CSE 4/546: Reinforcement Learning Spring 2022

Final Course Project

Group Members:

Affan Khan- 50432243
Hrishikesh Agrawal- 50428219
Ghulam Murtaza- 50419974

➔ Part 1: Setting up the environment

Description of the environment:

MAGridWorld is used as an environment for final project

Number of agents are: 2

Maximum Timesteps are: 16

Action space are 5 [“down”:0,”Up”:1,”right”:2,”left”:3,”No move”:4]

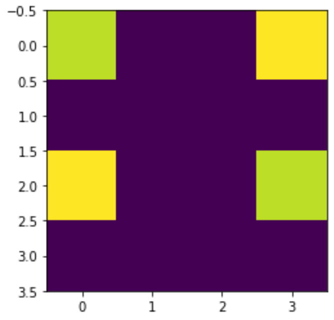
Goal at [0,3],[0,0]

For Deterministic Environment:

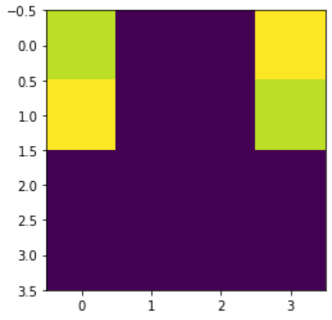
Actions taken are [1,1,1,0,1,3,4,1]

After execution:

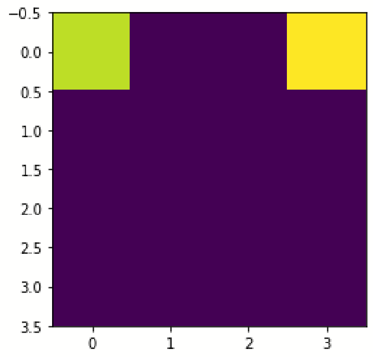
```
States are [[2, 0], [2, 3]]
Rewards are [-0.1, -0.1]
done [False, False] #agents did not reach the goal
```



```
Similarly executing again for same actions [1,1,1,0,1,3,4,1]
States are [[1, 0], [1, 3]]
reward is [-0.1, -0.1]
done [False, False] #agents did not reach the goa
```

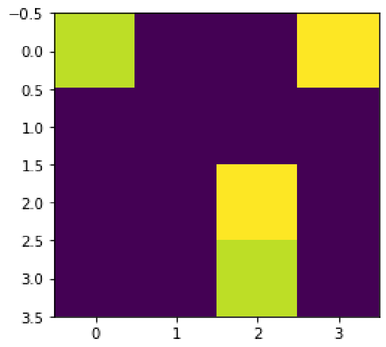


```
Thus, lets make the agents reach the goal by providing new actions after
above executions without resetting the environment again
list1= [2,2,1,2,4,4,4,4] #actions for agent1
list2= [0,0,3,3,3,1,1,1] #actions for agent2
```



```
state [[0, 3], [0, 0]]
reward [1, 1]
done [True, True] #both agents reached the goal
```

For Stochastic Environment:



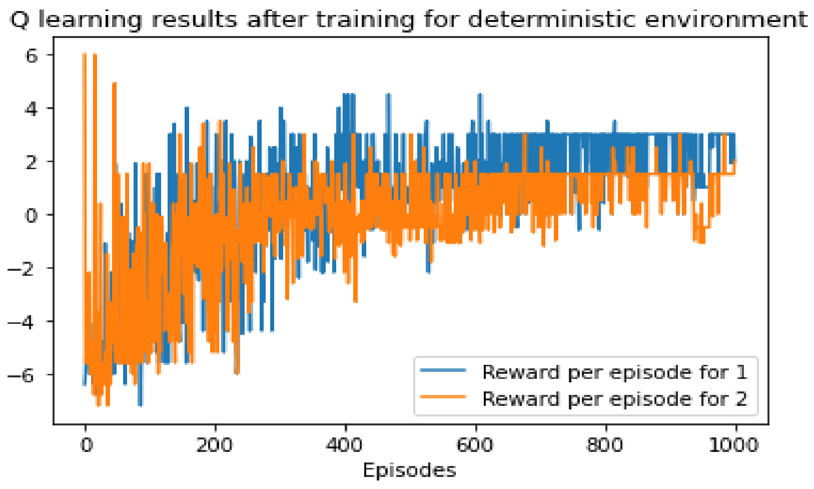
```
Reward:
[-0.5, -0.1]
[-0.1, -0.5]
[-0.5, -0.5]
[-0.1, -0.5]
[-0.1, -0.5]
[-0.1, -0.5]
[-0.1, -0.1]
[-0.5, -0.5]
[-0.5, -0.5]
[-0.5, -0.1]
[-0.5, -0.5]
[-0.1, -0.5]
[-0.5, -0.5]
[-0.5, -0.5]
[-0.5, -0.1]
[-0.5, -0.5]
```

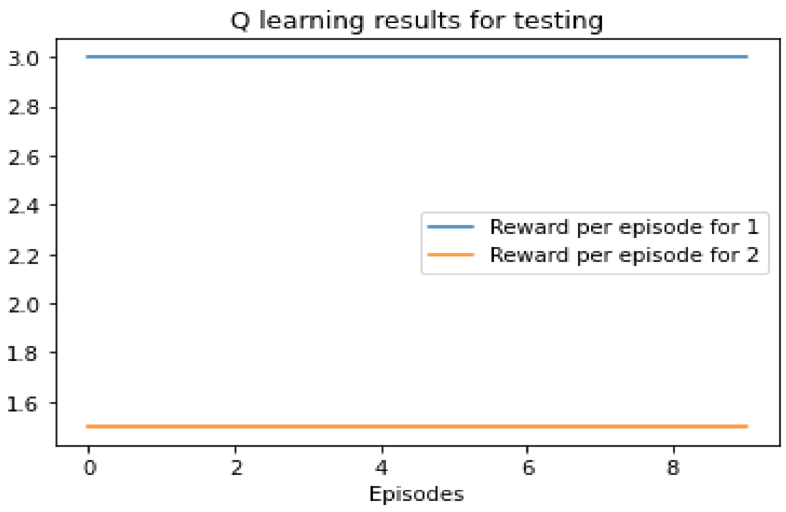
➔ **PART 2: Explore the existing baseline methods applied to solve it & Run it**

Method used is Q learning

Parameters are:

```
Alpha= 0.2
epsilon = 1
gamma = 0.9
epsilon decay = 0.9954
Epochs = 1000
```





- As we can see from the above figures, both the agent in the grid environment are successfully able to achieve the maximum rewards per episode as the training progress by achieving the optimal policy.
- The results while evaluation is also as expected, as by following only the learnt policy both the agents can achieve the maximum rewards.

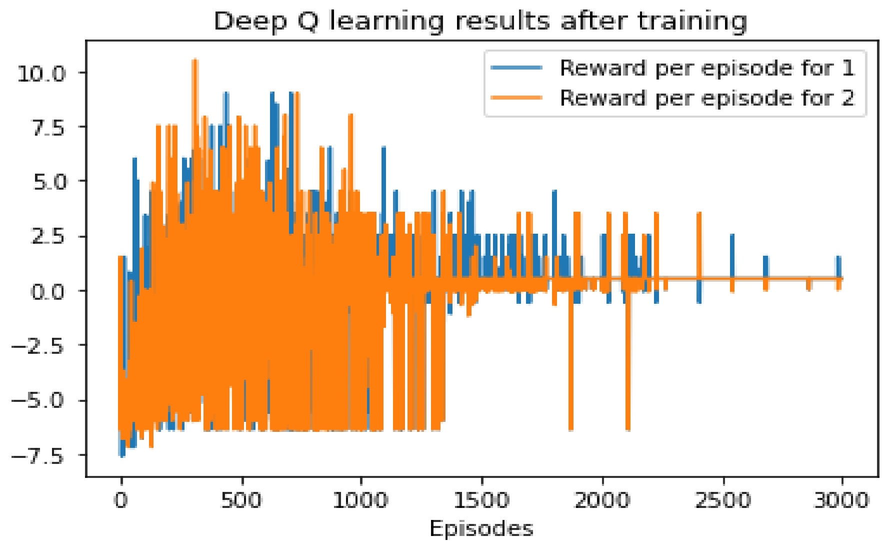
➔ **PART 3 Solve the environment using any deep RL methods (DQN, DDQN, AC, A2C, DDPG, TRPO, PPO, etc) and compare the results.**

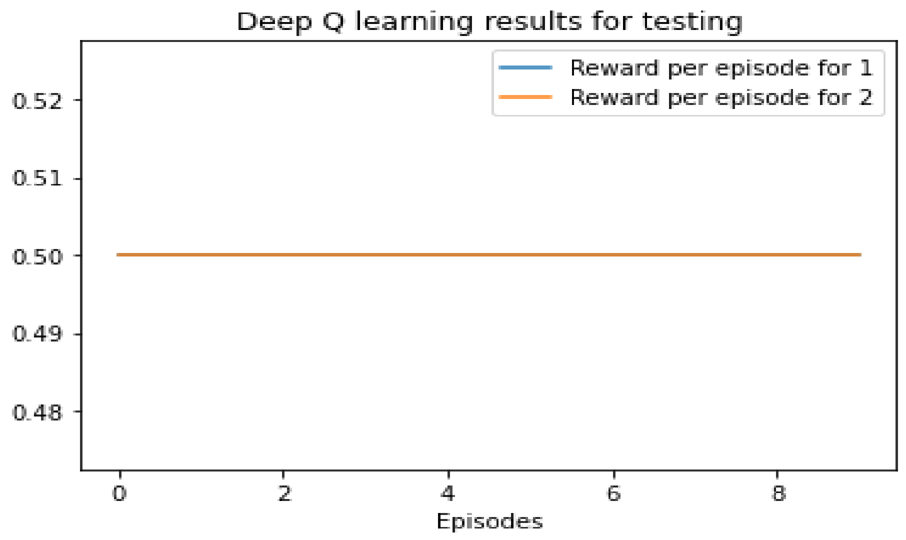
DQN model is used

```
Initial epsilon = 1
gamma = 0.99
epsilon_decay = 0.9977
episodes = 100
```

RESULTS:

```
0: [-4.0, -1.7000000000000002]
500: [-1.5, 1.9000000000000001]
1000: [-4.1, 0.0]
1500: [0.5, 0.5]
2000: [0.5, 0.5]
2500: [0.5, 0.5]
```





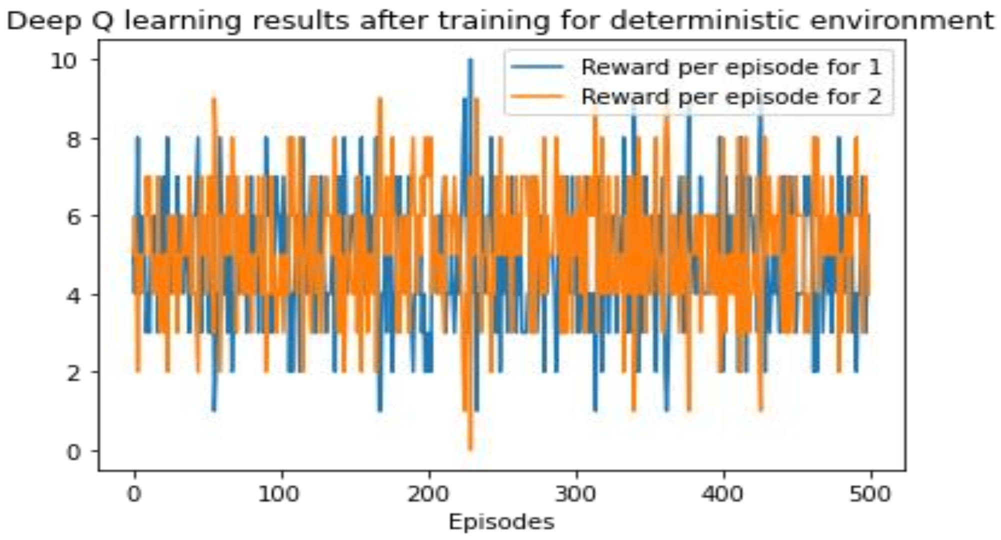
- Similarly, in DQN, as we can see from the above figures, both the agent in the grid environment are successfully able to achieve the maximum rewards per episode as the training progress by achieving the optimal policy.
- The results while evaluation is also as expected, as by following only the learnt policy both the agents can achieve the maximum rewards.
- The difference we saw when we trained the same environment using different policy is that, when trained by Q-learning algorithm the agents are maximizing their rewards by taking more no of steps. While, in case of the DQN algorithm the agents try to achieve the maximum rewards by taking less no of steps and they both reach the goal position at the same time thereby collaborating with each other better than in Q-learning.

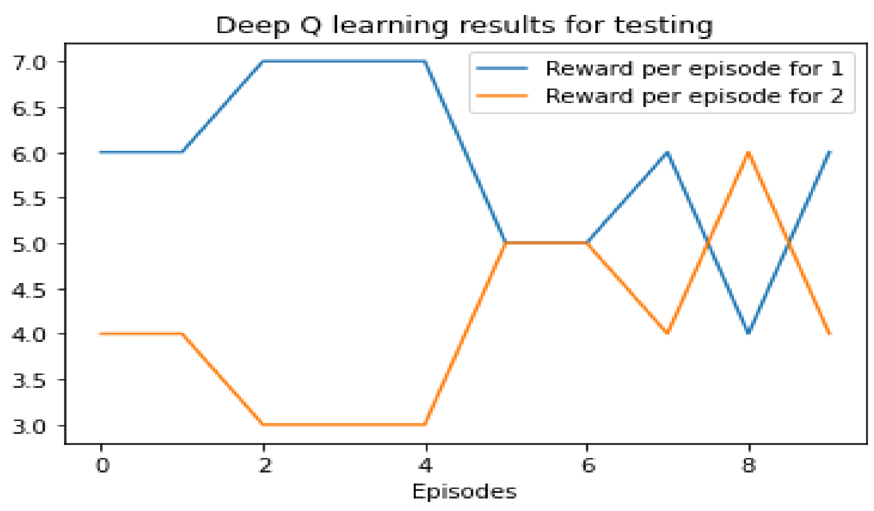
➔ **PART 4: Apply the algorithm from Part 3 to solve any of the existing MARL problems.**

ENVIRONMENT USED: PongDuel-v0

EPOCHS AND THEIR REWARDS

```
0: [5, 5]
50: [6, 4]
100: [5, 5]
150: [6, 4]
200: [3, 7]
250: [5, 5]
300: [4, 6]
350: [4, 6]
400: [2, 8]
450: [3, 7]
```





- So, for part 4 of our project, we were asked to perform the previously used algorithm that is DQN in an existing MARL environment, for this part we are using Pong duel.
- In the environment there are two levers playing pong against each other while competing against each other and win.
- Before training the agents, when they only took random actions, we saw that their was no fixed winner and the difference between the scores was also differing a lot.
- But after we trained the agents by DQN, we noticed that there's still no fix winner but after training both the agents try to win and the score difference gets significantly low, as they both learn optimal policy and they get almost the equal scores with a very low difference in score sometimes and even the same scores sometime.

➔ REFERENCES:

- **https://github.com/koulanurag/ma-gym**
- **https://github.com/LantaoYu/MARL-Papers**
- **Lecture slides.**