DLAI Project : Stock price forecasting with deep neural networks

July 19, 2021

Selim Yahia-Messaoud

Abstract

Stock price prediction is a highly complex and interesting topic. The purpose of this report is to present how deep neural networks can be applied to model financial markets.

1. Introduction

There exist two main type of strategies to do stock price forecasting : fundamental analysis and technical analysis. The former aims at measuring a financial asset's intrinsic value by relying on news, while the later uses quantitative data to identify trading opportunities. Latest advances in deep learning allow scientists and engineers to develop strategies at the intersection of those two, by combining social media and news analysis, with quantitative data. However, my research only covers the use of deep neural networks for technical analysis. I will present four different algorithms applied to stock price prediction. The first one is the Long-Short Term Memory (LSTM) algorithm, which has proven to be highly effective in time series forecasting. I will compare it to a convolutional neural network (CNN), then combination of both CNNs and LSTM will be studied. Finally, we will see what kind of results we can achieve by applying the Self Attention mechanism to this problem. The code used for this study is available at : https://github.com/selimym/Stock-price-forecasting-DLAI-Project.

2. Related Work

Banks and hedge funds have always been investing considerable resources to win the race of stock price prediction. While the best strategies may not be publicly shared for obvious reasons, a large amount of research articles studying the performance of the different deep learning algorithms applied to forecasting financial markets is available. (Sreelekshmy Selvin, 2017) compared LSTM and sliding CNNs and found at that the latter performed better thanks to its ability to identify changes in trend. Wenjie Lu & Wang (2020) and Jimmy Ming-Tai Wu (2021) both explored the combination of CNN and LSTM.

3. Method

The data used for this study is the daily open price of Alphabet stock (GOOG) from January 1, 2015 to June 14, 2021. No heavy processing was applied to the data. A 70-15-15 train, validation, test set split was performed, then the data was scaled down for faster training of the models. The scaling parameters were computed from the train set, and applied to all three data sets.

LSTM The first model trained consisted of 2 stacked LSTM layers, taking a sequence of length 10 as input. The time series data being relatively small, the network was trained quickly. Therefore I decided to train it with a small learning rate even if it took more epochs. This model serves as a baseline for the other experiments.

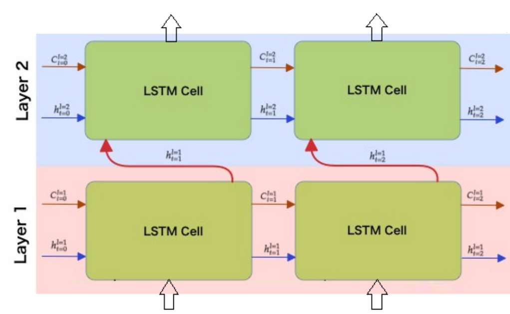

Figure 1. Architecture used for the LSTM model

CNN The second network studied was a classic convolutional neural network. The idea behind this network architecture is to identify specific patterns (see Investopedia), which are used by many technical analysts to anticipate the future direction of a security's price. 1D convolutions are good at capturing the relations between neighboring data points. Four convolutional layers were stacked together and followed by a global average pooling layer and a dense layer. The receptive field increases with the number of lay-

Email: Selim Yahia-Messaoud <yahiamessaoud.1965129@studenti.uniroma1.it >.

Deep Learning and Applied AI 2021, Sapienza University of Rome, 2nd semester a.y. 2020/2021.

ers, granting the model the ability to analyze small patterns to identify bigger trends. CNNs also have the advantage of being faster to train than LSTM models, because we can easily do most operations in parallel.

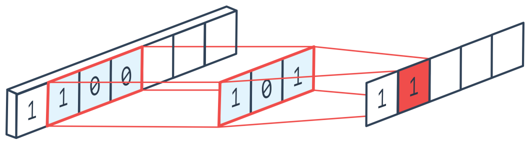

Figure 2. 1D Convolution feature extraction

CNN-LSTM I wanted to explore the performance of a model using both convolutional and LSTM layers. The goal is to combine the feature extraction ability of the CNN to identify patterns, and the good performance of LSTM on time series.

Self Attention Model Last but not least, I built a Transformer model using the self attention mechanism. Transformer have shown amazing results with Natural Language Processing tasks, and have been applied to many other problems since. I replaced the feed forward neural networks used in the encoder and decoder by convolutional layers, to see if we could get a performance improvement by capturing the information lying on the structure of the data. Compared to LSTM, Transformers benefit from faster training speed because operations can be done in parallel, and the self attention mechanism is supposed to limit the information loss observed in LSTMs. However, they generally need more data to be fully trained.

4. Experimental Results

Limits. The Mean Squared Error, computed between the prediction and the true security value, was used to train the models and compare their respective performance. All models performed quite well but none of them was able to greatly outperform all other methods.

Table. Tables can be used to report quantitative results, here is one random example:

Table 1. Model performance comparison (MSE)

Models	LSTM	CNN	CNN-LSTM	Self-Attention
MSE	0.0169	**0.0112**	0.028	0.353

As expected, all models showed really similar performance. Further architecture experiments and hyper parameters tuning may change which model performs the best. I was surprised to see that the convolutional neural network achieved the best results, but this confirms the observations of (Sreelekshmy Selvin, 2017). It appeared that all models performed better with an input sequence of size 10. For the LSTM models, this can be explained by the fact that this input sequence is encoded in a fixed size vector. Therefore, if sequences are too big it is harder from the LSTM to encode them without loosing too much information. The bad performance of the Transformer model may be caused by the limited amount of data available or because the architecture used was not optimal for this problem.

5. Conclusion and discussion

Popular deep learning models offer amazing performance with very limited parameter tuning and feature engineering. However, to obtain performance on par with algorithmic traders, advanced model designs and strategies are mandatory. A simple improvement over the models presented in this research would be to combine different models with stacking, to take advantage of the trends and patterns identified by each model, or to analyze different time frames with the same model. Models developed by quantitative researchers also generally use multi variate time series as inputs. Information that can be useful to improve the accuracy of the prediction can be the stock price of similar securities, the general sentiment regarding the studied company derived from news. Recently we have seen some companies going as far as using satellite imagery to have more insights on oil production (WSJ, 2019).

References

Investopedia. Introduction to technical analysis price patterns. URL https://www.investopedia.com/articles/technical/112601.asp.

Jimmy Ming-Tai Wu, Zhongcui Li, N. H. B. V. . J. C.-W. L. A graph-based cnn-lstm stock price prediction algorithm with leading indicators. *Springer*, 2021.

Sreelekshmy Selvin, R Vinayakumar, E. A. G. V. K. M. K. P. S. Stock price prediction using lstm,rnn and cnn-sliding window model. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017.

Wenjie Lu, Jiazheng Li, Y. L. A. S. and Wang, J. A cnn-lstm-based model to forecast stock prices. *Hindawi*, 2020.

WSJ. Want to make a big bet on oil prices? try measuring shadows — wsj. 2019. URL https://www.youtube.com/watch?v=wUgqJTTmVZI.