CSE546 - Reinforcement Learning

Assignment - 1 Part - 1

Venkata Sai Divya Pallineni
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
vpalline@buffalo.edu

1 Defining RL Environments

1.1 Deterministic Environment

A 5x5 Grid-World is created and when the Agent is in State (s) and performing an Action (a) the probability that it will reach state ($s\prime$) is either 0 or 1.

- **State (S) :-** Defined 25 states from [0][0] to [4][4] of which Agent's
  . Initial position = [0][4]
  . Terminal position = [2][2]

- **Action (A) :-** The agent can take four actions in a single time step by changing the x,y coordinates.
  . $A = \{Up, Down, Right, Left\}$

- **Transition Probability (P):-** When the Agent is in State (s) and performing an Action (a) the probability that it will reach state ($s\prime$) is either 0 or 1.
  . $P(s\prime, r|s, a) = \{0, 1\}$

- **Reward (R):-** 5 rewards of which for Gold chest(+10), Food (+25), Devil (-5), Dragon (-10) , End Goal (+50) same state(-1) and other states(1) are awarded.
  . $R = \{-10, -5, -1, +1 + 10, +25, +50\}$

- **Discount factor (γ):-** The agentś goal is to maximize the expected sum of rewards without any discounting.
  . $\gamma = 1$

- **Main Objective:-** The agentś main goal is to reach End Goal or terminal position which is state [2][2] to collect maximum reward of +50

1.2 Stochastic Environment

A 5x5 Grid-World is created and when the Agent is in State (s) and performing an Action (a) there is a randomness or uncertainty in which state the Agent will end-up.

- **State (S) :-** Defined 25 states from [0][0] to [4][4] of which Agent's
  . Initial position = Random or Uncertain
  . Terminal position = [4][0]

- **Action (A) :-** The agent can take four actions in a single time step by changing the x,y coordinates.
  . $$A = \{Up, Down, Right, Left\}$$

- **Transition Probability (P):-** When the Agent is in State (s) and performing an Action (a) the probability that it will reach state ($s\prime$) is either 0 or 1.
  . $$P(s\prime, r|s, a) = \{0, 1\}$$

- **Reward (R):-** 5 rewards of which for Gold chest(+10), Food (+25), Devil (-5), Dragon (-10) , End Goal (+50) same state(-1) and other states(1) are awarded.
  . $$R = \{-10, -5, -1, +1 + 10, +25, +50\}$$

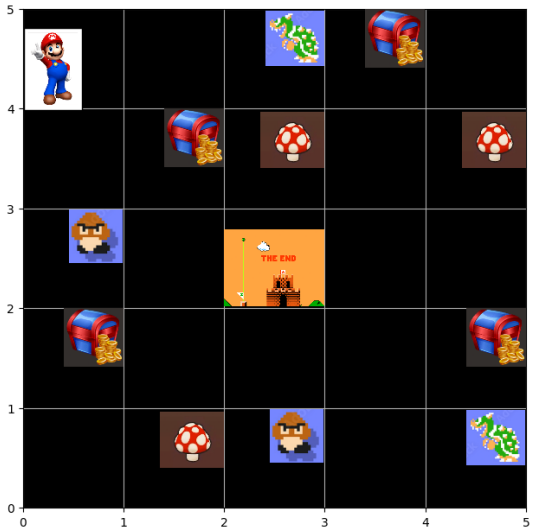
- **Discount factor (γ):-** The agentś goal is to maximize the expected sum of rewards without any discounting.
  . $$\gamma = 1$$

- **Main Objective:-** The agentś main goal is to reach End Goal or terminal position which is state [4][0] to collect maximum reward of +50

2 Visualizations

2.1 Deterministic Environment

2.1.1 Displaying Deterministic Environment with Agent in its initial position



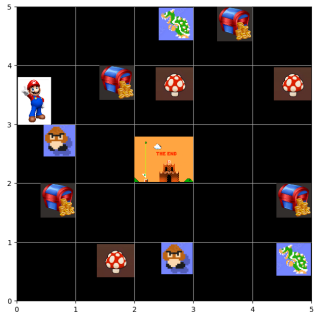
2.1.2 Cumulative rewards collected during 16 time steps

In [95]:

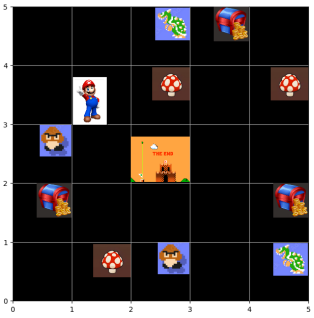
```
#Running the Mario GridWorld Game in Deterministic Environment
done = False
Actions={3:"Left",1:"Right",2:"Up",0:"Down"}
print("=========================================================================================================")
while not done:
    action = random.randint(0,3)
    reward, done, info = env.step(action)
    env.render()
    print("Timestep: {}".format(env.timeStep)+"\t\t\t Performing Action: "+Actions[int(action)])
    print(info)
    print("=========================================================================================================")
```

```
=========================================================================================================
Timestep: 1                     Performing Action: Left
Current Agent Position: [0,4] ; Current State Reward: 0 ; Total Cumulative Reward: 0
=========================================================================================================
Timestep: 2                     Performing Action: Right
Current Agent Position: [1,4] ; Current State Reward: 0 ; Total Cumulative Reward: 0
=========================================================================================================
Timestep: 3                     Performing Action: Down
Current Agent Position: [1,3] ; Current State Reward: 10 ; Total Cumulative Reward: 10
=========================================================================================================
Timestep: 4                     Performing Action: Right
Current Agent Position: [2,3] ; Current State Reward: 0 ; Total Cumulative Reward: 10
=========================================================================================================
Timestep: 5                     Performing Action: Down
Current Agent Position: [2,2] ; Current State Reward: -10 ; Total Cumulative Reward: 0
=========================================================================================================
Timestep: 6                     Performing Action: Down
Current Agent Position: [2,1] ; Current State Reward: 0 ; Total Cumulative Reward: 0
=========================================================================================================
Timestep: 7                     Performing Action: Left
Current Agent Position: [1,1] ; Current State Reward: 0 ; Total Cumulative Reward: 0
=========================================================================================================
Timestep: 8                     Performing Action: Down
Current Agent Position: [1,0] ; Current State Reward: 25 ; Total Cumulative Reward: 25
=========================================================================================================
Timestep: 9                     Performing Action: Up
Current Agent Position: [1,1] ; Current State Reward: 0 ; Total Cumulative Reward: 25
=========================================================================================================
Timestep: 10                    Performing Action: Up
Current Agent Position: [1,2] ; Current State Reward: 0 ; Total Cumulative Reward: 25
=========================================================================================================
Timestep: 11                    Performing Action: Down
Current Agent Position: [1,1] ; Current State Reward: 0 ; Total Cumulative Reward: 25
=========================================================================================================
Timestep: 12                    Performing Action: Down
Current Agent Position: [1,0] ; Current State Reward: 25 ; Total Cumulative Reward: 50
=========================================================================================================
Timestep: 13                    Performing Action: Down
Current Agent Position: [1,0] ; Current State Reward: 25 ; Total Cumulative Reward: 75
=========================================================================================================
Timestep: 14                    Performing Action: Right
Current Agent Position: [2,0] ; Current State Reward: -5 ; Total Cumulative Reward: 70
=========================================================================================================
Timestep: 15                    Performing Action: Right
Current Agent Position: [3,0] ; Current State Reward: 0 ; Total Cumulative Reward: 70
=========================================================================================================
Timestep: 16                    Performing Action: Right
Current Agent Position: [4,0] ; Current State Reward: 50 ; Total Cumulative Reward: 120
=========================================================================================================
```

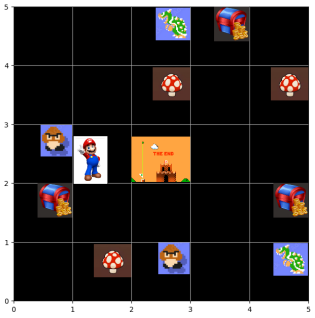
2.1.3 Visualization- Running environment for 16 time steps



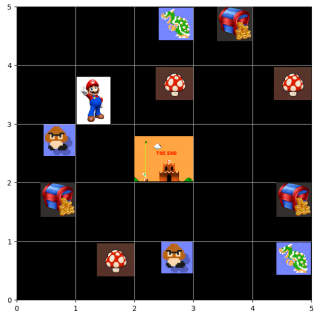
(a) Ts:1 Action:Left +0



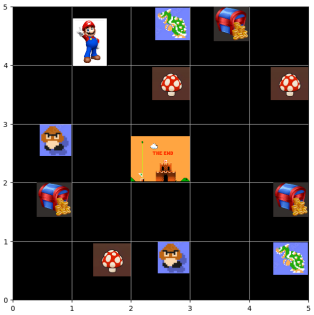
(b) Ts:2 Action:Right +0



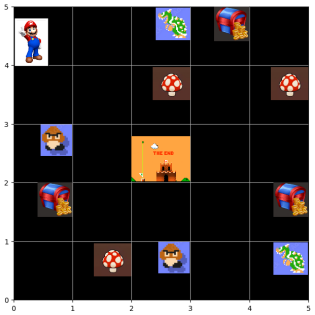
(c) Ts:3 Action:Down +10



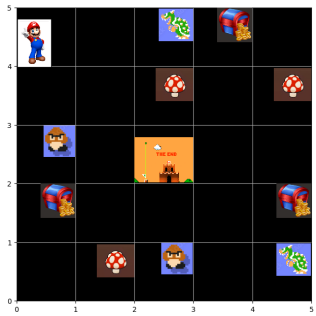
(d) Ts:4 Action:Right +10



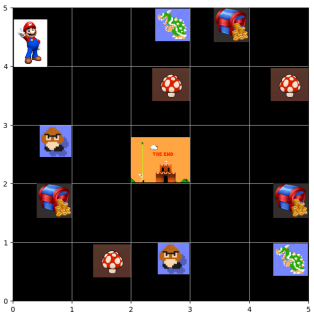
(e) Ts:5 Action:Down +0



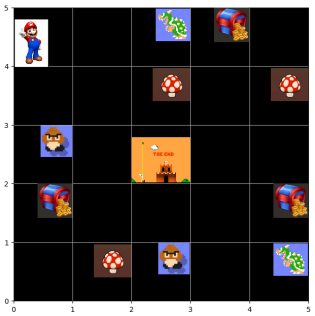
(f) Ts:6 Action:Down +0



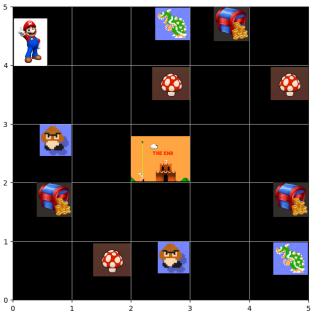
(g) Ts:7 Action:Left +0



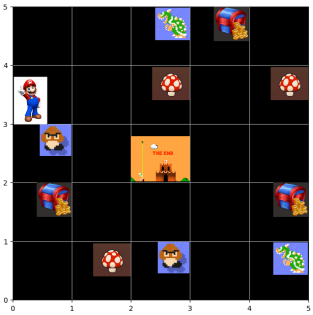
(h) Ts:8 Action:Down +25



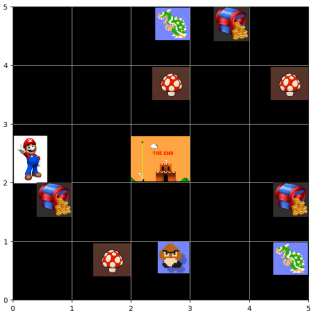
(i) Ts:9 Action:Up +25



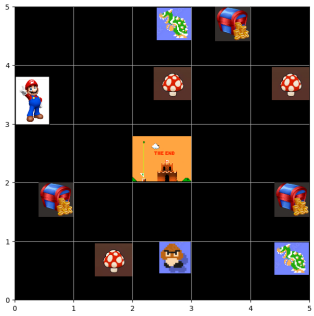
(j) Ts:10 Action:Up +25



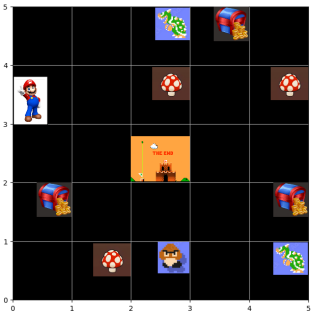
(k) Ts:11 Action:Down +25



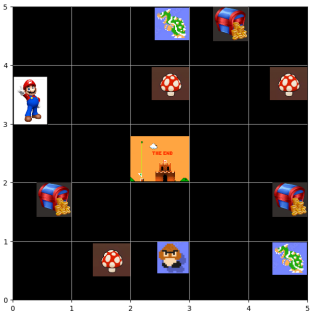
(l) Ts:12 Action:Down +50



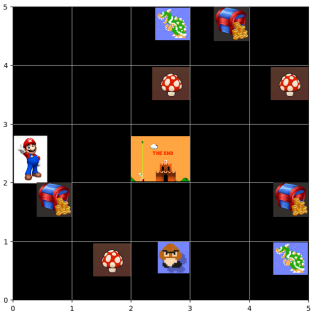
(m) Ts:13 Action:Down +75



(n) Ts:14 Action:Right +70



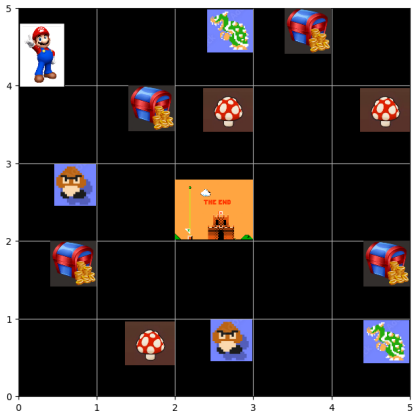
(o) Ts:15 Action:Right +70



(p) Ts:16 Action:Right +120

2.2 Stochastic Environment

2.2.1 Displaying Stochastic Environment with Agent in its initial position

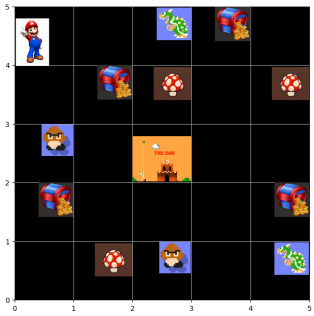


2.2.2 Cumulative rewards collected during 16 time steps

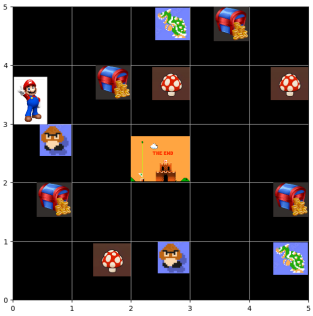
```
#Running the Mario GridWorld Game in Stochastic Environment
done = False
print("\n================================================================================================")
while not done:
    action = random.randint(0,3)
    reward, done, info = env.step(action)
    env.render()

    print("Timestep: {}".format(env.timeStep)+"\t\t\t Performing Action: {}\n".format(action))
    print(info)
    print("\n================================================================================================")
```

```
================================================================================================
Timestep: 1                     Performing Action: 0

Current Agent Position: [0,3] ; Current Reward: 0 ; Total Cumulative Reward: 0

================================================================================================
Timestep: 2                     Performing Action: 1

Current Agent Position: [0,4] ; Current Reward: 0 ; Total Cumulative Reward: 0

================================================================================================
Timestep: 3                     Performing Action: 3

Current Agent Position: [1,4] ; Current Reward: 0 ; Total Cumulative Reward: 0

================================================================================================
Timestep: 4                     Performing Action: 3

Current Agent Position: [2,4] ; Current Reward: 25 ; Total Cumulative Reward: 25

================================================================================================
Timestep: 5                     Performing Action: 1

Current Agent Position: [2,4] ; Current Reward: -1 ; Total Cumulative Reward: 24

================================================================================================
Timestep: 6                     Performing Action: 3

Current Agent Position: [4,4] ; Current Reward: 0 ; Total Cumulative Reward: 24

================================================================================================
Timestep: 7                     Performing Action: 3

Current Agent Position: [4,4] ; Current Reward: -1 ; Total Cumulative Reward: 23

================================================================================================
Timestep: 8                     Performing Action: 1

Current Agent Position: [4,4] ; Current Reward: -1 ; Total Cumulative Reward: 22

================================================================================================
Timestep: 9                     Performing Action: 0

Current Agent Position: [4,3] ; Current Reward: 25 ; Total Cumulative Reward: 47

================================================================================================
Timestep: 10                    Performing Action: 3

Current Agent Position: [4,3] ; Current Reward: -1 ; Total Cumulative Reward: 46

================================================================================================
Timestep: 11                    Performing Action: 3

Current Agent Position: [4,3] ; Current Reward: -1 ; Total Cumulative Reward: 45

================================================================================================
Timestep: 12                    Performing Action: 2

Current Agent Position: [3,3] ; Current Reward: 0 ; Total Cumulative Reward: 45

================================================================================================
Timestep: 13                    Performing Action: 1

Current Agent Position: [3,4] ; Current Reward: 10 ; Total Cumulative Reward: 55

================================================================================================
Timestep: 14                    Performing Action: 2

Current Agent Position: [0,0] ; Current Reward: 0 ; Total Cumulative Reward: 55

================================================================================================
Timestep: 15                    Performing Action: 1

Current Agent Position: [0,1] ; Current Reward: 10 ; Total Cumulative Reward: 65

================================================================================================
Timestep: 16                    Performing Action: 0

Current Agent Position: [0,0] ; Current Reward: 0 ; Total Cumulative Reward: 65

================================================================================================
```

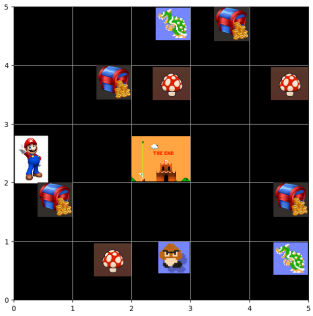
2.2.3 Visualization- Running environment for 16 time steps



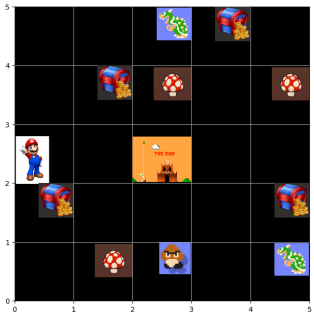
(q) Ts:1 R:0



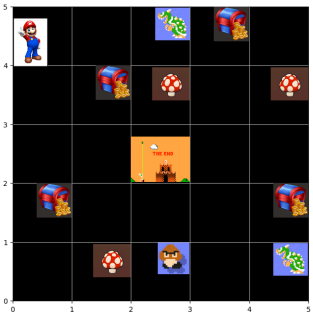
(r) Ts:2 R:0



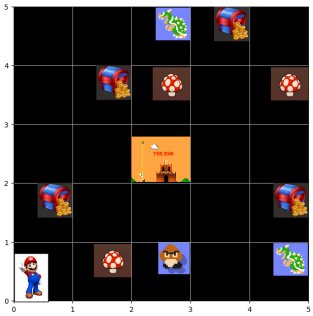
(s) Ts:3 R:0



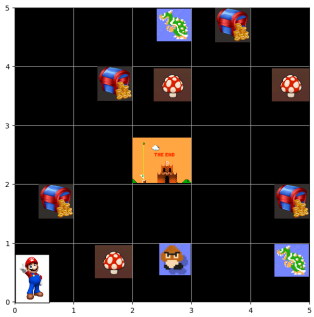
(t) Ts:4 R:+25



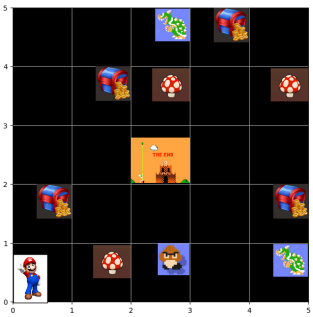
(u) Ts:5 R:+24



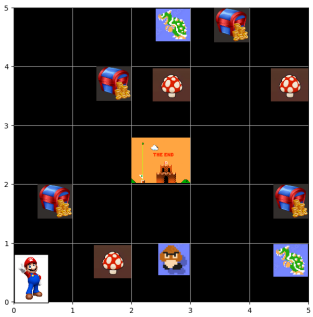
(v) Ts:6 R:+24



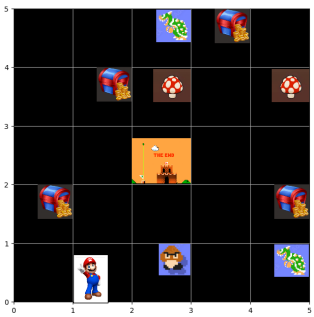
(w) Ts:7 R:+23



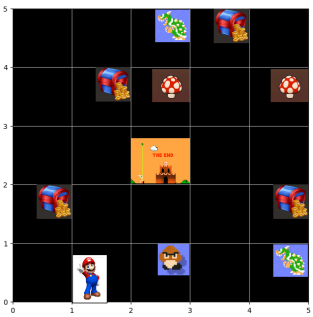
(x) Ts:8 R:+22



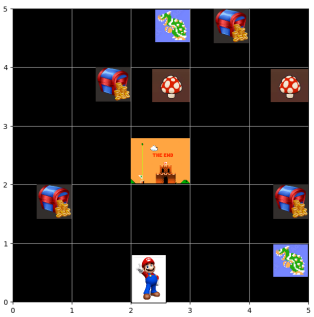
(y) Ts:9 R:+47



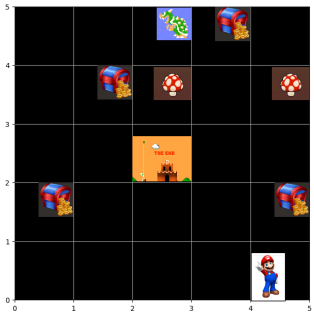
(z) Ts:10 R:+46



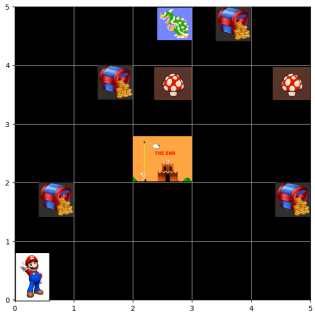
() Ts:11 R:+45



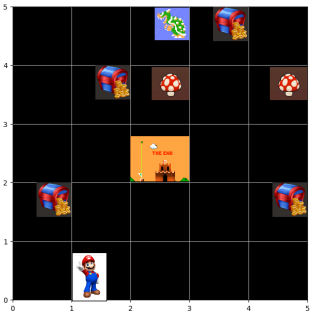
() Ts:12 R:+45



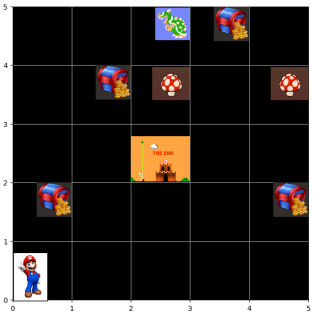
() Ts:13 R:+55



() Ts:14 R:+55



() Ts:15 R:+65



() Ts:16 R:+65

3 How did you define the stochastic environment?

In the current setup the Stochastic environment is designed in a way that the movement of the agent is random and uncertain using probability distribution. Also the immediate rewards disappear as the agent collects them.

4 What is the difference between the deterministic and stochastic environments?

1. Deterministic: When the Agent is in State (s) and performing an Action (a) the probability that it will reach state ($s\prime$) is either 0 or 1.

. $$P(s\prime, r|s, a) = \{0, 1\}$$

2. Stochastic: When the Agent is in State (s) and performing an Action (a) there is a randomness or uncertainty in which state the Agent will end-up.

. $$P(s\prime, r|s, a) = \{0, 1\}$$

5 Safety in AI

While defining the environment certain measures where taken in order to avoid agent running into error states while performing actions. For example, np.clip is used in environment definition to avoid agent going beyond the grid limit during any time-step. Taking care of such error states is very important because in real-world deployments it could lead to fatal incidents.

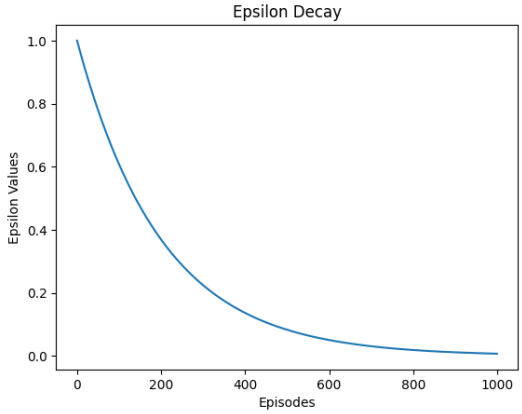
CSE546 - Reinforcement Learning

Assignment - 1 Part - 2

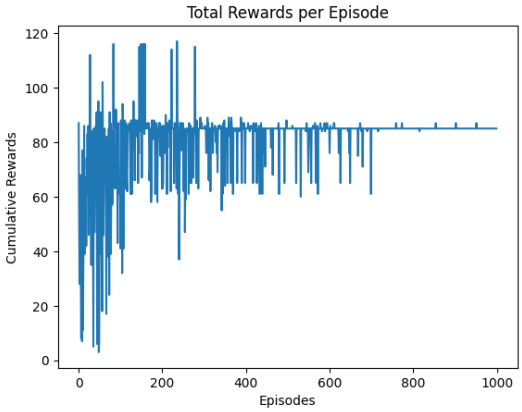
1 Applying Tabular Methods

1.1 Show and discuss the results after:

1.1.1 Applying Q-learning to solve the deterministic environment defined in Part 1. Plots should include epsilon decay and total reward per episode.

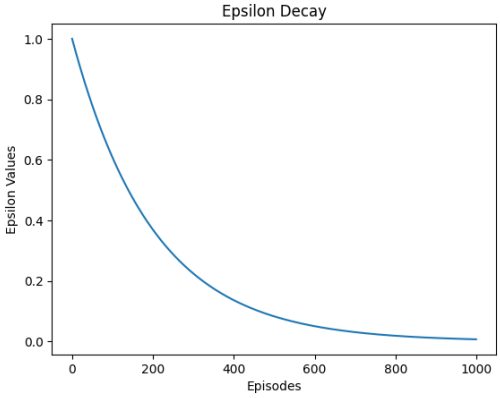


(a) Q-Learning: Epsilon Decay

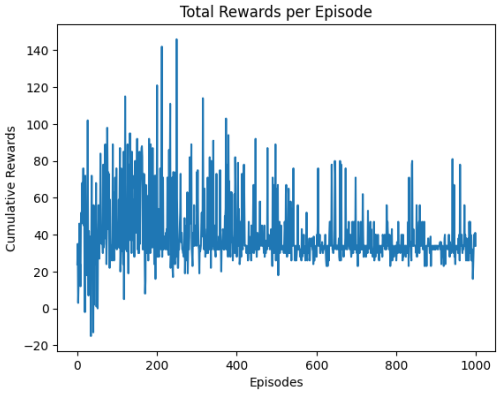


(b) Q-Learning: Total Rewards per Episode

1.1.2 Applying Q-learning to solve the stochastic environment defined in Part 1. Plots should include epsilon decay and total reward per episode.

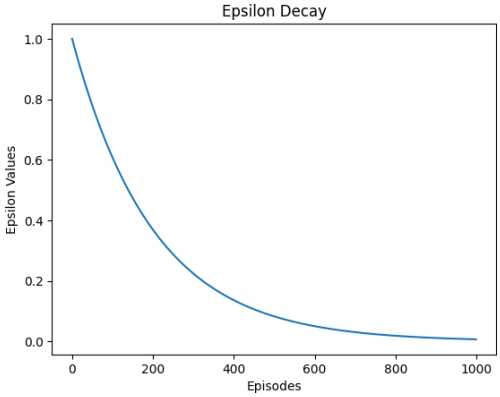


(c) Q-Learning: Epsilon Decay

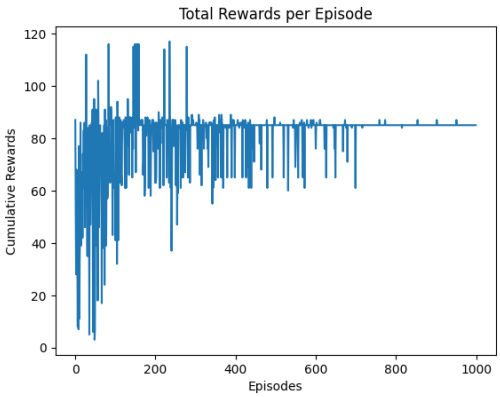


(d) Q-Learning: Total Rewards per Episode

1.1.3 Applying any other algorithm of your choice to solve the deterministic environment defined in Part 1. Plots should include total reward per episode.

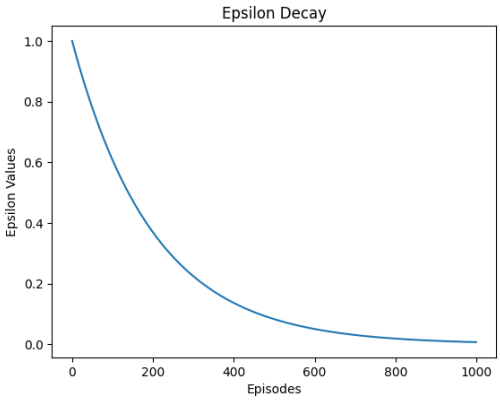


(e) SARSA: Epsilon Decay

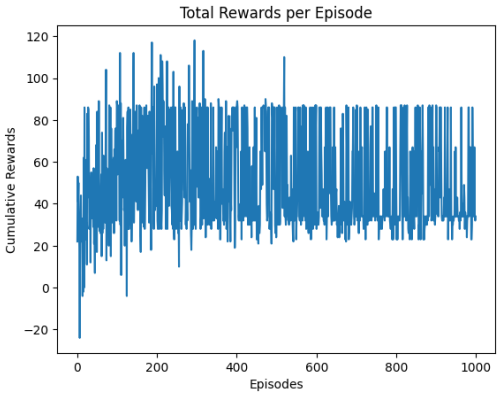


(f) SARSA: Total Rewards per Episode

1.1.4 Applying any other algorithm of your choice to solve the stochastic environment defined in Part 1. Plots should include total reward per episode.

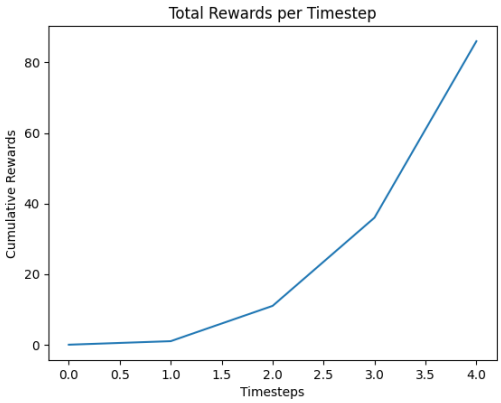


(g) SARSA: Epsilon Decay

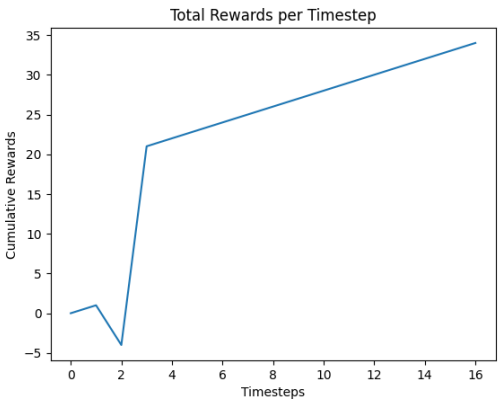


(h) SARSA: Total Rewards per Episode

1.1.5 Provide the evaluation results. Run your environment for at least 10 episodes, where the agent chooses only greedy actions from the learnt policy. Plot should include the total reward per episode.

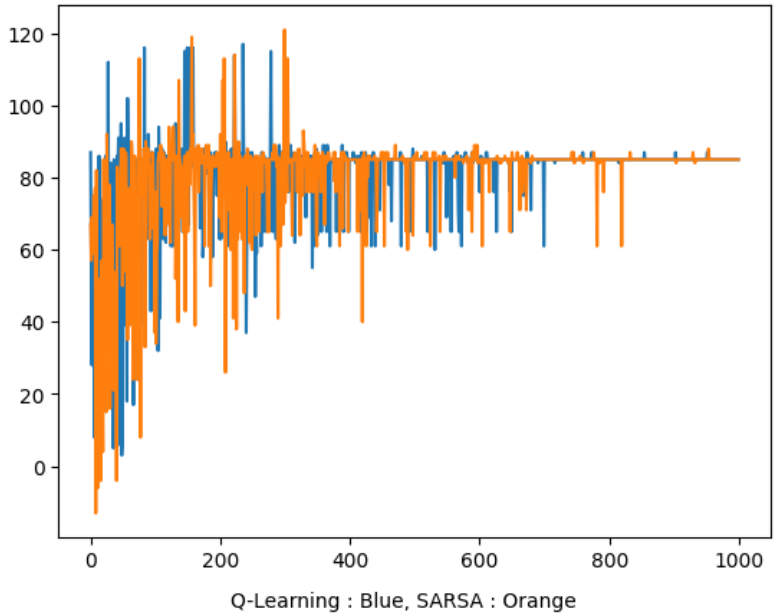


(i) Determinstic: Q-Learning Alg.



(j) Stochastic: Q-Learning Alg.

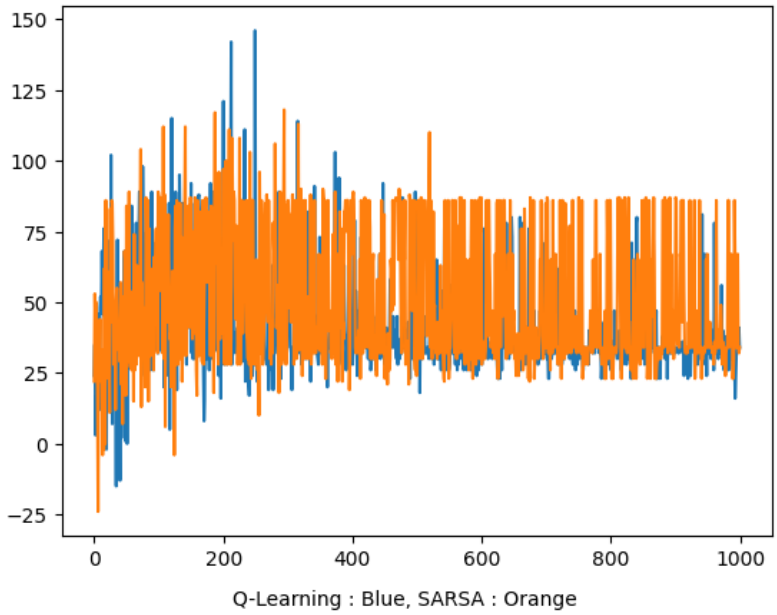
1.2 Compare the performance of both algorithms on the same deterministic environment (e.g. show one graph with two reward dynamics) and give your interpretation of the results.



(k) Determinstic: Q-Learning Alg. SARSA

At the time convergence, we can see that both SARSA and Q-Learning performed similarly.

1.3 Compare how both algorithms perform in the same stochastic environment (e.g. show one graph with two reward dynamics) and give your interpretation of the results.



(l) Stochastic: Q-Learning Alg. SARSA

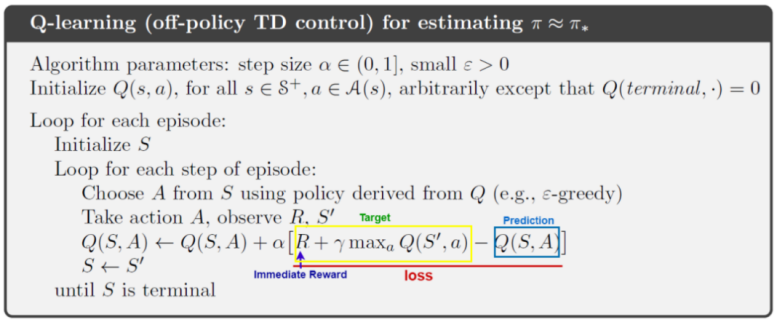
At the time convergence, we can see that both SARSA performed better than Q-Learning

1.4 Briefly explain the tabular methods, including Q-learning, that were used to solve the problems. Provide their update functions and key features.

Q-Learning: Is a model-free, off-policy TD method that does not require a model of the environment and can learn from actions that deviate from the optimal policy. It allows an agent to determine the best course of action based on its current state in the environment. The goal of Q-learning is to find the best action to take in a given state in order to maximize the future reward, and it does so by evaluating the quality of each action using a Q-value function.

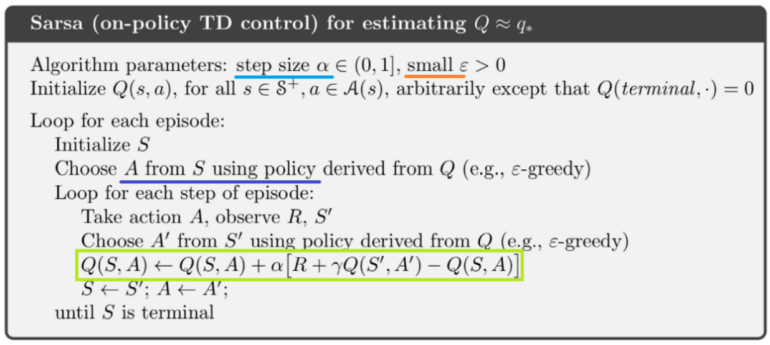
3 components of Q-Learning
1. State of the agent
2. Action that an agent takes when it is in a particular state
3. Reward that the agent will receive can be +ve or -ve, for each activity.

Reward function, $R : SXA- > R$



(m) Q-Learning: Update Function

SARSA: stands for State, Action, Reward, State, Action. It is a variation of the Q-learning algorithm that uses a modified update function. Unlike Q-learning, SARSA uses the State Action Reward State Action (SARSA) tuple to update its Q-value function. The updated Q-value is determined by the agent's immediate reward when transitioning from one state to another, as well as the next action chosen under the same policy. As a result, SARSA is considered a policy learning algorithm since the target policy and behavior policy are the same.



(n) SARSA: Update Function

1.5 Briefly explain the criteria for a good reward function. If you tried multiple reward functions, give your interpretation of the results.

In the current deterministic and stochastic environment cumulative reward function is used.

Once the rewards are collected they are removed from that state/position to avoid looping of agents.

Also a reward of -1 is given if the agent stays in the same position.

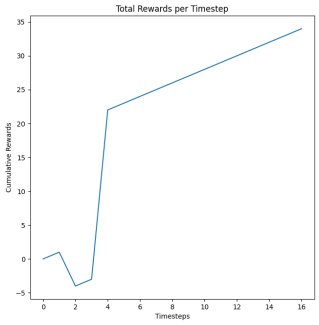
Hence by using an unambiguous reward function agent can collect maximum rewards.

1.6 Hyperparameter Tuning

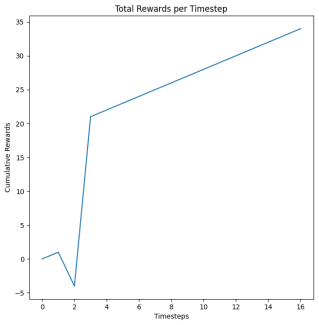
Among the Hyper Parameters used in Q-Learning, for the purpose of tuning discount-factor and learning rate are modified.

Stochastic Environment

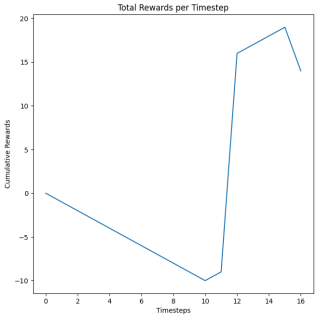
Tuning Discount Factor - 0.7, 0.8, 0.9



(o) Gamma=0.7

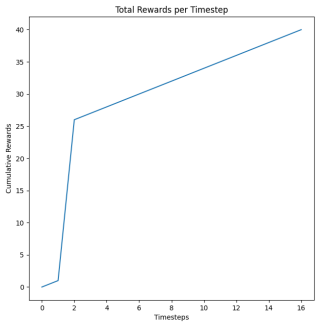


(p) Gamma=0.8

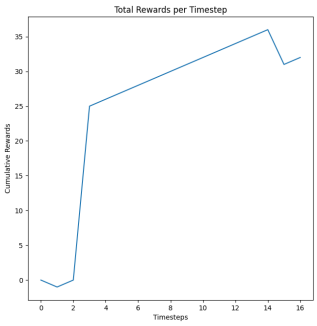


(q) Gamma=0.9

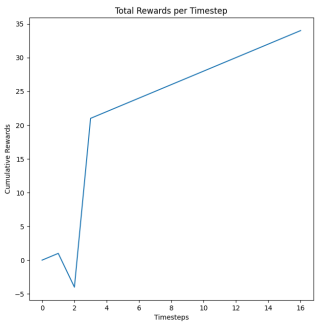
Tuning Learning Rate - 0.1, 0.15, 0.2



(r) alpha=0.1



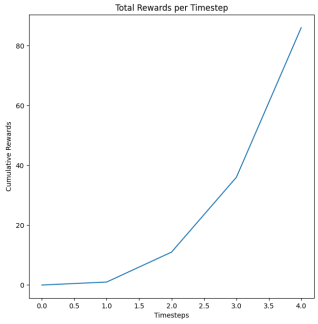
(s) alpha=0.15



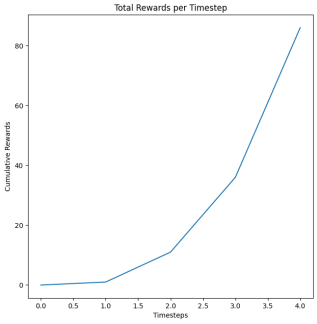
(t) alpha=0.2

Deterministic Environment

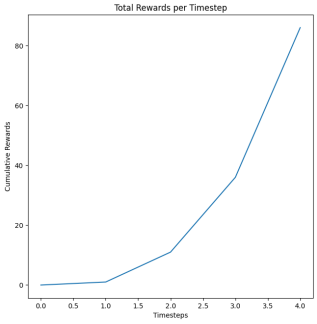
Tuning Discount Factor - 0.7, 0.8, 0.9



(u) Gamma=0.7

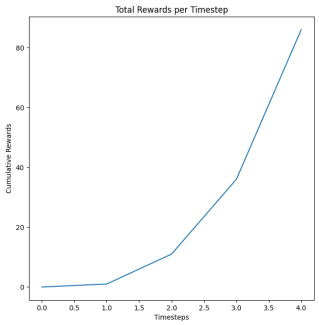


(v) Gamma=0.8

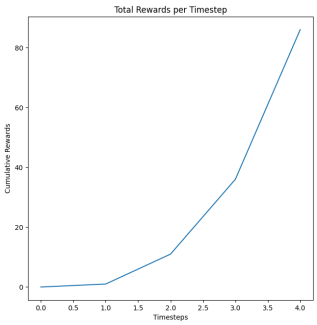


(w) Gamma=0.9

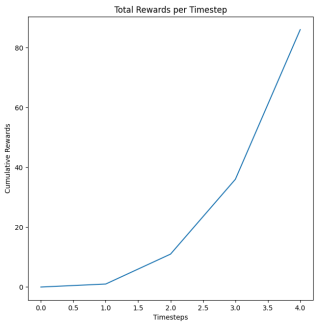
Tuning Learning Rate - 0.1, 0.15, 0.2



(x) alpha=0.1



(y) alpha=0.15



(z) alpha=0.2

CSE546 - Reinforcement Learning

Assignment - 1 Part - 3

1 Stock Trading Environment

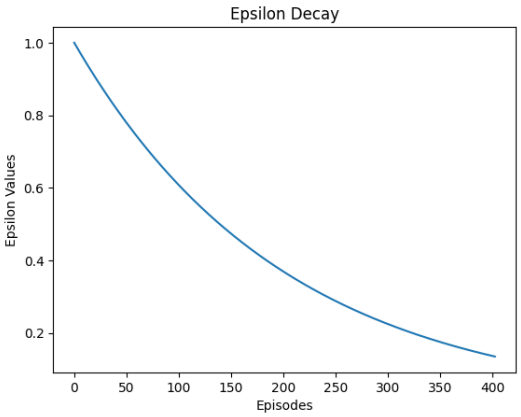
1.1 Show and discuss the results after applying the Q-learning algorithm to solve the stock trading problem. Plots should include epsilon decay and total reward per episode.

- Q-learning is a Off Policy Temporal Difference Algorithm. Using Q-learning we can find the optimal policy for the given Stock Trading Environment.
- Our goal is to use the Q-learning algorithm to find the optimal action-value function (Q-function) that maximizes the Total Account Value.
- Initially, exploration-exploitation is used to balance between exploring new actions and exploiting the learned Q-values to maximize outcomes.
- Figure(b) Exploration rate is controlled by a parameter called epsilon, which decays over time as the algorithm learns.
- Figure(a) depicts the Q-learning table which helps to find the optimal action-value function
- On comparing Figure(c) and Figure(d) we can that by following the optimal action-value function we were able to maximize our total stock or Account value

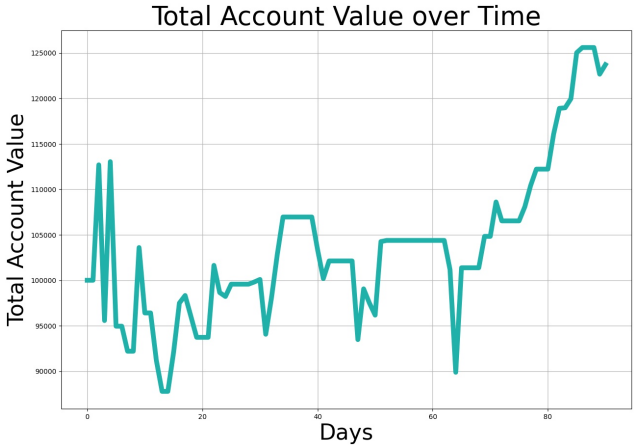
```
#Modifying learning rate gradually
for i in range(number_of_episodes): #number of episodes = 1000
    j = 100
    env.reset()
    observation=0
    #Updating learning rate gradually as the number of
        episodes increases
    if i%j == 0:
        hyper_parameters['alpha'] /= 10
    done = False
    while not done:
        if random.random() < hyper_parameters['epsilon']:
            action = random.randint(0,30)%3
        else:
            action = np.argmax(self.Q_Table[observation,:])

        nxt_observation, reward, done, truncated, info = env.
            step(action)
        self.update_table(observation,action,reward,
            nxt_observation)
        observation=nxt_observation
```

```
Q-Table:

[[20.76  5.32 14.45]
 [10.22 21.73 28.64]
 [15.28  2.56 11.78]
 [ 4.2  13.36 10.8 ]]
```

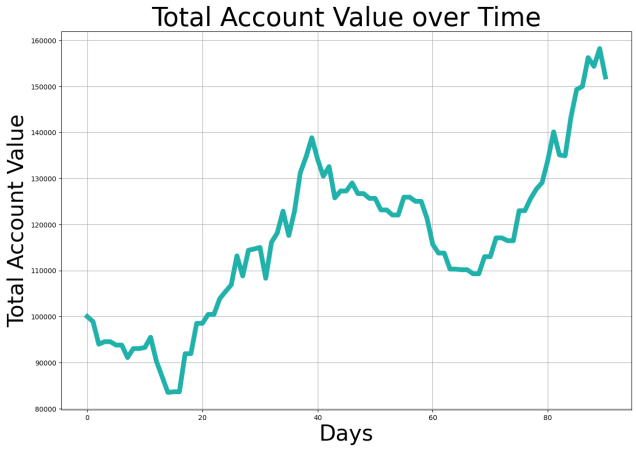
(a) Q-Learning Table



(b) Epsilon Decay



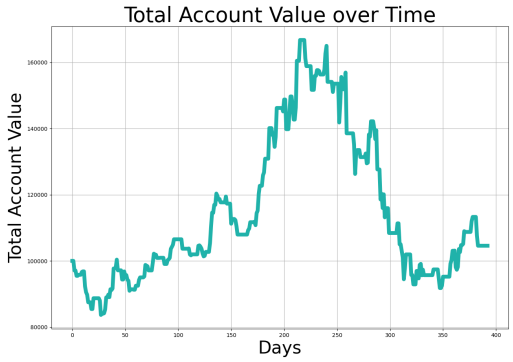
(c) Testing using Random Policy



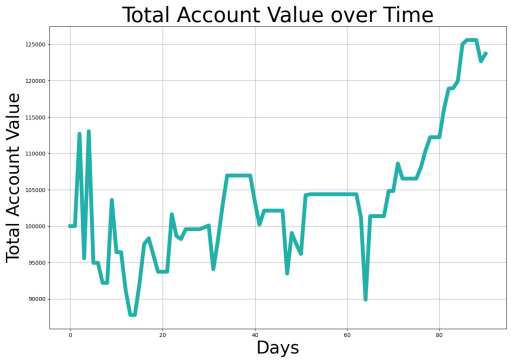
(d) Testing using Q-Learning policy

1.2 Provide the evaluation results. Evaluate your trained agent's performance (you will have to set the train parameter set to False), by only choosing greedy actions from the learnt policy. Plot should include the agent's account value over time.

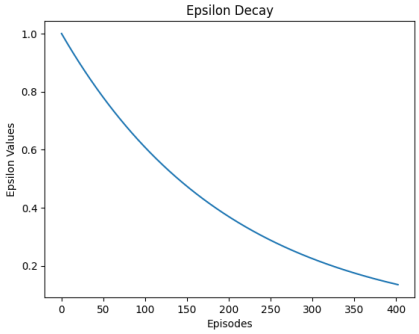
```
stock_trading_environment = StockTradingEnvironment('NVDA.csv', train=
    False,number_of_days_to_consider=10)
observation,info=stock_trading_environment.reset()
done = False
while not done:
    action = np.argmax(ql.Q_Table[int(observation),:])
    observation, reward, done, truncated, info =
        stock_trading_environment.step(action)
stock_trading_environment.render()
```



(e) Training using random policy



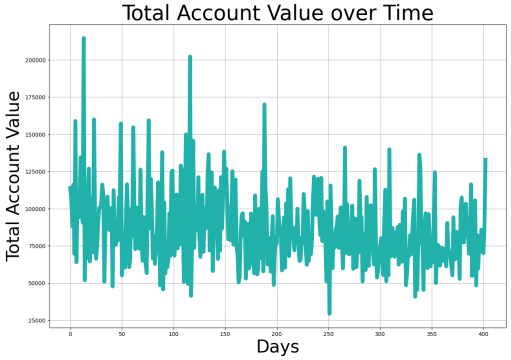
(f) Testing using Random Policy



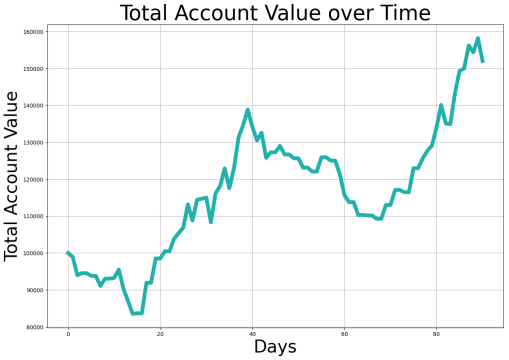
(g) Epsilon Decay

```
Q-Table:

[[20.76  5.32 14.45]
 [10.22 21.73 28.64]
 [15.28  2.56 11.78]
 [ 4.2  13.36 10.8 ]]
```

(h) Q-Learning Table



(i) Training using Q-Learning policy



(j) Testing using Q-Learning policy

2 Bonus Tasks

2.1 Git Expert [2 points]:

https://github.com/DivyaPallineni/CSE546_vpalline

2.2 CCR Submission:

Submitted Jupyter notebook that is executed on CCR

2.3 Grid-World Scenario Visualization:

As instructed visulizations are captured.

3 References

Resources shared in Piazza

1. spring_23_rl_lec_3_random_agent.ipynb
2. visualizing_rl_environments_and_representing_the_results

Report is prepared using NIPS template