Estimating the Inverse Function of Compound Options 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

Compound options are second order derivatives which give their holders the right for exercising over other derivatives. They are options on options. Compound options have many financial applications. Pricing methods for exotic options such as compounds are much more complex than the regular options. There are different models for pricing compound options. Simulating direct function of compound option pricing model based on the Black-Scholes model needs 7 input variables including current underlying asset price, basic option strike price, the time to expiration of the basic option, the volatility of the underlying asset price, the risk-free interest rate, compound option strike price, and time to expiration of the compound option. In this study, the inverse function of the compound option pricing model is going to be simulated using Artificial Neural Networks (ANNs). The inverse function of the compound option pricing model has 1 input and 7 outputs. This estimation is tricky because of the ANNs weakness in simulating the inverse functions where the number of input variables is less than output variables. ANNs cannot propagate error appropriately when the number of the outputs is more than the number of the inputs, so the simulation power of the network decreases. In this study, the method which has been introduced in previous chapter is going to be applied on the compound option pricing model as well. That method was introduced to improve the simulation power of ANNs for simulating the functions which have more output variables than input variables. Having an ANN with this capability, it could be possible to have a better estimation of the inverse function of compound option pricing model. The results show 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 and the Radial Basis Function Network performs better.
2. **Introduction**

Compound options, as type of exotic options are a complex type of derivatives. Compounds are options on options. A compound option, according to its type, gives its holder the right to buy or sell another option. Compound options are second order derivatives because they give the right for exercising over another derivative. A call option as a derivative financial instrument, gives its owner the right of buying the underlying asset at a pre specified date for a pre specified price. If that option can be exercised at a specified time, it called European call option. Pricing compound options, according to their natures, is more complex than pricing regular options (Geske, 1977, 1979).

There are different methods for valuation compound option prices. First of all, Geske derived the analytical valuation formula for pricing compound options for the Black-Scholes formula (Geske, 1977, 1979). Black-Scholes formula is used to price the options on the basis of the geometric Brownian motion (Black and Scholes, 1973). The number of option layers which reaches thorough the option structure into the underlying asset is called the fold number of a compound option. The basic compound options are 2 folds. The Geske’s proposed method is for 2 fold compound options. Geske assumed that the stock can be interpreted as an option on the firm, so the stock option can be considered as an option on another option, which that option is called compound option. He suggested a formula for pricing compound option using Fourier integrals when the firm value can be modeled as a geometric Brownian motion (Geske, 1979). In that study, he assumed a constant volatility to price a vanilla European call option on a European call option.
Some other valuation methods considering some other assumption have been done by other researchers. Fouque and Han calculate the compound option price using perturbation approximation. In this study, in order to approximate the price of a compound option, they used a combination of singular and regular perturbations techniques. They tried to handle the effect of stochastic volatility in their models (Fouque and Han, 2005). Gukhal proposed a valuation formula for pricing compound options when underlying asset follows a jump-diffusion process. In the proposed model, the parameters such as interest rate and volatility were constant (Gukhal, 2004). In another study by Li et al. Gukhal model has been extended. They model the valuation formula for pricing compound options when underlying asset follows a jump-diffusion process considering the time dependent parameters; because in reality the interest rate and volatility are not constant and they change by time.( Li et al., 2005). In another study Lee et al. introduced a generalized pricing formula for sequential compound options. They provide the deterministic parameters of the model such as interest rate and volatility time-dependent. In this case the sequential compound options can be more flexible comparing to the situation where all the parameters are constant (Lee et al., 2008). In another study, Zhang et al. proposed a formula for pricing compound options under fractional Brownian motion using partial differential equation (Zhang et al., 2011). Griebsch proposed a model for pricing European compound option prices when the volatility is stochastic. She used Fourier transform techniques in her model (Griebsch, 2012). In 1984, Geske and Johnson introduced a formula for modeling the multi-fold compound options (Geske and Johnson, 1984a). Lajeri-Chaherli introduced a formula for pricing 2-fold compound options (Lajeri-Chaherli, 2002), then the results have been extended to be applied for 2 fold compound option models where the parameters are varying with time or time-dependent parameters (Agliardi and Agliardi, 2003; Agliardi and Agliardi, 2005). Some other models have
been introduced for pricing the sequential compound call options (Thomassen and Wouwe, 2001; Chen, 2002; Chen, 2003).

Compound options have many financial applications. Other than their regular usage, pricing compound options, they can be used for pricing some other types of options which their payoffs are functions of European options at some times in the future or have exotic payoffs and financial instruments such as chooser options, capletions and floortions, forward start, Corporate debt, and cliquet options (Chen, 2003; Geske and Johnson, 1984b; Rubinstein, 1992; Musiela and Rutkowski, 1998; Griebsch, 2013). Compound options are used for pricing the American put options (Geske and Johnson, 1984a). One important application of compound options is their application in volatility risk hedging. Traders use compound options as a practical instrument to hedge the volatility risk. The role of volatility risk is crucial in managing a portfolio of derivative securities. This volatility risk is the risk of changing the value of options due to the unpredictable changes in the volatility of the underlying asset. It can be hedged by using future of volatility of volatility swaps (Griebsch, 2013). Compound options can be used as a modeling formula for a sequence of rights or decisions in finance, so any sequence of financial decisions, for instance R&D projects, can be modeled using compound options model. R&D projects can be considered as a sequential decision making process because in each stage which has its own objective and budget, according to the acquired results, you decide to continue the project or not (Gong et al., 2006). They are used for pricing sequential exchange options as well (Carr, 1988). The studies demonstrates some other applications of compound options such as capital budget decision making (Duan et al., 2003), valuating new drugs project(Casimon et al., 2004), inventory and production (Cortazar and Schwartz, 1993).
One method for calculating compound option prices based on its appropriate simulated model (such as Geske model or Black-Scholes model) is using numerical partial differential equations (PDE). Compound options are very sensitive to the values of volatilities. Considering this fact that volatility is not directly observed from the market and the complex behavior of volatility on the other hand, calculating compound option prices using numerical PDE methods can be very hard and complex. They are sometimes time consuming as well. For example, in a two-factor stochastic volatility model case, for pricing the compound option, the full stochastic volatility model should be specified at first, then two iterative three-dimensional PDEs should be solved. The first partial differential equation generates the price of the underlying option and the second one calculates the price of the compound option (Fouque and Han, 2005).

In all of the mentioned studies, the compound option price is studied as the output of a compound option pricing model, but none of them work on the inverse functions of the compound option pricing models. According to the vast and various comprehensive applications of compound pricing models in finance, having estimation on the inverse function of the compound option pricing model can be valuable. In some cases, it is necessary to estimate comprising variables or parameters of the compound option pricing model based on the compound option prices. The inverse function of compound option pricing model is going to be approximated in this study. Artificial Neural Networks (ANNs) are the tools which will be used for this estimation. ANNs enable the model to have fewer assumptions on the pricing model and increase the flexibility of the pricing models. This approach makes it possible to make pricing independent from any specific model.

Pricing the derivatives, especially options has been a field of study in the ANN 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).

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 this study, the method which was introduced in previous chapter is going to be applied to improve the simulation power of ANNs in simulating the inverse function of the compound option pricing model. The compound options pricing model will be developed based on the Black-Scholes model.

Simulating the direct model of compound option pricing model using ANNs is straightforward because the model has 7 inputs and one output and ANNs are very powerful in function approximation problem when the number of input variables is more than the number of output variables. The regular method of using ANNs cannot simulate the inverse function of these types of functions properly. One reason is that there are various combinations of values for the input variables which give the same value for a special value set of the output variables. Eventually, when the Artificial Neural Network is going to simulate the inverse function, there are different combinations of values for a specific set of the output variable, so it may generate the output which is different comparing to the desired output. Simulating the inverse function of compound option pricing model enables us to have an estimation of current underlying price, base option strike price, base option time to maturity, compound option strike price, compound option time to maturity, interest rate and implied volatility based mainly on the compound option price. The focus of this study is to simulate the inverse function of compound option pricing model using
Artificial Neural Networks (ANNs) to estimate the other variables of compound option pricing model.

Two types of ANNs are used to model the inverse function of the compound option pricing model including Feed Forward Back Propagation Network and Radial Basis Function Network.

To measure the performance of the proposed method for simulating the inverse function of the compound option pricing model using the ANNs, the system is simulated with both methods, one time with the current method of applying ANNs on the inverse function and one time using the suggested method. The results of each simulation are compared with each other. Actually, inverse function of the compound option pricing model is simulated first time with regular ANN model. Then, the simulation is repeated using 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 Networks and Radial Basis Functions Networks are compared with each other.

The rest of this study is organized as followings. Section 7.2, is going to explain the Compound option pricing model briefly. Section 7.3, 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 7.4. At the Section 7.5, the data of the study which will be used for the simulations will be presented and analyzed. Section 7.6, 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 7.7, as the final section of this chapter provides the conclusion of this study.
3. Compound Option Pricing Model

The derivatives can be categorized according to their features such as cash flows, time dependence, weak path dependence, strong path dependence, dimensionality, the order of the option and embedded decisions related to the options. Compounds are a type of exotic options. Exotic options are much harder to price than the regular options. Compounds are options which are applied on options (Wilmott, 2007).

A compound option, according to its type, gives its holder the right to buy or sell another option. Having a compound call option on a call option, there is the right to buy a call option at a specified time for a specified price. If the compound option is applied or exercised, there will be a call option to buy an underlying (a stock for example) at a specified time for a specified price. Compound option is a second order derivative because they give the right for exercising over another derivative. In this study, the compound options pricing model which has been developed based on the Black-Scholes model is considered as the base model. The Black-Scholes (BS) model is one of the most well-known option pricing models in finance. Fischer Black, Myron Scholes, and Robert Merton introduced the BS model for pricing the options in 1970 and the model absorbed lots of attentions in the field of option pricing (Black and Scholes, 1973). Although BS model can be considered for pricing the second order contracts but it is not a satisfactory model for real situation. In real situation, when a compound option is exercised, an option is born at the market price not at the calculated theoretical price.

Based on the BS model, in order to price the compound options, two steps should be followed. First, the underlying option should be priced according to the BS model as following.
Showing the basic option expiration time 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 Equation 7.1 as following (Black and Scholes, 1973; Wilmott, 2007; Hull, 2002):

\[
\frac{\partial f}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} + rS \frac{\partial f}{\partial S} - rf = 0 \tag{Eq. 7.1}
\]

The current value \( f(S_0, 0) \), where \( S_0 = S(t = 0) \), is the option price.

The second step is to price the compound option again based on the BS model. But in this step, the underlying price is the underlying option price. And the calculated option price based on the BS model is the compound option price (Black and Scholes, 1973; Wilmott, 2007; Hull, 2002).

Showing the expiration time of the compound option with \( T_{co} \) and the current time with \( t_{co} \) where \( t_{co} < T_{co} < T \), the current price of the underlying option, \( f(t_{co}) \), is known, but the price of the underlying option at maturity of the compound option, \( f(T_{co}) \), is not known. According to the Black and Scholes, \( f(T_{co}) \) can be considered as a random variable and the previous differential equation model, equation 7.1, can be replaced as the dynamics of the price \( f(t_{co}) \) as a function of time; Let \( Co(f(S(t), t), t_{co}) \) be the price of the compound option at time \( t_{co} \), or in short \( Co(f, t_{co}) \). Under suitable assumptions, the value of the compound option can be formulated as Eq. 7.2 (Black and Scholes, 1973; Wilmott, 2007; Hull, 2002):

\[
\frac{\partial Co}{\partial t_{co}} + \frac{1}{2} \sigma^2 f^2 \frac{\partial^2 Co}{\partial f^2} + rf \frac{\partial Co}{\partial f} - rCo = 0 \tag{Eq. 7.2}
\]
The current value $C_0(f_0, 0)$, where $f_0 = f(t = 0)$, is the Compound price (Black and Scholes, 1973; Wilmott, 2007; Hull, 2002).

The final compound option pricing differential equation can be rewritten as the equation 7.3 as following(Black and Scholes, 1973; Wilmott, 2007; Hull, 2002):

$$
\frac{\partial C_0}{\partial t} + \frac{1}{2} \sigma^2 \left( \frac{\partial f}{\partial t} \right)^2 + \frac{1}{2} \sigma^2 S^2 \left( \frac{\partial^2 f}{\partial S^2} \right)^2 + rS \left( \frac{\partial f}{\partial S} - rf \right)^2 \frac{\partial^2 C_0}{\partial S^2} + r \frac{\partial f}{\partial t} + \frac{1}{2} \sigma^2 S^2 \left( \frac{\partial^2 f}{\partial S^2} - rf \right)^2 +
$$

$$
rS \left( \frac{\partial f}{\partial S} - rf \right) \frac{\partial C_0}{\partial S} - rC_0 = 0 \quad \text{(Eq. 7.3)}
$$

Assuming that both underlying and underlying option price have the normal distributions, equation 7.1 and 7.2 can be simplified and be solved analytically. Assuming these, the basic European call option price based on the Black-Scholes formula is:

$$
C = S_0 N(d_1) - X e^{-rT} N(d_2) \quad \text{(Eq. 7.4)}
$$

Where

$$
d_1 = \frac{\ln(S_0/X) + (r + \sigma^2/2)T}{\sigma \sqrt{T}} \quad \text{(Eq. 7.5)}
$$

$$
d_2 = \frac{\ln(S_0/X) + (r - \sigma^2/2)T}{\sigma \sqrt{T}} = d_1 - \sigma \sqrt{T} \quad \text{(Eq. 7.6)}
$$

$N(x)$ 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 basic option strike price, $T$ is the basic option maturity, $\sigma$ is the volatility of the underlying asset price and $r$ is the risk-free interest rate(Black and Scholes, 1973; Wilmott, 2007; Hull, 2002).
After calculating the basic option price, the option price can be considered as the underlying price to calculate compound option price based on the analytical solution of the second BS model. So, the basic compound European call option price will be as:

$$C_o = CN(d_3) - X_{C_o}e^{-r_{C_o}T_{C_o}}N(d_4)$$

(Eq. 7.7)

Where

$$d_3 = \frac{\ln(C_{C_o}/X_{C_o})+(r+\sigma^2/2)T_{C_o}}{\sigma\sqrt{T_{C_o}}}$$

(Eq. 7.8)

$$d_4 = \frac{\ln(C_{C_o}/X_{C_o})+(r-\sigma^2/2)T_{C_o}}{\sigma\sqrt{T_{C_o}}} = d_3 - \sigma\sqrt{T_{C_o}}$$

(Eq. 7.9)

The variable $C_o$ is the Compound European call option price, $X_{C_o}$ is the compound option strike price, $T_{C_o}$ is the compound option maturity, $\sigma$ is the volatility of the underlying option price which is the same as the volatility of the underlying asset of the basic option which the compound option is calculating on that and $r$ is the risk-free interest rate (Black and Scholes, 1973; Wilmott, 2007; Hull, 2002).

In this study, the inverse function of the compound option pricing model, equation 7.3, is going to be simulated. Having a good simulation, the current underlying asset price, $S$, basic option strike price, $X$, the time to expiration or maturity of the basic option, $T$, the volatility of the underlying asset price, $\sigma$, the risk-free interest rate, $r$, compound option strike price, $X_{C_o}$, and maturity of the compound option, $T_{C_o}$ can be estimated mainly based on the compound European Call option Price, $C_o$, using ANNs.

The model has 7 independent variables and one dependent variable, so the data sets should include these seven variables and they will be fed into the networks according to each network’s design. Again, in order to generalize the model, it should be independent of any special underlying asset. The moneyness which is defined as the ratio of Strike price to the current price
of the underlying asset (Equation 7.10) is used for this generalization. The number of the input variables will be decreased to 6 from 7 considering this new variable, moneyness.

4. Developing inverse models

The forward function of the compound option pricing model uses the current underlying asset price, basic option Strike price, volatility of the underlying asset price, rate of interest, time to maturity of the basic option, compound option Strike price, and the time to maturity of the compound option as the inputs to calculate compound option price as the output of the network as following block diagram. As described before, the ratio of basic option strike price to the underlying asset price will be replaced with the moneyness. The schematic view of the direct function is as figure 7.1.

According to the explanations of section 6.5, considering figure 7.1, the direct function of the compound option pricing model can be considered as the case where $m < n$, so the inverse function is the system where $m > n$ as figure 7.2.

![Figure 1: schematic view of the Compound option direct function](image-url)
Again, considering the explanations of section 6.5, in this case the ANN can just find an approximate solution for the inverse function. In order to have a better solution, the system should be transformed to situation where \( m = n \). In order to match this, the suggested method, as section 6.5, is to add some dummy variables to the system to equalize the number of the inputs and outputs. In this manner, since there is one solution for the system, the ANN can work better. Actually, the set of inputs will be completed with the virtual dummy functions.

As described before, the inverse function of the compound option pricing model is going to be simulated. For this model, the compound option price is considered as the input, and the other 6 variables are considered as the output of the network (As explained before, two variables including current underlying asset price and strike price are combined to form one variable called moneyness). The final network which simulates inverse of compound option pricing model will have one input and six outputs as figure 7.2. Since in this system the number of the outputs is more than the number of the inputs, the ANN can find an approximate solution. According to the suggested method, the set of outputs of the direct function should be completed with dummy variables (functions). After this, the number of the inputs and outputs would be the same. So, five dummy variables should be added to the function as the inputs of the inverse function (output set of the direct function) as figure 7.3.
Shortly, the inverse function has one input and six outputs, so it needs five dummy variables to equalize the number of the inputs of the inverse function to the number of the outputs. Based on the system of equation 7.11, these dummy functions are functions of the input variables of the direct function so they 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 each of the input variables of the direct model (output variables of the inverse model) or a function of a combination of these variables (Eq.7.11) including moneyness, basic option time to maturity, basic option strike price, volatility, interest rate, compound option strike price and compound option time to maturity.

\[
\begin{align*}
    y_1 &= f_1(x_1, x_2, ..., x_n) \\
    y_2 &= f_2(x_1, x_2, ..., x_n) \\
    &\vdots \\
    y_m &= f_m(x_1, x_2, ..., x_n) \\
    y_{m+1} &= f_{m+1}(x_1, x_2, ..., x_n) \\
    &\vdots \\
    y_{n+m} &= f_n(x_1, x_2, ..., x_n)
\end{align*}
\]

Set of dummy functions

\[
\left\{ y_{m+i} = f_{m+i}(x_1, x_2, ..., x_n) \right\}_{i=1}^{N}
\]

as \( i = 1, ..., N \), \( x \in \mathbb{R}^n \), \( y \in \mathbb{R}^{n-m} \)

Eq. 7.11

Figure 3: Added dummy variables as the inputs of the inverse function
In this study, the inverse function of the compound option pricing model (without dummy variables) is considered as the base model. The results of the base model simulation will be compared with the suggested method results. Two different Artificial Neural Networks will be designed. One model for simulating the base inverse function of the compound option pricing model, as figure 7.2, and one which simulates the inverse function of compound option pricing model with the added dummy variables, as figure 7.3.

The base model has the European call compound option price as its input and the other 6 variables including moneyness, basic option time to maturity, basic option strike price, volatility, interest rate, compound option strike price and compound option time to maturity as its outputs. The suggested model has 5 dummy variables, as explained before, in addition as its inputs. Two different sets of dummy variables have been applied and tested in this study. For the first try, the dummy variables are defined as the functions of the 5 of the 5 output variables. The five dummy functions which are defined in this step are as (1: squared of the estimation of moneyness), (2: squared of the estimation of basic option time to maturity), (3: the complement of the estimation of the volatility (1 - the estimation of the volatility)), (4: squared of the estimation of the compound option time to maturity) and (5: squared of the estimation of the compound option strike price).

For the second try, the dummy variables are defined as the functions of the 3 of the 5 output variables. The five dummy functions which are defined in this step are as (1: the estimation of moneyness to the power of 2), (2: the estimation of the compound option strike price to the power of 2), (3: the complement of the estimation of the volatility (1 - the estimation of the volatility)), (4: the estimation of moneyness to the power of 3) and (5: the estimation of the compound option strike price to the power of 3).
Two networks should be designed for each of the both Feed Forward Back Propagation Network and Radial Basis Function Network. Finally, the study includes four ANNs, two FFBPNs and two RBFNs for simulating the base model and suggested model for each set of the dummy variables.

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 6 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 input 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 the number of the neurons of each hidden layer will be refined based on the performance of the training. The transfer
functions of the Feed Forward Back Propagation Network and the value of the spread will be refined to reach to the optimal level.

5. Results evaluation

As there are different values of the moneyness, interest rate, volatility and time to maturity for both basic option which the compound option is applied on that and also the compound option, the forecasted values of the networks cannot be compared with the appropriate set of the presented data (initial ones) which has been applied to the network. According to the model, various different sets of the forecasted values could generate the same compound option price so this point should be considered in results evaluation. In order to deplete this problem and have a good evaluation of the results precision, the option price will be calculated based on the forecasted values of the network (moneyness, interest rate, volatility and time to maturity) using the compound option pricing formulas analytically or using an ANN which is trained based on compound option pricing data which are generated using Compound pricing formula. This ANN simulates the Compound option pricing formula. The calculated compound option prices will be compared with the initial option prices. 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 compound option prices, for situation where there exist the explicit pricing model, compound formula has been used. In another run, the simulation process has been done using the simulated compound 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 compound formula. In this case, in order to check the results from the inverse function, the model outputs are fed into the compound formula to
calculate the compound option price and this price is compared with the network outputs to
determine the model precision.

The training and testing data series for the second run is done using the simulated compound
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 compound option pricing ANNs, which simulates
the compound formula, to calculate the compound option prices and these prices are compared
with the network outputs to determine the model precision again. This additional testing is done
to show that where there is not an explicit function, the direct function can 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
Compound option pricing model.

6. Data

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

1-The moneyness of the basic option, \( \frac{X}{S_0} \) (Strike price/Current underlying asset price), \( m \in [1, 1.1] \)
2-The volatility of the stock price, $\sigma \in [20\% \; 30\%]$

3-The risk-free interest rate, $r \in [1\% \; 5\%]$

4-The time to expiration for the basic option, $T \in [2\; 4]$ month

5-The time to expiration for the compound option, $T_{Co} \in [1\; 3]$ month

6-The compound option strike price, $X_{Co} \in [1.1 \; basic\; option\; price]$

A maximum value of 1.1 is considered for the moneyness. As described before, the current price is set equal to 1 for all the data points, so the strike price will be determined using these two values, moneyness and current price. Normally, for the normal stocks the range of volatility is between 20% and 40% (Hull, 2002), so in this study a range of 20% to 30% has been considered for generating random values for this variable. It is also considered a range of minimum 1% and maximum 5% for the rate of interest and a range of 2 to 4 month for the basic option maturity time. Since the maturity of the compound option should be less than the maturity of the basic option, a range of 1 to 3 months is considered for the compound option maturity time.

According to these ranges, random values have been generated in accordance with their predefined appropriate ranges. The data are generated using the uniform random generator function. The compound European call option price, $Co$, has been calculated using equations 7.4, 7.5, 7.6, 7.7, 7.8 and 7.9 which are based on the Black-Scholes formula for the compound options. 1000 series of data have been generated for each of the training and testing sets.
7.1 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 and tested. After training and testing the networks, the combination that provided the best results were chosen and are presented in table 7.1, 7.2, 7.3, and 7.4.

The result for the situation where the simulation and evaluation is done using compound formula is according to tables 7.1 and 7.2.

The result for the situation where the simulation and evaluation is done using the simulated compound formula with the ANNs, is according to tables 7.3 and 7.4.

For the first set of dummy variables, the network architectures for FFBPN for the inverse base case is a 4 layered network with the 1-4-5-6 architecture and for the inverse dummy added case is a 4 layered network with the 6-12-10-6 architecture.

For the second set of dummy variables, the network architectures for FFBPN for the inverse base case is a 4 layered network with the 1-4-5-6 architecture and for the inverse dummy added case is a 4 layered network with the 6-12-10-6 architecture. For both types of Artificial Neural Networks the results has been brought in Tables 7.1, 7.2, 7.3, and 7.4.
Table 1: Artificial Neural Networks training and testing results for first dummy variables set-
simulation and evaluation using compound formula

<table>
<thead>
<tr>
<th>System Type</th>
<th>Network Type</th>
<th>'MAPE'</th>
<th>'MSE'</th>
<th>'MAE'</th>
</tr>
</thead>
<tbody>
<tr>
<td>System with regular inputs</td>
<td>FFBPN</td>
<td>Training</td>
<td>8.22</td>
<td>4.12E-09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>8.04</td>
<td>4.96E-09</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Training</td>
<td>7.98</td>
<td>2.41E-09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>7.93</td>
<td>2.51E-09</td>
</tr>
<tr>
<td>System with added dummy variables</td>
<td>FFBPN</td>
<td>Training</td>
<td>4.02</td>
<td>1.37E-09</td>
</tr>
<tr>
<td>(function)</td>
<td></td>
<td>Testing</td>
<td>4.18</td>
<td>1.47E-09</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Training</td>
<td>2.02</td>
<td>3.02E-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>3.19</td>
<td>1.21E-09</td>
</tr>
</tbody>
</table>
Table 2: Artificial Neural Networks training and testing results for second dummy variables set—simulation and evaluation using compound formula

<table>
<thead>
<tr>
<th>System Type</th>
<th>Network Type</th>
<th>'MAPE'</th>
<th>'MSE'</th>
<th>'MAE'</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>System with regular inputs</strong></td>
<td>FFBPN</td>
<td>Training</td>
<td>8.22 E-09</td>
<td>4.69 E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>8.04 E-09</td>
<td>4.76 E-05</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Training</td>
<td>7.98 E-09</td>
<td>4.20 E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>7.93 E-09</td>
<td>4.29 E-05</td>
</tr>
<tr>
<td><strong>System with added dummy variables(function)</strong></td>
<td>FFBPN</td>
<td>Training</td>
<td>4.60 E-09</td>
<td>2.80 E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>4.76 E-09</td>
<td>2.76 E-05</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Training</td>
<td>4.79 E-09</td>
<td>2.25 E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>4.43 E-09</td>
<td>2.38 E-05</td>
</tr>
</tbody>
</table>
Table 3: Artificial Neural Networks training and testing results for first dummy variables set-simulation and evaluation using simulated compound formula with the ANNs

<table>
<thead>
<tr>
<th>System Type</th>
<th>Network Type</th>
<th>'MAPE'</th>
<th>'MSE'</th>
<th>'MAE'</th>
</tr>
</thead>
<tbody>
<tr>
<td>System with regular inputs</td>
<td>FFBPN</td>
<td>Training 10.15</td>
<td>5.24E-07</td>
<td>8.15E-04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing 9.04</td>
<td>1.05E-07</td>
<td>7.34E-04</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Training 5.69</td>
<td>3.26E-07</td>
<td>5.67E-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing 7.96</td>
<td>1.23E-07</td>
<td>6.31E-04</td>
</tr>
<tr>
<td>System with added dummy variables</td>
<td>FFBPN</td>
<td>Training 6.95</td>
<td>6.84E-09</td>
<td>6.95E-05</td>
</tr>
<tr>
<td>(function)-simulated Compound model</td>
<td></td>
<td>Testing 5.39</td>
<td>5.62E-09</td>
<td>5.09E-05</td>
</tr>
<tr>
<td>with ANN</td>
<td>RBFN</td>
<td>Training 4.57</td>
<td>1.31E-09</td>
<td>2.30E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing 5.10</td>
<td>2.94E-09</td>
<td>2.77E-05</td>
</tr>
</tbody>
</table>
Table 4: Artificial Neural Networks training and testing results for second dummy variables simulation and evaluation using simulated compound formula with the ANNs

<table>
<thead>
<tr>
<th>System Type</th>
<th>Network Type</th>
<th>'MAPE'</th>
<th>'MSE'</th>
<th>'MAE'</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>System with regular inputs</strong></td>
<td>FFBPN</td>
<td>Training</td>
<td>10.15</td>
<td>5.24E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>9.04</td>
<td>1.05E-07</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Training</td>
<td>5.69</td>
<td>3.26E-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>7.96</td>
<td>1.23E-07</td>
</tr>
<tr>
<td><strong>System with added dummy variables(function)-simulated Compound model with ANN</strong></td>
<td>FFBPN</td>
<td>Training</td>
<td>5.76</td>
<td>3.35E-09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>6.14</td>
<td>3.27E-09</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Training</td>
<td>5.62</td>
<td>4.31E-09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing</td>
<td>5.92</td>
<td>5.36E-09</td>
</tr>
</tbody>
</table>
Table 7.5 and 7.6 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 compound option pricing model for both the systems considering both sets of dummy variables and both evaluation processes.
Table 5: Final results comparison- simulation and evaluation using compound formula

<table>
<thead>
<tr>
<th>System Type</th>
<th>Network Type</th>
<th>'MAPE'</th>
<th>'MSE'</th>
<th>'MAE'</th>
</tr>
</thead>
<tbody>
<tr>
<td>System with regular inputs</td>
<td>FFBPN</td>
<td>Testing</td>
<td>8.04</td>
<td>4.96E-09</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Testing</td>
<td>7.93</td>
<td>2.51E-09</td>
</tr>
<tr>
<td>System with added dummy variable(function)</td>
<td>FFBPN</td>
<td>Testing</td>
<td>4.18</td>
<td>1.47E-09</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Testing</td>
<td>3.19</td>
<td>1.21E-09</td>
</tr>
<tr>
<td>System with added dummy variable(function)</td>
<td>FFBPN</td>
<td>Testing</td>
<td>4.76</td>
<td>2.29E-09</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>Testing</td>
<td>4.43</td>
<td>1.38E-09</td>
</tr>
</tbody>
</table>
Table 6: Final results comparison - simulation and evaluation using simulated compound formula with the ANNs

<table>
<thead>
<tr>
<th>System Type</th>
<th>Network Type</th>
<th>'MAPE'</th>
<th>'MSE'</th>
<th>'MAE'</th>
</tr>
</thead>
<tbody>
<tr>
<td>System with regular inputs</td>
<td>FFBPN</td>
<td>9.04</td>
<td>1.05E-07</td>
<td>7.34E-04</td>
</tr>
<tr>
<td></td>
<td>RBFN</td>
<td>7.96</td>
<td>1.23E-07</td>
<td>6.31E-04</td>
</tr>
<tr>
<td>System with added dummy variable(function)</td>
<td>FFBPN</td>
<td>5.39</td>
<td>5.62E-09</td>
<td>5.09E-05</td>
</tr>
<tr>
<td>First set</td>
<td>RBFN</td>
<td>5.10</td>
<td>2.94E-09</td>
<td>2.77E-05</td>
</tr>
<tr>
<td>System with added dummy variable(function)</td>
<td>FFBPN</td>
<td>6.14</td>
<td>3.27E-09</td>
<td>3.65E-05</td>
</tr>
<tr>
<td>Second set</td>
<td>RBFN</td>
<td>5.92</td>
<td>5.36E-09</td>
<td>3.93E-05</td>
</tr>
</tbody>
</table>
Comparing the testing results from the Artificial Neural Networks with regular inputs with the suggested system which has the first added dummy variable set 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 compound formula, in FFBPN model, the MAPE is equal to 8.04% while the MAPE for the new method is 4.18% and for RBFN the MAPE is equal to 7.93% while the MAPE for the new method is equal to 3.19%. The other evaluation measures such as MAE and MSE suggest the same. For the case of evaluation process with the simulated compound formula with ANN, in FFBPN model, the MAPE is equal to 9.04% while the MAPE for the new method is 7.9% and for RBFN the MAPE is equal to 7.96% while the MAPE for the new method is equal to 5.1%. 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 set 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 compound formula, in FFBPN model, the MAPE is equal to 8.04% while the MAPE for the new method is 4.76% and for RBFN the MAPE is equal to 7.93% while the MAPE for the new method is equal to 4.43%. The other evaluation measures such as MAE and MSE suggest the same. For the case of evaluation process with the simulated compound formula with ANN, in FFBPN model, the MAPE is equal to 9.04% while the MAPE for the new method is 6.14% and for RBFN the MAPE is equal to 7.96% while the MAPE for the new method is equal to 5.92%. 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 the new 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.

7. Conclusions

In this study, Artificial Neural Networks have been used to simulate the inverse function of the compound option pricing model which is a model for pricing the options on 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 demonstrate the better performance of new suggested method of using Artificial Neural Networks in simulating the inverse function of compound option pricing 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.
8. References

28. J. Hull, 2002, **FUNDAMENTALS OF FUTURES AND OPTIONS MARKETS**


42. Li, R., Meng, H., Dai, Y., “The Valuation of Compound Options on Jump-Diffusions with Time-Dependent Parameters”, IEEE, 0-7803-8971, p 1290-1294, 2205,


52. Po-Chang Ko(2009), Option valuation based on the neural regression model, Expert Systems with Applications 36, 464–471

