

Single Multiplicative Neuron Model Artificial Neuron Network Trained by Bat Algorithm for Time Series Forecasting

Eren Bas¹, Ufuk Yolcu², Erol Egrioglu^{1*}, Ozge Cagcag Yolcu³, Ali Zafer Dalar¹

¹Department of Statistics, Faculty of Arts and Science, Forecast Research Laboratory, Giresun University, Turkey

²Department of Statistics, Faculty of Science, Ankara University, Turkey

³Department of Industrial Engineering, Faculty of Engineering, Forecast Research Laboratory, Giresun University, Turkey

Abstract In recent years, artificial neural networks have been commonly used for time series forecasting. One popular type of artificial neural networks is feed forward artificial neural networks. While feed forward artificial neural networks give successful forecasting results, they have an architecture selection problem. In order to eliminate this problem, Yadav et al. (2007) proposed single multiplicative neuron model artificial neural network (SMNM-ANN). There are various learning algorithms for SMNM-ANN in the literature such as particle swarm optimization and differential evolution algorithm. In this study, differently from these learning algorithms, bat algorithm is used for the training of SMNM-ANN for forecasting of time series. The SMNM-ANN trained by bat algorithm is applied to two well-known real world time series data sets and its superior forecasting performance is proved by comparing with the results of other techniques suggested in the literature.

Keywords Artificial neural networks, Single multiplicative neuron model, Bat algorithm, Forecasting

1. Introduction

There are many models presented in time series literature for forecasting problem. Traditional models may fail to satisfy to analyze many data sets which have various assumptions. From this point of view, especially over the last decades, artificial neural networks (ANNs) that do not include such strict assumptions have been widely taken advantage of as an alternative forecasting approach. The most important characteristic of ANN is that it has the ability to learn from a source of information (data). Learning process of ANN is the process of getting the best values of weights and this process is called as the training of ANN. The training problem of ANN can be regarded as an optimization problem. In the literature, although Levenberg-Marquardt and Back Propagation (BP) learning algorithms have been used to train ANN, some heuristic algorithms such as genetic algorithms, particle swarm optimization, differential evolution algorithm, simulated annealing and taboo search algorithm are also used, to train ANN especially in recent years.

The first ANN model was proposed by [1]. Multilayer perceptron (MLP) was presented by [2] in order to solve a

non-linear problem. A MLP model is composed of an input and output, and one or more hidden layer(s). Various ANN models were put forward by [3-6]. Moreover generalized mean neuron model (GMN) and geometric mean neuron (G-MN) model were introduced by [7, 8], respectively. [9] presented an approach to get combining forecast by using ANN. In time series forecasting problem, multi-layer feed forward artificial neural networks (ML-FF-ANN) model have been most commonly used. The time series forecasting literature based on ANN was summarized by [10]. And also [11-14], for forecasting real-world time series, touched on the issue of network structure in their studies. Even though feed forward artificial neural networks can produce successful forecasting results, they have an architecture selection problem that can have a negative effect on the forecasting performance of ANN. Therefore the determination of the numbers of neurons in the layers in ML-FF-ANN is a crucial problem. [15] proposed single multiplicative neuron model artificial neural network (SMNM-ANN) that has a single neuron and does not have architecture selection problem. While [15] used back propagation learning algorithm, [16] used cooperative random particle swarm optimization (CRPSO) in the learning process of SMNM-ANN. And also, the modified particle swarm optimization method and differential evolution algorithm (DEA) were utilized to train MNM-ANN by [17, 18].

* Corresponding author:

erole1977@yahoo.com (Erol Egrioglu)

Published online at <http://journal.sapub.org/ajis>

Copyright © 2016 Scientific & Academic Publishing. All Rights Reserved

In this study, bat algorithm (BA) that is used in new studies frequently is firstly utilized for the training of SMNM-ANN in time series forecasting. BA is based on the echolocation features of micro-bats (Yang, 2010) and bats are very good at finding the prey. They can distinguish very small prey and obstacles even in the dark. The algorithm mimics the same process of how bats can find and detect the food source with the help of echolocation. To support the idea that SMNM-ANN trained by BA (BA-SMNM-ANN) has satisfied forecasting performance, two well-known real-life time series data were analyzed. Section 2 gives an introduction of BA-SMNM-ANN and an algorithm of training process. The implementation and the obtained results are presented in Section 3. And finally last section provides some discussions.

2. The Proposed Method

There are various studies that show SMNM-ANN is effectively used in time series forecasting. Different learning algorithms have been used in the training process of SMNM-ANN to get better forecasting results.

In this study, SMNM-ANN is firstly trained by BA. Since the proposed BA-SMNM-ANN model is basically a SMNM, it has same structure with SMNM and it can be demonstrated in Figure 1.

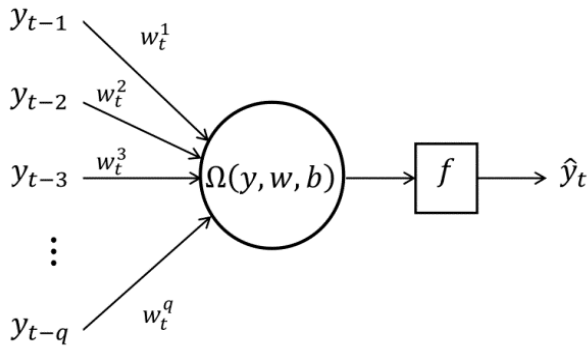


Figure 1. The structure of a BA-SMNM-ANN

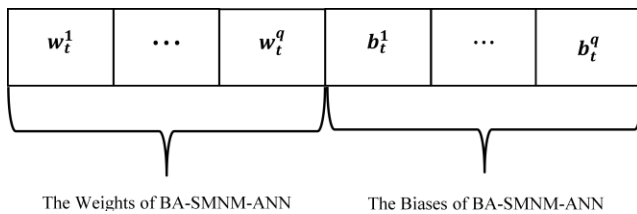


Figure 2. The structure of a bat

In Figure 1, function $\Omega(y, w, b)$ is the product of the weighted inputs and also f shows the activation function. Here, $y_{t-1}, y_{t-2}, \dots, y_{t-q}$ and \hat{y}_t are inputs and outputs of BA-SMNM-ANN, respectively. Moreover, q can be called as model order, n is the number of observations in training set of time series. The SMNM-ANN with q inputs given in Figure 1 has $q \times 2$ weights. Of these, q are the weights corresponding to the inputs ($w_t^i, i = 1, 2, \dots, q$;

$t = 1, 2, \dots, n$) and q to the sides of the weights ($b_t^i, i = 1, 2, \dots, q; t = 1, 2, \dots, n$). The optimization process can be also called as the training of BA-SMNM-ANN. The weights and biases are shown with w_t^i and b_t^i . Thus, each member of bat population has $2 \times q$ positions. The structure of a member of bat population is illustrated in Figure 2.

The algorithm of BA-SMNM-ANN is given steps by steps as below.

Step 1. Generate a set of bat population

Velocities (v_{ij}), positions (x_{ij}), rate (r), and loudness (A_0) are generated. The initial values of $A_0(0) = 0.8 r_0(0) = 0.05$ and $\alpha = 0.9$ and $\gamma = 0.9$ are constant parameters.

Step 2. Calculate the fitness value of each bat in the population.

To obtain fitness function values, firstly the outputs of the network are calculated by using following equations;

$$net_t = \prod_{i=1}^q (w_t^i \times y_{t-i} + b_t^i) \quad ; t = 1, 2, \dots, n \quad (1)$$

$$\hat{y}_t = f(net_t) = \frac{1}{1 + e^{-net_t}} \quad ; t = 1, 2, \dots, n \quad (2)$$

And also, root mean square error (RMSE) is calculated as the fitness function value;

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (3)$$

Step 3. Update variables for bats flying randomly

Following steps are applied for each bat.

Step 3.1. Generate f_i from uniform distribution with (f_{min}, f_{max}) parameters.

Step 3.2. Calculate velocities and positions of bats by using following formula.

$$v_{ij}^{t+1} = v_{ij}^t + (x_{ij}^t - pbest_j) * f_i \quad (4)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (5)$$

Step 3.3. u_1 is generated from uniform distribution with the parameters (0, 1). If $u_1 < r(t)$, positions of bats are updated by using following formula.

$$x_{ij}^{t+1} = x_{ij}^{t+1} + \phi * mean(A_0) \quad (6)$$

In this Equation, ϕ is a random number and it is generated from uniform distribution with the (-1, 1) parameters (-1, 1). Besides, A_0 is a vector and its elements are constituted by average loudness of all bats.

Step 3.4. u_2 is generated from uniform distribution with the parameters (0, 1). If $A_0(t) > u_2$ and new fitness value of the bat is smaller than the previous fitness value of the bat at the same time.

$$r(t + 1) = r(t) * (1 - \exp(-\gamma(t))) \quad (7)$$

$$A_0(t + 1) = \alpha * A_0(t) \quad (8)$$

$Pbest$ is updated. Otherwise $x_{ij}^{t+1} = x_{ij}^t$

Step 4. Check the stopping criteria.

If it is reached to the maximum number of iterations or the fitness value calculated from the bat population with the best fitness value. is less than a predetermined value (ϵ) the process is end, otherwise move to Step 3.

3. Implementation

To show the performance of the BA-SMNM-ANN, Australian Beer Consumption "(AUST)" data with 148 observations between years 1956 and 1994 was analyzed firstly.

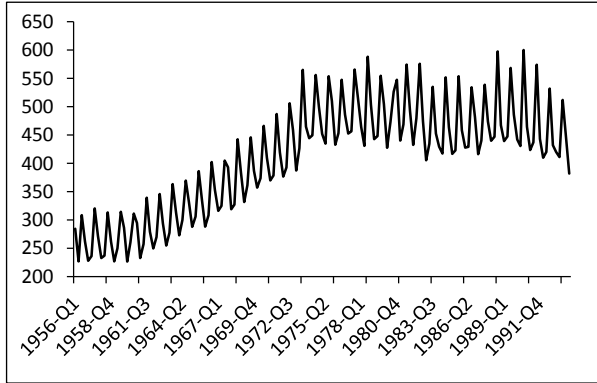


Figure 3. AUST data between the years of 1956-1994

The results obtained from various methods have been compared in respect to RMSE and Mean Absolute Percentage Error (MAPE) criteria given in Eq. 7.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (9)$$

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

Table 1. The results obtained from all methods for AUST data

Methods	RMSE	MAPE
BP-SMNM-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-SMNM-ANN	26.7831	0.0498
ML-FF-ANN	24.1052	0.0476
E-ANN	22.6581	0.0436
DEA-SMNM-ANN	19.7819	0.0372
BA-SMNM-ANN	17.7054	0.0323

The last 16 observations of the time series was taken as test data. In addition to the proposed method, AUST data, given in Figure 3, was analyzed by seasonal autoregressive integrated moving average (SARIMA), Winter's multiplicative exponential 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 single multiplicative neuron model neural network (BP-SMNM-ANN), single multiplicative neuron model artificial neural network based on particle swarm optimization (PSO-SMNM-ANN), single multiplicative neuron model artificial neural network based on differential evolution algorithm (DA-SMNM-ANN), Radial basis

artificial neural network (RB-ANN), Elman neural network (E-ANN) methods.

RMSE and MAPE criteria values for test set obtained from proposed model and other methods in the literature are given in Table 1.

With Table 1, we can clearly say that the proposed BA-SMNM-ANN 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 is given in Figure 4. According to this graph, it is obvious that the forecasts obtained from BA-SMNM-ANN are compatible with the real observations.

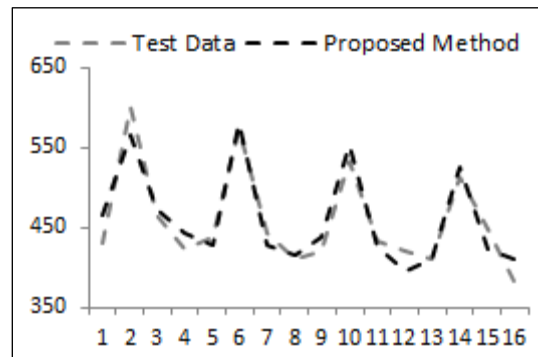


Figure 4. The graph of real observations and forecasts obtained from proposed method for AUST data

Finally, the proposed method was analyzed for Canadian lynx data between years 1821 and 1934. The last 14 observations of the time series was taken as test data. The logarithm (to the base 10) of the data was used in the analysis. The graph of the Canadian logarithmic lynx data is given in Figure 5.

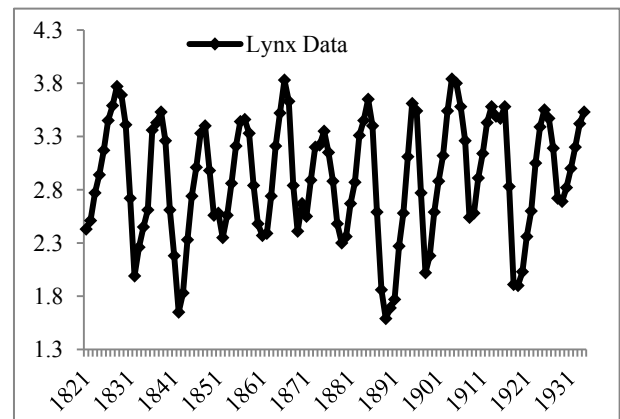


Figure 5. Logarithmic Canadian lynx data series (1821-1934)

This data set was also analyzed with SARIMA, WMES, RB-ANN, ML-FF-ANN, BP-SMNM-ANN, autoregressive integrated moving average (ARIMA), PSO-SMNM-ANN and [19]. During the analysis of Canadian lynx data, the model orders in other words the number of inputs of BA-SMNM-ANN (m) was changed from 2 to 4. The number of positions (np) in the population was changed from 30 to 100. The number of iteration was determined as 100.

At the end of the analysis, we conclude that the best result is obtained in the case where $m=4$, $np=30$.

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

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

Methods	MAPE
BP-SMNM-ANN	0.0807
RB-ANN	0.0636
WMES	0.0469
ML-FF-ANN	0.0376
[19]	0.0350
ARIMA	0.0312
PSO-SMNM-ANN	0.0293
BA-SMNM-ANN	0.0258

With Table 2, we can clearly say that the proposed BA-SMNM-ANN has the best performance among all methods in terms MAPE criteria.

4. Discussion and Conclusions

Several kind of ANN model is used in time series forecasting literature. Although ML-FF-ANN has been commonly used in the forecasting problem of time series, the architecture selection problem of it is crucial to get the reasonable forecasts. SMNM-ANN has not also such a problem because it has only one neuron. To train SMNM-ANN, there are various learning algorithms in the literature.

In this paper, for time series forecasting problems, SMNM-ANN is trained by BA. By favour of real-life time series implementations, the outstanding forecasting performance of BA-SMNM-ANN has been brought to light.

REFERENCES

- [1] McCulloch, W.S., Pitts, W., 1943, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5, 115–133.
- [2] Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986, Learning representations by back propagating errors, *Nature*, 323, 533-536.
- [3] Basu, M., Ho, T.K., 1999, Learning behavior of single neuron classifiers on linearly separable or nonseparable inputs, In *IEEE LICNN'99*.
- [4] Labib, R., 1999, New single neuron structure for solving non-linear problems", In *IEEE IJCNN'99*, 617–620.
- [5] Plate, T.A., 2000, Randomly connected sigma-pi neurons can form associator networks, *NETCNS: Network: Computation in Neural Systems*, 11, 321–322.
- [6] Zhang, C.N., Zhao, M., Wang, M., 2000, Logic operations based on single neuron rational model, *IEEE Transactions on Neural Networks*, 11, 739–747.
- [7] Yadav, R.N., Kumar, N., Kalra, P.K., John, J., 2006, Learning with generalized-mean neuron model, *Neurocomputing*, 69, 2026-2032.
- [8] Shiblee, M., Chandra, B., Kalra, P.K., 2010, Learning of geometric mean neuron model using resilient propagation algorithm, *Expert Systems with Applications*, 3, 7449-7455.
- [9] Aladag, C.H., Egrioglu, E., Yolcu, U., 2010, Forecast combination by using artificial neural networks, *Neural Processing Letters*, 32 (3), 269–276.
- [10] Zhang, G., Patuwo, B.E., Hu, Y.M. 1998, Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting*, 14, 35-62.
- [11] Sharda, R., 1994, Neural networks for the MS/OR analyst: An application bibliography, *Interfaces*, 24 (2), 116–130.
- [12] Weigend, A.S., Huberman, B.A., Rumelhart, D.E., 1990, Predicting the future: A connectionist approach, *International Journal of Neural Systems*, 1, 193–209.
- [13] Weigend, A.S., Huberman, B.A., Rumelhart, D.E., 1992, Predicting sunspots and exchange rates with connectionist networks. In: Casdagli, M., Eubank, S. (Eds.), *Nonlinear Modeling and Forecasting*, Addison-Wesley, Redwood City, CA, 395–432.
- [14] Cottrell, M., Girard, B., Girard, Mangeas, Y.M., Muller, C., 1995, Neural modeling for time series: a statistical stepwise method for weight elimination, *IEEE Transactions on Neural Networks*, 6(6), 1355–1364.
- [15] Yadav, R.N., Kalra, P.K., John, J., 2007, Time series prediction with single multiplicative neuron model, *Applied Soft Computing*, 7, 1157-1163.
- [16] L Zhao, Yang, Y., 2009, PSO-based single multiplicative neuron model for time series prediction, *Expert Systems with Applications*, 36, 2805-2812.
- [17] Aladag, C.H., Egrioglu, E., Yolcu, U., Dalar, A.Z., 2012, A new time invariant fuzzy time series forecasting method based on particle swarm optimization, *Applied Soft Computing*, 12, 3291-3299.
- [18] Bas, E., 2016, The training of multiplicative neuron model artificial neural networks with differential evolution algorithm for forecasting, *Journal of Artificial Intelligence and Soft Computing Research*, 6(1), 5-11.
- [19] Zhang, G., 2000, Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*, 50, 159-175.