Exploration of Single Stock RL Trading

Jeff Keith
Computer Science Dept
Ryerson University
Toronto, Canada
jkeith@ryerson.ca

Ian MacPherson
Computer Science Dept
Ryerson University
Toronto, Canada
ian.macpherson@ryerson.ca

Bretton Tan
Computer Science Dept
Ryerson University
Toronto, Canada
bretton.tan@ryerson

Paul Messina
Computer Science Dept
Ryerson University
Toronto, Canada
paul.messina@ryerson.ca

***Abstract*—The goal of this paper is to explore the performance of various reinforcement learning (RL) agents in a stock trading problem domain. We investigate single stock trading strategies and perform ablation studies by testing the network against stocks other than the one that it is originally trained on. We train on the Microsoft stock (NASDAQ:MSFT), and test the agent against Apple (NASDAQ:AAPL), Macy's (NYSE:M), GameStop (NYSE:GME), and the Fidelity S&P index mutual fund (MUTF:FXAIX). As expected, the network achieves good results when tested on other stocks that are more similar in industry, or can be considered to have some form of linear correlation. When tested against stocks with higher volatility, the agent has a more difficult time making decisions on whether to buy or sell the stock at a given time and price. We discuss what may be the cause of such phenomena and consider the possible causes of the variance in performance across differing stocks.[1]**

I. Introduction

Predicting the prices of stock is considered a very hard problem and machine learning methods have been used for many years to assist with stock trading by predicting stock prices and performing asset management. It has not been until recently that reinforcement learning techniques have been applied to this domain. The role of the stock market cannot be underestimated in the global financial system and being able to predict its movements has been a long sought after goal. Currently, Artificial Neural Networks (ANNs) have been used to predict stock prices more effectively than models which rely solely on traditional stock indicators like the Sharpe ratio. That being said ANNs have an unlying weakness because they are reliant on supervised data which is costly to prepare. As well, the network may fit to a data distribution that represents a bull market and not perform well during times of market volatility. The challenge for supervised learning models is that delayed rewards for long term goal seeking are difficult to represent and that they can only suggest actions not take them[1].

Reinforcement learning methods are structured around the Markov Decision Process (MDP) which is a mathematical technique of simplifying a decision making process where the future outcomes can be predicted based solely on the present state of the system. Stocks are often traded simply on the current market prices which means that they exhibit the markov property and so reinforcement learning algorithms can therefore be applied to the problem. There are many advanced RL systems that have been developed but to keep our investigation simple we chose to focus on trading of single stocks with two RL agents. We chose to train these agents on a range of stocks for a single stock strategy and compare their performance at run time when trading single stocks.

II. Related Work

Before we speak to the related research in general terms, we cannot move forward in this report without mentioning that large amounts of research in this area go unpublished. It is highly unlikely that a fund that employs RL techniques will release such strategies to the public. As well, it is in the best interest of these firms to keep their innovations a secret to obtain more funding to increase their assets under management (AUM) as increasing this number will increase the amount of money they are able to algorithmically trade and in turn ideally increasing overall returns. If such funds were to release their strategies, they may be liable if retail investors with limited expertise end up losing their life's savings.

While the research published may not come directly from those who implement reinforcement learning algorithms for stock trading in practice, many research groups have published papers which adapt different reinforcement learning strategies for stock trading. Such strategies have been discussed in [2][3][4][5][6]. Some interesting techniques include single stock trading, where an agent is trained to buy and sell a single stock for a profit. When training in this way, the agent's strategy is dependent completely on the patterns of the pre-chosen stock and may or may not generalize well to others. Other strategies aim to optimize for portfolio management which incorporates historical data from a range of stocks and potentially varying industries determining how to structure a portfolio of investments.

Related strategies employ traditional reinforcement learning techniques along with deep learning techniques to achieve state of the art results. Generally each of these works by training the agent on historical data of a specific stock, or group of stocks. Some may attempt to forecast the long term

[1] https://github.com/jkeithry/RL_Stock_Trading

prices of stocks, or engage in high-frequency stock trading. The size of the dataset is determined by how far back in time the reinforcement learning practitioner decides to train the agent.

High-frequency stock trading is characterized by trading stock in companies at a speed only achievable by computers. It can also be determined by the rate at which it turns over the equities in the portfolio. This is in contrast to a strategy commonly known as value investing, where an investor invests in a company based on its underlying value and holds on to the stock for a long period of time. Reinforcement learning techniques are implemented to exploit the underlying data movements of a stock or group of stocks for this purpose to be able to make high-stakes decisions and execute on these faster than a human may be able to understand the strategy being executed at a given time.

Deep reinforcement learning (DRL) has been shown to perform well on the task of algorithmic trading and some hand-picked strategies have been compiled in the FinRL library [7]. This work uses the methods compiled in the FinRL library to test various hypotheses and determine how profitable reinforcement learning solutions can be across various situations. The concepts described in this section will be tested against each other and investigated in further detail throughout this report.

III. Methods

All of our stock market data was obtained through the Yahoo Finance API and the stockstats python library. The stock trading model was trained on Microsoft stock data from January 1, 2009 to December 31, 2018. This timeline is chosen arbitrarily and may capture the bull market that came about after the 2008 financial crisis. The model was then tested on various stocks depending on the experiment. These stocks include:

- Microsoft (NASDAQ: MSFT)
- Macy's (NYSE:M)
- GameStop (NYSE:GME)
- Apple (NASDAQ:AAPL)

and the Fidelity® 500 Index Fund (MUTF:FXAIX) which tracks the S&P 500. Regardless of the stock being used in our experiments, we chose to run our experiments on stocks prices which range from January 1, 2019 to March 31, 2021 to test if the strategies learned by the models, whose training sets included the period after the 2008 financial crisis, were robust to the volatility inflicted in the early stages of the COVID-19 health crisis. One stock trading model is based on the soft actor critic (SAC) model. We also experimented with a Deep Deterministic Policy Gradient (DDPG) agent. The following hyper parameters were used in the training of the model: batch size of 64, buffer size of 100000, learning rate of 0.0001, learning start point of 100 steps, and an entropy regularization coefficient of 0.1.

IV. Experiments

In the first experiment, we tested our models on Microsoft stock. This was chosen as a high-growth technology stock which performs well in a sector with high levels of innovation. Since the model was trained on Microsoft stock, this test develops a baseline expectation of how our model will behave, which we can use to compare with the results of our other experiments.

Due to Microsoft's scale as a company, we hypothesize that the movements in this stock may have a wider effect or correlation to the tech sector, or stock market as a whole. In the second experiment, we test this hypothesis by comparing the performance of the agents trained on Microsoft stock to similar stocks to determine if the agents can perform well on stocks that may be correlated.

Apple and the Fidelity S&P 500 index fund were chosen as similar to Microsoft. Fidelity was chosen because the growth of the Microsoft stock may be considered to be loosely correlated to the overall growth of the S&P 500. Additionally, Apple is in the same industry. Due to these reasons, we surmise that the agents will be able to perform similarly on AAPL and FXAIX after being trained on Microsoft stock.

With the same hyperparameters optimized when training on the Microsoft stock, the second experiment aims to test the performance on a much more volatile stock from a differing industry. This is used to determine how accurately the network can generalize, or how the behaviour of the network might change with stocks of higher volatility. For this experiment, Macy's and GameStop stock are used to test the performance of the network.

V. Results

In the first experiment, we tested the model against Microsoft. the model bought stock for the first 89 days, proceeded to sell some stock on day 90, and then held onto its remaining stock for the rest of the run. As seen in Fig. 1. the account value of the model across the duration of the run. Given that Microsoft stock continues to climb over time, the value of the account given the actions taken by the model result in an increased account value by the end of the run.

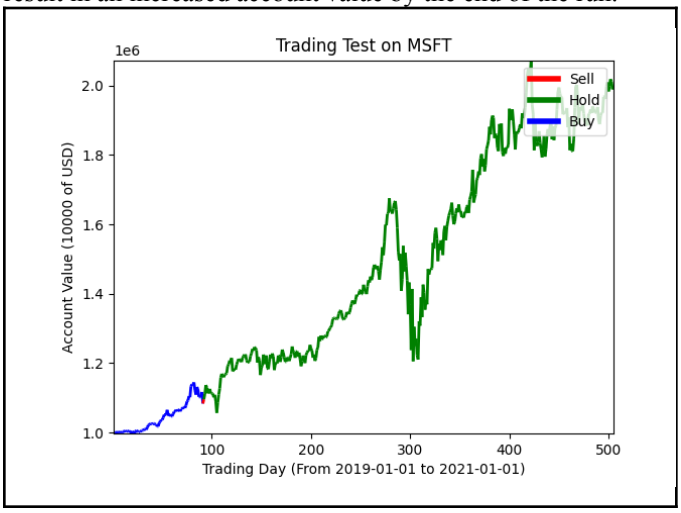

Fig. 1 Testing on MSFT.

In the second experiment, we test to see if the model can act in a positive way on a similar stock, in this case Apple (Fig. 2). The model buys stock for just over 200 days and then proceeds to hold onto its stock until the end of the run. Similar to Microsoft stock, the Apple stock tends to only increase over the range of time in the test set. This consistent increase paired with the purchasing and holding actions of the model result in a final account value that is significantly higher than the starting value, which we consider to be a success for our model.

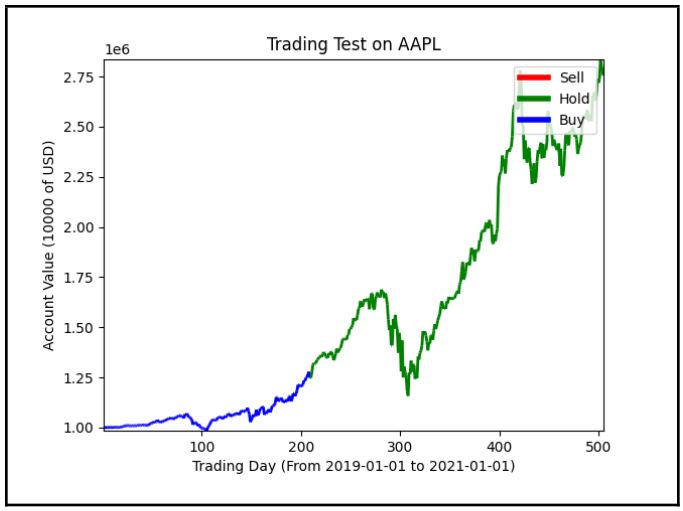

Fig 2. Testing on AAPL.

Continuing our second experiment, we then tested the model against the Fidelity S&P 500 index fund (Fig. 3). Interestingly, this stock performed the most similar to the microsoft stock out of all the experiments. The model bought stock for 106 days and then held onto its stocks until the end of the run. Looking at the behaviour of the stock and the actions taken by the model, we found that the total account value was higher by the end of the run, despite the significant dip that this stock took around day 300.

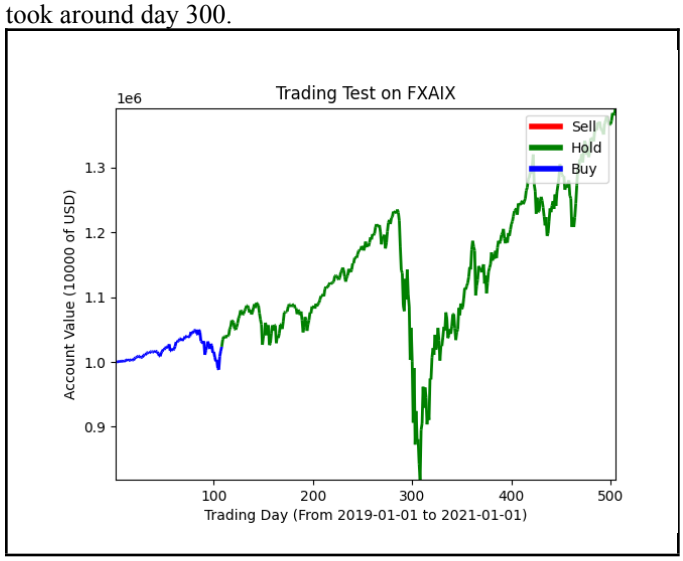

Fig 3. Testing on FXAIX.

The third experiment looks to examine how well the model will perform when tested on stock that is significantly different from the Microsoft stock it was trained on. The first stock chosen for this was Macy's (Fig. 4.). When looking at the model's actions, we found that the model consistently bought stock throughout the entire run. Given the high instability of Macy's stock, the model seemed to have a difficult time increasing the total account value by the end of the run. This reinforces the idea that the model might not be good at generalizing what it has learned to stocks that are significantly different. This may be due to the network learning a data distribution in a time of upward trends in the market that is dissimilar to the volatility encountered in more uncertain times.

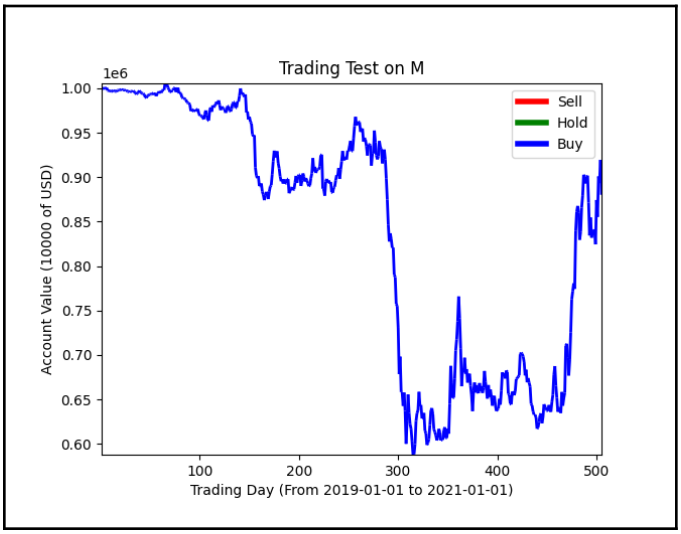

Fig 4. Macy's results.

We thought we would test the model on GameStop stock given that it has had a large increase in value over the past few months. We decided the models would probably have a difficult time trying to apply the actions it learned to data that is so noisy. As seen in Fig. 5. the model continues to buy stock throughout the entire run. Although the actions taken are exactly the same as experiment 3, due to the rise in popularity and the value of GameStop stock, the model was able to significantly increase its total account value by the end of the run.

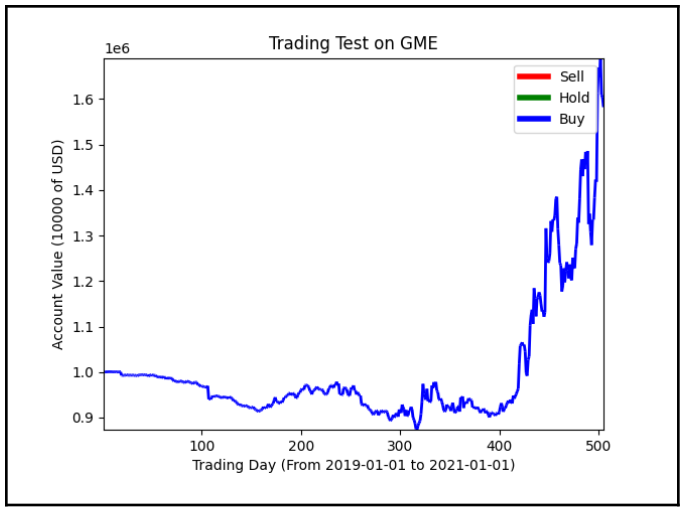

Fig 5. GameStop results.

Finally, the above figures reflect the results from our SAC model that was trained on MSFT. We also experimented with a DDGP model which resulted in almost identical results.

VI. Discussion

We found that the testing data includes a drop in account value about halfway through every run which coincides with the major events of the COVID19 pandemic. Our determination is that the stock market was greatly affected by these events, which caused a 30% drop in the S&P 500, leading to similar losses for our agents which were unable to fit to these damatric market changes even though they were trained during the 2008 financial crisis [8].

Looking at the results in Fig.1 and Fig 2, we see that the models were in fact able to increase the total account value by the end of the run. The models were able to buy stocks when they were cheaper and then hold onto them when they were more valuable thus increasing the account value. This further supports our hypothesis that the models should be able to successfully perform on stocks similar to the stock they were trained on. This may serve to prove that the agents do in fact learn an optimal strategy on the given stocks.

However, in our third experiment, with Macy's and GameStop (Fig 4. and Fig 5.), there are some mixed results. Testing on Macy's stock showed the model buying throughout the whole run and ending with a lower account value than what it started with. This is in contrast to the tests on GameStop stock which saw the model still buying throughout the whole run but ended with an account value much higher than the starting. What we can determine from these results is that the model has a difficult time deciding which actions to take when dealing with a stock that behaves significantly differently from what the model was trained on. We believe this is why the model only buys throughout the entire run in both tests. Our working hypothesis as to why the GameStop test ended with an increase to total account value is due to the large increase in value of GameStop stocks right before the end of the run. Based on performance from the Macy's test, it seems the model would have continued to buy no matter what was occurring with the stock. This lends to the idea that the model just managed to get lucky that all the stock it bought became significantly more valuable. In the future, we could perform more tests on stocks similar to Macy's and GameStop to see if the model continues to perform in the way we currently expect.

VII. Conclusions

From the experiments run to test the network, it seems that the network in fact learns some sort of strategy to make a profit when trained on Microsoft. In the cases of the test runs on Apple, FXAIX, and the test set from Microsoft, the network is able to nearly double the initial investment. On first glance this would lead one to believe that the agent is learning the most optimal strategy to turn a profit on the stock market.

However, we are not entirely convinced that this is the most optimal solution as the network seems to buy at the beginning, and hold the stock throughout the rest of training. Our suspicions arise due to the fact that the network does not buy or sell when the Microsoft, Apple, or Fidelity stock sees a sharp drop around the beginning of the COVID-19 health crisis. This could be due to the fact that the time span the agent was trained on generally saw an upward trend. However, these results give rise to confusion in comparison to Macy's or Gamestop stock. In these instances, the agent consistently buys throughout the volatility.

When considering these results it is also important to point to the fact that if the algorithms are learning something, it may not fall in line with our intuition - buy low, buy when others are selling, and sell when others are buying. This might be likened to optimal path-finding algorithms which may be optimal, but present themselves as counter-intuitive. With the upward trending market leading up to 2020, perhaps the network learned that over this span of time, selling is not as profitable as buying consistently and holding on to the shares. If any time in the stock's history is a good time to buy then always buying at the start, buying through more extreme downward or upward trends may be just as profitable of a strategy.

To continue to test the performance of the network, we suggest that for additional work, the network is trained on different market conditions such as a downward or bear market where perhaps most strategies result in a loss of profits. As well, it may be beneficial to collect specific snapshots in the history of a company's stock price where increased volatility is encountered. This would be done to increase the number of training examples for the network during volatile market conditions.

References

[1] Jae Won Lee, "Stock price prediction using reinforcement learning," ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), Pusan, Korea (South), 2001, pp. 690-695 vol.1, doi: 10.1109/ISIE.2001.931880.

[2] Thibaut Théate and Damien Ernst (2020). An Application of Deep Reinforcement Learning to Algorithmic Trading. CoRR, abs/2004.06627.

[3] Xiao-Yang Liu and Hongyang Yang and Qian Chen and Runjia Zhang and Liuqing Yang and Bowen Xiao and Christina Dan Wang (2020). FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. CoRR, abs/2011.09607.

[4] Mehran Taghian and Ahmad Asadi and Reza Safabakhsh (2021). A Reinforcement Learning Based Encoder-Decoder Framework for Learning Stock Trading Rules. CoRR, abs/2101.03867.

[5] Antonio Briola and Jeremy D. Turiel and Riccardo Marcaccioli and Tomaso Aste (2021). Deep Reinforcement Learning for Active High Frequency Trading. CoRR, abs/2101.07107.

[6] Evgeny Ponomarev and Ivan V. Oseledets and Andrzej Cichocki (2020). Using Reinforcement Learning in the Algorithmic Trading Problem. CoRR, abs/2002.11523.

[7] AI4Finance-LLC, “AI4Finance-LLC/FinRL,” *FinRL: A Deep Reinforcement Learning Library for Quantitative Finance*. [Online]. Available: https://github.com/AI4Finance-LLC/FinRL. [Accessed: 19-Apr-2021].

[8] F. Imbert and Y. In, “Dow drops 1,300 Points, S&P 500 loses 5% As coronavirus market SELL-OFF reaches new low,” 18-Mar-2020. [Online]. Available: https://www.cnbc.com/2020/03/17/stock-futures-fall-slightly-after-market-rebounds-on-hopes-for-1-trillion-stimulus.html. [Accessed: 19-Apr-2021].