Forecasting Movements in Stock Prices

Final Project Report for EEL5825
Professor Dr. Ying Ma
Fall 2021

James Laughridge
The University of Central Florida

1 Abstract

This paper offers an analysis of the use of a neural network for stock market prediction. The project aimed to make accurate predictions on the direction of future stock price based on recent stock price and volume data. Variations of architecture including the number of input features, hidden layers, neurons per layer and activation function were evaluated. The results suggest that neural networks can successfully predict short term future price direction using recent data. This project offers practical insights and potentially useful conclusions on real world applications of neural network use in stock trading.

2 Introduction

Much research has been done into the question of predictability in the stock market using neural networks. Historically the Efficient Market Hypothesis has posed that an efficient market fully reflects all the information available pertaining to it and cannot be predicted based on any previous information or behavior (Sewell, 20 January 2011). In 1889 George Gibson, in his book titled *The stock Markets of London, Paris and New York*, made the case against markets being predictable when he wrote that "the value which they [stock share prices] acquire may be regarded as the judgement of the best intelligence concerning them", meaning that the current price of a share of stock is already the accurate price given all the available information (Gibson, 1889). The idea that a stock's future price could be predicted using currently available information runs counter to this Efficient Market Hypothesis, implying that the current price is just transitory on the way to the true best price. In 1889 when George Gibson wrote his book he may very well have been correct, however due to advances in technology the stock market George Gibson was referring to is very different from the stock market we have today.

More recently there is support for the notion that there might be underlying deterministic attributes of future stock prices, although opinions on predictability still vary. Abhyankar Copeland, and Wong in their paper titled *Structure in Stock-Market Indexes*, concluded that there might be underlying deterministic, nonlinear, nonchaotic process in the data, but that the data process is still dominated if not swamped by noise (A. Abhyankar, 1997). Other attempts at using neural networks to predict stock prices have produced mixed results with limited success (Eunsuk Chong, 15 November 2015) (Jingyi Shen, 2020) (Cheng-Lung Huang, March 2009) (Kyoung-jae Kim, August 2000).

Being able to improve stock market predictions could directly improve order efficiency. This improvement would aid an individual trader or investor in market timing, increasing profit and investor confidence.

3 Background

3.1 Stock Market

Stocks trading has seen an increase in speed and accessibility in recent years. New technologies such as trading applications for smart phones have expanded access to the stock market, allowing almost anyone with a smart phone to place a trade from almost anywhere at any time. The types of trading can generally be classified into three categories, algorithmic trading, high frequency trading and day trading. Algorithmic trading describes any trading that uses a computer to identify trades based upon an established algorithm and automates trade execution. High Frequency Trading (HFT) is a form of algorithmic trading that occurs at extraordinarily high speeds, often holding a position for less than a second. In a Congressional report on HFT, it was estimated that half of all domestic stock trades in the United States fall into the category of HFT (Service, 19 June, 2014). Day trading describes trading where the same stock is bought and sold in the same day. Day trading can be manual or algorithmic and describes everything from large hedge funds to individuals manually trading on their computer, smart phone or tablet. There is another type of stock transaction known as investing. This is where a stock is bought and held for longer periods of time. This can include long term stock investing as well as retirement accounts. In the United States these investments can benefits from preferential tax treatment, such as long term capital gains or tax differed retirement investments.

3.2 Definitions

Trades can be classified as a “long” trade, meaning that a stock is bought and at a later time sold, or “short” meaning that a share is borrowed and sold, and eventually bought back to be returned to the original owner. “Volume” refers to the number of shares of a stock bought and sold each day. Regular trading hours are from 9:30am to 4:00pm eastern standard time. There are two relatively small fees charged on the sale of all shares of stock, charged by the Financial Industry Regulatory Authority and Securities and Exchange Commission. The sale of stock that was held for longer than a year is subject to the long term capital gains tax rate while profit from the sale of any stock held less than a year is treated as regular income. The “bid/ask spread” can be thought of as the difference between the price someone is willing to pay for a stock and the price someone else is willing to sell if for. “Slippage” occurs when the price you hoped to trade a stock from is less favorable than the price you expected to trade for. This difference can be due to numerous factors including latency, poor order routing, and low liquidity in the market.

3.3 Related Work

There have been numerous attempts to use neural networks to predict the prices in stock markets around the world. Three approaches, all using different techniques and considering different stock markets, all relied on datasets of daily stock prices ranging from two to ten years (Jingyi Shen, 2020) (Cheng-Lung Huang, March 2009) (Kyoung-jae Kim, August 2000).

However one similar study stood out for using price data from the South Korean stock market taken every five minutes. The study relied on five years of data. It was also notable for acknowledging that volume data corresponding to each stock price have been known to carry information about the future price movement. The study however did not include this data as a feature (Eunsuk Chong, 15 November 2015).

4 Methodology

4.1 Feature selection

Stock trades are made by humans and machines. The algorithms machines use to trade may be unknown however any system that is constrained by preprogrammed rules in which the same input results in the same output is inherently predictable. Human behavior on the other hand is governed by intuition and emotion, and while not as constrained, similar stimulus can be expected to produce similar reactions from humans. This makes trades by both humans and machines predictable. The trades being made by humans and machines are not limited to once a day and the data that both use is not limited to only a single data point per day either. Because these trades are occurring continuously and the charts commonly used in trading display historical data by the minute, the data used in this project was collected for each minute of trading during normal hours.

The highest and lowest traded stock price during a single minute, as well as the price at the beginning and end of each minute were selected as features. Also selected for inclusion as a feature was information on the trade volume for each minute.

4.2 Data Collection

Data for this project was sourced from the New York Stock Exchange. The S&P 500 index exchange-traded fund, stock listing ticker symbol SPY, was chosen for consideration in this project because of its high trade volume and portfolio diversification. Consistently high trade volume acts to dampen the effect of large single stock purchases, and diversification prevents an event in any single company or sector of the market from having undue impact on the model. Price data, including the open, close, high and low prices, as well as the volume of shares bought and sold, was collected for every minute during regular trading hours, from 1 December 2020 through 31 November 2021, resulting in 98,205 samples.

4.2 Data Preparation

Examining the data revealed that the volume data was extremely noisy, with single minute spikes as much as 4800% above the mean. The average volume also gradually increased over the course of the dataset. To prevent normalization from drowning out the differences between the samples near the mean several methods to reduce these large volume outliers were attempted. These methods included simple, cumulative and exponential moving averaging and exponential reduction, but ultimately in order to prevent these infrequent and dramatic values from diminishing the rest of the values in the dataset, any sample with a volume value outside of two standard deviations from the mean was removed. Once all the outliers had been removed the volume data was normalized using a moving window of one thousand samples.

The difference between the open and close price of the next minute was calculated. The difference was classified as positive, negative or the same and stored to use as the expected values for training and testing. The unnormalized open price for each minute was also stored for use in assessing the profitability of the model after trading.

The price values also exhibited a trend of increasing value over time. In order to normalize them a moving window of one thousand samples was again used.

The neural network was designed to use the data from a specified number of previous minutes when making its classifications. Each minute consisted of four normalized price values and two normalized volume values, meaning that the number of feature inputs was equal to six features per the number of minutes being considered. The code was written so that the number of minutes to be included could be changed between runs to allow for experimentally determining the optimal number of minutes for inclusion. This meant that for each run of the code, a new dataset was constructed that was as wide as the number of features for that test. After the new dataset was constructed and filled, the data was examined for any discontinuities resulting from previously removed volume samples. These discontinuities were also removed and the result was a new table consisting of sequential, normalized data for the number of minutes being considered.

The parameters for the neural network, including the number of epochs, hidden layers, neurons and activations functions were experimentally varied to determine the optimal result. Testing was conducted on 80% of the dataset with 20% reserved for testing.

Predictions were made using the entire dataset and the neural network parameters that produced the greatest accuracy. These predictions were then simulated as trades using the previously stored, unnormalized price and price difference data. Real world trade costs such as taxes, fees and simulated slippage were applied and the resulting profit was compared to equivalent long-term investing.

5 Results

5.1 Neural Network Parameter Optimization

The neural network architecture was optimized by adjusting each parameter independently and recording the accuracy for each. The first parameter tested was the number of epochs. After observing the training accuracy after each iteration over one thousand epochs, it was detected that the accuracy plateaued by approximately the first one hundred epochs. To prevent over fitting the data, one hundred epochs were used in all the subsequent testing.

The next architecture variation tested was the number of hidden layers. A neural network consisting of only a single hidden layer was tested first. The addition of a second hidden layer was accompanied by a decrease in accuracy with an increase in the time to train. In order to match the number of hidden layers with the complexity of the problem one hidden layer was determined to be the optimal solution.

Table 1: Number of Hidden Layers

Number of Hidden Layers	Training Accuracy	Testing Accuracy	Time to Train	Time to Test
1	95.63	92.53	50.4	0.3
2	91.02	90.92	60.4	0.3

While not directly an architecture parameter, the number of features was proportional to the number of minutes considered. Therefore the number of minutes included was varied from six down to a single minute. Using a single minute the testing and training accuracy was 54.36% and 53.28% respectively. Increasing the input to two minutes saw the largest increase in accuracy. Increasing to three minutes only increased the accuracy by 0.02% and 0.03% on the testing and training data, and any further increase in minutes produced slightly decreased accuracy. To again avoid overfitting two minutes was determined to be the optimal solution, meaning that the input layer to the network would consist of twelve neurons.

Table 2: Number of Minutes Considered

Number of Minutes	Training Accuracy	Testing Accuracy	Time to Train	Time to Test
1	54.36	53.28	57.08	0.3
2	96.24	92.87	66.63	0.3
3	96.26	92.90	59.75	0.2
4	95.68	92.61	52.50	0.3
5	95.63	92.53	50.4	0.3
6	95.94	92.70	46.39	0.3

The size of the hidden layer was tested. Because the number of input features was dependent on the number of minutes considered, the number of hidden neurons was also made dependent on the number of minutes. The parameter was tested by setting the ratio of hidden neurons and input neurons equal to one quarter, one half and one. The one half value produced the greatest accuracy.

Table 1: Number of Hidden Neurons

Number of Hidden Neurons	Training Accuracy	Testing Accuracy	Time to Train	Time to Test
¼ of features	91.04	90.81	49.9	0.3
½ of features	95.63	92.53	50.4	0.3
Equal to features	89.63	89.46	46.4	0.2

Finally the activation function was considered for both the hidden and output layer. The model was tested with every combination of Relu and sigmoid and a hidden layer using Relu with an output layer using sigmoid produced the best result.

Table 1: Activation Function

Hidden Layer	Output Layer	Training Accuracy	Testing Accuracy	Time to Train	Time to Test
Relu	Relu	47.20	46.91	41.3	0.3
Sigmoid	Sigmoid	85.01	84.01	48.4	0.3
Sigmoid	Relu	81.52	81.36	50.5	0.3
Relu	Sigmoid	95.63	92.53	50.4	0.3

The final neural network architecture was one hundred epochs, with twelve features, one hidden layer composed of six neurons and an output layer consisting of three neurons. The output neurons corresponded to future increases in price, decreases in price, and no future change to the price.

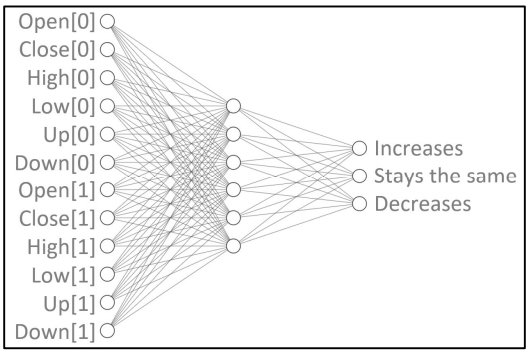

Figure 1: Neural Network Architecture

5.2 Stock Predictions

The practicality of using this optimized model for real world trading applications was evaluated. The trained model was used to make predictions on the entire dataset and the combination of the output data and previously stored price and price difference data was used to simulate trades. Tax fees and slippage were applied, and the result was compared to having simply purchased and held the same amount of stock for the same amount of time.

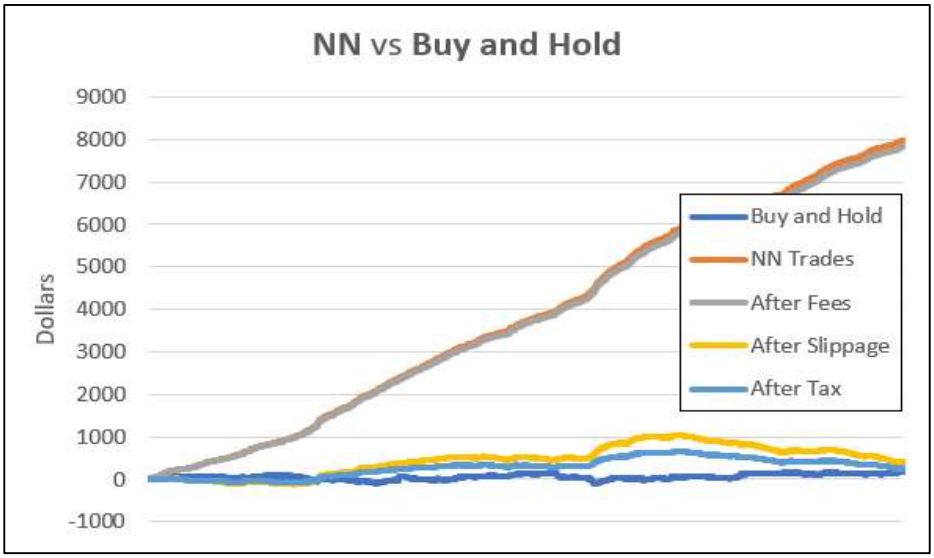

Figure 2: Trading based on neural network model predictions vs purchase and hold investing

In an attempt to increase the profitability of the system the per share profit target for each trade was increased from being anything greater than zero, to anything greater than one cent. The entire system was rerun and the result was a breakdown of the model. The neural network determined that it was more accurate to predict that every sample was below the profit target than it was to predict any actual trades.

7 Conclusions and Future work

This project proposed that a neural network is capable of making accurate stock market predictions using recent stock data. This approach differed from other attempts in that the interval of stock data was a single minute and volume data, in addition to price data, was included as a feature. The optimal parameters were experimentally determined. The results proved that the model was successful. Even using only a single minute of data produced a greater than 50% prediction accuracy. A second minute of data appeared to be the most important set of features, incorporating this data increasing the accuracy to over 90%. The project also differed from similar research in that a real world trading system using this model was simulated. After applying applicable taxes, fees and slippage to the simulation, the result was that a trading system based on this neural network model offers little advantage over the simple purchase and holding of an equivalent amount of stock.

Additional research could be done into the correlation of each feature to the output. Experimentation on dimension reduction such as principle component analysis could eliminate uncorrelated features or redundant features that are highly correlated to each other. Given the noisy nature of the input data and the many truly unpredictable external variables, anomaly detection could help eliminate unhelpful, unpredictable data from the dataset.

References

A. Abhyankar, L. S. (1997). *Uncovering Nonlinear Structure In Real-Time Stock-Market Indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100.* Journal of Business & Economic Statistics.

Cheng-Lung Huang, C.-Y. T. (March 2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications*, 1529-1539.

Eunsuk Chong, C. H. (15 November 2015). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 187-205.

Gibson, G. (1889). *The Stock Markets of London, Paris and New York.* New York: G. P. Putnam's Sons.

Jingyi Shen, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 66.

Kyoung-jae Kim, I. H. (August 2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, 125-132.

Service, C. R. (19 June, 2014). *High-Frequency Trading: Background, Concerns, and Regulatory Developments.* Washington D.C.: Congressional Research Service.

Sewell, M. (20 January 2011). *History of the Efficient Market Hypothesis.* UCL Department of Computer Science.

Code available at GitHub:

https://github.com/SkynetHawkeye/EEL5825_Final_Project/blob/afe9b5d525969c66df243e889a88b43a24cedbde/Forecasting_Movements_in_Stock_Prices.ipynb