CS230

Deep learning for relative stock performance prediction

Ruben Contesti, Sebastian Hurubaru, Jun Yan

Abstract

Over recent years several Deep learning methods have been applied to predict the stock market. At the same time, people find out that besides traditional finance data, some other unstructured data such as text data are also significantly correlated to the stock market, and thus they are good predictor candidates for the market prediction. In this work we combine the Tweet data and some traditional finance data to predict the relative performance of Apple, using a deep neural network structure based on wavelet transform, stacked auto-encoder and LSTM/GRU.

1 Introduction

Stock market prediction has always received a lot of attention and focus, since successful prediction could yield significant profit. At the same time, stock market prediction is also challenging, since the data is usually noisy and volatile. With the development and popularity of Deep learning, recently various Deep learning methods have been applied to finance data, and some of them turns out to have very good performance ([7], [14], [12], [1]).

However, there still exist many difficulties and challenges in this topic. One of the main challenges have been the use of unstructured data for the prediction of financial assets returns. While there have been a vast literature in terms of use of time series algorithm, the use of unstructured data has remain opaque. For instance there has recently been reported the use of satellite images for stock prices return. Hedge Funds have been using the images of JC Penney stores to detect if people were attending the store many months in advance that information has been made public. Another type of unstructured data that has recently reach notoriety has been the text data. One of the most successful hedge funds in the World, Two Sigma, recent launched the Kaggle competition about using news data to predict the market.

In this work, we try to combine the use of text data and the traditional finance data together for the market prediction. We make the use of Twitter information to extract a signal that would allow us predict next returns for a universe of stocks. Together with other financial features such as close price and volume, we use recurrent neural networks to predict stocks' relative performance.

2 Related Work

With the development of the deep learning methods and the increase of the amount of data, recently there are various applications of deep learning to finance prediction. For example, In [12] and [14], deep belief networks is used for time series forecasting. In [7], the authors use NLP and CNN to predict the influences of events on stock price.

On the NLP side, previously there have been several work shown that tweet data is correlated with market data ([3], [13]). Recently Google open sourced BERT [6], a state-of-art NLP pre-training, which provides bidirectional and contextual language representation. In our work, we use BERT to conduct our NLP analysis.

We borrow our network idea from [1]. In that paper, the authors first preprocess the data using wavelet transform, and then use the stacked autoencoder to generate deep features, and finally fit a LSTM for the prediction.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Features

We have gathered a data set from the beginning of 2009 until end of 2018, containing in total 2414 days of data. The data set containing 20 features which are separated in three categories described next.

The first category contains the historical trading data for Apple (APPL) and SPY. The data includes the Open, High, Low, Close, Adj Close prices as well as the Volume. Based on the data we also generated a new feature, relative performance.

The second category contains two types of macroeconomic indicators: interest rate and exchange rate. We choose the Effective Federal Funds Rate (DFF) as the interest rate and the US dollar index (DX-Y.NYB) as the proxy for the exchange rate, similar to what was described in [1]

The third category would be the sentiment extracted from the top tweets truncated to a 100 tweets a day, and only use those tweets with times between the last close of the market and the current opening. We do that so no future information leaks into our prediction.

The goal of our model is to predict the movement of APPL. Instead of directly predict the price, we predict the relative performance defined as follows

$$y_i = \frac{\text{APPL}_{i+1}/\text{APPL}_i}{\text{SPY}_{i+1}/\text{SPY}_i} - 1, \tag{1}$$

where APPL_i and SPY_i are respectively the close price of AAPL and SPY at day i. The relative performance describe the movement of Apple price relative to the SPY movement. We are interested in predicting relative performance, because we want to be "market neutral"; we do not want our predictions of apple to be influenced by the overall market.

4 Methods

We analyze the tweet data by BERT to extract a signal, which together with other features is our input data. Using the idea from [1], we build our model as a combination of wavelet transform, stacked autoencoder, and LSTM/GRU.

4.1 Tweet Data Analysis

We believe the use of transfer learning is a smart approach since we are standing on top of a giant's shoulder like Google who has already spend thousands of TPU hours training their models. As the BERT project states: "The other important aspect of BERT is that it can be adapted to many types of NLP tasks very easily. In the paper, we demonstrate state-of-the-art results on sentence-level (e.g., SST-2), sentence-pair-level (e.g., MultiNLI), word-level (e.g., NER), and span-level (e.g., SQuAD) tasks with almost no task-specific modifications.

In order to obtain the sentiment feature, we use the pre-trained weights of BERT and fine-tune it to predict movie reviews. We then use the predicted model with tweets. We finally we average the sentiment for each day and use it as a feature in the model.

We train a model with movie reviews mainly for two reasons. First unlike, tweets movie reviews are already labeled. Second, reviews are in length and in style very similar to tweets.

4.2 Network architecture

Wavelet Transform

Financial data is always noisy and fluctuating. Wavelet transform and thresholding is a common method to denoise the data, by first using wavelet functions of different scales to fit the data, and then throw out those wavelets with small weights (see more description in [8], [9], [15]). In our work we use wavelet transformation to pre-process the data. We chose the Daubechies as family functions as opposed to Haar, as used in [1], to construct a set of wavelet orthonormal basis functions that have become the cornerstone of wavelet applications today.

After applying multilevel decomposition using the maximum level of decomposition available for the data set and extracting the approximation coefficients array and per level details coefficient arrays, for each detail coefficient array we are using non-negative garrote threshold [10] with the standard deviation as threshold value and then we do the multilevel reconstruction to get the denoised data. We apply the wavelet transformation twice to get the best results as described in [1].

Stacked Autoencoder

When training a deep neural network, sometimes it is beneficial to use greedy layerwise approach to pretrain the model, and stacked autoencoder is one of such approaches ([2]). Each single layer autoencoder is a three-layer network, the input layer, the encoder layer, and the decoder layer. The idea is to train the parameters of the three-layer network, to make the decoder layer and the input layer close, and then throw out the decoder layer and use the encoder layer as the new feature.

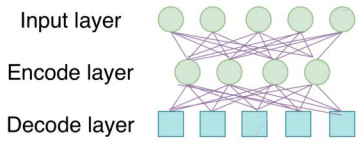

Figure 1: SAE Structure

The stacked autoencoder is just a multi-layer autoencoder, where each layer is the encoder layer of the last layer. The depth of the SAE is important and we set it to 5 as suggested in [4]. The autoencoder hidden layers have a size of 15, 12, and 10 respectively. So from the initial 20 features we generate a set of 10 deep features.

LSTM and GRU

Long short-term memory (LSTM) ([11]) and Gated Recurrent Units (GRU) ([5]) are variations of recurrent neural network (RNN), and they are common deep network architectures for fitting time series data. For traditional neural network, the units of the input vectors are assumed to be independent. However, for time series data, oftentimes there are strong correlation across the data, and RNN is introduced to address this dependence issue. Due to the vanishing gradient issue, RNN is hard to capture the long-term dependencies, and LSTM and GRU are come up to solve this problem.

The plot of the architecture is as follows:

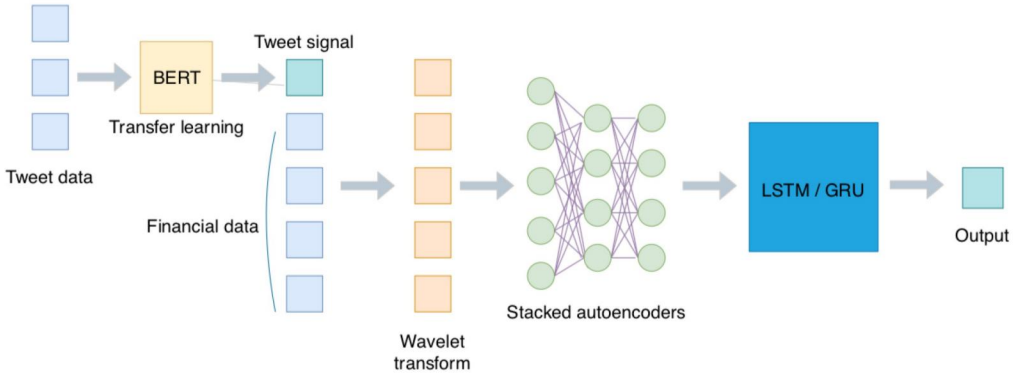

Figure 2: Architecture

5 Results and Discussion

5.1 Performance measurement

We are measuring the performance in two ways. First by getting the accuracy of our predictions model and secondly for each model we test the Kaggle score and the profitability .

5.1.1 Predictive performance accuracy

We are using three metrics to measure the accuracy of our models: RMSE, Theil U and MAPE. Denoting by y_t the real value and y_t^* the predicted value, these metrics are defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{N}(y_t - y_t^*)^2}{N}},\ \text{TheilU} = \frac{\sqrt{\frac{\sum_{t=1}^{N}(y_t - y_t^*)^2}{N}}}{\sqrt{\frac{\sum_{t=1}^{N}(y_t)^2}{N}} + \sqrt{\frac{\sum_{t=1}^{N}(y_t^*)^2}{N}}},\ \text{MAPE} = \frac{\sum_{t=1}^{N}\left|\frac{y_t - y_t^*}{y_t}\right|}{N}$$

5.1.2 Profitability performance

We design a simple buy or sell strategy using our prediction for the relative performance. If the prediction is positive, it means that buying Apple stock is better than buying SPY, therefore we buy one apple stock and sell it the next day, and sell one SPY stock and buy it the next day. Similar for the case the prediction is negative. The formula for the profitability P is

$$\text{P} = 100\sum_{i=1}^{N}\frac{\text{sgn}(y_i^*)(\text{APPL}_{i+1} - \text{APPL}_i - \text{SPY}_{i+1} + \text{SPY}_i) + \alpha(\text{APPL}_i + \text{SPY}_i + \text{APPL}_{i+1} + \text{SPY}_{i+1})}{\text{APPL}_i + \text{SPY}_i},$$

where α is the trading cost which we choose to be 0.0001.

5.2 Results

We have compiled in total 8 models: GRU/LSTM - a single layer of GRU/LSTM with 128 units; Wavelet GRU/Wavelet LSTM - the previous model with performing two wavelet transformation on the features; SAE Wavelet GRU/SAE Wavelet LSTM - the model above with encoding the twenty features in only ten deep features; Tweets SAE Wavelet GRU/Tweets SAE Wavelet LSTM - the model above with adding extracted sentiment from the daily tweets as feature

We trained all models with the following hyperparameters: 5000 epochs; initial learn rate of 0.05 and a decay of 0.0005; batch size of 64; dropout rate of 0.15

The results with the values for metrics defined above can be seen below

Model	RMSE	Theil U	MAPE	Kaggle score	Profitability
GRU	0.016241	0.012372	1.624109	0.299805	16.90
Wavelet GRU	0.018551	0.014131	1.855082	0.3453428	11.23
SAE Wavelet GRU	0.016203	0.012315	1.620265	0.344048	17.03
Tweets SAE Wavelet GRU	0.016249	0.01235	1.624864	0.344049	17.06

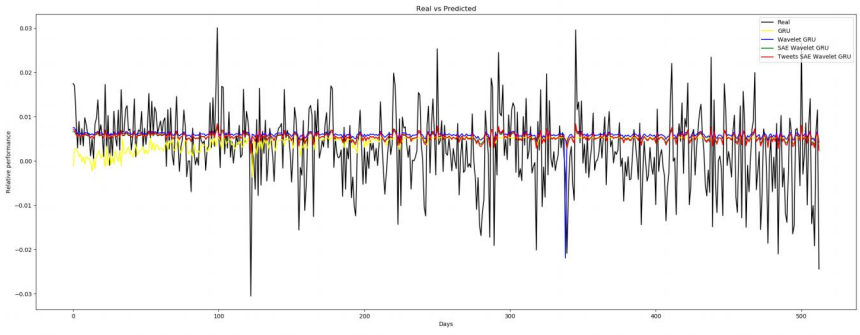

Figure 3: GRU Model Real Vs Predicted

Model	RMSE	Theil U	MAPE	Kaggle score	Profitability
LSTM	0.016772	0.01274	1.677205	0.343304	10.18
Wavelet LSTM	0.016558	0.012582	1.655797	0.301474	12.57
SAE Wavelet LSTM	0.016128	0.01226	1.612825	0.347782	17.04
Tweets SAE Wavelet LSTM	0.016002	0.012164	1.600169	0.347783	17.06

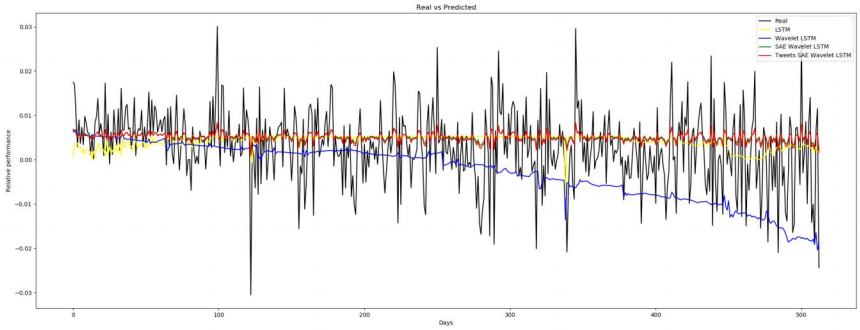

Figure 4: LSTM Model Real Vs Predicted

In the figure above we compare the real data and the predictions from the baseline model. As we can see, the prediction is smoother than the real data, which is quite noisy. The prediction follows the trend of the real data globally. The LSTM model seems do capture the idea of 'moving average'.

6 Conclusion and Future Work

In general we can observe that the best results according to both accuracy and profitability measures are a combination of using Wavelet transformation, Stacked Auto Encoders, Tweets sentiment and GRU/LSTM, with LSTM generating slightly better results than GRU. There are though some exceptions where the base GRU outperforms the GRU with wavelet transformation. Afuture step would be to choose different ways in removing the noise after performing the wavelet transformation, like just removing a specific number of details coefficient levels.

Finally, if we want to compare our results with those of the Kaggle competition we should make an adjustment to our scores because we worked with daily data and Kaggle competition worked with ten day returns. This adjustment implies multiplying our score by $\sqrt{10}$. That produces a score of 1.1. Taking into account that there are other differences that we are not taking into account like the noisier nature of our sentiment data or the fact that our universe only consists on one stock, this score would have placed us on the position 462 of the leader-board.

In the future, we would like to replicate the Kaggle competition more closely by using real news instead of tweets. Unfortunately that information isn't publicly available at the moment of writing this report.

Other approach that we would like to explore in the future is to weigh each tweet by the importance of the tweeter account behind it. For instance, we conjecture that a tweet from @POTUS taking about $AAPL would have more impact than those of somebody else.

7 Contributions

Ruben formed the team, came up with the idea and supported the team the whole time with his domain knowledge. Jun helped taming the complexity of the time series and did literature review. Sebastian implemented and trained the various models. The three of us worked evenly in generating the report and Jun created the final poster.

8 Code

Our current work can be found:
`https://github.com/SebastianHurubaru/deep_aapl_spy_rel_perf` [1]

[1] The GitHub repository is public

References

[1] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[3] Ray Chen and Marius Lazer. Sentiment analysis of twitter feeds for the prediction of stock market movement. *stanford edu Retrieved January*, 25:2013, 2013.

[4] Zhao X Wang G Gu Y Chen Y, Lin Z. Deep learning-based classification of hyperspectral data. *IEEE J Sel Top Appl Earth Observ Remote Sens*, 2014.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[8] David L Donoho and Iain M Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224, 1995.

[9] IM Dremin and AV Leonidov. Volatility dynamics of wavelet-filtered stock prices. *Bulletin of the Lebedev Physics Institute*, 35(1):1–5, 2008.

[10] Hong-Ye Gao. Wavelet shrinkage denoising using the non-negative garrote. *Journal of Computational and Graphical Statistics*, 7(4):469–488, 1997.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] Takashi Kuremoto, Shinsuke Kimura, Kunikazu Kobayashi, and Masanao Obayashi. Time series forecasting using a deep belief network with restricted boltzmann machines. *Neurocomputing*, 137:47–56, 2014.

[13] Venkata Sasank Pagolu, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. Sentiment analysis of twitter data for predicting stock market movements. In *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*, pages 1345–1350. IEEE, 2016.

[14] Furao Shen, Jing Chao, and Jinxi Zhao. Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing*, 167:243–253, 2015.

[15] Edward W Sun, Yi-Ting Chen, and Min-Teh Yu. Integrated wavelet denoising method for high-frequency financial data forecasting.