Bitcoin Stock Prediction Using Deep Learning and Sentiment Analysis

Arjun N. Chandra

Northeastern University

Abstract

This paper presents a comparison and selection of the best model from 2 deep machine learning solutions to predict the closing price of the Bitcoin cryptocurrency stock. The first framework is realized with a sequential model consisting of a linear stack of 2 Long Short-Term Memory (LSTM) layers and a single Linear Dense Layer. The second framework is a Multi-Layer Perceptron Classifier trained on the sentiment of the public company history and past prices as key data points. Cryptocurrency is an electronic and decentralized alternative to government-issued money, with Bitcoin as the best-known example of a cryptocurrency.

Keywords: Machine learning; Recurrent Neural Networks; Long Short-Term Memory; Deep Learning; Cryptocurrency; Bitcoin; Algorithmic Trading; Quantitative Finance; Natural Language Processing; Sentiment Analysis; Multi-Layer Perceptron Classifier

Introduction

There are several existing deep machine-learning approaches to financial market trading. The motivation for this project was derived by my curiosity to understand the power of recurrences i.e. feed the output back to the input, in an LSTM over a MLP (a type of feed-forward network) coupled with sentiment analysis of historical company articles that lacks this feature; on the prediction of stock prices. While the best-known example of a cryptocurrency is Bitcoin, there are more than 100 other tradable cryptocurrencies, called altcoins (meaning alternative to Bitcoin), competing each other and with Bitcoin (Bonneau et al., 2015). The motive behind this competition is that there are a number of design flaws in Bitcoin, and people are trying to invent new coins to overcome these defects hoping their inventions will eventually replace Bitcoin (Bentov et al., 2014; Duffield and Hagan, 2014). To June 2017, the total market capital of all cryptocurrencies is 102 billion in USD, 41 of which is of Bitcoin. Therefore, regardless of its design faults, Bitcoin is still the dominant cryptocurrency in markets. As a result, many altcoins cannot be bought with fiat currencies, but only be traded against Bitcoin. Hence, I chose Bitcoin as my commodity in order to make wiser future investments for my cryptocurrency portfolio.

Two natures of cryptocurrencies differentiate them from traditional financial assets, making their market the best test-ground for algorithmic portfolio management experiments. These natures are decentralization and openness, and the former implies the latter. Without a central regulating party, anyone can participate in cryptocurrency trading with low entrance requirements. One direct consequence is abundance of small-volume currencies. Affecting the prices of these penny-markets will require smaller amount of investment, compared to traditional markets. This will eventually allow trading machines to learn and take advantage of the impacts by their own market actions. Openness also means the markets are more accessible. Most cryptocurrency exchanges have application programming interface for obtaining market data and carrying out trading actions, and most exchanges are open 24/7 without restricting frequency of trading. These non-stop markets are ideal for machines to learn in the real world in shorter time-frames. (Jiang, Xu and Liang, 2017)

In a traditional recurrent neural network, during the gradient back-propagation phase, the gradient signal can end up being multiplied a large number of times (as many as the number of time steps) by the weight matrix associated with the connections between the neurons of the recurrent hidden layer. This means that, the magnitude of weights in the transition matrix can have a strong impact on the learning process.

If the weights in this matrix are small (or, more formally, if the leading eigenvalue of the weight matrix is smaller than 1.0), it can lead to a situation called vanishing gradients where the gradient signal gets so small that learning either becomes very slow or stops working altogether. It can also make more difficult the task of learning long-term dependencies in the data. Conversely, if the weights in this matrix are large (or, again, more formally, if the leading eigenvalue of the weight matrix is larger than 1.0), it can lead to a situation where the gradient signal is so large that it can cause learning to diverge. This is often referred to as exploding gradients.

These issues are the main motivation behind the LSTM model which introduces a new structure called a memory cell. A memory cell is composed of four main elements: an input gate, a neuron with a self-recurrent connection (a connection to itself), a forget gate and an output gate. The self-recurrent connection has a weight of 1.0 and ensures that, barring any outside interference, the state of a memory cell can remain constant from one time step to another. The gates serve to modulate the interactions between the memory cell itself and its environment. The input gate can allow incoming signal to alter the state of the memory cell or block it. On the other hand, the output gate can allow the state of the memory cell to have an effect on other neurons or prevent it. Finally, the forget gate can modulate the memory cell's self-recurrent connection, allowing the cell to remember or forget its previous state, as needed.

Several NLP problems end up taking a sequence and encoding it into a single fixed size representation, then decoding that representation into another sequence. For example, we might tag entities in the text, translate from English to French or convert audio frequencies to text. There is a torrent of work coming out in these areas and a lot of the results are achieving state of the art performance.

I believe the biggest difference between the NLP and financial analysis is that language has some guarantee of structure, it's just that the rules of the structure are vague. Markets, on the other hand, don't come with a promise of a learnable structure, that such a structure exists is the assumption that this project would prove or disprove (rather it might prove or disprove if I can find that structure).

Assuming that a structure exists; the idea of summarizing the current state of the market in the same way we encode the semantics of a paragraph seems plausible to me.

References

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. Neural Networks, IEEE Transactions on, 5(2), 157-166.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

Graves, A. (2012). Supervised sequence labelling with recurrent neural networks (Vol. 385). Springer.

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 142-150). Association for Computational Linguistics.

Yann LeCun, Yoshua Bengio & Geoffrey Hinton (2015). Deep Learning. Nature.