FINSEARCH 2023

Using Deep RL to optimize Stock Trading strategy and thus maximize Investment Return

MID TERM REPORT(ROADMAP 2)

-Arohan Hazarika(G8)

Introduction

In Roadmap 2, we have to build a model based on Reinforcement Learning which maximizes the return of an Investment. In this case, the environment is the market and our agent is the trader. The state space has various parameters like Account Balance(Cash in Hand), Cumulative Holding Shares of Stocks(Number of stocks held), Closing price of the stock on a day and financial indicators. Action space included BUY, SELL and HOLD. In our model, we have assumed that the trade is happening at the closing price each day.

State Space Representation

As mentioned above, the following parameters were used to define state space of the environment to capture market conditions:

- Account Balance: It is the cash in hand the agent has at any step with which he can buy stocks, Initial state in our testing and training had Rs.1,00,000 as the starting cash in hand of the agent. This parameter is important as our aim is to maximize the investment return and the cumulative return percentage is calculated with respect to it.
- Cumulative Holding Shares of Stocks: It is the total number of stocks that the agent holds at the moment, BUY leads to increment of this parameter by 1, SELL does the opposite(decrement) and HOLD doesn't change its value. This parameter has been kept to calculate Assets(Explained below) based on which we can say that the agent is in profit or loss.
- Closing Price of the Stock on a day: It is the price at which a stock closes on a day and our trade is being performed on this price.
- Simple Moving Average 15 Days (SMA15): It is a parameter which calculates the difference between Current Day Closing Price and SMA15 which can be used to predict whether we should give a Buy/Sell/Hold signal. The agent will learn this trend as he proceeds.
- Simple Moving Average 30 Days (SMA30): It is a parameter which calculates the difference between Current Day Closing Price and SMA15 which can be used to predict whether we should give a Buy/Sell/Hold Signal. The agent will learn this trend as he proceeds. The point of using two SMAs is to basically compare both of them and predict a better signal as the longer SMA gives us an idea about a longer picture of the market and the shorter SMA is for giving weightage to just recent trends. So a mix of both would be better.
- RSI: It's a financial indicator which gives us an idea about the momentum or how fast the market is going up or down. Based on different values of RSI, our agent will learn when it's good to BUY or SELL as he proceeds.

Action Space

Action Space is defined by BUY, SELL or HOLD.

BUY: If the agent buys a stock, then his/her account balance gets reduced and his number of holding stocks increases.

SELL:If the agent sells a stock, then his/her account balance gets increased and his number of holding stocks decreases.

HOLD: This is equivalent to doing nothing.

In this model, there is a limitation that the agent can buy/sell only one stock at a transaction(in a day),done to limit the action space and reduce complexity whereas it would have been more realistic if we would also include the number of stocks to be bought/sold.

All the parameters(financial indicators) are calculated using the closing price data of the market and each timestep in the model is a day.

Rewards

A term **Asset** has been defined in the model which is the sum of Account Balance and the total stock price(it's the product of number of holding stocks and stock value).

Rewards are defined as follows:

- At each step, the reward is the percentage change in the Asset value of the agent.
- However, if the Asset value goes below 98% of the initial Asset(in my training case, initial Asset was Rs.100000), the agent receives a -200 reward penalty which punishes it and gives a more stricter corrective action.

Inspiration has been taken from the Research Paper “Model based reinforcement learning for stock trading optimization” by Huifang Huang, Ting Gao, Luxuan Yang, Yi Gui, Jin Guo, Peng Zhang, September 2021.

In the future, if we include more financial indicators, better prediction and more realistic decisions could be taken by the Agent but it would involve more computation time, higher complexity and more advanced RL Algorithms.

Financial Indicators

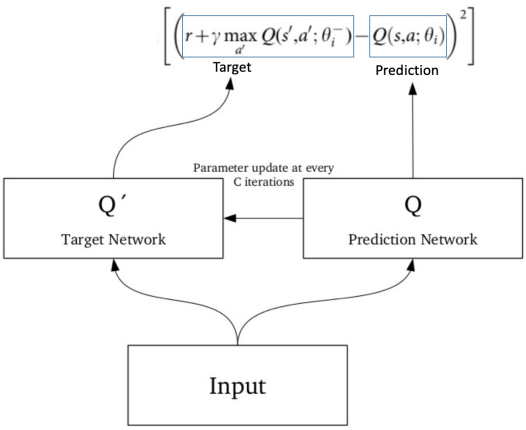
SMA: A simple moving average (SMA) is an arithmetic moving average calculated by adding recent prices and then dividing that figure by the number of time periods in the calculation average. The difference of a SMA and the current price of the stock can give us a soft signal about the trend whether the stock will go up or down.

RSI: The RSI is an indicator which takes into account the Average Loss and Average Gain over a 14 day period which helps us to determine whether the stock's value is in downtrend or uptrend. If RSI value is less, around 30 or less, the trend is downtrend(loss) and we shall sell the stock and vice versa if RSI value is around 70 or more and if it's around 50, it can be kind of a neural signal.

The purpose of using combinations of different financial indicators as signals is good as the agent gets an idea what step can be taken based on the trend. Combinations of indicators are always better as the chances of false signals are less compared to use of a single indicator to predict the trend.

RL Algorithm

We have used Deep Q Learning with the help of neural networks and Epsilon-Greedy Policy as the exploration policy. In deep reinforcement learning, Deep Q Learning algorithm is an algorithm where an agent learns by updating Q values for different actions for a given state by changing weights and biases using a neural network based on its experience. There are two networks Target and Prediction which give values for calculation of Loss Function(also known as Cost Function) which gives an idea to the neural network how good/bad are its predictions.



We used deep Q Learning algorithm as its a relatively simple Deep RL algorithm and could be applied easily. In the future, we intend to use more complex RL algorithms. Moreover, Deep Q Learning was working more or less fine which will be explained in detail in the Results and Analysis section.

I took help from the website https://keon.github.io/deep-q-learning/ for learning the Deep Q Learning Algorithm.

Model Parameters

Epsilon: The exploration rate for the Epsilon-Greedy Algorithm which is related to exploration vs exploitation.

Discount Rate: Parameter which is used for providing weightage to recent rewards or how much does a future reward affect the current Q value.

Learning Rate: Parameter used in Gradient Descent Algorithm(for minimizing the cost function) which is related to the speed at how the neural network learns.

Results and Analysis

I had trained the model for 600 timesteps(in slots of 300 steps each) which is equivalent to 1.5 years of data and the following results were obtained.

For the first 300 episodes, we can see that the agent is more or less able to maintain the initial Asset though it has made a 10% loss after the completion of the episode but it's fine as the agent is in the very earlier stage and we can see that Asset hasn't dropped very significantly throughout the journey which is due to heavy penalty of -200 for dropping below Rs.98,000 total assets. Epsilon was kept initially 1 but with a decay rate of 0.995 so that epsilon decreases at every time step with minimum epsilon of 0.1. Discount doctor was kept at 0.95 so that future rewards are taken into account significantly and the learning rate was 0.005.

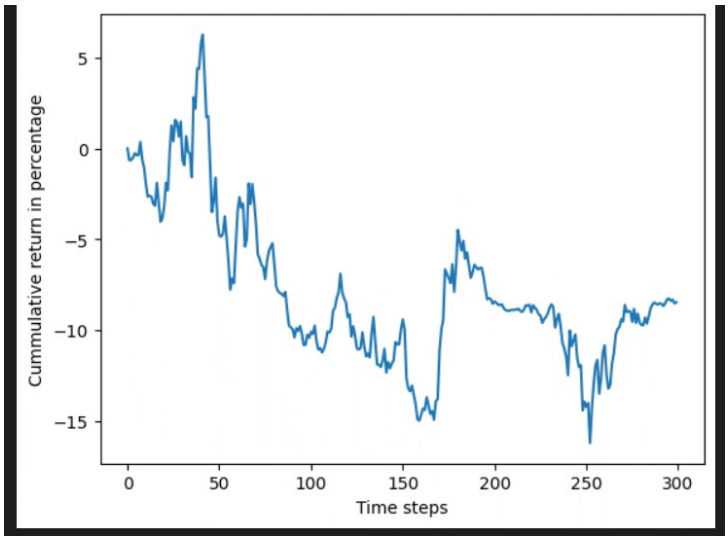


Fig: Cumulative Return Percentage

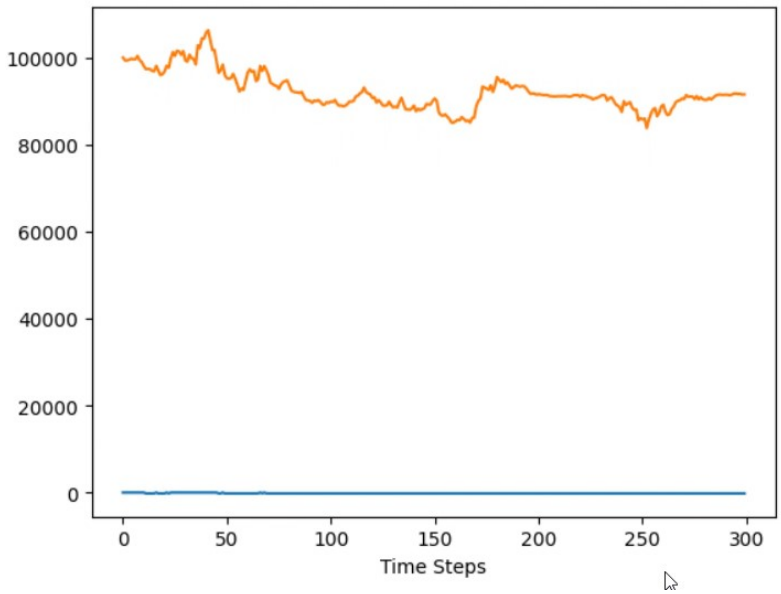


Fig: Assets

For the second set of 300 episodes, the results were very promising as the final profit percentage was 50% and throughout the journey we can see that cumulative percentage was on the positive side most of the time though it dropped a little below 0 for some time in between. Here epsilon was started with the value of 0.5 and rest all parameters were kept the same, this was done keeping in mind that the agent had gained some experience and shall exploit(choose the best action according to the Q values) also.

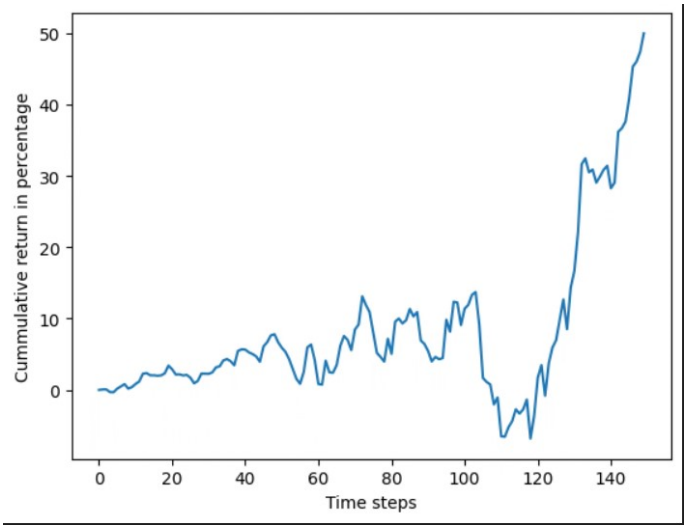


Fig: Cumulative Return Percentage

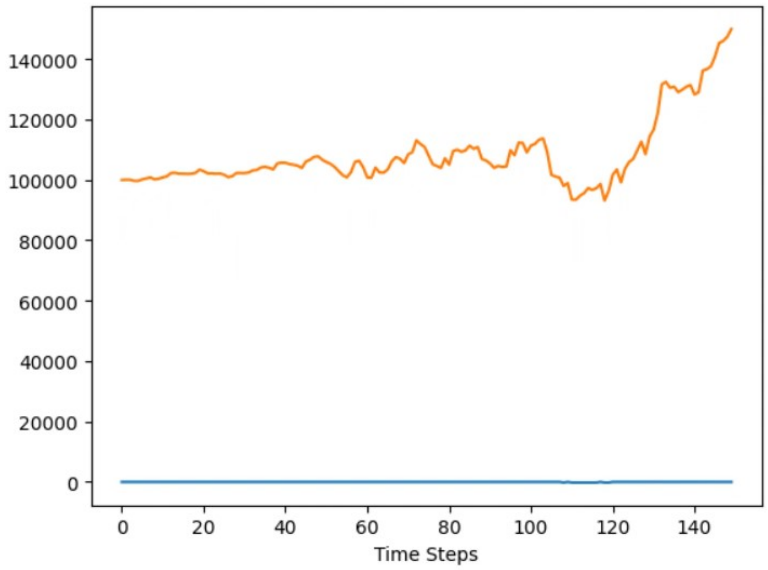


Fig: Assets

Then the agent was tested for 500 steps, following results were obtained.

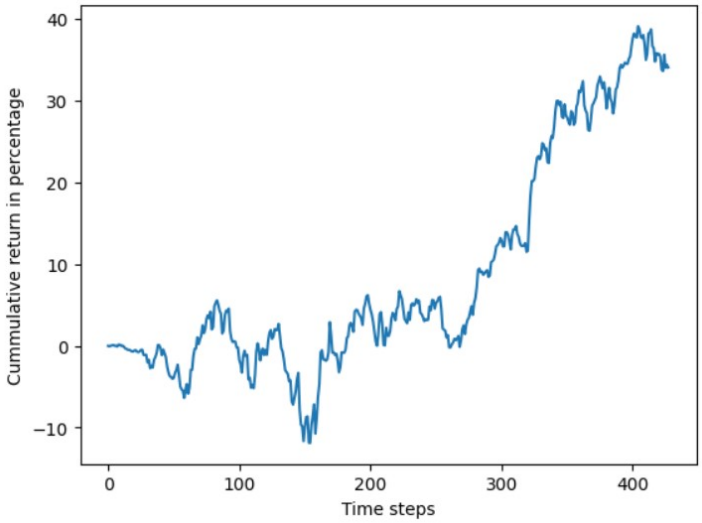


Fig: Cumulative Return Percentage

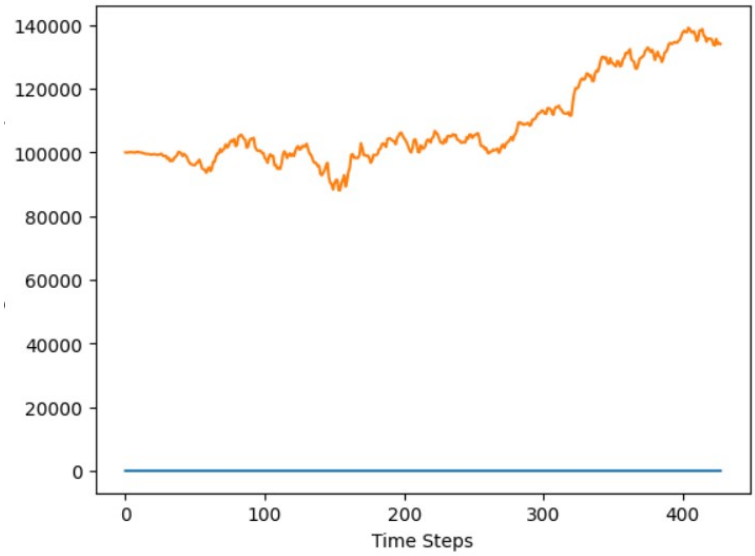


Fig: Assets

Some my observations and thoughts are that since stock market is a stochastic environment(at least for our model), epsilon should be kept in between 0.5 and 0.1 only and shall not be dropped below 0.1 so that our agent also has that exploration factor into account and is not too deterministic. As we can see in the above diagrams, in some cases cumulative return percentage dropped below 0 (Loss!!), hence it can lead to losses too but as we can see it hasn't dropped below minus 10-minus 15 %(and this percentage would decrease if initial Asset value is higher in value) due to heavy penalty of -200.