FINANCIAL DATA FORECASTING USING RECURRENT NEURAL NETWORKS

Cătălina-Lucia COCIANU

Bucharest University of Economic Studies

catalina.cocianu@ie.ase.ro

Mihai-Șerban AVRAMESCU

Bucharest University of Economic Studies

mihai_serban.avramescu@yahoo.com

Abstract. *The work reported in the paper focuses on financial data forecasting in terms of one-step-ahead nonlinear auto-regressive model (NAR). The main aim is to obtain a suitable method to forecast the exchange rate between EURO and US Dollar. In our work, the prediction task is carried out by three recurrent neural networks, namely the standard NAR neural network, Long Short Time Memory (LSTM) and stacked LSTM respectively. We evaluated the performances of the obtained algorithms using the mean absolute percentage error, the root mean square error and a trend-based measure. The performance of the proposed methods is evaluated by a long series of tests, the results being very encouraging as compared to similar developments. The test results enable us to conclude that the LSTM neural network outperforms the standard NAR one. A series of experimentally derived conclusions and some suggestions for further work are exposed in the final part of the paper.*

Keywords: LSTM Networks, NAR Model, NAR Networks, statistical metrics

JEL classification: C45, C63, CO2

DOI: 10.12948/ie2019.01.03

1. Introduction

Modern financial markets are chaotic and very hard to predict systems mainly due to their volatile features and non-linearity. Therefore, stock market prediction is regarded as one of the most challenging applications among time series forecasting and it is under continuous investigations for years. Nowadays, the field of financial forecasting is developing fast, the financial data being the main valuable source of information for the investors and traders.

One of the most successful techniques in solving financial data prediction problems belong to machine learning (ML) field and include Artificial Neural Networks (ANN). More recently, a series of research works suggested that deep nonlinear topology should be used in time series forecasting area. ([1], [2]) An improvement over traditional machine learning, deep learning (DL) models as for instance convolutional neural networks, deep belief networks and stacked auto-encoders can successfully model complex real-world data and achieve better performances than traditional ML approaches. (see: [3], [4])

Recurrent Neural Networks (RNNs) belong to the ANN class and exhibit temporal dynamic behavior. One of the main properties of RNNs is the capacity of using their own internal memories to operate over sequences of vectors (input data, outputs or both). Note that the output vector's contents are influenced not only by the current input sequence, but also depend on the entire history of past inputs. Long Short Time Memory (LSTM) is a particular type of RNN with feedback links attached to some of its layers. LSTM is a suitable learning

environment in case of non-stationary time series and it has the ability to predict data when there are arbitrary sized time steps. (see: [5], [6], [7], [8])

The rest of the paper is organized as follows. Next section is dedicated to NAR model. A brief description of LSTM and stacked LSTM is provided in the third section of the paper. In the final part of the paper are exposed experimental results, comparative analysis, conclusions and suggestions for future work.

2. The NAR Model

The NAR model (Nonlinear Auto Regressive) is a type of RNN well suited for modelling non-linear time series as well as for one-step and multi-step ahead prediction respectively. The forecasting model in case of non-stationary time series usually uses Partial Autocorrelation Function (PACF) to establish the delay variable.

In the following we denote by:

- T - the total number of time periods
- Y - the variable to be forecast
- d_Y - the corresponding delays.

The delay variable of a time series X corresponds to the lag d , where the PACF function computed for X drops immediately after the d^{th} lag. For $1 \leq t \leq T$, Y_t is the value at the moment of time t , and we denote by $\hat{Y}_t$ the predicted value of Y_t . The NAR model is given by [9].

$$\hat{Y}_{(t+p)} = f\left(Y_t^{(d_Y)}\right) \quad (1)$$

where

- f is a non-linear function
- $Y_t^{(d_Y)} = \{Y_t, Y_{t-1}, Y_{t-2}, \dots, Y_{t-d_Y+1}\}$

In the following we consider $p=1$, the equation (1) describing the one-step ahead prediction model. The non-linear function f can be computed by a neural network, the most commonly choice being the NAR networks.

3. LSTM Neural Network

RNN is a neural networks category specialized in processing sequential type data, as for instance time series. The most important properties of the RNNs are:

- RNN have an output value for every time step;
- RNN have recurrent links at the neurons from the hidden layer(s);
- RNN have the capacity to read an entire sequence of data and return one single output value.

Among the main algorithms that can be implemented using this neural network model are: Real-Time Recurrent Learning (RTRL) and Long Short-Term Memory (LSTM). [10]

If we denote by $(v_1, \dots, v_T)$ an input sequence, RNN computes a sequence of hidden states and an output vector z using the following algorithm:

for t from 1 to T do

$$u_t \leftarrow W_{hv}v_t + W_{hh}h_{t-1} + b_h$$

$$h_t \leftarrow e(u_t)$$

$$o_t \leftarrow W_{oh}h_t + b_o$$

$$z_t \leftarrow g(o_t)$$

end for

Note that in the above algorithm the matrices W_{hv} , W_{hh} , and W_{oh} correspond to the network weights values and the vectors b_h , b_o , h_0 correspond to the network biases values. RNN determines a hidden state sequence $h_{1,T}$ and an outputs sequence $z_{1,T}$. $e(\cdot)$ and $g(\cdot)$ are the non-linear components of the hidden neural, respectively, of the output neural layer and h_0 is the vector containing the parameters in which is saved the initial state of the hidden layer. [11]

Long Short Term Memory is a type of RNN architecture that allows the long-time learning of time steps dependencies. The learning process is accomplished using specific memory blocks located in the recurrent hidden layer. A memory cell consists of four units: an input gate, a forget gate, an output gate, and a self-recurrent neuron. [12]

Let us denote by $x = (x_1, \dots, x_T)$ an input vector and we denote by $y = (y_1, \dots, y_T)$ the output sequence. The mathematics behind the most common model of the LSTM architecture is described as follows [12].

The function used to remove a part from cell state information is described as

$$f_t = \sigma(W_f \circ [y_{t-1}, x_t] + b_f) \quad (2)$$

where W_f is the weight matrix, b_f is the bias vector.

The vector containing the new possible values is defined using a tanh layer

$$i_t = \sigma(W_i \circ [y_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c \circ [y_{t-1}, x_t] + b_c) \quad (4)$$

where W_i and W_c are the weight matrices, b_i and b_c are the biases, and y_{t-1} is the value of the memory cell at time t .

Next we compute c_t , an update to the state of the memory cell c_{t-1} as follows

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (5)$$

and the output is given by

$$y_t = o_t \cdot \tanh(c_t) \quad (6)$$

$$o_t = \sigma(W_o \circ [y_{t-1}, x_t] + b_o) \quad (7)$$

where W_o is the weight matrix, b_o is the bias vector, and o_t is the value of the output gate at the time t .

Note that, in order to make LSTM deeper, additional hidden layers can be considered. The obtained RNN is called LSTM stacked. The idea behind the use of multi hidden layers is to recombine already learned representation (the information obtained from prior layers) at higher levels of abstraction of input samples at various time scales. It was only experimentally established that stacked LSTMs are sometimes more useful than shallower ones. (see: [13], [14])

4. Experimental Results

We have implemented standard NAR, LSTM and stacked LSTM neural networks to forecast the exchange rate between EURO and US Dollar, recorded between October 1998 and October 2018. We evaluated the performances of the obtained algorithms using the mean absolute percentage error, the root mean square error and a trend-based measure. We denote by $\{Y(1), \dots, Y(T)\}$ the set of correct values of Y and we assume that $\{\hat{Y}(1), \dots, \hat{Y}(T)\}$ is the collection of the forecasted values. The mean absolute percentage error (MAPE) is defined by

$$MAPE = 100 \times \frac{1}{T} \sum_{i=1}^T \frac{|Y(i) - \hat{Y}(i)|}{Y(i)} \quad (8)$$

The root mean square error (RMSE) is expressed as

$$RMSE = \sqrt{\frac{1}{T} \sum_{i=1}^T (Y(i) - \hat{Y}(i))^2} \quad (9)$$

while the prediction of change in direction (POCID) is given by

$$POCID = 100 \times \frac{\sum_{i=1}^{T-1} D_i}{T} \quad (10)$$

where

$$D_i = \begin{cases} 1, & \text{if } (Y(i+1) - Y(i))(\hat{Y}(i+1) - \hat{Y}(i)) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

The PACF analysis of the EURO/USD exchange rates led us to the conclusion that the window $\{1,2,3,4\}$ should be considered to correctly set the delay variable. The tests mean results are show in Table 1, Table 2, Table 3 and Table 4 and displayed in Figure 1, Figure 2 and Figure 3.

Table 1. NAR, delay = 1

	MAPE (%)	RMSE	POCID
NAR – NN	0.48 ± 0.7%	0.0073 ± 1.3%	48.95 ± 0.2%
LSTM	0.59 ± 13.3%	0.0089 ± 12.9%	47.42 ± 3.8%
LSTM Stacked	0.71 ± 9.1%	0.0105 ± 6.7%	48.87 ± 1.7%

Table 2. NAR, delay = 2

	MAPE(%)	RMSE	POCID
NAR – NN	0.47 ± 0.4%	0.0063 ± 0.1%	48.35 ± 0.7%
LSTM	0.32 ± 8.7%	0.0046 ± 2.9%	77.26 ± 0.8%
LSTM Stacked	0.93 ± 14.5%	0.0134 ± 16.4%	60.96 ± 4.9%

Table 3. NAR, delay = 3

	MAPE(%)	RMSE	POCID
NAR – NN	0.47 ± 0.3%	0.0063 ± 0.2%	47.93 ± 0.3%
LSTM	0.32 ± 24.4%	0.0048 ± 22.9%	70.45 ± 0.8%
LSTM Stacked	0.42 ± 32.7%	0.0083 ± 30.7%	67.05 ± 2.3%

Table 4. NAR, delay = 4

	MAPE(%)	RMSE	POCID
NAR - NN	0.47 ± 0.1%	0.0063 ± 0.8%	48.56 ± 1.0%
LSTM	0.36 ± 14.8%	0.0055 ± 14.3%	62.97 ± 4.3%
LSTM Stacked	0.49 ± 15.2%	0.0071 ± 17.7%	67.53 ± 2.0%

Figure 1 - NAR NN Forecast with delay = 2

Figure 1. NAR NN Forecast with delay = 2

Figure 2. LSTM Forecast with delay = 2

Figure 3. LSTM Stacked Forecast with delay = 2

5. Conclusions

The reported work aimed to develop suitable methods for financial data forecasting in terms of one-step-ahead nonlinear auto-regressive model. We have analyzed the exchange rate between EURO and US Dollar using samples recorded between October 1998 and October 2018. The prediction task has been carried out by three recurrent neural networks, the standard NAR neural network, LSTM and stacked LSTM respectively. The performance of the proposed

methods was evaluated by a long series of tests, the test results enabling us to conclude that the LSTM neural network outperforms the standard NAR one.

Since the experimentally derived results are quite encouraging, we are going to extend our study taking into account more complex prediction models and learning methodologies. Between possible extensions, we will consider the NARX model, technical indicators and several data mining techniques to better represent the available information and, consequently, to obtain more accurate algorithms.

References

- [1] C. Cocianu and M. Avramescu, „New Approaches of NARX-Based Forecasting Model. A Case Study on CHF-RON Exchange Rate”, *Informatica Economica*, Vol. 22, no. 2/2018, ISSN 1453-1305.
- [2] C. Cocianu and H. Grigoryan, *Machine Learning Techniques for Stock Market Prediction. A Case Study Of Omv Petrom Economic Computation and Economic Cybernetics Studies and Research*, Vol. 50, No. 3/2016, ISSN 0424-267X, pp. 63-82
- [3] C. Cocianu and H. Grigoryan, *An Artificial Neural Network for Data Forecasting Purposes*, *Informatica Economica*, Vol. 19, no. 2/2015, ISSN 1453-1305
- [4] S. Addai, “Financial Forecasting Using Machine Learning”, DOI: 10.13140/RG.2.1.1186.0083, 2016.
- [5] S. Siami-Namini and A. Siami Namin, “Forecasting Economics and Financial Time Series: ARIMA vs. LSTM”, arXiv:1803.06386v1, 2018.
- [6] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions”, *FAU Discussion Papers in Economics*, No. 11/2017, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, Erlangen, 2017.
- [7] A. Kumar Rout, P.K. Dash, R. Dash and R. Bisoi, “Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach”, *Journal of King Saud University – Computer and Information Sciences*, 2015.
- [8] Q. Gao, “STOCK MARKET FORECASTING USING RECURRENT NEURAL NETWORK”, A Thesis Presented to the Faculty of Graduate School at the University of Missouri-Columbia, 2016.
- [9] N. Zhang, T. Alemayehu and P. Behera, “Nonlinear Autoregressive (NAR) Forecasting Model for Potomac River Stage using Least Squares Support Vector Machines (LS-SVM)”, *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-4 Issue-9, February 2015.
- [10] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.
- [11] I. Sutskever, "Training Recurrent Neural Networks," Ph.D. Thesis, University of Toronto, 2013.
- [12] W. Bao, J. Yue and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory.” *PloS one*, 2017
- [13] Y. Goldberg, "A Primer on Neural Network Models for Natural Language Processing." *J. Artif. Intell. Res.(JAIR)* 57 (2016): 345-420.
- [14] J. Brownlee, *Stacked Long Short-Term Memory Networks*. Internet: <https://machinelearningmastery.com/stacked-long-short-term-memory-networks/> , August 18, 2017 [10.02.2019].