CSE 4/574: Introduction to Machine Learning
Fall 2021

Prof. Sargur Srihari

Assignment 3 - Deep Reinforcement Learning

Due Date: 5th December, 11:59pm

1 Assignment Overview

The goal of the assignment is to learn the trends in stock price and perform a series of trades over a period of time and end with a profit. In each trade you can either buy/sell/hold. You will start with an investment capital of $100,000 and your performance is measured as a percentage of the return on investment.

You will use the Q-Learning algorithm for reinforcement learning to train an agent to learn the trends in stock price and perform a series of trades. You will implement Q-learning algorithm from scratch. The purpose of this assignment is to understand the benefits of using reinforcement learning to solve the real world problem of stock trading.

1.1 Dataset

You have been given a dataset on the historical stock price for Nvidia for the last 5 years. The dataset has 1258 entries starting 10/27/2016 to 10/26/2021. The features include information such as the price at which the stock opened, the intraday high and low, the price at which the stock closed, the adjusted closing price and the volume of shares traded for the day.

1.2 Environment

The environment which calculates the trends in the stock price is provided to you along with the documentation. Your task is to use the Q-learning algorithm to figure out a trading strategy and increase your total account value over time.

Environment Structure:

Init method: This method initializes the environment.

Input parameters:

1. file_path: Path of the CSV file containing the historical stock data.
2. train: - Boolean indicating whether the goal is to train or test the performance of the agent.
3. number_of_days_to_consider = Integer representing whether the number of days the for which the agent considers the trend in stock price to make a decision

Reset method: This method resets the environment and returns the observation.

Returns: observation: - Integer in the range of 0 to 3 representing the four possible observations that the agent can receive. The observation depends upon whether the price increased on average in the number of days the agent considers, and whether the agent already has the stock or not.

Step method: This method implements what happens when the agent takes the action to Buy/Sell/Hold.

Input parameter: action: - Integer in the range 0 to 2 inclusive.

Returns:

1. observation: - Integer in the range of 0 to 3 representing the four possible observations that the agent can receive. The observation depends upon whether the price increased on average in the number of days the agent considers, and whether the agent already has the stock or not.
2. reward: - Integer/Float value that's used to measure the performance of the agent.
3. done: - Boolean describing whether or not the episode has ended.
4. info: - A dictionary that can be used to provide additional implementation information.

Render method: This method renders the agent's total account value over time. Input parameter: mode: 'human' renders to the current display or terminal and returns nothing.

Note: You can't make changes to the environment.

Task: Implementing Q-learning

1.3 Implementing Q-learning

Implement the Q-learning algorithm from scratch. A general structure for implementing the Q-learning algorithm is provided in the code file, however you are welcome to modify it/follow your own.

1.4 Producing a Trading Strategy

Apply the Q-learning algorithm to the stock trading environment to generate a trading strategy for maximizing the agent's total account value over time.

In your report:

1. Show and discuss the results after applying the Q-learning algorithm to solve the stock trading problem. Plots should include epsilon decay and total reward per episode.
2. Briefly describe the stock trading environment that's provided to you (e.g. possible actions, states, agent, goal, rewards, etc).
3. Provide the evaluation results. Run your trained agent (you will have to set the train parameter to false) and evaluate the agent's performance, where the agent chooses only greedy actions from the learnt policy. Plot should include the agent's account value over time.

2 Deliverables

Submit your work using UBLearns.

2.1 Report

The report should be delivered as a separate pdf file. You may include comments in the Jupyter Notebook, however you will need to duplicate the results in the separate pdf file.

2.2 Code

Python is the only code accepted for this project. You can submit the code in Jupyter Notebook or Python script. You can submit multiple files, but they all need to have a clear name. After executing command python main.py in the first level directory or Jupyter Notebook, it should generate all the results and plots you used in your report and should be able to be printed out in a clear manner. Additionally you can submit the trained parameters, so that the grader can fully replicate your results.

3 References

- NIPS Styles (docx, tex)
- Overleaf (LaTex based online document generator) - a free tool for creating professional reports
- Lecture slides

4 Final Submission [Due date: 5th December 2021]

Add your report pdf and ipynb/python script to a zip file *ubitname_assignment3.zip* (e.g. *soumyyak_assignment3.zip*) and upload it to UBlearns. After the assignment is graded, you may be asked to demonstrate it to the instructor if your results or reasoning in your report are not clear enough.

5 Important Information

The standing policy of the Department is that all students involved in any academic integrity violation (e.g. plagiarism in any way, shape, or form) will receive an F grade for the course. The catalog describes plagiarism as "Copying or receiving material from any source and submitting that material as one's own, without acknowledging and citing the particular debts to the source, or in any other manner representing the work of another as one's own.". Updating the hyperparameters or modifying the existing code is not part of the assignment's requirements and will result in a zero. Please refer to the UB Academic Integrity Policy.