European Journal of Operational Research 270 (2018) 654–669

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Computational Intelligence and Information Management

Deep learning with long short-term memory networks for financial market predictions

Thomas Fischer[1,*], Christopher Krauss[1]

Department of Statistics and Econometrics, University of Erlangen-Nürnberg, Lange Gasse 20, 90403 Nürnberg, Germany

ARTICLE INFO

Article history:
Received 12 May 2017
Accepted 27 November 2017
Available online 5 December 2017

Keywords:
Finance
Statistical arbitrage
LSTM
Machine learning
Deep learning

ABSTRACT

Long short-term memory (LSTM) networks are a state-of-the-art technique for sequence learning. They are less commonly applied to financial time series predictions, yet inherently suitable for this domain. We deploy LSTM networks for predicting out-of-sample directional movements for the constituent stocks of the S&P 500 from 1992 until 2015. With daily returns of 0.46 percent and a Sharpe ratio of 5.8 prior to transaction costs, we find LSTM networks to outperform memory-free classification methods, i.e., a random forest (RAF), a deep neural net (DNN), and a logistic regression classifier (LOG). The outperformance relative to the general market is very clear from 1992 to 2009, but as of 2010, excess returns seem to have been arbitraged away with LSTM profitability fluctuating around zero after transaction costs. We further unveil sources of profitability, thereby shedding light into the black box of artificial neural networks. Specifically, we find one common pattern among the stocks selected for trading – they exhibit high volatility and a short-term reversal return profile. Leveraging these findings, we are able to formalize a rules-based short-term reversal strategy that yields 0.23 percent prior to transaction costs. Further regression analysis unveils low exposure of the LSTM returns to common sources of systematic risk – also compared to the three benchmark models.

1. Introduction

Prediction tasks on financial time series are notoriously difficult, primarily driven by the high degree of noise and the generally accepted, semi-strong form of market efficiency (Fama, 1970). Yet, there is a plethora of well-known capital market anomalies that are in stark contrast with the notion of market efficiency. For example, Jacobs (2015) or Green, Hand, and Zhang (2013) provide surveys comprising more than 100 of such capital market anomalies, which effectively rely on return predictive signals to outperform the market. However, the financial models used to establish a relationship between these return predictive signals, (the features) and future returns (the targets), are usually transparent in nature and not able to capture complex non-linear dependencies.

In the last years, initial evidence has been established that machine learning techniques are capable of identifying (non-linear) structures in financial market data, see Huck (2009, 2010), Takeuchi and Lee (2013), Moritz and Zimmermann (2014), Dixon, Klabjan, and Bang (2015), and further references in Atsalakis and Valavanis (2009) as well as Sermpinis, Theofilatos, Karathanasopoulos, Georgopoulos, and Dunis (2013). Specifically, we expand on the recent work of Krauss, Do, and Huck (2017) on the same data sample for the sake of comparability. The authors use deep learning, random forests, gradient-boosted trees, and different ensembles as forecasting methods on all S&P 500 constituents from 1992 to 2015. One key finding is that deep neural networks with returns of 0.33 percent per day prior to transaction costs underperform gradient-boosted trees with 0.37 percent and random forests with 0.43 percent. The latter fact is surprising, given that deep learning has “dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains” (LeCun, Bengio, and Hinton, 2015, p. 436). At first sight, we would expect similar improvements in the domain of time series predictions. However, Krauss et al. (2017, p. 695) point out that “neural networks are notoriously difficult to train” and that it “may well be that there are configurations in parameter space to further improve the performance” of deep learning.

In this paper, we primarily focus on deep learning, and on further exploring its potential in a large-scale time series prediction problem. In this respect, we make three contributions to the literature.

* Corresponding author.
E-mail addresses: thomas.g.fischer@fau.de (T. Fischer), christopher.krauss@fau.de (C. Krauss).

[1] The authors have benefited from many helpful discussions with Ingo Klein and three anonymous referees.

https://doi.org/10.1016/j.ejor.2017.11.054
0377-2217/

- First, we focus on long short-term memory (LSTM) networks, one of the most advanced deep learning architectures for sequence learning tasks, such as handwriting recognition, speech recognition, or time series prediction (Graves et al., 2009; Graves, Mohamed, & Hinton, 2013; Hochreiter & Schmidhuber, 1997; Schmidhuber, 2015). Surprisingly, to our knowledge, there has been no previous attempt to deploy LSTM networks on a large, liquid, and survivor bias free stock universe to assess its performance in large-scale financial market prediction tasks. Selected applications, as in Xiong, Nichols, and Shen (2015), focus on predicting the volatility of the S&P 500, on forecasting a small sample of foreign exchange rates (Giles, Lawrence, & Tsoi, 2001), or on assessing the impact of incorporating news for specific companies (Siah and Myers (2016)). We fill this void and apply LSTM networks to all S&P 500 constituents from 1992 until 2015. Hereby, we provide an in-depth guide on data preprocessing, as well as development, training, and deployment of LSTM networks for financial time series prediction tasks. Last but not least, we contrast our findings to selected benchmarks from the literature – a random forest (the best performing benchmark), a standard deep neural net (to show the value-add of the LSTM architecture), and a standard logistic regression (to establish a baseline). The LSTM network outperforms the memory-free methods with statistically and economically significant returns of 0.46 percent per day – compared to 0.43 percent for the RAF, 0.32 percent for the standard DNN, and 0.26 percent for the logistic regression. This relative advantage also holds true with regard to predictional accuracy where a Diebold–Mariano test confirms superior forecasts of the LSTM networks compared to the applied benchmarks. Our findings are largely robust to microstructural effects. Specifically, when we implement the LSTM strategy on volume-weighted-average-prices (VWAPs) instead of closing prices, we see a decline in profitability, but the results are still statistically and economically significant. The same holds true for a weekly implementation with lower turnover – even after introducing a one-day-waiting rule after the signal. Only as of 2010, the edge of the LSTM seems to have been arbitraged away, with LSTM profitability fluctuating around zero after transaction costs, and RAF profitability dipping strictly into the negative domain.
- Second, we aim at shedding light into the black box of artificial neural networks – thereby unveiling sources of profitability. Generally, we find that stocks selected for trading exhibit high volatility, below-mean momentum, extremal directional movements in the last days prior to trading, and a tendency for reversing these extremal movements in the near-term future.
- Third, we synthesize the findings of the latter part into a simplified, rules-based trading strategy that aims at capturing the quintessence of the patterns the LSTM acts upon for selecting winning and losing stocks. A strategy that buys short-term extremal losers and sells short-term extremal winners leads to daily returns of 0.23 percent prior to transaction costs – so only about 50 percent of the LSTM returns. Regression analyses on systematic risk factors unveil a remaining alpha of 0.42 percent of the LSTM prior to transaction costs and generally lower exposure to common sources of systematic risk, compared to the benchmark models.

The remainder of this paper is organized as follows. Section 2 briefly covers the data sample, software packages, and hardware. Section 3 provides an in-depth discussion of our methodology, i.e., the generation of training and trading sets, the construction of input sequences, the model architecture and training as well as the forecasting and trading steps. Section 4 presents the results and discusses our most relevant findings in light of the existing literature. Finally, Section 5 concludes.

2. Data, software, hardware

2.1. Data

For the empirical application, we use the S&P 500 index constituents from Thomson Reuters. For eliminating survivor bias, we first obtain all month end constituent lists for the S&P 500 from Thomson Reuters from December 1989 to September 2015. We consolidate these lists into one binary matrix, indicating whether the stock is an index constituent in the subsequent month. As such, we are able to approximately reproduce the S&P 500 at any given point in time between December 1989 and September 2015. In a second step, for all stocks having ever been a constituent of the index during that time frame, we download daily total return indices from January 1990 until October 2015. Return indices are cum-dividend prices and account for all relevant corporate actions and stock splits, making them the most adequate metric for return computations. Following Clegg and Krauss (2018), we report average summary statistics in Table 1, split by industry sector. They are based on equal-weighted portfolios per sector, generated monthly, and constrained to index constituency of the S&P 500.

2.2. Software and hardware

Data preparation and handling is entirely conducted in Python 3.5 (Python Software Foundation, 2016), relying on the packages numpy (Van Der Walt, Colbert, & Varoquaux, 2011) and pandas (McKinney, 2010). Our deep learning LSTM networks are developed with keras (Chollet, 2016) on top of Google TensorFlow, a

Table 1
Average monthly summary statistics for S&P 500 constituents from January 1990 until October 2015, split by industry. They are based on equal-weighted portfolios per industry as defined by the Global Industry Classification Standards Code, formed on a monthly basis, and restricted to index constituency of the S&P 500. Monthly returns and standard deviations are denoted in percent.

Industry	No. of stocks	Mean return	Standard deviation	Skewness	Kurtosis
Industrials	80.6	0.99	5.36	−0.19	1.71
Consumer services	72.6	1.07	5.27	−0.20	2.59
Basic materials	35.2	0.90	6.31	−0.02	2.24
Telecommunications	10.7	0.92	6.50	0.34	4.76
Health care	41.3	1.33	4.40	−0.40	1.18
Technology	50.3	1.41	8.50	−0.06	1.11
Financials	78.0	1.13	6.17	−0.39	2.44
Consumer goods	65.2	1.04	4.53	−0.44	3.02
Oil and gas	31.2	1.00	6.89	−0.03	1.06
Utilities	34.6	0.85	4.54	−0.43	1.72
All	499.7	1.04	4.78	−0.49	2.01