Forecasting Exchange Rates with Mixed Models

Laura Maria BADEA (STROIE)
Department of Economie Informatics and Cybernetics
The Bucharest University of Economic Studies
Dorobantilor no. 15-17, Bucharest, 010552
ROMANIA
laura.maria.badea@gmail.com

Abstract: Gaining accuracy in exchange rate forecasting applications provides true benefits for financial activities. Supported today by the advancements in computing power, machine learning techniques provide good alternatives to traditional time series estimation methods. Very approached in time series forecasting are Artificial Neural Networks (ANNs) which offer robust results and allow a flexible data manipulation. When integrating both, the "white-box" feature of conventional methods and the complexity of machine learning techniques, forecasting models perform even better in terms of generated errors. In this study, input variables (independent variables) are selected using an ARIMA technique and are further employed in differently configured multilayered feed-forward neural networks using Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization algorithm to perform predictions on EUR/RON and CHF/RON exchange rates. Results in terms of mean squared error highlight good results when using mixed models.

Key-Words: machine learning techniques, Artificial Neural Networks, ARIMA, exchange rate forecasting, learning algorithms, BFGS optimization

1. Introduction

Current financial and economic activities provide incentives for continuous productivity increase, determining business practitioners to be in a permanent search for new techniques and technologies that allow them to reach certain results at a higher success rate. As traditional forecasting methods proved certain limitations, machine learning techniques have started to gain popularity, supported also by the ubiquitous character of computers. Artificial Neural Networks are self-adaptive computational methods offering solutions to a series of financial and economic applications and providing good results in exchange rate forecasting problems.

This paper analyzes forecasting results of models built with mixed Artificial Neural Networks integrating both, the transparency of conventional models when selecting input variables and the complexity of advanced training algorithms.

In section two of this paper a brief reviews of ANNs performances in exchange rate applications is presented based on previous research. Section three discusses the main aspects regarding the datasets and sampling techniques. Section four provides model building details and results obtained when forecasting exchange rates with mixed neural networks. Section five of this paper will conclude on the main findings.

2. Neural Networks in exchange rate forecasting applications

Since the proposal of ARIMA models, introduced in [1], these have become reachable time series estimation techniques, providing good results in forecasting applications. However, alongside the development of the business environment and its ever-growing requirements, it also became necessary to search for models that provide more accurate outputs. Artificial Neural Networks are machine learning techniques that have gained success over the past decades supported by improved results compared with traditional time series estimation techniques.

ANNs have been tested within several exchange rate forecasting applications over the past two decades. [2], [3], [4], [5], [6], [7], [8] and [9] have all reported that Neural Networks generate better results compared with traditional ARIMA models. [10] surveyed 45 articles published between 1993 and 2004 that analyze the results of ANNs in exchange rates forecasting applications. Over half of these papers use the gradient descent back-propagation learning method, the most common type of algorithms used to train ANNs (see [11]). However, in case of noisy and limited data series, the gradient descent back-propagation approach is very exposed to local

minima problems, and therefore, very sensitive to the learning rate and momentum rate parameters. Thus, over the past years, several actions have been employed in finding optimized techniques to improve the accuracy of the classical back-propagation algorithm.
According to [12], local optimization can be divided into the following classes: non-derivative methods, first derivative (gradient) methods, and second derivative methods. Gradient descent is a first derivative method and uses gradient information calculated from the optimization function to determine the search direction on the response surface. The latter of all three optimization classes uses the second derivatives of the function (the Hessian) to determine the search direction. Examples of second derivative methods are: discrete Newton, truncated Newton, quasi-Newton, and Levenberg-Marquardt.
One quasi-Newton optimization method providing good results when training Neural Networks is Broyden-Fletcher-Goldfarb-Shanno (BFGS) introduced in 1970 ([13], [14], [15], [16]). Compared with the gradient descent method, BFGS technique uses a line search to calculate the size of the step, progressing thus towards the optimal solution at a higher success rate and at a higher speed. Moreover, compared with other second derivative approaches, BFGS algorithm requires less computational resources by using an approximation rather than an explicit calculation of the Hessian inverse matrix, which is updated after each step of the algorithm.
Starting from the Newton equation below, the search direction is determined in equation (2):

$$A_k d_k = -\nabla E(W_k) \quad (1)$$

Where, k is the current iteration, A_k is the Hessian matrix, d_k is the search direction and $-\nabla E(W_k)$ is the gradient of the weigh vector W_k,

$$d_k = -H_k \nabla E(W_k) \quad (2)$$

Where, H_k is the inverse of the Hessian matrix A_k ?
The approximation formula of the Hessian inverse matrix used by BFGS algorithm is given by:

$$H_{k+1} = \left(I - \frac{S_k Y_k^T}{Y_k^T S_k} \right) H_k \left(I - \frac{Y_k S_k^T}{Y_k^T S_k} \right) + \frac{S_k S_k^T}{Y_k^T S_k} \quad (3)$$

Where, H_{k+1} and H_k are the current and previous approximations of the Hessian inverse matrix, I is the identity matrix, $S_k = W_{k+1} - W_k$, and $Y_k = \nabla E(W_{k+1}) - \nabla E(W_k)$.
The process is performed iteratively and after each update of H_k, a line search is performed in order to identify the appropriate size of the step, α. After this, the new weight vector is determined using the equation below:

$$W_{k+1} = W_k + \alpha d_k \quad (4)$$

This process is performed until a certain stopping criterion is met. For instance, [12] suggest testing the performance of the training process on an additional validation data set which prevents the over-fitting phenomenon from affecting the model's performance on the out-of-sample dataset.

3. Dataset

The analyzed datasets consist of 2129 daily exchange rates for Romanian national currency (RON) against EUR and CHF. These were procured from the official website of the National Bank of Romania and illustrate the values available within the time interval 1st of January 2005 – 28th of February 2013. Missing values resulted from non-transactional days, such as legal holidays, were replaced by those from previous available days.

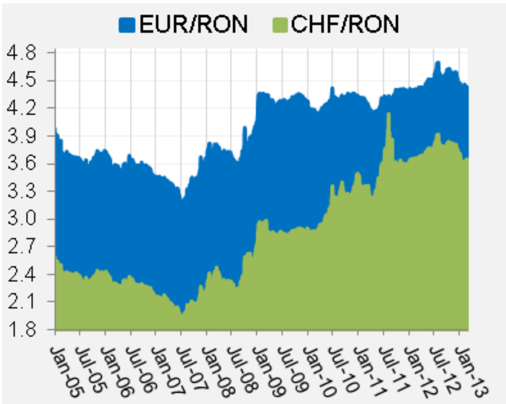

Figure 1. Exchange rates evolution

Figure 1 provides the evolution of the selected exchange rates, EUR/RON and CHF/RON, within the specified time interval, highlighting a nonlinear and volatile character in case of both time series.
Within the timeframe January 2005 – July 2007, the national currency of Romania faced an appreciation trend, offering confidence of a

balanced economy. However, beginning with 2008, the point when the worldwide financial and economic downturn installed, RON experienced significant depreciations. The highest volatility is observed in case of CHF/RON exchange rate (Table 1), which reached a significant amplitude over the analyzed timeframe.

Table 1. Statistics

	EUR/RON	CHF/RON
Min	3.1112	1.8741
Max	4.6481	4.0898
Amplitude	1.5369	2.2157
Average	3.9227	2.7587
Standard deviation	0.4001	0.5876

Glancing at Figure 1, it is visible that both time series have a non-stationary character. Although with ANNs there is usually no prior intensive dataset cleanup required, in order to avoid unbalances resulted from extreme values, the exchange rates were transformed using the first differences of the log time series, reaching eventually stationary time series (see Figure 2):

$$d\_\log\_EUR/RON_t = \ln(EUR/RON_t) - \ln(EUR/RON_{t-1}) \quad (5)$$

Where, EUR/RON_t is the exchange rate at time point t and EUR/RON_{t-1} is the exchange rate from the previous available day.

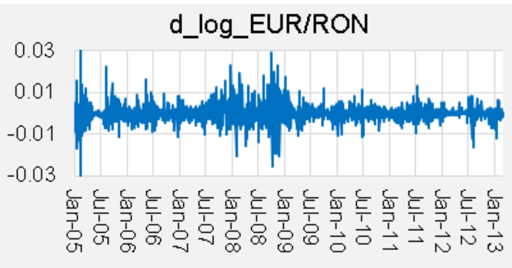

Figure 2 a). d_log_EUR/RON time series

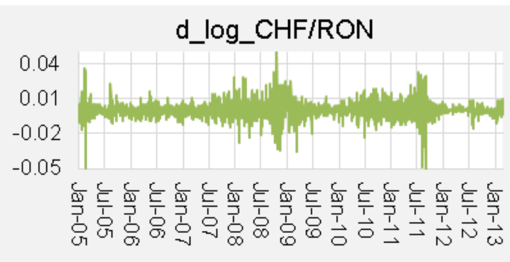

Figure 2 b). d_log_CHF/RON time series

For development purposes, each time series was split as follows:

- Training set: 67%, used for model development,
- Validation set: 20%, used for model assessment,
- Test set: 13%, used for out-of-sample evaluation.

As there is no rule of thumb in selecting the partitioning ratios, a common practice was approached, suggesting to use 2/3^{rds} of the available data into training set. Thus, considering that Figure 3 describes the time line, the first 67% of the exchange rates would fall into training set, the next 20% into validation, and the remaining 13% would enter test data set.

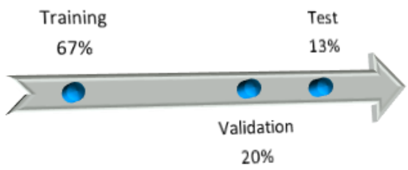

Figure 3. Partitioning scheme

4. Model building

Considering the reviews performed by [10] in respect with exchange rate forecasting applications, short-term forecasting horizons (such as 1 to 3 periods ahead) provide better results compared with those cases when medium or long-term prediction horizons are chosen. This paper aims to predict one step (one day) ahead, based on previous exchange rates, using ANNs with BFGS optimization algorithm. The selected performance measure is the mean squared error metric (MSE) and will be used to select the best model in terms of prediction:

$$MSE = \frac{\sum_{t=1}^{n}(EUR/RON_t - EUR/RON^{estimated}{}_t)^2}{n} \quad (6)$$

Where, EUR/RON_t is the exchange rate at time point *t*, $EUR/RON^{estimated}{}_t$ is the estimated value for time point *t*, and *n* is the number of observations within the dataset.

4.1 Input variable selection using ARIMA(p,d,q)

An Autoregressive Integrated Moving Average (ARIMA) model predicts future values of a time series by a linear combination of its past values and a series of errors (also known as random shocks).

$$\phi_p(B)(1-B)^d Y_t = \mu + \theta_q(B)\varepsilon_t \qquad (7)$$

Where, t indexes time, Y_t is the time series, B is the backshift operator ($BY_t = Y_{t-1}$), p is the order of the autoregressive part, d is the order of integration (differencing), q is the order of moving averaging, μ is a constant, ε_t is a white noise series, $\phi_p(B)$ is the autoregressive operator such that $\phi_p(B) = 1 - \phi_1 B - ... - \phi_p B^p$, and $\theta_q(B)$ is the moving average operator such that $\theta_q(B) = 1 - \theta_1 B - ... - \theta_q B^q$.

This method has been intensely used ever since the 1970's, with good estimation results being also very easy to implement.

The basic steps of *ARIMA(p,d,q)* method stand in:

- Plotting the series Y_t against time line to identify possible seasonality and outliers;
- Making transformations (differencing and identify *d*) on the data series in order to eventually reach stationary data;
- Identify the orders *p,* and *q* using autocorrelation and partial autocorrelation functions;
- Estimate model parameters and test model consistency using different tests.

Using the available time series and choosing a confidence interval of 95%, several values for the orders *p* and *q* were tested up to 5 periods. Eventually, an ARIMA(3,1,0) model without a constant proved significance in case of both currencies (Table 2 a) and b)). Parameters were estimated using the training data sample. Thus, this indicates that the fourth observation is explained by the values observed within the past three periods.

Table 2 a). Parameter estimation for EUR/RON exchange rate

Dependent Variable: D_LOG_EUR_RON
Method: Least Squares
Sample (adjusted): 1/07/2005 6/10/2010
Included observations: 1415 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.221028	0.026455	8.354851	0.0000
AR(2)	-0.138573	0.026846	-5.161750	0.0000
AR(3)	-0.108547	0.026417	-4.108975	0.0000

Table 2 b). Parameter estimation for CHF/RON exchange rate

Dependent Variable: D_LOG_CHF_RON
Method: Least Squares
Sample (adjusted): 1/07/2005 6/10/2010
Included observations: 1415 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.114559	0.026441	4.332577	0.0000
AR(2)	-0.086312	0.026499	-3.257136	0.0012
AR(3)	-0.112513	0.026421	-4.258441	0.0000

4.2 Forecasting with ANNs

Developing a prediction model with Neural Networks requires performing certain configurations for the input nodes, the number of hidden layers and hidden nodes, the activation functions in the hidden and output layers, the training algorithm and the stopping conditions. For this study the multilayered feed-forward neural network was selected using BFGS training algorithm.

The input nodes of the network were assigned based on the ARIMA model that resulted significant in the previous sub-section. Thus, the fourth period will be explained using the values observed within the previous three days. Recent work on the estimation of EUR/RON exchange rate using ANNs ([17]) used a trial and error approach to select the relevant input periods.

Regarding the number of hidden layers, only one was included, as with the appropriate number of hidden nodes results are good in most of the cases ([18]).

On each stationary time series were further trained 100 networks in which the number of hidden nodes was varied between 2 and 30. In the hidden layer, logistic sigmoid (log) and hyperbolic-tangent sigmoid (tanh) functions were used in turns and in the output layer, apart from these two, the linear activation function (lin) was also considered. The stopping condition was set to either of the following events, meaning that whichever happens first, leads to ending the training process: 500 cycles, or a variation of the average error for 20 consecutive epochs below 0.0000001. The best 5 networks in terms of mean squared error on the test data set were retained for each currency (EUR/RON and CHF/RON).

4.3 Results

Results of the selected neural networks were compared with those obtained when using ARIMA technique alone. All five Neural Networks retained provide better results on the out-of-sample data set. Table 3 provides the results for EUR/RON time series, which reached the lowest MSE on the test set when including 19 hidden nodes in the middle layer. Using logistic activation function in the hidden layer and hyperbolic tangent in the output layer, MLP 3-19log-1tanh reaches on the test set a MSE that is by 3% lower compared with ARIMA(3,1,0) results.

Table 3. EUR/RON forecast results

Net. name	Training MSE	Valid MSE	Test MSE
MLP 3-19log-1tanh	3.164E-04	1.423E-04	1.329E-04
MLP 3-19tanh-1log	3.201E-04	1.423E-04	1.334E-04
MLP 3-2log-1tanh	3.314E-04	1.427E-04	1.336E-04
MLP 3-16log-1log	3.138E-04	1.428E-04	1.337E-04
MLP 3-13log-1log	3.268E-04	1.424E-04	1.337E-04
ARIMA(3,1,0)	3.238E-04	1.484E-04	1.366E-04

In case of CHF/RON time series, the best five ANNs using BFGS training algorithm also reach lower MSEs on the test sample compared with ARIMA technique alone (Table 4). This time, the error is only by 1.5% lower for MLP 3-17tanh-1lin using hyperbolic tangent activation function in the hidden layer and linear function in the output layer.

Table 4. CHF/RON forecast results

Net. name	Training MSE	Validation MSE	Test MSE
MLP 3-17tanh-1lin	2.686E-04	1.038E-03	1.266E-04
MLP 3-16tanh-1lin	2.689E-04	1.036E-03	1.271E-04
MLP 3-4log-1lin	2.598E-04	1.001E-03	1.274E-04
MLP 3-14tanh-1lin	2.760E-04	1.044E-03	1.275E-04
MLP 3-2tanh-1log	2.768E-04	1.045E-03	1.285E-04
ARIMA(3,1,0)	2.665E-04	1.044E-03	1.285E-04

5. Conclusion and future developments

Neural Networks are complex machine learning techniques, providing good results in exchange rate forecasting applications. However, the network configuration process can sometimes be time consuming and can eventually lead to senseless results. Selecting the number of input values is one of the tasks that is usually performed in an experimental manner, after employing several trial and error comparisons in which the number of inputs is varied. This, nevertheless, is computationally very expensive as for each new simulation the iterative process must be retaken. To overcome these drawbacks, this paper has shown how to use conventional method ARIMA to select input variables and then build ANN forecasting models which generate good results in terms of forecasting errors.

References

[1] G.E.P. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day, 1970

[2] J. T. Yao, H.-L. Poh, and T. Jasic, Foreign Exchange Rates Forecasting with Neural Networks, *International Conference on Neural Information Processing (ICONIP'96),* Hong Kong, Sept. 24-27, 1996, pp. 754-759

[3] T. Hill, M. O'Connor and W. Remus, Neural Network Models for Time Series Forecasts, *Management Science*, Vol. 42, 1996, pp. 1082-1092

[4] G. Zhang and M.Y. Hu, Neural Network Forecasting of the British Pound/US Dollar Exchange Rate, *International Journal of Management Science*, Vol. 26, 1998, No. 4, pp. 495-506

[5] C.L. Dunis and M. Williams, Modelling and Trading the EUR/USD Exchange Rate: Do Neural Network Models Perform Better?, *Derivatives Use, Trading and Regulation*, Vol. 8, No. 3, 2002, pp. 211–239

[6] J. Kamruzzaman and R.A. Sarker, Comparing ANN Based Models with ARIMA for Prediction of Forex Rates, *ASOR Bulletin*, Vol. 22, 2003, pp. 2-11

[7] C. Panda and V. Narasimhan, Forecasting Exchange Rate Better with Artificial Neural Network, *Journal of Policy Modeling*, Vol. 29, 2007, pp. 227–236

[8] C.L. Dunis, J. Laws and G. Sermpinis, The Robustness of Neural Networks for Modelling and Trading the EUR/USD Exchange Rate at the ECB Fixing, Journal of Derivatives & Hedge Funds, Vol. 15, 2009, pp. 186-205

[9] A.N. Kia, M. Fathian and M.R. Gholamian, Using MLP and RBF Neural Networks to Improve the Prediction of Exchange Rate Time Series with ARIMA, *International Journal of Information and Electronics Engineering*, Vol. 2, No. 4, 2012, pp. 543-546

[10] L. Yu, S. Wang, W. Huang and K.K. Lai, Are Foreign Exchange Rates Predictable? A Survey from Artificial Neural Networks Perspective, *Scientific Inquiry: A Journal of International Institute for General Systems Studies*, Vol. 8, No. 2, 2007, pp. 207-228

[11] G.S. Atsalakis and K.P. Valavanis, Surveying Stock Market Forecasting Techniques – Part II: Soft Computing Methods, *Expert Systems with Applications*, Vol. 36, 2008, pp. 5932-5941

[12] R.C. Schweitzer and J.B. Morris, *A Tutorial on Neural Networks Using the*

ISSN 2067 – 4074

Broyden-Fletcher-Goldfarb-Shanno (BFGS) Training Algorithm and Molecular Descriptors With Application to the Prediction of Dielectric Constants Through the Development of Quantitative Structure Property Relationships (QSPRs), Army Research Laboratory, 2000

[13] C. Broyden, The Convergence of a Class of Double-Rank Minimization Algorithms, *Journal of Inst. Mathematical Applications*, Vol. 6, 1970, pp. 222-231

[14] R. Fletcher, A New Approach to Variable Metric Algorithms, *Computer Journal*, Vol. 13, 1970, pp. 317-322

[15] D. Goldfarb, A Family of Variable-Metric Methods Derived by Variational Means, *Mathematical Computations*, Vol. 24, 1970, pp. 23-26

[16] D. Shanno, Conditioning of Quasi-Newton Methods for Function Minimization, *Mathematical Computations*, Vol. 24, 1970, pp. 647-656

[17] B. Oancea and S.C. Ciucu, Time Series Forecasting Using Neural Networks, Proceedings of *The 7th International Scientific Conference Challenges of the Knowledge Society*, Bucharest, 2013, pp. 1401-1408

[18] S. Knerr, L. Personnaz, G. Dreyfus, Handwritten Digit Recognition by Neural Networks with Single-Layer Training, *IEEE Transactions on Neural Networks*, Vol.3, No. 6, 1992, pp. 962-968