
NEURAL NETWORKS, MACHINE LEARNING

THE EFFECT OF INTRODUCTION OF THE NON-LINEAR CALIBRATION FUNCTION AT THE INPUT OF THE NEURAL NETWORK

Piotr Romanowski

Abstract: *The paper presents the experiment on the time series whose elements are month values of BIS effective exchange rate of USD from January 1994 till March 2010. A tendency of BIS (Bank of International Settlements) effective exchange rate to increase or decrease is an expected value.*

First, a process of building of the neural network for events forecasting is presented, that is the selection of networks' architecture and parameters. Next, the effect of adding data calibrated by nonlinear input function to input data calibrated linearly is described. The nonlinear input function - hyperbolic tangent was accepted. Hyperbolic tangent sigmoid transfer function and log sigmoid transfer function are commonly used as transfer functions in neural networks.

Keywords: *neural network, time series.*

ACM Classification Keywords: *I.2.8 Data calibration.*

Introduction

Forecasting is one of the tools helping in taking decisions. The accuracy of forecasting, that is the agreement of the forecasted value with the real future actual value, is the measure of the correctness of the forecasting. New forecasting methods [Armstrong, 1992], [Kimberly, 2010], that will enable the increase of accuracy are the subject of many works. Artificial neural networks are commonly used tools for forecasting. Typical examples of prediction by the use of neural networks are areas of finance market, meteorology, medicine and many others. Many factors, such as the frequency with which data should be sampled, the number of data points

which should be used in the input representation, the time window size etc. influence the accuracy of prognosis. Moreover, the architecture and parameters of neural network are additional factors when neural networks are used [Gill,1978], [Graves, 2009], [Williams, 1989]. Figure 1 [Smith, 2003] presents typical network architecture.

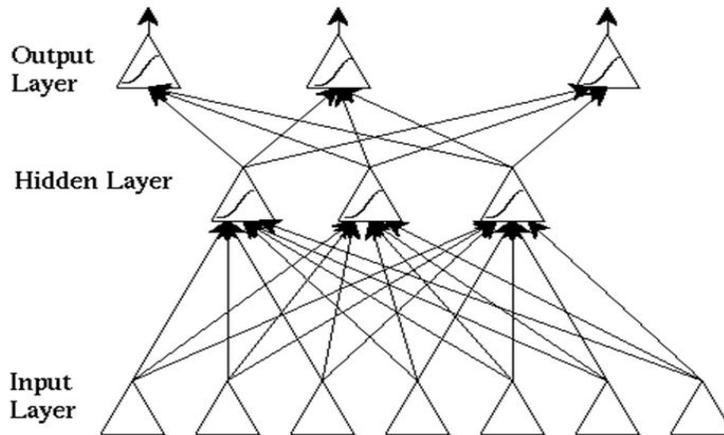


Fig. 1. Typical network architecture.

The process of neural network teaching is aimed at error minimization that is the minimization of differences between output signals and values of the training sample. During this process the optimal set of weights of individual neurons is searched. It is analogous to the process of minimizing of functional in a numerical solution of set of differential equations. The selection of architecture and parameters of the neural network and data preparation are similar to functional minimization, but more complicated.

Time series prediction

Time series are vectors, $x(t)$, $t=0,1,\dots,n$, where t represents consecutive time [Chen, 2002], [Oppenheim, 1999]. Work of neural networks above time series can be described as finding a function $f: R^N \rightarrow R$ such as to obtain an estimate of a x time $t+d$, from the N time steps back from time t , so that:

$$x(t+d) = f(x(t), x(t-1), \dots, x(t-N+1)) \quad (1)$$

Experiment

Data

A subject of experiments was a set of average BIS effective exchange rate of USD from January 1994 till March 2010. It is worth mentioning that exchange rates are regarded as very difficult to predict [Stein, 1994]. In the paper, the whole set was named USD rate, its elements USD rate[i], $i=1, \dots, 192$ (the length of USD rate =192).

Usually, data calibration to an interval from 0 to 1 is the first step in data preparation. The following equations are used:

$$\text{USDcalibrated} = (\text{USD rate} - \min(\text{USDrate}))/\text{range};$$

Where $\text{range} = \max(\text{USDrate}) - \min(\text{USDrate})$

Or for value from -1 to 1,

$$\text{USDcalibrated} = 2 * (\text{USDrate} - \min(\text{USDrate} - \text{range}/2)) / \text{range} \text{ (used in this work)}$$

Figure 2. presents USD rate after calibration to the interval [-1; +1].

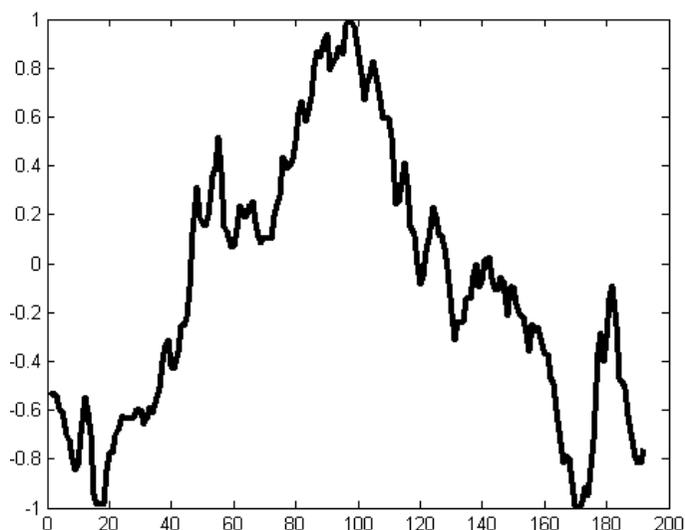


Fig. 2. USD rate after calibration.

The set of input elements was created according to equation (1). Accepting time window length $N=15$, 177 input vectors of N elements were obtained. A one layer input network (neuron number = N) and a one neuron output layer were accepted. Input layer neurons emit the output signal modified by a Log-sigmoid transfer function

$\text{logsig}(n) = 1/(1+\exp(-n))$ [Matlab 7.1]. Output layer neurons emit a linear $\text{purelin}(n) = n$ signal. Output functions are presented in Figure 3.

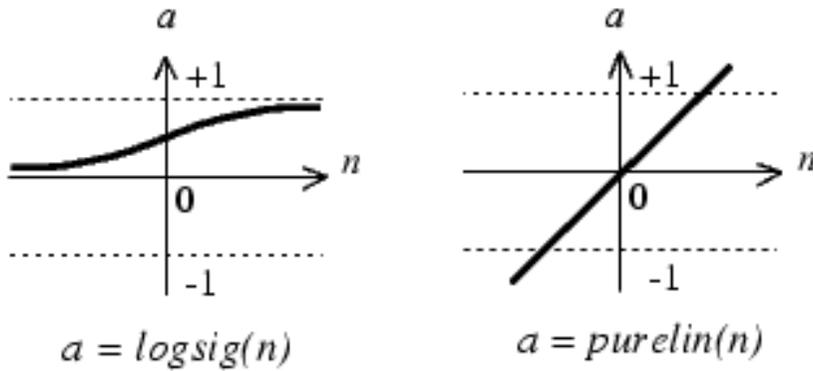


Fig. 3. Output functions.

A `trainlm` training function that updates weight and bias values according to Levenberg – Matquard optimization was used for network training [Lampton, 1997], [Marquardt, 1963]. `Trainlm` is often the fastest backpropagation algorithm in the MATLAB toolbox, and is highly recommended as a first choice supervised algorithm, although it does require more memory than other algorithms [Williams, 1989]. The network was trained and tested at the same set of vectors.

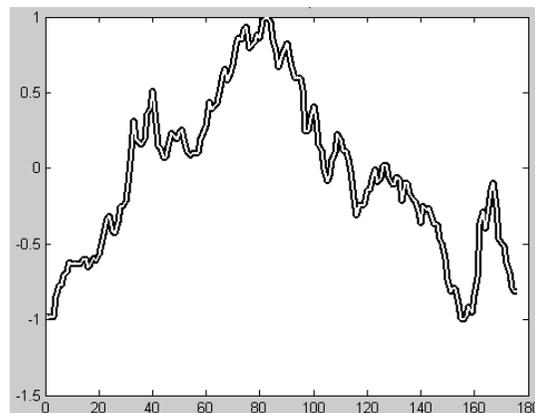


Fig.4. Output of the network can be seen as a thin, white line on the bold, black line which represents the measured data.

The following parameters were accepted:

The number of epochs = 100

Train Parameter goal = $1e-12$ (error coefficient)

Size of time window = 15.

The network was trained well, values forecasted by the network agree with measured data (see Figure 4).

Mean absolute error of approximation was $4.56608e-005$

Standard deviation of the approximation error was 0.000172702

Then, the set of input vectors was divided into a set of 160 training vectors and a set of 17 test vectors. After training, the results of prediction on the test set are not satisfactory (see Figure 5).

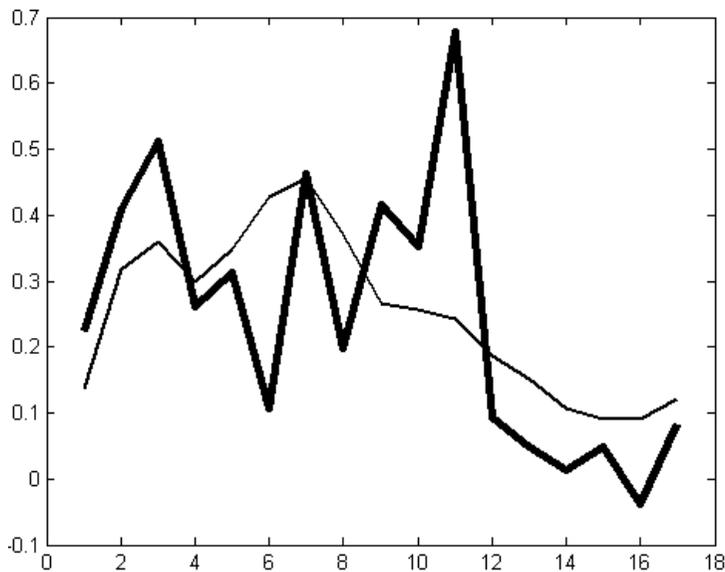


Fig. 5. The comparison of the real (thin line) and predicted (bold line) values of BIS effective exchange rate of USD.

Calculations were carried on for 100 epochs. Mean absolute approximation error was 52.8957 and standard deviation of the approximation error was 81.9857 .

The proportion of the number of correct answers of the network to the number of test vectors was accepted as an efficiency of the network. The correct answer is the answer that has the same tendency as the test data value of USD exchange rate (1 - increase, 0 - else).

Next the set of 160 training vectors was divided into a set of 143 training vectors and 17 test vectors. It was assumed, that these 17 vectors are unknown.

Below, a short description of data preparation and selection of architecture and network parameters are presented.

Data preparation

The aim of the data preparation was establishing the optimal size of the time window and the length of the training vector establishing. 10 tests for each size of the time window from 1 to 20 were carried out, the obtained mean values are presented in Figure 6. Than a size of the time window equal 4 was accepted.

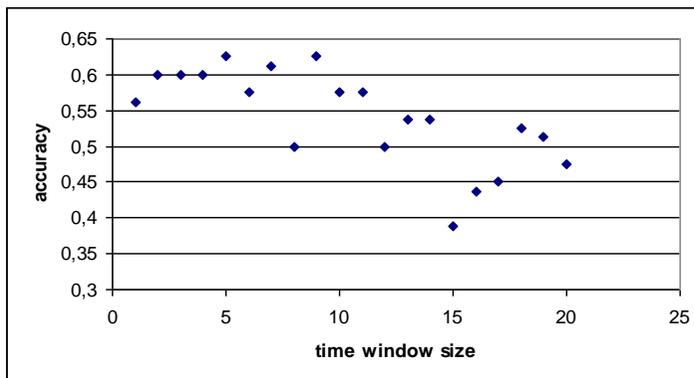


Fig.6. The accuracy of the network as a function of the time window size.

Next, the influence of the length of the training vector on the efficiency of the network was tested. Results are presented in Figure 7. For the further calculations the maximal length of the training vector equal 154 was accepted.

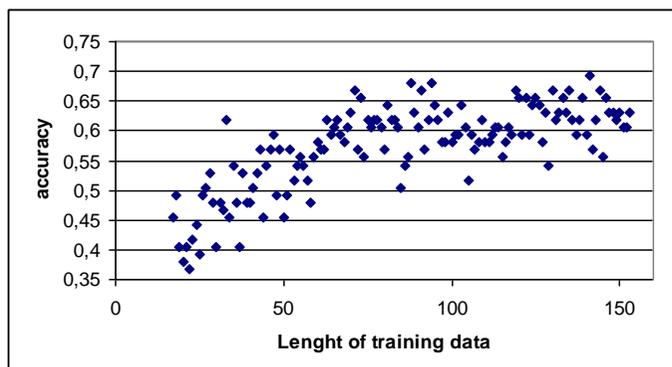


Fig.7. The accuracy of the network as a function of the length of the training vector.

In the paper the linear data calibration was accepted for choosing network's parameters and architecture. Data were projected into the [-1; +1] range, that means :

DATA calibrated = $2 * (\text{DATA} - \min(\text{DATA}) - \text{range} / 2) / \text{range}$.

Selection of architecture and network parameters.

Determination of the optimal number of training epochs is presented in Figure 8. Finally 50 training epochs were accepted.

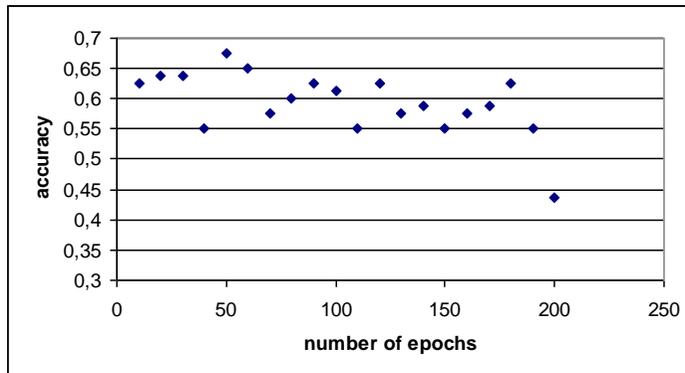


Fig.8. The influence of the number of the training epochs on the accuracy. Each point represents the mean value of 10 calculations

Consecutively, the results of the network extension of one hidden layer were investigated. Several kinds of layers with different number of neurons and different output functions were investigated, but as a mean accuracy always decreased, the idea of network extension was rejected. Figure 9 presents the result one of some calculations. Each point represents the mean value of 10 calculations.

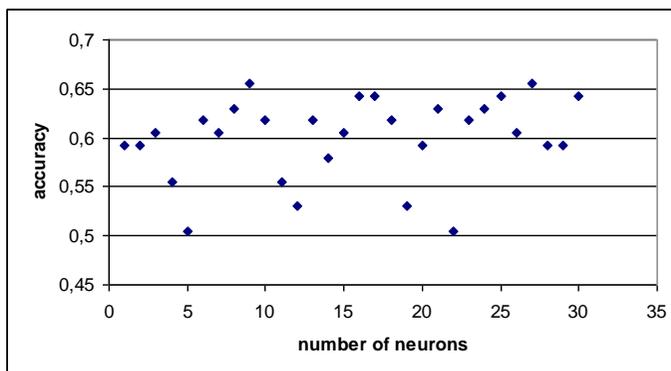


Fig. 9. The effect of introduction of the hidden layer with the changeable number of neurons.

Finally the log-sigmoid function was accepted for input layer and purelin for the output.

Neural network

Finally, the network with time size window = 4 (the number of neurons equals the length of the input vectors), with log-sigmoid transfer function and output layer with one neuron that emits the linear signal was accepted. The number of the training epochs equal 50 was accepted.

100 calculations for the accepted network were performed. The mean efficiency of the network was 0.6220. Figure 10. presents one of the plots that compare the network answers with the real values of the similar accuracy.

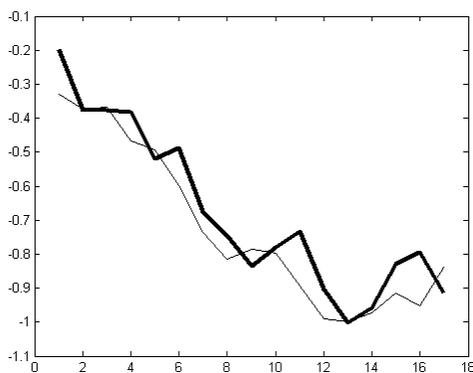


Fig. 10. The comparison of the real values of the BIS effective exchange rate of USD (the thin line) with the values calculated by the network (bold line)

Introduction of a non-linear calibration function

The input vectors included 4 consecutive mean month values of x =BIS effective exchange rate of USD calibrated to the interval $[-1,+1]$. The values calibrated by the non-linear function $x' = \tanh(\beta \cdot x)$ were added to them (see Figure 11).

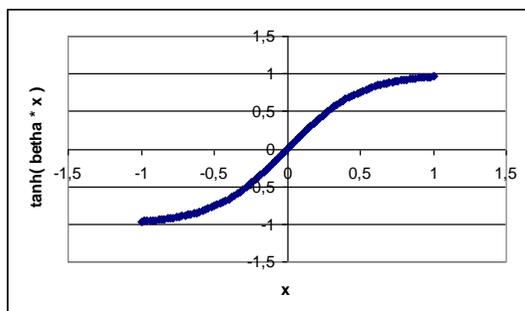


Fig. 11. Non-linear calibration function $x' = \tanh(\beta \cdot x)$, $\beta = 2.0$.

The input vector includes now eight elements, instead of four. The input layer consists of eight neurons. Figure 12. presents results of such extension. Ten runs were performed for each value of beta coefficient from 0.5 to 3.5 (step 0.1). For each beta value of the accuracy increased.

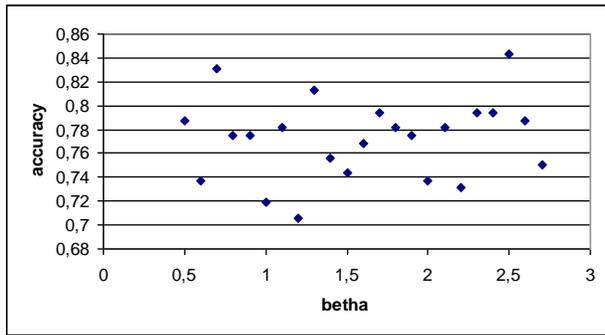


Fig. 12. The network accuracy after introduction of non-linear calibrated data.

Beta coefficient equal 2.0 was accepted.

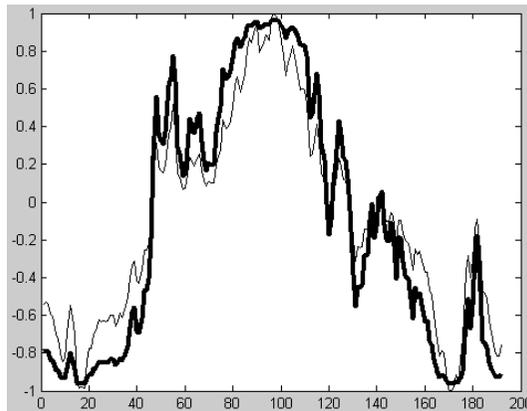


Fig. 13. Data calibrated lineary - thin line and data calibrated by the use of non-linear function $x' = \tanh(\text{beta} \cdot x)$ function - bold line.

After 100 runs, the mean accuracy was 0.6740. This result is 5 per cent better that the accuracy for linear calibration (0.6220).

Two methods of calibration (linear and non- linear) were tested on 17 test vectors. After 100 runs, the mean accuracy was 0.8247 for linear calibration and 0.8667 for adding non-linear calibration. So, adding non-linear calibration increased the accuracy of 4.2 per cent.

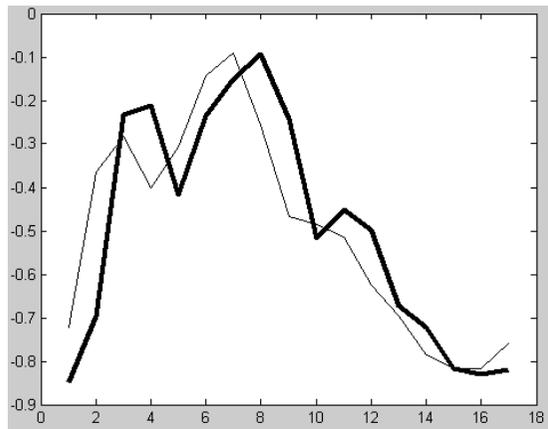


Fig. 14. Comparison of the network answer (bold line) with the real values (thin line) after one of 100 runs when the non – linear input function was used.

The test on the new set of input vectors gave higher accuracy. Probably, it is not due to the increase of the length of the training vector (see Figure 7.) but to the appearance of better recognized data.

Conclusions

The analyzed data are regarded as difficult to predict. On the other hand, there is a strong need of creating methods for forecasting the currency market. The significant difference of accuracy for different test vectors (65% and 85%) justify the statement, that one must be very careful using neural networks at the currency market.

The increase of the input vectors calibrated linearly with the values calibrated non – linearly caused the increase of accuracy by 4.6 per cent.

It seems, that results presented in this paper encourage further experiments and calculations.

Bibliography

- [Armstrong , 1992] J.S.Armstrong, F.Collopy. Error Measures For Generalizing About Forecasting Methods: Empirical Comparisons. In: International Journal of Forecasting 8: 69–80, 1992.
- [Chen, 2002] Y.Chen, G.Dong, J.Han, B.W.Wah, J.Wang. Multidimensional regression analysis of time-series data streams. In: Proc. 2002 Int. Conf. Very Large Data Bases (VLDB'02), 323-334, 2002.
- [Gill,1978] P.E.Gill, W.Murray. Algorithms for the solution of the nonlinear least-squares problem. In: SIAM Journal on Numerical Analysis 15 (5): 977–992, 1978.
- [Graves, 2009] A.Graves, J.Schmidhuber. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. Advances in Neural Information Processing Systems 22 In: NIPS'22, p 545-552, Vancouver, MIT Press, 2009.

- [Kimberly, 2010] E.Kimberly. Production Planning and Inventory Control. In: McGraw-Hill. ISBN 0-412-03471-9, 2010.
- [Lampton, 1997] M.Lampton. Damping-Undamping Strategies for the Levenberg-Marquardt Nonlinear Least-Squares Method. In: Computers in Physics Journal, 11(1):110–115, Jan./Feb, 1997.
- [Marquardt, 1963] D.W.Marquardt. An algorithm for least-squares estimation of nonlinear parameters. In: Journal of the Society for Industrial and Applied Mathematics, 11:431–441, 1963.
- [Oppenheim, 1999] A.V.Oppenheim, R.W.Schafer, J.A.Buck. Discrete-time signal processing. In: Upper Saddle River, N.J. Prentice Hall. pp. 468–471, 1999.
- [Smith, 2003] L.Smith. An Introduction to Neural Networks, <http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html>. Centre for Cognitive and Computational Neuroscience, Department of Computing and Mathematics, University of Stirling. 2003.
- [Stein, 1994] J.L.Stein. The Evolution of the Real Value of the U.S. Dollar Relative to the G7 Currencies. In: Ronald MacDonald. 1994.
- [Williams, 1989] R.J.Williams. Complexity of exact gradient computation algorithms for recurrent neural networks. In: Technical Report Technical Report NU-CCS-89-27, Boston: Northeastern University, College of Computer Science, 1989.
- [MATLAB 7.1] MATLAB Version 7.1 (R14SP3).

Authors' Information



Piotr Romanowski - Rzeszow University, Faculty of Mathematics and Natural Sciences / The Institute of Computer Science. Al. Rejtana 16A, 35-310 Rzeszów, Poland; e-mail: proman@univ.rzeszow.pl

Major Fields of Scientific Research: numerical methods, time series, artificial Intelligence