Forex Forecasting: A Comparative Study of LLWNN and NeuroFuzzy Hybrid Model

Article - March 2013
DOI: 10.5120/11188-6451

3 authors:

Puspanjali Mohapatra
International Institute of Information Techno...
6 PUBLICATIONS  7 CITATIONS
SEE PROFILE

Anirudh Munnangi
University of Cincinnati
3 PUBLICATIONS  2 CITATIONS
SEE PROFILE

Tapas Kumar Patra
College of Engineering & Technology, Bhuba...
7 PUBLICATIONS  10 CITATIONS
SEE PROFILE

All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Anirudh Munnangi
Retrieved on: 12 November 2016
Forex Forecasting: A Comparative Study of LLWNN and NeuroFuzzy Hybrid Model

Puspanjali Mohapatra  
(Asst. Prof)  
Dept of Computer Science and Engineering  
IIIT Bhubaneswar

Munnangi Anirudh  
(Reader)  
Dept of Electronics and Telecommunication Engineering  
IIIT Bhubaneswar

Tapas Kumar Patra  
(Director)  
Dept of Instrumentation & Electronics Engineering  
College of Engineering Technology  
Bhubaneswar, India

ABSTRACT

This paper shows how the performance of the basic Local Linear Wavelet Neural Network model (LLWNN) can be improved with hybridizing it with fuzzy model. The new improved LLWNN based Neurofuzzy hybrid model is used to predict two currency exchange rates i.e. the U.S. Dollar to the Indian Rupee and the U.S. Dollar to the Japanese Yen. The forecasting of foreign exchange rates is done on different time horizons for 1 day, 1 week and 1 month ahead. The LLWNN and Neurofuzzy hybrid models are trained with the backpropagation training algorithm. The two performance measurers i.e. the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE) show the superiority of the Neurofuzzy hybrid model over the LLWNN model.

Keywords

Forex, Foreign Exchange Market, LLWNN, LLWNN based Neurofuzzy hybrid model, Back Propagation training algorithm.

1. INTRODUCTION

The foreign exchange market which is usually referred to as Forex [1] is a large business with a large turnover in which trading takes place round the clock and all over the world. International business is totally controlled by international transactions which are settled in the near future. Therefore the exchange rate forecasting is required to evaluate the foreign denominated cashflows involved in international transactions. It can determine the benefits and risks attached to the international business environment. Currency exchange refers to the trading of one currency against another. Generally consumers consult to the currency exchange to convert their own home currency to the desired foreign currency.

Forex was established in 1971 when floating exchange rates began to materialize. In terms of trading volume, it is the world’s largest market with daily trading volumes in excess of $1.5 trillion U.S. Dollars. It is the most liquid market. The major currencies involved in the Forex transactions are the U.S. Dollar(USD), Japanese Yen(JPY), Euro(EUR), Swiss Franc(CHF), British Pound (GBP), Canadian Dollar(CAD) and Australian Dollar(AUD) where most of the currencies are rated against the USD. Trading between two nondollar currencies occur by first trading one against USD and then trading the USD against the second nondollar currency. Here the exchange rate is referred to as the cross rate.

The Forex economic time series is highly dynamic, nonlinear, complicated, random, chaotic and nonparametric in nature. Further the timeseries is highly nonstationary and have frequent structural breaks[2] and they are also affected by many economic factors, political events, government policies, investors expectations, psychology of traders and investors. The quantity and quality of the data points, the presence of various external issues, inflation rate make the modeling process a very difficult task. These complexities motivated the researchers to use various statistical models like ARMA, ARIMA[3,4], ARCH, GARCH, GARCH-M,EGARCH, IGARCH [5], Box and Jenkins approach[6] along with various soft computing and evolutionary computing methods[7,8]. Artificial Neural Network(ANN), fuzzy set theory, Support Vector Machine(SVM) etc. are considered under soft computing techniques where as the various evolutionary learning algorithms include Genetic algorithms(GA)[9], Ant Colony Optimization(ACO)[10], Particle Swarm Optimization (PSO)[11,12,13], Differential Evolution (DE)[14,15], Bacterial Foraging Optimization (BFO)[16] etc. The Literature survey reveals that different types of ANN’s like Radial Basis Function(RBF) [17], Multi Layer Perceptron (MLP) [18], Recurrent Neural Network(RNN) [19], Time delay Neural Network (TDNN) [20], Machine Learning techniques [21], Functional Link Artificial Neural Network(FLANN) [22,23,24], Local Linear Wavelet Neural Network (LLWNN) [25], Evolutionary Neurofuzzy NN[26,27], and various Neurofuzzy hybrid models have been used for time series forecasting. Statistical method can handle only linear data, they become unable to follow the non-linear pattern hidden within the exchange rate data. In this paper, a local linear wavelet neural network is proposed in which the connection weights between the hidden layer units and output units are replaced by the local linear model. At first the LLWNN model is trained with the backpropagation training algorithm. Later on a LLWNN based neurofuzzy hybrid model is designed and it is also trained with the same back propagation training algorithm. When compared simulation results of Neurofuzzy hybrid model shows it’s superiority over the LLWNN model. Here LLWNN model can recognize the patterns and adapt themselves to some extent to cope with the changing environment while the Neurofuzzy hybrid model incorporate the human knowledge and expertise for inferencing and decision making.

This paper is organized as follows. Section 2 deals with the basic principle of LLWNN. The basic principle of Neurofuzzy hybrid model is dealt in section 3. Section 4 deals with the BP learning algorithm used for both models. Performance of both models are discussed in section 5. Outputs of both...
2. BASIC PRINCIPLE OF LOCAL LINEAR WAVELET NEURAL NETWORK (LLWNN)

In Local Linear Wavelet Neural Network (LLWNN) the number of neurons in the hidden layer is equal to the number of inputs and the connection weights between the hidden layer units and output units are replaced by a local linear model. From the literature of LLWNN, It is known that the local linear model provides a more parsimonious interpolation in a high dimensional space providing it’s suitability for time series forecasting. This local capacity of the LLWNN model provides some advantages like learning efficiency and the structure transparency. This model suffers from shortcoming that for higher dimensional problems many hidden layer units are needed. Here the LLWNN model is chosen for 1 day, 1 week and 1 month ahead forecasting of Forex. The connection weights between the hidden layer and output layer of conventional wavelet neural networks are replaced with local linear model to form LLWNN model. The architecture of LLWNN model is shown in the Fig. 1.

Generally wavelets are represented in the following form.

$$\psi_i = \{a_i|^{-1/2}\psi_i(\frac{x-b_i}{a_i}): a_i, b_i \in \mathbb{R}^n, i \in \mathbb{Z}\}$$  \hspace{1cm} (1)

From the literature of LLWNN, the architecture of proposed LLWNN based Neurofuzzy hybrid model is shown in Fig. 2. The Neurofuzzy hybrid system is responsible for the determination of the knowledge base which consists of the following subsystems of developing the membership function, defining a fuzzy reasoning mechanism, the number of rule and rule base. The proposed LLWNN based Neurofuzzy hybrid model uses a LLWNN based combination of input variables. Each fuzzy rule corresponds to a sub-LLWNN, comprising a link. The LLWNN model realizes a fuzzy IF-THEN rule in the following form and the same “P” number of patterns “Xp” is passed through the linear combiner and multiplied with the weight to generate the partial sum.

Rule j:

IF $x_1$ is $A_{ij}$ and $x_2$ is $A_{ij}$ . . . . and $x_i$ is $A_{ij}$ . . . . and $x_N$ is $A_{ij}$

THEN $\hat{y}_j = \sum_{k=1}^{M} w_{ij} \phi_k$

$$= w_{ij} \phi_i + w_{ij} \phi_j + \ldots + w_{ij} \phi_M$$  \hspace{1cm} (6)

where $x_i$ and $\hat{y}_i$ are the input and local output variables, respectively;

$A_{ij}$ is the linguistic term of the precondition part with Gaussian membership function, $N$ is the number of input variables, $w_{ij}$ is the link weight of the local output, $\phi_k$ is the basis function of input variables, $M$ is the number of basis function, and rule $j$ is the $j^{th}$ fuzzy rule.

$$x = \sqrt{p_1^2 + p_2^2 + ... + p_n^2}$$  \hspace{1cm} (3)

Instead of the straight forward weight $w$, a linear model $v_j = (w_{ij} + w_{ij} x_1 + \ldots + w_{ij} x_n)$ is introduced. The activities of the linear models $v_j (i=1,2,\ldots,M)$ are determined by the associated locally active wavelet functions $\psi_i (x)$ ($i=1,2,\ldots,M$), thus $v_j$ is only locally significant.

The standard output of Wavelet Neural Network is

$$f(x) = \sum_{i=1}^{M} w_i \psi_i(x)$$

$$= \sum_{i=1}^{M} w_i |a_i|^{-1/2} \psi_i(\frac{x-b_i}{a_i})$$  \hspace{1cm} (4)

where $\psi_i$ is the wavelet activation of $i^{th}$ unit of hidden layer and $w_i$ is the weight connecting the $i^{th}$ unit of hidden layer to the output layer unit.

In LLWNN model the output in the output layer is

$$Y = \sum_{i=1}^{M} (w_{i0} + w_i x_1 + \ldots + w_i x_n) |a_i|^{-1/2} \psi_i(\frac{x-b_i}{a_i})$$  \hspace{1cm} (5)

3. BASIC PRINCIPLE OF LLWNN BASED NEUROFUZZY HYBRID MODEL

The architecture of proposed LLWNN based Neurofuzzy Hybrid Model is shown in Fig. 2. The Neurofuzzy hybrid system is responsible for the determination of the knowledge base which consists of the following subsystems of developing the membership function, defining a fuzzy reasoning mechanism, the number of rule and rule base. The proposed LLWNN based Neurofuzzy hybrid model uses a LLWNN based combination of input variables. Each fuzzy rule corresponds to a sub-LLWNN, comprising a link. The LLWNN model realizes a fuzzy IF-THEN rule in the following form and the same “P” number of patterns “Xp” is passed through the linear combiner and multiplied with the weight to generate the partial sum.

Rule j:

IF $x_1$ is $A_{ij}$ and $x_2$ is $A_{ij}$ . . . . and $x_i$ is $A_{ij}$ . . . . and $x_N$ is $A_{ij}$

THEN $\hat{y}_j = \sum_{k=1}^{M} w_{ij} \phi_k$

$$= w_{ij} \phi_i + w_{ij} \phi_j + \ldots + w_{ij} \phi_M$$  \hspace{1cm} (6)

where $x_i$ and $\hat{y}_i$ are the input and local output variables, respectively;

$A_{ij}$ is the linguistic term of the precondition part with Gaussian membership function, $N$ is the number of input variables, $w_{ij}$ is the link weight of the local output, $\phi_k$ is the basis function of input variables, $M$ is the number of basis function, and rule $j$ is the $j^{th}$ fuzzy rule.
The operation functions of the nodes in each layer of the LLWNN model is now described. In the following description, $u^{(l)}_i$ denotes the output of a node in the $l$th layer.

Layer 1:
No computation is performed in layer 1. Each node in this layer only transmits input values to the next layer directly.

$$u^{(1)}_i = x_i$$  
(7)

Layer 2:
Each fuzzy set $A_{ij}$ is described here by a Gaussian membership function. Therefore, the calculated membership value in layer 2 is

$$u^{(2)}_{ij} = \exp \left( -\frac{[u^{(1)}_i - m_{ij}]^2}{\sigma_{ij}^2} \right)$$  
(8)

where $m_{ij}$ and $\sigma_{ij}$ are the mean and variance of the Gaussian membership function, respectively, of the $j$th term of the $i$th input variable $x_i$.

Layer 3:
Nodes in layer 3 receive one-dimensional membership degrees of the associated rule from the nodes of a set in layer 2. Here, the product operator described earlier is adopted to perform the precondition part of the fuzzy rules. As a result, the output function of each inference node is

$$u^{(3)}_j = \prod_i u^{(2)}_{ij}$$  
(9)

where the $\prod_i u^{(2)}_{ij}$ of a rule node represents the firing strength of its corresponding rule.

Layer 4:
Nodes in layer 4 are called consequent nodes. The input to a node in layer 4 is the output from layer 3, and the other inputs are calculated from the LLWNN that has used the function $\tanh(\cdot)$, as shown in Fig. 1. For such a node

$$u^{(4)}_j = u^{(3)}_j \sum_k w_{kj} \phi_k$$  
(10)

where $w_{kj}$ is the corresponding link weight of the LLWNN and $\phi_k$ is the functional expansion of input variables.

Layer 5:
The output layer in node 5 acts as the defuzzification layer. So, the final output of the LLWNN model $y$ is expressed as

$$y = (y_1 + Fz_{11} + y_2 + Fz_{22}) / (Fz_{11} + Fz_{22})$$  
(11)

where $y_1$ and $y_2$ are the output from the LLWNN model and $Fz_{11}$ and $Fz_{22}$ are the output from the NeuroFuzzy hybrid model or output from layer 3.

4. BACK PROPAGATION (BP) LEARNING ALGORITHM

Back propagation algorithm is one of the tested supervised learning algorithm. It minimises the objective function by adjusting the link weights used to develop the models. The gradient of the cost function with respect to that particular weight parameter is calculated and the parameters are updated with the negative gradient.

$$E = \frac{1}{2} \left[ y(t) - w_{10} \psi_1(x) - w_{11} p_1 \psi_1(x) - \cdots - w_{m0} p_m \psi_m(x) - w_{m1} p_1 \psi_m(x) - \cdots - w_{mn} p_n \psi_m(x) \right]^2$$  
(12)

where $y(t)$ is the desired value. A weight updation from $i^{th}$ to $(i+1)^{th}$ iteration i.e., from $w(t)$ to $w(t+1)$ is given by

$$w(t+1) = w(t) + \Delta w(t) = w(t) + (-\eta \frac{\partial E}{\partial w(t)})$$  
(13)

where $\frac{\partial E}{\partial w}$ for all weights are described by equation (14) to (19).

$$\frac{\partial E}{\partial w_{ij}} = w_{ij} + \eta \cdot e \cdot \left( \frac{1}{2} \right) \cdot \left( x_1^2 + x_2^2 + \cdots + x_n^2 \right) \cdot \exp \left( -\frac{\left( (x_1 - c_1)^2 + (x_2 - c_1)^2 + \cdots + (x_n - c_1)^2 \right)}{\sigma_{11}^2} \right)$$  
(14)

For $\forall j \neq 0$,

$$\frac{\partial E}{\partial w_{ij}} = w_{ij} + \eta \cdot e \cdot \left( \frac{1}{2} \right) \cdot \left( x_1^2 + x_2^2 + \cdots + x_n^2 \right) \cdot \exp \left( -\frac{\left( (x_1 - c_1)^2 + (x_2 - c_1)^2 + \cdots + (x_n - c_1)^2 \right)}{\sigma_{11}^2} \right) \cdot x_j$$  
(15)

\[ \text{i.e.,} \]

$$\frac{\partial E}{\partial w_{10}} = w_{10} + \eta \cdot e \cdot \left( \frac{1}{2} \right) \cdot \left( x_1^2 + x_2^2 + \cdots + x_n^2 \right) \cdot \exp \left( -\frac{\left( (x_1 - c_1)^2 + (x_2 - c_1)^2 + \cdots + (x_n - c_1)^2 \right)}{\sigma_{11}^2} \right)$$  
(16)

$$\frac{\partial E}{\partial w_{22}} = w_{22} + \eta \cdot e \cdot \left( \frac{1}{2} \right) \cdot \left( x_1^2 + x_2^2 + \cdots + x_n^2 \right) \cdot \exp \left( -\frac{\left( (x_1 - c_1)^2 + (x_2 - c_1)^2 + \cdots + (x_n - c_1)^2 \right)}{\sigma_{11}^2} \right)$$  
(17)

$$\frac{\partial E}{\partial w_{20}} = w_{20} + \eta \cdot e \cdot \left( \frac{1}{2} \right) \cdot \left( x_1^2 + x_2^2 + \cdots + x_n^2 \right) \cdot \exp \left( -\frac{\left( (x_1 - c_1)^2 + (x_2 - c_1)^2 + \cdots + (x_n - c_1)^2 \right)}{\sigma_{11}^2} \right)$$  
(18)

$$\frac{\partial E}{\partial w_{11}} = w_{11} + \eta \cdot e \cdot \left( \frac{1}{2} \right) \cdot \left( x_1^2 + x_2^2 + \cdots + x_n^2 \right) \cdot \exp \left( -\frac{\left( (x_1 - c_1)^2 + (x_2 - c_1)^2 + \cdots + (x_n - c_1)^2 \right)}{\sigma_{11}^2} \right)$$  
(19)

Other weights are also updated like this. $\eta$ is the learning rate used in LLWNN model.
Fig 2: Architecture of proposed LLWNN based NEUROFUZZY Hybrid Model

Table 1. Details of Forex DataSets

<table>
<thead>
<tr>
<th>FOREX DATA SETS</th>
<th>Total Training Samples</th>
<th>Total Testing Samples</th>
<th>Training Sample</th>
<th>Testing Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dollar to Rupee</td>
<td>1st-MAY 2008 to 1st-MAY 2011</td>
<td>1st-June 2011 to 1st-June 2012</td>
<td>1095</td>
<td>366</td>
</tr>
<tr>
<td>Dollar to Yen</td>
<td>1st-MAY 2008 to 1st-MAY 2011</td>
<td>1st-June 2011 to 1st-June 2012</td>
<td>1095</td>
<td>366</td>
</tr>
</tbody>
</table>

5. STUDY OF PERFORMANCE OF LLWNN and LLWNN BASED NEUROFUZZY MODEL

The daily Forex data for Dollar to Rupee and Dollar to Yen are considered here as the experimental data. Here the models are forecasting for 1 day, 1 week and 1 month ahead. All the inputs are normalized within a range of [0, 1] using the following formula.

$$X_{\text{norm}} = \frac{X_{\text{orig}} - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$ (20)

Where $X_{\text{norm}}$ → normalized value, $X_{\text{orig}}$→original exchange rate, $X_{\text{min}}$ and $X_{\text{max}}$ are the daily minimum and maximum prices of the corresponding Forex data.

Details of the FOREX datasets are given in table 1.
Here each Forex dataset is divided into two sets, one for training and one for testing. The duration of the sample, total number of samples, number of training samples and testing samples are given in table 1. Different lagged values are taken as input to the proposed model. The daily, weekly and monthly periodicity and their trend are taken into consideration. However we can also consider other technical indicators as inputs to the models. Here in this paper simple moving average is used as the technical indicator for both the models. Training of the LLWNN and Neurofuzzy models are carried out using the BP algorithm given in Section 4 and the optimum weights are obtained. Then using the trained model, the forecasting performance is tested using test patterns for 1 day, 1 week, and 1 month ahead. The Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are used to measure the performance of the proposed models.

The MAPE is defined as

\[ \text{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \]  

(21)

\[ \text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \]  

(22)

where \( y_i \) and \( \hat{y}_i \) are the desired and predicted value respectively. 'N' represents the total number of samples under test.

6. MODEL OUTPUT

The performance of BP based LLWNN and Neurofuzzy model is given in Table 2 and Table 3. Model outputs for various time horizons during training and testing are presented here from fig. 3-16.
Fig 7: MAPE during 1 day ahead training (Dollar to Rupee)

Fig 8: One day ahead prediction during testing of Neurofuzzy Hybrid Model (Dollar to Rupee)

Fig 9: One week ahead prediction during testing of Neurofuzzy Hybrid Model (Dollar to Rupee)

Fig 10: One month ahead prediction during testing of Neurofuzzy Hybrid Model (Dollar to Rupee)

Fig 11: One day ahead prediction during training (Dollar to Yen)

Fig 12: Error during 1 day ahead training (Dollar to Yen)
Table 2. Performance of BP based LLWNN model (Training and Testing)

<table>
<thead>
<tr>
<th>FOREX Data</th>
<th>Duration</th>
<th>RMSE (train)</th>
<th>MAPE (%) (train)</th>
<th>RMSE (test)</th>
<th>MAPE (%) (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOLLAR to RUPEE</td>
<td>1-day</td>
<td>0.0235</td>
<td>5.2125</td>
<td>0.0685</td>
<td>4.284</td>
</tr>
<tr>
<td></td>
<td>1-week</td>
<td>0.0387</td>
<td>4.9467</td>
<td>0.1315</td>
<td>7.86</td>
</tr>
<tr>
<td></td>
<td>1-month</td>
<td>0.0242</td>
<td>3.7232</td>
<td>0.1735</td>
<td>9.8955</td>
</tr>
<tr>
<td>DOLLAR to YEN</td>
<td>1-day</td>
<td>0.0379</td>
<td>3.5469</td>
<td>0.1424</td>
<td>6.76</td>
</tr>
<tr>
<td></td>
<td>1-week</td>
<td>0.0405</td>
<td>4.2679</td>
<td>0.1538</td>
<td>9.0047</td>
</tr>
<tr>
<td></td>
<td>1-month</td>
<td>0.0419</td>
<td>4.7473</td>
<td>0.1983</td>
<td>12.586</td>
</tr>
</tbody>
</table>

7. ANALYSIS OF RESULT

The experiment undertaken in this paper has taken into account two models, two data sets and three time horizons. The results are presented in terms of target vs. predicted values and error convergence speed. A comparative analysis of the performance of both the models for 1 day, 1 week and 1 month in advance is presented. The two normalized Forex indices i.e. Dollar to Rupee and Dollar to Yen are shown in Fig. 3. Fig. 4-10 and Fig. 11-16 deals with various plots of Dollar to Rupee and Dollar to Yen datasets respectively. One day ahead training prediction error, RMSE and MAPE plots for both the models for Dollar to Rupee dataset are presented in Fig 4-7 respectively. Fig. 8-10 represents 1 day, 1 week and 1 month ahead testing prediction of proposed Neurofuzzy hybrid model for Dollar to Rupee dataset. Similarly 1 day ahead training prediction error, RMSE and MAPE plots for both the models for Dollar to Yen dataset are presented in Fig 11-14 respectively. Fig. 15-16 represents 1 day and 1 week ahead testing prediction of proposed Neurofuzzy hybrid model for Dollar to Yen dataset. The performance measurers RMSE and MAPE for Dollar to Rupee, Dollar to Yen during training and testing of BP based LLWNN model and BP based Neurofuzzy hybrid model during different time horizons are mentioned in Table 2 and 3 respectively. The performance of Neurofuzzy hybrid model is found to be better in comparison to simple LLWNN model.

8. CONCLUSION

Accurate Forex forecasting is always a very challenging task. The proposed Neurofuzzy hybrid model trained with back propagation is giving good result as per the recorded RMSE, and MAPE values during testing for one day, one week and one month ahead respectively in comparison to the simple LLWNN model trained with back propagation. The Neurofuzzy hybrid model is proved to be better in terms of prediction accuracy, error convergence speed and is more capable to handle more uncertainties in comparison to simple LLWNN. Further the prediction performance of LLWNN and Neurofuzzy hybrid model is to be verified with GA, PSO and DE based training algorithms.
9. REFERENCES


