

Jason Ma and Joe Nunez

CS 152

Prof. Sweedyk

December 7, 2017

Final Project Writeup

Our project focused on trying to develop a trading algorithm based on Neural Networks. We based our networks on previous work by Abhishek Kar¹ and a Github Repository² who worked with a multi-layer perceptron and LSTM neural network respectively. In regards to Kar's work, we were able to mimic his results by recreating his network and running it on the same dataset from his paper. Additionally, we were able to get similar results on other datasets³ as well. Under the advice of Professor Sweedyk, we attempted to implement the LSTM network from the paper and were able to reproduce decent results, which were slightly worse than those on the repository but better than the multi-layer perceptron, but with a much longer training time.

Since our initial goal was to implement trading strategies, we tried transferring our work over to Quantopian, so that we could use their framework to backtest our network and visualize the returns. Unfortunately, Quantopian's Python does not include Keras, which we used to implement our network, so we could not implement the LSTM. We then attempted to recreate our multi-layer perceptron using sklearn, but after a few hours of learning how to handle data from Quantopian's pipeline, we realized that Quantopian did not include sklearn's neural network library either, so we then moved onto trying to develop our own trading system.

¹ https://people.eecs.berkeley.edu/~akar/IITK_website/EE671/report_stock.pdf

² <https://github.com/BenjiKCF/Neural-Network-with-Financial-Time-Series-Data>

³ <https://www.kaggle.com/pablocastilla/predict-stock-prices-with-lstm>

Our trading system was relatively simple, we reused our test set, and fed it into the network day by day. If the predicted price would go up by some threshold percentage relative to the previous value, we would invest all our capital into it, and if it went down by a similar percentage, we would short the stock - this means if the actual price goes down, we gain money, and if it goes up, we lose money. Initially, our tests were wildly successful, often netting more than a 10x increase in our starting money, which indicated some aspect of our testing was incorrect⁴. Thus, after some tweaking, we were able to see variable results. For the most part, our strategy made money (about 1.5x), though occasionally, we lost about 10% of our starting capital.

From this project, we learned many things about the difficulties of implementing neural networks using third party platforms, and the difficulties of creating a system to test our neural network with simulated trading. Though going forward, we already have a system established to test our neural network, which means that we will be able to easily test the changes from modifying our network further if we continue this project. However, Quantopian has a system called Zipline, which might allow us to more faithfully test our networks (Zipline simulates the change in stock price during the time delay between when a trade is submitted and executed), though we aren't sure if it is compatible with any neural network, and we didn't have time to explore it. The most direct expansions of our work would be to test an LSTM on our mock trading system (which would require AWS due to the increased training time of LSTMs), and to use Zipline to test other datasets and more faithful trading simulations (if compatible).

⁴ For a while, we accidentally were calculating return based off of our projected value instead of the real value, which meant that the network worked great if the market behaved exactly like the network.

Overall, we each spent about 6-7 hours on this project per week. Though, early on it was only about 4-5 hours since there were relatively few problems in implementing the projects from the paper. Later on, there were some weeks where each of us spent nearly 12 hours, which was mostly due to Quantopian and our issues porting our networks to Quantopian. Overall, we think our results are quite promising, and that given more time, we could possibly get even better results.