Homework 2: Autonomous Agents

I. Introduction

The problem in question is to establish a sequence of actions for agents to reach the targets present in a 5 by 10 gridworld. The goal of this project is to observe the behaviour of the agents by varying the reward function.

II. Algorithm

Q-learning is used for this project. The q table is stored as a dictionary of dictionaries. Each element of the q table is in the form:

$$self.q_table[(x,y)] = 'UP' : value, 'DOWN' : value, 'LEFT' : value, 'RIG \quad (1)$$

The initial rewards for each state action pair in the q table is zero. With probability less that epsilon, the agent chooses to explore and randomly picks an action. The value of epsilon considered in 0.05. For probability greater that epsilon, the agent chooses the action with the maximum value for that state, from the q table. If multiple maximum values exist, it randomly chooses an action. The agent then takes the action, and the reward of that action is updated according to Q-learning.

$$Q(S,A) \leftarrow Q(S,A) + alpha * (R + gamma * maxQ(S',a) - Q(S,A)) \quad (2)$$

The value of gamma is taken as one, which means that reward is decided based on next action only, effect of future actions on reward is not considered. The learning rate(alpha) taken is 0.1.The reward is -1 for every action that does not lead to the target state. If the agent reaches the wall, it remains in that state and additionally takes a reward of -1.

After every action the q table is updated. The algorithm terminates when either the target is reached or the number of steps exceeds the upper bound. An upper bound of 1000 steps was used for this project.

III. Part 1- Results

For this scenario, Q-learning is used for one agent and one target. The agent is able to reach the target.

One class of Agent and one class of Gridworld is used in this. The trials are run in the Agent class. Figure 1 shows the results of the q learning algorithm for single agent- single target scenario. The left column shows the trial number and right column value shows number of steps taken to reach the target.As the number of trials increases, the number of steps taken to reach the target gradually decreases, barring some outliers.

No. of trials	No. of steps
1	47
2	18
3	2708
4	3
5	5
6	8
7	5
8	3
9	5
10	3
11	3
12	3
13	5
14	3
15	3

Fig. 1. Single target result

IV. Part 2- Results

For this scenario, two agents and two targets are used. Each agent gets an individual reward based on the rewards it individually captures. Both the agents are more likely to go to target one. A few times, one the agents reaches target 2. It is observed that the agents do not benefit from each other as each agent seeks to maximize its own reward.Each trial is stopped only when either agent reaches a target or the maximum limit on number of steps is exceeded.

To introduce the second agent, I've used two classes of agents, one for each. Only one class of agent is used. The trials are run with both the agents in a separate class known as $Game_play$. Figure 2 shows a sample result which is the output of the terminal. Agent 1 has reached target 1 as in the second matrix in the figure. It can be seen from the figure that both agents receive their own individual rewards.

```
Current position of the agent = (3, 1)
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 2. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
Current position of the agent1 = (3, 1)
Current position of the agent2 = (1, 1)
number of steps= 8
reward1= -1.0
reward2= 20.0

Process finished with exit code 0
```

Fig. 2. Single reward result

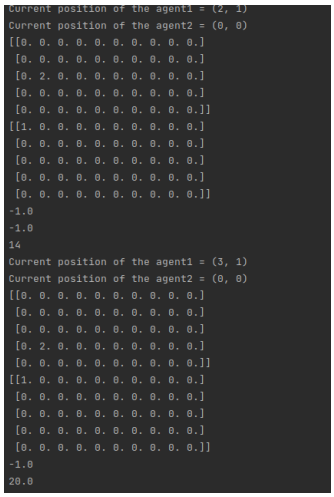

Fig. 3. Single reward agent cooperation

Figure 3 shows that both agents are not benefiting from each others behaviours. Both agents are trying to go to the same target, that is target 1 to maximize their individual rewards. As a result they are not maximizing the total reward of system.

V. Part 3-Results

For this scenario, a global reward is set for both the agents. Both the agents get an equal reward when a target is achieved. For each action of any agent, when the target is not reached a -1 reward is received by both agents. If a target is reached by one, both agents receive the same reward of +20. Each trial is stopped only when either agent reaches a target or the maximum limit on number of steps is exceeded. It is observed that agent 1 always tends to go towards target 2 and agent 2 moves around target 1.

To introduce a global reward system, I have modified the way the rewards are updated for both agents by including the other agents rewards for each agent. That is, agent 1 acquires its own rewards and also the rewards of agent 2, and vice-versa. Figure 4 shows that agent 1 has reached target 2. With the global reward assignment system, it is observed that the agents are more likely to reach target 2. It can be seen from the figure that both agents have received the same reward.

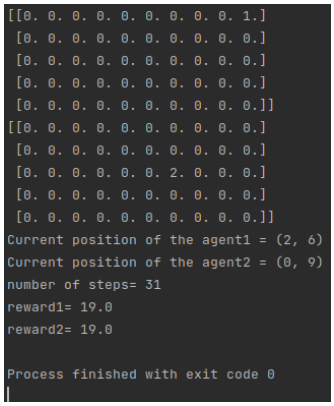

Fig. 4. Global reward result

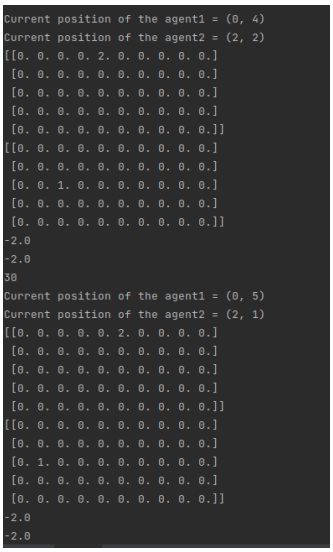

Fig. 5. Global reward cooperation result

Figure 5 shows both the agents are benefiting from each others behaviours.As seen in this figure, agent 1 is trying to move towards target 2 whereas agent 2 is trying to move towards target 1.

VI. Analysis

The q-learning algorithm used for this project is able to reach the targets. Part 1 results show that as the number of trials (training sets) are increased,the number of steps required to reach the target shows a generally decreasing trend barring a few outliers. This confirms that the agent learns from its behaviour.

A second target and agent is introduced in part 2 and 3. As seen in part 2 results, when individual rewards are assigned to agents, both the agents seek to maximize their own rewards. They do not benefit from others behaviours. Both agents tried to move to the nearest target, which is target 1. There is no cooperation.

Part 3 results are interesting. As a global reward assignment is used, both the agents benefit from each others behaviours. One agent tries to move to target 2 and the other moves towards target 1. The agents cooperate implicitly. The cooperation is intentional, as bad moves for any one agent results in negative rewards (-1)being assigned to both the agents. This system performs better as agents are more likely to reach both targets as seen from the results.

VII. Note

$q_learning.py$ contains my code for part 1. $sep_reward.py$ contains my code for part2. $experiment.py$ contains my code for part 3.