# Comparing of Three Meta-heuristic Algorithm for Artificial Neural Network Training with Case Study of Stock Price Forecasting

Majid Rostami

Department of Mathematics, Islamic Azad University of Naragh Arak, I.R.Iran

Mehrnaz Piroozbakht

Department of Industrial Engineering, Islamic Azad University of Naragh Arak, I.R.Iran

***Abstract*—Meta-heuristic algorithms, inspired by nature, are used in various scientific fields. One application for these algorithms is neural network training. Neural networks (NNs) are also important in a research such as forecasting. In financial markets, the first and most important factor the investors could confront is stock price; which in turn leads to the subject of evaluation and prediction of future prices. Ability of neural networks in the field of forecasting and the value of prediction in all areas has attracted the researchers towards the way of increasing the accuracy of predictions using neural networks. In this paper, considering the number of neurons in the hidden layer 6, 8, 11, and14, the network performance in predicting stock prices are compared according to these circumstances and in three trained situations with Genetic Algorithm (GA), Particle Swarm Optimizer (PSO) and Birds Mating Optimizer(BMO) algorithms. In all cases, performance is compared based on the Mean Square Error (MSE), and finally, a training algorithm is introduced for more accurate prediction of stock prices using neural networks. In this paper, the network training process is presented in a simpler explanation.**

***Keywords: Neural Network; Neural Network training algorithm; Meta-heuristic optimizer algorithm; Stock price forecasting.***

## I. INTRODUCTION

Artificial Neural Network (ANN) training is one of the major challenges in using a prediction model[1].The viewpoint of training in order to solve the problem and recognition of complicated patterns are really challenging for researchers. NNs are the simple computational tools for modeling with data structures. NNs employ the training data to learn patterns hidden inside the data and use them to get outputs and different results[2].NN training process is an optimization task with the aim of finding a set of weights to minimize an error measure. Owing to this fact that search space is high dimensional and multimodal which is usually polluted by noises and missing data, the problem of NN training needs powerful optimization techniques[3]. Gradient based algorithms are the most frequent training algorithms with several drawbacks[1]. The gradient-based algorithms are susceptible to be converged at local optima, because they are local search methods that the final result depends strongly on the initial weights. If the initial weights are located near local optima, the algorithm would be stuck at them. In contrast with conventional methods, meta-heuristic algorithms do not use any gradient information, and have more chance to avoid local optima by sampling simultaneously multiple regions of search space[3].

Many articles and studies predict the stock price index using neural networks. Among them, “Introducing a prediction model in total stock price index using neural networks approach (Case Study: Tehran Stock Exchange)“[2], “Stock price forecasting with neural network”[4], “stimulating of stock price forecasting through neural network, and comparing them with mathematical prediction methods”[5], “Forecasting of Tehran Stock Exchange with neural network” [6]and “Stock price forecasting of Persian Oil Company using Neural Network and regression model”[7] and so on, could be named.

In all articles mentioned, the issue of" forecasting the stock price for next day" using neural networks, is modeled. When modeling a problem in the form of a neural network, what should be identified, are the number of inputs, the number of outputs, the number of hidden layers, the number of neurons in each hidden layer, type of activation functions for neurons of the network's hidden layer(s) and network's output layer, the values of network weight sand biases (Fig. 1).Independent variables of the problem are identified as the inputs of the model(independent variables in each study mentioned above are different according to the studies conducted by researchers).

The dependent variable of the problem is the "predicted price of the stock for next day" (output of the network). Other characteristics of the network structure (type of activation functions, the number of hidden layers, the number of neurons in each hidden layer), are achieved through trial and error. Weight and bias values in a neural network depend on the training algorithm used to train the neural network.

It should be noted that the neural network accuracy greatly depends on the weights, type of activation functions and the number of neurons in hidden layer. As noted above ,type of activation function, the number of hidden layers and the number of neurons in each hidden layer are obtained through trial and error and the values of biases and weights are dependent on the type of training algorithm. Considering the scope of this paper, investigation of accuracy level of stock price forecasting using neural networks, we focus on neural network training algorithms.

In this paper we investigate the network training and its relevance to the problem of optimization, along with the neural network training through a case study to predict the stock price and to compare the three meta-heuristic learning algorithms.

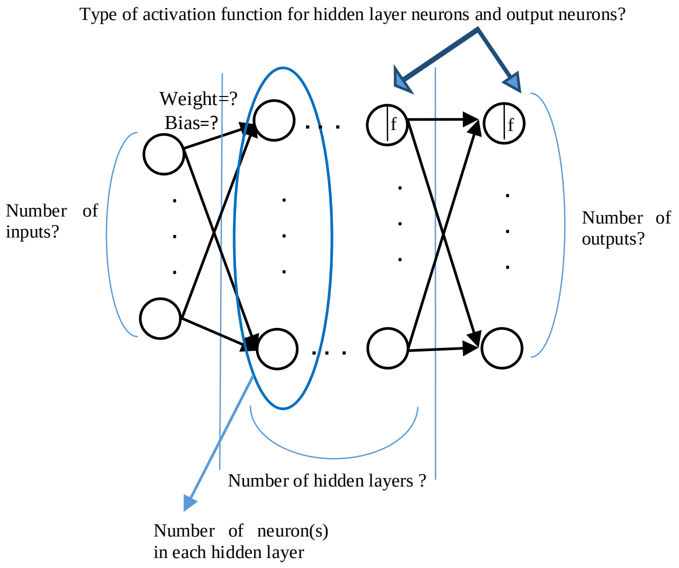



Figure1. Parameters should be determined in order to design an artificial Neural Network for a problem

## II. LITERATURE AND HISTORY

### *A. History of NN Training Algorithm*

Since 2000, a lot of efforts related to NNs training algorithms have been done. Aim of these efforts has been increasing of accuracy ofNN forecasting. This algorithms are divided into three groups: metaheuristic algorithms, hybrid algorithms based on metaheuristic algorithms and gradient based algorithms (in order to resolving problems of gradient based algorithms), and algorithms constructed by combination of two or more than two metaheuristic algorithms.
Researches related to first case:

Kaviani et al., train NN with PSO algorithm[8]. Ragers uses GA algorithm to calculate the weights of NN[9]. Also bayesian method is used for NN training[10].Sinem (2011) and Jonas (2013) present application of Self-adaptive Global Best Harmony Search (SGHS) algorithm for the supervised training of feed-forward neural networks (NNs).Accuracies of SGHS algorithm is compared with five harmony search variants and the standard back-propagation algorithm.

Results show that the proposed algorithm lends itself very well to training of NNs and it is also highly competitive with the compared methods[11, 12].Adam

et al. (2014), perform detailed comparison of the performance of nature-inspired optimization methods and Levenberg–Marquardt (LM) algorithm in ANNs training, based on the case study of water temperature forecasting in a natural stream, namely Biala Tarnowska River in southern Poland. Based on this case study, the only methods that seem competitive to LM algorithm in terms of the final performance (but not speed) are Differential Evolution algorithms that benefit from the concept of Global and Local neighborhood-based mutation operators[13]. Askarzadeh and Rezaadeh (2012) present a novel metaheuristic algorithm. At first, they use it for modeling of proton exchange membrane fuel cell (PEMFC) system[14]. Then, the proposed algorithm, named BMOANN, is employed for NN training and compared with other algorithms. The results are as follows:

CNNE > BMOANN > COOP > GANetbest> SVM-best > CCSS = EDTs > OC1-best > MGNN

Also, they use this algorithm to build an ANN-based model for proton exchange membrane fuel cell (PEMFC) system. Also, this model is trained with PSOand BP algorithms. BMOANN yields better result than the other ANNs[3].

Researches related to second group of NN training algorithms:

Adam (2014), shows a drawback of Differential Evolution algorithms, and clarifies why these methods usually perform poorer than classical Levenberg–Marquardt algorithm. And to tackle this drawback, it is proposed to combine the Differential Evolution with Global and Local Neighborhood-based Mutation Operators algorithm with the Trigonometric mutation operator. The impact of Differential Evolution population size, the initialization range and bounds on Neural Networks performance is also discussed[15].

Yaghini et al. (2013), propose an algorithm that merges the global ability of metaheuristics and the local greedy gradient based algorithm resulting in a superior hybrid method. Opposition based learning and random perturbation help population diversification during the iteration. Effectiveness of this algorithm is compared with several algorithms[1]

Efforts done in relation to third group of algorithms:

Mirjalili et al. (2012), propose a hybrid of PSO and Gravitational Search Algorithm (GSA) to resolve slow searching speed in the last iterations. GSA and PSOGSA are used for NN training and the results are compared with NN trained by PSO. Results show PSOGSA outperforms both PSO and GSA for NN training in terms of converging speed and avoiding local minima and also an NN trained with PSOGSA has better accuracy than one trained with GSA[16].

*B. Bird Mating Optimizer Algorithm (BMO)*

Given the novelty of the algorithm, a brief description is represented. For more explanation refer to Askarzadeh and Rezazadeh, 2013[3, 14].

In this algorithm, there is a population of birds named a society. Members of this society are named as birds. Every bird is a solution for the algorithm. The first population of birds is produced randomly. Every iteration of algorithm is as follows:

Birds are divided into four groups: pa, mg, pg and pr birds. “pa” birds are female and “mg”, “pg”, “pr” birds are male. The presence of each bird in community is known in percentage. Usually, “pa”and“pr” birds include less, and “pg”and“mg” birds are more percentage. At first, we calculate the fitness function of any solution. Then, we order the birds (solutions) according to their fitness values.

The best birds (the birds with most fitness function) are "pa" birds. Then the best of remain birds are respectively "mg" and "pg" birds and the last group is "pr" (worsts of the society). “pr” birds are deleted and replaced by solutions obtained by Eq. (1). In this equation, $z$ is chaos variable and its initial value is a random number between 0 and1 (not the points of 0.25, 0.50 and 0.75). At every iteration, the parameter $z^{gen}$ is firstly to be updated by Eq. (2):

For j=1:d

$$x(\xi,j)= l(j) + z^{gen} *u(j)-l(j)) \quad (1)$$

End

$$z^{gen+1} = 4 * z^{gen} * (1 - z^{gen}) \quad (2)$$

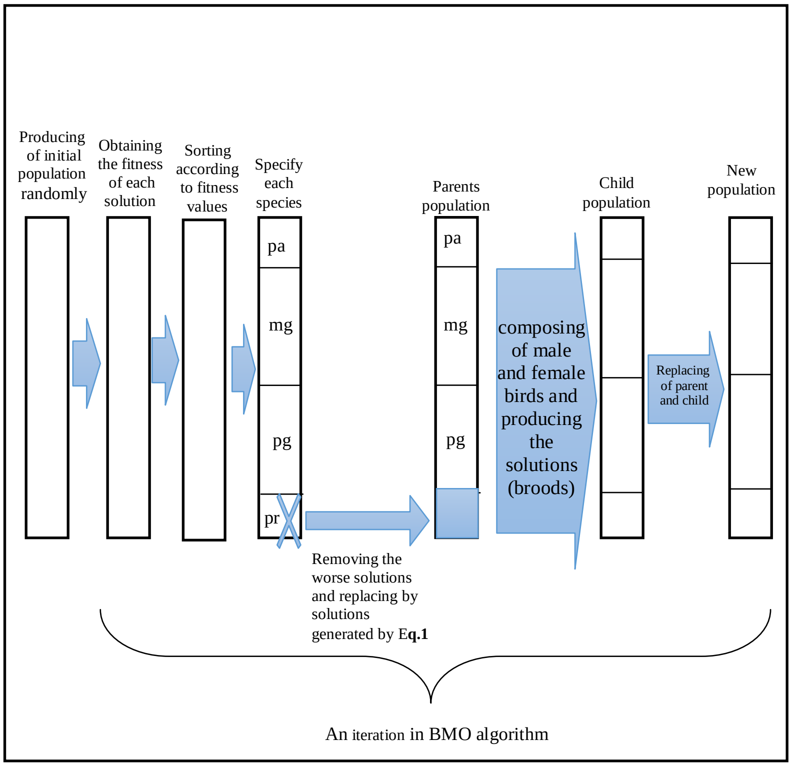



Figure2: Description of a repetition of BMO algorithm

After preparing the parent birds, it is time for brood production.
We show the set of birds ina society by λ, so that $\lambda=\mu \cup \xi \cup \psi \cup \kappa$, where $\mu, \kappa, \psi$ and $\xi$ represent the set of "mg", "pg", "pa",and "pr" birds, respectively. Each bird is shown by a vector$\vec{x}(\lambda)$=(x (λ, 1), x (λ, 2),.., x (λ, d)).
In all formulas(3) to(8), *d* expresses the dimension. *J*, is the variable index for which is the maximum. *W*, expresses the weight in each iteration, with a changing value(according to time) and shows the importance of selected bird.$r_1$,$r_2$ , $r_3$ and$r_4$ are normally distributed random numbers between 0 and 1,$m_w$denotes mutation weight, and *u(j)* and *l(j)* are the upper and lower bounds of variable jth, respectively. Each gene of the brood may be produced by mutation in the bird gene. The probability of mutation is controlled by a factor named mutation control factor, *mcf*, which varies between 0 and 1. This factor helps the algorithm maintains the diversity and avoids premature convergence[3].

1) *Mating of "pa" Bird:*

"Pa" bird mates with one or more males from the best ones ("mg" birds). It's supposed that by mating a "pa" bird with multiple males only one brood is produced. A "pa" bird selects a male by use of an annealing function (3). In order to increase the probability of raising a good brood, a predefined percentage of "mg" birds with better qualities are selected for mating.

$$pr = exp(\frac{-\Delta f}{T}) \qquad (3)$$

Where *pr* is the probability of selecting, $\Delta f$ denotes the absolute differencebetween the objective functions (fitness functions) of the "pa" bird and "mg" one, and *T* is an adjustable parameter to control the probability. Then a random number between 0 and 1is generated and compared with the calculated probability. If it is less than the calculated probability, that "mg" bird is selected formatting. Otherwise, the selection of that male is failed.
After selecting of "mg" bird(s) and mating, brood genes are produced according to (4):

For j=1:d

If $r_1 < mcf$

x(brood,j)= x(ψ,j)+w $*\sum_{i=1}^{n_{em}} r_i *$ (x($em_i$,j)- x(ψ,j))

Else

x(brood,j)=x(ψ,j)+$m_w$*($r_2 - r_3$)*(u(j)-l(j)) (4)

End

End

where$\vec{x}$(ψ) is the "pa" bird, $\vec{x}$($em_i$,j)، is the ith elite male, and $n_{em}$ denotes the number of interesting elite males.

2) *Mating of "mg" Bird:*

"Mg" bird mates just with one female. The bird selects its pair among females,"pa"birds, using the roulette wheel approach (eq.5). In roulette wheel approach, the selection probability of the bird *kth* from a group including *m* birds is defined by formula (5). Based on its selection probability, each female bird is devoted a range between 0 and 1. The female birds with better qualities have wider range than the others.

Then, a random number is uniformly generated between 0 and 1. That range which includes the generated number is specified and the corresponding bird is selected as the interesting bird (pair). Birds with better quality have more chance of being selected.

$$p_k = \frac{1/\text{fit}(\vec{x})}{\sum_{i=1}^{m} 1/\text{fit}(\vec{x})} \quad (5)$$

After selecting a female and mating, the brood genes are obtained according to (6):

For j=1:d

If $r_1 < mcf$

x(brood,j)=x($\mu$,j)+w*$r_2$*(x(ef,j)-x($\mu$,j))

Else (6)

x(brood,j)=x($\mu$,j)+$m_w$*($r_3 - r_4$)*(u(j)-l(j))

End

End

where x(brood,j), x($\mu$), and x(ef) are, respectively, the produced brood, "mg" bird, and interesting female.

3) *Mating of "pg" Bird:*

"Pg" bird mates with one or more female birds. It selects the pairs among "pa" birds through annealing method. The resulted brood genes are obtained through pseudocode (7).

For j=1:d

If $r_1 < mcf$

x(brood,j)= x($\kappa$,j)+w *$\sum_{i=1}^{n_{ef}} r_i$ * (x($ef_i$,j)- x($\kappa$,j))

Else (7)

x(brood,j)=x($\kappa$,j)+$m_w$*($r_2 - r_3$)*(u(j)-l(j))

End

End

Where x($\kappa, j$) and x($ef_i$,j) are, respectively, the $jth$ jenes of "pg" bird and $jth$ jenes of $i$th elite female, $n_{ef}$ denotes the number of elite females, and $r_i$ are normally distributed random numbers between 0 and 1. A "pg" bird combines the information of solutions.

4) *Mating of "pr" bird:*

As mentioned, "pr" birds, are replaced with a set of solutions resulted from eq. (1), at the beginning of the algorithm. Now, the new birds come into mating process. The method of selecting a female bird and mating is the same as "mg" birds. The brood genes are obtained as follows:

For j=1:d

If $r_1 < mcf$

x(brood,j)=x($\xi$,j)+w*$r_2$*(x(ef,j)-x($\xi$,j))

Else (8)

x(brood,j)=x($\xi$,j)+$m_w$*($r_3 - r_4$)*(u(j)-l(j))

End

End

where x($\xi$,j) denotes the "pr" bird.

Next step is replacement. In this step, the bird decides whether to replace the brood or not?If brood has better genes than bird, bird leaves and brood enters the community. Otherwise, bird stays and brood may not enter the community.BMO algorithm pseudo code is shown in Fig. 3.

**Initialization:**
**Determine** the society size, percentage of "mg", "pg", "pr", and "pa" birds, maximum number of generations, and the other parameters
**Do**
**Compute** objective (fitness) function of the birds
**Sort** birds based on their objective function
**Partition** the society into males and females
**Specify** "mg", "pg", and "pa" birds
**Remove** the worst birds and generate "pr" birds based on the chaotic sequence
**Compute** objective function of the "pr" birds
**For** *i*= 1 **to** number of "mg" birds
**Select** interesting bird
**Produce** the brood based on Eq. (6)
**Next** *i*
**For** *i*= 1 **to** number of "pg" birds
**Select** interesting elite birds
**Produce** the brood based on Eq. (7)
**Next** *i*
**For** *I* = 1 **to** number of "pa" birds
**Select** interesting birds
**Produce** the brood based on Eq. (4)
**Next** *i*
**For** *i*= 1 **to** number of "pr" birds
**Select** interesting bird
**Produce** the brood based on Eq. (8)
**Next** *i*
**Compute** objective function of the broods
**Perform** replacement step
**Update** the parameters
**Until** termination criterion is

Figure 3: Pseudocode of BMO algorithm[3]

## III. RESEARCH METHODOLOGY

### A. *NN Architecture Used in This Study*

The architecture of ANN which is used in this article is acquired from Makyan and Mousavi, 2012[7]. It is considered as follows: Number of nodes in the input layer is 4. The inputs are the Brent oil price, maximum price, minimum price, first (opening) price. Closing price is the output of network. Feed forward neural network using in this study, has three layers. According to Makyan and Mousavi, 2012[7], the number of hidden nodes is fourteen. But here, the number of neurons in the hidden layer is considered 6, 8, 11, and 14, and the performance ofthe network is compared according to the sestates training with three methaheuristic algorithm (GA, PSO, BMO).Hidden units employ hyperbolic tangent as their activation function, while output units make use of linear function. If we want to use this model for time series prediction, we will have:

$$P_t = f(P_{max,t-1}, P_{min,t-1}, P_{open,t-1}, P_{oil,t-1}) \qquad (9)$$

where, $P_t$is the closing price at time *t*, and it is defined as output of network. $P_{max,t-1}$ , $P_{min,t-1}$ , $P_{open,t-1}$,denote maximum, minimum and opening stock price of “Oil Industry Investment Company” at time *t-1*. $P_{oil,t-1}$, is the Brent oil price at time*t-1*.

As shown in equation (9), the closing price is considered as the model’s output, and four variables of the maximum price, the minimum price, the opening price, and the Brent oil price are supposed as inputs to the neural network model.

### B. *Input Data and Classification*

Data are on a daily base in the range of seventh September 2013andsixth September 2014(That is, the total number of datais243).Training and test data sets should also be determined. Therefore, the data are randomly divided into two groups of training data and test data. 70% of data is training data, and the remaining30% is the test data (That is, 170 total data are trainingdataand73 data are test data).

Learning in the network is performed through the training data set. The accuracy of the results depends largely on the size of the training set[7].

### C. *NN Training Process*

In order to explain steps in training processof NN, suppose we have a network with structure2-4-1 (Fig.4), and the weights and biases are only unknown values in this network.

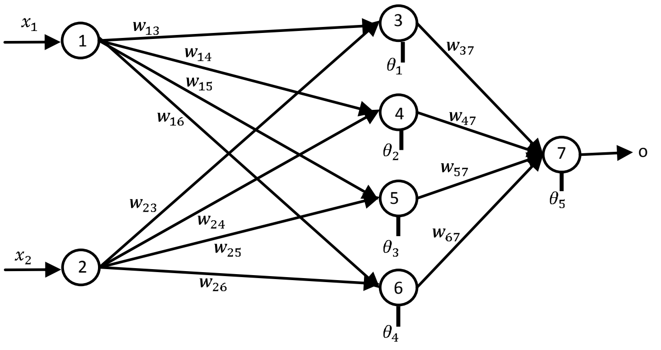



Figure 4.: A network with architecture 2-4-1

In encoding and programming (in this study, written in MATLAB environment), after determining the structure of the neural network, the network without weight and bias is entered into the training phase. All weights and biases in network should be encoded due to enter meta-heuristic algorithm. According to Zhang et al, 2007[17],there are three encoding strategy (vector encoding strategy, matrix encoding strategy and binary encoding strategy)among which the matrix encoding strategy is more suitable for neural networks.

For the network presented in Fig. 4, the weight and bias vectors areas follows:

($w_{67}$, $w_{47}$, $w_{57}$, $w_{13}$,$w_{14}$, $w_{15}$, $w_{16}$, $w_{14}$, $w_{23}$, $w_{24}$, $w_{25}$, $w_{26}$, $w_{37}$, $\theta_1$, $\theta_2$, $\theta_3$, $\theta_4$, $\theta_5$).

This vector is a solution (gene, particle, bird) for meta-heuristic algorithm (Genetic, Particle Swarm Optimizer, Bird Mating Optimizer) that is supposed to train the network.
If we use the matrix encoding strategy, it can be written as:

$$W_1 = \begin{bmatrix} w_{13} & w_{23} \\ w_{14} & w_{24} \\ w_{15} & w_{35} \\ w_{16} & w_{26} \end{bmatrix}, W_2 = [w_{37} \quad w_{47} \quad w_{57} w_{67}]$$

$$, B_1 = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}, B_2 = [\theta_5]$$

where $W_1$ is the hidden layer weight matrix, $W_2$ is the output layer weight matrix, $B_1$ is the hidden layer bias matrix, $B_2$ is the output layer bias matrix.
Training algorithm, at the beginning of its first iteration, produces a random population of weights and biases. It creates a better population than the previous one, at the end of each iteration. As a result, the weights and biases are different in each iteration of training network algorithm. But, how would the algorithm understand that these weight and bias values are optimum for the neural network?

A good criterion for evaluating of the neural network performance is the amount of forecasting error. That is, if $y^k$ is the actual value and $\hat{y}^k$ is the networkoutput value(the predicted value) for the $k^{th}$ training data, the network error will be obtained for the $k^{th}$ data in equation(10): (in this example, $\hat{y}^k$ is the outputof neuronnumber 7inFigure 4 and could be calculated using a mathematical relationship).

$$E_k = \sum_{i=1}^{m} (\hat{y}_i^k - y_i^k)^2 \quad (10)$$

$$MSE = \sum_{k=1}^{q} \frac{E_k}{q} \quad (11)$$

Where, $q$ is the number of training data, $m$is the number of network outputs (in this example it is assumed equal to one). Equation (11) indicates the network output for total training data as well as the fitness value of the best solution vector (gene, particle, bird) in an iteration of meta-heuristic algorithm.
Fitness (bird(i))=MSE

1) *How a neural network is trained?*

Consider one data of company as (($x_1$, $x_2$, $x_3$, $x_4$), y).Each data belongs to one day. As mentioned, the data are on a daily basis in the range of seventh September 2013 and sixth September 2014. 70% of the data is training data. In this research the network chooses the training data accidentally. An artificial neural network, after receiving an input (including variables of the maximum price, the minimum price, the opening price, and the Brent oil price), gives the output(the predicted closing price of the stock for next day) or$\hat{y}$ vector.

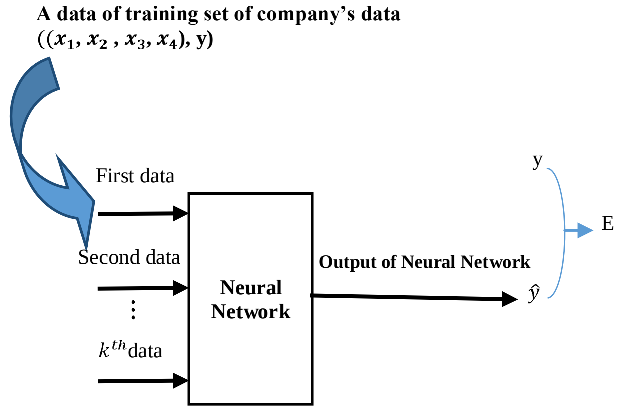



Figure 5. A network with four inputs,($x_1$,$x_2$,$x_3$,$x_4$), and one output, $\hat{y}$

Network error (E), is the difference between the actual value, y, and the predicted value, $\hat{y}$.

In each training iteration, the number of errors, E, is equal to the size of the training set. According to equation (11), the mean square error of these errors is the output of only one iteration of network training algorithm.

In this paper, three training algorithms are used to train the network. Each training algorithm is repeated 50times. The output resulted from the50-th iteration includes an algorithm optimal solution (or final output (final MSEtr)), which is the output of the neural network code's or program's running.

2) *The relationship between the neural network training through algorithm optimization!*

As it has been said, equation (11) is the output of a training iteration. We aim to minimize the amount of this output. Therefore, we face with an optimization problem. We can say that neural network training means to solve an optimization problem in which equation(11) is goal function, and neural network weights and biases are decision variables that are supposed to be optimized. The optimization process in this paper is conducted by meta-heuristic algorithms (GA, PSO and BMO).

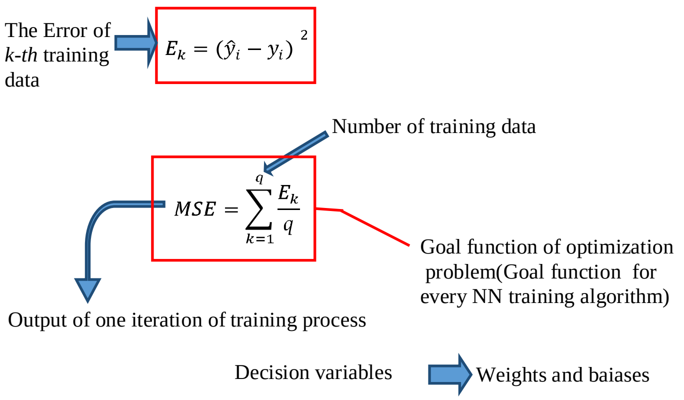



Figure 6. The relation between the neural network training and the optimization problem

### D. *Testing of NN*

Now, the neural network has been completed. Neural network, after the training process, may include a set of trained and optimized weights and biases. This trained network is examined using testing set. At this stage, 30% of the data set, selected accidentally as the testing set at before, will enter network which has specified weights and biases. The network would result in a$\hat{y}$ and also *E*, for each data of testing set. In this paper, the MSEts for 73testingdata would be obtained due to evaluate the network performance.

### E. *Data Collection*

Brent oil price can be acquired from the Eranico Website[18].Maximum price, Minimum price, First (opening) price of Oil Industry Investment Company have been extracted from Tehran Securities Exchange Technology Management Co.[19].Time period in this study is from seventh September 2013 to sixth September 2014. Data has been collected and stored in EXCEL environment.

MATLAB environment is implemented to code FNNGA, FNNPSO and FNNBMO.

### F. *Assumptions and Parameter Values*

In the experiments, parameter setting of FNNBMO, FNNPSO and FNNGA algorithms is as follows:
For FNNGA, percentages of reproduction and crossover are 15% and 50%, respectively. It is assumed that every chromosome is randomly initialized. Parent selection strategy using in this algorithm is Imperialist Competitive Algorithm (ICA).

Parameter setting of FNNBMO algorithm is according to the (Askarzadeh and Rezazadeh,2013); “Mg”, “pg”,”pa”, and “pr” birds make 50%, 40%, 5%, and 5% of the society, respectively; T, w, and $m_w$are defined as decreasing linear functions where $T_{max}$ = 300,$T_{min}$ = 50, $w_{max}$ = 2.5, $w_{min}$ = 0.5, $m_{w,max}$ = 0.01, and $m_{w,min}$ = 0.0001; *mcf* is selected 0.9.
For FNNPSO, $c_1$ and $c_2$ are set to 2, $r_1$ and $r_2$ are two random numbers in the interval [0,1], w is equal to 1, and the initial velocities of particles are randomly generated in the interval [0,1].
Minimum and maximum values of variables in each algorithm are set to -1 and 1, respectively. It should be noted that the parameter setting is based on trial and error and no attempt has made to optimize it. For two algorithms, the society size is set to 100 and Maximum number of generations (epochs) is set to 50.

Due to the fact that the nature of metaheuristic algorithms is stochastic, the results obtained in one attempt will differ from the results obtained in another attempt. Therefore, the performance analysis must be statistically based. Results of FNNGA, FNNPSO and FNNBMO are compared based on average, median and standard deviation of the Mean Square Error (MSE) for training set and testing set over 50 independent runs. Termination criterion of NN training is maximum epochs.

## IV. RESULT AND DISCUSSION

Table 1 shows the percent values of minimum, maximum, average, standard deviation for the Mean Square Error of testing set and training set for three algorithm. These values are in percentage.
As can be seen from Table1, for the three algorithms, MSEts for8 neurons shows better values. In Table 2, the yield of three training algorithm, with the best network architecture (8 neurons in the middle layer), are compared based on the average of "Mean Square Error" of test data. According to the table:

FNNPSO>FNNBMO>FNNGA

TABLE II.COMPARISON OF THREE NNTRAINING ALGORITHM IN TERMS OF AVERAGE TESTING ERROR RATE(%) ON STOCK PRICE FORECASTING PROBLEM

| Algorithm | FNNGA | FNNBMO | FNNPSO |
|---|---|---|---|
| Test error rate(%) | 4.21 | 2.47 | 2.44 |

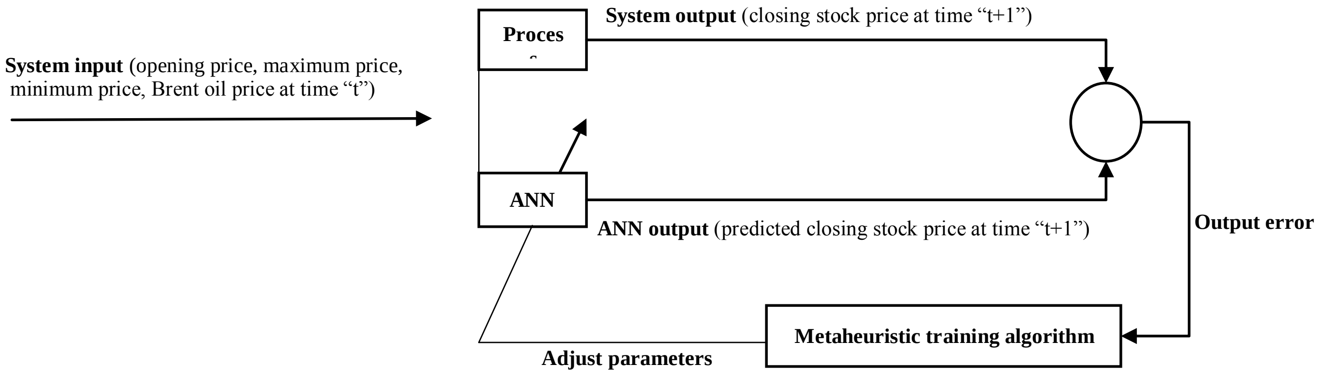
## V. CONCLUSION

The aim of this article is to compare three training neural network algorithms through the case study of stock price forecasting. As previously mentioned, from 2010 to now, new and different efforts have been done in area of NN training algorithms. Examination of accuracy of these algorithms in forecasting with NN, for example stock price forecasting, to access a more accurate forecast, is a good research scope. Hybrid of mentioned training algorithms in order to achieve a new algorithm which has better convergence speed and can escape from local optimum traps, is an interesting research domain.

TABLE I.AVERAGE, MEDIAN, STANDARD DEVIATION, AND BEST OF MSE FOR ALL TESTING SET AND TRAINING SET OVER 50 INDEPENDENT RUNS FOR FNNGA, FNNBMO, FNNPSO IN ORDER TO MODELING OF STOCK PRICE FORECASTING

| Algorithm | Number of hidden layer neuran | Training set | | | | | Testing set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Ave | Std. | Mid | Min | Max | Ave | Std. | Mid |
| FNNPSO | 6 | 1.02 | 3.52 | 1.91 | 0.58 | 1.82 | 0.96 | 9.56 | 3.34 | 2.34 | 2.44 |
| | 8 | 0.93 | 3.55 | **1.88** | 0.71 | 1.7 | 0.72 | 7.51 | **2.44** | 1.56 | 1.99 |
| | 11 | 1.04 | 4.42 | 2.2 | 0.83 | 2.02 | 0.82 | 7.76 | 2.72 | 1.6 | 2.25 |
| | 14 | 0.8 | 4.67 | 2.11 | 0.86 | 2.12 | 0.8 | 10.82 | 3.32 | 2.18 | 2.59 |
| FNNBMO | 6 | 1.15 | 4.54 | 2.25 | 0.79 | 2 | 0.69 | 13.66 | 3.7 | 2.96 | 2.46 |
| | 8 | 0.93 | 4.06 | 2.3 | 0.77 | 2.18 | 0.6 | 8.03 | **2.47** | 1.71 | 1.87 |
| | 11 | 0.93 | 4.88 | 2.33 | 0.96 | 2.26 | 0.72 | 8.5 | 2.85 | 2.04 | 2.07 |
| | 14 | 1 | 4.56 | **2.2** | 0.94 | 2.23 | 0.64 | 10.32 | 2.93 | 2.17 | 2.34 |
| FNNGA | 6 | 2.23 | 5.1 | 3.7 | 0.82 | 3.67 | 2.14 | 8.48 | 5 | 1.59 | 4.49 |
| | 8 | 2.09 | 5.05 | **3.56** | 0.82 | 3.4 | 3.4 | 10.49 | **4.21** | 1.59 | 4.01 |
| | 11 | 2.08 | 5.48 | 3.97 | 0.87 | 4.12 | 1.52 | 8.28 | 4.21 | 1.43 | 3.93 |
| | 14 | 2.45 | 5.51 | 4.13 | 0.89 | 4.16 | 1.74 | 9.48 | 4.73 | 1.75 | 4.47 |

SCHMENTIC DIAGRAM




REFRENCES


1. Yaghini, M., M.M. Khoshraftar, and M. Fallahi, *A hybrid algorithm for artificial neural network training.* Engineering Applications of Artificial Intelligence, 2013. **26**(1): p. 293-301.
2. Pakdin Amiri , A., M. Pakdin Amiri , and M. Pakdin Amiri *Introdusing a prediction Model in Total Stock Price Index Using Neural Network Aproach(Case Study :Tehran Stock Index).* journal of iran's Economic Essays, 2009. **6**(11): p. 83-108.
3. Askarzadeh, A. and A. Rezazadeh, *Artificial neural network training using a new efficient optimization algorithm.* Applied Soft Computing, 2013. **13**(2): p. 1206-1213.
4. Heidari Zare, B. and H. kordlouie *stock price forecasting with Neural Network.* PAJOUHESHGAR (JOURNAL OF MANAGEMENT), 2010. **7**(17): p. 49-56.
5. Toloie Ashlaghi, A. and S. Haghdoost, *stimulating of stock price forecasting through neural network, and comparing them with mathematical prediction methods.* Economic Bulletin, 2007. **8**(25): p. 237-252.
6. Sinaei, H.A., S. Mortazavi, and y. Teimouri Asl *Forecasting of Tehran Stock Exchange with Neural Network.* journal of the accounting and auditing review2005. **12**(3).
7. Makyan , S.N. and F.s. Mousavi *Stock price forecasting for Pars Oil Company using neural network and regression (Case study: Pars Oil Company).* Quarterly Journal of Economic Modelling, 2012. **2**(18).
8. Kashefi Kaviani , A., S.a. Poormousavi Kani, and A. Jahanbani Ardakani. *Multilayer neural network training using PSO algorithm.* in *First Joint Congress on Fuzzy and Intelligent systems* 29 August to 31 August 2007. Iran.
9. Rogers, L.L., F.U. Dowla, and V.M. Johnson, *Optimal field-scale groundwater remediation using neural networks and the genetic algorithm.* Environmental Science & Technology, 1995. **29**(5): p. 1145-1155.
10. Kumar, P., S. Merchant, and U.B. Desai, *Improving performance in pulse radar detection using Bayesian regularization for neural network training.* Digital signal processing, 2004. **14**(5): p. 438-448.
11. Kalderstam, J., et al., *Training artificial neural networks directly on the concordance index for censored data using genetic algorithms.* Artificial intelligence in medicine, 2013. **58**(2): p. 125-132.
12. Kulluk, S., L. Ozbakir, and A. Baykasoglu, *Self-adaptive global best harmony search algorithm for training neural networks.* Procedia Computer Science, 2011. **3**: p. 282-286.
13. Piotrowski, A.P., et al., *Comparing large number of metaheuristics for artificial neural networks training to predict water temperature in a natural river.* Computers & Geosciences, 2014. **64**: p. 136-151.
14. Askarzadeh, A. and A. Rezazadeh, *A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer.* International Journal of Energy Research, 2013. **37**(10): p. 1196-1204.
15. Piotrowski, A.P., *Differential Evolution algorithms applied to Neural Network training suffer from stagnation.* Applied Soft Computing, 2014. **21**: p. 382-406.16. Mirjalili, S., S.Z.M. Hashim, and H.M. Sardroudi, *Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm.* Applied Mathematics and Computation, 2012. **218**(22): p. 11125-11137.
17. Zhang, J.-R., et al., *A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training.* Applied Mathematics and Computation, 2007. **185**(2): p. 1026-1037.
18. *Brent oil price.* Available from: http://www.eranico.com/fa/oil/167/1.
19. *Tehran securities Exchange Technology Management.* Available from: http://en.tsetmc.com/Site.aspx?ParTree=121C.