Effectiveness of firefly algorithm based neural network in time series forecasting

Article · July 2016
DOI: 10.4314/bajopas.v9i1.2

2 authors, including:

Ahmed Baita Garko
Federal University Dutse, Nigeria

Some of the authors of this publication are also working on these related projects:

- Design and Implementation of Automated Teller Machine (ATM) for Banking Transaction Review on Withdrawals and Depositing. View project
- Design and Implementation of an Automated Staff Appraisal System View project
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 benchmarks 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 land
scape of the objective function optimized.
The variation of attractiveness is then defined as
(Yang 2009):
\[ \beta(i) = \beta_0 e^{-\gamma d^2} \]  
\[ (1) \]
where \( \gamma \) is distance and \( \beta_0 \) the attractiveness as
\( \gamma = 0 \), and \( \gamma \) is light absorption coefficient.
For a firefly \( i \) attracted by a brighter one say \( j \), its
movement is determined as;
\[ x_{ij}(t+1) = x_{ij}(t) + \beta_{ij}(t) [x_{jk}(t) - x_{ij}(t)] + \alpha(t) \xi(t) \]  
\[ (2) \]
Where \( k = 1 \ldots D \), \((D \) is the dimension of the
problem), \( \alpha(t) \) control the step size while \( \xi(t) \) 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;

Create and randomly initialize population of fireflies
Initialize algorithm parameters \( \alpha, \beta_0 \) and \( \gamma \)
repeat
  for each firefly \( i = 1 \) to \( n \)
    for each firefly \( j = 1 \) to \( n \)
      if \( f(x_i) < f(x_j) \) // Objective function \( f(x) = (x_1, x_2, \ldots, x_D)^T \)
        move firefly \( i \) towards \( j \) using equation 2
        vary attractiveness with distance \( \gamma \) using equation 1
        evaluate new solutions and update the new solution;
      end if
    end for
  end for
end repeat

B. Artificial Neural Networks in Time series forecasting
Basically, NN is the realization of a non-linear
mapping \( f_{ANN}: \mathbb{R}^I \rightarrow \mathbb{R}^O \) 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 \( x_{t-n}, x_{t-n+1}, \ldots, x_{t} \) to some future value \( x_t \);\n\[ x_t = f(x_{t-n}, x_{t-n+1}, \ldots, x_{t}) + e_t \]  
\[ (3) \]
\[ u_{R:n} = f_{O:n}(\sum_{j=1}^{J} w_{R,j} f_{Y,j}(\sum_{k=1}^{K} w_{k,j} x_k)) \]  
\[ (4) \]
where \( x_t, \ j = 1 \ldots J \) are the units in the input and
hidden layer respectively; the \((J+1)^{th} \) input and
\((J+1)^{th} \) hidden units are bias units for AN in the
next level; \( w_{R,j} \) is the weight between output unit
\( O_k \) and hidden unit \( Y_j \); \( w_{k,j} \) is the weight between
hidden unit \( Y_j \) and input unit \( x_k \); \( f_{O,k} \) and \( f_{Y,j} \) are
the activation functions of output unit \( O_k \) and hidden
unit \( Y_j \) respectively.

where \( e_t \) 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). It 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;

\[ u_{R:n} = f_{O:n}(\sum_{j=1}^{J} w_{R,j} f_{Y,j}(\sum_{k=1}^{K} w_{k,j} x_k)) \]  
\[ (4) \]
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 (TE), or over the generalisation set to get the generalisation error (GE).

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);

\[ \frac{dx(t)}{dt} = \frac{ax(t-tau)}{1+x^k(t-tau)} - bx(t) \]  \hspace{1cm} (5)

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

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

\[ x(n+1) = x(n) + \mu \times (1 - x(n)) \times x(n) \] \hspace{1cm} (6)

c) Sunspot: The series has 289 data points representing the total annual measure of sunspot from 1770 to 1988, obtained online from Time Series Data Library (Hyndman and Hyndman 2013). The series has a strong seasonal pattern and somewhat constant trend as shown in Fig. 3.

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 partition chronologically into training and generalization set in the ratio 70:30.

The firefly algorithm is then used to move firefly through the weight space in order to minimise the MSE.
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. $\alpha$ 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. Inert weight $\omega$, 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 $2 \times 10^6$ 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 presents 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>
<thead>
<tr>
<th>Problem</th>
<th>Training Algorithm</th>
<th>Training Error</th>
<th>Generalisation Error</th>
<th>Generalisation Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MSE</td>
<td>Confidence</td>
<td>MSE</td>
</tr>
<tr>
<td>Mackay Glass</td>
<td>RPROP</td>
<td>2.24E-06</td>
<td>2.42E-07</td>
<td>2.18E-06</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>2.06E-06</td>
<td>4.55E-07</td>
<td>1.99E-06</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>7.11E-07</td>
<td>1.25E-07</td>
<td>7.27E-07</td>
</tr>
<tr>
<td>Logistic Map</td>
<td>RPROP</td>
<td>4.39E-01</td>
<td>1.26E-01</td>
<td>4.85E-01</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>4.46E-04</td>
<td>1.02E-04</td>
<td>4.83E-04</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>7.29E-04</td>
<td>2.04E-05</td>
<td>7.03E-04</td>
</tr>
<tr>
<td>Sunspot</td>
<td>RPROP</td>
<td>1.11E-01</td>
<td>8.91E-02</td>
<td>2.44E-01</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>2.15E-02</td>
<td>3.63E-04</td>
<td>6.28E-02</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>2.22E-02</td>
<td>9.98E-05</td>
<td>7.65E-02</td>
</tr>
</tbody>
</table>

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 benchmark time series. Results obtained suggested that FA trained NN outperformed both PSO and RPROP trained NNs in two 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


