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 V2.5. 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 place. 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, 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]:

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

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

\[ \hat{F}(x) = (t(k) - a(k))^{T}(t(k) - a(k)) = e^{T}(k)e(k) \] (2)

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

\[ w_{ij}^{m+1} = w_{ij}^{m}(k) - \alpha \frac{\partial \hat{F}}{\partial w_{ij}} \] (3)

\[ b_{i}^{m+1} = b_{i}^{m}(k) - \alpha \frac{\partial \hat{F}}{\partial b_{i}} \]

In order to simplify the expression, we define:

\[ s_{i}^{m} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} \] (4)

There are:

\[ \frac{\partial \hat{F}}{\partial w_{ij}^{m}} = s_{i}^{m}a_{j}^{m-1}, \frac{\partial \hat{F}}{\partial b_{i}^{m}} = s_{i}^{m} \] (5)

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

\[ w_{ij}^{m+1} = w_{ij}^{m}(k) - \alpha s_{i}^{m}a_{j}^{m-1} \]

\[ b_{i}^{m+1} = b_{i}^{m}(k) - \alpha s_{i}^{m} \] (6)

Application of Jacobian matrix:

\[ \frac{\partial n_{i+1}^{m+1}}{\partial n_{i}^{m}} \]

\[ \frac{\partial n_{i+1}^{m+1}}{\partial n_{i}^{m}} = \begin{bmatrix} \frac{\partial n_{i+1}^{m+1}}{\partial n_{i}^{m}} & \cdots & \frac{\partial n_{i+1}^{m+1}}{\partial n_{i}^{m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial n_{i+1}^{m+1}}{\partial n_{i}^{m}} & \cdots & \frac{\partial n_{i+1}^{m+1}}{\partial n_{i}^{m}} \end{bmatrix} \] (7)
In order to simplify the expression, make:

\[ f^n(m^n_j) = \frac{\partial f^m(n^n_j)}{\partial n^n_j} \]

As written in matrix form:

\[
\begin{pmatrix}
  f^n(n^n_1) & \cdots & 0 \\
  \vdots & \ddots & \vdots \\
  0 & \cdots & f^n(n^n_m)
\end{pmatrix}
\]

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

\[
\frac{\partial n^{m+1}_i}{\partial n^m_j} = \sum_{l} w^{m+1}_{i,l} \left( a^m_l + b^{m+1}_l \right)
\]

\[
= w^{m+1}_{i,j} \frac{\partial a^m_l}{\partial n^m_j} + w^{m+1}_{i,j} \frac{\partial b^{m+1}_l}{\partial n^m_j}
\]

\[
= w^{m+1}_{i,j} f^n(n^n_j)
\]

The Jacobian matrix (7) can be rewritten as:

\[
\frac{\partial n^{m+1}}{\partial n^m} = W^{m+1} F^n(n^n)
\]

Further derivation, get recurrence formula of \(s^m^n\):

\[
s^m^n = \frac{\partial f^n}{\partial n^n} \left( \frac{\partial n^{m+1}}{\partial n^m} \right) = F^n(n^n) \cdot \frac{\partial f^n}{\partial n^n} = F^n(n^n) s^m^n
\]

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

\[
s^M \Rightarrow s^{M-1} \Rightarrow s^{M-2} \Rightarrow \cdots \Rightarrow s^2 \Rightarrow s^1
\]

At last, the starting point is obtained by recursion

\[
s^M = -2F^M(n^M)(t - a)
\]

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>
<thead>
<tr>
<th>Month/day/time</th>
<th>EUR/USD</th>
<th>Month/day/time</th>
<th>EUR/USD</th>
<th>Month/day/time</th>
<th>EUR/USD</th>
</tr>
</thead>
<tbody>
<tr>
<td>8210000</td>
<td>1.2343</td>
<td>8221600</td>
<td>1.2457</td>
<td>4220000</td>
<td>1.3048</td>
</tr>
<tr>
<td>8210100</td>
<td>1.2341</td>
<td>8221700</td>
<td>1.2473</td>
<td>4221000</td>
<td>1.3027</td>
</tr>
<tr>
<td>8210200</td>
<td>1.2340</td>
<td>8221800</td>
<td>1.2463</td>
<td>4222000</td>
<td>1.3027</td>
</tr>
<tr>
<td>8210300</td>
<td>1.2346</td>
<td>8221900</td>
<td>1.2449</td>
<td>4223000</td>
<td>1.3028</td>
</tr>
<tr>
<td>8210400</td>
<td>1.2347</td>
<td>8222000</td>
<td>1.2450</td>
<td>4230000</td>
<td>1.3052</td>
</tr>
<tr>
<td>8210500</td>
<td>1.2344</td>
<td>8222100</td>
<td>1.2442</td>
<td>4230100</td>
<td>1.3047</td>
</tr>
<tr>
<td>8210600</td>
<td>1.2346</td>
<td>8222200</td>
<td>1.2475</td>
<td>4230200</td>
<td>1.3047</td>
</tr>
<tr>
<td>8210700</td>
<td>1.2350</td>
<td>8222300</td>
<td>1.2467</td>
<td>4230300</td>
<td>1.3058</td>
</tr>
<tr>
<td>8210758</td>
<td>1.2351</td>
<td>8230000</td>
<td>1.2468</td>
<td>4230400</td>
<td>1.3062</td>
</tr>
<tr>
<td>8210900</td>
<td>1.2350</td>
<td>8230100</td>
<td>1.2464</td>
<td>4230500</td>
<td>1.3066</td>
</tr>
<tr>
<td>8211000</td>
<td>1.2356</td>
<td>8230200</td>
<td>1.2488</td>
<td>4230600</td>
<td>1.3064</td>
</tr>
<tr>
<td>8211100</td>
<td>1.2356</td>
<td>8230300</td>
<td>1.2527</td>
<td>4230700</td>
<td>1.3063</td>
</tr>
<tr>
<td>8211200</td>
<td>1.2353</td>
<td>8230400</td>
<td>1.2517</td>
<td>4230758</td>
<td>1.3057</td>
</tr>
<tr>
<td>8211300</td>
<td>1.2357</td>
<td>8230500</td>
<td>1.2526</td>
<td>4230900</td>
<td>1.3052</td>
</tr>
<tr>
<td>8211400</td>
<td>1.2355</td>
<td>8230600</td>
<td>1.2531</td>
<td>4231000</td>
<td>1.3045</td>
</tr>
<tr>
<td>8211500</td>
<td>1.2366</td>
<td>8230700</td>
<td>1.2531</td>
<td>4231100</td>
<td>1.3048</td>
</tr>
<tr>
<td>8211600</td>
<td>1.2405</td>
<td>8230758</td>
<td>1.2532</td>
<td>4231200</td>
<td>1.3034</td>
</tr>
<tr>
<td>8211700</td>
<td>1.2409</td>
<td>8230900</td>
<td>1.2531</td>
<td>4231300</td>
<td>1.3043</td>
</tr>
<tr>
<td>8211800</td>
<td>1.2411</td>
<td>8231000</td>
<td>1.2543</td>
<td>4231400</td>
<td>1.3046</td>
</tr>
</tbody>
</table>
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 values in the future. In this case, we selected t = 1, N = 10. Specific division method is shown in Table 2:

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, x_2, x_3, \ldots, x_{10}$</td>
<td>$x_1$</td>
</tr>
<tr>
<td>$x_2, x_3, x_4, \ldots, x_{11}$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$x_{n-9}, x_{n-8}, x_{n-7}, \ldots, x_n$</td>
<td>$x_{n+1}$</td>
</tr>
</tbody>
</table>

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.

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.

![BP neural network model](image1)

**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.

![Testing Results for Test 1](image2)

**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.

![Testing Results for Test 2](image3)

**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**


