ATLANTIS PRESS
Advances in Social Science, Education and Humanities Research (ASSEHR), volume 106
International Conference on Education Science and Economic Management (ICESEM 2017)

Prediction of Short Term Exchange Rate Using BP Neural Network

Jun Liu
Department of Information Science and Technology
Tianjin University of Finance and Economics
Tianjin 300222, P.R.China
liujun@tjufe.edu.cn

***Abstract*—Exchange rate prediction accuracy is often about the survival of an enterprise or even national economic environment security. The effect of predicting the short-term exchange rate with general statistical method is not ideal. This paper proposed a multi-layer BP neural network model to predict short-term exchange rate. We conducted simulation experiments in Matlab 2010b, using data from China Merchants Bank foreign exchange market analysis software V2.8. The results show a better goodness of fit. It indicates that BP neural network can be of practical use in predicting short term exchange rate with high rate of accuracy.**

Keywords—BP neural network; Exchange rate prediction; Time series; Matlab simulation

I. INTRODUCTION

As the global economic integration trend to strengthen gradually, the volatility of exchange rate becomes the ties and bridges of international economic exchanges and occupies more and more important position. The exchange rate has become an important comprehensive price index with international economic transactions. The appropriate exchange rate can promote the growth of the national economy, keep the good reputation of the country, maintain a stable price level, and improve the international competitiveness.

The accuracy of exchange rate prediction is often about the survival of an enterprise or even national economic environment security. The sooner you make predictions, you can take a preemptive opportunity on the deal. But a slight deviation will bring great economic losses. Therefore, research on intrinsic nature of the currency market and price fluctuation of exchange rate prediction is of great theoretical significance and application value.

II. BP NEURAL NETWORK

The basic principle of BP neural network model is through the establishment of a lot of mapping relation, by Back Propagation, constantly adjust model within the weights and thresholds to minimize the error of the network within acceptable error range, so as to establish the training sample and make the network training to the ideal state.

Multilayer BP Neural Network is LMS (Further Mean Square) algorithm. In the case of multiple output, we get general form LMS algorithm [1]:

$$F(\mathbf{x}) = E[\mathbf{e}^T\mathbf{e}] = E[(\mathbf{t}-\mathbf{a})^T(\mathbf{t}-\mathbf{a})] \tag{1}$$

When using the k iteration of mean square error instead of the mean square error expectations, we get:

$$\hat{F}(\mathbf{x}) = \left(\mathbf{t}(k)-\mathbf{a}(k)\right)^T\left(\mathbf{t}(k)-\mathbf{a}(k)\right) = \mathbf{e}^T(k)\mathbf{e}(k) \tag{2}$$

The approximate mean square error of the steepest descent method is:

$$\begin{aligned} w_{i,j}^m(k+1) &= w_{i,j}^m(k) - \alpha\frac{\partial\hat{F}}{\partial w_{i,j}^m} \\ b_i^m(k+1) &= b_i^m(k) - \alpha\frac{\partial\hat{F}}{\partial b_i^m} \end{aligned} \tag{3}$$

In order to simplify the expression, we define:

$$s_i^m = \frac{\partial\hat{F}}{\partial n_i^m} \tag{4}$$

There are:

$$\frac{\partial\hat{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1}, \frac{\partial\hat{F}}{\partial b_i^m} = s_i^m \tag{5}$$

The equation (5) into equation (3), get:

$$\begin{aligned} w_{i,j}^m(k+1) &= w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \\ b_i^m = (k+1) &= b_i^m(k) - \alpha s_i^m \end{aligned} \tag{6}$$

Application of Jacobian matrix:

$$\frac{\partial n^{m+1}}{\partial n^m} \equiv \begin{pmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \cdots & \frac{\partial n_1^{m+1}}{\partial n_{S^m}^m} \\ \vdots & \ddots & \vdots \\ \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_1^m} & \cdots & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_{S^m}^m} \end{pmatrix} \tag{7}$$

In order to simplify the expression, make:

$$f^m\left(n_j^m\right)=\frac{\partial f^m\left(n_j^m\right)}{\partial n_j^m} \tag{8}$$

As written in matrix form:

$$\mathbf{F}^m(\mathbf{n}^m)=\begin{pmatrix} f^m\left(n_1^m\right) & & 0 \\ & \ddots & \\ 0 & & f^m\left(n_{S^m}^m\right) \end{pmatrix} \tag{9}$$

Consider the element (i, j) in matrix, then

$$\begin{aligned} \frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial\left(\sum_{l=1}^{S^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1}\right)}{\partial n_j^m} \\ &= w_{i,j}^{m+1}\frac{\partial a_j^m}{\partial n_j^m} \\ &= w_{i,j}^{m+1}\frac{\partial f^m\left(n_j^m\right)}{\partial n_j^m} \\ &= w_{i,j}^{m+1} f^m\left(n_j^m\right) \end{aligned} \tag{10}$$

The Jacobian matrix (7) can be rewritten as:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m}=\mathbf{W}^{m+1}\mathbf{F}^m\left(\mathbf{n}^m\right) \tag{11}$$

Further derivation, get recurrence formula of $\mathbf{s}^m$:

$$\mathbf{s}^m=\frac{\partial \hat{F}}{\partial \mathbf{n}^m}=\left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m}\right)^T\frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}}=\dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T\frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}}=\mathbf{F}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})\mathbf{s}^{m+1} \tag{12}$$

Equation (12) shows the characteristics of the BP neural network, back propagation process is simulated by symbol as follows:

$$\mathbf{s}^M \Rightarrow \mathbf{s}^{M-1} \Rightarrow \mathbf{s}^{M-2} \Rightarrow \cdots \Rightarrow \mathbf{s}^2 \Rightarrow \mathbf{s}^1 \tag{13}$$

At last, the starting point is obtained by recursion

$$\mathbf{s}^M=-2\mathbf{F}^M\left(\mathbf{n}^M\right)\left(\mathbf{t}-\mathbf{a}\right) \tag{14}$$

Put equation (14) into equation (1), the results can be obtained.

III. Prediction of Short Term Exchange Rate

A. *Data acquisition and preprocessing*

We acquired data from China Merchants Bank foreign exchange market analysis software V2.8. In this case, we selected euro/dollar exchange rate per hour during the closing price as the sample on August 16, 2012 to April 24, 2013, and produced a total of 4290 samples. (Sample data date format: "8210000" is 00:00 on August 21.)

Sample data as shown in Table 1 (due to the limitation of length, only list the first set of data and the last set of data, intermediate data omitted):

TABLE I. Original input sample data

Month/day/time	EUR/USD	Month/day/time	EUR/USD	Month/day/time	EUR/USD
8210000	1.2343	8221600	1.2457	4222000	1.3048
8210100	1.2341	8221700	1.2473	4222100	1.3027
8210200	1.2340	8221800	1.2463	4222200	1.3027
8210300	1.2346	8221900	1.2449	4222300	1.3028
8210400	1.2347	8222000	1.2450	4230000	1.3052
8210500	1.2344	8222100	1.2442	4230100	1.3047
8210600	1.2346	8222200	1.2475	4230200	1.3047
8210700	1.2350	8222300	1.2467	4230300	1.3058
8210758	1.2351	8230000	1.2468	4230400	1.3062
8210900	1.2350	8230100	1.2464	4230500	1.3066
8211000	1.2356	8230200	1.2488	4230600	1.3064
8211100	1.2356	8230300	1.2527	4230700	1.3063
8211200	1.2353	8230400	1.2517	4230758	1.3057
8211300	1.2357	8230500	1.2526	4230900	1.3052
8211400	1.2355	8230600	1.2531	4231000	1.3045
8211500	1.2366	8230700	1.2531	4231100	1.3048
8211600	1.2405	8230758	1.2532	4231200	1.3034
8211700	1.2409	8230900	1.2531	4231300	1.3043
8211800	1.2411	8231000	1.2543	4231400	1.3046

TABLE 1, cont.					
8211900	1.2424	8231100	1.2538	4231500	1.3058
8212000	1.2433	……		4231600	1.2987
8212100	1.2457	4200100	1.3066	4231700	1.2981
8212200	1.2470	4200200	1.3075	4231800	1.2987
8212300	1.2478	4200300	1.306	4231900	1.2983
8220000	1.2469	4200400	1.3058	4232000	1.3000
8220100	1.2480	4200500	1.3050	4232100	1.2982
8220200	1.2475	4200600	1.3050	4232200	1.3020
8220300	1.2465	4200700	1.3050	4232300	1.3014
8220400	1.2462	4200758	1.3050	4240000	1.3011
8220500	1.2471	4220900	1.3064	4240100	1.3007
8220600	1.2469	4221000	1.3077	4240200	1.3001
8220700	1.2469	4221100	1.3074	4240300	1.2991
8220758	1.2474	4221200	1.3076	4240400	1.2997
8220900	1.2471	4221300	1.3069	4240500	1.2998
8221000	1.2463	4221400	1.3072	4240600	1.3001
8221100	1.2463	4221500	1.3063	4240700	1.3002
8221200	1.2460	4221600	1.3036	4240758	1.2993
8221300	1.2467	4221700	1.3043	4240900	1.2998
8221400	1.2460	4221800	1.3043	4241000	1.2993
8221500	1.2463	4221900	1.3041	4241031	1.2990

We selected the top 4000 samples to form the training set, the after 290 samples as test set. In the training process, we randomly selected about 75% as the actual training set (3000 samples) in the top 4000 samples, 15% (600 samples) as a validation set, 10% as a test set (400 samples).

Data in the training set could be divided into input array using the time series analysis method [2] [3]. A mapping is established from N points in the continuous time series with time interval for t to x_{N+t} values in the future. In this case, we selected t = 1, N = 10. Specific division method is shown in Table 2:

TABLE II. SAMPLE DIVISION METHOD

Input Data	Expected Results
$(x_1, x_2, x_3, \cdots, x_{10})$	x_{11}
$(x_2, x_3, x_4 \cdots, x_{11})$	x_{12}
$\vdots$	$\vdots$
$(x_{t-9}, x_{t-8}, x_{t-7} \cdots, x_t)$	x_{t+1}

The main factors influencing the time series prediction effect are dependences of future data on existing data future.

When sequence has the higher self correlation, applying time series forecasting model to forecast will achieve good results. Durbin-Watson method is commonly used as inspection methods [4] [5]. Self correlation of data can be obtained using Matlab software [6].

B. *Establishing BP neural network model*

We set up a three layers BP network with multiple inputs in Matlab 2010b: the input layer, hidden layer and output layer. The hidden layer has 15 neurons; Output layer outputs a result each time, as shown in Fig.1.

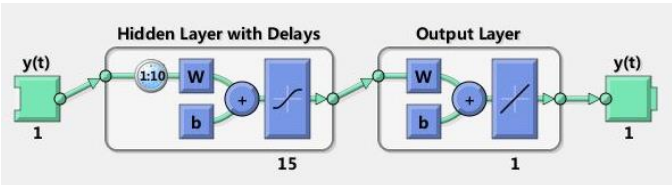

Fig. 1. BP neural network model

The normalized data input to the neural network, in order to train network weight b and bias values w, performance index of mean square error (mse), the target for mse<=0.0001 , initial weights and bias value were randomly selected in (0, 1). Take one value for every 2 times training. Error decreases with the increase of the number of training, and tends to level off. Then w and b change slightly and the network has been basically stable. w and b change slightly and the network has been basically stable.

Figure 2 shows the target data and forecast data fitting charts, where the upper left chart is 3000 training data, the upper right chart is 600 validation data, the lower left chart is 400 test data, the lower right chart is 4000 sample data. The success rate of each figure fitting has reached more than 99%,

it indicates that the network has reached the prediction requirements, and good stability.

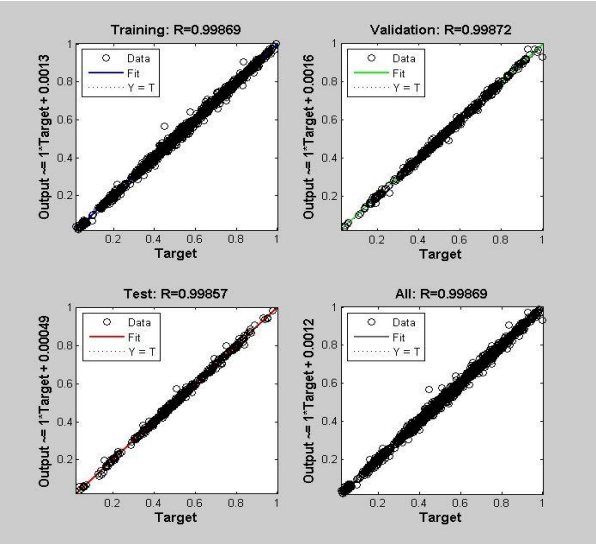

Fig. 2. BP neural network model

C. *Testing the BP network*

Test 1: In order to further determine the stability and accuracy of network, we reselected 800 data from the samples of BP network training set to test the network. Figure 3 shows the fitting effect of expected value and predictive value. We can see only 10 points of error is more obvious, but far lower than 0.1.

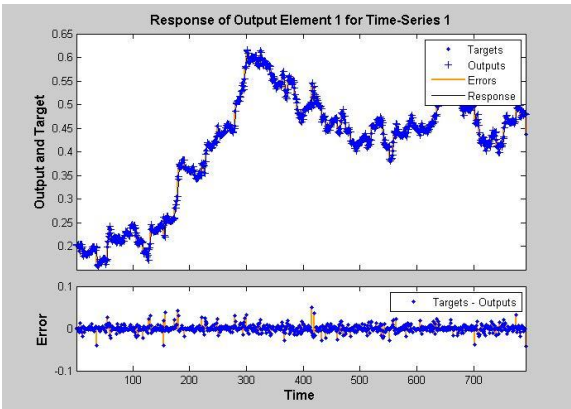

Fig. 3. Testing Results for Test 1

It indicates that the BP network is completely suitable for short-term exchange rate forecasting.

Test 2: We selected other 290 samples uninvolved in the previous test to test network. Testing results are shown in Figure 4. Almost all of the forecast data are fallen from the original data points within the range of error less than 0.05. It indicates that the BP neural network can achieve fairly good short-term exchange rate forecasting effect through the strict training.

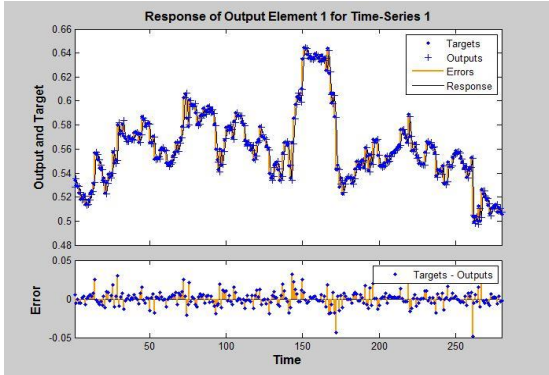

Fig. 4. Testing Results for Test 2

IV. CONCLUSIONS

In view of the exchange rate prediction problem, this paper puts forward the application of BP neural network model and using the method of time series analysis to forecast short-term exchange rate. Data chose August 16, 2012 to April 24, 2013 euro/dollar exchange rate per hour during the closing price for the samples. Simulation experiments are conducted in Matlab 201b and get a better goodness of fit. Thus the BP neural network has very good prediction effect for those data conforming to the time series analysis, and has the very high self-study and adaptability. It makes it possible to get the trend of exchange rate changes over time in the future based on the analysis of historical data. This will provide strong support for the enterprise and the government decision-making, especially for those sensitive to short-term exchange rate change and requiring high prediction accuracy.

ACKNOWLEDGMENT

The research work was supported by my undergraduate student, Gai Li (Student ID: 2009111318).

REFERENCES

[1] D. Graupe, Principles of Artificial Neural Networks, 3rd Edition, World Scientific Publishing Co Pte Ltd, 2013

[2] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung, Time series analysis : forecasting and control, Fifth Edition, John Wiley & Sons, 2015

[3] S. Galeshchuk, “Neural networks performance in exchange rate prediction”, Neurocomputing, vol.172, pp.446-452, Jan. 2016

[4] D.C. Moosmayer, Y.L. Chong, M.J. Liu, B. Schuppar, “A neural network approach to predicting price negotiation outcomes in business-to-business contexts”, Expert Systems with Applications, vol.40, no.8, pp.3028-3035, 2012

[5] M. Rehman, Mehreen, G.M. Khan, S.A. Mahmud, “Foreign Currency Exchange Rates Prediction Using CGP and Recurrent Neural Network” Proc of International Conference on Future Information Engineering, Beijing, China, IERI Procedia, vol.10, pp.239-245, July 2014

[6] Z.H. Yan, L. Zeng, “The BP Neural Network with MATLAB”, Proc of 2013 International Conference on Electrical,Control and Automation Engineering, Hong Kong, China, pp.565-569, Dec. 2013.