Artificial neural networks approach to the forecast of stock market price movements

Luca Di Persio
University of Verona
Department of Computer Science
strada le Grazie, 15, Verona
Italy
luca.dipersio@univr.it

Oleksandr Honchar
University of Verona
Department of Computer Science
strada le Grazie, 15, Verona
Italy
alex.gonchar@univr.it

Abstract: In this work we present an Artificial Neural Network (ANN) approach to predict stock market indices. In particular, we focus our attention on their trend movement *up or down*. We provide results of experiments exploiting different Neural Networks architectures, namely the Multi-layer Perceptron (MLP), the Convolutional Neural Networks (CNN), and the Long Short-Term Memory (LSTM) recurrent neural networks technique. We show importance of choosing correct input features and their preprocessing for learning algorithm. Finally we test our algorithm on the S&P500 and FOREX EUR/USD historical time series, predicting trend on the basis of data from the past n days, in the case of S&P500, or minutes, in the FOREX framework. We provide a novel approach based on combination of wavelets and CNN which outperforms basic neural networks approaches.

Key–Words: Artificial neural networks, Multi-layer neural network, Convolutional neural network, Long short-term memory, Recurrent neural network, Deep Learning, Stock markets, Time series analysis, financial forecasting

1 Introduction

During recent years, Artificial Neural Networks (ANN) have been interested by a renewed interest by academicians, as well as by practitioners of various type. The turning point is represented by the winning solution for ImageNet challenge in 2012, which was based on deep convolutional network trained on GPUs, see [1]. Then, ANN approaches to deep learning have been used in lot of different areas such, e.g., autonomic car driving, medicine, bots, playing games, finance, physics, etc. Within the financial arena, machine learning is already applied to, e.g., credit scoring, portfolio management, algorithmic trading, automated underwriting of loans, automated financial document classification, separation systems, etc. In the present work we are dealing with the ANN-machine learning approach to the study of stock price movements prediction.

The paper is structured as follows: in Sec. 2 we provide a short review already known techniques to stock price prediction based on *classic* machine learning algorithms, logistic regression or support vector machine; then, in Sec. 3, we discuss input data for algorithms, feature selection and preprocessing, eventually, in Sec. 4, we provide a detailed analysis of our numerical investigations based on the analysis of S&P500 and FOREX time series.

2 Existing approaches

The approaches used to forecast future directions of share market prices are historically splitted into two main categories: those that rely on technical analysis, and those that rely on fundamental analysis. Technical analysis uses only historical data such as, e.g., past prices, volume of trading, volatility, etc., while fundamental analysis is based on external information like, e.g., interest rates, prices and returns of other assets, and other macro/micro-economic parameters.

The *machine learning* approach to the latter problem has been declined in several forms, especially during recent years, also aiming to find an effective way to predict sudden financial crashes as, e.g., the one happened during the 2007-08 biennium. As an example, good results have been obtained using linear classifiers as the logistic regression one, which has been used to predict the Indian Stock market, see [2], or with respect to the S&P500 index, see [3]. More complicated techniques, as large-margin classifier or Support Vector Machine (SVM), was the best choice for prediction before the rise of neural networks. The latter uses a *kernel trick* which allows to consider our input data into a higher-dimensional space, where it is linearly separable. Successful applications of SVM can be found, e.g., for the study of the *Korea composite* stock prices market, see [4], and the NIKKEI 225 index, see [5], showing how such a method outperforms logistic regression, linear discriminant analysis and even some neural network approaches. Another popular approach based on decision trees, i.e. the *random forest* one, was also applied to this problem, see, e.g., [6], with respect to the BM&F/BOVESPA stock market, resulting in a high accuracy obtained by using as input features different technical indicators as simple and exponential moving averages, rate of change, stochastic oscillator, etc. The aforementioned references are all characterized by an accuracy in stock price movement forecasting that ranges between 70% and 90%. In what follows, we show that is possible to achieve the same accuracy with neural networks, working with preprocessed open/close/high/low data, also morking with high frequent, intra-day, data.

3 Data processing

In what follows we focus our attention on the S&P500, exploiting related data from 1950 to 2016 (16706 data points), and on the FOREX EUR/USD minute-by-minute data from 2015 (372210 data points). Usually stock market data looks like on Figure 1 and 2, where are displayed close prices of every day (Fig.1), resp. of every minute (Fig.2).

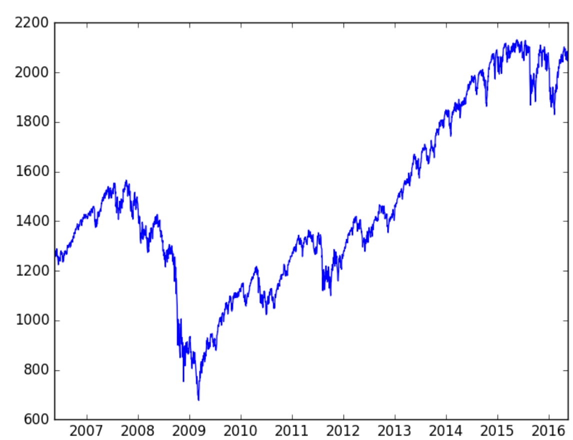

Figure 1: S&P500 index data from 2006 to 2016.

Our goal is the prediction of movements based on some information from previous data. Suppose we are going to predict if the close price of the next day, resp. minute, is larger or smaller than the previous one, based on the last 30 days, resp. minutes, of observations. Appropriate time window and prediction horizon should be chosen during hyper parameter optimization stage. First we have to scale or normalize our input data. The input vectors of the training data are normalized in such a way that all the features have zero-mean, and unitary variance. Usually

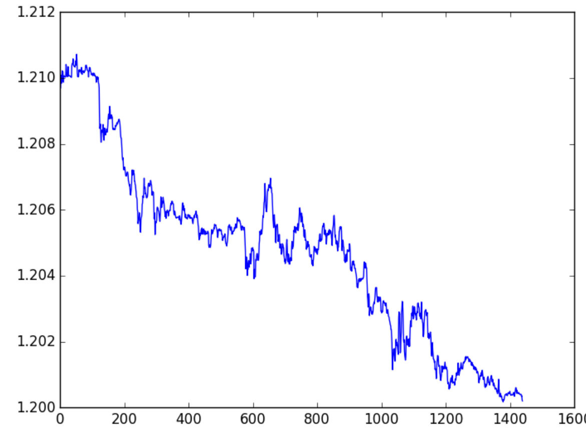

Figure 2: Forex EUR/USD pair 1 January of 2015 data.

data are scaled to be in the range [-1; 1], or [0; 1], in the neural network approaches such renormalization strongly depends on the activation function of every *neuron*. A different approach, see, e.g., [7], is about considering not raw open or close ticks, but calculating return during the day, resp. during the minute, and then using this data as the training set. In Fig. 3, we have reported the plot of returns for the S&P500 index.

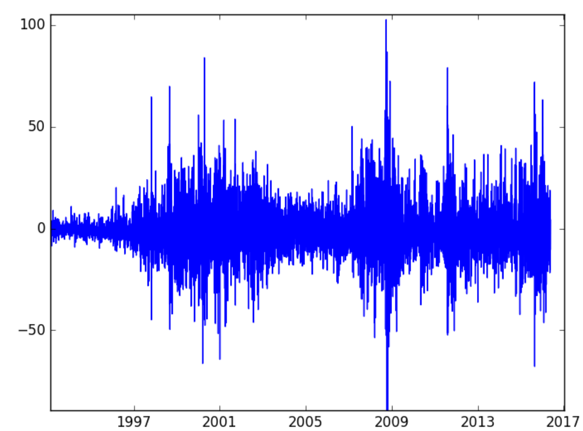

Figure 3: S&P500 index daily returns data.

In our research we use return prices as more representative data with normalization for stock price movement forecasting problem. In particular, we normalized our time series to have zero mean and unit variance maiking use of the *sklearn library*, see [9]. All existing time series have been splitted, accordingly to the latter, in train, 80%, resp. in test dataset, the remaining 20%; moreover we use 10% of training set for hyper parameter optimization. Every element of the train data set is a normalized time series with length of 30. Based on such a subdivision, we want to predict the transition [1; 0] if price goes, resp. [0; 1] if it goes down next day / minute, according with the particular index chosen.

4 Experimental results

In this section we provide the computational results related to the training process. All NN were trained using Keras deep learning framework, for Python. Every NN for S&P500 data was trained for 10 epochs, while we use 5 epochs for FOREX, exploiting the Adam optimization algorithm, see [8]. Computational time was reduced by using GPU hardware, in particular we made use of a GTX 860M, to speed up tensor multiplication, as well as other mathematical routines.

4.1 Multilayer perceptron architecture (MLP)

We use MLP with two hidden layers and test different layer sizes to determine the optimal size from the point of view of data modelling. We choose the rectified linear unit (ReLU) function as activation function, and, between two hidden layers, one dropout layer is included to prevent overfitting. The architecture of this model is shown in Fig. 4.

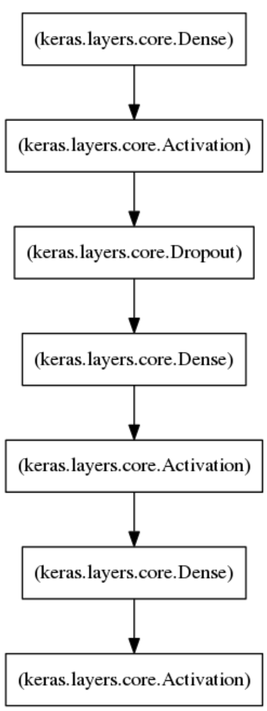

Figure 4: MLP architecture.

4.2 Convolutional neural network architecture (CNN)

We use CNN as sequential combination of 1D convolutional and max-pooing layers, choosing hyper parameters as follows:

- Number of filters = 64;
- Pool length = 2;
- Subsample length = 1;
- Activation function ReLU.

We provide experiment results with 1 and 2 hidden convolutional layers. The architecture of this model is shown in Fig. 5.

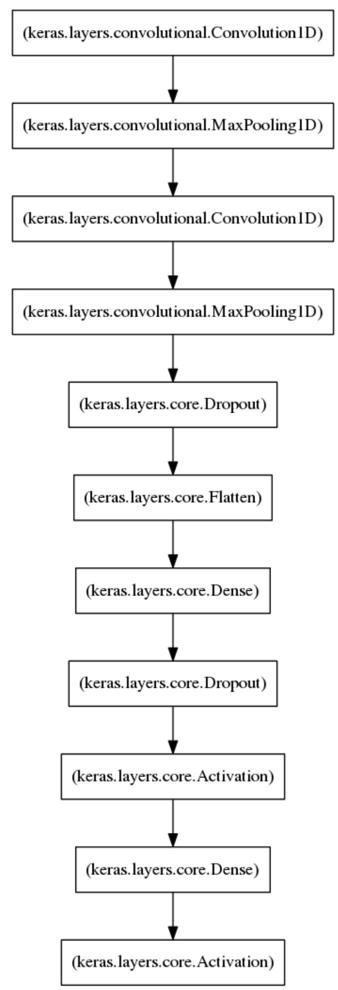

Figure 5: CNN architecture.

4.3 Recurrent neural network architecture (RNN)

As RNN, we choose the LSTM architecture, with 2 stacked recurrent layers and softmax in the end. Number of hidden neurons equals 100 while the activation functions are the hyperbolic tangent and the inner activations hard sigmoid functions. The architecture associated to the latter model is shown in Fig. 6.

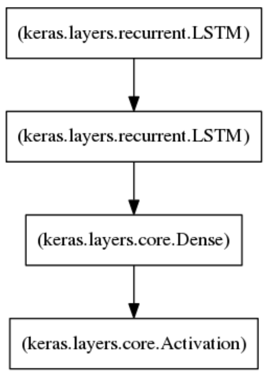

Figure 6: RNN architecture.

4.4 Experimental Wavelet + CNN (WCNN)

One of the non-traditional approaches for financial time series prediction with NN is the Wavelet-NN. Basically it is feature extraction using wavelet transform and passing them into the MLP, like in [10]. CNN showed better convergence and accuracy in previous experiments so we passed time series preprocessed with Haar wavelets to the 2-layer CNN.

4.5 Results

Detailed results of the described training-NN, are reported in Fig.7 and Fig.8.

	S&P500	Forex
MLP (100-50 hidden)	0.66	0.49
MLP (500-250 hidden)	0.58	0.48
CNN (2 layer)	0.56	0.47
RNN (2 layer)	0.57	0.48
Wavelet CNN	**0.55**	**0.45**

Figure 7: Log-loss after training different architectures

	S&P500	Forex
MLP (100-50 hidden)	0.59	0.81
MLP (500-250 hidden)	0.6	0.81
CNN (2 layer)	0.59	0.81
RNN (2 layer)	0.61	0.79
Wavelet CNN	**0.62**	**0.83**

Figure 8: Accuracy after training different architectures

5 Conclusion

In this work different artificial neural network approaches, namely MLP, CNN, and RNN, have been applied to stock market price movement forecasting. We compared results trained on a daily basis, for the S&P500 index, as well as in a minute-by-minute setting, in the case of Forex returns, reporting that convolutional neural networks (CNN) can model financial time series better then other architectures.
We also implemented novel Wavelet + CNN algorithm which outperforms other neural network approaches. It shows, that feature preprocessing is one of the most crucial parts in stock price forecasting.
We would like to underline that we achieved much better results, namely 62% accuracy maximum versus 83%, with Forex due to availability of more data that ca be used to implement a more effective learning patterns. In future researches, we aim at using more extended financial time series, and related features, to improve the neural networks training. Techincal indicators, moving averages and stochastic oscillators, will be also taken into account to increase accuracy of prediction.

References:

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, *NIPS 2012, Nevada*, 2012

[2] A. Dutta, G. Bandopadhyay and S. Sengupta, Prediction of Stock Performance in the Indian Stock Market Using Logistic Regression, *International Journal of Business and Information*, 2012

[3] J. Chen , M. Chen and Nan Ye, Forecasting the Direction and Strength of Stock Market Movement, *Technical report*, 2013

[4] Y. Wang and In-Chan Choi, Market Index and Stock Price Direction Prediction using Machine Learning Techniques: An empirical study on the KOSPI and HSI, *Technical report*, 2013

[5] Wei Huang, Yoshiteru Nakamori, Shou-Yang Wang, Forecasting stock market movement direction with support vector machine, *Computers and Operations Research, archive Volume 32, Issue 10*, 2005, pp. 2513–2522

[6] Marcelo S. Lauretto, B. C. Silva and P. M. Andrade, Evaluation of a Supervised Learning Approach for Stock Market Operations, *arXiv:1301.4944[stat.ML]*, 2013

[7] V.V.Kondratenko and Yu. A Kuperin, Using Recurrent Neural Networks To Forecasting of Forex, *arXiv:cond-mat/0304469 [cond-mat.dis-nn]*, 2003

[8] D. P. Kingma, J. Lei Ba, Adam: A method for stochastic optimization, *3rd Int. Conf. for Learning Representations, San Diego*, 2015

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research Volume 12*, 2011, pp.2825–2830

[10] Chong Tan, Financial Time Series Forecasting Using Improved Wavelet Neural Network, *Master Thesis, University of Aarhus*, 2009