MATEC Web of Conferences **68**, 19001 (2016) DOI: 10.1051/matecconf/20166819001
*ICIEA 2016*

# Forex Market Prediction Using NARX Neural Network with Bagging

Nima Shahbazi[1], Masoud Memarzadeh[2] and Jarek Gryz[1]

[1]*Department of Electrical Engineering and Computer Science, York University, Canada*
[2]*Department of Mathematics, Tehran University, Iran*

**Abstract.** We propose a new methodfor predicting movements in Forex market based on NARX neural network withtime shifting bagging techniqueand financial indicators, such as relative strength index and stochastic indicators. Neural networks have prominent learning ability but they often exhibit bad and unpredictable performance for noisy data. When compared with the static neural networks, our method significantly reducesthe error rate of the responseandimproves the performance of the prediction. We tested three different types ofarchitecture for predicting the response and determined the best network approach. We applied our method to prediction the hourly foreign exchange rates and found remarkable predictability in comprehensive experiments with 2 different foreign exchange rates (GBPUSD and EURUSD).

## 1 Introduction

The Forex Market is the largest and most liquid market around the word with an estimated $2 trillion traded every day.Modelling the system is difficult as currency market is characterized by significant non-linearity and low signal to noise ratio. Also, foreign exchange rates are affected not only by economic news but alsopolitical and psychological factors. Not surprisingly, predicting the movement of foreign exchange rates or finding a trend in the market is very difficult.

There are several neural network approaches [1] and [2] which ledto successful predictions. However, these approaches use moving averages for the input of their networks and do not use any bagging which,as we will showlater, improves the response prediction. It has been shown in [3] that the relationship between neural networks and conventional statistical approaches for time series forecasting are complementary. It has also been indicated in [4] that traditional statistical techniques for forecasting have reached theirlimit inapplications with nonlinearities in the data set (such as stock indices).Recently, Dynamic Neural Networks such as NARX have been used in many businessapplications especially when the problem domain involves predictions [5] but again, there is no bagging or indicator selection.

This paper proposes a new method for predictingforex market movement by using NARX neural network with bagging technique. Instead of using the next close price or moving averages, we uses RSI (Relative Strength Index) indicator for the output response. We feed data aboutForex news to the networkasthere is strong correlation between the news piece and the stock price behavior[6].

Our contributions can be described as three phases:

*Phase 1*: We find the most relevant financial indicators to be fed into the network by sensitivity analysis regarding the response, the input/output indicators, and the architecture.Then we train the network with news data with static neural network (Section 0).

*Phase 2*: Using the same inputs and output,we use the NARX neural Network architecture (Section 0). By usingdynamic network both the error rates as well as the directional success improve(Section 0).

*Phase 3*: In the last phase, time shifting bagging method (Section 0)is added to the system and shown to improve the test error and directional success rate on the test data.

After presenting the experimental results, a discussion on future research concludes the paper.

## 2 Architecture and learning algorithm

### 2.1 NARX neural network as a forecasting tool for Forex

Neural Networks (NN) based prediction is a suitable tool for forecasting in non-linear time seriesapplications. With neural network approach to time series it is not necessary to know any information regarding the cause that generates the signal.

The architectural approach proposed here to deal with Forex market is based on *Nonlinear Autoregressive models with eXogenous input* (NARX model)[7]. All other dynamic networks have either been focused networks, with the dynamics only at the input layer, or feed-forward networks. NARX is a recurrent dynamic network, with feedback connections enclosing several layers of the network. It is based on the linear ARX model, which is commonly used in time-series modeling.This is a powerful class of models which has been demonstrated to be well suited for modeling nonlinear systems and especially time series. It has been shown that in NARX networks learning is more effective than in other neural networks and that these networks converge much faster and generalize better than the other networks [8].

The defining equation for the NARX model is:

$$y(t) = F(y(t-1), \dots y(t-2), y(t-n_y), u(t-1), \quad (1)$$
$$u(t-2), \dots, u(t-n_u))$$

where the next value of the dependent output signal $y(t)$ is regressed on previous values of the output signal and previous values of an independent $u(t)$(exogenous) input signal. One can implement the NARX model by using a feed-forward neural network to approximate the function $F$.

There is another important configuration for NARX model (useful in training) that we use in our approach. Let the output of the NARX network be an estimate of the output of some nonlinear dynamic system that we are trying to model. The output is fed back to the input of the feed-forward neural network as part of the standard NARX architecture. Since the true output is available during the training of the network one could create a series-parallel architecture [9] in which the true output is used instead of feeding back the estimated output as shown in Figure 1. This has two advantages. First, the input to the feed-forward network is more accurate. Second, the resulting network has purely feed-forward architecture and static back propagation can be used for training as our dynamic network is now a feed-forward neural network without any feedback.

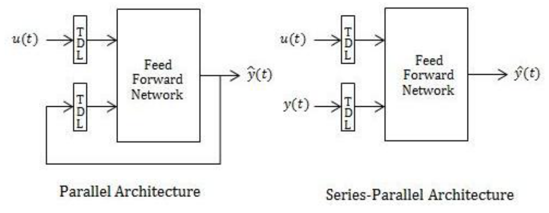

**Figure 1.** Serial/parallel architecture for NARX model.

### 2.2 Learning algorithm

For learning, a dynamic back-propagation algorithm is needed to evaluate the gradients. This is more computationally than static back-propagation. In addition, the error can be larger for dynamic than for static networks. Training is more likely to fall into local minima [10]. But asshownin Figure 1we can use instead a series-parallel architecture.Thus, it is possible to use true output instead of the estimated output to train the network with the feedback connections.

The process of training a neural network involves iteratively presenting it with the input data so that it is calibrated and can be used later as a forecasting tool. The objective of the training is to minimize a defined error function, which implies that the neural network fit the input data, given the expected result as outputs. In this approach we use RMSE (Root Mean Square Error) aserror function:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_i)^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2} \quad (2)$$

where $t_i$ is the target and $y_i$ is the predicted value. In this paper, the training function that updates the weights and bias values is the Levenberg-Marquardt function. In general, in function approximation problems, for networks that contain up to a few hundred weights, the Levenberg-Marquardt algorithm will have the fastest convergence. This advantage is especially noticeable if very accurate training is required. Also,it was shown that the neural network training can be made more efficient if certain preprocessing steps on the network inputs and targets are performed. The *normalization* of the input and target values simplifies the problem of the outliers in the network. The normalized inputs and targets that are returned will all fall in the interval[-1, 1].Another factor in the neural network isthe transfer function. The best transfer function is chosen by the best performance seen on the test result. In this problem Tangent-Sigmoid makes the learning phase faster and increases the neural network accuracy.

## 3 Model construction and evaluation

### 3.1 Dataset and input/output predictors

The prediction is done hourly. The dataset used here is pulled from Alpari-Forex.com[11]. For the news release data, two sources on relevant currencies are used: (1) Fxstreet.com[12], economic calendar from 2001; and (2) Bloomberg.com [13], economic calendar from 2001.

We define six hour window, three hours before the related news release and three hours after the news release, and gather the data (Open, Low, High and Close) every minute in the six hour window for training network. We divided the data into (70%, 15% and 15%) with 70% use for training, 15% for validation,and the rest for testing.

Once the raw data have been chosen by six hour window, a set of indicators based on those values might be developed. In this work we used five indicators for input/output of the neural network shown in Table 1. For all formulas below, the subscript *t±1* indicates time *t* plus or minus one hour.

Differential RSI (Relative Strength Index) as Input/Output

1. A technical momentum indicator that compares the magnitude of recent gains to recent losses in an attempt to determine overbought and oversold conditions of an asset. RSI ranges from 0 to 100. We use differential RSI as:

$$DRSI(t)_{input} = RSI_t - RSI_{t-1} \quad (3)$$

The output of the network will be:

$$DRSI(t+1)_{ouput} = RSI_{t+1} - RSI_t \quad (4)$$

2. Differential EMA (Exponential Moving Average) as Input

**Table 1.** Selected features and their formulae: C, closing price; H, high price; L, low price; $LL_n$, lowest low price in the last n samples; $HH_n$, highest high in the last n samples; Up, upward price change; Dw, downward price change.

| Name of features | Formula |
|---|---|
| STOCH % D | $\sum_{i=0}^{n-1} \frac{C_t - LL_{t-5}}{HH_{t-5} - LL_{t-5}} \times \frac{100}{n}$ |
| PR( Willia m %R) | $\frac{H_n - C_t}{H_n - L_n} \times 100$ |
| RSI | $100 - {}^{100}/_{(1 + (\sum_{i=0}^{n-1} Up_{t-i}/_n)/(1 + (\sum_{i=0}^{n-1} Dw}$ |
| EMA | $EMA_t = EMA_{t-1} + \alpha \times (Price_t - EMA_{t-1})$ |
| MACD | $EMA_{fast} - EMA_{slow}$ |

Exponential moving average (EMA), also known as an exponentially weighted moving average (EWMA), is a type of infinite impulse response filter that applies weighting factors that decrease exponentially. We use Differential EMA as:

$$DEMA(t)_{input} = EMA_t - EMA_{t-1} \tag{5}$$

3. Differential STOCH (Stochastic Oscillator %D) as Input

In technical analysis of securities trading, the stochastic oscillator is a momentum indicator that uses support and resistance levels. The term stochastic refers to the location of the current price in relation to its price range over a period of time. This method attempts to predict price turning points by comparing the closing price of a security to its price range. STOCH ranges from 0 to 100. We use Differential STOCH as:

$$DSTOCH(t)_{input} = STOCH_t - STOCH_{t-1} \tag{6}$$

4. MACD (Moving Average Convergence/Divergence) as Input

MACD makes use of moving averages of different time frames to indicate momentum changes and swings in the mood of the crowd, to give buying and selling signals that catches the big moves. The MACD indicator measures the difference between two moving averages (EMA):

$$MACD(t)_{input} = MACD_t - MACD_{t-1} \tag{7}$$

5. Differential PR (William %R) as Input

Williams's %R, or just %R, is a technical analysis oscillator showing the current closing price in relation to the high and low of the past N values (for a given N). Its purpose is to tell whether a stock or commodity market is trading near the high or the low, or somewhere in between, of its recent trading range. The oscillator is on a negative scale, from -100 (lowest) up to 0 (highest). We use Differential PR as:

$$DPR(t)_{input} = DPR_t - DPR_{t-1} \tag{8}$$

In this paper, the hourly period sample for each of the indicator is given in Table 2.

**Table 2.** Period (in hours) samples used in this work.

| Name of features | Period |
|---|---|
| STOCH %D | 5:sampling<br>3:smoothing |
| PR( William %R) | 14 |
| RSI | 14 |
| EMA | 14 |
| MACD | Fast:12 Slow=26 |

## 3.2 Construction phases and trading strategy results

In order to track the improvement of the method we divided the problem into three phases.All networks were trained between January 2001 and May 2014. For each experiment, a range of different training parameters and numbers of hidden nodes were tried, and the results are listed here. For each experiment, error in predicting training and test set outputs are reported, in terms of RMSE, along with the directional success.

While the accuracy of a model may be measured using RMSE (the error in directional success) what is ultimately needed is a measure of the effectiveness of the model in relation to its use in driving decisions to buy/sell shares. As the output of the model is $DRSI(t+1) = RSI_{t+1} - RSI_t$, the trading strategy will be:

$$\begin{cases} if\ DRSI(t+1) > \alpha\ Buy\ Signal \\ if\ DRSI(t+1) < \alpha\ Sell\ Signal \end{cases} \tag{9}$$

Here $\alpha$ is non-negative tuning parameter. The value of the tuning parameter is traded off between the number of trades and directional success that can be evaluated from the back testing result on historical data (we ended up using 1.22 for $\alpha$). In all phases, a range of different parameter settings and configurations of hidden nodes were evaluated and the most successful ones are documented. The results given in tables are for both

GBPUSD/EURUSD rates. The training parameters lay within the following ranges:

1. Learning rate, (0.001 to 0.3)
2. Momentum, (0.002 to 0.4)

### *3.2.1 Phase 1*

In this phase we use static feed forward network for training. The prediction errors on train/test sets are shown in Table 3.

**Table 3.** Phase 1 train/test errors.

| Set | RMSE | |
|---|---|---|
| | GBPUSD | EURUSD |
| Training Set | 3.92 | 3.66 |
| Test Set | 4.72 | 4.42 |

### *3.2.2 Phase 2*

In the second phase a series-parallel architecture as shown in Figure 1is used, so that now NARX is used as the network architecture.The improvements in RMSE from using this network are shown inTable 4.

**Table 4.** Phase 2 train/test errors.

| Set | RMSE | |
|---|---|---|
| | GBPUSD | EURUSD |
| Training Set | 3.76 | 3.36 |
| Test Set | 4.32 | 4.02 |

### *3.2.3 Phase 3*

In the last phase of the work we used the time shifting bagging method. Time shifting bagging method works as follows: we gather the data in a minute and produce the hourly rates based on M1 (minutes) data. We can sample the data in 60 different ways for hourly sampling. Thus, we can start sampling in M1 data at time t and sample repeatedly at times t-60, t-120, t-180, …. Therefore, the sampling has 60 steps starting at time t. Or we can start the sampling in M1 data at time t-1, and sample repeatedly at times t-61, t-121, t-181, … . Clearly, we can do this kind of sampling in 60 different ways to prepare the data for the input of the network. In this work we sample in 12 different ways starting at t, t-5, t-10… and t-55 and compute each of the indicators based on these datasets. Finally, we train the network and average the result of the output for 12 ways. Prediction errors are shown in Table 5.

**Table 5.** Phase 3 train/test errors GBPUSD.

| Prediction Error | RMSE | |
|---|---|---|
| | GBP | USD |
| Training Set | 3.53 | 3.23 |
| Test Set | 4.17 | 3.87 |

The results in Table 5 shows that the error rates for train/test improve remarkably when the time shifting average method is used.There is a significant change in test error rate when bagging method is used in combination with NARX neural network. InTable 6the directional success for all the phases is listed.

**Table 6.** Directional success on all the phases.

| Phase No. | Directional success rate on test date | |
|---|---|---|
| | GBPUSD | EURUSD |
| 1 | 54% | 57% |
| 2 | 60% | 63% |
| 3 | 68% | 72% |

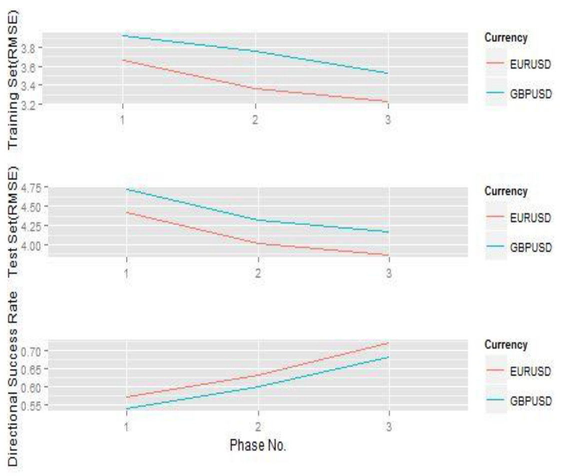

**Figure 2.** Directional success rate, test set and training set as a function of phase No. for each currency.

## 3.3 Discussion

The purpose of Phase 1 was to determine the performance of a neural network trained solely on the hourly time series data. This showed thedirectional success of 54%(GBPUSD) and 57%(EURUSD). The results indicate that this is useful; yet the success rate is not good enough to attempt to use this method for profit in the real market.

In Phase 2, the architecture was changed to NARX neural network and the directional success rate became 60%(GBPUSD) and 63%(EURUSD). By changing the method to time shifting bagging, as we did in Phase 3, we improved the rate to 68%(GBPUSD) and 72%(EURUSD) which is a significant rate of success.

## 4 Conclusions

There is an enormous range of indicators that are not accounted for in this paper. The analysis presented here shows only the benefit of considering NARX neural network and time shifting bagging method with some external indicators when predicting Forex market movement. The use of other indicators or even different bagging methods should be explored further. Also, this study was on two pairs of currencies only and should be repeated on other currency combinations.

## References

1. J. Yao and C. Tan, "A case study on using neural networks to perform Technical Forecasting of Forex," Neurocomputing, **34**, 79-98 (2000)
2. F. Ferna´ndez-Rodrı´guez, C. Gonza´lez-Martel and S. Sosviall-Rivero, "On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid Stock Market," Economics Letters, **69**, 89-94 (2000)
3. H. White, "Learning in artificial neural networks: A statistical perspective," Neural Computing, **1**, 425-464 (1989)
4. A. Refenes and G. F. A. Zapranis, "Stock performance modeling using neural networks: A comparative study with regression models," Neural Network, **5**, 961-970 (1994)
5. L. Haizhon and K. Robert, "A Dynamic Neural Network Method for Time Series Prediction Using the KIII Model," in Proceeding of the 2003 International Joint Conference on Neural Networks, (2003)
6. G. Fung, "The Predicting Power of Textual Information on Financial Markets," IEEE Intelligent Informatics Bulletin, **5**, 1-10 (2005)
7. S. Haykin, Neural Networks and Learning Machines (3rd Edition), Pearson Education, (2008)
8. T. Lin, C. L. Giles, B. G. Horne and S. Kung, "A Delay Damage Model Selection Algorithm for NARX Neural Networks," IEEE Transactions on Signal Processing, “Special Issue on Neural Networks”, **45**, 2719-2730 (1997)
9. K. S. Narendra and K. Parthasarathy, ""Learning Automata Approach to Hierarchical Multiobjective Analysis,"," IEEE Transactions on Systems, Man and Cybernetics, **20**, 263-272 (1991)
10. K. S. Narendra and K. Parthasarathy, ""Learning Automata Approach to Hierarchical Multiobjective Analysis,"," IEEE Transactions on Systems, Man and Cybernetics, **20**, 263-272 (1991)
11. "Alpari," 1 09 2013. [Online]. Available: www.alpari-forex.com.
12. "FXstreet," 1 09 2013. [Online]. Available: www.fxstreet.com.
13. "Bloomberg," 01 09 2013. [Online]. Available: www.bloomberg.com.