Assignment 3

Aakar Sood

Dec 10, 2021

1 Assignment Overview

The goal of the assignment is to implement Q-Learning environment to learn the trends of stock market. After making the algorithm learn the trends the overall goal is to make a profitable trade over a period of time.

2 Dataset

We are provided with historical data of NVIDIA stock price for 5 years. The dataset has 1258 entries and span from 10/27/2016 to 10/26/2021. In the dataset we have features such as at what price the stock opened and closed, intraday high and low, as well as volume of shares traded on that specific day.

3 Environment

The environment consist of 3 actions and 4 states. The actions are 'BUY', 'SELL', and 'HOLD'. Each of the actions are encoded from 0 to 2 respectively. We also have 4 states defined in the environment which are 'Increase in price', 'Decrease in price', 'Stock held' and 'Stock not held'. These are again encoded from 0 to 3 respectively for algorithm to understand. We are also provided with $100k investment to invest in stock price.

4 Q-Learning

Q-Learning is a type of reinforcement learning algorithm which helps in determining the best action possible in a given state. For our problem we have a 4x3 matrix for Q-Learning which basically indicates the 4 states and 3 actions we have in our environment. During the initialization of Q-Learning table we initialize every state-action combination by value 0. This table acts as a reference table for the agent to understand as to what will be next best possible action. The values in the Q table are known as q table and are updated using the rewards it receives when it takes an action. Rewards here are associated to whether the action of agent is profitable or incur losses. If it incur losses the rewards will be negative or else positive. Below is the formula for updating q-values:

$$Q(s,a) = (1-\alpha) * Q(s,a) + \alpha * (r + \gamma * Q(s',a'))$$

In the above equation α is the learning rate, γ is the discount factor, r is the rewards it obtained from the previous action, Q(s,a) indicates the state and action in which agent was earlier and Q(s',a') indicates the state and action the agent is in right now. The agent generally takes the action which maximizes its rewards and this method is known as exploitation but in certain scenario we want the agent to take a random step which can be categorized as exploration on the agent's part. In the initial phases of learning we want our agent to explore more and in the later stages we want our agent to exploit more. This can be controlled using the epsilon parameter which is nothing but the probability of an agent to explore.

4.1 Training

In the training phase we have initialized our agent with certain parameters and there values is as follow:
- $\alpha = 0.6$
- $\gamma = 0.9$
- no episodes = 1000
- decay = 0.001

The training of the agent ran for 1000 iterations as mentioned above. Below are some of the graphs for training:

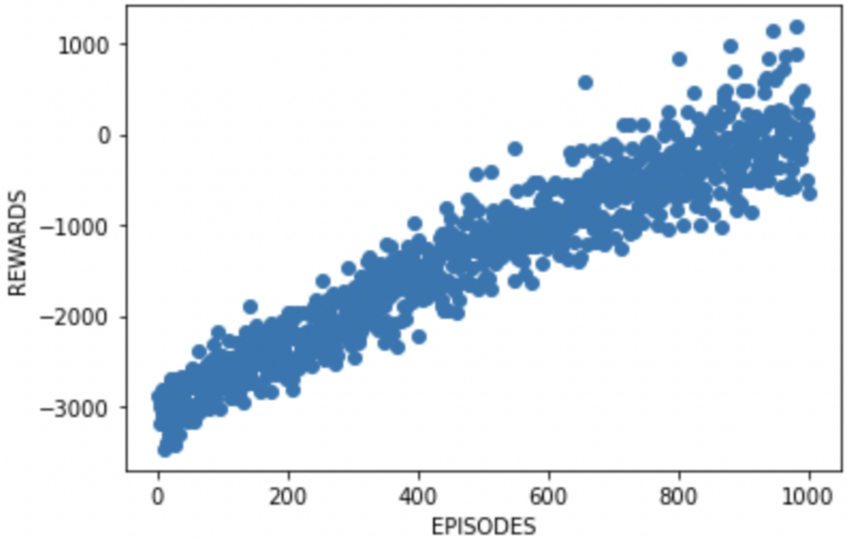

Figure 1: This is an image showing the rewards earned per episode.

From the above graph we can see that the model initial started with negative rewards and then slow learning iteration by iteration increases the rewards value

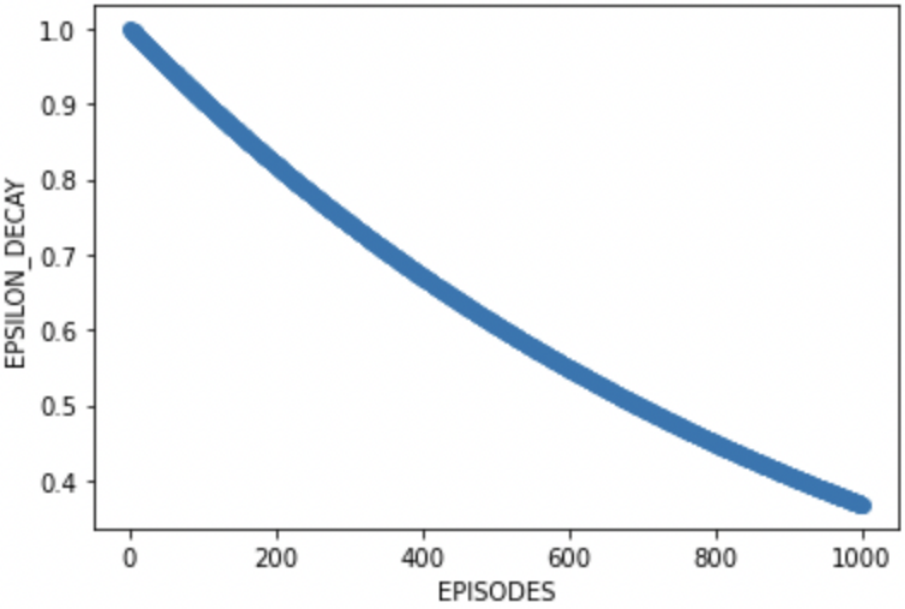

Figure 2: This is an image showing how agent's exploration probability is decreasing per episode.

In the above graph we can see how exponentially the probability of exploration for model is decreasing per episode.

4.2 Testing

After training the agent for 1000 iterations we have created a Q-table consisting of the next best possible action based on the given state. We run our agent on the data for one episode.
Below is the graph representing the total reward graph over 250 days:

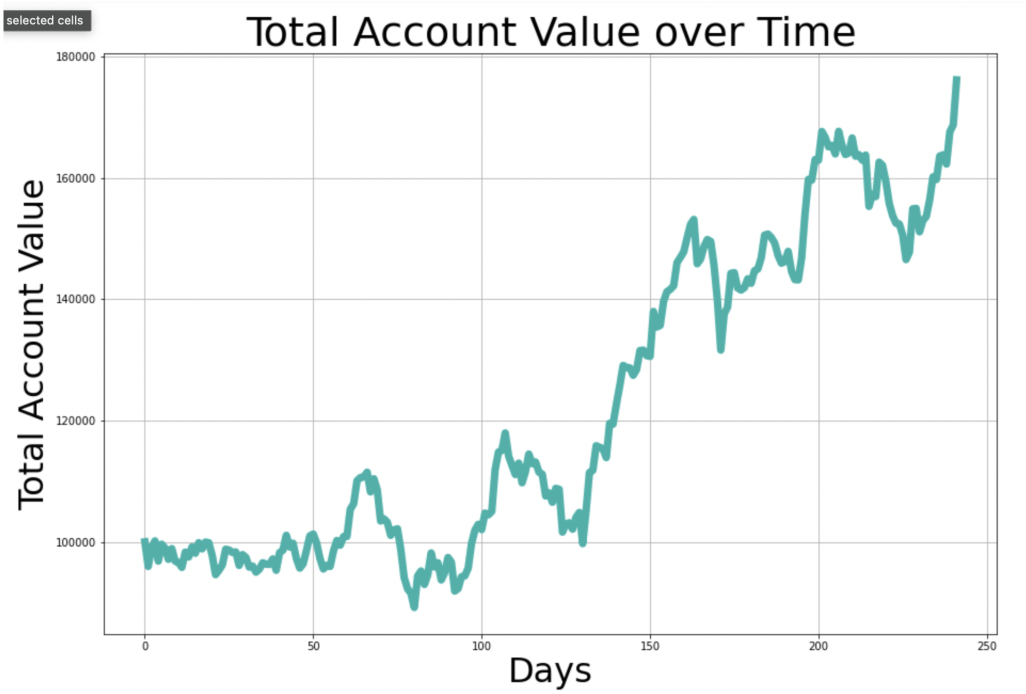

Figure 3: This is an image showing the rewards earned over a span of 250 days.

The total balance over time is coming out to be $176207.54 which basically means that over a span of 250 days the agent started from $100k and is ending with $176k making a profit of $76k