



Bayero Journal of Pure and Applied Sciences, 9(1): 6 - 10

Received: January, 2016

Accepted: May, 2016

ISSN 2006 – 6996

EFFECTIVENESS OF FIREFLY ALGORITHM BASED NEURAL NETWORK IN TIME SERIES FORECASTING

Abdulkarim, S. A. and Garko, A. B.

Computer Science Department, Federal University Dutse, Jigawa State, Nigeria
(sakwami@live.com; abgarko@gmail.com) 08037713775

ABSTRACT

Global optimization techniques such as Particle Swarm Optimizers (PSO) and Genetic Algorithm (GA) are now widely used for training Artificial Neural Networks (NN), particularly in time series forecasting problems. Firefly algorithm (FA) is a relatively new addition to the family of population based optimization technique that has shown promising result in a number of problems. In this work, we evaluate the effectiveness of FA trained NN in time series forecasting. In the experiments, three well known time series were used to evaluate the performance. Results obtained were compared with results from both PSO and Resilient Propagation (RPROP) trained NNs. FA based NN performed very well in forecasting all the time series considered, outperforming the bench-marks in two out of the three problems.

Keywords: Time series, Artificial Neural Network, Firefly Algorithm, Particle Swarm Optimization, Overfitting

INTRODUCTION

Time series forecasting is hot research area which has significant practical applications in many fields (Guoqiang Zhang et al. 1998). A number of international research journals are dedicated to forecasting. Initially, only statistical methods were used to analyze time based observations (series) to develop appropriate model that can be used to predict future values of that series. Auto Regressive Moving Average (ARIMA) is arguably the most popular statistical model use in forecasting. Being a linear model, ARIMA can model variety of time series with ease and simplicity, except nonlinear time series. Unfortunately, most real life time series are nonlinear. This led to the development of nonlinear statistical techniques such as Non-linear Auto Regressive (NAR) model (GP Zhang 2007). Even though these models can handle nonlinear time series, they are characterized to have huge mathematical complexities and largely depend on the specific knowledge of how the time series concern is generated, which is usually unpredictable, thereby limiting their application in general time series forecasting (Gooijer and Kumar 1992). As an alternative nonlinear model that can be adaptively formed based on the features of presented data without the prior knowledge of input output relationship, Artificial Neural Network (NN) gain huge attention and widely applied in forecasting time series (Adhikari and Agrawal 2012). Even though NN has been successfully used in many forecasting problems, its performance heavily depends on the selected NN architecture and training algorithm. There is no systematic way of determining NN architecture, it is normally selected via experimentation while gradient descent based back propagation algorithm is often

used as the training method. Backpropagation has been used with success in many applications (Engelbrecht, 2007). However, it has the tendency of converging on local minima and sometimes have slow convergence rate. To mitigate these issues, several improved versions of the algorithm such as Scaled Conjugate Gradient (SCG) (Møller, 1993) and Resilient propagation (RPROP) (Riedmiller and Braun, 1993) were developed.

To avoid issue of gradient descent based algorithms (local optimization algorithms), global optimization algorithms have been proposed and successfully used in training NNs (Rakitianskaia and Engelbrecht 2009). Examples include Genetic Algorithm (GA) (Whitley, 1994), Artificial Bee Colony algorithm (ABC) (Karaboga, 2010) and Particle Swarm Optimizers (PSO) (Hu *et al.*, 2004). Firefly algorithm (FA) is a relatively new population based optimization algorithm based on the idealized behavior of flashing characteristics of fireflies (Yang, 2009). It was empirically shown to outperform PSO in some optimization problems. Recently (Brajevic and Tuba, 2015) investigated its applicability to training NN in classification problems and compared its performance with GA and ABC. In this paper, we investigated the performance of Firefly Algorithm (FA) trained ANN in time series forecasting, bench-marking the result against that obtained from PSO and RPROP trained NNs.

Subsequent sections of the paper are organized as follows; Section II presents the necessary background information for the study, section III describe the methodology adopted, results are presented and discussed in section IV, section V concludes the paper.

BACKGROUND

A. Firefly Algorithm

Firefly algorithm (FA) is a meta-heuristic optimization algorithm developed by Xin-Shen Yang in 2008, inspired by the flashing behavior of fireflies based on the following assumptions (Yang 2009);

1. All fireflies are unisexual; they get attracted to each other regardless of their sex.
2. Attractiveness is proportional to their brightness, which both decreases as distances increases. For any two fireflies, the less brighter one is attracted by the brighter one. Fireflies moves randomly when there is no brighter one.
3. Brightness of a firefly is determined by the landscape of the objective function optimized. The variation of attractiveness is then defined as (Yang 2009);

$$\beta(r) = \beta_0 e^{-\gamma r^2}$$

where r is distance and β_0 the attractiveness as $r = 0$, and γ is light absorption coefficient.

For a firefly i attracted by a brighter one say j , its movement is determined as;

$$x_{ik}(t+1) = x_{ik}(t) + \beta_0 e^{-\gamma r_{ij}^2} [x_{jk}(t) - x_{ik}(t)] + \alpha(t) \epsilon_{ik}(t) \quad (2)$$

Where $k = 1 \dots D$, (D is the dimension of the problem), $\alpha(t)$ control the step size while $\epsilon_{ik}(t)$ is a vector of random numbers at time t . It is worth noting that when $\beta_0 = 0$, the algorithm becomes a simple random walk and when $\gamma = 0$, it correspond to a variant of particle swarm optimization (PSO) (Nandy, Bengal, and Bengal 2012).

Firefly algorithm is given below;

$$(1)$$

```

Create and randomly initialize population of fireflies
Initialize algorithm parameters  $\alpha, \beta_0$ , and  $\gamma$ 
repeat
  for each firefly  $i = 1:n$ 
    for each firefly  $j = 1:n$ 

      if  $f(x_j) < f(x_i)$  // Objective function  $f(x) = (x_1, x_2, \dots, x_D)^T$ 

        move firefly  $i$  towards  $j$  using equation 2
        vary attractiveness with distance  $r$  using equation 1
        evaluate new solutions and update the new solution;
      end if
    end
  end
end
    
```

B. Artificial Neural Networks in Time series forecasting

Basically, NN is the realization of a non-linear mapping $f_{ANN}: R^I \rightarrow R^O$ where I and O are the dimensions of the input and desired output space respectively. The function f_{ANN} is usually a complex function of a set of non-linear functions (Engelbrecht 2007). In principle, we can say NN is a nonlinear mapping of n past observations $x_{t-1}, x_{t-2}, \dots, x_{t-n}$ to some future value x_t ;

$$x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-n}) + \epsilon_t \quad (3)$$

$$o_{k,n} = f_{o_k} \left(\sum_{j=1}^{J+1} w_{kj} f_{y_j} \left(\sum_{i=1}^{I+1} v_{j,i} x_i \right) \right) \quad (4)$$

where $x_i, i = 1..I$ and $y_j, j = 1..J$ are the units in the input and hidden layer respectively; the $(I + 1)$ th input and $(J + 1)$ th hidden units are bias units for AN in the next level; w_{kj} is the weight between output unit

where ϵ_t is the error at time t .

Most forecasting problems employs the traditional NN often called feedforward NN (FNN). FNN is nothing but an acyclic collection of computing units (called artificial neuron). Its structure allows information flow in only one direction from a set of input units, through a set of hidden units to the set of output units. The output of a FNN is computed with a single forward pass through the network for any given input pattern x_n as;

o_k and hidden unit y_j ; $v_{j,i}$ is the weight between hidden unit y_j and input unit x_i ; f_{o_k} and f_{y_j} are the activation functions of output unit o_k and hidden unit y_j respectively.

C. Firefly Algorithm for NN training

Firefly can easily be applied to NN training just like PSO. Each firefly is used to represent a candidate solution to the NN training problem (i.e. a vector of all the weights and biases of a NN). Fitness of each firefly is calculated by substituting its position into the NN, and mean squared error (MSE) over the training set to obtain the training error (T_E), or over the generalisation set to get the generalisation error (G_E).

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \tag{5}$$

Initial conditions $a = 0.2$, $b = 0.1$ and $\tau = 30$ and $x(t) = 0.9$ for $0 \leq t \leq \tau$ were used. 500 points dataset was generated for this study. Plot of the series is shown in Fig. 1.

c) ;

$$x(n+1) = x(n) + Gx(n)(1-x(n)) \tag{6}$$

Plot of the logistic map time series is displayed in Fig. 2.

d) *Sunspot*: The series has 289 data points representing the total annual measure of sunspot

The firefly algorithm is then used to move firefly through the weight space in order to minimise the MSE.

MATERIALS AND METHODS

Three well known bench-mark time series were selected for the experiments. They are;

a) *Mackay Glass*: This data set is a solution of the Mackey-Glass delay-differential equation (Lapedes and Farber 1987);

b) *Logistic Map*: The series was generated by iterating the logistic map equation 150 times starting from a random initial value set to 0.1 and with $G = 3$. Logistic map equation is define as

from 1770 to 1988, obtained online from Time Series Data Library (Hyn-d-man and Hyndman 2013). The series has a strong seasonal pattern and somewhat constant trend as shown in Fig. 3.

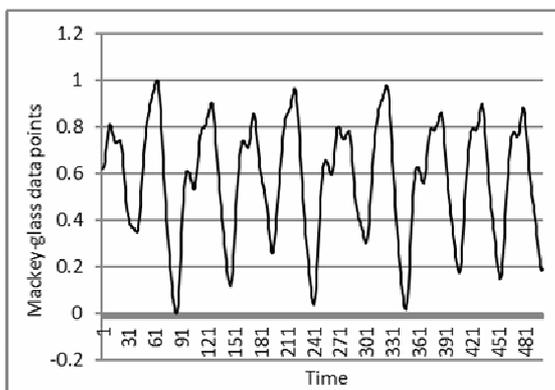


Fig. 1: Mackay-Glass Time Series

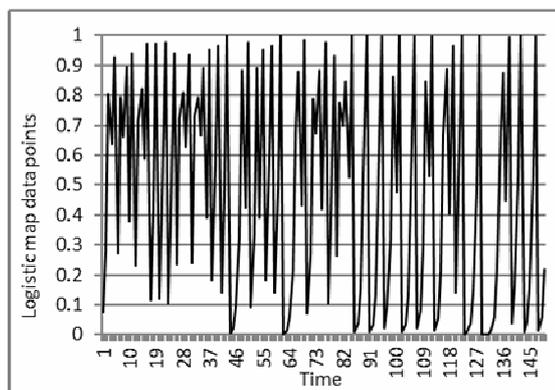


Fig. 2: Logistic Map Time Series

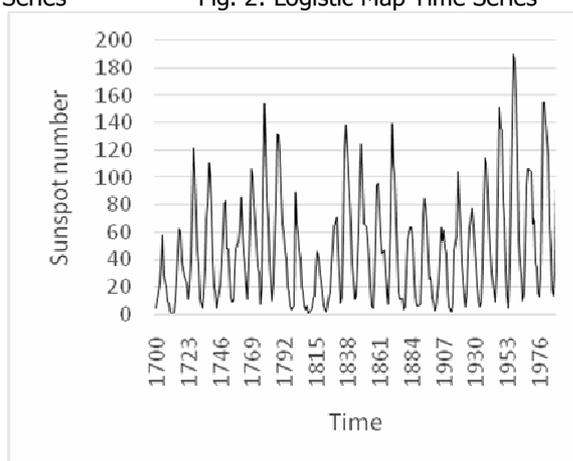


Fig. 3: Sunspot Time Series

All datasets were scaled to the range $[-1,1]$ to lie within the range of activation function used and standardized so that their mean is close to zero. Each dataset is then partitioned chronologically into training and generalization set in the ratio 70:30.

FNN with single hidden layer was used in the experiments. Modified hyperbolic tangent functions were employed in the hidden and output layer as suggested by (LeCun 1989), define as

$$f(net) = 1.7159 \tanh\left(\frac{2}{3}net\right) \tag{7}$$

For each problem, hidden units were iteratively optimized on the training set. The numbers in the range [2, 12] were considered. A single output unit NN was used for all problems since we considered only one step ahead forecasting.

For the FA, a swarm of 20 fireflies were used to trained the NN. α was set to varies linearly from 0.2 to 0, $\beta_0 = 0.2$ and $\gamma = 1.0$. Standard PSO was used in the experiment. A swarm of 20 particles was employed. Inertia weight w , was set to 0.72 and acceleration coefficients fixed at $c_1 = c_2 = 1.49$ as suggested by (Bergh 2001).

We used Computational Intelligence Library (CILib) (Cloete, Engelbrecht, and G 2008) to carry out all the experiments. Reported results were averages over 30 simulations. Stopping condition for each algorithm was set to 2×10^4 function evaluations. MSE was used as the performance measure. Training and Generalisation errors were used to assess the

accuracy of the forecasting model. Even though no effort was taken to avoid overfitting, Generalization factor developed by (Röbel 1994) to measure overfitting was reported. Overfitting is when NN lost its generalization ability due to memorization of training pattern.

RESULTS AND DISCUSSION

Table 1 present the results of forecasting Mackay Glass Time Series, Logistic Map Time Series and Sunspot Time Series using three different models.

In forecasting Mackay Glass, FA trained NN had the lowest generalisation and training errors compared to both PSO and RPROP trained NNs. All the three forecasting methods showed no sign of over fitting.

PSO trained NN outperformed FA and RPROP in forecasting Logistic Map as shown in the result table. FA-NN performed very well also, came second closely behind PSO-NN

For Sunspot, FA trained NN had lowest errors compared to the other two.

Table 1: Mean Errors of Forecasting Model

Problem	Training Algorithm	Training Error		Generalisation Error		Generalisation Factor	
		MSE	Confidence	MSE	Confidence	MSE	Confidence
Mackay Glass	RPROP	2.24E-06	2.42E-07	2.18E-06	2.35E-07	9.74E-01	1.18E-03
	PSO	2.06E-06	4.55E-07	1.99E-06	4.43E-07	9.65E-01	4.22E-02
	FA	7.11E-07	1.25E-07	7.27E-07	1.34E-07	1.04E+00	3.29E-01
Logistic Map	RPROP	4.39E-01	1.26E-01	4.85E-01	1.54E-01	1.08E+00	3.18E-02
	PSO	4.46E-04	1.02E-04	4.83E-04	1.13E-04	1.12E+00	1.69E-01
	FA	7.29E-04	2.04E-05	7.03E-04	2.13E-05	9.65E-01	3.29E-02
Sunspot	RPROP	1.11E-01	8.91E-02	2.44E-01	2.02E-01	2.03E+00	6.78E-02
	PSO	2.15E-02	3.63E-04	6.28E-02	4.31E-03	2.93E+00	2.10E-01
	FA	2.22E-02	9.98E-05	7.65E-02	2.02E-03	3.45E+00	8.36E-02

CONCLUSION

The aim of the study is to investigate the forecasting accuracy of firefly trained NN, and then compare the performance with those obtained from 2 established forecasting models; PSO trained NN and RPROP trained NNs. Experiments were carried out using three bench mark time series. Results obtained suggested that FA trained NN outperformed both PSO and RPROP trained NNs in two out of three forecasting problems. PSO-NN performed better than the other two in one out of three problems. RPROP-NN had the worst performance in all three problems. FA being a

fairly new algorithm has proven from the empirical study to be a very effective NN training method for time series forecasting problems. Future studies should evaluate its performance in training recurrent NNs and in dynamic time series forecasting problems.

CONTRIBUTIONS

The first author (Salihu A. Abdulkarim) proposed the topic, wrote the background information necessary for the study, the methodology adopted, discussed the results obtained and wrote entire manuscript. The second author (Ahmed B. Garko) conducted the experiments and presented the results.

REFERENCES

Adhikari, R, and RK Agrawal. (2012). Forecasting Strong Seasonal Time Series with Artificial Neural Networks. *Journal of Scientific and Industrial*
http://www.researchgate.net/publication/230583150_Forecasting_Strong_Seasonal_Time_Series_with_Artificial_Neural_Networks/file/32bfe51227916a4f90.pdf (October 28, 2013).
 Bergh, Frans Van den. (2001). An Analysis of Particle

Swarm Optimizers. University of Pretoria, South Africa.
 Brajevic, Ivona, and Milan Tuba. (2015). Training Feed-Forward Neural Networks Using Firefly Algorithm Faculty of Computer Science 2 Training Feed-Forward Artificial Neural Networks. *Recent Advances in Knowledge Engineering and Systems Science Training*. 156–61.

- Cloete, T, A P Engelbrecht, and Pampará G. (2008). CIlib: A Collaborative Framework for Computational Intelligence Algorithms – Part I. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*. IEEE International Joint Conference on IEEE, , 1750–57.
- Engelbrecht, AP.(2007). *Computational Intelligence: An Introduction*.
- Gooijer, JG De, and K Kumar. (1992). Some Recent Developments in Non-Linear Time Series Modelling, Testing, and Forecasting. *International Journal of Forecasting*. <http://www.sciencedirect.com/science/article/pii/S016920709290115P> (April 30, 2014).
- Hu, X, Y Shi, and R Eberhart. (2004). Recent Advances in Particle Swarm. *Evolutionary Computation, 2004.* http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1330842 (January 24, 2013).
- Hyn-d-man, R.J, and R.J. Hyndman. (2013). Time Series Data Library." <https://datamarket.com/data/list?q=cat:g24provider:tsdl> (December 5, 2015).
- Karaboga, D. (2010). Artificial Bee Colony Algorithm. *scholarpedia*. http://scholarpedia.org/article/Artificial_bee_colony_algorithm (May 25, 2016).
- Lapedes, AS, and RM Farber. 1987. "How Neural Nets Work. *Neural information processing systems*. <http://papers.nips.cc/paper/59-how-neural-nets-work> (May 15, 2014).
- LeCun, Y. (1989). Generalization and Network Design Strategies. *Connections in Perspective. North-Holland, Amsterdam*. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.476.479&rep=rep1&type=pdf> (November 23, 2015).
- Møller, MF. (1993). A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Neural networks*. <http://www.sciencedirect.com/science/article/pii/S0893608005800565> (May 25, 2016).
- Nandy, Sudarshan, West Bengal, and West Bengal. (2012). Analysis of a Nature Inspired Firefly Algorithm Based Back-Propagation Neural Network Training. *International Journal of Computer Applications* 43(22): 8–16.
- Rakitianskaia, Anna, and AP Andries Engelbrecht. (2009). Training Neural Networks with PSO in Dynamic Environments. ... , 2009. *CEC'09. IEEE*
- Riedmiller, M, and H Braun. (1993). "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. *Neural Networks, 1993., IEEE* http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=298623 (May 6, 2014).
- Röbel, A. (1994). The Dynamic Pattern Selection Algorithm: Effective Training and Controlled Generalization of Backpropagation Neural Networks." *Technical report, Technische Universität Berlin*. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.9352> (November 8, 2015).
- Whitley, D. (1994). "A Genetic Algorithm Tutorial." *Statistics and computing*. <http://link.springer.com/article/10.1007/BF00175354> (May 25, 2016).
- Yang, Xin-she. (2009). Firefly Algorithms for Multimodal Optimization." *Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences* 5792: 169–78.
- Zhang, GP. (2007). A Neural Network Ensemble Method with Jittered Training Data for Time Series Forecasting. *Information Sciences*. <http://www.sciencedirect.com/science/article/pii/S0020025507003003> (November 2, 2013).
- Zhang, Guoqiang (1998). Forecasting with Artificial Neural Networks: The State of the Art." *International journal of forecasting* 14(1): 35–62. <http://www.sciencedirect.com/science/article/pii/S0169207097000447> (October 28, 2013).