Adv. Math. Fin. App., 2024, **9**(2), P.723-736

Advances in Mathematical Finance & Applications

www.amfa.iau-arak.ac.ir

Print ISSN: 2538-5569

Online ISSN: 2645-4610

Original Research

Option Pricing with Artificial Neural Network in a Time Dependent Market

Mehran Araghi[a], Elham Dastranj[b,*], Abdolmajid Abdolbaghi Ata'abadi[c], Hossein Sahebi Fard[d]

[a,b,d] *Department of Mathematics, Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Semnan, Iran*

[c]*Department of Management, Faculty of Industrial Engineering and Management, Shahrood University of Technology, Shahrood, Semnan, Iran.*

ARTICLE INFO

Article history:
Received 2023-03-09
Accepted 2024-02-23

Keywords:
Option Pricing
Mikhailov and Nogel Model
Artificial Neural Network
Activation Function

ABSTRACT

In this article, the pricing of option contracts is discussed using the Mikhailov and Nogel model and the artificial neural network method. The purpose of this research is to investigate and compare the performance of various types of activator functions available in artificial neural networks for the pricing of option contracts. The Mikhailov and Nogel model is the same model that is dependent on time. In the design of the artificial neural network required for this research, the parameters of the Mikhailov and Nogel model have been used as network inputs, as well as 700 data from the daily price of stock options available in the Tehran Stock Exchange market (in 2021) as the network output. The first 600 data are considered for learning and the remaining data for comparison and conclusion. At first, the pricing is done with 4 commonly used activator functions, and then the results of each are compared with the real prices of the Tehran Stock Exchange to determine which item provides a more accurate forecast. The results obtained from this research show that among the activator functions available in this research, the ReLU activator function performs better than other activator functions.

1 Introduction

In financial mathematics, the pricing of options is one of the most important parts. An option is a type of contract between the holder and the grantor of the option, so that the holder of the option buys from the grantor the right to buy or sell a certain asset at a certain time and at a certain price. In the option contract, both parties grant each other a point. The holder pays the assignor an amount under the title of condition fee, which is actually the option price, and the assignor grants the option holder the right to buy or sell the desired asset at a certain time and at a certain price. That is, for this contract to be fair, the holder must pay an amount as the price of the option to the grantor of the option at the time of

* Corresponding author. Tel.: 09112433899
E-mail address: *elham.dastranj@shahroodut.ac.ir*

signing the contract. So the issue of option pricing, in the purchase and sale of option transactions, is investigated under different models [6].

Fischer Black, Myron Scholes, and Robert Merton (1970) initiated a major leap forward in the pricing of options contracts. The result of their work was to present a model named Black-Scholes model. This model had a significant impact on option pricing and risk hedging. The Black-Scholes model played a pivotal role in the success of financial engineering in the 1980s and 1990s and boosted the derivatives pricing market. But it lost its effectiveness in October 1978 when the European and American stock markets crashed. This model of volatility was calculated using the past changes in the price of the underlying asset, if the goal is to measure the volatility in the future [8].Therefore, in order to eliminate these criticisms, in new models such as the Heston model, volatility is considered a stochastic process, so that pricing on options contracts can be done more accurately. To analyze currency option contracts, Heston (1993) presented a model called the Heston model, which was highly scalable and used for more complex situations. Another strength of Heston model is the reversion to the mean process, which justifies the reversion to the mean property of volatility in financial markets. According to this feature, the return of an asset always moves towards the average market return, or more precisely, towards the long-term average return of the same asset. That is, returns that are lower or higher than the average always return to the average. In this model, stock price volatility is not constant and is calculated based on a random process. Kristofferson (2009) added another stochastic process to Heston model and improved this model. The double Heston model also has two stochastic volatility processes, compared to the standard Heston model, it has more flexibility for extreme prices. Making the parameters of this model dependent on time, Mikhailov and Nogel improved Heston model so that the price of short-term options is closer to the real market. Neural network science researchers, McCulloch and Pitts (1940), conducted studies on the internal communication ability of a neuron model. The result of their work was to present a computational model based on a simple pseudo-neuronal element. At that time, other scientists such as Donald Hebb were also working on the rules of adaptation in neuronal systems. Donald Hebb (1949) proposed a learning rule for adapting connections between artificial neurons. Verbus (1971) published a back propagation algorithm in his doctoral dissertation, and finally Rosenblatt (1986) rediscovered this technique. Nowadays, Artificial Neural Networks are widely studied with the aim of achieving human-like efficiency. These networks are composed of linear and non-linear computing elements that operate in parallel. Among the most important features of artificial neural networks is the ability to learn, generalize and parallel processing. So artificial neural networks can learn complex tasks that are difficult for rule-based systems .Artificial neural networks are inspired by the human brain system. In the human brain, there are billions of neurons, through which learning is done, and humans can predict future phenomena using the past they have experienced. Artificial neural network is used as a method to simulate neurons.In order to investigate the pricing of option trading using the Mikhailov and Nogel model and artificial neural networks, articles that are close to the mentioned subject in terms of content were studied. Topics such as artificial neural network, option pricing, Heston and Mikhailov and Nogel model or a combination of them.In recent years, several financial studies have been conducted on options pricing. In [2] the hypothetical options pricing have been done based on fractional Heston model in Iran’s gold market.European options with transaction cost under some Black-Scholes markets are priced with the gold asset as underlying asset in [3]. In [13] European option pricing is driven when zero-coupon bond is considered as underlying asset in a market under Hull-White model. In [15], the optimized artificial neural network is trained on a dataset created by a complex financial model and tested on three models: Black-Scholes, Heston, and Brent's iterative rooting method, and shows that artificial neural networks It can significantly decreases the computation time. Combining artificial neural networks and two Black-Scholes and Heston models in [7], it is concluded that artificial neural networks can be considered as an efficient alternative to existing quantitative models for option pricing. Results available in [11] dedicates the pricing of Bitcoin options sheets are systematically overpriced by classical methods, while there is a significant improvement in price prediction using neural network models. In this research, a number of daily option prices were extracted from the Tehran stock market, and an artificial neural network was designed using the parameters of the Mikhailov and Nogel model. Then the initial learning of the artificial neural network, pricing on the extracted options was done by 4 activation functions. In the sequel their results were compared with the actual value to determine which activator function has better performance than the other functions.This research consists of 6 parts. In section 2, European call option is presented and in section 3 Mikhailov and Nogel model as a time dependent Heston model and parameter estimation are presented. In section 4 artificial neural network and considered activation functions are discussed. The results of option pricing with neural network in a real market and error analysis are examined in section 5. The paper is conclude in section 6.

2 European call option

Option contracts are usually divided into two categories: purchase option and put option. The price specified in the contract for buying or selling in the future is called the agreed price (E) and the time set for the execution of the contract is called the maturity date (t). The value of each call option contract for the underlying asset at the price S is calculated from equation (1) [6]

$$c(t,E) = \max(S-E,0)\,, \tag{1}$$

The value of each put option contract is obtained from

$$p(t,E) = \max(E-S,0)\,. \tag{2}$$

Trading options are also divided into two types, European and American, in terms of contract execution time, the difference being that the European trading option can only be executed at the time of maturity, but the American trading option can also be executed before the maturity date [1]. In this research, European options are used for pricing.

3 Mikhailov and Nogel model

In Heston model, the price process of the basic asset, namely S_t, at moment t is

$$dS_t = \mu(S_t)dt + \sqrt{v_t(S_t)}\,dZ_{t_1}, \qquad t \in [0,T]\,, \tag{3}$$

For volatility in this model, another stochastic differential equation is considered

$$dv_t = \kappa(\theta - v_t)dt + \sigma\sqrt{v_t}(S_t)dZ_{t_1}, \qquad t \in [0,T]\,. \tag{4}$$

where S_t is the basic asset price process, v_t is the basic asset price volatility process, Z_{t_1} and Z_{t_2} are the Brownian process with correlation coefficient ρ. θ is the long-term mean, κ is the rate of return to the mean, and σ is the variance of the volatility process. The Mikhailov and Nogel model is actually Heston model, which is dependent on time and is expressed as follows [9].

$$dS_t = rS_tdt + \sqrt{v_t}S_tdW_{1,t}\,, \tag{5}$$

$$dv_t = \kappa_t(\theta_t - v_t)dt + \sigma_t\sqrt{v_t}dW_{2,t}\,, \tag{6}$$

where S_t is the basic asset price change process and v_t is the standard deviation process of the basic asset price and r is the risk-free interest rate and $\kappa_t > 0$ is the rate of return to the average of the basic asset price process, $\theta_t > 0$ is the long-term average of the basic asset price process, $\sigma_t > 0$ is the price fluctuation process It is a basic asset. Also, $W_{1,t}$ and $W_{2,t}$ are two Brownian processes with correlation coefficient ρ_t, that is, the following relationship holds

$$\mathbb{E}^{\mathbb{P}}\left[dW_{1,t}dW_{2,t}\right] = \rho_t dt\,. \tag{7}$$

The differential equation related to the Mikhailov and Nogel model is

$$\frac{\partial C}{\partial t} + \frac{1}{2}v_t\frac{\partial^2 C}{\partial S^2} + \left(r_t - \frac{1}{2}v_t\right)\frac{\partial C}{\partial S} + \rho_t\sigma_t v_t\frac{\partial^2 C}{\partial v\partial S} + \frac{1}{2}\sigma_t^2 v_t\frac{\partial^2 C}{\partial v^2} - r_t C + [\kappa_t(\theta_t - v_t) - v_t]\frac{\partial C}{\partial v} = 0,\ [16]\,. \tag{8}$$

3.1 Parameter estimation

At first, the following points should be taken into account; The initial value of v_t, called v_0, is always positive. κ the rate of return to the mean of the underlying asset price process, σ the underlying asset price volatility process, and θ the long-term mean of the underlying asset price process all have positive values. and Feller's condition, which is expressed as $2\kappa\theta = \sigma^2$ [12].

Theorem: The MLE estimation for calculating parameters κ θ and σ in each time step is

$$\hat{\kappa} = \frac{2}{\delta}\left(1 + \frac{P\delta}{2}\frac{1}{n}\sum_{k=1}^{n}\frac{1}{V_{k-1}} - \frac{1}{n}\sum_{k=1}^{n}\sqrt{\frac{v_k}{v_{k-1}}}\right), \tag{9}$$

$$\hat{\sigma} = \sqrt{\frac{4}{\delta}\frac{1}{n}\sum_{k=1}^{n}\left[\sqrt{v_k} - \sqrt{v_{k-1}} - \frac{\delta}{2\sqrt{V_{k-1}}}(P - \kappa v_{k-1})\right]^2}\,, \tag{10}$$

$$\hat{\theta} = \frac{P + \frac{1}{4}\sigma^2}{\kappa}, \tag{11}$$

$$\hat{P} = \frac{\frac{1}{n}\sum_{k=1}^{n}\sqrt{v_{k-1}v_k} - \frac{1}{n^2}\sum_{k=1}^{n}\sqrt{\frac{v_k}{v_{k-1}}}\sum_{k=1}^{n}v_{k-1}}{\frac{\delta}{2} - \frac{\delta}{2}\frac{1}{n^2}\sum_{k=1}^{n}\frac{1}{v_{k-1}}\sum_{k=1}^{n}v_{k-1}}, \tag{12}$$

where $n = \frac{T}{\delta}$ and δ is an arbitrary positive number.[12]

4 Artificial neural network

Artificial neural network is built as a method to simulate neurons or a network of neurons in the brain, which is generally made of three main parts; Input layer, hidden layer, output layer. In the artificial neural network system, the processing elements are called artificial neurons or nodes. In this model, the flow of information from input to output units is one-way. Data passes through multiple nodes without any information feedback.

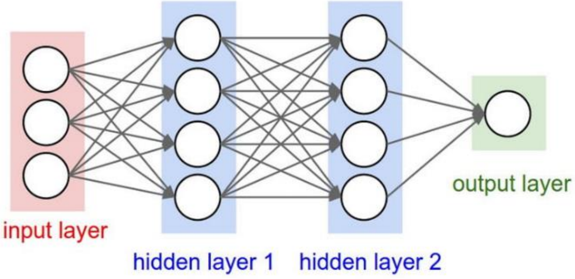


Fig. 1: Simple simulation of an artificial neural network [4]

First, the primary signals or data enter the input layer, and then they are modified by multiplying by a modified weight and enter the node in the next layer, and the modified signals in each node are added together and modified again by multiplying by a new weight until the final output. In this process, the system seeks to find the most optimal weights so that the output of the system is as close to the expected output as possible. This operation is performed for a large number of initial data to obtain the final optimal weights. The components of the artificial neural network are: Input, hidden and output layer, weights, bias, activation and cost function.

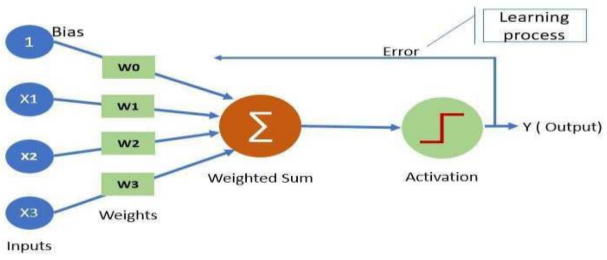


Fig. 2: Mathematical scheme of a neuron [5]

In the artificial neural network of Figure 2, the value of the output layer is calculated from

$$y = f(x) = f(\sum x_i w_i + B)\,, \qquad (i = 1,2,\dots,m). \tag{13}$$

In this system, interaction (Bias) is an additional parameter that is defined as the bias of artificial neural network. In this way, it is added to the product of variables and weights and helps the model to direct the activator function to the positive or negative side. The role of bias is similar to the role of constant value in linear function [4].

4.1 Feed forward

In the feed forward propagation stage, the information flow is forward. In other words, in feed forward propagation, through the input layer, data enters the network and by applying calculations on them using activation functions in hidden layers, the output of each layer is transferred to its next layer until finally the output of the network is determined in the last layer. In the feed forward propagation stage, the activator function is considered as a "port" that sends the inputs of each layer to the next layer [4].

4.2 Back propagation

The purpose of the back propagation step is to reduce the value of the cost function by adjusting the values of the weights and biases of the network. The gradients of the cost function determine the amount of changes according to parameters such as activation function, weights, bias and other relevant items.

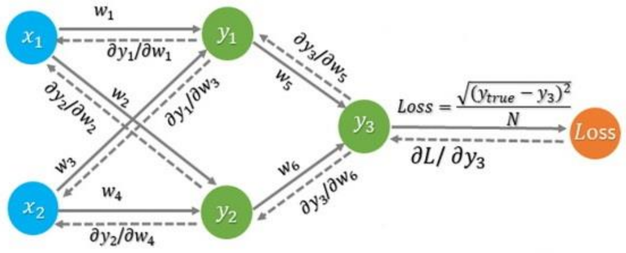

x_1
x_2
y_1
y_2
y_3
w_1
w_2
w_3
w_4
w_5
w_6
$\partial y_1/\partial w_1$
$\partial y_2/\partial w_2$
$\partial y_1/\partial w_3$
$\partial y_2/\partial w_4$
$\partial y_3/\partial w_5$
$\partial y_3/\partial w_6$
$Loss = \frac{\sqrt{(y_{true} - y_3)^2}}{N}$
$\partial L/\ \partial y_3$
Loss

Fig. 3: Computation method of an artificial neural network [4]

4.3 Loss function

The task of the cost function, as an important part of the artificial neural network system, is to check the predicted error rate with the actual value and correct the existing weights. The cost function for artificial neural networks is defined as follows

$$cost(t) = -\frac{1}{m}\sum_{t=1}^{m}\sum_{k=1}^{k}[y_k^{(t)}\log\left(\left(h_\theta\left(x^{(t)}\right)\right)_k\right) + (1 - y_k^{(t)})\log(1 - \left(h_\theta\left(x^{(t)}\right)\right)_k)] + \quad (14)$$
$$\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\theta_{j.i}^{(l)}\right)^2, \ [4].$$

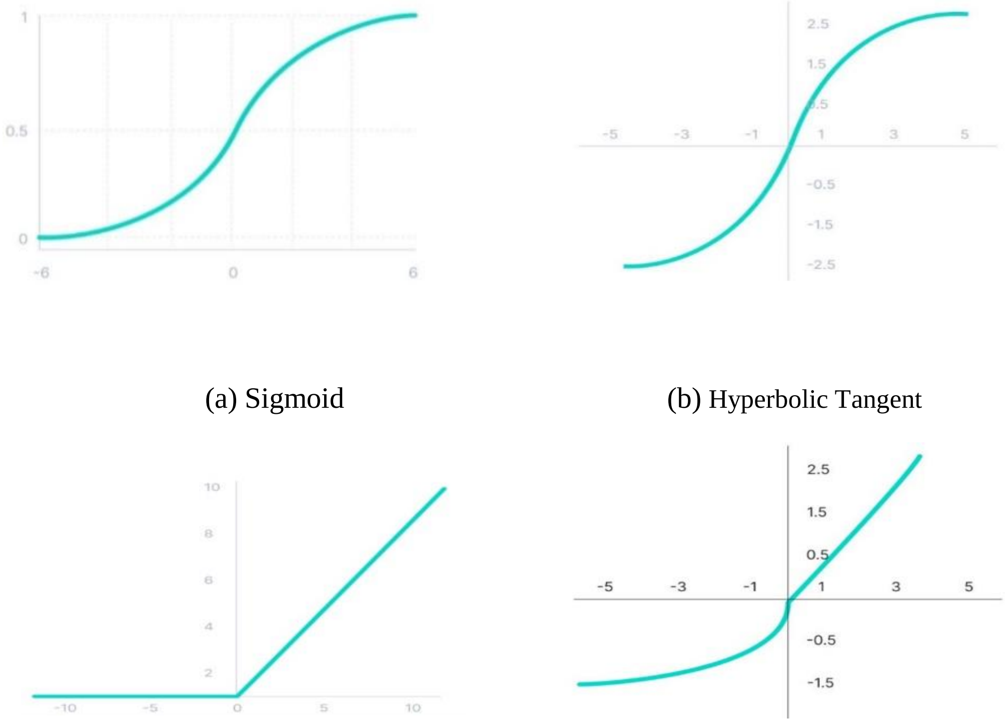
In relation (15) L is the total number of layers in the neural network, s_l the number of nodes in the first layer (not including the bias node), k the number of output classes, $\theta^{(l)}$ the weight matrix of the first layer, $y_k^{(t)}$ the actual value of the tth node in the output layer and $h_\theta\left(x^{(t)}\right)$ is the prediction value of the artificial neural network in the tth node of the output layer. Now, if we consider a simple classification and a class ($k = 1$) and regularization is ignored, then our cost function will be calculated as

$$cost(t) = y^{(t)}\log\left(h_\theta\left(x^{(t)}\right)\right) + \left(1 - y^{(t)}\right)\log(1 - h_\theta\ (x^{(t)}))\ . \quad (15)$$

4.4 Activation function

The function that is placed on the neuron to affect the inputs of the neuron and produce the desired output is called the activation function of that neuron. The main component of artificial neural networks is the activation functions. Without the activation functions, the work of artificial neural network will be like linear functions; In this case, it will not work for complex models. While artificial neural networks try to solve problems that are nonlinear and complex in nature. So the existence of the activation function, which is also called the transfer function, is necessary and unavoidable. Because the nodes of the input layer are combined with each other and the output is transferred to the neural network activation function. In other words, the values obtained by multiplying the weights with the input features are added together and the resulting value is transferred to the activation function. The activation function in neural networks makes the linear combination of inputs nonlinear and maps the input values to the space with a certain interval based on the type of activation function. In other words, the activator function decides which neuron will be active and which neuron will be inactive. In the following, the concepts related to the activator function will be examined.

There are many activation functions for neural networks, some of them are discussed [14].



(a) Sigmoid (b) Hyperbolic Tangent

(c) ReLU (d) SELU

Fig. 4: Activation functions

4.4.1 Sigmoid

This nonlinear activation function transforms its input to a value between 0 and 1. The larger the input value, the closer the output value of this function is to 1. While if the input value of this function is very small (negative number), the output value of the sigmoid function will be closer to zero. The sigmoid function is considered a monotonic function, but the derivative of this function is not a monotonic function. This function is considered as one of the most widely used nonlinear activation functions. The mathematical function of this activator is

$$f(x) = \frac{1}{1+e^{-x}}. \tag{16}$$

4.4.2 Hyperbolic Tangent

This function is very similar to the sigmoid activation function and the curve of this function is similar to s. The only difference between this function and the sigmoid function is the output range of this function, which maps its input value to the range between -1 and 1. The mathematical function of this activator is

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \tag{17}$$

4.4.3 ReLU (Rectified Linear Unit)

The ReLU function is very famous in the field of deep learning and is used most of the time. Although this function looks like a linear function, this function is differentiable and can be used in the backward propagation step. This function does not activate all nodes at the same time. In other words, the nodes are inactive when the input value of this function is less than zero. The mathematical function of this activator is

$$f(x) = \max(0.x)\,. \tag{18}$$

4.4.4 SELU (Scaled Exponential Linear Unit)

The SELU function performs network normalization. In other words, this function keeps the mean and variance values of each layer. This function uses positive and negative input values to shift the mean value while negative values were ignored in ReLU functions. In this function gradient is used to adjust the variance value. The mathematical function of this activator is

$$f(\alpha, x) = \lambda \begin{Bmatrix} x & (x \geq 0) \\ \alpha(e^x - 1) & (x < 0) \end{Bmatrix}. \tag{19}$$

In the artificial neural network system, the more the number of hidden layers and nodes, the more optimal the resulting weights and the more accurate the prediction, but if their number is too high, it reduces the generalization power of the network. There is no specific formula for calculating the number of hidden layers and nodes in each layer, and it is calculated by trial and error, but equation (20) is often used for the number of hidden layer nodes;

$$N_h = \frac{N_s}{\alpha(N_i + N_o)}, \tag{20}$$

where N_h is the number of hidden layer neurons, N_s is the number of samples in the training data, N_i and N_o the number of input and output neurons respectively and α the scaling factor.

5 Approximate Solutions in Real Market

In this research, first, 700 real data from stock option prices in Tehran stock market were prepared, and then the first 600 data were used to train the artificial neural network. These data are placed in the output layer and the parameters of the Mikhailov and Nogel model are placed in the input layers. Thus, for each output data, 9 parameters in the Mikhailov and Nogel model must be calculated in order to complete one step of training. The parameters calculation of the Mikhailov and Nogel model is shown in Table 1.

Table 1: Parameters of Model [12]

Definition	Parameter
0.004 × time to maturity	t
Strike price	E
Daily final price	S
0.0008 × time to maturity	r
Volatility process of underlying asset price	v_t
Correlation between S_t and v_t	ρ

t is the remaining time until maturity and because the option contracts are 6 months, its maximum value is 0.5 and minimum is assumed to be zero. To calculate the daily changes of t, just divide 0.5 by 125 (a stock market year is 252 days, so 6 stock market months are considered 125 days) $(0.5 \div 125 = 0.004)$. In this way, t = 0.5 means 6 months left until the end of the term (125×0.004=0.5) and t = 0 means the contract has ended, and in the same way, for example t = 0.3 means 75 stock market days or 112 normal days are left until the end of the contract. (75×0.004=0.3) has been used to calculate the expected return from the banks' risk-free interest rate in 1400. The risk-free interest rate is reported as 18% in 1400, which becomes 9% for 6 months (to calculate the daily changes of "r", just divide 0.09 by 125) $(0.09 \div 125 \approx 0.0008)$. Therefore, the range of variable efficiency is [0,0.09], which changes from 0.09 to 0 in a decreasing manner. To estimate κ_t, σ_t and θ_t, v_t must be calculated first. In such a way that the standard deviation of the basic asset price from the first day to the 10th day (v_1) and then the standard deviation of the basic asset price from the second day to the eleventh day (v_2) and after that from the third day to the twelfth day (v_3) and so on. The order to standard deviation of the price of the underlying asset in the last 10 days (v_t) was calculated and placed in the v_t column. Then the v_t column was divided into 10 categories and using the relations (10), (11) and (12), the mentioned parameters were calculated for each category separately. That is, first for v_1 to v_{10} and then for v_{11} to v_{20} and similarly for v_{t-10} to v_t, all three parameters were estimated. In Figure 5, the designed model of the artificial neural network of this research can be seen.

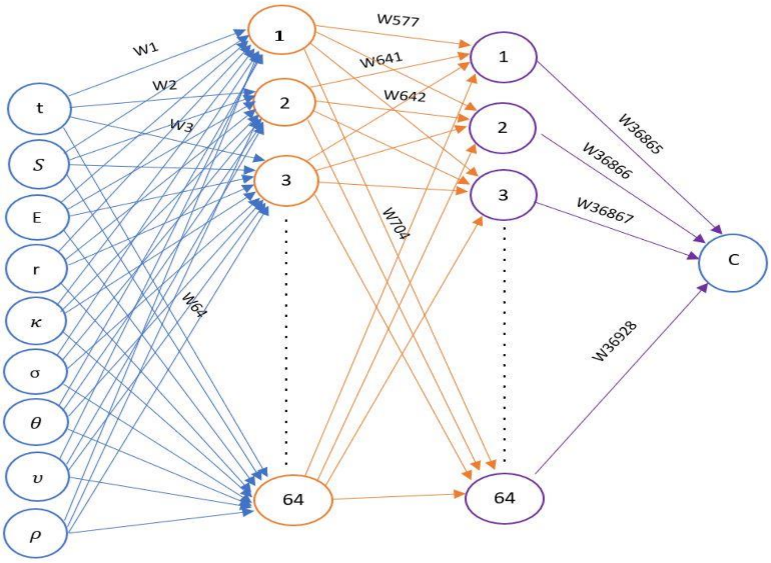
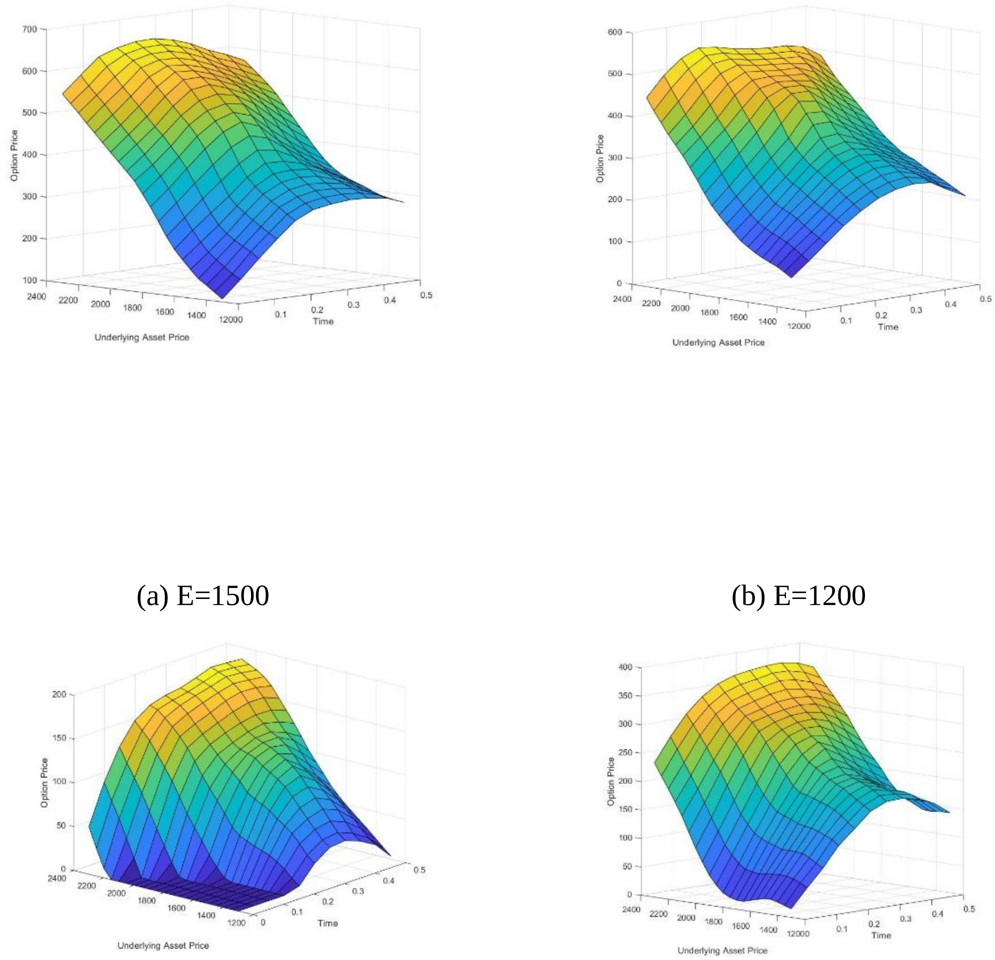


Fig. 5: The considered neural network model

As shown in Figure 5, this network consists of an input layer, 2 hidden layers and an output layer. The input layer, which has 9 nodes, is equal to the number of parameters of the Mikhailov and Nogel model. Also, each hidden layer consists of 64 nodes, which make the calculations very complicated and difficult to make a more accurate prediction. In the last layer, there is a node in which the price of the option contract is placed. Equation (20) was used to calculate the number of hidden layer neurons. That is the total number of data in the training sample (610 data) divided by the total number of neurons in the input and output layer (10 neurons), and the result was 61. (α=1) Then with a little trial and error, it was observed the number 64 is considered the most optimal number for predicting the neural network and numbers greater than that and less than that are less accurate in predicting the price.

In this section, you can see the result of artificial neural network training which was done by 600 data.



(a) E=1500

(b) E=1200

(c) E= 2500

(d) E=2000

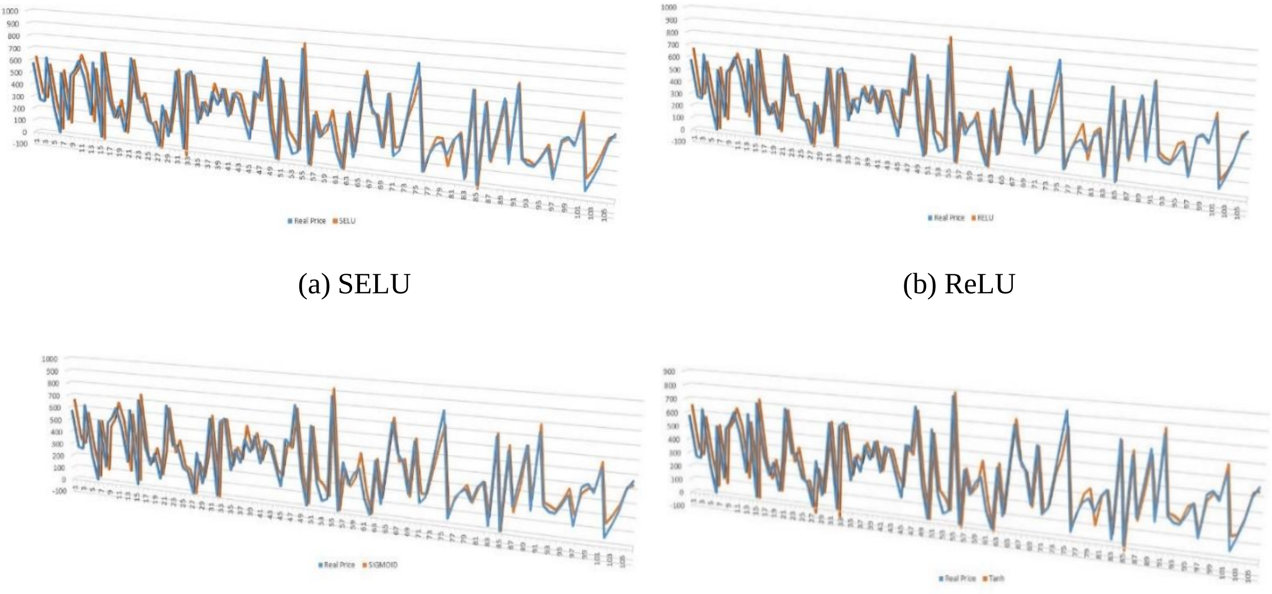
Fig. 6: The learning result by neural networks

Figure 6 shows the impact of the stock price and remaining time to maturity on the option price. Other parameters are assumed constant. The difference between the drawn diagrams is due to the difference in the agreed price mentioned in the contract. In all graphs, the trend of option price changes with respect to time is first increasing and then decreasing. That is, at the beginning of the contract, the option price increases as the expiration time decreases, but after that, the option price also decreases as the expiration time decreases. The change trend of the option price is increasing compared to the underlying stock price. That is, with the increase in the basic stock price, the option price also increases. Of course, it should be noted that as long as the basic stock price is lower than the agreed price, the option price has little value and has little changes, and when the basic stock price is higher than the agreed price, the option price also rises and increases. In the designed artificial neural network system, 4 commonly used activator functions available in artificial neural networks are used, and the training and prediction process is checked by all 4 functions separately. It should be noted that the data used for training and prediction are considered the same for all 4 functions. In the next step, we analyze the performance of these 4 functions to determine which one provides a more accurate forecast and is more suitable for the market.To check the performance of these functions, first, the amount of errors of each function is calculated using the RMSE (Root Mean Square Error) method [10]; (c_i the actual price and q_i the predicted price)

$$e_i = c_i - q_i \, , \tag{21}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \, . \tag{22}$$

The performance results of these functions can be seen in the graphs below, where the prediction of each of these functions is compared with the actual value of the market;



(a) SELU

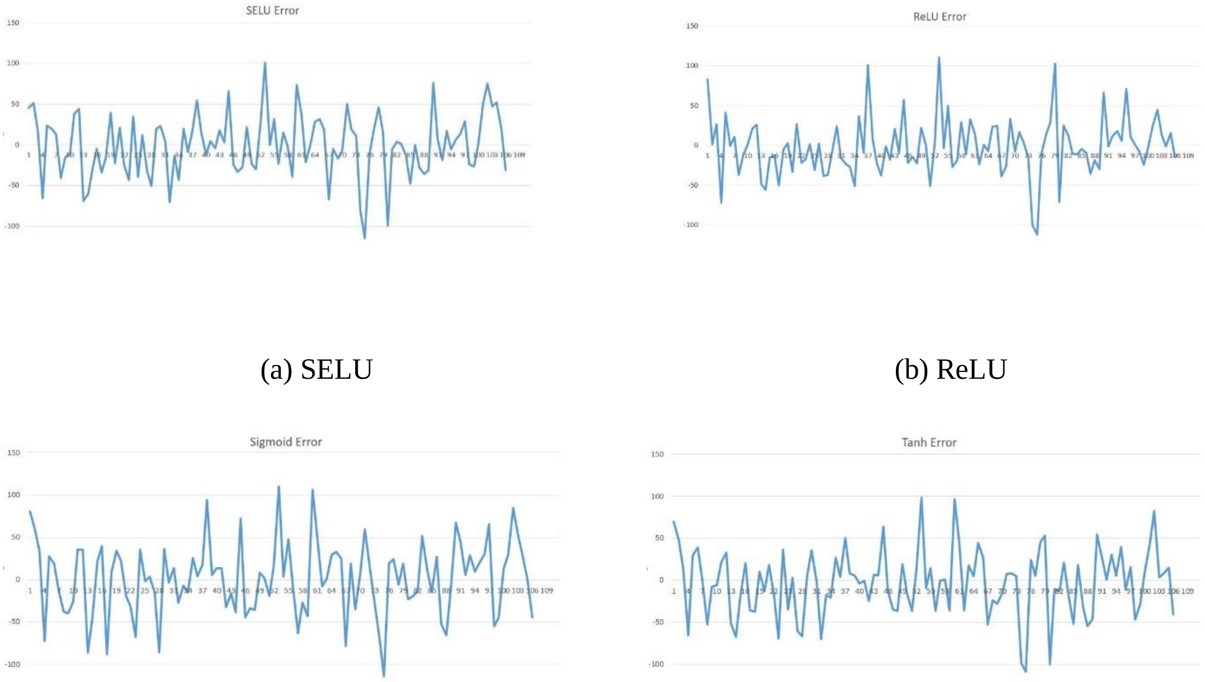
(b) ReLU

(c) Sigmod

(d) Tanh

Fig. 7: The comparison of neural network prediction with actual market price

In Figure 7, you can see the accuracy of the prediction of the activator functions, which are compared with the actual value. Considering that all functions are predicted with high accuracy, it is difficult to distinguish from the figure which function performed better. Therefore, in Figure 8, the error value of these functions is drawn on the graph, which makes the comparison easier.



(a) SELU

(b) ReLU

(c) Sigmoid

(d) Tanh

Fig. 8: The errors of activation functions

According to Figure 8, it can be seen that the ReLU function has less deviation than other functions and its error is less. More details to compare the difference between the used functions and the real price can be seen in Table 2.

Table 2: Comparison of activation functions [Researcher's findings]

Real price	ReLU	Errors	SELU	Errors	Tanh	Errors	Sigmoid	Errors
274	275/5446	1/5446	322/2493	48/2493	300/1324	26/1324	332/9395	58/9395
131	94/3548	36/6451	78/1182	52/8817	95/2562	35/7437	93/3235	37/6765
549	550/1474	1/1474	542/9693	6/0306	540/9936	8/0064	523/3686	25/6314
630	651/477	21/477	651/7065	21/7065	679/0558	49/0558	665/3094	35/3094
484	510/0044	26/0044	516/8265	32/8265	546/3494	62/3494	519/6682	35/6682
191	142/1177	48/8822	139/5809	51/419	125/2397	65/7602	104/8571	86/1429
628	572/829	55/171	560/9797	67/0203	576/7271	51/2729	582/2881	45/7119
711	698/7699	12/23004	731/2762	20/2762	725/7449	14/7449	750/2635	39/2635
192	187/0587	4/9413	154/5762	37/4237	171/0093	20/9906	200/6189	8/6189
292	295/0389	3/0389	302/2323	10/2323	283/6824	8/3175	326/0997	34/0997
86	52/7241	33/2758	73/0421	12/9579	82/7172	3/2827	108/2782	22/2782
373	374/5111	1/51114	409/5271	36/5271	392/4873	19/4873	408/4270	35/4270
200	168/9358	31/0641	165/4225	34/5774	182/0275	17/9725	198/4610	1/5389
174	176/3338	2/3338	176/7212	2/72127	183/1125	9/11258	177/2652	3/26526

As can be seen in table (2), in the first column, the real price of the call option sheets available in the market is given. In the next columns, the value of the price prediction of the same call option sheets made by the activator functions is placed. In front of the prediction of each activation function, its error rate (the difference between the prediction and the actual value of the first column) is also written. In the sequel, using the equation (21) and (22), the total error value of each activator function was calculated (in Table 2, 14 cases among 106 cases are given).

6 Conclusions

In this research using stock option contracts of the Tehran Stock Exchange and the Mikhailov and Nogel model for pricing option, it was possible to provide up to 600 data. It was prepared for learning the artificial neural network and 4 considered activation functions were learned it. Then, the number of 100 option contracts was collected to compare the predictive power of these 4 functions, and using the Mikhailov and Nogel model, the input parameters for these 100 purchase option contracts were calculated and provided to them. Comparing the prediction of these functions with the actual value available in the market, which can be seen in Table 2 and Figure 8 shows the ReLU function has better performance and more accurate prediction.According to the results obtained in this research as well as the results obtained in a research titled " *Neural network models for Bitcoin option pricing* " and another research with the title "*Provide an Improved Factor Pricing Model Using Neural Net-works and the Gray Wolf Optimization Algorithm* " It can be concluded that the pricing on option sheets or common stock sheets using the neural network method and the Relu activator function is much more accurate and faster than the classical methods, and Its use is recommended for financial market participants [11],[17].

References

[1] Björk, T., Arbitrage theory in continuous time, *Oxford university press*, 2009;

[2] Dastranj, E., Sahebi Fard, H., Abdolbaghi, A., Power option pricing under the unstable conditions (Evidence of power option pricing under fractional Heston model in the Iran gold market*), Physica A: Statistical Mechanics and its Applications*, 2020; 537:122690.

[3] Dastranj, E., Sahebi Fard, H., Exact solutions and numerical simulation for Bakstein-Howison model, *Computational Methods for Differential Equations*, 2022; 10(2):461-474.

[4] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., Pineau, J., An introduction to deep reinforcement learning, *Foundations and Trends in Machine Learning*, 2018; 11(3-4): 219-354. doi:10.1561/2200000071

[5] Haykin, S., Neural networks and learning machines, *Pearson Education India*, 2009.

[6] Hull, J., *R*isk management and financial institution*s*, 2012; (Vol. 733). John Wiley & Sons.

[7] Kaustubh, Y., Formulation of a rational option pricing model using artificial neural networks, *S*outheastCon, IEEE, 2021. doi:10.20944\preprints202012.0426.v1

[8] MacBeth, J.D., Merville, L.J., An empirical examination of the Black-Scholes call option pricing model, *The journal of finance*, 1979; 34(5), 1173-1186.

[9] Mikhailov, S., Nogel, U., Heston’s stochastic volatility model: Implementation, calibration and some extensions, *Wilmott magazine*, 2004.

[10] Mitra, S.K., An option pricing model that combines neural network approach and Black-Scholes formula, *Global Journal of Computer Science and Technology*, 2012; 12(4), P. 7-15.

[11] Pagnottoni, P., Neural network models for Bitcoin option pricing, *Frontiers in Artificial Intelligence*, 2019, *2*, 5.

[12] Ren, P., Parametric estimation of the Heston model under the indirect observability framework, *Diss. University of Houston*, 2014.

[13] Sahebi Fard, H., Dastranj, E., Abdolbaghi Ataabadi, A., Analytical and numerical solutions for the pricing of a combination of two financial derivatives in a market under Hull-White model, *Advances in Mathematical Finance and Application*s, 2022; 7(4), 1013-1023.doi: 10.22034/amfa.2021.1902303.1447

[14] Sharma, S., Sharma, S., Athaiya, A., Activation functions in neural networks, *Towards Data Sci*, 2017; 6(12): 310-316.

[15] Shuaiqiang, L., Oosterlee, C.W., Bohte, S.M., Pricing options and computing implied volatilities using neural networks, risks, 2019; 7(1):16.

[16] Vasilev, G.S., *Time-dependent Heston model*, arXiv preprint arXiv, 2014, 1402.5679.

[17] Zohrabi, A., Tehrani, R., Souri, A., & Sadeghi Sharif, S.J., Provide an improved factor pricing model using neural networks and the gray wolf optimization algorithm, *Advances in Mathematical Finance and Applications*, 2023, 8(1).