CSCI357 Final Report

Chengcheng Ding, Andrew Whitig, Jacky Lin, Hanzheng Wang

Department of Computer Science,

Bucknell University,

Lewisburg, PA 17837

CSCI357 Final Report

Introduction

The United States stock market is a large and complicated process which in some way impacts not only the daily lives of more than 300 million Americans but also the world economy. The American stock exchange is a market where shares of ownership in companies are bought and sold. It is a system which exemplifies the United States' capitalist economic policies and the free market of trade on which those policies sustain. For this reason, the US stock market and the role of AI in the sock market has become the focus of our inquiry. Throughout this paper we will gain a greater understanding of the role that AI can play in shaping the stock market and general US economic policy. This includes discussion of the ethical considerations of AI in the stock market and the decision making power that AI holds. To that end, we will explore the methods of predicting stock prices and determining when and how much of a stock to purchase at any given moment.

Previous Work

The applications of accurate stock price prediction are alluring. One can imagine putting one's feet up and running a program which successfully executes trades and accumulates massive wealth without lifting a finger. Unfortunately this fantasy is unlikely to ever become a reality. Stock prices are volatile and at times subjective making them extremely difficult to predict. For this reason, a wide variety in approaches and perspectives have been taken in the pursuit of AI stock price prediction. In this section we will provide an overview of the most common approaches and a survey into the current state of stock price prediction. While not exhaustive, they will paint a picture of how price prediction is performed and to what degree of success it is performed.

One type of model that has been used to specifically predict the price of the S&P 500 index is hybrid system which combines a heuristic approach with a neural network predictor [4]. The S&P 500 index is a collection of the largest 500 which can serve as an approximation for the overall stock market. The heuristic portion of the agent serves to provide examples to a neural network which predicts the price of the index. The rules based system focuses on expert predictions about the futures value of the index. A

futures contract is a contract made now to have the option to purchase/sell some amount of a stock at a predetermined price in the future. Expert future contracts inherently hold their predictions about the direction of a stocks movement. From these futures predictions and 10 technical indicators, a series of rules are constructed. If a particular combination of indicators have occurred, a rule is activated and the neural network predicts only on similar cases historically.In addition to the rules guided neural network, the results from a four neural network voting combination are used to identify the long term trends of the futures values. This AI system ended up being slightly above 50% accurate on the four year period it was trained on from 1989 to 1993.

Another model to predict the future price of stocks relies on machine learning, just as our purchasing model does [3]. This model treats the updating price of a stock as a continually updating set of states. This aligns well with a reinforcement model which experiences states continually and updates based on its previous accuracy. –insert name of person– designs a reinforcement learning stock prediction which follows this system. The system learns a policy which determines what set of actions to perform from a specific state. So for each state s_t where t is the current time, the action a_t follows as a function of the state. In this way, the model trains the function by updating how far the current functions result is from the optimal result. In –inert name's– model, a score for reinforcement is determined by not only evaluating the difference from the most recent action but also the all previous differences. The model uses a coefficient γ which takes value between zero and then for the function $\sum_{i=1}^{n} \gamma^{n-i} r_i$ where r_i is the reward of action a_i, the older rewards will eventually trend to zero and eliminate the effect of older rewards on the current decisions. From these rewards the new state is acquired using a bootstrapping process similar to dynamic programming, which is a process similar to dynamic programming where states are stepped through one at a time until the one with the most benefit is located. This state then becomes the agents next prediction. The states s_t consist of several indicators. They include the opening price, high price, low price, and closing price for a day as well as the number of share traded during the day, or volume. These states are identical to the indicators which can be seen later in our implementation and are basic metrics. Finally, since these discrete states are unlikely to ever repeat, a neural network is used to approximate and generalize the reinforcement learning process. This model worked somewhat well but

ultimately struggled because it is unable to adjust to extreme positive and negative values and does not perform well when using shortened or lengthened time intervals.

Model

Our approach to modelling the stock market uses a network of agents interacting to simulate the stock market. In our system, we have created an environment for one or more agents to interact with a supply of stocks in an overall marketplace. Each of these agents consists of a two AI models in sequence. The first model in the sequence is a neural network. This network is a LSTM network and is used to predict the future value of a stock. The second model in the sequence is a Q-Learning model. This model is used to determine the appropriate action to take at any given moment in time. With these two models, our agents are able to work autonomously to evaluate and purchase stocks.

Stock Prices Prediction

The first AI model in our approach to stock price prediction is a LSTM-based deep recurrent neural network (DRNN) Architecture. The LSTM layer is able to handle not only a single data point each time, but also a sequence of data. The input of each node in LSTM layer includes both the current input data to the node as well as the output from previous node [1] (See Fig. 1). In this way, the LSTM layer is able to connect prior information to the present task.

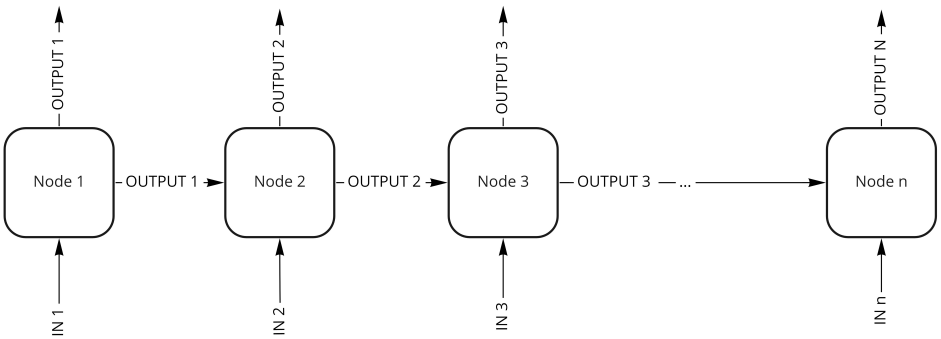


Figure 1. Structure of LSTM layer

Before making decisions, such structures help the neural network to include all the information. Given our task to predict prices of stock market, our assumption is that there are some hidden connections between the current price and the historical prices.

To retain such a historical connection and given the characteristics of LSTM layer, we use this structure to predict the future price. The stacked LSTM architecture we use consists of multiple layer of LSTMs (See Fig. 2). After each layer of LSTM, we add a dropout layer. Such layers prevent neural network from over-fitting by randomly dropping out some nodes from neural network at each step. Thus, during each training process, only the remaining node in each layer is trained.

Figure 2. Structure of LSTM layer

To train this neural network, we use real stock price information from Yahoo! Finance API. This data set contains historical information about the performance of a stock including volume and indicator variables of the daily closing cost, high cost, and low cost (See Fig. 3). We will normalize the data using robust scaler. Since any outliers in data may affect the accuracy of the prediction, we try to reduce such effects. The robust scaler has features using statistics that are robust to outliers. It removes the median and scales the data according to the interquartile range (IQR). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).

Stock Trading System

The second system in our model is stock trading system, where we have a virtual environment that simulates the real stock market by implementing it using real stock prices and a agent implementing using q-learning method.

Q-learning is a type of reinforcement learning. Within this method, agent learns to provide a policy that maximize the total reward. During the training process, the agent updates the Q value for each action given a specific state. The equation we use for update the Q value for each action in any given state is given by [5]

$$Q_{n+1}(s,a) = Q_n(s,a) + \alpha \times (r_n + \gamma \times \max\{Q_n(s') - Q_n(s,a,t)\} \tag{1}$$

where (1) s represent current state; (2) a represent the current action; (3) $Q_{n+1}(s,a)$ is the updated Q-value for an action a given a state s; (4) $Q_n(s,a)$ is the current Q-value for an action a given a state s; (5) α is our learning rate. The higher the α, the faster

	Date	Open	High	Low	Close	Adj Close	Volume
0	1980-12-12	0.128348	0.128906	0.128348	0.128348	0.100922	469033600
1	1980-12-15	0.122210	0.122210	0.121652	0.121652	0.095657	175884800
2	1980-12-16	0.113281	0.113281	0.112723	0.112723	0.088636	105728000
3	1980-12-17	0.115513	0.116071	0.115513	0.115513	0.090830	86441600
4	1980-12-18	0.118862	0.119420	0.118862	0.118862	0.093463	73449600

Figure 3. historical Price of Apple Inc.

we learn; (6) γ is our discount factor that controls how much we take the utility of our next state into account; (7) $\max\{Q_n(s') - Q_n(s, a, t)\}$ allows us to consider the best possible case of using an action a given a state s.

In this paper, we use Q-learning to train an agent so that the agent is able to buy, sell, or hold stock at appropriate time. here, the state for each time consists of recent prices of stock. The action space is a list of floating numbers. These numbers represent the percentage of maximum possible shares agent could buy or sell. A positive percentage represents buying some shares at current price, a negative percentage represents selling some shares of current holding shares at current price, and the zero percentage represents no action and holds all shares. For each iteration, our agent picks one actions from this list of percentages and execute them. Then, we will calculate Q-value for the action providing with the reward earned and input state, using the Eq. 1.

General Model

Generally, the combination of two of our models can be concluded as following (See Fig. 4) our agent obtain recent prices from stock environment. Then agent uses stock price prediction system to do a price prediction on the prices of future several days. Next, the predicted result is treated as input into stock trading system, where agent operate selling and buying actions accordingly.

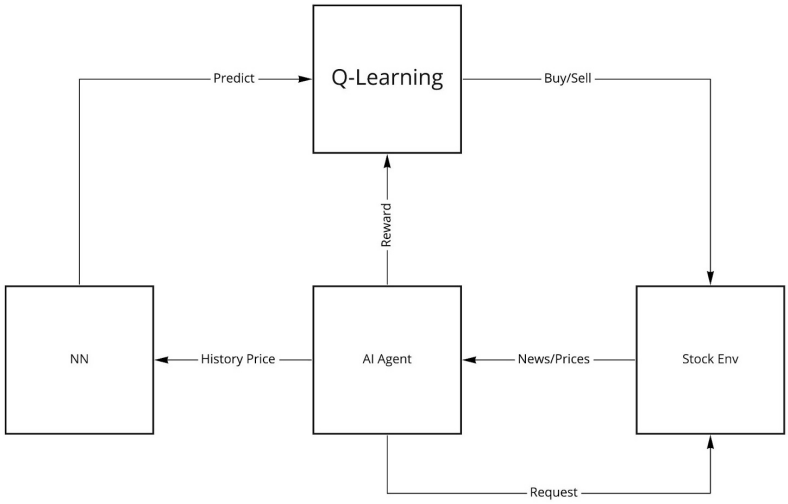


Figure 4. General Design of AI Agent

Result

Prediction Performance

huber loss function is used in this paper. huber function is less sensitive to outliers than mean square error. The huber loss function describes the penalty incurred and defines the loss function piece-wise by [2]:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases} \tag{2}$$

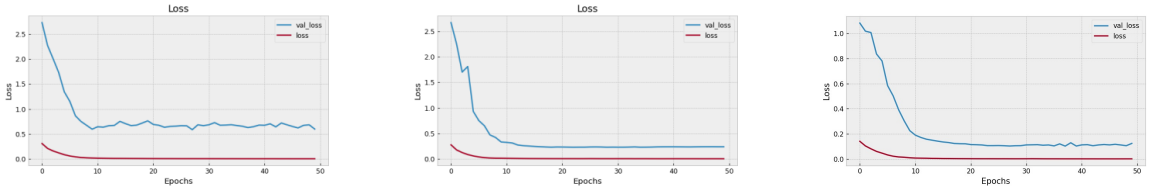
The reason we use this loss function is due to the fluctuations in stock price. As shown in Fig. 6, the red line indicates the actual stock price, which has many tiny fluctuations from day to day. Huber loss function helps neural network to ignore tiny fluctuations and to pay more attention on general trend of prices. The training loss and validation loss is shown in Fig. 8. Fig. 5a shows losses as a function of epochs measuring with mean square error and using shallow LSTM structure (using one or two layer of lstm). Fig. 5b shows losses as a function of epochs measuring with mean square error and using stacked LSTM structure (using 5 layers of lstm). Fig. 5c shows losses as a function of epochs measuring with Huber loss and using stacked LSTM structure (or DRNN) (using 5 layers of lstm).

In all cases shown is Fig. 8, the loss is dramatically decreasing within first 10

epochs and gradually slows down after 10 epochs. Difference is that, in the end, validation loss levels off around 0.1 and training loss levels off at 0.003 when we measuring with Huber loss and using stacked LSTM structure, which is significantly lower than other cases.

Advantages of applying this function is that predicted price is closely fit in actual data price trend, which is indicated as blue line in Fig. 6. Drawback is that neural network cannot accurate predict each tiny fluctuation happening. This drawback is extremely obvious that when there is a sudden increase or drop in actual price, the prediction result usually ignore it. This scenario is shown in Fig. 6 at time 2019-01.

We also find that a DRNN has a better performance compare to shallow LSTM structure. By comparing Fig. 5a and Fig. 5b, we find that with DRNN structure, the loss levels off with value around 0.25, while with shallow LSTM structure, the loss levels off with value above 0.5.



(a) Losses of Prediction vs. Epochs (MSE + LSTM)

(b) Losses of Prediction vs. Epochs (MSE + DRNN)

(c) Losses of Prediction vs. Epochs (Huber + DRNN)

Figure 5. Losses of Prediction of Different Methods

Trading Performances

In this section, we are introducing some important parameters in our model that impacts on performance of our agent. For parameter we discuss below, we train 1000 agents at the same time and measure the mean of their performance for 10 independent runs. We use worth, which refers to the total assets of an agent, as our matrix to measure the performances of agents.

Operation Length. Operation Length refers to total number of times to operate during each training epoch. Generally speaking, this factor controls the upper limitation of how much an agent can possibly earn during an epoch. Fig. 7a is shown an example of total worth as a function of epochs. To obtain the upper limited of

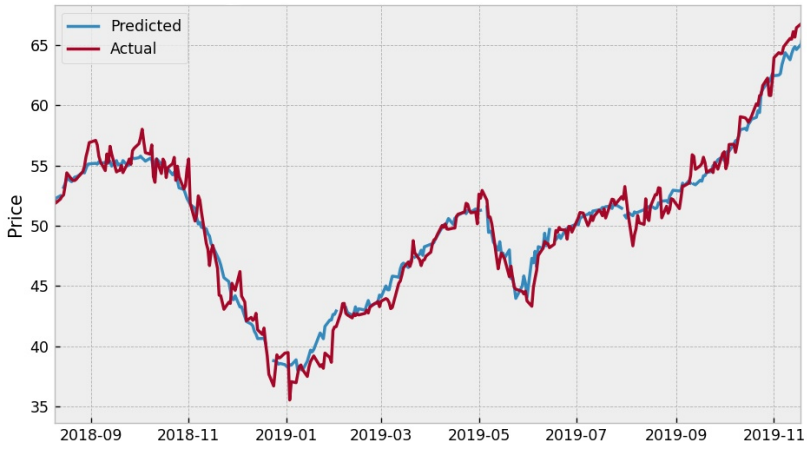


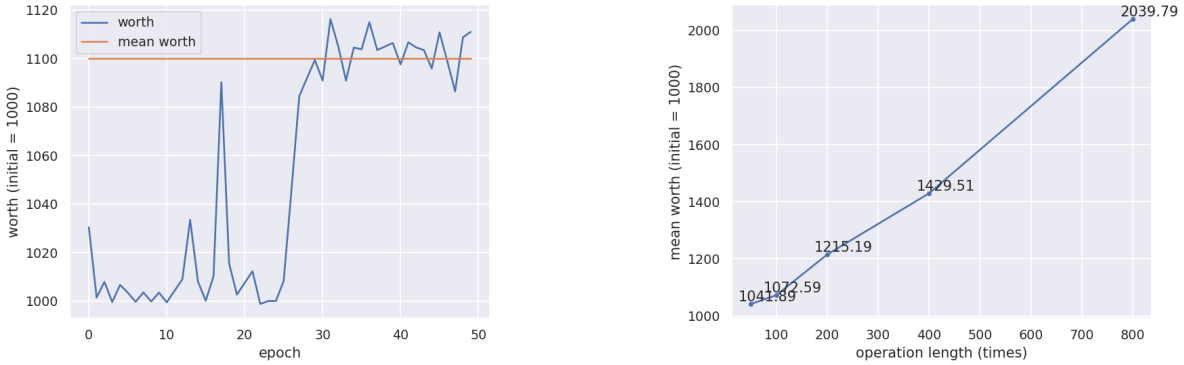
Figure 6. Prediction Result

performance of our agent. We only take data points after the performance leveled of, such as taking values after epoch 25 into account. By dropping data on early training process, we avoid inaccurate measurement of performance of an agent.

In this way, we calculate mean worth for 10 independent runs of our agent. The result is shown in Fig. 7b. The mean worth shows a linear increase as a function of increases in operation length. The longer the operation length is, the more worth an agent is able to earn. This linear relation can be expressed as $w(l) = w_0 + l + \delta$, where $w(l)$ is the worth with l operation length; w_0 is the initial funds an agent has; l is the operation length; and δ is some small variation.

Observation Length. Observation length refers to how many days an agent can look back into before performing an agent. This variable is also the same as the state size in Q-table or length of input state. Fig. 8a shows worth with respect to observation length. We measure the mean and standard deviation of worth for 10 independent runs for 1000 agent. We find that the average worth is close to each other with a value around 1100. However, the standard deviation varies. As shown in Fig. 8a, when observation length is above 80 days, the standard deviation increase by a large amount. Meaning that The performance from agent to agent or in different runs is largely fluctuated.

Discount rate. Discount factor, γ, controls how much we take the utility of our next state into account. It determines how much the reinforcement learning agents cares about rewards in the distant future relative to those in the immediate future. We



(a) Mean Worth Vs. Epochs

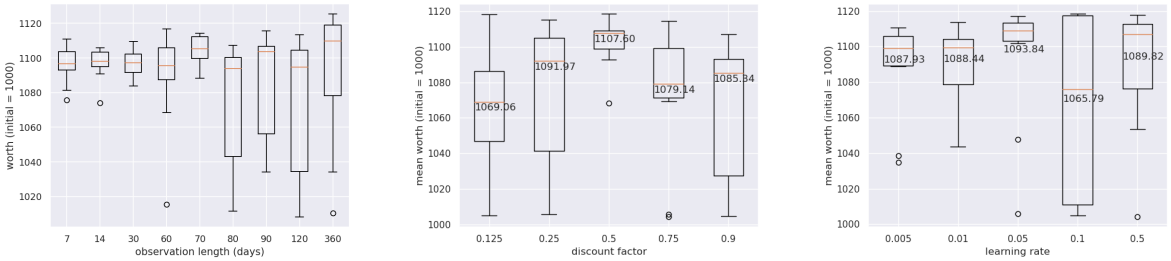
(b) Mean Worth Vs. Operation Length

Figure 7. Operation Length

vary γ from 0.125 to 0.9 for our agent and find that the peak value occurs at $\gamma = 0.5$ as shown in Fig. 8b. At this value, our agent have a highest mean and median with the lowest standard deviation of 10 independent runs. Median decay at value $\gamma > 0.5$ and $\gamma < 0.5$. At the same time, the standard deviation increases.

Learning rate. Learning rate, α, controls how fast our agent learns. If learning rate is too high the learning process jumps over minima. If learning rate is too low, it takes too long to converge or get stuck in an undesirable local minimum. Thus, it is essential to obtain an optimal learning rate. We examine α varying from 0.005 to 0.5. Except for $\alpha = 0.1$, the median of mean worth is 1066, median of mean worth does not vary significantly, which ranges from 1088 to 1094, we can conclude that the learning that does not contribute much to the median of mean worth. However, the standard deviation is varying from value to value. Thus, to find an optimal solution, we find that although there are some outliers at $\alpha = 0.05$, we have lowest standard deviation and highest median value of mean worth at this point.

Epsilon. Epsilon, ϵ, refers to the probability that an agent performs a random action from given action space. Random action helps agent avoid bouncing within a plateau. By applying the same method as we use to determine the impact of operation length, we examine the mean worth as a function of epsilon as shown in Fig. 9. We find that the global maximal mean worth happens at $\epsilon = 0.01$. Mean worth decay on both side at point $\epsilon = 0.01$.



(a) Worth Vs. Obs Length (b) Worth Vs. Discount Rate (c) Worth Vs. Learning Rate

Figure 8. Box Plot of Mean Worth

Utopian and Dystopian

Utopia

Because AI has better calculation, data collection and comprehension, it can make more precise prediction than humans generally. If we use AI in our stock market agent, prediction of stocks made by our stock market agent will more precise. So that such dangers like stock market crashes and crisis will decrease dramatically. The stock market will be more stable under the help from AI agent. As we continue to make progress in AI technique, when better AI stock market agent appears in the future, we could imagine that we will have an eternally peaceful stock market and economy.

Dystopia

AI might cause harm to humans without reason. The logic and rationale behind AI decisions may remain opaque to the person concerned and even the person responsible for the decisions. For example, the AI stock market agent might report the fault prediction intentionally so that it will cause chaos in the public brought people nightmare when they buy the stocks which the agent recommend. If stock market agents make AI-based decision, they must be able to let public know how their decisions be made and explain how they come to this conclusion. The decisions must be traceable and explainable to be contestable.

Discussion

In this paper, we solve the problem of trading stock using two AI system in sequence. Starting with a DRNN model used for prediction future price and following a

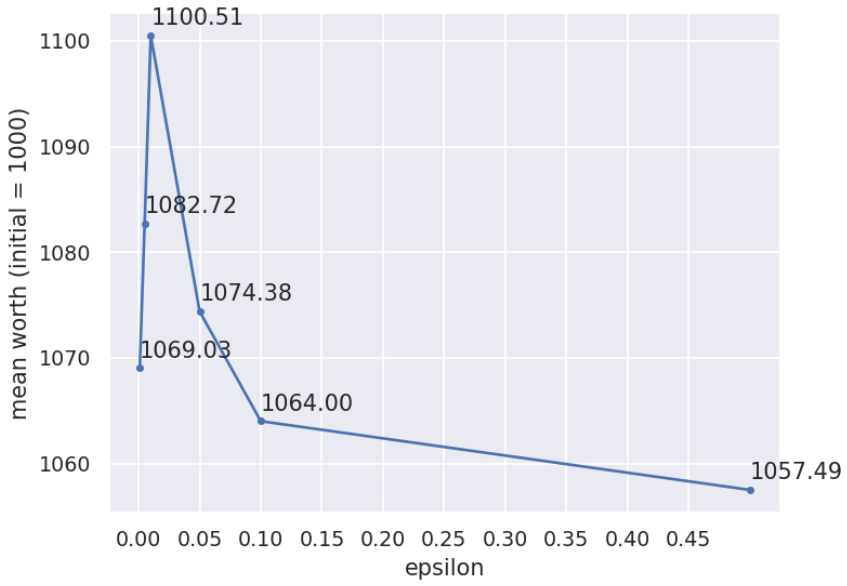


Figure 9. Mean Worth Vs. Epsilon

Q-learning model used for dealing with stock trading system, we examine the performances of these two systems by checking prediction loss and mean worth under different structures and hyper-parameters. Some significant parameters in our Q-learning system includes operation length, epsilon, learning rate, and discount rate. By tuning them into local optimal value, we obtain an optimal solution to our system.

Ethical Concerns

As we get a breakthrough on our project technically, we begin to consider the influence of AI to humanity:

Unemployment. Compared with humans, AI has better calculation and data analysis, so finally it will have more precise prediction of the trend of the stock market. If widely used, AI will dominate the labor force in this realm by winning the competition with humans. It will take away numerous jobs from humans who are engaging in the same area with AI.

Social Instability Risk. Although AI's prediction has a higher correctness rate than humans, it is not one hundred percent correct. Even if it has the visible possibility to make mistakes, more people still trust it because AI has better prediction than humans. As a result, the wrong price prediction may lead to terrible outcomes, so that causes chaos and fears in the stock market even in the whole public.

Inequality Wealth Distribution. Wealth created by machine might be unevenly distributed. Use of AI can help individuals or companies succeed in the stock market according to the correct prediction by it. But also, use of AI makes people lose their jobs who engage in similar jobs. So the gap of wealth difference is enlarged by use of AI.

Higher Labor Requirement. An additional concern about the inequality of AI systems is that it puts individuals who lack the capital and education to create, train, and maintain AI systems at a disadvantage in terms of efficiency and time to those who do. In this way, AI within the stock market serves to perpetuate existing disparities in the ability of the wealthy and the non-wealthy to maintain their long term economic health

Future Work

Besides our work in this paper, future work may focus on the following point:

1. Apply different matrices to measure the performance of agent. For example, instead of measure the worth rate of agents, Compound Annual Growth Rate may be interested to study.

2. Although we examine action spaces with different length, result does not show significantly different. But we do not exhaustively examine all different values. There may be something interesting to study by varying percentages in action space.

References

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

Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, *35*(1), 73–101. https://doi.org/10.1214/aoms/1177703732

Lee, J. W. (2001). Stock price prediction using reinforcement learning. *1*, 690–695 vol.1. https://doi.org/10.1109/ISIE.2001.931880

Tsaih, R., Hsu, Y., & Lai, C. C. (1998). Forecasting s&p 500 stock index futures with a hybrid ai system. *Decision Support Systems*, *23*(2), 161–174. https://doi.org/https://doi.org/10.1016/S0167-9236(98)00028-1

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3-4), 279–292. https://doi.org/10.1007/BF00992698