Time Series Forecasting using Artificial Neural Networks

[*]Rajasekhar Nannapaneni, Raghavendra V. Kulkarni

Faculty of Engineering and Technology, M.S. Ramaiah University of Applied Sciences, Bengaluru-560054

*Contact Author Email: raja.nannapaneni@gmail.com

Abstract

Time series datasets contain a collection of observations in time and this additional temporal component differentiates time series with normal datasets. Studies of forecasting time series datasets are particularly popular as they improve financial planning, sales prediction, average rainfall forecast, internet traffic rates, business trend forecasting, weather forecast, contagious disease spread, etc. Time series forecasting facilitates the decision process in various domains such as business by predicting timely sales information thereby improving management of product manufacturing output. In medical domain, predicting patient health improves medical treatment. The environmental and physical time series forecasting such as rainfall, disasters prediction would improve the contingency plans. On an overall basis time series prediction improves the way humans live. Artificial neural networks are increasingly becoming successful in predicting the time series data through regression-based models. In this article, a benchmark time series model called Mackey-Glass chaotic time series is predicted using feedforward multi-layer neural network trained using the error backpropagation algorithm. The architecture and algorithm proposed in this article can easily be extended to any real-world time series prediction requirements.

Key Words: Artificial Neural Networks, Mackey-Glass, Time Series, Forecasting

1. INTRODUCTION

Time series analysis is a crucial research area that is attracting researchers' attention in the recent past. Timebound data is being generated continuously in numerous fields including businesses, natural phenomena, medical devices or industrial production output. Data collected sequentially or in an orderly fashion over time is known as time series data and analyzing a time series data helps in improving decision making in the concerned sector [1]. Improved decision making in a process can be achieved by improving the forecasting of future values of relevant parameters based on their current values [2].

Time series data that cannot be modelled using a mathematical equation or a function is known as a chaotic time series. Most natural phenomena, like disease spread, annual rainfall and industrial sensor data, which are often corrupted by noise, fall into the category of chaotic time series. A chaotic time series is non-stationary, and it appears to be random due to noise and other external influences.

Mackey-glass chaotic time series is a benchmark chaotic time series that has 1201 data points in time [3]. The objective is to model this dynamic process or time series and forecast future values.

Artificial neural networks (ANNs) are increasingly becoming popular in forecasting or predicting time series data. An ANN model has been proposed in this paper to predict the future values of Mackey-glass chaotic time series. This paper is an abridged version of [4]. The ANN model and the training method used in the paper performs better than some traditional statistical methods.

2. BACKGROUND AND RELATED WORK

An ANN is biologically inspired as its building block – artificial neuron – derived from McCulloch Pitts model aims to emulate a biological neuron. Though ANNs are unable to emulate perfect behavior of a human brain, they can perform a few other tasks that humans do better than the traditional programming or rule-based systems [5]. ANNs can be designed with various combinations of interconnections of neurons resulting in various topologies or structures with varying computations and connection weights that can perform powerful human-like tasks.

ANNs are naturally good in pattern (image or speech) recognition and association tasks where they almost behave like or sometimes even better than humans. Their importance can be understood as they tend to classify/categorize unseen data and are also able to predict unknown future trends. They learn by examples and map inputs to output patterns. They are also fault tolerant and are robust in executing the tasks [6].

Traditionally, several methods have been developed for the prediction of future values, and to model a given time series [7]. These include auto regressive integrated moving average and other statistical methods. However, the performance of statistical and moving average methods are not always encouraging, especially with higher order complexities and when the given dynamic processes are non-stationary. The nature of any given dynamic process is usually non-stationary due to noise and other environmental conditions.

Neural networks have been extensively used for time series prediction [8]. Forecasting a time series has also been addressed using regression models, such as auto regressive integrated moving average (ARIMA) model.

However, ARIMA model works well when the given function is linear and stationary [9]. Mackey-Glass time series is non-linear and a non-stationary function.

3. PROPOSED ANN ARCHITECTURE AND ALGORITHM

The Mackey glass time series can be predicted using feedforward multi-layer perceptron neural network. The feedforward neural network that was architected for this prediction has two hidden layers where the first hidden layer has 6 neurons; the second hidden layer has 3 neurons and the output layer has 1 neuron. The neurons in Figure 1 are fully connected such that all the inputs at input layer are applied to all the first hidden layer neurons, the outputs of the first hidden layer neurons are applied to all the second hidden layers and all the outputs of the second hidden layers are applied to output layer neuron.

The activation function of the hidden layer neurons is log sigmoid function and the activation function of the output layer is linear function. The Mackey glass time series is a nonlinear function and hence to predict this function it is required to introduce non-linearity throughout the network which can be accomplished by using log sigmoid as the activation function. The log sigmoid function is differentiable and hence its derivative can be used to compute the error during backpropagation.

Since the time series prediction is a regression, the usage of linear function as activation function at output justifies that we need continuous real values rather than binary values. The selection of 6 neurons in the first hidden layer and 3 neurons in the second hidden layer is better justified in Section 3.3 where it is shown that the mean square error during training is of the order 10-5 which supports the suitability of this architecture to predict the given time series.

The network architecture with single layers didn't achieve the error of the order 10-5 and networks with multiple layers or other architectures with more neurons increases the complexity and resource requirements. Hence the 2-hidden layer feedforward network fits just with right resources to predict and simultaneously achieves reasonable mean square error.

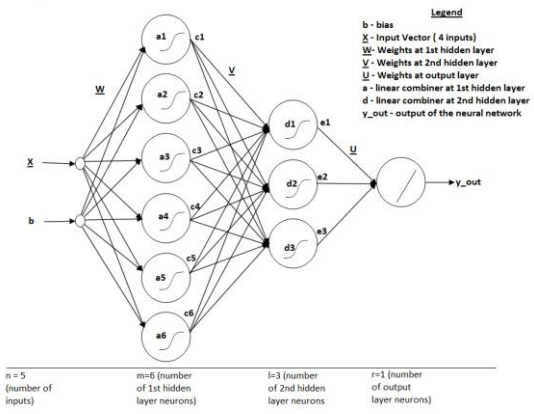

Fig.1 Feedforward multi-layer perceptron neural network architecture

The training algorithm that was used in the feedforward neural network to predict the Mackey-glass time series is backpropagation. There are multiple different algorithms available however backpropagation algorithm is sufficient to achieve best results for the prediction of time series. The backpropagation algorithm includes derivatives of the activation function and since the neural network architecture considered in Section 3.1 has differentiable activation functions such as log sigmoid and linear function, it is relatively simple to use backpropagation algorithm to solve the time series prediction.

The objective is to predict $(t+6)^{th}$ value of time series using the t, $(t-6)^{th}$, $(t-12)^{th}$ and $(t-18)^{th}$ values of the time series. The Mackey-glass time series has a total of 1201 values of which the first 600 points are used for training the network and the remaining 600 points will be used for testing the results of the network.

The feedforward neural network with error back propagation has two phases called forward phase and backward phase. In the forward phase, the inputs are propagated forward towards the output and in the backward phase the error is computed at the end of each layer and propagated forward towards the inputs.

3.1 Forward Phase:

The following are steps involved in forward training phase:

$n = 5$ (number of inputs including the bias)
$m = 6$ (number of neurons in 1st hidden layer)
$l = 3$ (number of neurons in 2nd hidden layer)
$r = 1$ (number of neurons in output layer)
X is the input vector and is of the order of $(n \times 1)$ which is (5×1) in this case and given by,

$$X = \begin{bmatrix} x(t) \\ x(t-6) \\ x(t-12) \\ x(t-18) \\ b \end{bmatrix}$$

W is the weight matrix of the 1st hidden layer and is of the order $(m \times n)$ which is (6×5) in this network and given by

$$W = \begin{bmatrix} w11 & w12 & w13 & w14 & w15 \\ w21 & w22 & w23 & w24 & w25 \\ w31 & w32 & w33 & w34 & w35 \\ w41 & w42 & w43 & w44 & w45 \\ w51 & w52 & w53 & w54 & w55 \\ w61 & w62 & w63 & w64 & w65 \end{bmatrix}$$

The activation vector a for the first hidden layer is of the order $(m \times 1)$ which is (6×1) and is given by,

$$a = \begin{bmatrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \end{bmatrix} = W \times X$$

$$= \begin{bmatrix} w11 & w12 & w13 & w14 & w15 \\ w21 & w22 & w23 & w24 & w25 \\ w31 & w32 & w33 & w34 & w35 \\ w41 & w42 & w43 & w44 & w45 \\ w51 & w52 & w53 & w54 & w55 \\ w61 & w62 & w63 & w64 & w65 \end{bmatrix} \text{x} \begin{bmatrix} x(t) \\ x(t-6) \\ x(t-12) \\ x(t-18) \\ b \end{bmatrix}$$

The decision vector c at the output of 1st hidden layer is obtained by passing activation vector a through log sigmoid function and hence is given by,

$$c = \begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{bmatrix} = \frac{1}{1+e^{-a}}$$

V is the weight matrix of the second hidden layer and is of the order $(l \times m)$ which is (3×6) in this network and given by,

$$V = \begin{bmatrix} v11 & v12 & v13 & v14 & v15 & v16 \\ v21 & v22 & v23 & v24 & v25 & v26 \\ v31 & v32 & v33 & v34 & v35 & v36 \end{bmatrix}$$

The activation vector d for the second hidden layer is of the order $(l \times 1)$ which is (3×1) and is given by,

$$d = \begin{bmatrix} d1 \\ d2 \\ d3 \end{bmatrix} = V \times c =$$

$$\begin{bmatrix} v11 & v12 & v13 & v14 & v15 & v16 \\ v21 & v22 & v23 & v24 & v25 & v26 \\ v31 & v32 & v33 & v34 & v35 & v36 \end{bmatrix} \times \begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{bmatrix}$$

The decision vector e at the output of the second hidden layer is obtained by passing activation vector d through log sigmoid function given by,

$$e = \begin{bmatrix} e1 \\ e2 \\ e3 \end{bmatrix} = \frac{1}{1+e^{-d}}$$

U is the weight matrix of the output layer and is of the order $(r \times l)$ which is (1×3) in this network and given by,

$$U = [u11 \quad u12 \quad u13]$$

The output y_{out} is a scalar and is given by,

$$y_{out} = U \times e = [u11 \quad u12 \quad u13] \times \begin{bmatrix} e1 \\ e2 \\ e3 \end{bmatrix}$$

This concludes the forward phase of the training and error computation in reverse direction is detailed in Section 3.2.

3.2 Reverse Phase:

In the backward phase the error is computed at each node of the network and is propagated back to front of the network and weight vectors are adjusted accordingly.

$y_{des} = x(t+6)$ is the desired output and the y_{out} (the actual output in forward phase) must align closer to y_{des}. Hence the error at the output layer (e_y) can be computed as,

$$e_y = y_{des} - y_{out}$$

The decision error (e_e) at the output of the second hidden layer is given by,

$$e_e = U' \times e_y = \begin{bmatrix} u11 \\ u21 \\ u31 \end{bmatrix} \times e_y$$

The activation error (e_d) at the second hidden layer is computed as,

$$e_d = e\,(1-e) \times e_e = \begin{bmatrix} e1 \\ e2 \\ e3 \end{bmatrix} \times (1 - \begin{bmatrix} e1 \\ e2 \\ e3 \end{bmatrix}) \times e_e$$

The decision error (e_c) at the ouput of the first hidden layer is given by,

$$e_c = V' \times e_d = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \\ v31 & v32 & v33 \\ v41 & v42 & v43 \\ v51 & v52 & v53 \\ v61 & v62 & v63 \end{bmatrix} \times e_d$$

The activation error (e_a) at the first hidden layer is computed as,

$$e_a = c\,(1-c) \times e_c = \begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{bmatrix} \times (1 - \begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{bmatrix}) \times e_c$$

This concludes the error computation in reverse direction and Section 3.3 details how the weights are updated so that network is ready to be used in testing.

3.3 Neural network Weight Updates:

As the weights are computed in reverse direction the weight deltas are computed and weights are updated accordingly. There are two specific parameters called weight gain (γ_g) and weight momentum (γ_m) which can be used to tune the priority of previous epoch delta weights and current epoch delta weights. The equations for delta weights are given by,

ΔU (epoch) = $\gamma_g \times e' \times e_y + \gamma_m \times \Delta U$ (epoch -1)
ΔV (epoch) = $\gamma_g \times c' \times e_d + \gamma_m \times \Delta V$ (epoch -1)
ΔW (epoch) = $\gamma_g \times X' \times e_a + \gamma_m \times \Delta W$ (epoch -1)

The weights are updated in ONLINE training process rather than batch training to increase the efficiency in predicting the time series. The weight update equations are given by,

$$U = U + \Delta U$$
$$V = V + \Delta V$$
$$W = W + \Delta W$$

The training is performed for 8000 epochs and the weights are updated online after each training input sample. The training algorithm (error backpropagation), the number of epochs and training method used (online training) can be easily justified for predicting the given time series as this is not only a simple approach but also efficient. The error backpropagation is chosen as all the activation functions used by neurons in the feedforward network are differentiable and the online training method resulted mean square error of the order of 10^{-5} which can be considered efficient.

4. RESULTS AND DISCUSSION

Figure 2 details the plot of time series prediction in MATLAB where the actual Mackey-Glass time series desired output is plotted in RED. The training output is in BLUE and is plotted for input samples from 1 to 600 and the testing output is represented in GREEN and is

plotted for predicted samples from 601 to 1201. Figure 2 shows that the training output in BLUE very closely coincides with the desired output RED and similarly the testing output in GREEN also very closely approximates the desired output in RED. Hence, the testing can be said to be successful.

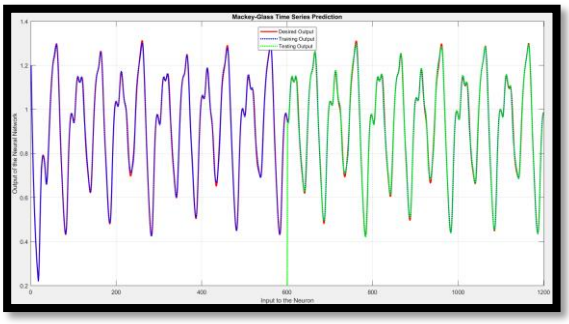

Fig. 2 Plot of Mackey glass time series prediction

Figure 3 details the mean square error obtained during the training procedure at each epoch and the mean square error decreases over the epochs. The total of 8000 epochs were sufficient and the code has achieved the mean square error of 5.2 x 10^{-5} during testing phase.

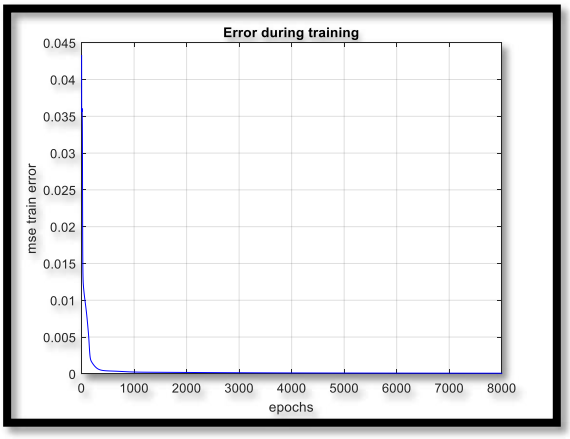

Fig. 3 Plot of MSE during training process

5. CONCLUSION

The Mackey-Glass time series can be efficiently predicted by multilayer perceptron network with error backpropagation algorithm. There are multiple different network topologies and different algorithms to predict this time series.

The mean square error achieved during testing phase was of the order 10^{-5} which is a very good measure to state the efficiency of the network architecture (mentioned in Fig.1) and the algorithm used detailed in Section 3.2. It is also important to consider the number of training epochs used to train a network which in this case was 8000 epochs that was enough to result in desired network. The network topology of $5 \times 6 \times 3 \times 1$ was one of the appropriate networks to achieve the above mentioned MSE with 8000 epochs. There are other tuning parameters such as learning gain and learning momentum which are tuned appropriately for this time series.

It can be concluded that the multilayer perceptron network with two hidden layers and error back propagation algorithm can predict this time series with reasonable computing power and simple architecture.

REFERENCES

[1] Kirchgässner Gebhard, Wolters Juergen, Hassler Uwe, (2007) *Introduction to Modern Time Series Analysis*, Springer Heidelberg, New York.

[2] Maindonald John, (2009) Time Series Analysis with Applications in R, *International Statistical Review*.

[3] Mackey M. C., Glass L., (1977) Oscillation and chaos in physiological control systems, *Science*, 197, pp. 287-289.

[4] Rajasekhar Nannapaneni, (2018) Prediction using Artificial Neural Networks, *Dell EMC Proven Professional Knowledge Sharing,* May, 2018, from https://education.emc.com/_content/_common/docs/ks_articles/2017KS_Rajasekhar-Prediction_using_Artificial_Neural_Networks.pdf

[5] Šíma J., (2001) The Computational Capabilities of Neural Networks. In: Kůrková V., Neruda R., Kárný M., Steele N.C. (eds) *Artificial Neural Nets and Genetic Algorithms*. Springer, Vienna.

[6] Jack V. Tu, (1996) Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, In *Journal of Clinical Epidemiology*, 49(11), pp. 1225-1231.

[7] Lewis N.D., (2017) Neural networks for time series forecasting with R, *CreateSpace Independent Publishing Platform*.

[8] Cai X., Zhang N., Venayagamoorthy G. K., Wunsch II D.C., (2007) Time series prediction with recurrent neural networks trained by a hybrid PSO-EA algorithm, *Neurocomputation*, 70(13-15), pp. 2342-2353.

[9] Chindanur Narendra Babu, Eswara B., (2015) Performance comparison of four new ARIMA-ANN prediction models on internet traffic data, *J. of Telecommunications and Information Technology,* 1, pp. 67-75.