CSE 546 - Reinforcement Learning
Project Checkpoint Report

Arthav Ashok Mane
UBID: 50426537
arthavas@buffalo.edu

Indra Teja Pidathala
UBID: 50478945
indratej@buffalo.edu

Abstract

This document is presented as part of the final project checkpoint submission for the course CSE546: Reinforcement Learning, taught by Prof. Alina Vereshchaka in Fall 2022.

Topic: Multi-agent Reinforcement Learning

1 Objective

The main objective of this project is to study multi-agent reinforcement learning methods and solve an environment with multi-agent reinforcement learning. We plan to study how different reinforcement learning techniques such as tabular methods (eg: Q-Learning) and deep RL methods (eg: DQN) perform while solving an environment with multiple agents.

Project Management Tool: `https://rl-project-ub.atlassian.net/jira/software/projects/RLP/boards/1`

2 Multi-Agent GridWorld Environment

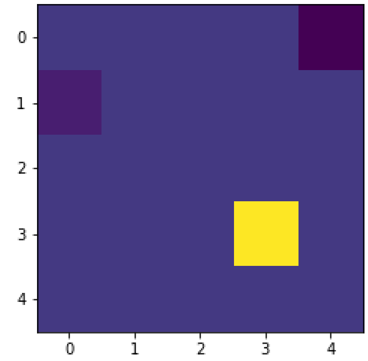


Figure 1: Render of the defined grid world environment with 2 agents (dark blue) and 1 target (yellow)

For this environment, we've created a 5x5 grid world environment. The number of agents chosen is 2, and each agent can choose to perform one of 4 actions: up, down, left and right to move around in the grid. The agent and target position have been fixed. Fig. 1 shows a render of this environment. The goal of the agents in this environment is to reach the yellow target as soon as possible. A render of the different rewards obtained for a state is shown in Fig. 2. Each step yields a reward of -0.1 and on hitting the boundary of the grid yields a reward of -1. On being diagonally adjacent to the target we get a reward of +2.5 whereas on being directly adjacent we get a reward of +5. Reaching the target yields a reward of +20.

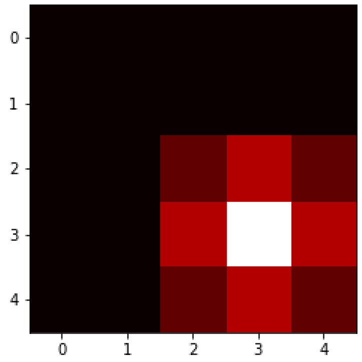


Figure 2: Different rewards for each states in the above mentioned grid environment. Rewards are black = -0.1, maroon = 2.5, red = 5 and white (target) = 20

3 Tabular Method: Q-Learning

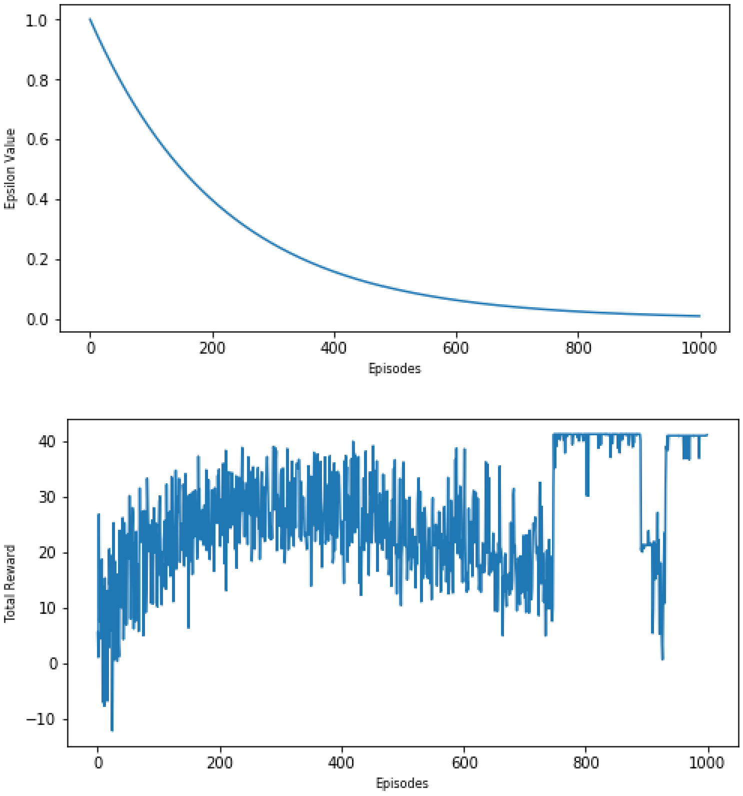
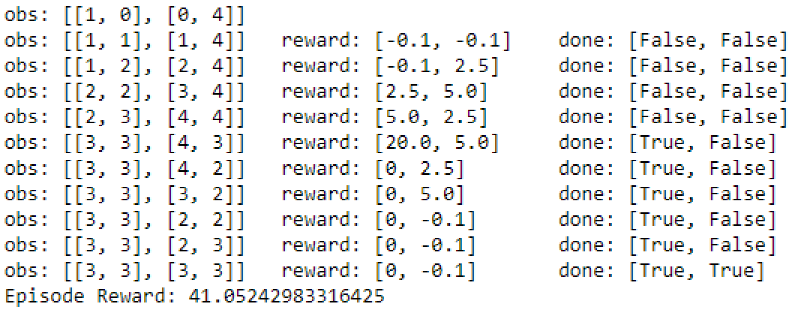


Figure 3: Solving the above mentioned grid environment using Q-Learning. (above) Epsilon decay during training. (below)

For the tabular method, we've used Q-Learning to solve the multi-agent environment. Both the agents share the same Q table in this method to collaboratively find the optimum policy to reach the goal the quickest. During training, the agents followed the epsilon-greedy policy to choose actions to perform. Fig. 3 shows the epsilon decay during training and the discounted cumulative episode reward obtained for each episode during training. After about 800 episodes, the Q table seems to have the reached the optimum Q* policy.



```
obs: [[1, 0], [0, 4]]
obs: [[1, 1], [1, 4]]   reward: [-0.1, -0.1]    done: [False, False]
obs: [[1, 2], [2, 4]]   reward: [-0.1, 2.5]     done: [False, False]
obs: [[2, 2], [3, 4]]   reward: [2.5, 5.0]      done: [False, False]
obs: [[2, 3], [4, 4]]   reward: [5.0, 2.5]      done: [False, False]
obs: [[3, 3], [4, 3]]   reward: [20.0, 5.0]     done: [True, False]
obs: [[3, 3], [4, 2]]   reward: [0, 2.5]        done: [True, False]
obs: [[3, 3], [3, 2]]   reward: [0, 5.0]        done: [True, False]
obs: [[3, 3], [2, 2]]   reward: [0, -0.1]       done: [True, False]
obs: [[3, 3], [2, 3]]   reward: [0, -0.1]       done: [True, False]
obs: [[3, 3], [3, 3]]   reward: [0, -0.1]       done: [True, True]
Episode Reward: 41.05242983316425
```

Figure 4: Evaluation of the Q-learning policy learned for the grid environment.

The learned policy is evaluated and the results are shown in Fig. 4. As seen, the agents reach the target quickly with the maximum possible reward.