

THE TRAINING OF MULTIPLICATIVE NEURON MODEL BASED ARTIFICIAL NEURAL NETWORKS WITH DIFFERENTIAL EVOLUTION ALGORITHM FOR FORECASTING

Eren Bas

*Department of Statistics, Giresun University,
Giresun, 28100, Giresun, Turkey*

Abstract

In recent years, artificial neural networks have been commonly used for time series forecasting by researchers from various fields. There are some types of artificial neural networks and feed forward artificial neural networks model is one of them. Although feed forward artificial neural networks gives successful forecasting results they have a basic problem. This problem is architecture selection problem. In order to eliminate this problem, Yadav et al. (2007) proposed multiplicative neuron model artificial neural network. In this study, differential evolution algorithm is proposed for the training of multiplicative neuron model for forecasting. The proposed method is applied to two well-known different real world time series data.

Keywords: Artificial neural networks; multiplicative neuron model; differential evolution algorithm; forecasting

1 Introduction

There are many models proposed in the literature on the forecasting problem. Conventional forecasting models, one of these models, may be inadequate to solve many data due to various assumptions that they require. In this respect, artificial neural networks (ANN) that do not include the assumptions valid for conventional forecasting models are widely used as an alternative to conventional time series approaches.

ANN is generally defined as mathematical algorithms inspired by biological neural networks and can learn from the samples and generalize them. It can be said that the most important feature of ANN is the ability to learn from a source of information (data). Learning process of ANN is the process of finding the best values of weights and

this process is called as the training of ANN. Finding the best weights in ANN can be considered as an optimization problem. The most used learning algorithms are Levenberg-Marquardt and Back Propagation (BP) learning algorithms. However, the other methods used in the training of ANN are heuristic algorithms such as genetic algorithms, particle swarm optimization, simulated annealing and taboo search algorithm. The first ANN model was proposed by McCulloch and Pitts [1]. The studies on ANN began with a single-layer neural network. Single-layer neural networks consist of only the input and output layers and the output function of single-layer neural networks are linear.

However, it was not possible to solve a non-linear problem with single-layer neural networks. In later years multilayer perceptron (MLP) model was developed by Rumelhart et al. [2] in order to

solve a non-linear problem. A MLP model consists of an input, one or more hidden and an output layer. The main objective of the MLP model is to reduce the difference between the output produced by the network and with the expected output of the network (error). In later years, several ANN models were proposed by Basu and Ho [3], Labib [4], Plate [5] and Zhang et al. [6]. In addition to these types of ANN, there are some ANN models in the literature such as generalized mean neuron model (GMN) proposed by Yadav et al. [7] and geometric mean neuron (G-MN) model proposed by Shiblee et al. [8].

Multi-layer feed forward artificial neural networks (ML-FF-ANN) model is the most popular type of artificial neural networks among all of these methods. ML-FF-ANN is generally used for forecasting problem in time series analysis Aladag et al. [9]. The neural network forecasting literature was also summarized by Zhang et al. [10]. And also, it is well known that one major application area of ANNs is forecasting. Sharda [11], Weigend et al. [12], Weigend et al. [13] and Cottrell et al. [14] addressed the issue of network structure for forecasting real-world time series.

The determination of the numbers of neurons in the layers in ML-FF-ANN is an important problem. This problem is also called as architecture selection problem. In order to eliminate this problem, Yadav et al. [15] proposed multiplicative neuron model based artificial neural network (MNM-ANN) that this neuron model has a single neuron and it does not have architecture selection problem. While Yadav et al. [15] used back propagation learning algorithm, Zhao and Yang [16] used cooperative random particle swarm optimization (CRPSO) in the learning process of MNM-ANN. Therefore, the modified particle swarm optimization method proposed by Aladag et al. [17] is utilized to train MNM-ANN.

In this study, a new learning algorithm based on differential evolution algorithm for the training of MNM-ANN (MNM-ANN-DEA) is proposed. And also, the proposed method was supported by two well-known different real-life time series data and its superior forecasting performance was shown.

The remainder of this paper is organized as follows: short information about multiplicative neuron model (MNM) is given in Section 2. DEA is briefly summarized in the Section 3. The proposed method

is introduced in Section 4. Under the main title of the applications in the fifth section, the application of the proposed method was examined by considering two different real life time series data. Finally, Section 6 presents conclusions and discussions.

2 Multiplicative Neuron Model

The single multiplicative artificial neuron model which is a special case of neuron model was first presented by Yadav et al. [15]. This model has only one neuron and the general structure of the model, which includes only five inputs, is illustrated in Figure 1.

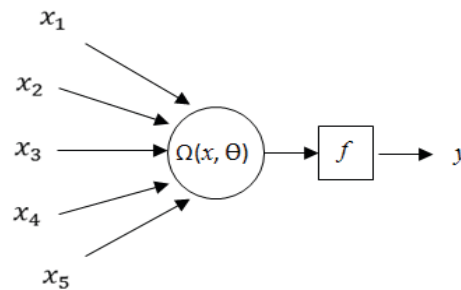


Figure 1. Multiplicative neuron model

In this model, $x_i (i = 1, \dots, 5)$ is input pattern and when output values are computed by using multiplication function instead of sum function.

The operator $\Omega(x, \theta)$ is a multiplicative operation which is composed of multiplication of weighted inputs. Activation function and target of the model are represented by f and y , respectively. In the literature, for training the multiplicative neuron model, there are several learning algorithms. BP and cooperative particle swarm optimization learning algorithms were used in the process of multiplicative neuron model by Yadav et al. [15] and Zhao and Yang [16], respectively.

3 Differential Evolution Algorithm

DEA was proposed by Price and Storn [18]. DEA is a heuristic algorithm based on the population. It is similar with genetic algorithm in terms of operators such as mutation and cross-over operators. These operators are also used in the process of DEA, respectively. All these operators are applied to all chromosomes in the population in DEA and a new

chromosome is obtained. After using the mutation and cross-over operations the interested chromosome and the new chromosome are compared with a determined fitness value. Chromosome with a better fitness value is transferred to a new generation according to the objective function. The best chromosome is taken as the optimal solution at the end of the process of DEA. This information should be found enough in this section because DEA is discussed in more detail in the section of proposed method. Those who want more information can look at the study of Price and Storn [18].

4 The Proposed Method

In the literature, there are many ANN methods for forecasting of time series. One of these methods is ML-FF-ANN. Besides, ML-FF-ANN has some problems. One of the most important problems in ML-FF-ANN is architecture selection problem. In order to eliminate this type of problem Yadav et al. [15] proposed MNM-ANN. There are some learning algorithms proposed in the literature for MNM-ANN to improve its performance.

In this study, a new learning algorithm based on DEA for MNM-ANN called MNM-ANN-DEA is proposed, firstly.

The algorithm of the proposed method is given step by step as below.

Algorithm

Step 1. Determine the parameters of DEA

The parameters used in the structure of DEA are determined. These are the number of chromosomes (cn), gene numbers (gn) and cross-over ratio (cor).

Step 2. Generate the initial population

A chromosome structure for the proposed learning algorithm is given in Figure 2.

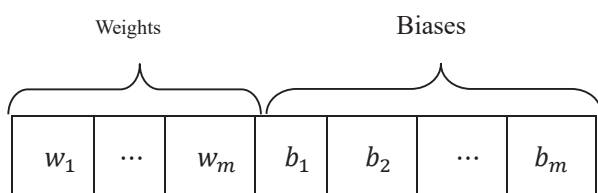


Figure 2. The structure of a chromosome

In the Figure 2, the first m gene represents the weights and the second m gene represents the biases. Totally, there is $(2 \times m)$ gene in a chromosome. For the generation of the initial population of chromosomes, random numbers are generated from the continuous uniform distribution. $(2 \times m)$ random numbers are generated from $U(0, 1)$ distribution for the $(2 \times m)$ genes in a chromosome.

Step 3. Calculate the fitness function value for each chromosome.

The calculation of the fitness function value for the chromosomes is given by the steps between 3.1 and Step 3.3 when the number of iterations is k .

Step 3.1. The weights and biases of MNM-ANN-DEA are obtained from $(2 \times m)$ genes of the chromosomes i at the iteration k . The formulas of weights and biases of MNM-ANN-DEA were given in Equations 1 and 2, respectively.

$$w_1 = x_{i,1}^k, w_2 = x_{i,2}^k, \dots, w_m = x_{i,m}^k \quad (1)$$

$$b_1 = x_{i,m+1}^k, b_2 = x_{i,m+2}^k, \dots, b_m = x_{i,2m}^k \quad (2)$$

In these Equations, $x_{i,m}^k$ represents the m^{th} gene of the chromosome i at the iteration k .

Step 3.2. The net values for each learning sample of MNM-ANN-DEA are calculated like in Equation 3 by using weights and bias values.

$$net_t = \prod_{j=1}^m w_j y_{t-j} + b_j, t = 1, \dots, n \quad (3)$$

In this equation n is the number of learning samples. If time series has N observations n is equal to $N - m$ ($n = N - m$). The outputs of the network are calculated as in Equation 4.

$$o_t = \frac{1}{1 + \exp(-net_t)}, t = 1, 2, \dots, n \quad (4)$$

Step 3.3. The fitness function value which is Root Mean Square Error ($RMSE$) criteria given in Equation 5 of each chromosome is calculated by using the output values.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{n}} \quad (5)$$

Step 4. Mutation and Crossover operations

Mutation and Crossover operations are applied for each chromosome in the initial population, respectively.

Step 4.1. The application of Mutation operation

For applying mutation operation in DEA, firstly four chromosomes are selected. The first one of these four chromosomes is called as current chromosome and the remaining three chromosomes are selected randomly except current chromosome.

The first two of these selected three chromosomes are subtracted each other and it is called as the difference vector. Then, difference vector is multiply by F and a new chromosome is obtained (In general the parameter F gets values between 0 and 2. We take this F value as 0.8 which is general value in the literature in this study).

This new chromosome is called as the weighted difference vector. The weighted difference vector is summed with the last of these selected three chromosomes and mutation operation is completed. This new created chromosome is called as the total vector. Thus, the chromosome to be used in the crossover operation is created with the help of mutation operation.

Step 4.2. The application of Crossover

To apply crossover operation in DEA, the total vector obtained at the end of the mutation operation is compared with current chromosome and nominee chromosome is obtained. While nominee chromosome is obtained, each gene of total vector and current chromosome is evaluated one by one. First at all, a crossover rate (cor) is determined. Then, a random number is generated from between 0 and 1 with the help of uniform distribution. If this random number is smaller than the crossover rate, the gene is taken from total vector. If it is not, the gene is taken from current chromosome and nominee chromosome is generated and the fitness value of nominee chromosome is calculated.

Step 5. The Comparison of fitness values

Nominee chromosome and interested chromosome are compared in terms of fitness values. That is, the chromosome which has the smaller RMSE value used as evaluation function is transferred to a new generation.

Step 6. Stopping criteria

If it is reached to the maximum number of iterations or the fitness value calculated from the chromosome with the best fitness value is less than a predetermined value (ϵ) the process is end, otherwise move to Step 3. If the algorithm is stopped, the optimal weight and bias values are taken from the chromosome with the best fitness value.

5 Applications

To show the performance of the MNM-ANN-DEA, the proposed method is applied to two real life time series data. The first time series data is Australian Beer Consumption “(AUST)” data with 148 observations between years 1956 and 1994 and the second one is Canadian lynx data between years 1821 and 1934. The performance of the proposed method was evaluated with several methods suggested in the literature. For this purpose, RMSE criteria given in Equation 5 and Mean Absolute Percentage Error (MAPE) criteria given in Equation 6 were used to compare the methods.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (6)$$

In this Equation, n , y_t and \hat{y}_t show the number of training samples, observed values and the forecasting values.

5.1 Analysis with Australian beer consumption data

The proposed method was firstly applied to the Australian beer consumption data (AUST) time series data with 148 observations between the years 1956 and 1994 which is shown in Figure 3. The last 16 observations of the time series was taken as test data. In addition to the proposed method, AUST data, which is shown in Figure 3, is examined by using seasonal autoregressive integrated moving average (SARIMA), Winter’s multiplicative exponen-

tial smoothing (WMES), Multi-layer feed-forward neural network (ML-FF-ANN), Multilayer neural network based on particle swarm optimization (ML-PSO-ANN), Back propagation learning algorithm based on multiplicative neuron model neural network (BP-MNM-ANN), multiplicative neuron model artificial neural network based on particle swarm optimization (PSO-MNM-ANN), Radial basis artificial neural network (RB-ANN), Elman neural network (E-ANN) methods.

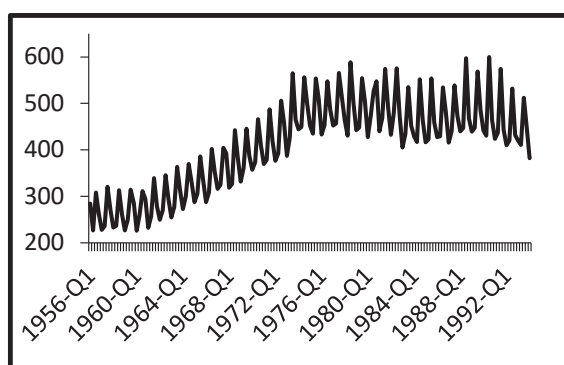


Figure 3. AUST data between the years of 1956-1994

During the analysis of AUST time series data,

- The model orders in other words the number of inputs of MNM-ANN-DEA (m) was changed from 4 to 8.
- cn is experimented as from 10 to 100 with increment 10
 - cor is experimented respectively 0.1 to 1 with increment 0.1
 - For all possible case, DEA is executed 100 times in MATLAB.

At the end of the process, we obtained 500 different solutions. Then the parameters (m , cn , cor) with the smallest $RMSE$ value were taken as the best solution among these solutions.

We conclude that the best result is obtained in the case where $m=5$, $cn=70$, $cor=0.2$.

In addition, $RMSE$ and $MAPE$ criteria values for test set obtained from proposed method and other methods in the literature were given in Table 1.

It is clearly seen in Tab. 1, that the proposed method has the best performance among all methods in terms of both $RMSE$ and $MAPE$ criteria.

Besides, the graph of the real observations and the forecasts obtained from proposed method for test set was given in Figure 4. According to this graph, the forecasts obtained from the proposed approach are very accurate.

Table 1. The $RMSE$ and $MAPE$ values obtained from all methods for original AUST data

Method	$RMSE$	$MAPE$
BP-MNM-ANN	74.2551	0.0983
WMES	53.3295	0.1072
SARIMA	47.0367	0.0949
ML-PSO-ANN	44.7780	0.0856
RB-ANN	41.7000	0.0686
PSO-MNM-ANN	26.7831	0.0498
ML-FF-ANN	24.1052	0.0476
E-ANN	22.6581	0.0436
Proposed Method	19.7819	0.0372

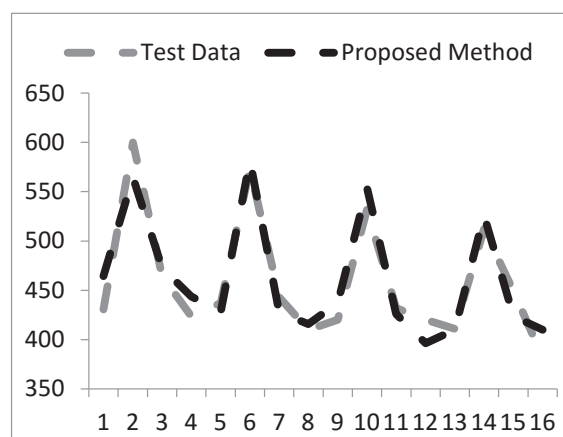


Figure 4. The graph of real observations and forecasts obtained from proposed method for AUST data

5.2 Analysis with Canadian lynx data

The proposed method is also applied to Canadian lynx data consisting of the set of annual numbers of lynx trappings in the Mackenzie River District of North-West Canada for the period from 1821 to 1934. The last 14 observations of the time series was taken as test data. This data has also been extensively analyzed in the time series literature. We use the logarithm (to the base 10) of the data in the analysis. In addition to the proposed approach, logarithm of Canada lynx data, which is shown in Figure 5, is examined by Autoregressive integrated moving average (ARIMA), WMES, ML-FF-ANN,

RB-ANN, Zhang [19], Aladag et al. [20] and Yolcu et al. [21].

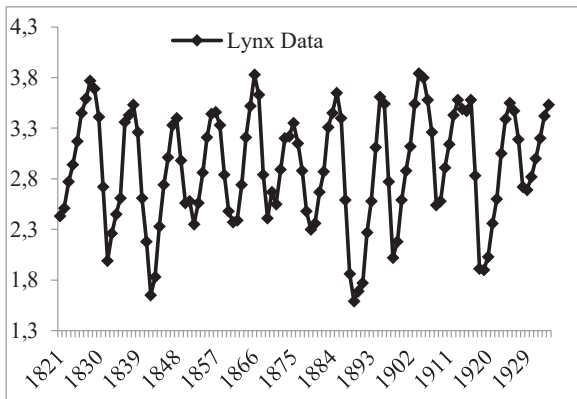


Figure 5. Logarithmic Canadian lynx data series (1821-1934)

During the analysis of Canadian lynx data,

- The model orders in other words the number of inputs of MNM-ANN-DEA (m) was changed from 1 to 4.
- cn is experimented as from 10 to 100 with increment 10
- cor is experimented respectively 0.1 to 1 with increment 0.1
- For all possible case, DEA is executed 100 times in MATLAB.

Table 2. The forecasting results obtained from the proposed method for Logarithmic Canadian Lynx Data

Method	RMSE	MAPE
ARIMA	0.1261	0.0312
WMES	0.1568	0.0469
[21]	0.0826	0.0217
RB-ANN	0.2170	0.0636
[20]	0.0970	0.0251
ML-FF-ANN	0.1423	0.0376
[19]	0.1324	0.0350
Proposed Method	0.0814	0.0238

At the end of the process, we obtained 500 different solutions. Then the parameters (m , cn , cor) with the smallest $RMSE$ value were taken as the best solution among these solutions.

We conclude that the best result is obtained in the case where $m=3$, $cn=90$, $cor=0.7$.

In addition, $RMSE$ and $MAPE$ criteria values for test set obtained from proposed method and other methods in the literature were given in Table 2.

It is clearly seen in Table 2, the proposed method has the best performance among all methods in terms of both $RMSE$ and $MAPE$ criteria.

Besides, the graph of the real observations and the forecasts obtained from proposed method for test set was given in Figure 6. According to this graph, the forecasts obtained from the proposed approach are very accurate.

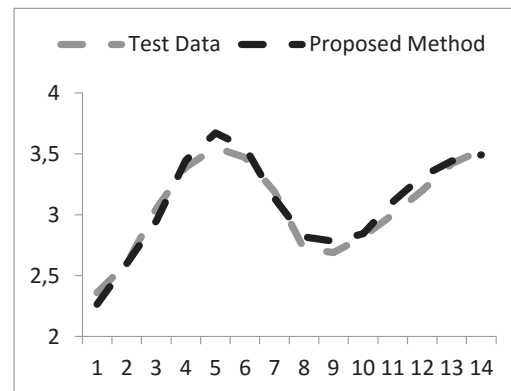


Figure 6. The graph of real observations and forecasts obtained from proposed method for Canadian lynx data

6 Conclusion and Discussion

In the literature, different types of ANN have been used in various applications. Especially, ML-FF-ANN has been widely used in the forecasting problem of time series. Although ML-FF-ANN gives successful forecasting results and uses in time series problems it has also some basic problems. One of these problems is the architecture selection problem which plays an active role in the forecasting performance of the network. There are some learning algorithms in the literature for MNM-ANN used in order to eliminate this problem.

In this paper, a new learning algorithm called MNM-ANN-DEA was proposed for MNM-ANN. In the proposed method, the training of MNM-ANN was performed by DEA, firstly. And also, the proposed method was applied to two well-known real life time series data and its superior forecasting performance was proved.

In the future studies, different artificial intelligence techniques can be used for the training

of MNM-ANN to improve the forecasting performance.

References

- [1] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5, 1943, 115–133.
- [2] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Representations by Backpropagating Errors, *Nature*, 323, 1986, 533–536.
- [3] M. Basu, T.K. Ho, Learning behavior of single neuron classifiers online linearly separable or nonseparable inputs, In *IEEE LICNN'99*, 1999.
- [4] R. Labib, New single neuron structure for solving non-linear problems, In *IEEE IJCNN'99*, 1999, 617–620.
- [5] T.A. Plate, Randomly connected sigma-pi neurons can form associator networks, *NETCNS: Network: Computation in Neural Systems*, 11, 2000, 321–322.
- [6] C.N. Zhang, M. Zhao, M. Wang, Logic operations based on single neuron rational model, *IEEE Transactions on Neural Networks*, 11, 2000, 739–747.
- [7] R.N. Yadav, N. Kumar, P.K. Kalra, J. John, Learning with generalized-mean neuron model. *Neurocomputing*, 69, 2006, 2026–2032.
- [8] M. Shiblee, B. Chandra, P.K. Kalra, Learning of geometric mean neuron model using resilient propagation algorithm, *Expert Systems with Applications*, 37, 2010, 7449–7455.
- [9] C.H. Aladag, E. Egrioglu, U. Yolcu, Forecast combination by using artificial neural networks, *Neural Processing Letters*, 32 (3), 2010, 269–276.
- [10] G. Zhang, B.E. Patuwo, Y.M. Hu, Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14, 1998, 35–62.
- [11] R. Sharda, Neural networks for the MS/OR analyst: An application bibliography, *Interfaces*, 24 (2), 1994, 116–130.
- [12] A.S. Weigend, B.A. Huberman, D.E. Rumelhart, Predicting the future: A connectionist approach, *International Journal of Neural Systems*, 1, 1990, 193–209.
- [13] A.S. Weigend, B.A. Huberman, D.E. Rumelhart, Predicting sunspots and exchange rates with connectionist networks. In: Casdagli, M., Eubank, S. (Eds.), *Nonlinear Modeling and Forecasting*. Addison-Wesley, Redwood City, CA, 1992, 395–432.
- [14] M. Cottrell, B. Girard, Y. Girard, M. Mangeas, C. Muller, Neural modeling for time series: a statistical stepwise method for weight elimination, *IEEE Transactions on Neural Networks*, 6(6), 1995, 1355–1364.
- [15] R.N. Yadav, P.K. Kalra, J. John, Time series prediction with single multiplicative neuron model, *Applied Soft Computing*, 7, 2007, 1157–1163.
- [16] L. Zhao, Y. Yang, PSO-based single multiplicative neuron model for time series prediction, *Expert Systems with Applications*, 36, 2009, 2805–2812.
- [17] C.H. Aladag, E. Egrioglu, U. Yolcu, A.Z. Dalar, A new time invariant fuzzy time series forecasting method based on particle swarm optimization, *Applied Soft Computing*, 12, 2012, 3291–3299.
- [18] R. Storn, K. Price, *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, Technical Report TR-95-012, International Computer Science Institute, Berkeley, 1995.
- [19] G. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*, 50, 200, 159–175.
- [20] C.H. Aladag, E. Egrioglu, C. Kadilar, Forecasting nonlinear time series with a hybrid methodology, *Applied Mathematic Letters*, 22, 2009, 1467–1470.
- [21] U. Yolcu, E. Egrioglu, C.H. Aladag, A new linear & nonlinear artificial neural network model for time series forecasting, *Decision Support Systems*, 2013, 1340–1347.



Eren Bas is Assistant Professor at Giresun University, Turkey. He received the B.Sc. degree in statistics from the University of Ondokuz Mayıs Turkey, in 2009. He received the M.Sc. degree (M.Sc. thesis title: A new approach based on the genetic algorithm for fuzzy time series analysis) in statistics at the Institute of Science of the University of Ondokuz Mayıs, Turkey, in 2011 and the PhD.

degree in statistics at the Institute of Science of the University of Ondokuz Mayıs, Turkey, in 2014. His research interests include; time series analysis, fuzzy time series, artificial neural networks, heuristic and evolutionary algorithms.