PART B

Problem Statement: Automated Trading Strategy using DDQN

Background: In finance, automated trading systems are used to execute trades in financial markets based on predefined rules. These systems often employ quantitative strategies that rely on analyzing market data to make trading decisions. However, designing effective trading strategies manually can be challenging due to the complex and dynamic nature of financial markets.

Objective: The objective is to develop an automated trading system that can learn to buy, sell or hold the stocks directly from historical market data.

Dataset Choice Instructions:

1. You need to select a trading dataset of Klines daily data only as it suits the problem statement. Select the historical data of 30 mins for one month for any stock from the list.
2. The link for downloading the dataset can be found here https://data.binance.vision/?prefix=data/spot/daily/klines/. The details of Klines daily data is given on github link Details on how to get Binance public data **(Use only Klines dataset only)**

Problem Design:

State Space: The state space will contain the data about the stocks such as its close value, high & low value, volume, Number of trades and so on.

Action Space: (buy, hold or sell) The Agent has defined functions for buying and selling options. The get_state and act functions employ a neural network to generate the neural network's next state. 1 indicates a buy call, 2 indicates a sell call, and 3 indicates a hold. In each cycle, the state is determined, and an action is done that either buys or sells stocks.

Environment: Design a trading environment with all the parameters initialized for the agent to learn. The agent will have an initial amount to invest in stock be 100000.

Rewards: The rewards are then determined by adding or subtracting the value generated by using the call option. The action taken in the previous state influences the action taken in the next. The total profit variable contains information about the entire rewards.

Implementation:

1. Print its statistics.
2. **Implement DDQN on the dataset**.
3. **Implement only the DRL approach with DDQN**.
    a. Design a Trading Environment.
    b. State the state space and action space
    c. Clearly define the parameters used for training an AI agent.
        i. Number of episodes
        ii. Max capacity of replay memory
        iii. Batch size
        iv. Period of Q target network updates
        v. Discount factor for future rewards
        vi. Initial value for epsilon of the e-greedy
        vii. Final value for epsilon of the e-greedy
        viii. Learning rate of ADAM optimizer, and etc..
    d. Define the functions for Buy, Sell and Hold actions.
    e. Implement a replay buffer for storing the experiences.
    f. Design the Main Network
    g. Target Network
4. Plot the graph for agents for buying and selling of the stock.
5. Summarizing your findings.

References:

1. https://www.analyticsvidhya.com/blog/2021/01/bear-run-or-bull-run-can-reinforcement-learning-help-in-automated-trading/
2. https://github.com/ThibautTheate/An-Application-of-Deep-Reinforcement-Learning-to-Algorithmic-Trading/tree/main
3. https://medium.com/datapebbles/building-a-trading-bot-with-deep-reinforcement-learning-drl-b9519a8ba2ac