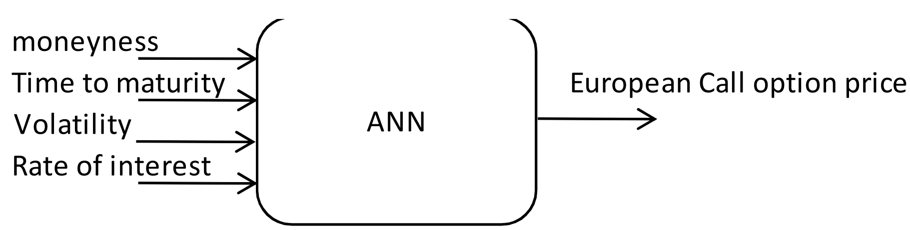

Figure 5.1: schematic view of the BS direct function

Figure 5.2: a system with n input variables and m output variables

Each of the output variables can be a function of input variables of the system. The system equation can be written as equation 6.6 as the following system of equations:

$$y_1 = f_1(x_1, x_2, \dots, x_n)$$

$$y_2 = f_2(x_1, x_2, \dots, x_n)$$

$$\dots$$

$$y_m = f_m(x_1, x_2, \dots, x_n)$$

$$x \in \mathbb{R}^n \quad y \in \mathbb{R}^n$$

Eq.6.6

If this system is considered as a Artificial Neural Network, there can be infinite solutions for the system when $m < n$. If $= n$, there can be just one solution for the system. If $> n$, there may exist one approximate solution for the system. In other words, when $m < n$, the direct function has many solutions, so the ANN cannot properly find an exact solution for the inverse function. Since the direct function has many solutions for each case, the inverse function should approximate one solution among all the possible combinations of the direct function solutions. In this case, the ANN probably can find an approximate solution for the inverse function of the system. In the case of BS model, According to figure 6.1, the direct function of the Black-

Scholes model can be considered as the case where $m < n$, so the inverse function is the system where $m > n$ as figure 6.3.

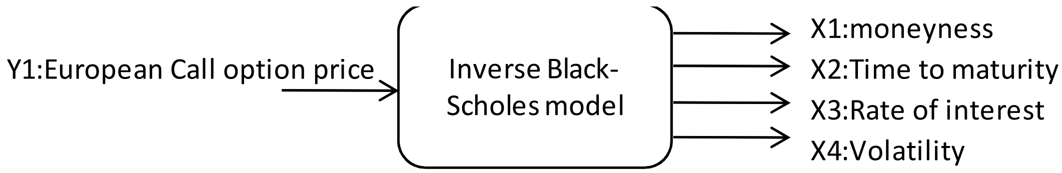

Figure 5.3: inverse function of BS model

As mentioned before, in this case, the ANN can find an approximate solution for the inverse function. One reason is that there are various combination of values for the input variables which gives the same value for a special set of the output variable. In conclusion, when the Artificial Neural Network is going to simulate the inverse function, there are different combinations of values for the output variables which are related to that special value of the input variable set to the inverse function so the Artificial Neural Network has numerous set of related values for a special input values so it may generate the output which is very noisy comparing to the desired output. In order to have a better solution, the system needs to be changed to situation where $m = n$. In order to match this, the suggested method in this paper is to add some dummy variables to the system to equalize the number of the inputs and outputs. In this manner, as there is a solution for the system, the ANN can work better. Actually the set of "Y"s will be completed with the virtual dummy functions as :

$$y_{m+1}, y_{m+2}, \cdots, y_n$$

So that i sentences, or i functions should be added to the current m functions as:

$$m + i : i = 1, \ldots, N \qquad where: \ N = n - m$$

So the system will be as figure 6.4:

Figure 5.4: System with n input and n output

The final set of formula for system will be as equation 6.7.

$$
\begin{gathered}
y_1 = f_1(x_1, x_2, \dots, x_n) \\
y_2 = f_2(x_1, x_2, \dots, x_n) \\
\cdots \\
y_m = f_m(x_1, x_2, \dots, x_n) \\
\text{Set of dummy functions} \begin{cases} y_{m+1} = f_{m+1}(x_1, x_2, \dots, x_n) \\ \dots \\ y_{m+i} = f_{m+i}(x_1, x_2, \dots, x_n) \\ \dots \\ y_{n=m+N} = f_n(x_1, x_2, \dots, x_n) \end{cases} \\
\cdots \\
as\ i = 1, \dots, N \qquad N = n - m \\
x \in \mathbb{R}^n \ \ y \in \mathbb{R}^n
\end{gathered}
$$

Eq.6.7

As described before, the inverse function of Black-Scholes model is going to be simulated. The Call option price is considered as the input, and the other 4 variables are considered as the outputs of the network (As explained before, two variables, Current underlying asset price and strike price are combined to form one variable as the moneyness). The final network which simulates inverse if BS model will have one input and four outputs as figure 6.3. The number of the outputs is more than the number of the inputs for this system, so the ANN can find an approximate solution. In order to overstate this and according to the suggested method, the set of "Y"s of the direct function will be completed with dummy variables (functions). Eventually, the number of the inputs and outputs would be the same. Three dummy variables are considered as the inputs of the inverse function (output set of the direct function) as figure 6.5.

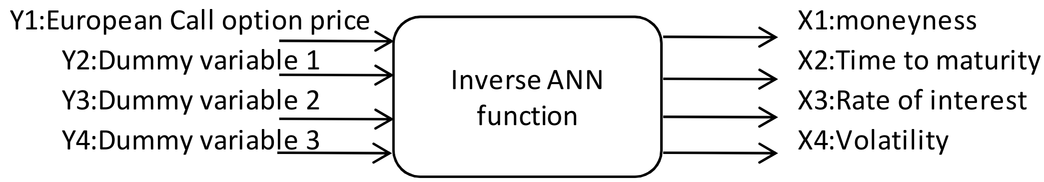

Figure 5.5: System with dummy variables as the inputs of the inverse function

Shortly, the model has one input and four outputs, so three dummy variables should be added to system to equalize the number of the inputs and outputs of the inverse function. Based on the system of equation 6.7, these dummy functions are functions of the input variables of the direct function. These functions can be defined according to the rationale of the field which the system is defined on that. Each of these dummy variables can be a function of the input variables of the direct model (output variables of the inverse model) or a function of a combination of these variables (Eq.6.7) including moneyness, time to maturity, volatility and interest rate.

In the study, the basic inverse function of BS model (without dummy variables) is considered as the base model. In other words, two different Artificial Neural Networks will be designed, one simulating the basic inverse function of BS model as figure 6.3 as the base model and one which simulates the inverse function of BS model considering the added dummy variables as figure 6.5.

The base model has the European call option price as its input and the other 4 variables including moneyness, time to maturity, rate of interest and implied volatility as its outputs. The suggested model has 3 dummy variables as explained before in addition to call option price as its inputs. In this study, two different sets of dummy variables are tested. For the first try, dummy variables are considered as functions of the 3 of the 4 output variables. The three dummy functions which are defined in this step are as (1: squared of the estimation of moneyness), (2: the complement of the estimation of the volatility (1 - the estimation of the volatility)) and (3: squared of the estimation of the time to maturity).

For the second try, dummy variables are considered as functions of the 2 of the 4 output variables. The three dummy functions which are defined in this step are as (1: the squared root of the estimation of moneyness), (2: the complement of the estimation of the volatility (1 - the estimation of the volatility)) and (3: the estimation of moneyness to the power of 3).

Since both Feed Forward Back Propagation Network and Radial Basis Function Network are going to be used is this study, an appropriate design is used for each of the two pre-mentioned models for each of the two types of networks. It means that four ANNs, two FFBPNs and two RBFNs for simulating the base model and suggested model for each set of the dummy variables, are designed and tested.

According to the provided general overview of the simulation process, the input variables of the ANNs should be determined. The input data are preprocessed to be in range [0,1] because ANNs perform better when the inputs are in this range.

According to different architecture of Radial Basis Functions Networks and Feed Forward Back Propagation Networks, their designs will be different. The number of the neurons in the input layer assumed equal to the number of the input variables for the first run. According to the ANNs design, the number of the neurons in the last layer or output layer should be equal to the number of the output variables as well, so they are considered equal to the number of the output variables which in this model is 4 for both networks. A 3 layered network is considered for the initial design of the Forward Back Propagation Network. The number of the layers will be changed in order to reach the best design to get the best results. The number of the neurons in the first layer, input layer, and the number of the hidden layers and their assigned neurons will be changed during the training process as the parameters of the ANNs to improve the results. The range of the output variable of the study is more than one, so “pure linear” function is considered as the transfer function for the output layer. “Logarithmic sigmoid” transfer function is also considered for the input and other hidden layers for Feed Forward Back Propagation Network. Training algorithm is set as the Levenberg-Marquardt algorithm, and a goal of 1e-15 is set as the network goal for the mean squared error performance function.

In contrary with the Feed Forward Back Propagation Network, the Radial Basis Function Network has just 3 layers, input layer, radial basis layer and output layer. The number of the neurons in input layer is equal to the number of the input variables and the number of the neurons in the output layer is equal to the number of the output variables. The number of the neurons in the radial basis layer is also determined automatically during the training process, so it does not need to be determined.

Again, because the range of the output variable of the study is more than one, “pure linear” function is considered as the transfer function for the output layer. “radial Basis function” transfer function is also the pre specified transfer function for the radial basis layer in Radial Basis Function Networks. Spread is another parameter of the RBFN. The spread should be large enough to to allow the active input regions of the radial basis neurons overlap enough. The Training algorithm is set as the Levenberg-Marquardt algorithm, and a goal of 1e-5 is set as the network goal for the mean squared error performance function.

After finishing the networks designs, the training data can be fed into the neural networks. 1000 data points are generated randomly for each of the training and testing sets.

Training the networks is started by applying different combinations of data sets as the inputs to the networks. After each run, the network compares the forecasted moneyness, interest rate, volatility, and time to maturity with the desired ones. It calculates and feeds the error backward. The neurons reset their weights each time the errors are fed back. After training each network, the network parameters including the number of the hidden layers and neurons in each hidden layer are changed based on the performance of the training. The transfer functions of Feed Forward Back Propagation Network and the value of the spread of the Radial Basis Function Network are changed to reach to the optimal level.

6. Results evaluation

Since there are different values of the moneyness, interest rate, volatility and time to maturity which are related to each value of European call option price, the forecasted values of the networks cannot be compared with the appropriate set of the presented data (initial ones) which have been applied to the network. Various sets of the forecasted values could generate the same option price, so this point should be considered in results evaluation. In order to deplete this problem and have a good evaluation of the precision of the results, the option price is calculated based on the forecasted values of the network (moneyness, interest rate, volatility and time to maturity) using the Black-Scholes formula analytically or using an ANN which is trained based on option pricing data which are generated using BS formula. This ANN simulates the BS formula. The calculated option price will be compared with the initial option price. The calculated error will be considered as the measure of the performance of the networks.

Actually, the simulation and testing process has been done in two manners. One time, inorder two generate the option prices, for situation where there exist the explicit pricing model, BS formula has been used. In another run, the simulation process has been done using the simulated BS formula with ANNs, for modeling the situation where the explicit pricing model does not exist. To explain more, it can be said that the training and testing data series for the first run is done using the BS formula. In this case, in order to check the results from the inverse function, the model outputs are fed into the BS formula to calculate the option price and this price is compared with the network outputs to determine the model presision.

The training and testing data series for the second run is done using the simulated BS formula with ANNs. In this case, in order to check the results from the inverse function, the model outputs are fed into into the simulated option pricing ANNs, which simulates the BS formula, to calculate the option prices and these prices are compared with the network outputs to determine the model presision again. This additional testing is done to show that where there is not an explicit function, the direct function can just be simulated using ANNs and these ANNs can be used for testing the inverse function.

Results evaluation is done based on three measures including the mean absolute error (MAE) which is calculated according to equation 4.12; the mean squared error (MSE) which is calculated according to equation 4.13; and the mean absolute percentage error (MAPE) which is calculated according to equation 4.14. Generally, these measures calculate the difference between the ANNs results and actual ones, which have been calculated analytically based on the BS model.

7. Empirical results and discussion

Different designed networks with different justifications such as hidden layers, neurons in each layer, transfer functions, and spreads have been trained. After training and testing the networks, the combination that provided the best results were chosen and are presented in tables 6.1, 6.2, 6.3, and 6.4.

The result for the situation where the simulation and evaluation is done using BS formula is according to tables 6.1 and 6.2.

The result for the situation where the simulation and evaluation is done using the simulated BS formula with the ANNs, is according to tables 6.3 and 6.4.

For the first set of dummy variables, the network architectures for FFBPN for the inverse base case is a 3 layered network with the 1-5-4 architecture and for the inverse dummy added case is a 5 layered network with the 4-5-6-5-4 architecture.

For the second set of dummy variables, the network architectures for FFBPN for the inverse base case is a 3 layered network with the 1-5-4 architecture and for the inverse dummy added case is a 5 layered network with the 4-5-6-5-4 architecture.

For both types of Artificial Neural Networks and for both types of evaluation processes the results has been brought in Tables 6.1, 6.2, 6.3, and 6.4.

Table 7.1: Artificial Neural Networks training and testing results for first dummy variables set-simulation and evaluation using BS formula

System Type	Network Type		'MAPE' %	'MSE'	'MAE'
System with regular inputs	FFBPN	Training	9.05	2.57E-06	1.11E-03
		Testing	10.64	2.11E-06	1.04E-03
	RBFN	Training	13.42	1.74E-06	9.12E-04
		Testing	8.51	4.07E-06	1.05E-03
System with added dummy variables(function)	FFBPN	Training	3.42	7.61E-07	6.29E-04
		Testing	3.35	7.27E-07	6.38E-04
	RBFN	Training	0.71	4.42E-08	1.50E-04
		Testing	0.95	1.75E-07	2.10E-04

Table 7.2: Artificial Neural Networks training and testing results for second dummy variables set-simulation and evaluation using BS formula

System Type	Network Type		'MAPE' %	'MSE'	'MAE'
System with regular inputs	FFBPN	Training	9.05	2.57E-06	1.11E-03
		Testing	10.64	2.11E-06	1.04E-03
	RBFN	Training	13.42	1.74E-06	9.12E-04
		Testing	8.51	4.07E-06	1.05E-03
System with added dummy variables(function)	FFBPN	Training	3.61	1.05E-06	6.53E-04
		Testing	3.60	1.28E-06	7.26E-04
	RBFN	Training	3.40	3.50E-07	4.47E-04
		Testing	3.36	5.53E-07	4.99E-04

Table 7.3: Artificial Neural Networks training and testing results for first dummy variables set-simulation and evaluation using simulated BS formula with the ANNs

System Type	Network Type		'MAPE' %	'MSE'	'MAE'
System with regular inputs	FFBPN	Training	14.45	2.49E-06	1.18E-03
		Testing	12.46	3.42E-06	1.26E-03
	RBFN	Training	11.90	1.63E-06	9.28E-04
		Testing	10.88	4.03E-06	1.05E-03
System with added dummy variables(function)-simulated BS model with ANN	FFBPN	Training	5.64	9.78E-07	6.27E-04
		Testing	3.51	1.30E-06	6.61E-04
	RBFN	Training	4.64	2.00E-07	3.24E-04
		Testing	2.59	5.51E-07	4.95E-04

Table 7.4: Artificial Neural Networks training and testing results for second dummy variables set-simulation and evaluation using simulated BS formula with the ANNs

System Type	Network Type		'MAPE' %	'MSE'	'MAE'
System with regular inputs	FFBPN	Training	14.45	2.49E-06	1.18E-03
		Testing	12.46	3.42E-06	1.26E-03
	RBFN	Training	11.90	1.63E-06	9.28E-04
		Testing	10.88	4.03E-06	1.05E-03
System with added dummy variables(function)	FFBPN	Training	9.11	2.21E-06	1.08E-03
		Testing	4.86	2.50E-06	1.13E-03
	RBFN	Training	8.87	5.10E-07	5.26E-04
		Testing	4.16	1.29E-06	7.24E-04

Table 6.5 and 6.6 compares the testing results from the two types of Artificial Neural Networks for both of the systems (the regular system which has the regular inputs and the system with the added dummy variables among the inputs), for both sets of dummy variables for both evaluation processes. It can be seen that both Artificial Neural Networks including Feed Forward Back Propagation Network and Radial Basis Function Network exhibit a good performance in simulating the inverse function of the Black-Scholes model for both the systems considering both sets of dummy variables and both evaluation methods.

Table 7.5: Final results comparison- simulation and evaluation using BS formula

System Type	Network Type		'MAPE' %	'MSE'	'MAE'
System with regular inputs	FFBPN	Testing	10.64	2.11E-06	1.04E-03
	RBFN	Testing	8.51	4.07E-06	1.05E-03
System with added dummy variable(function) First set	FFBPN	Testing	3.35	7.27E-07	6.38E-04
	RBFN	Testing	0.95	1.75E-07	2.10E-04
System with added dummy variable(function) Second set	FFBPN	Testing	3.60	1.28E-06	7.26E-04
	RBFN	Testing	3.36	5.53E-07	4.99E-04

Table 7.6: Final results comparison-simulation and evaluation using Simulated BS model with ANN

System Type	Network Type		'MAPE' %	'MSE'	'MAE'
System with regular inputs	FFBPN	Testing	12.46	3.42E-06	1.26E-03
	RBFN	Testing	10.88	4.03E-06	1.05E-03
System with added dummy variable(function) First set	FFBPN	Testing	3.51	1.30E-06	6.61E-04
	RBFN	Testing	2.59	5.51E-07	4.95E-04
System with added dummy variable(function) Second set	FFBPN	Testing	4.86	2.50E-06	1.13E-03
	RBFN	Testing	4.16	1.29E-06	7.24E-04

Comparing the testing results from the Artificial Neural Networks with regular inputs with the suggested system which has the first added dummy variable as the input, it can be concluded that the Artificial Neural Networks with the added dummy variables outperform the model which does not have the added dummy variables. For the case of evaluation process with the BS formula, in FFBPN model, the MAPE is equal to 10.64% while the MAPE for the new method is 3.35% and for RBFN the MAPE is equal to 8.51% while the MAPE for the new method is equal to 0.95%. The other evaluation measures such as MAE and MSE suggest the same. For the case of evaluation process with the simulated BS formula with ANN, in FFBPN model, the MAPE is equal to 12.46% while the MAPE for the new method is 3.51% and for RBFN the MAPE is equal to 10.88% while the MAPE for the new method is equal to 2.59%. The other evaluation measures such as MAE and MSE suggest the same.

Comparing the testing results from the Artificial Neural Networks with regular inputs with the suggested system which has the second added dummy variable as the input, it can also be concluded that the Artificial Neural Networks with the added dummy variables outperform the model which does not have the added dummy variables. For the case of evaluation process with the BS formula, in FFBPN model, the MAPE is equal to 10.64% while the MAPE for the new method is 3.60% and for RBFN the MAPE is equal to 8.51% while the MAPE for the new method is equal to 3.36%. The other evaluation measures such as MAE and MSE suggest the same. For the case of evaluation process with the simulated BS formula with ANN, in FFBPN model, the MAPE is equal to 12.46% while the MAPE for the new method is 4.86% and for RBFN the MAPE is equal to 10.88% while the MAPE for the new method is equal to 4.16%. The other evaluation measures such as MAE and MSE suggest the same.

These comparisons show that the new method is working better than the traditional method and suggested method can decrease the simulation error.

Comparing the Feed Forward Back Propagation Network with the Radial Basis Function Network results with each other for the new system, it can be seen that Radial Basis Function Network is doing better than the Feed Forward Back Propagation Network in this regard.

8. Conclusions

In this study, Artificial Neural Networks have been used to simulate the inverse function of the Black-Scholes model which is a model for pricing the options. Feed Forward Back Propagation Network and Radial Basis Function Network were the two different types of Artificial Neural Networks which have been used. The simulated inverse function using the regular method of using the Artificial Neural Networks was compared with the new suggested method of simulating the inverse functions. The results prove the better performance of new suggested method of using Artificial Neural Networks in simulating the inverse function of Black-Scholes model. According to the measures, the Radial Basis Function Network is working better than the Feed Forward Back Propagation Network too. An important contribution of this part of research is suggesting a new method for increasing the precision of the simulation power of the inverse functions where the number of the inputs is less than the number of the outputs in the Artificial Neural Network models.

9. References

1. Davies PC. “Neural network techniques for financial time series analysis”. In: Lederman J, Klein RA, editors. Virtual trading. Chicago, IL: Probus Publishing Co.; 1995. p. 81– 7.
2. Donaldson, R.G. and M. Kamstra, 1996a, A new dividend forecasting procedure that rejects bubbles in asset prices, Review of Financial Studies 9, 333–383.
3. Donaldson, R.G. and M. Kamstra, 1996b, Forecast combining with neural networks, Journal of Forecasting 15, 49–61.
4. F. Black and M. Scholes, “The pricing of options and corporate liabilities,“ Journal of Political Economy, vol. 81, no. 3, pp.637-654, 1973.
5. Geigle DS, Aronson JE. 1999. An artificial neural network approach to the valuation of options and forecasting of volatility. Journal of Computational Intelligence in Finance 7(6): 19–25
6. Guido Montagna, Marco Morellia, Oreste Nicrosinia, Paolo Amatoc, Marco Farina(2003), Pricing derivatives by path integral and neural networks, Physica A 324, 189 – 195
7. H. Amilon, “A neural network versus Black–Scholes: a comparison of pricing and hedging Performances”, Journal of Forecasting, vol. 22, pp. 317-335, 2003.
8. Hamid, S. A., & Iqbal, Z. (2004). Using neural networks for forecasting volatility of S&P 500 Index futures prices. Journal of Business Research, 57, 1116–1125.
9. Haoffi, Z., Guoping, X., Fagting, Y., & Han, Y. (2007). A neural network model based on the multi-stage optimization approach for short-term food price forecasting in China. Expert System with Application, 33, 347–356.
10. Hornik, K., 1989. Multilayer feed-forward networks are universal approximators. Neural Networks 2, 359–366.
11. Hutchinson, J.M., A.W. Lo and T. Poggio, 1994, A nonparametric approach to pricing and hedging derivative securities via learning networks, Journal of Finance 49, 851–889.
12. J. Bennell and C. Sutcliffe, “Black-Sholes versus artificial neural networks in pricing FTSE 100 options,” Intelligent Systems In Accounting, Finance and Management, vol. 12, pp. 243-260, 2004.
13. J. Hull, 2002, FUNDAMENTALS OF FUTURES AND OPTIONS MARKETS
14. Jasic, T., & Wood, D. (2004). The profitability of daily stock market indices trades based on neural network predictions: Case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965–1999. Applied Financial Economics, 14, 285–297.
15. Jingtao Yao, Yili Li, Chew Lim Tan(2000), Option price forecasting using neural networks, Omega 28, 455-466
16. Kalman, B.L., Kwasny, S.C., 1997. High performance training of feed-forward and simple recurrent networks. Neurocomputing 14, 63–83.
17. Kanas, A. (2001). Neural network linear forecasts for stock returns. International Journal of Finance & Economics, 13, 317–354.
18. Kanas, A., & Yannopoulos1, A. (2001). Comparing linear and nonlinear forecasts for stock returns. International Review of Economics & Finance, 10, 383–398.
19. Ko, Po-Chang. (2009). Option valuation based on the neural regression model. Expert System with Application, 36, 464–471.
20. Kohonen, T. (1987). Self-organization and associative memory (2nd ed.). Berlin: Springer-Verlag.
21. L. Xua, M. Dixon, B.A. Eales, F.F. Cai, B.J. Read, J.V. Healy(2004), Barrier option pricing: modeling with neural nets, Physica A 344, 289–293
22. Li, M., (2008). “Approximate inversion of the Black-Scholes formula using rational functions”, European Journal of Operational Research 185 (2008) 743–759
23. Li, J. H., Anthony, N. M., & Porod, W. (1989). Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube. IEEE Transactions on Circuits and Systems, 36(11), 1405–1422.
24. M. Malliaris and L. Salchenberger, “A neural network model for estimating option prices,” Journal of Applied Intelligence, vol. 3, pp. 193-206, 1993.
25. Maasoumi, E., & Racine, J. (2002). Entropy and predictability of stock market returns. Journal of Econometrics, 107, 291–312.
26. Makridakis, S. (1993). Accuracy measures: Theoretical and practical concerns. International Journal of Forecasting, 9, 527–529.

27. Marco J. Morelli, Guido Montagna, Oreste Nicrosini, Michele Treccani, Marco Farina(2004), Paolo Amato, Pricing financial derivatives with neural networks, Physica A 338, 160 – 165
28. Mary Malliaris, Linda Salchenberger,1996, Using neural networks to forecast the S &P 100 implied volatility, Neurocomputing 10 (1996) 183-195
29. P. Wilmott, 2007, "Paul wilmott introduces quantitative finance", second edition
30. Panayiotis C. Andreou, Chris Charalambous, Spiros H. Martzoukos(2008), Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters, European Journal of Operational Research 185, 1415–1433
31. Po-Chang Ko(2009), Option valuation based on the neural regression model, Expert Systems with Applications 36, 464–471
32. Rapach, D. E., & Wohar, M. E. (2005). Valuation ratios and long-horizon stock price predictability. Journal of Applied Econometrics, 20, 327–344.
33. Rene Garcia, Ramazan Gencay(2000), Pricing and hedging derivative securities with neural networks and a homogeneity hint, Journal of Econometrics 94, 93-115
34. Saxena, A., Valuation of S&P CNX Nifty Options: Comparison of Black-Scholes and Hybrid ANN Model, http://www2.sas.com/proceedings/forum2008/162, 2008
35. Shachmurove, Y. (2005). Business applications of emulative neural networks. International Journal of Business, 10, 303–322.
36. Shafiee, H., Mayorga, R.V., Khan, S., "Forecasting Return Volatility of Crude Oil Futures Prices using Artificial Neural Networks", International Research Journal of Applied Finance, ISSN 2229 – 6891, Vol – III Issue – 9 September, 2012
37. Shively, P. A. (2003). The nonlinear dynamics of stock prices. Quarterly Review of Economics and Finance, 13, 505–517.
38. Tseng, C. H., Cheng, S. T., Wang, Y. H., & Peng, J. T. (2008). Artificial neural network model of the hybrid EGARCH volatility of the Taiwan stock index option prices. Physica A, 387, 3192–3200.
39. Wang, Y. H. (2009). Nonlinear neural network forecasting model for stock index option price: Hybrid GJR-GARCH approach. Expert System with Application, 36, 564–570.
40. Yi-Hsien Wang (2009), Nonlinear neural network forecasting model for stock index option price: Hybrid GJR–GARCH approach, Expert Systems with Applications 36, 564–570
41. Yu, L., Wang, S., & Keung, L. (2009). A neural-network-based nonlinear meta modeling approach to financial time series forecasting. Applied Soft Computing, 9, 536–574.
42. Zhang, G. P., & Berardi, V. L. (2001). Time series forecasting with neural network ensembles: An application for exchange rate prediction. Journal of the Operational Research Society, 52, 652–664.
43. Zhang, G., Patuwo, E.B., Hu, M.Y., 1998. Forecasting with artificial neural networks: the state of the art. Int. J. Forecasting 14, 35–62.