

# Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting

Lean Yu<sup>1,2</sup>, Shouyang Wang<sup>1,2</sup>, and Kin Keung Lai<sup>3,4</sup>

<sup>1</sup> Institute of Systems Science, Academy of Mathematics and Systems Sciences,  
Chinese Academy of Sciences, Beijing 100080, China

<sup>2</sup> School of Management, Graduate School of Chinese Academy of Sciences,  
Chinese Academy of Sciences, Beijing 100039, China  
{yulean, sywang}@amss.ac.cn

<sup>3</sup> College of Business Administration, Hunan University, Changsha 410082, China

<sup>4</sup> Department of Management Sciences, City University of Hong Kong,  
Tat Chee Avenue, Kowloon, Hong Kong  
mskklai@cityu.edu.hk

**Abstract.** This study proposes a novel forecasting approach – an adaptive smoothing neural network (ASNN) – to predict foreign exchange rates. In this new model, adaptive smoothing techniques are used to adjust the neural network learning parameters automatically by tracking signals under dynamic varying environments. The ASNN model can make the network training process and convergence speed faster, and make network's generalization stronger than the traditional multi-layer feed-forward network (MLFN) model does. To verify the effectiveness of the proposed model, three major international currencies (British pounds, euros and Japanese yen) are chosen as the forecasting targets. Empirical analyses reveal that the proposed novel forecasting model outperforms the other comparable models. Furthermore, experimental results also show that the proposed model is an effective alternative approach for foreign exchange rate forecasting.

## 1 Introduction

The difficulty in predicting foreign exchange rates, due to their high volatility and complexity, has long been an imperative concern in international financial markets as many econometric methods are unable to produce significantly better forecasts than the random walk (RW) model [1]. Recent studies provide some evidence that nonlinear models are able to produce better predictive results, ameliorating the performance of the simple RW model. Of the various nonlinear models, the artificial neural network (ANN) model has emerged as a strong alternative for predicting exchange rates. As claimed by Grudnitski and Osburn [2], neural networks are particularly well suited for finding accurate solutions in an environment characterized by complex, noisy, irrelevant or partial information. Literature documenting this research effort is quite diverse and involves different architectural designs. Some examples are presented. Early applications of neural networks in forecasting chaotic time series have been performed by Lapedes and Farker [3]. Weigend et al. [4] and Refenes et al. [5] ap-

plied multilayer forward network (MLFN) models in their forecasts of foreign exchange prices. Weigend's model performance was tested in terms of accuracy, giving support to nonrandom behavior. Refenes' work extended Weigend's research by adding a validity test to the model's performance and compared the results with those of the forward rate, thereby providing added support to the forecasting ability of neural networks in the foreign exchange market. Tenti [6] applied recurrent neural network (RNN) models to forecast exchange rates. Hsu et al. [7] developed a clustering neural network (CNN) model to predict the direction of movements in the USD/DEM exchange rate. Their experimental results suggested that their proposed model achieved better performance relative to other indicators. De Matos [8] compared the strength of a MLFN with that of a RNN based on the forecasting of Japanese yen futures. Likewise, Kuan and Liu [9] provided a comparative evaluation of MLFN's performance and an RNN for the prediction of an array of commonly traded exchange rates. In a more recent study by Leung et al. [10], MLFN's forecasting accuracy was compared with the general regression neural network (GRNN). The study showed that the GRNN possessed a greater forecasting strength relative to MLFN with respect to a variety of currency rates. Zhang and Berardi [11] adopted an ensemble method for exchange rate forecasting and obtained better results than those under a single network model. Chen and Leung [1] used an error correction neural network (ECNN) model to predict exchange rates and good forecasting results can be obtained with their model.

Although a handful of studies exist on neural network applications in foreign exchange markets, most of the literature focuses on the MLFN [1-5, 8-10, 12-15]. However, there are several limitations to the MLFN. For example, convergence speed of the MLFN algorithm is often slow, thus making the network learning time long. Furthermore, it is easy for the optimal solution to be trapped into local minima thus making generalization capability weak. Therefore, we propose an adaptive smoothing technique to overcome these limitations to predict the daily exchange rates for three major internationally traded currencies: British pounds, euros and Japanese yen. In order to provide a fair and robust evaluation of the ASNN model relative to performance, the forecasting performance of the proposed ASNN model is compared with those of the MLFN model, which is used as the benchmark model. The rest of this article is organized as follows. Section 2 describes the ASNN model in detail. Section 3 gives an experiment and reports the results. And Section 4 concludes the article.

## **2 Adaptive Smoothing Neural Network for Forecasting**

### **2.1 The Introduction of Neural Networks**

Artificial neural networks (ANNs) – originally developed to mimic neural networks, in particular the human brain – are composed of a number of interconnected simple processing elements called neurons or nodes. Each node receives an input signal from other nodes or external inputs; after processing the signals locally through a transfer function, a transformed signal is output to other nodes or final outputs. ANNs are characterized by the network architecture; that is, the number of layers, the number of nodes in each layer and how the nodes are connected. In a popular form, the multi-

layer feed-forward network (MLFN), all nodes and layers are arranged in a feed-forward manner. The first or the lowest layer is an input layer where external information is received. The last or the highest layer is an output layer where the network produces the model solution. In between, there are one or more hidden layers which are critical for ANNs to identify the complex patterns in the data. All nodes in adjacent layers are connected by acyclic arcs from a lower layer to a higher layer. ANNs are already one of the types of models that are able to approximate various nonlinearities in the data, and this makes them popular with academics and practitioners.

However, there are several drawbacks to the popular MLFN. First of all, the convergence speed of the MLFN algorithm is often slow, thus making network learning time long. Second, it is easy for the optimal solution obtained to be trapped into local minima, thus making generalization capability weak. Finally, the question of how to select reasonable network architecture is still an intractable problem.

In view of the above problems, in the following subsection we propose a novel algorithm to improve the MLFN by introducing adaptive smoothing techniques.

**2.2 The Adaptive Smoothing Neural Network Model**

In this study, adaptive smoothing techniques are used to adjust the neural network learning parameters automatically in terms of tracking signals under dynamic varying environments. This yields a new weight adjustment algorithm in virtue of quality control (QC) concept. In MLFN, model errors are usually the squared error or mean squared error (MSE). But using these error metrics makes it difficult to capture deviations between actual values and network output values (or expected values). In the process of neural network learning, adaptive smoothing algorithms can utilize ordinary error and mean absolute deviation (MAD) as a supplement of error measure to adjust the network’s parameters (i.e., learning weights). With the aid of cumulative ordinary error (COE), MAD, and derivative tracking signal (TS), an adaptive smoothing neural network model can be formulated.

Assume that a network with  $m$  layers has  $n$  nodes, the transfer function of every node is usually a sigmoid function (i.e.,  $f(x) = \frac{1}{1 + e^{-x}}$ ),  $y$  is an output from the output layer,  $O_i$  is an output of any unit  $i$  in a hidden layer,  $W_{ij}$  is the weight on connection from the  $j$ th to the  $i$ th unit. Suppose that there are  $N$  sample pairs  $(x_k, y_k)$  ( $k = 1, 2, \dots, N$ ), the output of unit  $i$  connected with the  $k$ th sample is  $O_{ik}$ , the input of unit  $j$  connected with the  $k$ th sample is

$$net_{jk} = \sum_i W_{ij} O_{ik} \tag{1}$$

And the output of unit  $j$  connected with the  $k$ th sample is

$$O_{jk} = f(net_{jk}) \tag{2}$$

Here the error function is the squared error, i.e.,  $E = \frac{1}{2} \sum_{k=1}^N (y_k - \hat{y}_k)^2$ , the cumulative ordinary error (COE) is  $COE(N) = \sum_{k=1}^N (y_k - \hat{y}_k)$ , where  $y_k$  is the actual value and  $\hat{y}_k$  is the network output value. Let  $E_k$  and  $COE_k$  be a squared error and an ordi-

nary error connected with the  $k$ th sample, then  $E_k = (y_k - \hat{y}_k)^2$  and  $COE_k = (y_k - \hat{y}_k)$ . Clearly,  $COE(N) = COE(N - 1) + COE_N$ . Meanwhile, the mean absolute deviation ( $MAD$ ) and tracking signal ( $TS$ ) are defined as

$$MAD(N) = \frac{\sum_{k=1}^N |y_k - \hat{y}_k|}{N} \tag{3}$$

$$TS = \frac{COE(N)}{MAD(N)} \tag{4}$$

If  $TS$  is “large”, this means that  $COE(N)$  is large relative to the mean absolute deviation  $MAD(N)$ . This in turn says that the network output is producing errors that are either consistently positive or consistently negative. That is, a large value of  $TS$  implies that the network output is producing forecasts that are either consistently smaller or consistently larger than the actual values that are being forecast. Since an “accurate” forecasting system should be producing roughly one half positive errors and one half negative errors, a large value of  $TS$  indicates that the forecast output is not reliable. In practice, if  $TS$  exceeds a control limit, denoted by  $\theta$ , for two or more consecutive periods, this is taken as a strong indication that the forecast errors have been larger than an accurate forecasting system can reasonably be expected to produce. In our study, the control limit  $\theta$  is generally taken to be  $3\sigma$  for a neural network model with the aid of the ‘ $3\sigma$  limits theory’ proposed by Shewhart [16].

If the error signal indicates that adjustment action is needed, there are several possibilities. One possibility is that the model needs to be changed. To do this, input variables may be added or deleted to obtain a better representation of the time series. Another possibility is that the model being used does not need to be changed, but the estimates of the model’s parameters need to be changed. When using a neural network model, this is accomplished by changing parameters (i.e., model weights and bias).

Now we present the parameter adjustment process. Define the error gradient  $\delta_{jk} = \frac{\partial E_k}{\partial net_{jk}}$ , then

$$\frac{\partial E_k}{\partial W_{ij}} = \frac{\partial E_k}{\partial net_{jk}} \cdot \frac{\partial net_{jk}}{\partial W_{ij}} = \frac{\partial E_k}{\partial net_{jk}} \cdot O_{ik} = \delta_{jk} \cdot O_{ik} \tag{5}$$

(i) If  $j$  is the output node,  $O_{jk} = \hat{y}_k$ , then

$$\delta_{jk} = \frac{\partial E_k}{\partial net_{jk}} = \frac{\partial E_k}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial net_{jk}} = -(y_k - \hat{y}_k) \cdot f'(net_{jk}) \tag{6}$$

(ii) If  $j$  is not the output node, then

$$\delta_{jk} = \frac{\partial E_k}{\partial net_{jk}} = \frac{\partial E_k}{\partial O_{jk}} \cdot \frac{\partial O_{jk}}{\partial net_{jk}} = \frac{\partial E_k}{\partial O_{jk}} \cdot f'(net_{jk}) \tag{7}$$

$$\frac{\partial E_k}{\partial O_{jk}} = \sum_m \frac{\partial E_k}{\partial net_{mk}} \cdot \frac{\partial net_{mk}}{\partial O_{jk}} = \sum_m \frac{\partial E_k}{\partial net_{mk}} \cdot \frac{\partial}{\partial O_{jk}} \sum_i W_{mi} \cdot O_{ik} = \sum_m \frac{\partial E_k}{\partial net_{mk}} W_{mj} = \sum_m \delta_{mk} \cdot W_{mj} \tag{8}$$

Thus

$$\begin{cases} \delta_{jk} = f'(net_{jk}) \sum_m \delta_{mk} W_{mj} \\ \frac{\partial E_k}{\partial W_{ij}} = \delta_{jk} \cdot O_{ik} \end{cases} \quad (9)$$

The error  $\delta_{jk}$  is propagated back to the lower layers in terms of Equations (6) and (9).

In order for the network to learn, the value of each weight has to be adjusted in proportion to each unit's contribution to the total error in Equations (6) and (9). The incremental change in each weight for each learning iteration is computed using Equations (10) and (11) in the following:

$$\Delta W_{ij} = c_1 \cdot \delta_{jk} \cdot O_{ik} + c_2 \cdot \varphi_{jk} \quad (10)$$

where  $c_1$  is a learning rate ( $0 \leq c_1 < 1$ ),  $c_2$  is a positive constant that, being less than 1.0, is the smoothing rate to smooth out the weigh changes; and

$$\varphi_{jk} = \begin{cases} 0, & |TS| \leq \theta, (\theta = 3\sigma[16] \text{ or } \theta = 4 \cdot MAD [17]); \\ -COE(N), & |TS| > \theta \text{ and } TS \leq 0; \\ COE(N), & |TS| > \theta \text{ and } TS > 0. \end{cases} \quad (11)$$

It should be noted that there is a difference between our weight adjustment and the traditional momentum term. The traditional momentum term is only used to accelerate the neural network learning speed, while our weight adjustment cannot only increase learning speed but can also adjust the network search path and speed network convergence and improve neural network learning performance.

For convenience, we give the detailed algorithm for ASNN in the sequel:

- (1) Initialize random weights to avoid saturation in the learning process.
- (2) Iterate the following procedures, until error goals are satisfactory
  - a. For  $k=1$  to  $N$ 
    - (i) Compute  $O_{ik}, net_{jk}, COE(N), MAD(N)$  and  $\hat{y}_k$  (forward process)
    - (ii) Compute  $\delta_{jk}$  from the output layer to the preceding layer inversely (backward process)
  - b. For any nodes in the same layer, compute  $\delta_{jk}$  according to Equations (6) and (9)
- (3) Adjust weights with Equations (10) and (11) in terms of error gradient and tracking signals.

This completes the introduction of the ASNN algorithm. Usually, we can obtain the following benefits relative to traditional MLFN algorithms. First of all, learning error limits can be controlled via the corresponding program, making the search space smaller and learning accuracy higher. Second, model parameters can be adjusted adaptively in term of tracking signals, thus making network learning more efficient. Third, the search path can be adjusted by a smoothing factor and making it easier to obtain the network optimal solution than by using the MLFN algorithm.

To summarize, adaptive smoothing neural networks can adjust the model parameters adaptively and automatically via tracking signals, thus making the network search and convergence speed faster and avoiding local minima as far as possible.

### 2.3 ASNN for Time Series Forecasting

An adaptive smoothing neural network can be trained by the historical data of a time series in order to capture the nonlinear characteristics of the specific time series. The model parameters (such as connection weights and nodes biases) will be adjusted iteratively by a process of minimizing the forecasting errors (e.g., MSE). For time series forecasting, the computational form of the ASNN model with three-layer network connection is expressed as

$$x_t = a_0 + \sum_{j=1}^q w_j f(a_j + \sum_{i=1}^p w_{ij} x_{t-i}) + \xi_t \quad (12)$$

where  $a_j$  ( $j = 0, 1, 2, \dots, q$ ) is a bias on the  $j$ th unit, and  $w_{ij}$  ( $i = 1, 2, \dots, p; j = 1, 2, \dots, q$ ) is the connection weight between layers of the model,  $f(\bullet)$  is the transfer function of the hidden layer,  $p$  is the number of input nodes and  $q$  is the number of hidden nodes. Actually, the ASNN model in (12) performs a nonlinear functional mapping from the past observation  $(x_{t-1}, x_{t-2}, \dots, x_{t-p})$  to the future values  $x_t$ , i.e.,

$$x_t = g(x_{t-1}, x_{t-2}, \dots, x_{t-p}, v) + \xi_t \quad (13)$$

where  $v$  is a vector of all parameters and  $g$  is a nonlinear function determined by the network structure and connection weights. Thus, in some senses, the ASNN model is equivalent to a nonlinear autoregressive (NAR) model [15]. To verify the effectiveness of the ASNN model, a simulation study is presented in the following section.

## 3 Experiment Study

### 3.1 Data Sources

We use three different datasets in our forecast performance analysis. The data used are daily and are obtained from Pacific Exchange Rate Service (<http://fx.sauder.ubc.ca/>), provided by Professor Werner Antweiler, University of British Columbia, Vancouver, Canada. They consist of the US dollar exchange rate against each of the three currencies (EUR, GBP and JPY) with which it has been studied in this research. We take the daily data from 1 January 2000 to 31 October 2002 as in-sample data sets, and we take the data from 1 November 2002 to 31 December 2002 as evaluation test sets or out-of-sample datasets (partial data sets excluding holidays), which are used to evaluate the good or bad performance of the predictions, based on evaluation measurements. In order to save space, the original data are not listed in the paper, detailed data can be obtained from the website. In addition, to examine the forecasting performance, the normalized mean squared error (NMSE) [15] and directional change statistics of exchange rate movement ( $D_{stat}$ ) [14, 15] are employed here.

### 3.2 Experimental Results

When the data are prepared, we begin the ASNN model's training and learning process. In these experiments, we prepare 752 data (two years' data excluding public holidays). We use the first 22 months' data to train and validate the network, and use the

last two months' data for prediction testing. For convenience, the three-day-ahead forecasting results of three major international currencies using the proposed ASNN model are shown in Table 1.

**Table 1.** Forecast performance evaluation for the three exchange rates

Exchange rates	British pounds		Euros		Japanese yen	
	MLFN	ASNN	MLFN	ASNN	MLFN	ASNN
<i>NMSE</i>	0.5534	0.1254	0.2137	0.0896	0.2737	0.1328
<i>D<sub>stat</sub></i> (%)	55.00	77.50	57.50	72.50	52.50	67.50

As can be seen from Table 1, we can conclude that: (i) from the viewpoint of *NMSE* indicator, the ASNN model performs consistently better than the MLFN model; (ii) furthermore, the *NMSE* of the MLFN model is much larger than that of the ASNN model, indicating that adaptive smoothing techniques can effectively control error changes and significantly improve network performance; and (iii) from the *D<sub>stat</sub>* point of view, the correct number of direction of exchange rate movements increases when using the ASNN model. Among these, the increase in the British pound rate is the largest, while the increase in the Japanese yen rate is the smallest. This suggests that there may be some additional factors that need to be studied in relation to the Japanese yen. One possible reason is that the Japanese yen exchange rate is more volatile than that of the British pound; another might be that the market for yen is bigger and more efficient than the market for British pounds. However, we also find that it is feasible to predict exchange rates using the ASNN model, and that the results are promising.

## 4 Concluding Remarks and Future Research

This exploratory research examines the potential of using an ASNN model to predict main international currency exchange rates. Our empirical results suggest that the ASNN forecasting model may provide better forecasts than the traditional MLFN forecasting model. The comparative evaluation is based on a variety of statistics such as *NMSE* and *D<sub>stat</sub>*. For all currencies included in our empirical investigation, the ASNN model outperforms the traditional MLFN model in terms of *NMSE* and *D<sub>stat</sub>*. Furthermore, our experimental analyses reveal that the *NMSE* and *D<sub>stat</sub>* for three currencies using the ASNN model are significantly better than those using the MLFN model. This implies that the proposed ASNN forecasting model can be used as a feasible solution for exchange rate prediction.

However, our work also highlights some problems that need to be addressed further. For example, the accuracy of rolling forecasting is still unsatisfactory for certain currencies, such as the Japanese yen. Of course, the above problems show possible directions for future work in formulating a generic adaptive smoothing neural network prediction model for exchange rate prediction as follows:

(i) As foreign exchange markets constitute a very complex system, more factors that influence the exchange rate movement should be considered in future research.

(ii) A new adaptive smoothing algorithm that improves the traditional MLFN model should be added to neural network software packages so that users working in other domains can more easily utilize new neural network models in their work.

## References

- [1] Chen, A.S., Leung, M.T.: Regression neural network for error correction in foreign exchange forecasting and trading. *Computers and Operations Research*, 31, (2004) 1049-1068.
- [2] Grudnitski, G., Osburn, L.: Forecasting S&P and gold futures prices: an application of neural networks. *Journal of Futures Market*, 13, (1993) 631-643.
- [3] Lapedes, A., Farber, R.: Nonlinear signal processing using neural network prediction and system modeling. Theoretical Division, Los Alamos National Laboratory, *NM Report*, (1987) No. LA-UR-87-2662.
- [4] Weigend, A.S., Huberman, B.A., Rumelhart, D.E.: Generalization by weight-elimination with application to forecasting. In: Lippman, R.P., Moody, J.E. and Touretzky, D.S. (Eds), *Advances in Neural Information Processing Systems*, 3, Morgan Kaufman, San Mateo, CA, (1991) 875-882.
- [5] Refenes, A.N., Azema-Barac, M., Chen, L., Karoussos, S.A.: Currency exchange rate prediction and neural network design strategies. *Neural Computing and Applications*, 1, (1993) 46-58.
- [6] Tenti, P.: Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*, 10, (1996) 567-581.
- [7] Hsu, W., Hsu, L.S., Tenorio, M.F.: A neural network procedure for selecting predictive indicators in currency trading. In: Refenes, A.N. (Ed), *Neural Networks in the Capital Markets*, New York: John Wiley and Sons, (1995) 245-257.
- [8] De Matos, G.: Neural networks for forecasting exchange rate. *M. Sc. Thesis*. The University of Manitoba, Canada (1994).
- [9] Kuan, C.M., Liu, T.: Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics*, 10 (4), (1995) 347-364.
- [10] Leung, M.T., Chen, A.S., Daouk, H.: Forecasting exchange rates using general regression neural networks. *Computers and Operations Research*, 27, (2000) 1093-1110.
- [11] Zhang, G.P., Berardi, V.L.: Time series forecasting with neural network ensembles: an application for exchange rate prediction. *Journal of the Operational Research Society*, 52, (2001) 652-664.
- [12] Brooks, C.: Linear and nonlinear (non-) forecastability of high frequency exchange rates. *Journal of Forecasting* 16, (1997) 125-145.
- [13] Gencay, R.: Linear, nonlinear and essential foreign exchange rate prediction with simple technical trading rules. *Journal of International Economics*, 47, (1999) 91-107.
- [14] Yao, J.T., Tan, C.L.: A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, 34, (2000) 79-98.
- [15] Yu, L.A., Wang, S.Y., Lai, K.K.: A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Computers and Operations Research*, (2004) In Press.
- [16] Shewhart, W. A.: *Economic Control of Quality of Manufactured Product*, New York, (1931).
- [17] Chase, R.B., Aquilano, N.J., Jacobs, R.F.: *Production and Operations Management: Manufacturing and Services*, McGraw-Hill, (1998).