Neural Network Prediction of Short-Term Dynamics of Futures on Deutsche Mark, Libor and S&P500

Ludmila Dmitrieva[1], Yuri Kuperin[1,2] and Irina Soroka[3]

[1] Department of Physics, Saint-Petersburg State University,
Ulyanovskaya str. 1, 198094 Saint-Petersburg, Russia
`kuperin@JK1454.spb.edu`
[2] School of Management, Saint-Petersburg State University
per.Dekabristov 16, 199155 Saint-Petersburg, Russia
[3] Baltic Financial Agency,
Nevsky pr. 140, 198000 Saint-Petersburg, Russia
`phosphor@ok.ru`

Abstract. The talk reports neural network modelling and its application to the prediction of short-term financial dynamics in three sectors of financial market: currency, monetary and capital. The methods of nonlinear dynamics, multifractal analysis and wavelets have been used for preprocessing of data in order to optimise the learning procedure and architecture of the neural network. The results presented here show that in all sectors of market mentioned above the useful prediction can be made for out-of-sample data. This is confirmed by statistical estimations of the prediction quality.

1 Introduction

In this talk we consider dynamic processes in three sectors of the international financial markets - currency, monetary and capital. Novelty of an approach consists in the analysis of financial dynamics by neural network methods in a combination with the approaches advanced in econophysics [1]. The neural network approach to the analysis and forecasting of the financial time series used in the present talk is based on a paradigm of the complex systems theory and its applicability to the analysis of financial markets [2,3]. The approach we used is original and differs from approaches of other authors [4-7] in the following aspects. While choosing the architecture of a network and a stratagy of forecasting we carried out deep data preprocessing on the basis of methods of complex systems theory: fractal and multifractal analysis, wavelet-analysis, methods of nonlinear and chaotic dynamics [1,2,3]. In the present talk we do not describe stages and methods of this data preprocessing. However the preliminary analysis has allowed to optimize parameters of the neural network, to determine horizon of predictability and to carry out comparison of forecasting quality of different time series from various sectors of the financial market.

Specifically we studied dynamic processes in the currency, monetary, capital markets in the short-term periods, predicting daily changes of prices of the following financial assets: futures on a rate US dollar - DM (it is designated as DM), futures on the rate of interest LIBOR on eurodollars (ED), futures on American stock index Standard & Poor's 500 (SP).

2   Method of Prediction and Network Architecture

It should be noted that the success or failure of a neural network predictor depends strongly on the user definition of the architecture of the input and desired output. For prediction of data sets under consideration the neural network we had used two inputs. As the first input we used the daily returns expressed as follows: to 1,5% return there corresponded the value 1.5. As the second input we used the mean values for the last 5 days. The presence of noise in analyzed time series can degenerate the learning and generalisation ability of networks. It means that some smoothing of time series data is required. We used the simplest techniques for smoothing, i.e. 5-days moving average shifted backwards for one day. Thus such neural network aims to predict smoothed daily return to next day. Among all possible configurations of neural nets we chose the recurrent network with hidden layer feedback into input layer known as the Elman-Jordan Network (see Fig.1).
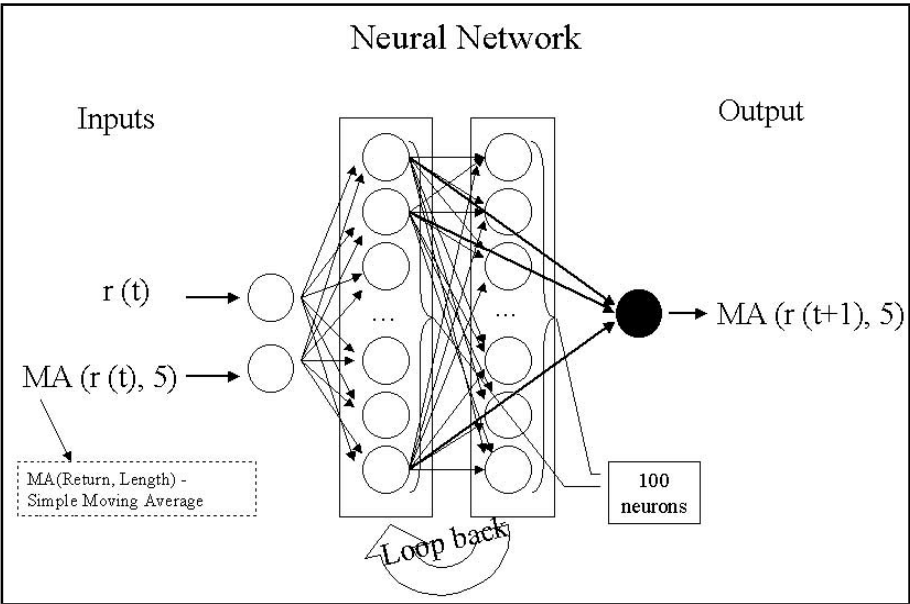

Fig. 1. Architecture of the Elman-Jordan neural network used for prediction

To our opinion this is one of the most powerful recurrent network. This type of backpropagation network has been successfully used in predicting financial markets because recurrent networks can learn sequences, so they are powerful tools for time series data processing. They have the slight disadvantage of taking longer to train. A backpropagation network with standard connections responds to a given input pattern with exactly the same output pattern every time the input pattern is presented. A recurrent network may respond to the same input pattern differently at different times, depending upon the patterns that have been presented as inputs just previously. Thus, the sequence of the patterns is as important as the input pattern itself. Recurrent networks are trained in the same manner as standard backpropagation networks except that patterns must always be presented in the same order; random selection is not allowed. The one difference in structure is that there is one extra slab in the input layer that is connected to the hidden layer just like the other input slab. This extra slab holds the contents of one of the layers as it existed when the previous pattern was trained. In this way the network sees previous knowledge it had about previous inputs. This extra slab is sometimes called the network's "long term" memory.

The Elman-Jordan network has logistic $f(x)=1/(1+\exp(-x))$ activation function of neurons in its hidden (recurrent) layer, and linear activation function of neurons in its output layer. This combination is special in that two-layer networks with these activation functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. In the present research we changed the logistic activation function by the symmetric logistic function $f(x)=(2/(1+\exp(-x)))-1$. This do not change the properties of the network in principle but allows to speed up the convergence of the algorithm for the given types of series. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fitted increases in complexity. In the network we used the hidden layer consisted of 100 neurons.

One of the hard problems in building successful neural networks is knowing when to stop training. If one trains too little the network will not learn the patterns. If one trains too much, the network will learn the noise or memorise the training patterns and not generalise well with new patterns. We used calibration to optimise the network by applying the current network to an independent test set during training. Calibration finds the optimum network for the data in the test set which means that the network is able to generalise well and give good results on out-of-sample data.

3   Neural Network Prediction of Returns

We divided each analysed time series into 3 subsets: training set, test set and production set. As the training set, i.e. the set on which the network was trained to give the correct predictions, we took the first 900 observations. The test set was used for preventing the overfitting of the network and for calibration and included the observations numbering from 901 up to 1100. The production set included observations, which "were not shown" to the network and started from the 1101-th observation up to the end of the series.

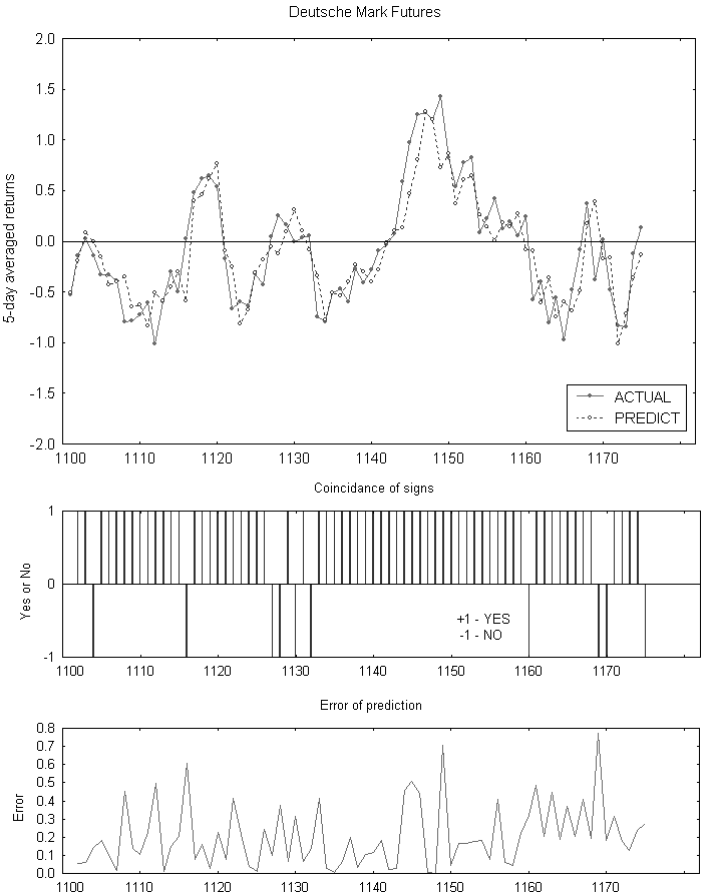The results of predictions for training sets of all analysed financial assets are given in Fig1, Fig2, Fig.3.

Fig. 2. Results of neural network prediction of 5-days moving average of returns for Deutsche mark futures: actual versus predicted returns in percents *(top figure)*, coincidence of signs of predicted and actual returns *(middle figure)*, absolute value of the actual minus predicted returns in percents (*bottom figure)*
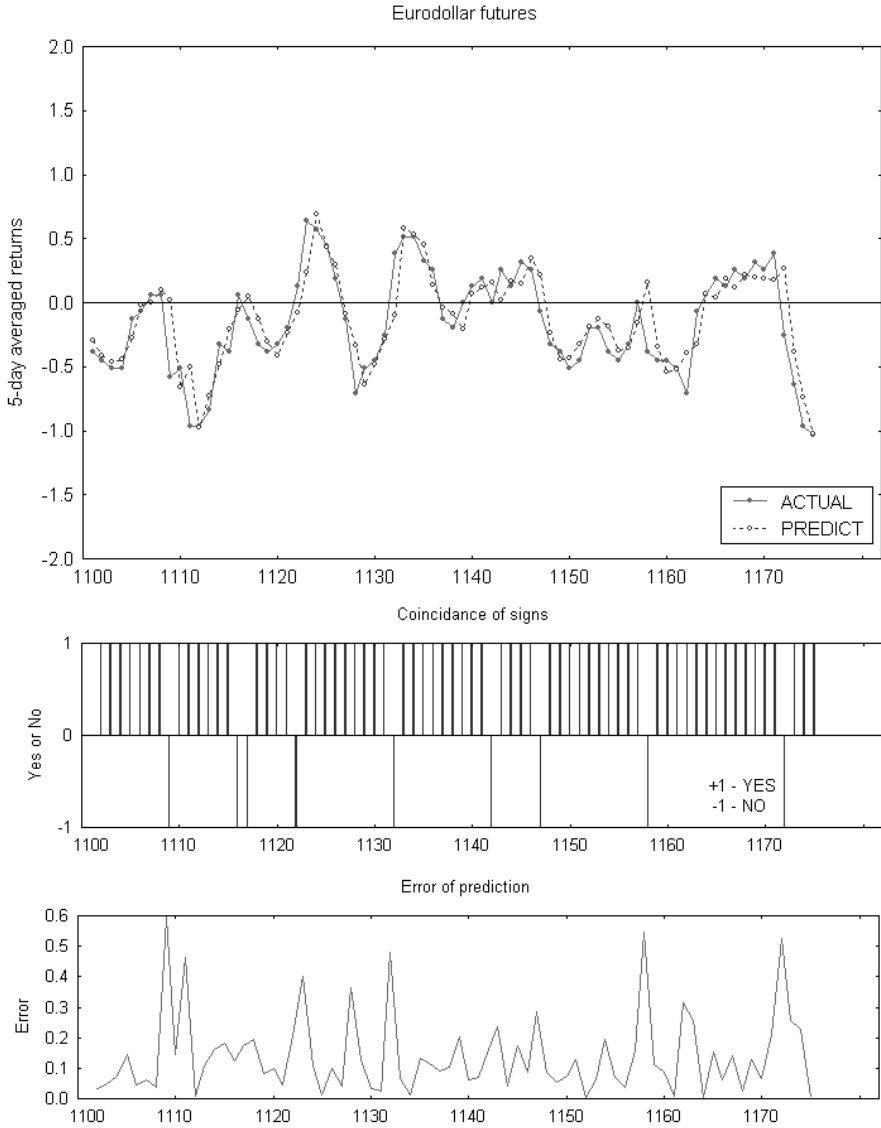
Fig. 3. Results of neural network prediction of 5-days moving average of returns for Eurodollar futures: actual versus predicted returns in percents *(top figure)*, coincidence of signs of predicted and actual returns *(middle figure)*, absolute value of the actual minus predicted returns in percents *(bottom figure)*
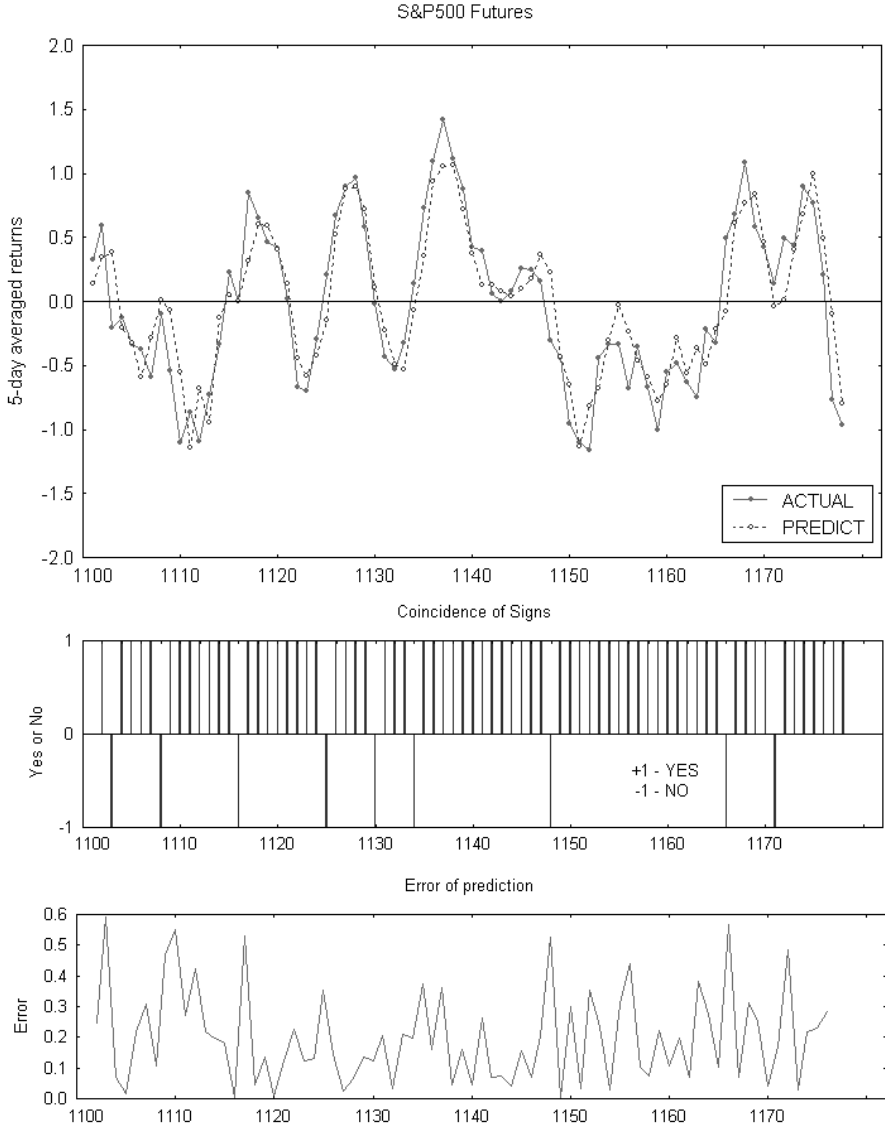
Fig. 4. Results of neural network prediction of 5-days moving average of returns for S&P500 futures: actual versus predicted returns in percents *(top figure)*, coincidence of signs of predicted and actual returns *(middle figure)*, absolute value of the actual minus predicted returns in percents (*bottom figure*)

The quality of prediction was estimated by the following parameters:

- *Training time and number of learning epochs* - the quantities showing how long the network can improve its predictions to achieve the best results on the test set. By the learning epoch we mean the single presentation to the network of all samples from training set. These parameters can vary depending on the given learning rate and momentum. Leaning rate and momentum are established on the basis of desirable accuracy of the prediction. For the neural network we used both these parameters were equal to 0.003.
- *Coefficient Q* compares the accuracy of the model to the accuracy of a trivial benchmark model or trivial predictor wherein the prediction is just the mean of all of the samples. A perfect fit would result in an Q value of 1, a very good fit near 1, and a very poor fit less than 0. If neural model predictions are worse than one could predict by just using the mean of sample case outputs, the coefficient Q value will be less than 0.
- *R-squared* - the coefficient of determination which is a statistical indicator usually applied to regression analysis being the ratio of predicted values variation to the actual values variation.
- *Mean absolute error* - this is the mean over all patterns of the absolute value of the actual minus predicted values.
- *Max absolute error* - this is the maximum of actual values - predicted values of all patterns.
- *% of proper predictions of returns signs* – this is the ratio of number of samples for which signs of predicted values coincide with signs of actual ones to the number of considered samples.

For details the reader is referred to the linear statistical analysis literature [8].

The above characteristics of neural network prediction quality for the analysed series are given in table 1. The table consists of three blocks. The upper one gives characteristics of the network training. The middle one refers to the whole time series, which includes training, test and production sets. The bottom block describes only the results for production sets.

The table shows that the best predictions are obtained for S&P500 futures, the worse predictions are obtained for the Eurodollar (ED) futures. Deutsche mark (DM) futures has intermediate predictions. This follows from values of coefficient Q for production set (see the bottom block of table 1) although it hardly can be seen by sight from Fig.3 and Fig.4. It should be noted that despite the approximately equal quality of learning (see values of coefficient Q in the middle block of table 1) the training time for S&P500 is five times bigger than that for DM and training time for DM futures is four times bigger than training time for ED. This obviously means that to find hidden regularities in S&P500 futures is noticeably complicated than in DM futures and all the more in ED futures. At the same time the best quality of prediction is obtained just for S&P500 and the worse for ED. All this points out that hidden regularities found by neural network in S&P500 preserve their character much more longer than that found in ED. In other words ED futures have more unsteady hidden regularities what results in the worse quality of predictions.

Table 1. Numerical characteristics of neural network predictions quality (N denotes the number of learning epochs, τ stands for training time, N_{whole} denotes the number of samples in the whole time series while N_{prod} stands for the number of samples in production set, % of signs means the percent of proper predictions of returns signs)

Characteristics	S&P500 futures	DM futures	ED futures
N	30512	6779	1873
τ (hours)	19	4	1
N_{whole}	1173	1170	1170
Q	0,7408	0,7594	0,7436
r–squared	0,7431	0,7612	0,7452
mean abs.er., %	0,182	0,196	0,179
max abs.er., %	2,172	1,291	2,281
% of signs	86	83	83
N_{prod}	77	74	74
Q	0,8032	0,5897	0,4517
r–squared	0,8062	0,6319	0,5697
mean error, %	0.217	0,279	0,201
max.error, %	0.799	1,046	1,234
% of signs	88	86	88

In summary one should mention that ultimate goal of any financial forecasting is profitability. The latter is always connected with some trading rule and/or the money management strategy. This problem is out the scope of the present talk (see, however our recent paper [9]).

References

1. Mantegna, R.N., Stenley, H.E:. An Introduction to Econophysics. Correlations and Complexity in Finance. Cambridge University Press (2000)
2. LeBaron, B.: Chaos and Nonlinear Forecastability in Economics and Finance. Philosophical Transactions of the Royal Society of London 348 (1994) 397-404
3. Peters E.E.: Chaos and Order in Capital Market. John Wiley&Sons (1996)
4. Baestaens, D.E., Den Bergh, W.-M.Van, Wood, D: Neural network solutions for trading in financial markets. Pitman Publishing (1994)
5. Refenes, A.-P. (ed.): Neural Networks in the Capital Markets. John Wiley&Sons (1995)
6. Poddig, Th.: Developing Forecasting Models for Integrated Financial Markets using Artificial Neural Networks. Neural Network World 1 (1998) 65 – 80
7. Poddig, Th., Rehkugler, H. A.: World Model of Integrated Financial Markets using Artificial Neural Networks. Journal of Neurocomputing 10 (1996) 251-273
8. Dougherty, Ch.: Introduction to Econometrics. Oxford University Press (1992)
9. Kuperin, Yu.A., Dmitrieva L.A. and Soroka, I.V.: Neural Networks in Financial Market Dynamics Studies. Working paper series № 2001-12, Center for Management and Institutional Studies, St.Petersburg State University, St.Petersburg (2001) 1-22