



# Study on Exchange Rate Forecasting Using Recurrent Neural Networks

Yuxi Ye

School of Finance, Harbin University of Commerce, Harbin, China

**Email address:**

874490053@qq.com

**To cite this article:**

Yuxi Ye. Study on Exchange Rate Forecasting Using Recurrent Neural Networks. *International Journal of Economics, Finance and Management Sciences*. Vol. 5, No. 6, 2017, pp. 300-303. doi: 10.11648/j.ijefm.20170506.14

**Received:** September 1, 2017; **Accepted:** October 19, 2017; **Published:** November 20, 2017

---

**Abstract:** It focuses on the problems of forecasting exchange rate that is a nonlinear time series. A dynamics systems approach and the recurrent neural networks (RNN) were employed to modeling this nonlinear time series. The delay time was calculated using mutual information method and embedding dimension was confirmed by false nearest neighbors. The dataset was reconstructed from source time series for trained and verified the neural networks model. The quadratic optimization criterion was considered which the neural networks weights update algorithm were derived using gradient descent method for hidden layer; recurrent layer and output layer. The calculation flow chart was designed for neural networks learning and emulation. The reliability and stability of neural networks was confirmed by testing dataset. The results of simulation showed that the recurrent neural networks were preferably performance for prediction the change of exchange rate.

**Keywords:** Neural Networks, Forecasting Exchange Rate, Nonlinear Time Series

---

## 1. Introduction

Exchange rate is a very important economic variable; the fluctuation of exchange rate has great effects on the waxing and waning of national income, the development of industry and agriculture, the change of interest rate, etc. As a result, the forecasting of exchange rate has attracted wide attention. People have developed many methods and models in exchange rate forecasting [1~3]. It is proved that neural network not only has strong ability in nonlinear mapping, which can approximate primitive function in any precision; but can also learn from large amounts of history data to find laws in some behavior changes. There are masses of reference make study on artificial neural network in exchange rate forecasting [4~8]. Although the study shows that the result is better than results in other methods, the training network is caught in local minimal point when it is connected with weight. Lyapunov's theory in stability on dynamic system is used to derive the learning algorithm in neural network weight and make sure the neural network learning does not converge to the local minimal. The paper creates a recurrent neural network by Matlab toolbox. Finally, accomplishing the goal uses middle price at RMB exchange rate to predict middle price in the future. The features and contents of this paper are

included. Section 2 the forecasting model of recurrent networks was proposed. Section 3 the neural networks weights update algorithm was deduced by gradient descent method. Section 4 the prediction calculate flow was designed and the RMB to US dollars exchange rate simulation was proceed, and then prediction error was analysis.

## 2. Neural Network Forecasting Model

Nonlinear forecasting equation can be described as [9~12]:

$$y(k+1) = f[y(k), \dots, y(k-p+1)] + v(k) \quad (1)$$

where,  $f(\cdot)$  is a nonlinear function;  $y(k), \dots, y(k-p+1) \in \mathbb{R}$  is output variable in nonlinear function and output time delay in different orders described in equation (1).  $p$  is order in time delay during input or output. If  $\mathbf{E}(v(k) | y(k), y(k-1), \dots) = 0$  and  $\mathbf{v}(k)$  has limited variance  $\sigma^2$ , then there is a given minimum mean square error optimal estimation in series [13~14]  $y(k), \dots, y(k-p); k \geq p+1$ . That is:

$$\hat{y}(k+1) = \mathbf{E}(y(k+1) | y(k), y(k-1), \dots, y(k-p)) = f(y(k), y(k-1), \dots, y(k-p+1)) \quad (2)$$

Where: there is mean square error and unknown nonlinear function  $f$ . As a result, it can use Recurrent Neural Networks to approximate the function  $f$ . The structure of recurrent neural network shows [15~16] in figure 1. Where: weights  $W_{i,j}^H, i=1,2,\dots,p, j=1,2,\dots,m$  are from input layer to hidden layer in neural network, the weights  $W_j^O, j=1,2,\dots,m$  are from hidden layer to output layer, the weight  $W_l^R$ . Hidden layer neuron activation function  $\varphi(\cdot)$  is Sigmoid function. The output layer is a linear combination.

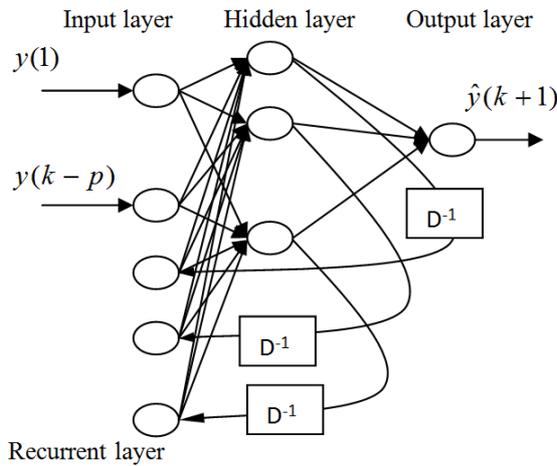


Figure 1. Structure diagram of recurrent neural network.

Figure 1 represents that the neural network approximate equation (1). That is:

$$\hat{y}(k+1) = f(y(k), y(k-1), \dots, y(k-p+1)) = \sum_{j=1}^m W_j^O \Phi(\sum_{l=1}^s W_l^R x_j(k-1) + \sum_{h=1}^p W_{j,h}^H U_h(k)) \quad (3)$$

In equation (3), network parameters  $W_{j,h}^H, W_j^O, W_l^R$  can be acquired through neural network learning.  $U_h(k) = [y(k), y(k-1), \dots, y(k-p+1)]^T$ ,  $x_j(k), j=1,2,\dots,m$  is output signal in hidden layer. In order to derive the learning algorithm in neural network, the equation (3) can be revised as:

$$\begin{aligned} Y_i(k) &= \sum_{j=1}^m W_j^O x_j(k) \\ x_j(k) &= \varphi(S_j(k)) \\ S_j(k) &= \sum_{l=1}^s W_l^R x_j(k-1) + \sum_{h=1}^p W_{j,h}^H U_h(k), \\ j &= 1, 2, \dots, m \end{aligned} \quad (4)$$

### 3. The Weights Update Algorithm

The error function is  $\varepsilon_i(k) = y_i(k) - \hat{y}_i(k)$ . Defining cost function as:

$$E_k(k) = \frac{1}{2} \sum_{i=1}^m \varepsilon_i^2(k) = \frac{1}{2} \sum_{i=1}^m (y_i(k) - \hat{y}_i(k))^2 \quad (5)$$

Where:  $y_i(k)$  is  $k$  time's output vector of source time series,  $\hat{y}_i(k)$  is  $k$  time's output vector of neural network. The weights adjust algorithm is

$$\begin{aligned} \theta(k) &= \theta(k-1) - \eta \frac{\partial E_k}{\partial \theta} \\ &= \theta(k-1) + \eta \varepsilon_i(k) \frac{\partial \hat{y}_i(k)}{\partial \theta} \end{aligned} \quad (6)$$

Where: the  $\theta$  is weight for all neural networks. The  $\theta$  is hidden layer weight.

$$\frac{\partial \hat{y}(k)}{\partial W_j^O} = x_j(k) \quad (7)$$

The  $\theta$  is output layer weight.

$$\frac{\partial \hat{y}(k)}{\partial W_{i,j}^H} = W_j^O P_j(k) \quad (8)$$

The  $\theta$  is recurrent layer weight.

$$\frac{\partial \hat{y}(k)}{\partial W_j^R} = W_j^O Q_{i,j}(k) \quad (9)$$

Where:  $P_j(k) = \frac{\partial X_j(k)}{\partial W_j^R}$ ;  $Q_{i,j}(k) = \frac{\partial X_j(k)}{\partial W_{i,j}^H}$

The equation (10) and (11) are solvable

$$\begin{aligned} P_j(k) &= \varphi'(S_j(k)) [X_j(k-1) + W_j^R P_j(k-1)] \\ P_j(0) &= 0 \end{aligned} \quad (10)$$

$$\begin{aligned} Q_{i,j}(k) &= \varphi'(S_j(k)) [U_h(k) + W_j^R Q_{i,j}(k-1)] \\ Q_{i,j}(0) &= 0 \end{aligned} \quad (11)$$

The global error function is

$$E = \sum_{k=1}^m E_k \quad (12)$$

According to all above, the calculated step is designed. Step (1) all layer weights are initialized, set training rate  $\eta$  and  $x_j(0) = 0, P_j(0) = 0, Q_{i,j} = 0, k = 1$

Step (2) the delay time of time series is computed by mutual

information and embedded dimension of time series confirmed by false nearest neighbors.

Step (3) the data pair  $\{U_p(k), Y_m(k)\}$  is formed using results of step (2).

Step (4) the neural networks weights are trained by data pair  $\{U_p(k), Y_m(k)\}$

Step (5) the  $S_j(k), X_j(k), Y_i(k)$  is calculated using equation (4)

Step (6) the error of weights training is computed using equation (5)

Step (7) using equations (5), (6), (7), (8), (9) and (10) to adjust all of weights

Step (8) set  $k = k + 1$ , go to step (2) and continue to calculate until all datasets finish.

Step (9) if  $\sum_{k=1}^m E_k > \varepsilon, \varepsilon > 0$ , set  $k = 1$  again, return step

(2), when  $E$  less then a position  $\varepsilon$ , the weights training produce finished.

Step (10) the testing dataset is used to confirm the performances of the neural networks.

### 4. The Time Series Prediction Simulation

A time series has 478 samples  $\{y_1, y_2, y_3, \dots, y_{478}\}$ . The two datasets are formed to use samples of source time series, the first dataset has 390 samples which are used to train the neural networks, the second dataset has 88 samples which are used to verify the neural networks. The embedded dimension [17] shows in figure 2, the time delay [18] shows in figure 3.

From figure 1 and figure 2, the time delay is 15 and embedded dimension is 4, the data pair  $\{U_p(k), Y_m(k)\}$  is constructed to form the first dataset in order to input the neural networks, it is  $15 \times 20 \times 1$ .

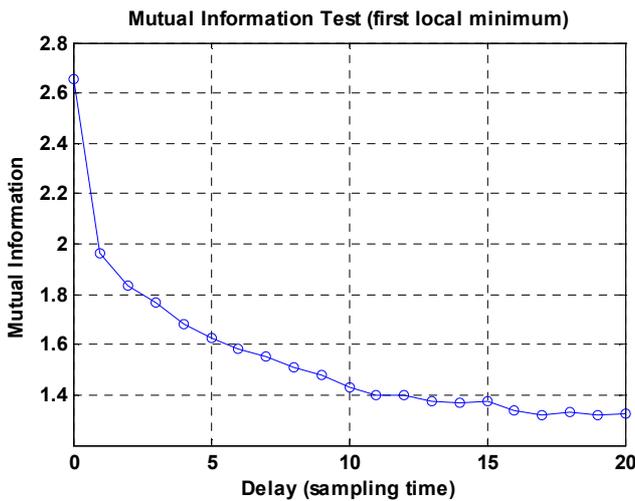


Figure 2. The Time delay curve.

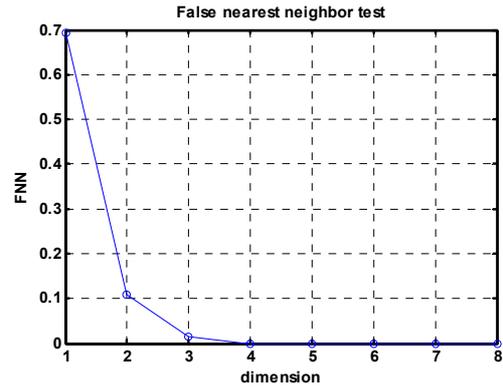


Figure 3. The embedded dimension curve.

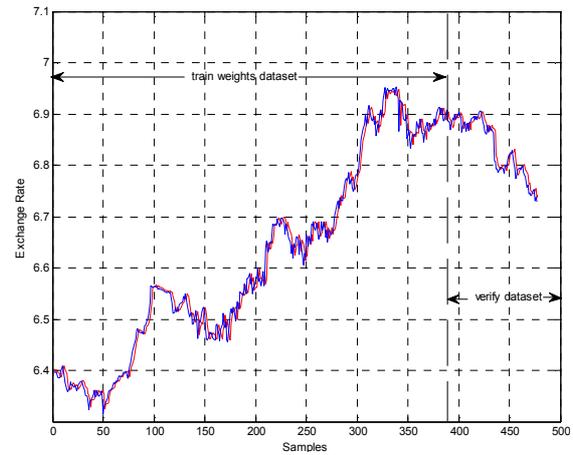


Figure 4. The neural networks train and verify curve.

The simulation output shows figure 4, the absolute error of train is in figure 5, and histogram is in figure 6 and figure 7, in the figure 4 the neural networks are better to predict the exchange rate. In the figure 5 may know that only 8 epochs the neural networks absolute error is close to zero and the error is normal distribution in figure 6 and figure 7. The neural networks training algorithm may be satisfy real time calculate on line. The weights updated algorithm has a high speed to train neural networks and not convergence in the local minimal. The algorithm may apply to modeling and control dynamics systems, also.

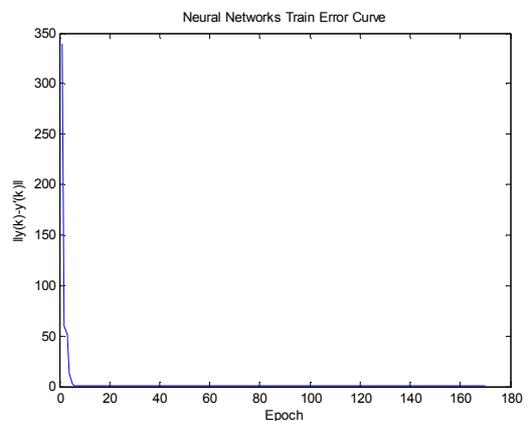


Figure 5. Absolute error of train Neural Networks.

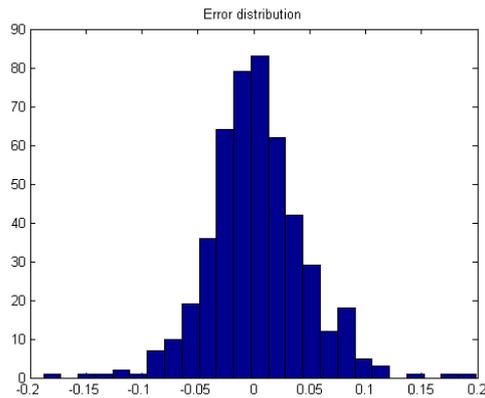


Figure 6. The histogram of error distribution.

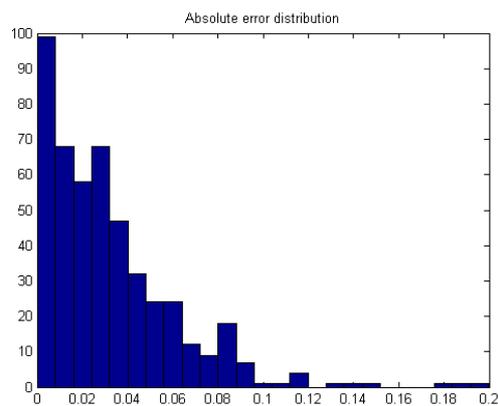


Figure 7. The histogram of absolute error distribution.

## 5. Conclusion

The neural networks have recurrent neurons to feedback the hidden neuron states in order to speed up the weights update and the fast convergence. The recurrent neurons can memorize learning message that the neural networks can not locate in local minimal point during training process. When apply the recurrent neural networks to forecasting middle rate of RMB to US dollars, it is clear that the neural networks has some feedback neurons have better performances then others methods. This paper proposed method may use modeling chaotic time series or nonlinear dynamics model.

## References

- [1] Xing Chengdong, Zhang Quan. Research on the Impact of RMB Exchange Rate Fluctuation on Chinese Stock Market-From the Perspective of Index. *International Journal of Economics, Finance and Management Sciences*. 2015 3(5): 641-645.
- [2] Md. Zahangir Alam, Muhammad Abdur Rahim. Nexus between stock exchange index and exchange rates. *International Journal of Economics, Finance and Management Sciences*. 2013; 1(6): 330-334.
- [3] Diebold, F. X., Nason, J. A.. Nonparametric exchange rate prediction. *Journal of International Economics* 28, 315–332.1990.
- [4] Y. Kajitani, K. W. Hipel, and A. I. McLeod, Forecasting nonlinear time series with feed-forward neural networks: a case study of Canadian lynx data, *Journal of Forecasting*, vol. 24, no. 2, pp. 105–117, 2005.
- [5] T. Takahama, S. Sakai, A. Hara, and N. Iwane, Predicting stock price using neural networks optimized by differential evolutionwith degeneration, *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 5021–5031, 2009.
- [6] D. Xiao and J. Wang, Modeling stock price dynamics by continuum percolation system and relevant complex systems analysis, *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 20, pp. 4827–4838, 2012.
- [7] M. Zounemat-Kermani, Principal component analysis (PCA)for estimating chlorophyll concentration using forward andgeneralized regression neural networks, *Applied Artificial Intelligence*, vol. 28, no. 1, pp. 16–29, 2014.
- [8] R. Ebrahimpour, H. Nikoo, S. Masoudnia, M. R. Yousefi, and M. S. Ghaemi, Mixture of mlp-experts for trend forecasting of time series: a case study of the tehran stock exchange, *International Journal of Forecasting*, vol. 27, no. 3, pp. 804–816, 2011.
- [9] D. O. Faruk, A hybrid neural network and ARIMA model for water quality time series prediction, *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 586–594, 2010.
- [10] M. Tripathy, Power transformer differential protection using neural network principal component analysis and radial basis function neural network, *Simulation Modelling Practice and Theory*, vol. 18, no. 5, pp. 600–611, 2010.
- [11] P.-C. Chang, D.-D. Wang, and C.-L. Zhou, A novel model by evolving partially connected neural network for stock price trend forecasting, *Expert Systems with Applications*, vol. 39, no. 1, pp. 611–620, 2012.
- [12] T. H. Roh, Forecasting the volatility of stock price index, *Expert Systems with Applications*, vol. 33, no. 4, pp. 916–922, 2007.
- [13] G. E. Batista, E. J. Keogh, O. M. Tataw, and V. M. de Souza, CID: An efficient complexity-invariant distance for time series, *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 634–669, 2014.
- [14] M. Zounemat-Kermani, Principal component analysis (PCA)for estimating chlorophyll concentration using forward and generalized regression neural networks, *Applied Artificial Intelligence*, vol. 28, no. 1, pp. 16–29, 2014.
- [15] Z. Q. Guo, H. Q. Wang, and Q. Liu, Financial time series forecasting using LPP and SVM optimized by PSO, *Soft Computing*, vol. 17, no. 5, pp. 805–818, 2013.
- [16] H. L. Niu and J. Wang, Financial time series prediction by a random data-time effective RBF neural network, *Soft Computing*, vol. 18, no. 3, pp. 497–508, 2014.
- [17] Kennel M B, Brown R, Abarbanel H D I. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 1992, 45: 3403-3411.
- [18] Lin T, Horne B G, Tino P, Giles C L. A delay damage model selection algorithm for NARX neural networks. *IEEE Transactions on Signal Processing*, 1997 45(11), 2719-2730.