



# Effects of direct input–output connections on multilayer perceptron neural networks for time series prediction

Yaoli Wang<sup>1</sup> · Lipo Wang<sup>2</sup> · Qing Chang<sup>1</sup> · Chunxia Yang<sup>1</sup>

Published online: 20 November 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

Feedforward neural network prediction is the most commonly used method in time series prediction. In view of the low prediction accuracy of the conventional BPNN model when the time series data contain a certain linear relationship, this paper describes a neural network approach for time series prediction, that is BPNN–DIOC (back-propagation neural network with direct input-to-output connections). Eight different datasets were used to verify the validity of BPNN–DIOC model in time series prediction. In this paper, the BPNN was extended to four variants based on the presence or absence of output layer bias and input-to-output connections firstly, and the prediction accuracy of eight datasets are analyzed by statistic method secondly. Finally, the experimental results demonstrate that the BPNN–DIOC has better prediction accuracy compared to the conventional BPNN while the output layer bias has no significant effect. Therefore, the input-to-output connections can significantly improve the prediction ability of time series.

**Keywords** Time series prediction · BPNN–DIOC · Linear relationship · Prediction accuracy

## 1 Introduction

Time series refers to the sequence of the same statistical indicators according to time sequence occurring within the same time interval and has the characteristics of large data volume, large noise, and fast data update. It is a very important and complex data that are widely existed in various fields, such as gross domestic product (GDP) (Jovic et al. 2019), stock price (Wang et al. 2011; Zhu and Wang 2010; Gupta and Wang 2010; Fang et al. 2014), population (Chi et al. 2019), unemployment rate (Li et al. 2014), traffic flow (Hou and

Mai 2013; Yang and Hu 2016), precipitation (Ramana et al. 2013), carbon dioxide concentration (Besteiro et al. 2017) and so on (Wang et al. 2001; 2001).

Time series prediction is forecasting the future data based on the existing historical data. Since the time series contains lots of information and rules, it is very important to find the hidden rules in this field and predict the unknown situation in advance more precisely (Taylor et al. 2006; Li and Shi 2010; Camara 2016). Through accurate forecasting results, people can arrange work and take measures ahead of time to prevent unfavorable situations and minimize losses (Samsudin et al. 2010). For example, stock prices forecast can enable investors to effectively avoid risks (Selvamuthu et al. 2019); predicting precipitation can enable people to do preventive work in advance (Devi et al. 2017); forecasting power load could provide certain decision-making direction for power participants (Bozkurt et al. 2017).

Time series prediction is to use statistical techniques and methods to establish a mathematical model which uses past values as input and future values as outputs, then find out the function that satisfies the changes of the sequence data. Subsequently, quantitatively estimate the future development trend of the data (Ding et al. 2008; Jia 2014). In previous time series prediction studies, most scholars first judged the attributes of the sequence data and then selected the appropriate model for prediction. If the time series data approximately

Communicated by L. Wang.

✉ Lipo Wang  
elpwang@ntu.edu.sg  
Yaoli Wang  
wangyaoli@tyut.edu.cn  
Qing Chang  
changqing@tyut.edu.cn  
Chunxia Yang  
1662010618@qq.com

<sup>1</sup> College of Information and Computer, Taiyuan University of Technology, 030600 Taiyuan, China

<sup>2</sup> School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

satisfy linear, a linear prediction method can be used, mainly including an autoregressive model, a moving average model, a self-sliding moving average model and so on. These models require linear functional relationships of future and historical values of time series; otherwise, the above linear prediction method will cause large prediction errors; therefore, the nonlinear method should be used when data satisfy nonlinear. However, the time series data collected under actual conditions is usually complex and nonlinear. Artificial neural network has the advantages of self-organization and strong nonlinearity in solving complex nonlinear problems (Li et al. 2013; Szoplik 2015; Doucoure et al. 2016). It can actively find the rule from the sample data and approximate the nonlinear function with arbitrary precision. These advantages of neural network make it obtain good prediction effect in nonlinear prediction, and it is widely used in time series prediction.

The above explains a time series with a linear or nonlinear relationship. When the nonlinear time series contains a certain non-negligible linear relationship, the BP neural network with highly nonlinear fitting characteristics may not be able to express the implicit relationship between the sample data perfectly and completely. On the contrary, it will lead to a decline in the prediction accuracy of time series and accuracy of the prediction result. To solve this problem, this paper adopts an improved BP neural network (BPNN–DIOC), which adds input-to-output connections based on BPNN. In addition, eight groups of time series datasets were used to compare the prediction performance of BPNN–DIOC network and BPNN.

## 2 Description of neural network

### 2.1 Back-propagation neural network (BPNN)

Artificial Neural Network (ANN) is a data processing system consisting of a large number of simple and highly interconnected processing elements (Cui et al. 2005). It is the abstraction and simulation of the human brain, which can imitate the human brain for complex parallel information processing, grasp the internal rules of the data and achieve a highly nonlinear mapping.

The most widely used artificial neural network is BPNN, which is a multilayer forward network based on error back-propagation (BP) learning algorithm. In the practical application of BPNN, the specific structure of BPNN should be determined first, namely the number of hidden layers and neurons required by the input layer, hidden layer and output layer. For determining the number of hidden layers: Kolmogorov theorem indicates that only three layers of BPNN can approximate any continuous function, so it is generally enough to select one layer of hidden layer (Hornik et al.

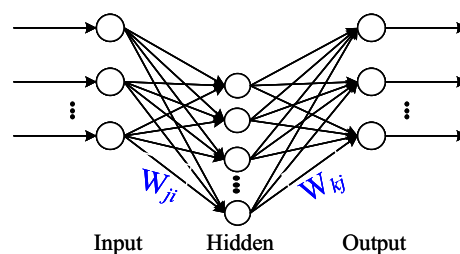


Fig. 1 The topology of BPNN

1989); for determining the number of neurons each layer, the number of input and output nodes depends on the dimension of the training sample, and there is no definite selection method to determine the number of nodes in the middle hidden layer.

Figure 1 shows the structure of BPNN, it can be seen that the BPNN is composed of three parts: the input layer, the hidden layer and the output layer. There is no connection between the neurons in the same layer and between the neurons in the non-adjacent layers and only a forward connection between neurons in adjacent layers. Obviously, the BPNN has an outstanding nonlinear mapping ability, which is shown that the relationship between input and output neurons is represented by  $n$  nonlinear terms ( $n$  is the number of neurons in the hidden layer). Therefore, the corresponding output of BPNN is:

$$O_k = \sum_{j=1}^n w_{kj} y_j + \gamma_k \quad (1)$$

$$y_j = f \left( \sum_{i=1}^m w_{ji} x_i + \theta_j \right) \quad (2)$$

where  $O_k$  is the output vector and  $y_j$  is the hidden layer output;  $n$  is the number of hidden nodes and  $m$  is the number of input layer neurons;  $w_{kj}$  is the weight between the hidden and output nodes;  $w_{ji}$  is the weight between the input to hidden nodes;  $\gamma_k$  is the threshold of the output layer neurons and  $\theta_j$  is the threshold of the hidden neurons;  $f$  is the transfer function of the hidden neurons.

### 2.2 Back-propagation neural network with direct input-to-output connections (BPNN–DIOC)

We have been using BPNN to achieve a nonlinear mapping between input and output. However, most problems are a combination of nonlinear and linear problems in real life, so BPNN may not be able to express the implicit relationship between input and output sample data completely and accurately.

In fact, not only the learning algorithm affects the prediction accuracy and generalization ability of the BPNN, but

also the network topology structure has a certain influence on the prediction performance. In other word, both learning algorithm and network topology structure have certain influence on prediction performance when BPNN is used for prediction and estimation. and the generalization ability of the network for unknown samples is also affected by the topology of the network.

### 2.2.1 Overview of previous work

Peng et al. (1992) proposed an improved neural network algorithm, including the combined representation of linear and nonlinear terms that map input to output. Pao et al. (1994) proposed a random vector functional-link (RVFL) network with a combination of random weights and functional links that has direct connection from input layer to output layer. Looney (1996) extended the radial basis function neural network (RBFNN) architecture to a more robust radial basis function link network (RBFLN), which also has the direct connection from input layer to output layer and can obtain more accurate results than RBF neural networks. However, such networks have not been fully researched and developed since then. Ren et al. (2016) and Zhang and Suganthan (2016) demonstrated that the RVFL network that adds input-to-output connections in RWSLFN can improve the network's generalization ability compared with RWSLFN network through the examples of prediction and classification, that is, the input-to-output connections in the network has a significant positive impact on the prediction effect of the network.

### 2.2.2 Structure of BPNN–DIOC

Inspired by the above work, BPNN–DIOC model is adopted in this paper, which improves the capability of BPNN with highly nonlinear fitting capability to solve the nonlinear and linear synthesis problems.

Figure 2 shows the structure of the BPNN–DIOC. Obviously, the BPNN–DIOC adds the direct linear input-to-output connections based on the conventional BPNN and reveals the combination function of the linear and nonlinear mapping of the input variables. Therefore, BPNN–DIOC shows the relationship between the input and output, which is expressed by  $m$  linear terms and  $n$  nonlinear terms approximately. Therefore, the corresponding output of BPNN–DIOC is:

$$O_k = \sum_{i=1}^m \beta_{ki} x_i + \sum_{j=1}^n w_{kj} y_j + \gamma_k \quad (3)$$

where  $\beta_{ki}$  is the linear connection weight from the input-to-output neurons, the remaining parameters are shown in Formula 1 and 2.

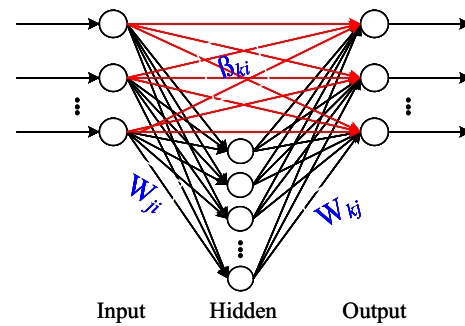


Fig. 2 The topology of BPNN–DIOC

Similar to BPNN, BPNN–DIOC uses training algorithms to adjust network parameters through an iterative process. Nevertheless, the main difference between these two models is that the direct input-to-output connections of the BPNN–DIOC simulates the linear components of the data compares to BPNN model.

## 3 Experimental configuration

### 3.1 Datasets

This paper selects 8 groups of common time series data to explore the performance of BPNN–DIOC model in time series prediction. Their statistics features include the length, min/max, median, average, and standard deviation of each datasets shown in Table 1.

### 3.2 Variations on BPNN

The difference between BPNN–DIOC and BP neural network is whether there is direct mapping between input layer and output layer. In this paper, four different network models are obtained based on whether input-to-output connections and output layer thresholds are added in the BPNN. The four different configurations of BPNN and their formulas are shown in Table 2. M1, M3 indicate that the input layer is not connected to the output layer.

M2, M4 model indicate that the input layer is connected to the output layer. In Table 2,  $h$  is the output of hidden layer neurons;  $O$  is the output of the output layer neurons;  $X$  is the input of the network;  $W_1$  is the connection weights from the input layer to the hidden layer;  $W_{21}$  is the connection weights from the input layer to the output layer;  $W_{22}$  is the connection weights from the hidden layer to the output layer;  $\theta$  is the threshold of the hidden neurons;  $\beta$  is the threshold of the output neurons;  $f$  is the transfer function.

**Table 1** Summary of the eight time series datasets

Sequence number	Name	Length	Min	Median	Max	Mean	SD
1	CO2 concentration	192	317.25	328.295	341.19	328.464	5.9627
2	Milk production	168	553	761	969	754.7083	102.2045
3	Atvinnulausir eftir menntun	185	1441	5112	16822	6407	4538
4	Number of unemployed	414	0.1515	2.1021	9.2871	2.7553	2.2585
5	Monthly lake erie levels	600	10	14.858	20	14.9931	2.0121
6	Social media payroll index	207	107078	108714	115068	109384	2205
7	Electricity load demand	1440	5676.57	7811.27	10398.73	7717.34	895.9621
8	Electricity price	1200	26.33	64.44	410.39	80.0491	49.5126

**Table 2** BPNN with different configurations

Model	Output layer bias	Input-to-output connections	Formula
M1	×	×	$h = f(\sum XW_1 + \theta)$ $O = \sum W_{22}h$
M2	×	✓	$h = f(\sum XW_1 + \theta)$ $O = \sum W_{21}X + \sum W_{22}h$
M3	✓	×	$h = f(\sum XW_1 + \theta)$ $O = \sum W_{22}h + \beta$
M4	✓	✓	$h = f(\sum XW_1 + \theta)$ $O = \sum W_{21}X + \sum W_{22}h + \beta$

### 4 Assessment on eight time series data

Time series prediction is to speculate the future value based on historical data. If time series is  $\{x_n\}$ , its general form can be described as:

$$x_{n+k} = f(x_n, x_{n-1}, \dots, x_{n-(m-1)}) \tag{4}$$

where  $k$  is the number of prediction steps;  $m$  represents the input dimension of the model. When  $k = 1$ , it is the simplest single-step prediction; when  $k > 1$ , it is the multi-step prediction. This article only discusses single-step prediction of time series, that is, multiple time steps that are used for rolling predict the next time step.

#### 4.1 Select input and output variables

In this paper, 8 datasets are selected, of which dataset 1, dataset 2, dataset 3, dataset 4 and dataset 5 are monthly datasets, i.e., one data per month; dataset 6 is a weekly dataset, i.e., one data per week; dataset 7 and dataset 8 are one data every half hour. The input and output pattern of the sample is shown in Table 3.

According to the control variable method, the same initial conditions were used for different models to remove the influence of the initial conditions on the experimental results. The number of neurons in the hidden layer was tested from

**Table 3** The input and output patterns for neural network training

	Input	Output
1	$x_1 \sim x_{12}$	$x_{13}$
2	$x_2 \sim x_{13}$	$x_{14}$
3	$x_3 \sim x_{14}$	$x_{15}$
⋮	⋮	⋮
n	$x_n \sim x_{n+11}$	$x_{n+12}$

1:30 to find out the best accuracy of the test set and obtain the number of neurons in the hidden layer under the optimal precision of the test set. Due to the randomness of neural network training, each network structure was trained 10 times, and then the average prediction accuracy of the test set was calculated. Finally, the optimal topology structure was obtained for rolling prediction in 8 datasets.

#### 4.2 Error measures

There are too many factors that affect data, including predictability, unknown and all kinds of unexpected conditions. Therefore, errors will inevitably occur in the prediction work. Errors are impossible to eliminate, only try to improve the prediction method or learning algorithm to reduce them. In this paper, in order to analyze the prediction effect of the four different neural networks, Root Mean Square Error (RMSE)

**Table 4** Weights and thresholds of linear neural network after training

	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5	Dataset6	Dataset7	Dataset8
$w_1$	0.1444	0.2153	-0.0095	-0.025	0.1155	0.0711	-0.0116	-0.0899
$w_2$	0.2215	0.1339	-0.0118	0.2088	0.1009	0.0829	0.1044	-0.0558
$w_3$	0.0666	-0.1163	0.0849	0.1471	0.1379	0.1197	-0.0545	0.0559
$w_4$	0.0114	0.2882	-0.1273	0.0237	-0.0574	-0.0538	-0.0753	-0.1183
$w_5$	0.0527	0.0641	0.1034	0.1207	0.185	0.0539	-0.0491	0.127
$w_6$	-0.0195	-0.1267	-0.1403	-0.0002	-0.015	0.0826	0.0597	-0.0954
$w_7$	-0.0876	-0.0416	-0.0647	-0.1145	-0.112	-0.0396	-0.0706	0.0125
$w_8$	0.0817	0.0742	0.2284	0.1286	-0.0589	0.0486	0.1056	0.1443
$w_9$	0.1234	-0.1944	0.0279	-0.0352	0.0823	0.1766	0.0383	0.1129
$w_{10}$	0.0503	0.0851	0.0742	0.1503	0.0844	0.1594	0.0291	0.0553
$w_{11}$	0.0992	0.2279	0.4522	0.125	0.1113	0.1194	0.2518	0.2076
$w_{12}$	0.2233	0.3322	0.3341	0.236	0.2845	0.0559	0.2699	0.4408
$b$	0.0371	0.0531	0.0232	0.0141	0.077	0.1424	0.2794	0.029

and Mean Absolute Percentage Error (MAPE) are used to measure the predictive performance of the network. They are defined as shown in Eqs. 6 and 7.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{k=1}^n (T_k - O_k)^2} \quad (5)$$

$$\text{MAPE} = \frac{1}{n} \sum_{k=1}^n \left| \frac{T_k - O_k}{T_k} \right| \times 100 \% \quad (6)$$

where  $T$  is the target vector,  $O$  is the output vector and  $n$  is the length of data. MAPE is an extension of MSE and MAPE is the preferred measure for industry practitioners.

### 4.3 Prediction results and analysis

For each time series, the first 70% was used for training and the remaining 30% was used for testing. Due to the randomness of neural network training, each network structure was trained 10 times.

#### 4.3.1 Linear analysis

The linear neural network has a similar structure as the single-layer perceptron, which is also composed of the input layer and the output layer, and the neurons in the output layer have the ability of information processing. The only difference between them is that the activation function of the perceptron is a hard transfer function, while the linear neuron uses the linear transfer function purelin, so the output of the linear neural network can be arbitrary value, rather than only two values. The output of linear neural network can be calculated by the following formula 7:

$$y = \text{purelin}(v) = \text{purelin}(\mathbf{w} \cdot \mathbf{p} + b) = \mathbf{w} \cdot \mathbf{p} + b \quad (7)$$

It can be seen from the above formula that the linear neural network can be approximated as a linear function, but cannot complete the calculation of approximating a nonlinear function.

In order to analyze whether there is a linear factor in the system, this paper uses linear neural networks to predict time series first. Table 4 shows the weights and thresholds of the network after the training of each dataset, that is to say, for each dataset, it can be expressed as a linear relationship as shown in Equation 9.

$$x_{13} = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_{12}x_{12} + b \quad (8)$$

#### 4.3.2 Performance evaluation between different models

In this paper, the experiments are simulated on MATLAB 2012a version. For four different BPNN variant networks, the transfer functions of the hidden layer and the output layer are logsig and purelin, respectively, and the network training function is traingda. Set the number of hidden neurons to 1–30 and train the network in turn, and then the average value of RMSE and MAPE was taken as the prediction precision of each network structure. Furthermore, in order to compare the results of the experiment, the four different models of each datasets were trained in the same initial conditions.

#### Hidden layer nodes Optimization

The experiment is repeated by adjusting the number of hidden nodes under the other parameters remain unchanged and the optimal number of hidden neurons is determined according to the minimum output error in the training process. The number of optimal hidden neurons required for each dataset of four different network structures is shown in Table 5. Figures 3 and 4 show the RMSE and MAPE change

**Table 5** Optimal hidden nodes

	M1	M2	M3	M4
Dataset 1	8	1	11	1
Dataset 2	12	4	11	3
Dataset 3	13	2	13	4
Dataset 4	18	1	18	1
Dataset 5	13	2	13	1
Dataset 6	18	2	17	3
Dataset 7	9	1	12	1
Dataset 8	7	2	8	2

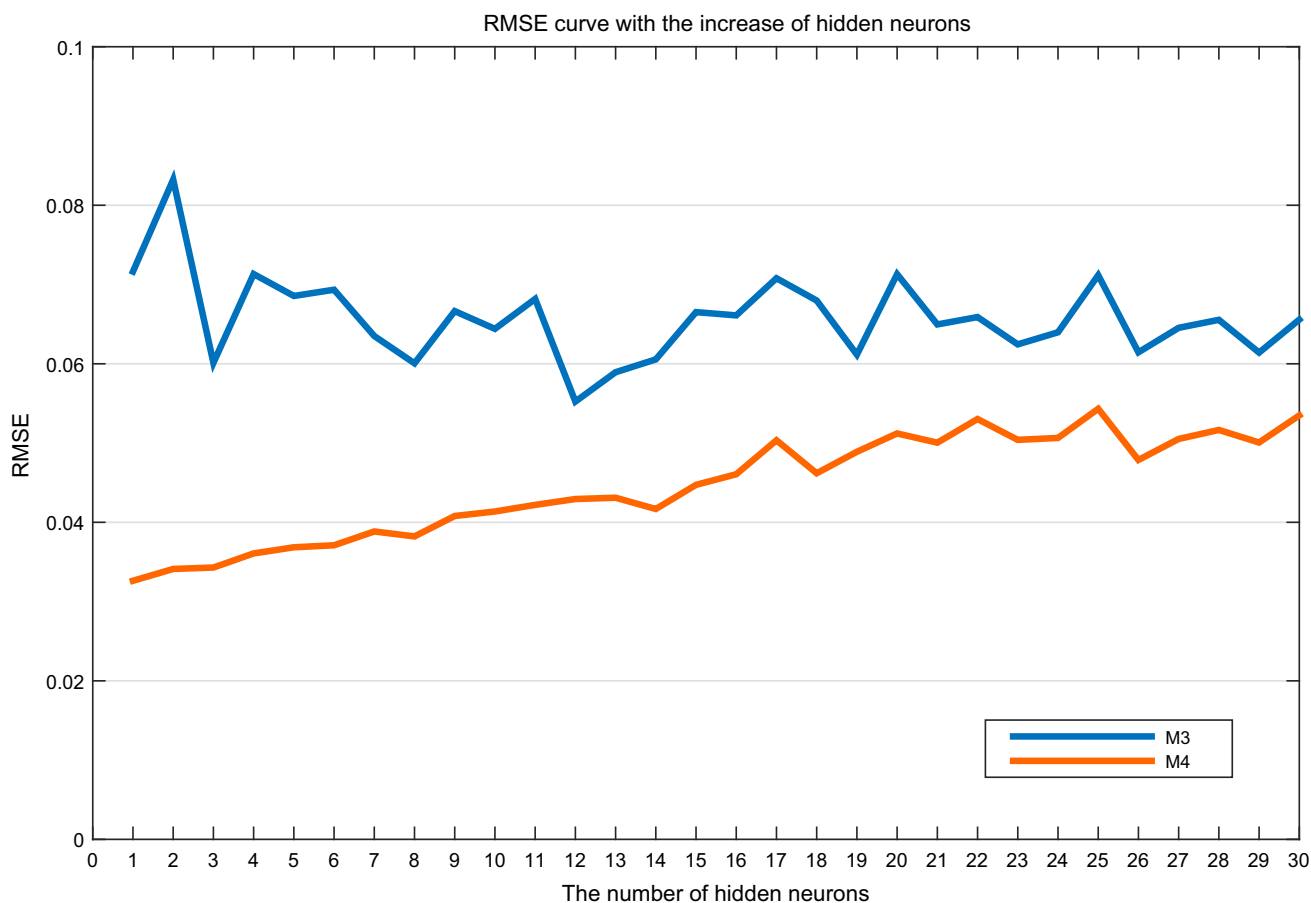
curves of dataset 7 with the increase in hidden neurons during the training of M3 and M4 models, respectively.

Apparently, the number of hidden neurons needed of BPNN–DIOC model is much less than the conventional BPNN. So it is useful to add that input-to-output connections based on BPNN reduce the number of neurons required by the hidden layer and delete some input-to-hidden weights that are less importance for training results of network. Consequently, the BPNN–DIOC model simplifies the network

structure greatly and reduces the amount of weight adjustment.

### Performance Optimization

The prediction results of 8 datasets using linear neural networks are shown in the right column of Table 6. The performance of four models was measured by RMSE and MAPE, and the average RMSE and MAPE is tabulated in Table 6. It can be seen that, compared with BPNN, 1–8 data sets with input and output connected network of RMSE and MAPE decreased significantly. However, there is no significant difference in the prediction results of the network with or without the output threshold. The prediction structure of the linear neural network is shown in Fig. 5. Figure 6 is the improved percentage of RMSE for BPNN–DIOC and the RMSE for linear neural network. It can be seen from Table 6 and Fig. 6, in dataset 1, the prediction result using linear neural network is similar to that of BPNN, indicating that there is a certain linear predictability between the data. Moreover, BPNN–DIOC greatly improves the prediction accuracy compared with BPNN, and the RMSE is reduced from 0.0992 to 0.0329. In dataset 2, the prediction result using linear neural

**Fig. 3** The RMSE curve of M3 and M4 models in dataset 7



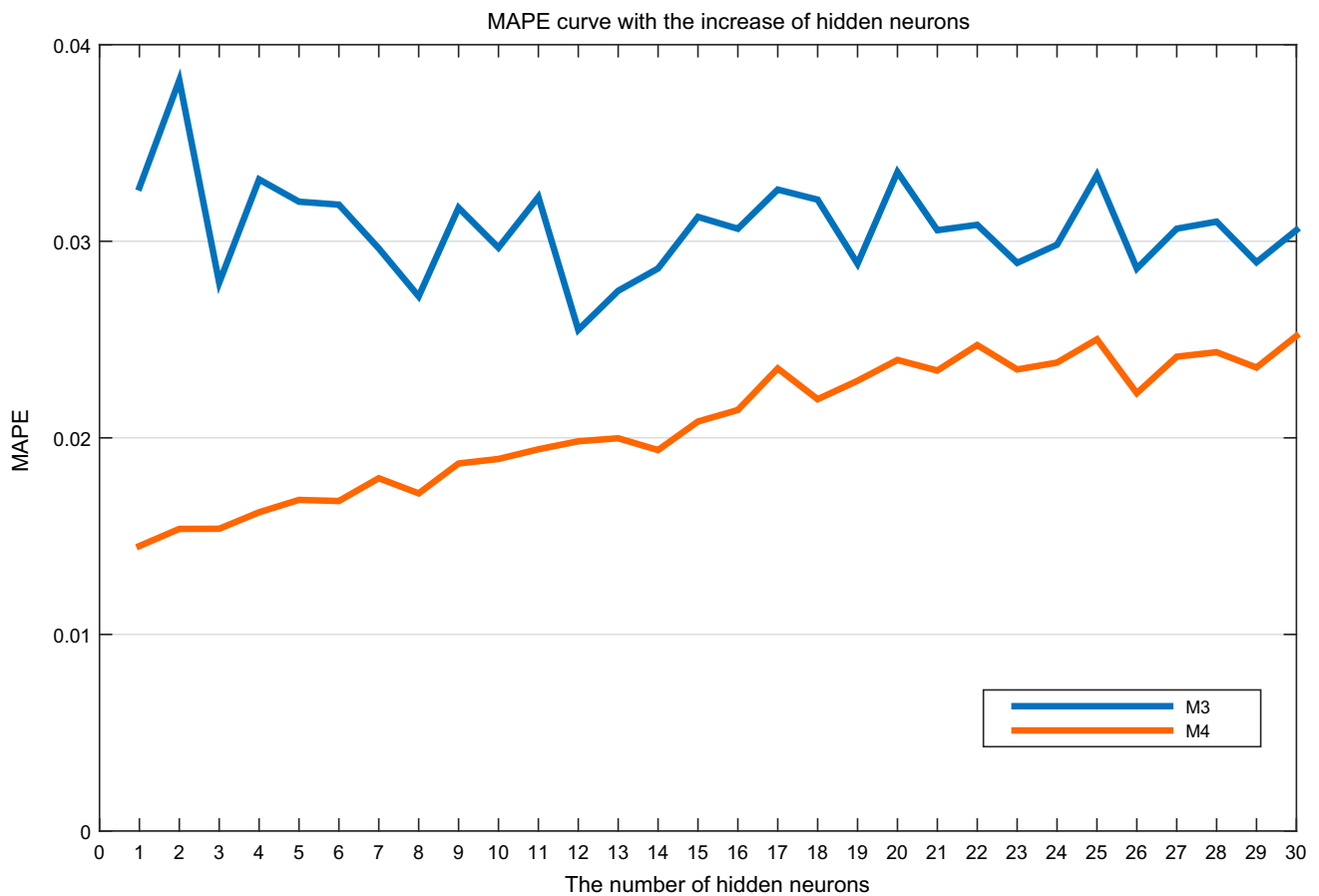


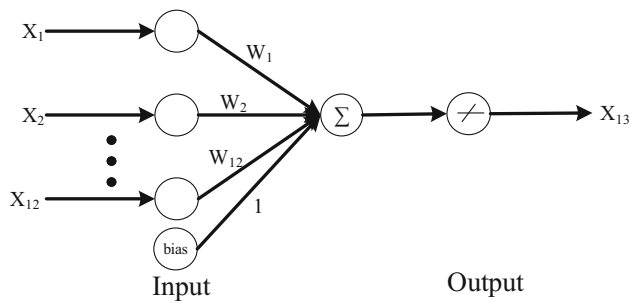
Fig. 4 The MAPE curve of M3 and M4 models in dataset

Table 6 The average RMSE and MAPE of four different models and linear neural network

	M1		M2		M3		M4		Linear neural network	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Dataset 1	0.1003	0.0062	0.0319	0.0019	0.0992	0.0061	0.0329	0.002	0.1041	0.0062
Dataset 2	0.0803	0.0323	0.0667	0.0269	0.0821	0.0324	0.0662	0.0274	0.1437	0.0563
Dataset 3	0.0659	0.1094	0.0385	0.0614	0.0653	0.1089	0.0383	0.0635	0.0817	0.1315
Dataset 4	0.1357	0.1914	0.106	0.1626	0.1348	0.19	0.107	0.1604	0.1527	0.2233
Dataset 5	0.0581	0.0323	0.0423	0.0227	0.0592	0.0335	0.0427	0.0232	0.1231	0.0678
Dataset 6	0.0765	0.0054	0.0536	0.0038	0.0784	0.0056	0.0517	0.0036	0.0893	0.0063
Dataset 7	0.0525	0.0242	0.0318	0.0141	0.0552	0.0225	0.0326	0.0145	0.1078	0.0517
Dataset 8	0.0763	0.2286	0.0687	0.1722	0.0761	0.2141	0.0678	0.167	0.0891	0.2465

network is poor, which is quite different from the prediction result of BPNN, indicating that there is no obvious linear predictability between the data. Moreover, BPNN–DIOC has a smaller improvement in prediction accuracy compared with BPNN, and the RMSE is reduced from 0.0821 to 0.0662. In dataset 3, the prediction result using linear neural network is similar to that of BPNN, indicating that there is a certain linear predictability between data. Moreover, BPNN–DIOC greatly improves the prediction accuracy compared

with BPNN, and the RMSE is reduced from 0.0653 to 0.0383. In dataset 4, the prediction result using linear neural network is similar to that of BPNN, indicating that there is a certain linear predictability between data. Moreover, BPNN–DIOC greatly improves the prediction accuracy compared with BPNN, and the RMSE is reduced from 0.1348 to 0.1070. In dataset 5, the prediction result using linear neural network is poor, which is quite different from the prediction result of BPNN, indicating that there is no obvious linear predictabil-



**Fig. 5** Prediction structure of linear neural network

ity between the data. Moreover, BPNN–DIOC has a smaller improvement in prediction accuracy compared with BPNN, and the RMSE is reduced from 0.0592 to 0.0427. In dataset 6, the prediction result using linear neural network is similar to that of BPNN, indicating that there is a certain linear predictability between data. Moreover, BPNN–DIOC greatly improves the prediction accuracy compared with BPNN, and the RMSE is reduced from 0.0784 to 0.0517. Since datasets 7–8 are collected every half hour, the 12-dimensional input and 1-dimensional output structure cannot well interpret the relationship between the data. Therefore, the prediction result of linear neural network has little relationship with the prediction results of BPNN–DIOC, but it can be seen that the prediction result of BPNN–DIOC is still superior to that of BPNN. Therefore, for data that have a linear relationship, BPNN–DIOC plays an important role in time series prediction, it can obtain better prediction accuracy than BPNN.

The Wilcoxon signed-ranks test is a non-parametric alternative to the paired  $t$  test, which ranks the differences in

performances of two classifiers for each data set, ignoring the signs, and compares the ranks for the positive and the negative differences. The differences are ranked according to their absolute values. Ranks of  $d_i = 0$  are split evenly among the sums; if there is an odd number of them, one is ignored. Let  $N$  be the number of pairs.

$$R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i) \quad (9)$$

$$R^- = \sum_{d_i < 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i) \quad (10)$$

$$T = \min(R^+, R^-) \quad (11)$$

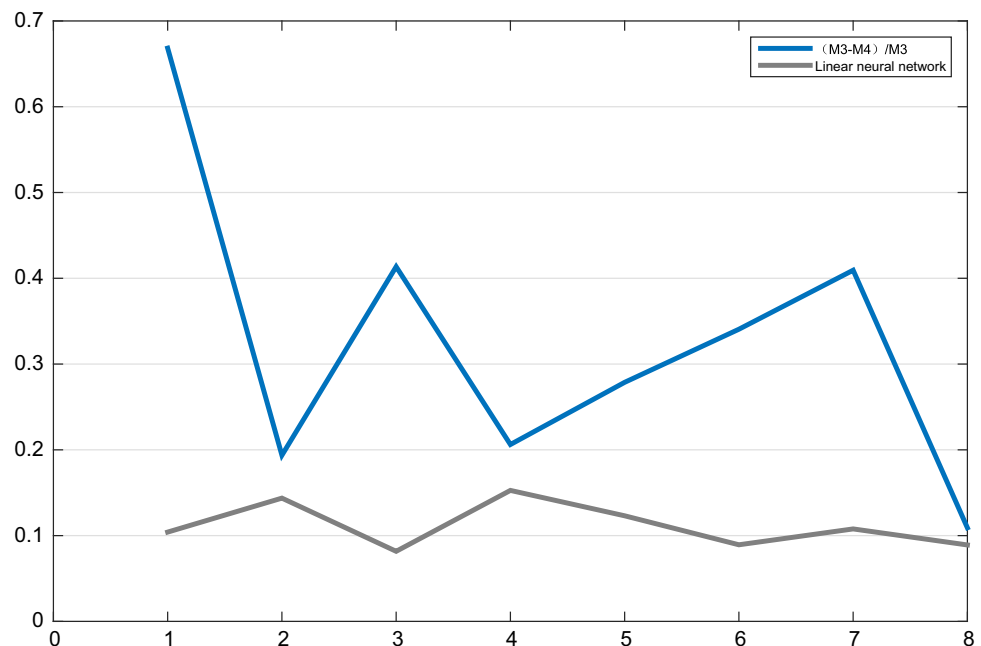
$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}}, \quad (N(N+1)/2) > 25 \quad (12)$$

With  $\alpha = 0.05$ , the null-hypothesis can be rejected if  $z$  is smaller than a given value.

In order to explore whether the output bias has a significant effect on prediction results, we applied the Wilcoxon signed-rank test on the two pairs: M1 and M3, M2 and M4. The  $p$  values shown in Tables 7 and 8 are bigger than 0.05, indicating that the output bias has no significant effect on prediction effect.

In order to explore whether the input-to-output connections has a significant effect on prediction results, we applied the Wilcoxon signed-rank test on the two pairs: M1 and M2, M3 and M4. The  $p$  values shown in Tables 7 and 8 are smaller

**Fig. 6** Improved percentage of RMSE for BPNN–DIOC and the RMSE for linear neural network





**Table 7** Wilcoxon signed-rank test of the BPNN whether has the output bias

Dataset 1–8	RMSE	MAPE
M1 versus M3	0.3281	0.2656
M2 versus M4	0.9141	0.9219

**Table 8** Wilcoxon signed-rank test of the BPNN whether has the input-to-output connections

Dataset 1–8	RMSE	MAPE
M1 versus M2	0.0078	0.0078
M3 versus M4	0.0078	0.0078

than 0.05, indicating that the input-to-output connections has a significant effect on prediction effect.

In general, BPNN–DIOC can improve the prediction accuracy owing to the input-to-output connections. It strengthens the description of the linear relationship in the time series data of the entire network and improves the generalization ability of the network.

## 5 Conclusion

When the target system contains linear component, the traditional method is to use BPNN network with highly nonlinear fitting characteristics to approximate the system. There is no doubt that the effect of using nonlinear network to approximate linear system is worse than that of linear model. Therefore, this paper adopts BPNN–DIOC for time series prediction, joining a linear connection between the input layer and the output layer of the based BPNN, a linear and nonlinear combined network is formed to enhance the generalization ability of the network and fully express the implicit relationship between the input and output samples. This paper discusses the influence on the input-to-output connections and output layer bias on the prediction results based on 8 sets of datasets, the following conclusions can be drawn:

1. During network training, BPNN–DIOC can reduce the number of neurons required by the hidden layer by adding input-to-output connections than BPNN, it deletes some input-to-hidden weights that are less important for training results of network. The total number of connections in BP networks could be reduced if  $(m + q) \times p > m \times q$ , where  $m$  is the number of input layer nodes,  $q$  is the number of output layer nodes and  $p$  is the reduction in the number of the hidden layer nodes. So the BPNN–DIOC could simplify the network structure greatly.
2. The direct input-to-output connections can improve the prediction accuracy significantly. Moreover, the better

linear fitting of the data, the better prediction effect of BPNN–DIOC. However, the output bias has no significant effect on network prediction result.

3. The prediction result of linear neural network has relationship with the prediction results of BPNN–DIOC. In general, BPNN–DIOC plays an important role in time series prediction for data that has a linear relationship, which can obtain a better prediction accuracy than BPNN.

Therefore, adding a connection from input-to-output based on BPNN can map the input to the output of the network more completely and describe the characteristics of the time series data more accurately. Thence, the BPNN–DIOC network provides a more general framework for prediction model.

**Acknowledgements** This study was funded by Natural Science Foundation of Shanxi Province (Grant No. 201801D121141) and Joint Research Fund for Overseas Chinese Scholars and Scholars in Hong Kong and Macao (Grant No. 61828601).

## Compliance with ethical standards

**Conflict of interest** Lipo Wang is Guest Editor of SI: ICNC-FSKD 2017. Other authors have no conflict of interest.

**Human participants and/or animals** This research does not involve human nor animals.

**Informed consent** I consent the journal to review the paper. I inform that the manuscript has not been submitted to other journal for simultaneous consideration. The manuscript has not been published previously. The study is not split up into several parts to increase the quantity of submissions and submitted to various journals or to one journal over time. No data have been fabricated or manipulated (including images) to support my conclusions. No data, text, or theories by others are presented as if they were of my own. Proper acknowledgements to other works are provided, and I use no material that is copyrighted. I consent to submit the paper, and I have contributed sufficiently to the scientific work and I am responsible and accountable for the results.

## References

- Besteiro R, Arango T, Ortega JA, Rodríguez MR, Fernández MD, Velo R (2017) Prediction of carbon dioxide concentration in weaned piglet buildings by wavelet neural network models. *Comput Electron Agric* 143:201–207
- Bozkurt Biricik G, Tayşi ZC (2017) Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market. *PLoS One* 12(4):e0175915
- Camara A (2016) Time series forecasting using statistical and neural networks models. LAP LAMBERT Academic Publishing, Berlin
- Chi G, Wang D, Hagedorn AD (2019) Future interstate highway system demands: predictions based on population projections. *Case Stud Transp Policy* 7(2):384–394
- Cui X, Potok TE, Palathingal P (2005) Document clustering using particle swarm optimization. In: *Proceedings 2005 IEEE swarm intelligence symposium 2005, SIS 2005*, pp 185–191

- Devi SR, Arulmozhiarman P, Venkatesh C (2017) ANN based rainfall prediction—a tool for developing a landslide early warning system. *Adv Cult Living Landslides* 3:175–182
- Ding G, Zhong SS, Li Y (2008) Time series prediction using wavelet process neural network. *Chin Phys B* 17(6):1998–2003
- Doucoure B, Agbossou K, Cardenas A (2016) Time series prediction using artificial wavelet neural network and multi-resolution analysis: application to wind speed data. *Renew Energy* 92:202–211
- Fang Y, Fataliyev K, Wang LP, Fu XJ, Wang Y (2014) Improving the genetic-algorithm-optimized wavelet neural network approach to stock market prediction. In: 2014 International joint conference on neural networks (IJCNN 2014), pp 3038–3042
- Gupta S, Wang LP (2010) Stock forecasting with feedforward neural networks and gradual data sub-sampling. *Aust J Intell Inf Process Syst* 11:14–17
- Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2(5):359–366
- Hou Y, Mai Y (2013) Chaotic prediction for traffic flow of improved BP neural network. *Indones J Electr Eng Comput Sci* 11(3):1682–1690
- Jia J (2014) Financial time series prediction based on BP neural network. *Appl Mech Mater* 631–632:31–34
- Jovic S, Miladinovic JS, Micic R, Markovic S, Rakic G (2019) Analysing of exchange rate and gross domestic product (GDP) by adaptive neuro fuzzy inference system (ANFIS). *Phys A* 513:333–338
- Li G, Shi J (2010) On comparing three artificial neural networks for wind speed forecasting. *Appl Energy* 87(7):2313–2320
- Li S, Hao Q, Yue Y, Liu H (2013) Prediction for chaotic time series of optimized BP neural network based on modified PSO. *Comput Eng Appl* 49(6):697–702
- Li Z, Xu W, Zhang L, Lau RYK (2014) An ontology-based web mining method for unemployment rate prediction. *Decis Support Syst* 66:114–122
- Looney CG (1996) Radial basis functional link nets as learning fuzzy systems. RES report. University of Nevada, Department of Computer Science
- Pao YH, Park GH, Sobajic DJ (1994) Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* 6(2):163–180
- Peng TM, Hubele NF, Karady GG (1992) Advancement in the application of neural networks for short-term load forecasting. *IEEE Trans Power Syst* 7(1):250–257
- Ramana RV, Krishna B, Kumar SR, Pandey NG (2013) Monthly rainfall prediction using wavelet neural network analysis. *Water Resour Manage* 27(10):3697–3711
- Ren Y, Suganthan PN, Srikanth N, Amaratunga G (2016) Random vector functional link network for short-term electricity load demand forecasting. *Inf Sci* 367:1078–1093
- Samsudin R, Shabri A, Saad P (2010) A comparison of time series forecasting using support vector machine and artificial neural network model. *J Appl Sci* 10(11):950–958
- Selvamuthu D, Kumar V, Mishra A (2019) Indian stock market prediction using artificial neural networks on tick data. *Financ Innov* 5(1):16
- Szoplík J (2015) Forecasting of natural gas consumption with artificial neural networks. *Energy* 85:208–220
- Taylor JW, de Menezes LM, McSharry P (2006) A comparison of univariate methods for forecasting electricity demand up to a day ahead. *Int J Forecast* 22(1):1–16
- Teo KK, Wang LP, Lin ZP (2001) Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization. In: Computational science—ICCS 2001, proceedings Pt 2. vol 2074, pp 310–317
- Wang JZ, Wang JJ, Zhang ZG, Guo SP (2011) Forecasting stock indices with back propagation neural network. *Expert Syst Appl* 38(11):14346–14355
- Wang LP, Teo KK, Lin ZP (2001) Predicting time series with wavelet packet neural networks. In: 2001 IEEE international joint conference on neural networks (IJCNN 2001). pp 1593–1597
- Yang H, Hu X (2016) Wavelet neural network with improved genetic algorithm for traffic flow time series prediction. *Optik Int J Light Electron Opt* 127(19):8103–8110
- Zhang L, Suganthan PN (2016) A comprehensive evaluation of random vector functional link networks. *Inf Sci* 367:1094–1105
- Zhu M, Wang LP (2010) Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms. In: 2010 International joint conference on neural networks (IJCNN 2010)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.