Utilizing a Deep Neural Network in the Pursuit of Forecasting the United States Stock Market

Alexander R. Mihalcea

Software Engineering, Eastlake High School

Advanced Projects; *Siddhartha Puri*

Project Collaborators: *Robert Ren, Aidan Suen, Kelyan Wandji, Nathan Headon-Weeks, Eamon Senapati, Angus Tian*

5/30/2024

Background

The Stock Market

The stock market is an aggregation of individuals who buy and sell equity in publicly traded companies. By becoming involved in the stock market, someone may choose to invest some of their capital into a company, often by purchasing a number of shares—which represent a percentage ownership in the company— and in doing so they become a shareholder. The financial performance of the company, whether positive or negative, is subsequently reflected in their own equity. Many types of shares also come with added benefits, from routine payments based on the company's performance—referred to as dividends—to voting rights in major company decisions.

Investors vs Traders

Many people invest some amount of money into a set of companies, and proceed to keep their shares for an extended period of time. These individuals, often referred to as investors, may periodically invest more into their portfolio, but they usually keep their stocks for several years, only selling when they require liquidity or suspect the market is set to decline. More risk tolerant investors may finance young startups—where massive risk is weighed against massive reward—and are referred to as venture capitalists.

Conversely, other individuals, usually called traders, prefer a more fast paced and involved strategy, consisting of selling and buying shares based on the current behavior of the stock in question. The time scale can vary, from those that make trades minute-by-minute, called day traders, to those that buy and sell stocks on the scale of days to weeks, often called swing traders.

Basics of Stock Trading

Traders, whether individuals or those working for an investment firm, rely on several key strategies to predict how the price of a security will change in the near future to make more advantageous trades. These can be broken down into two categories: ***Technical Analysis***, and ***Fundamental Analysis***.

Technical Analysis consists of the appraisal of a security's past prices and numerous technical indicators to ascertain a prediction of its future trajectory. It usually involves an in depth analysis of a stock chart, searching for patterns rooted in the principles of market dynamics, like the balance between buyers and sellers. These can include lines of support and resistance, as well as numerous stock chart patterns which may indicate a reversal in a stock's trend.

Fundamental Analysis, on the other hand, is more about looking at a company as a whole. This usually includes an analysis of its financial statements, its performance against competitors, its anticipated products, and even the market as a whole. Due to its nature, fundamental analysis is less useful in the micro time scales of day trading, but comes into play with investing or swing trades.

Understanding Technical Analysis

Within the domain of technical analysis, stock prices are presumed to follow principles of market dynamics, primarily the relation between buyers and sellers. As such, understanding how this supply/demand relation translates to a stock's value is key to understanding this method of analysis.

To begin, a support level is a hypothetical line drawn on a stock chart, which can be thought of as a floor which the stock has failed to fall below, instead bouncing off it multiple times. The support level represents the point where demand increases to meet supply, and thus the stock price falls no further.

A resistance level is precisely the opposite of a support, acting as a type of ceiling which the stock has bounced off of multiple times, and represents the point where supply increases to match demand, meaning the stock price rises no further.

A stock will tend to bounce between a support and resistance for some time, until eventually it breaks through one of the levels. If, say, it breaks through the resistance “ceiling”, it represents that demand is beating supply. Usually, the stock price will soon fall back to test the resistance level, and if it bounces off the top of it, that means that the stock will likely continue to rise. The opposite is true if a stock breaks through the support “floor”, indicating that it is likely to continue falling.

Many other, more complex stock chart patterns exist, but they all fundamentally boil down to the relation between buyers and sellers. These interactions and the subsequent patterns they create are used by technical analysts to predict when the direction of a stock is set to reverse, as to buy or sell accordingly.

The Potential for Machine Learning in Technical Analysis

Given the complex patterns involved in technical analysis, a sufficiently complex neural network could have the potential to learn these stock patterns—and potentially even more, yet unknown patterns—as to provide a more accurate prediction than that of a human analyst. Such a neural network could, given all the relevant stock data, learn to understand any stock pattern, as well as the relation between different technical indicators and stock behavior, and potentially even lead to new insights into the behavior of the stock market, and how it relates to our current understanding of markets and economics. It could also help to prove or disprove any number of economic hypotheses relating to the stock market, including the Efficient Market Hypothesis (EMH) and the Random Walk Hypothesis. While the use of neural networks in the purview of fundamental analysis—through methods like sentiment analysis—may have some potential viability, the more concrete and mathematical nature of technical analysis makes it a more fitting candidate for the employment of a neural network. This network would likely need to be a deep neural network to have the capability of learning the complex nature of stock behaviors, meaning substantial data and computational power would likely be paramount to the training of such a network.

Project and Development Overview

Project Goal

We sought out to create a program that offers a user a forecast of a desired stock, as to allow for more advantageous trades. The core idea of the project was a dual faceted approach, where an advanced technical analysis system is combined with a neural network to provide a more accurate prediction of a desired stock. This writer focused on the machine learning aspect of the program. The original plan was to train a neural network on historic data from every company in the S&P 500, by providing segments of 100 stock values as an input, and having the model predict the 100 stock values that would follow.

Overview of Miscellaneous Data Systems

To obtain and utilize relevant training data, I [*Mihalcea*] wrote several code systems. I began by scraping the last ~700 days worth of hourly stock values for every company in the S&P 500. For each company, this equated to roughly 3000 stock values, the final 500 of which are spliced out to be used as validation data. I converted the data into a separate list for every type of stock value saved (open, close, high, low). After this, I used a moving window approach to split the data into two subsequent list-of-lists, where one contains all 100 sequential stock value sections, and the second contains the corresponding following 100 sequential stock value sections. Essentially, I ended up with two lists, where for each index **i**, the first list at **i** contains 100 stock values, and the second list at **i** contains the 100 stock values that followed [See **fig 3** for illustration]. I also wrote systems that normalize each company's stock data as to be between 0 and 1, in an attempt to remove the effect value magnitude would have on the model. After all of this data manipulation, I ended up with roughly 1.6 million 100-stock-sets, around 1.4 million for training and 200,000 for validation.

First Attempt: Utilizing an FNN

I firstly attempted to make progress utilizing a Forward Feed Neural Network (FNN), which had multiple hidden layers, would take 100 stock close values as inputs, and would output the predicted next value. This value would be appended to the end of the input list, and the process would repeat until 100 total future values were predicted.

Over the course of several weeks, I attempted countless variations regarding model architecture and training settings, but was unable to get the model to predict even its own training data. Eventually there was nothing else left to try, and I decided to scrap this approach.

Second Attempt: Utilizing an LSTM

I decided the best approach to try next would be shifting over to a Long Short Term Memory (LSTM) neural network. I changed the input to include high, low, and open values, in addition to close values. I also discarded the one value at a time looping prediction system, and instead had the model predict all 100 future close values at once. The model had multiple hidden layers, and unlike the previous FNN model, took significant amounts of time to train—an issue which becomes relevant later—meaning I had to train on a small subset of the available training data, usually 100,000 stock sets. However, once trained, this model showed great promise in predicting its training data with a decent amount of accuracy and consistency [**fig 1**], in contrast to the previous FNN model. However when expanded to new data outside its training set, this model showed little to no predictive capability [**fig 2**]. Attempts to change the model to improve predictive capability failed, and increasing the amount of training data was impracticable, as the training time for all 1.4 million training sets would likely be on the order of weeks. Several attempts at optimizing system resource usage to improve training time failed.

Being suspicious of overfitting being the culprit, I modified the model's training, changing it from a flat 100 epochs to a dynamic system, which checks the models accuracy on validation data after each epoch, and ends training once the model's accuracy on validation data begins declining. Essentially, this makes the model train until it begins overfitting.

However this new training system revealed a startling problem: The LSTM model would begin overfitting almost immediately, often after the first few epochs. Increasing the training set to 250,000 failed to improve this issue. This essentially meant that the model, despite its ability to predict training data well, was unable to learn how to predict new data whatsoever.

After significant research and subsequent attempts at solving the issue, I came to the conclusion that at least one of the following three explanations are responsible:

(1) My relative inexperience in the field of machine learning is causing me to be unable to properly construct and train the model;

(2) My lack of computational power making me unable to use a larger training set is limiting the models ability to learn, at least with its current architecture;

(3) The "Efficient Market" hypothesis[1] is in fact an accurate description of the stock market, making the goal of a neural network which is able to learn stock market patterns to predict future behavior truly impossible.

If either option **(2)** and or **(3)** was the cause of the issue, there would be nothing else I could do, and potentially the project as a whole would be untenable.

I brought up my concern regarding the machine learning portion of this project likely being a deadend. The project team collectively came to the conclusion that the best option would be to assign more personnel to work on machine learning, in the hopes of a new idea or approach bearing fruit. I remained involved with the machine learning team, providing insight into approaches I've attempted, but took a mostly backseat approach to the development process, focusing instead on academic research and documentation—including this paper. Ren, Wandji, and Suen each began working on their own attempts at a neural network. Mr. Ren decided to work on his own LSTM model, separate from my attempt. Mr. Suen began working on a Generative Adversarial Network (GAN), a model type which showed promise in predicting stock markets in several papers I reviewed. Mr. Wandji worked on his own RNN, as well as a Gated Recurrent Unit (GRU) model.

(TO BE CONTINUED)

---

[1] The efficient market hypothesis postulates that a security's value is always representative of all available information. Thus it is impossible to predict a stock's future behavior, as it will be the result of information which does not yet exist. There is empirical evidence both supporting and disputing this hypothesis, and a clear consensus has yet to be reached.

Analysis and Retrospective

Existing literature

Much literature exists on the very problem we attempted to solve with this project, most of which report positive results in predicting the stock market using their models. For example, a paper by Staffini [2] reported impressive predictive capabilities using a Generative Adversarial Network (GAN), while a paper by Noel [3] showed promising results using a Non-linear Autoregressive with Exogenous Input (NARX) model. However it is important to consider that on multiple occasions seemingly impressive machine learning papers get published, only for crucial errors in the testing methodology to be later discovered.

For example, consider a neural network trained to accurately classify if an image of a skin spot is melanoma or not. In this example, numerous non-obvious mistakes can lead to the model having a near 100% accuracy on the testing data, while being useless in real life. Perhaps the model learned that the images where a ruler is present nearly always are of melanoma, or that the meta-data of the images contained information regarding their classification, or different cameras or lighting conditions were used depending on the type of skin spot. In any of these cases, the model would be nearly useless at predicting any new data, as it learned to, in effect, cheat.

While I do not have the technical knowledge to ascertain whether that is the case in the literature seeking to predict the stock market, I also cannot rule it out, especially considering the astronomical amount of variables which a neural network could potentially exploit during training, as well as the suspiciously overfit prediction curves that nearly all the papers I received presented [See **fig 4**]. No paper I could find actually had their models make real trades in the market, and to my knowledge no such paper exists.

---

[2] Staffini, A. (2022). https://doi.org/10.3389/frai.2022.837596
[3] Noel, D. (2023, June 18). https://doi.org/10.48550/arXiv.2306.12969

On Predicting the Stock Market

Our project was heavily constrained by numerous factors, including experience, computing power, and funding. However, I am unconvinced that these were the sole reasons for our inability to accomplish our goal. Our project, at its core, was about predicting a stock's value based on its past behavior. Short term behaviors of a stock tend to follow patterns, often representing a supply/demand balance between buyers and sellers. This leads to many known stock chart patterns, which can occasionally hint towards what a stock's future behavior will vaguely be. However, the stock market, just like any free economy, is ultimately just an aggregation of human decisions. Thus, during any trade, it is less about you as an agent predicting the behavior of an independent phenomenon, but rather you predicting the behavior of other agents, each of which is also attempting to predict the behavior of the others. All of these variables make it seem an untenable goal to predict, with any meaningful accuracy, how a securities value will fluctuate based on previous values.

Ultimately, for any neural network claimed to have predictive capabilities of the stock market, the true litmus test is not its accuracy on testing data, but its ability, in the real world, to consistently make advantageous trades better than random chance. If such a system exists, or were to be developed, it would be comparable to a money printer, and whichever investment firm attains said system would have astronomical profits. However, given the zero-sum nature of stock trading, such an advantage would be short lived, as competing firms would invest astronomical resources into their own neural networks to avoid imminent extinction. Once numerous accurate neural networks are involved in the market, they will likely cease to be useful, as the market they were trained to predict—one run by human decisions—has fundamentally changed. Each of these neural networks would now need to predict what all the others will predict, a goal which quickly descends into an infinite regression, in some ways similar to the famous “three-body” physics problem. **These reasons, in my opinion, preclude the possibility of a neural network which can accurately and consistently “beat the market”.**

Figures

Fig 1

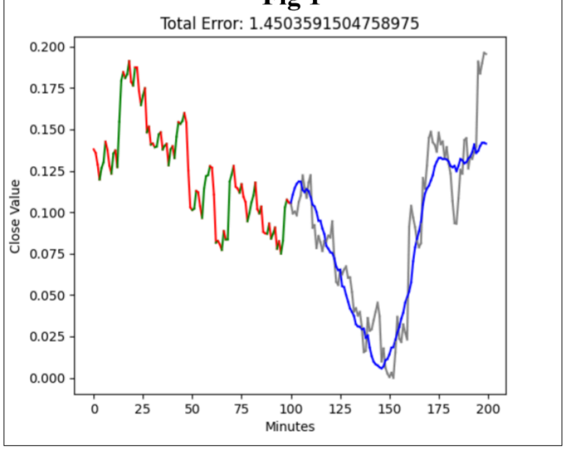


Fig 2

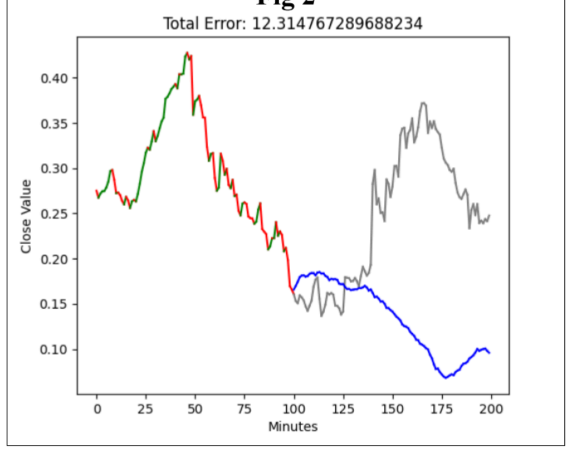
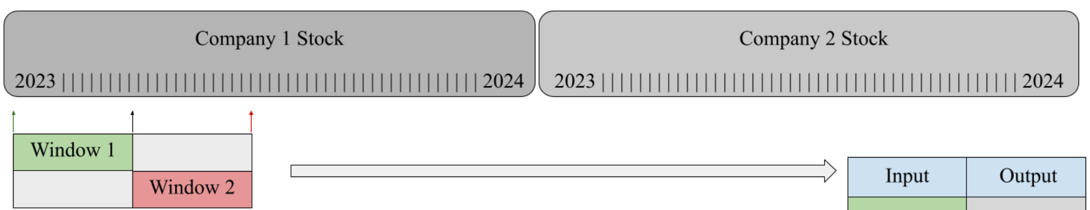


Fig 1, **Fig 2**: Our LSTM model, fig 1 = training data, fig 2 = testing data. **red**/**green** = provided data; grey = real data; **blue** = predicted

---

Fig 3



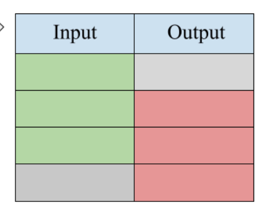


Fig 3: Illustration of moving window data splitting approach. Window 1 and 2 move together, each being appended to corresponding training list.

Fig 4

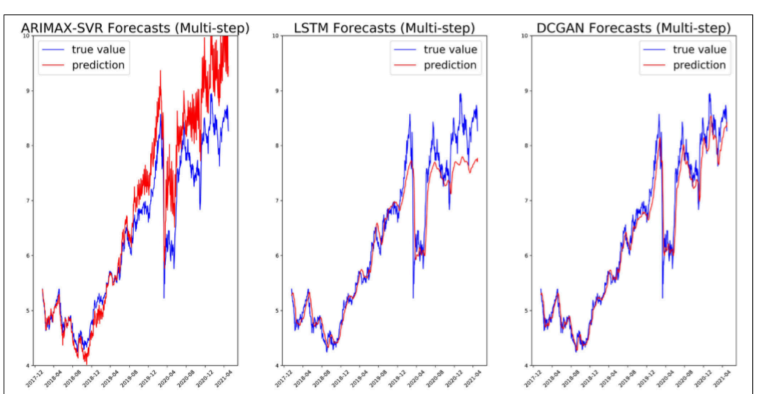


FIGURE 6 | Multi-step (5 days) forecasts on the ENEL test set for ARIMAX-SVR **(left)**, LSTM **(center)**, and our GAN architecture **(right)**.

Fig 4: Figures in paper by Staffini 2022, showing their models predictions on testing data. Seemingly overfit.

Citation:
Figure 6;
Staffini A (2022) Stock Price Forecasting by a Deep Convolutional Generative Adversarial Network. Front. Artif. Intell. 5:837596. doi: 10.3389/frai.2022.837596