Time Series Long-term Forecasting Using Tensor Product Functional Link Neural Network

Waddah Waheeb
School of Computing,
Asia Pacific University of Technology and Innovation
Kuala Lumpur, Malaysia
waddah.waheeb@staffemail.apu.edu.my

***Abstract*— In this paper, a tensor product functional link neural network (TP-FLNN) was applied on the Mackey-Glass chaotic time series in the long-term forecasting. The forecasting performance of TP-FLNN was compared with simple forecasting methods namely naive, drift and average methods. The obtained results show the possibility to consider TP-FLNN in long-term forecasting.**

Keywords*— functional link neural network, tensor product, Mackey-Glass chaotic time series, long-term forecasting

I. Introduction

Time series is a data which recorded sequentially over time. Hourly energy consumption, daily product sales, monthly sunspot numbers, and annual deaths from homicides are examples of time series data. Future forecasts of a time series can be done by using current and past observations.

Neural Network (NN) is a machine learning method which has been used in time series forecasting. NN is inspired by biological nervous systems; it has the ability to learn from data and adjust its learning parameters to build a model that can forecast future observations.

Various types of NNs methods have been applied for time series forecasting. Functional link neural network (FLNN) is a feedforward neural network with a single layer and a simple learning rule. The output is only a function of the current input. In FLNN, supplementary inputs are used to extend the structure of the network which helps to solve complex problems. The information in the FLNN is propagated through the network in a forward direction, from input neurons to output neurons.

Tensor product FLNN (TP-FLNN) is one type of FLNN where each element in the input vector is multiplied by other components in this input vector [1]. During learning, TP-FLNN uses historical data to build a forecasting model in order to forecast future observations.

TP-FLNN was used in time series forecasting [2,3]. However, no much attention has been paid to explore the efficiency of TP-FLNN in long-term forecasting. Therefore, the contributions in this paper are:

- To explore the efficiency of TP-FLNN in long-term forecasting.
- Comparison of the forecasting performance of TP-FLNN with three benchmarks forecasting methods, namely naive, drift and average methods in forecasting the Mackey-Glass time series.

The paper is organized as follows. A brief overview of TP-FLNN is given in Section. Section 3 presents the experimental design used in this paper. Results and discussion are presented in Section 4. Finally, the conclusion is given in section 5.

II. Related work

Functional link neural network (FLNN) was introduced by Giles and Maxwell [4]. FLNN extends the structure of the network by introducing extra inputs. The tensor product and functional expansion models are two common FLNN models [1]. In the tensor product FLNN model (TP-FLNN), each element in the input vector is multiplied by other elements in this input vector. On the another hand, the functional expansion model expands the dimension of the inputs by using a set of functions to deal with the problem at hand. However, it is not easy to determine which set of functions can be used to expand input dimensions. Therefore, in this paper, we only consider the TP-FLNN. Fig. 1 shows an example of a third-order TP-FLNN which has 3 external inputs and 4 high order inputs.

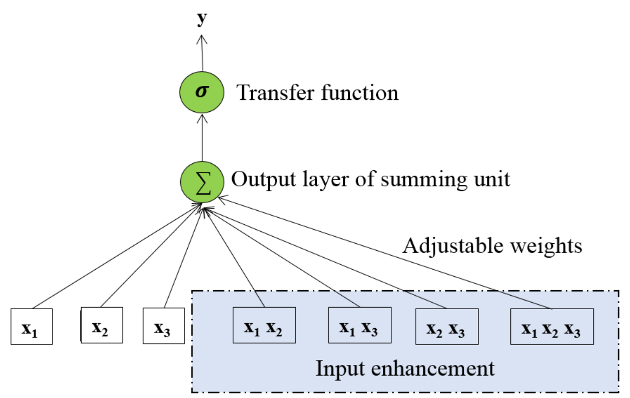

Fig. 1 Tensor product functional link neural network.

The learning algorithm for TP-FLNN is as follows:

For any given input,

- Calculate network output using Eq. (1):

$$y = \sigma(W_0 + \sum_i W_i x_i + \sum_i \sum_j W_{ij} x_i x_j + \sum_i \sum_j \sum_k W_{ijk} x_i x_j x_k) \tag{1}$$

where σ is a transfer function, W_0 are the biases, W_i, W_{ij} and W_{ijk} are weights between input and output nodes, x is a component of input vector X.

- Calculate weight changes:

$$\Delta W_i = \eta(d_i - y_i) y_i (1 - y_i) X_k \tag{2}$$

where η is the learning rate and *d* is the desired output.

- Update the weights:

$$W_i = W_i + \Delta W_i \tag{3}$$

- Continue until termination condition is satisfied.

III. Experimental Design

A. Time series used

The Mackey-Glass (MG) time-delay differential equation time series was used in this paper, which is found in mgdata.dat in MATLAB [5]. MG is given by the following equation:

$$\frac{dx}{dt} = \beta x(t) + \frac{\alpha x(t-\tau)}{1+x^{10}(t-\tau)} \tag{4}$$

where α=0.2, β=−0.1, x(0)=1.2, and τ=17. With this initial values the series shows chaotic behaviour.

The inputs, output, and the number of training and testing samples used in this paper are shown in Table I. Fig. 2 shows the interval used for training (i.e., blue points) and testing (i.e., red points).

TABLE I
MACKEY-GLASS TIME SERIES INFORMATION

Setting	Value
Input data	$x(t - 18), x(t - 12), x(t - 6), x(t)$
Output data	$x(t + 84)$
Training samples#	500
Out-of-sample samples#	500

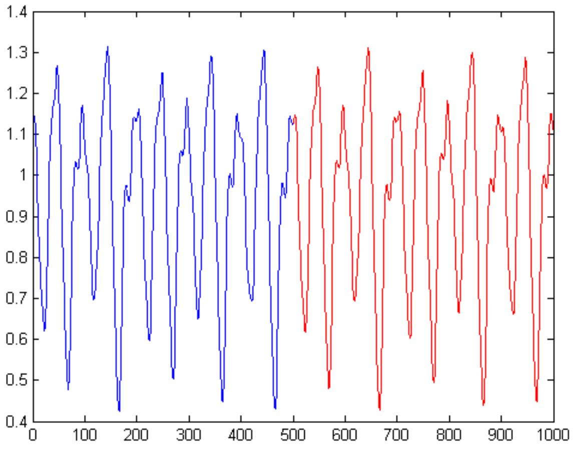

Fig. 2 Mackey-Glass time series.

TABLE II
NETWORK TOPOLOGY

Setting	Value
Learning rate	[0.01-1]
Momentum	[0.4-0.8]
Transfer function	Sigmoid function
Stopping criteria	• After accomplishing the 7th order network learning or, • Maximum number of epochs =3000 or, • MSE's threshold is 0.000001.

B. Network topology and training

Table II shows settings used to train TP-FLNN in this paper. Since sigmoid transfer function was used in this paper, the data was normalized to be within the new range is [0.2, 0.8] using the minimum and maximum normalization method, which given by Eq. (5):

$$\hat{x} = (\max{}_2 - \min{}_2) * \left(\frac{x - \min_1}{\max_1 - \min_1} \right) + \min{}_2 \tag{5}$$

where $\hat{x}$ is the normalized value of *x*, min_1 and max_1 are the minimum and maximum values of all observations, and min_2 and max_2 refer to the minimum and maximum values of the new range.

C. Performance Metrics

In this paper, the Root Mean Squared Error (RMSE) metric was used as performance metric. RMSE is given by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{6}$$

where N, y and ^y represent the number of out-of-sample data, actual output and network output, respectively.

IV. Results and Discussion

The simulation results for the 1-step forecasting of the time series used are presented in this section. Three simple forecasting methods were used for comparison, namely naive, drift and average methods [6].

As seen in Table III, RMSE testing of TP-FLNN is the lowest. The good results achieved by TP-FLNN compared to the simple forecasting method is due to the supplementary inputs extended with the structure of TP-FLNN which helps to produce more accurate results. It is good to note that the results presented in Table III are the de-normalized results.

The best forecasting for TP-FLNN using out-of-sample data is shown in Fig. 3. As seen in Fig. 3, to some extent TP-FLNN can follow the dynamics behavior of the time series. This shows the possibility to use TP-FLNN in long-term forecasting.

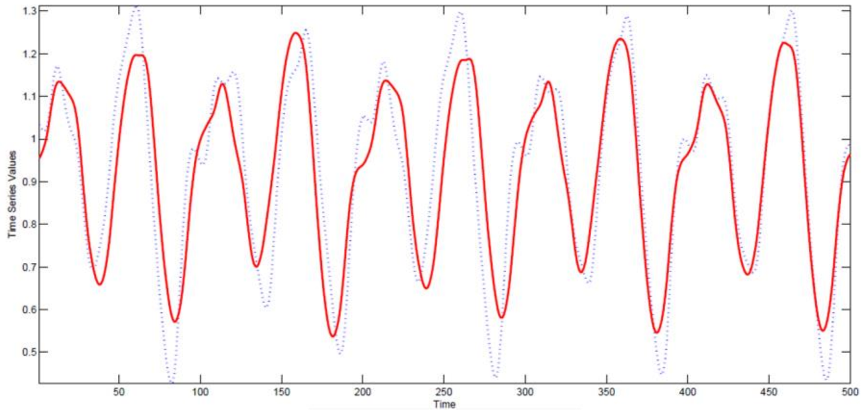

Fig. 3. Best out-of-sample forecasting for TP-FLNN.

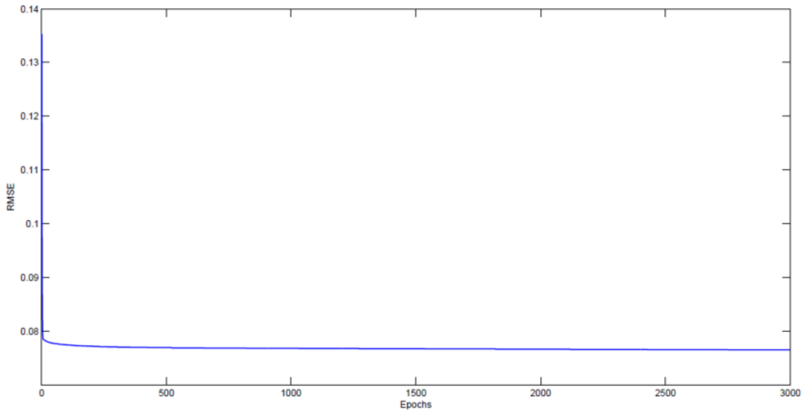

Fig. 4. Learning curves for the best TP-FLNN simulation with MG time series.

TABLE IIIII
BEST OUT-OF-SAMPLE FORECASTING RESULT COMPARISON.

Model	**RMSE**
Naive	0.2519
Drift	0.2550
Average	0.2282
TP-FLNN	**0.1021**

Finally, the evolution of RMSE during the learning of TP-FLNN is shown in Fig. 4. It can be seen from the figure that the learning curve for TP-FLNN is remarkably stable and RMSE continuously reduced.

V. CONCLUSIONS

This paper presents an evaluation of tensor product functional link neural network (TP-FLNN) for long-term forecasting. The simulations were conducted using the Mackey-Glass time-delay differential equation time series. This study compared TP-FLNN with naive, drift and average methods. Results showed that TP-FLNN can be considered for long-term forecasting.

With regards to future works, further comparison with other forecasting models is needed. In addition, more time series can be used for further analysis.

REFERENCES

[1] Y. Pao, "Adaptive pattern recognition and neural networks," 1989.

[2] W. Waheeb, and R. Ghazali, "Chaotic time series forecasting using higher order neural networks," International Journal on Advanced Science, Engineering and Information Technology 6, no. 5, pp. 624-629, 2016.

[3] W. Waheeb, and R. Ghazali, "A new genetically optimized tensor product functional link neural network: an application to the daily exchange rate forecasting," Evolutionary Intelligence 12, no. 4, pp. 593-608, 2019.

[4] C. L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks," Applied optics, vol. 26, pp. 4972-4978, 1987.

[5] MATLAB. [Online] Available: http://www.mathworks.com/examples/fuzzy-logic/mw/fuzzy-ex38166291-predict-chaotic-time-series.

[6] R. J. Hyndman, G. Athanasopoulos, Forecasting: principles and practice, OTexts, 2018.