A study on FOREX forecasting steps utilizing neural network model

H. Babaee[1], A. khosravi[1], A. Alfi [2]
alireza_khosravi@yahoo.com
hb.babaee@gmail.com
[1] Faculty of Electrical and Computer Engineering, Noushirvani University of Technology, Babol 47135-484, Iran
2 Faculty of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood 36199-95161, Iran

Paper Reference Number: 0104-710
Name of the Presenter: Hossein Babaee

Abstract

In this paper, forecasting steps of foreign currency exchange rate (FOREX) utilizing neural network is proposed and assessed. At first the concept of foreign exchange market is introduced and the influence aspects in forecasting FOREX are addressed. In this regard, the neural network is described as a powerful tool for modeling the market. The process of modeling is represented step by step. Finally, the prediction market with artificial model is analyzed and its applicability and beneficially of the proposed method is shown.

Key words: Foreign currency exchange rate, Artificial neural network, Forecasting

1. Introduction

Creating many international businesses, the globalization has made the international trade, international financial transactions and investment to rapidly grow. Globalization is followed by foreign exchange market also known as FOREX (Abraham et. al, 2003). In the past, foreign exchange rates were only determined by the balance of payments. The balance of payments was merely a way of listing receipts and payments in international transactions for a country. Payments involve a supply of the domestic currency and a demand for foreign currencies. Receipts involve a demand for the domestic currency and a supply of foreign currencies. The balance was determined mainly by the import and export of goods. Thus, the prediction of the exchange rates was not a complex issue in the past. Unfortunately, the interest rates, and local and international supply-demand factors had become more relevant to each currency later on. On the top of this, the fixed foreign exchange rates were abandoned and a floating exchange rate system was implemented by the industrialized nations in 1973. Recently, further liberalization of trades is being discussed in General Agreement on Trade and Tariffs (Kamruzzaman and Sarker, 2004).

As international currency market is very sensitive to all forms of worldwide current affairs and governmental economics statistics releases. Such news frequently causes imbalances in the demand and supply of currencies and therefore leads to fluctuations of the exchange rates and etc (Quah et. al, 1995).Thus Traders must predict FOREX price movements in order to sell at top range and to buy at bottom range (Lai et. al, 2005). Hence we need to use various decision techniques for successful trading. In the following section, we start with a brief introduction to the FOREX. Then we describe the neural networks. We also show how an ANN can be as a predictor. At the same time, we provide a brief description of different techniques for prediction, and explain steps of NN architecture for accurate prediction and successful trading. Finally, some conclusions are made.

2.What is FOREX?

The Foreign Exchange market, also referred to as the "FOREX" or "FX" market is the largest financial market in the world, with a daily average turnover of well over US$1 trillion - 30 times larger than the combined volume of all U.S. equity markets. Unlike other financial markets, the FOREX market has no physical location or central exchange. It is an over-the-counter market where buyers and sellers including banks, corporations, and private investors conduct business. A true 24-hour market, FOREX trading begins each day in Sydney, and moves around the globe as the business day begins in each financial center, first to Tokyo, London, and New York. Unlike any other financial market, investors can respond to currency fluctuations caused by economic, social and political events (, etc.) at the time they occur - day or night. The huge number and diversity of players involved make it difficult for even governments to control the direction of the market. The unmatched liquidity and around-the-clock global activity make FOREX the ideal market for active traders (Web page, 2009). At the FOREX market, traders create prices by buying and selling currencies to exporters, importers, portfolio managers, and tourists. Each currency has two prices: a bid price at which a trader is willing to buy and an offer price at which a trader is willing to sell. If being in the major money centers banks traders deal in two way prices, for both buying and selling. In market-making banks worldwide much of the trading take place by direct dealing, while the rest takes place through brokers. Today computerized services electronically match buy and sell orders using an automated brokerage terminal (Abraham et. al, 2003).

2.1. Influence aspects in forecasting FOREX

The nature of the FOREX market is generally complex and volatile. The volatility or rate fluctuation depends on many factors. Some of factors include economic changes, demand and supply, interest rate, financing government deficits, changing hands of equity in companies, ownership of real estate, employment opportunities, social upheavals, political events, unexpected phenomena, news about worldwide turmoils, merging and ownership of large financial corporation or companies (Abraham, et. al, 2003; Quah, et. al, 1995).

3.Artificial neural networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well (Stergiou and Siganos, 2008). In other words Neural Networks are a class of nonlinear model that can approximate any nonlinear function to an arbitrary degree of accuracy and have the potential to be used as forecasting tools in many different areas. The most commonly used neural network architecture is multilayer. It consists of an input layer, an output layer and one or more intermediate layer called hidden layer. All the nodes at each layer are connected to each node at the upper layer by interconnection strength called weights (Kamruzzaman and Sarker, 2004).

3.1. Why neural networks?

The Artificial Neural Networks, the well-known function approximators in prediction and system modeling, has recently shown its great applicability in time-series analysis and forecasting (Kamruzzaman and Sarker, 2004).

ANN assists multivariate analysis. Multivariate models can rely on grater information, where not only the lagged time series being forecast, but also other indicators (such as technical, fundamental, inter-marker etc. for financial market),are combined to act as predictors. In addition, ANN is more effective in describing the dynamics of nonstationary time series due to its unique non-parametric, non-assumable, noise-tolerant and adaptive properties. ANNs are universal function approximators that can map any nonlinear function without a priori assumptions about the data (Kamruzzaman and Sarker, 2004).

Neural Networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include: 1) Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience, 2) Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time, 3) Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability, 4) Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage (Stergiou and Siganos, 2008)

3.2. Traits of one network for forecasting

Traits of one system for prediction must be: have very high forecasting accuracy, good interaction or flexibility, user-oriented by its implementation, it’s trading recommendations to be reliable. There should be adequate organization and processing of forecasting data. Preprocessing and proper sampling of input data can have impact on the forecasting performance. Choice of indicators as inputs through sensitivity analysis could help to eliminate redundant inputs. Furthermore, NN forecasting results should be used wisely and effectively.

Another important attribute of a neural net is its ability to adapt itself to new patterns, emerging in the marketplace (Chan et. al, 1995).

3.3. Neural network forecasting model

Forecasting is a process that produces a set of outputs by given a set of variables. The variables are normally historical data. Basically, forecasting assumes that future occurrences are based, at least in part, on presently observable or past events. It assumes that some aspects of the past patterns will continue into the future. Past relationships can then be discovered through study and observation. The basic idea of forecasting is to find an approximation of mapping between the input and output data in order to discover the implicit rules governing the observed movements (JingTao et. al, 2001). Then for forecasting with NN we must train algorithm to NN, that a training algorithm is used to attain a set of weights that minimizes the difference the target and actual output produced by the network. There are many different neural net learning algorithms found in the literature (Kamruzzaman and Sarker, 2004).
The performance of a neural network depends on learning algorithms with these factors, e.g., initial weights chosen, different learning parameters (learning rate and the momentum factor, etc.) used during training and the number of hidden units (Kamruzzaman and Sarker, 2003).
Figure 1 shows a samples diagram of the neural network where the above deficiency we must for having the best of forecasting my model determine w_i (initial weights), number of hidden layers, output layers.

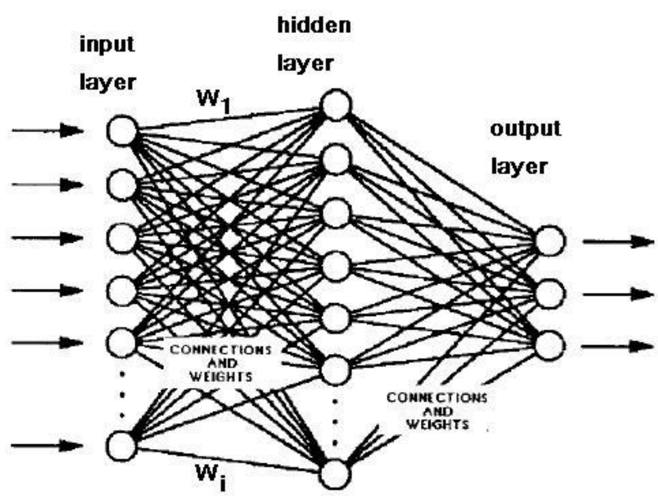

Fig 1: Simple BPNN

4.How to modeling?

The basic idea of forecasting is to find an approximation of mapping between the input and output data in order to discover the implicit rules governing the observed movements. For instance, the forecasting of exchange rate can be described in this way. Assume that u_i represents today's price, v_i represents the price after ten days. If the prediction of exchange rate after ten days could be obtained using today's exchange rate, then there should be a functional mapping u_i to v_i, where $v_i = f_i(u_i)$. That $f(\,)$ function is nonlinear generally. Using all (u_i, v_i) pairs of historical data, a general function $f(\,)$ which consists of $f_i(\,)$ could be obtained, that is $v = f(u)$. More generally, $\vec{u}$ which consists of more information in today's price could be used in function $f(\ )$. As NNs are universal approximators, we can find a NN simulating this $f(\ )$ function. The trained network is then used to predict the movements for the future (JingTao et. al, 2001),

4.1. Different types of forecasting model

There are many different neural net learning algorithms found in the literature. No study has been reported to analytically determine the generalization performance of each algorithm. In this study, we experimented with three different neural network learning algorithms, namely standard Backpropagation (BP), Scaled Conjugate Gradient Algorithm (SCG) and Backpropagation with Regularization (BPR). In the following we describe these algorithms briefly (Kamruzzaman and Sarker, 2004).

Standard Backpropagation (SBP): Backpropagation updates the weights iteratively to map a set of input vectors $(x_1, x_2,..., x_p)$ to a set of corresponding output vectors $(y_1, y_2,..., y_p)$. The input x_p is presented to the network and multiplied by the weights. All the weighted inputs to each unit of upper layer are summed up, and produce output governed by the following equations.

$$\mathrm{y_p = f(w_o h_p + \theta_o)} \tag{1}$$

$$\mathrm{h_p = f(w_h x_p + \theta_h)} \tag{2}$$

where w_o and w_h are the output and hidden layer weight matrices, h_p is the vector denoting the response of hidden layer for pattern 'p', θ_o and θ_h are the output and hidden layer bias vectors, respectively and $f(.)$ is the sigmoid activation function. The cost function to be minimized in SBP is the sum of squared error defined as

$$\mathrm{E = \frac{1}{2}\sum_p (t_p - y_p)^T (t_p - y_p)} \tag{3}$$

where t_p is the target output vector for pattern 'p'. The algorithm uses gradient descent technique to adjust the connection weights between neurons. Denoting the fan-in weights to a single neuron by a weight vector w, its update in the t-th epoch is governed by the following equations.

$$\mathrm{\Delta w_t = -\eta \nabla E(w)\big|_{w=-w(t)} + \alpha\, \Delta w_{t-1}} \tag{4}$$

The parameters η and α are the learning rate and the momentum factor, respectively. The learning rate parameter controls the step size in each iteration. For a large-scale problem BP learns very slowly and its convergence largely depends on choosing suitable values of η and α by the user.

Scaled Conjugate Gradient (SCG): The error surface in BP may contain long ravines with sharp curvature and gently sloping floor which causes slow convergence. In conjugate gradient methods, a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. In steepest descent search, a new direction is perpendicular to the old direction. This approach to the minimum is a zigzag path and one step can be mostly undone by the next. In conjugate gradient methods, a new search direction spoils as little as possible the minimization achieved by the previous direction and the step size is adjusted in each iteration. The general procedure to determine the new search direction is to combine the new steepest descent direction with the previous search direction so that the current and previous search directions are conjugate. Conjugate gradient techniques are based on the assumption that, for a general nonquadratic error function, error in the neighborhood of a given point is locally quadratic. The weight changes in successive steps are given by the following equations.

$$w_{t+1} = w_t + \alpha_t d_t \tag{5}$$

$$d_t = -g_t + \beta_t d_{t-1} \tag{6}$$

with

$$g_t = \nabla E(w)\Big|_{w = wt} \tag{7}$$

$$\beta_t = \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}} \ or\ \beta_t = \frac{\Delta g_{t-1}^T g_t}{g_{t-1}^T d_{t-1}} \ or\ \beta_t = \frac{\Delta g_{t-1}^T g_t}{g_{t-1}^T g_{t-1}} \tag{8}$$

where d_t and d_{t-1} are the conjugate directions in successive iterations. The step size is governed by the coefficient α_t and the search direction is determined by β_t. In scaled conjugate gradient the step size α_t is calculated by the following equations.

$$\alpha_t = -\frac{d_t^T g_t}{\delta_t} \tag{9}$$

$$\delta_t = d_t^T H_t d_t + \lambda_t \|d_t\|^2 \tag{10}$$

where λ_t is the scaling co-efficient and H_t is the Hessian matrix at iteration t. λ is introduced because, in case of non-quadratic error function, the Hessian matrix need not be positive definite. In this case, without λ, δ may become negative and weight update may lead to an increase of error function. With sufficiently large λ, the modified Hessian is guaranteed to be positive $(\delta > 0)$. However, for large values of λ, step size will be small. If the error function is not quadratic or $\delta < 0$, λ can be increased to make $\delta > 0$. In case of $\delta < 0$, Moller suggested the appropriate scale coefficient $\bar{\lambda}_t$ to be

$$\bar{\lambda}_t = 2\left(\lambda_t - \frac{\delta_t}{\|d_t\|^2}\right) \tag{11}$$

Rescaled value $\bar{\delta}_t$ of δ_t is then be expressed as

$$\bar{\delta}_t = \delta_t + (\bar{\lambda}_t - \lambda_t)\|d_t\|^2 \tag{12}$$

The scaled coefficient also needs adjustment to validate the local quadratic approximation. The measure of quadratic approximation accuracy, Δ_t is expressed by

$$\Delta_t = \frac{2\{E(w_t) - E(w_t + \alpha_t d_t)\}}{\alpha_t d_t^T g_t} \tag{13}$$

If Δ_t is close to 1 then the approximation is a good one and the value of λ_t can be decreased. On the contrary if Δ_t is small, the value of λ_t has to be increased. Some prescribed values suggested in are as follows

$$\begin{cases} \text{for } \Delta_t > 0.75 & \lambda_{t+1} = \lambda_t / 2 \\ \text{for } \Delta_t < 0.25 & \lambda_{t+1} = 4\lambda_t \\ \text{otherwise} & \lambda_{t+1} = \lambda_t \end{cases} \tag{14}$$

Bayesian Reguralization (BPR): A desired neural network model should produce small error not only on sample data but also on out of sample data. To produce a network with better generalization ability, MacKay proposed a method to constrain the size of network parameters by regularization. Regularization technique forces the network to settle to a set of weights and biases having smaller values. This causes the network response to be smoother and less likely to overfit and capture noise. In regularization technique, the cost function F is defined as

$$F = \gamma E_D + (1-\gamma)E_W \tag{15}$$

where E_D the same as is E defined in Eq. (3), $E_w = \|w\|^2 / 2$ is the sum of squares of the network parameters, and $\gamma(< 1.0)$ is the performance ratio parameter, the magnitude of which dictates the emphasis of the training on regularization. A large γ will drive the error E_D to small value whereas a small γ will emphasize parameter size reduction at the expense of error and yield smoother network response. One approach of determining optimum regularization parameter automatically is the Bayesian framework. It considers a probability distribution over the weight space, representing the relative degrees of belief in different values for the weights. The weight space is initially assigned some prior distribution. Let $D = \{x_m, t_m\}$ be the data set of the input-target pair, m being a label running over the pair and M be a particular NN model. After the data is taken, the posterior probability distribution for the weight $p(w|D,\gamma,M)$ is given according to the Bayesian rule.

$$p(w|D,\gamma,M) = \frac{p(D|w,\gamma,M)p(w|\gamma,M)}{p(D|\gamma,M)} \tag{16}$$

where $p(w|\gamma,M)$ is the prior distribution, $p(D|w,\gamma,M)$ is the likelihood function and $p(D|\gamma,M)$ is a normalization factor. In Bayesian framework, the optimal weight should maximize the posterior probability $p(w|D,\gamma,M)$, which is equivalent to maximizing the function in Eq.(14). The performance ration parameter is optimized by applying the Bayes' rule

$$p(\gamma|D,M) = \frac{p(D|\gamma,M)p(\gamma|M)}{p(D|M)} \tag{17}$$

If we assume a uniform prior distribution $p(\gamma|M)$ for the regularization parameter γ, then maximizing the posterior probability is achieved by maximizing the likelihood function $p(D|\gamma,M)$. Since all probabilities have a Gaussian form it can be expressed as

$$p(D|\gamma,M) = (\pi/\gamma)^{-N/2}[\pi/(1-\gamma)]^{-L/2} Z_F(\gamma) \tag{18}$$

where L is the total number of parameters in the NN. Supposing that F has a single minimum as a function of w at w^* and has the shape of a quadratic function in a small area surrounding that point, Z_F is approximated as

$$Z_F \approx (2\pi)^{L/2} \det^{-1/2} H^* \exp(-F(w^*)) \tag{19}$$

where $H = \gamma \nabla^2 E_D + (1-\gamma)\nabla^2 E_W$ is the Hessian matrix of the objective function. Using Eq. (18) into Eq. (17), the optimum value of γ at the minimum point can be determined. Foresee and Hagan propose to apply Gauss-Newton approximation to Hessian matrix, which can be conveniently implemented if the Lebenberg-Marquart optimization algorithm is used to locate the minimum point. This minimizes the additional computation required for regularization.

5. Conclusion

In this paper, the artificial neural network-based method for predicting the foreign currency exchange rates is presented. Based on this method, the FOREX market and influence aspects in forecasting FOREX were addressed. Moreover, the ANN technique and beneficially of ANN for forecasting were discussed. The NN forecasting models were also explained and different types of forecasting models were described and compared. The adaptively and simplicity in implementation of ANN depicted its ability for forecasting the foreign currency exchange rates.

References

Abraham, A., Chowdhury, M. U., & Petrovic-Lazarevic S. (2003).Australian FOREX Market Analysis Using Connectionist Models. *International Journal of Management*, 29, 18-22.

Chan,Keith C. C., & Teong, Foo Kean(1995). Enhancing Technical Analysis In The FOREX Market Using Neural Networks.*IEEE International Conference on Neural Networks*, 2.

Gan, W. S., & Ng, K. H. (1995). Multivariate FOREX forecasting using artificial neural networks. *IEEE International Conference on Neural Networks*, 2.

JingTao , Y. A. O., North, P., & Tan C. L. (2001). Guidelines for financial forecasting with neural networks. *International Conference of Neural Information Processing,* Shanghai, China.

Kamruzzaman J., Sarker R. A. (2004). ANN-Based Forecasting of Foreign Currency Exchange Rates. *Neural Information Processing-Letters and Reviews*, 3(2), 49-58.

Kamruzzaman, J., Sarker, R., & Australia, A. (2003). Forecasting of currency exchange rates using ANN: A case study. *Proceedings of the IEEE International Conference on Neural Network and Signal Processing*, pp. 793-797.

Lai, K. K., Yu, L., & Wang, S. Y. (2005). A neural network and web-based decision support system for FOREX forecasting and trading. *Lecture Notes in Artificial Intelligence,* vol. 3327, 243-253.

Luca, C. (1997). Technical analysis applications in the global currency markets. *NewYork Institute of Finance Press.*

Quah, T. S., Tan, C. L., Teh, H. H., & Sriniivasan, B. S. (1995). Utilizing a neural logic expert system in currency option trading. *Expert Systems with Applications Journal*, 9(2), 213-222.

Stergiou C. & Siganos D. (2008). Neural Networks., web page.

Web page, (2009). www.realtimeFOREX.com. (accessed on May 2009).