Reinforcement Learning lab report

DIYUAN DAI, EID: DD33653,

Department of Computer Science, The University of Texas at Austin

1 1. INTRODUCTION

Q learning uses a traditional algorithm to create a Q-table to help the agent find the following action to be taken. The most significant difference between supervised learning and unsupervised learning is that there is no given set of data for the agent to learn in reinforcement learning. The environment is constantly changing, and the reinforcement learning agent must make a series of action decisions in a changing climate, not a certain action decision. Combining a series of decisions is a strategy, and reinforcement learning is a process of updating strategies through continuous interaction with the environment.

In this assignment, the goal is to implement an agent that can walk down a sidewalk that contains litter to be picked up and obstacles to be avoided. I use separate reinforcement modules, one for litter, one for obstacles, and one for staying on the sidewalk in a 6x25 grid world. Different value is used in the map as different objects: obstacles (-10), litter(5), sidewalk(2), final(20)

2 2. Q-LEARNING ALGORITHM

In the algorithm of Q learning, we use Q reality and Q estimation for each update, and the fascinating thing about Q learning is that in the reality of Q(s1, a2), it also contains a maximum estimate of Q(s2). The maximum estimate of the decay of the next step and the current rewards are regarded as the reality of this step. It's amazing. Finally, let's talk about the meaning of some parameters in this algorithm. Epsilon greedy is a strategy used in decision-making. For example, when epsilon = 0.9, it means that in 90% of the cases, I will choose the behavior according to the optimal value of the Q table, and use the random selection behavior 10% of the time. Alpha is the learning rate to determine how much of the error is to be Learned, alpha is a number less than 1. Gamma is the attenuation

$$Q^*(s,a) = \sum_r rP(r|s,a) + \gamma \sum_{s' \in S} P(s'|s,a) \max_{a'} Q^*(s',a')$$

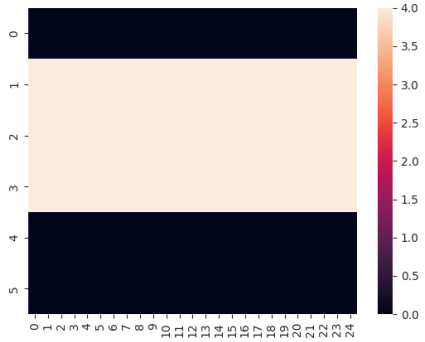
value of future rewards.

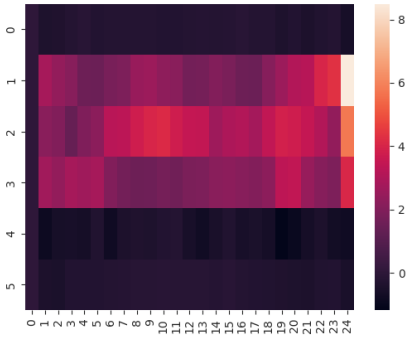
While $Q*$ stands for new Q-value. α stands for learning rate, γ stands for discount rate and Σ part means to select the action which will give the biggest reward and change state from s to s'.

I used seaborn package to plot the map in grids and draw heatmaps.

3 SIDEWALK MODULE

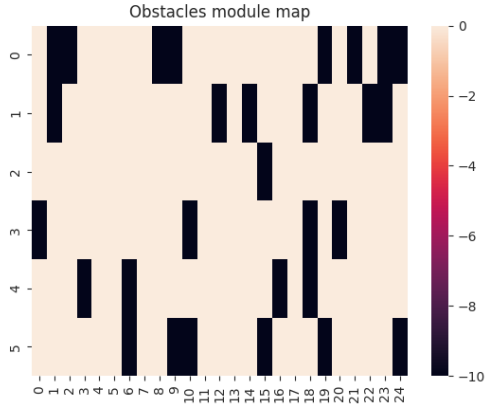
According to the problem statement, I set 1,2,3 row as sidewalk and provide them with a reward value 3, and the rest of the map is 0. Since there are no final goals in this map, I set a 1000 steps train and the agent learned pretty well to step on the sidewalk.

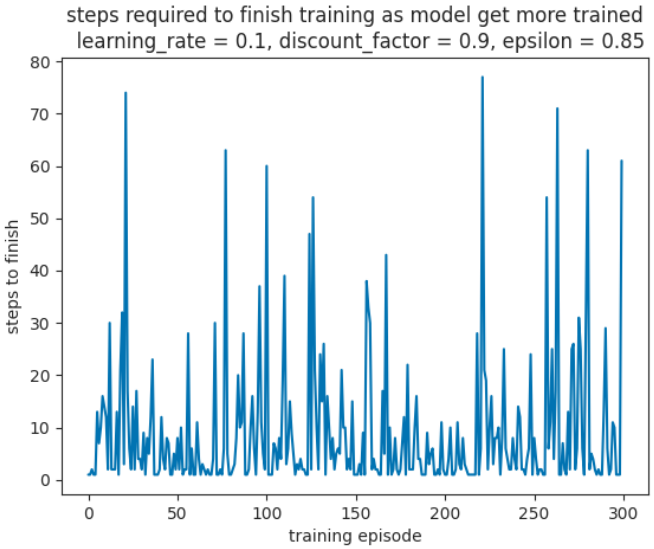


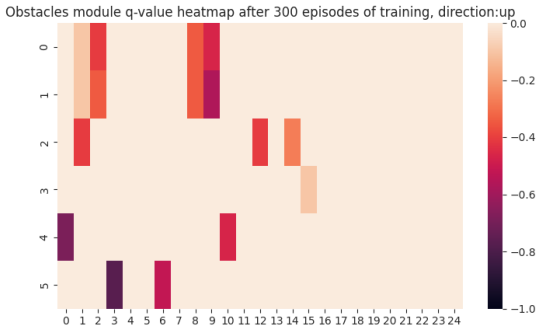


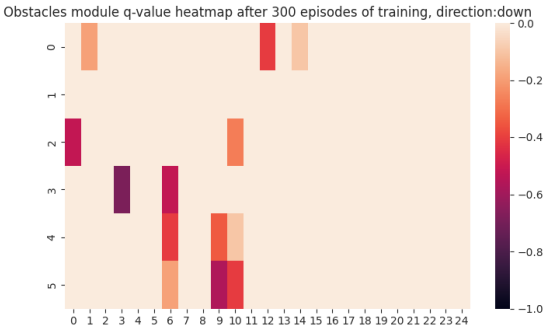
4 OBSTACLE MODULE

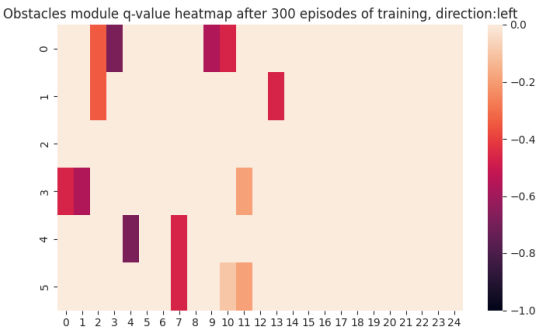
According to the problem statement, I randomly picked some grids to be Obstacle by setting the reward value on that grid to be negative value and the rest of the map is 0. Once the agent steps into an obstacle, a penalty is applied and the agent will keep the training until it takes 500 steps in this train. We can clearly see that the heatmap is avoiding all the Obstacle after training.

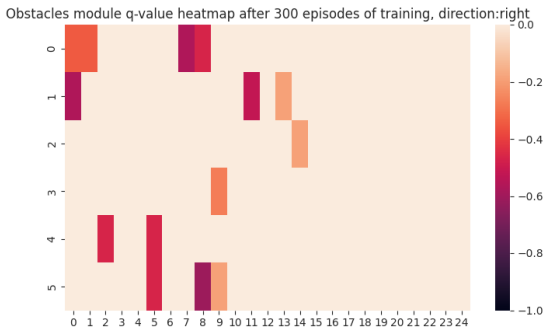






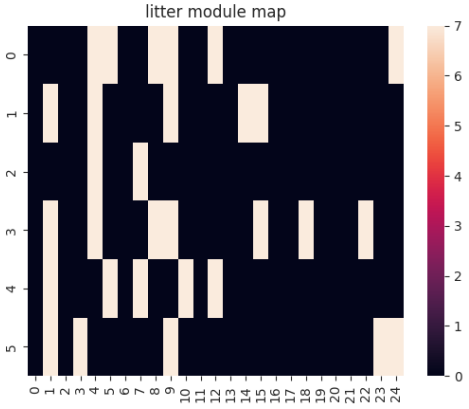


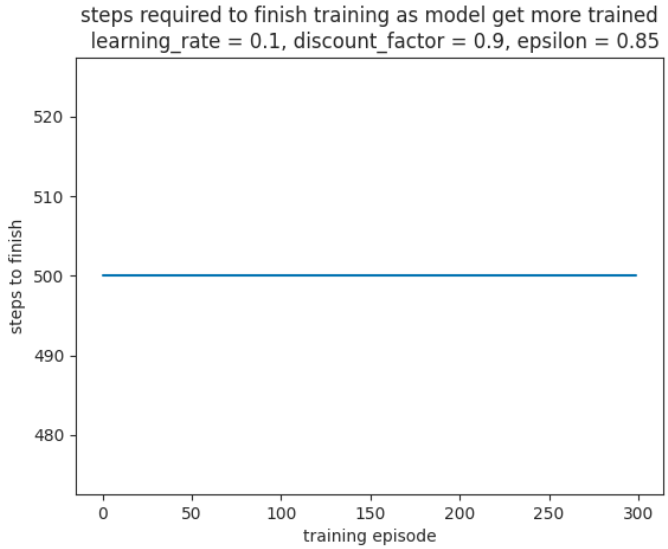


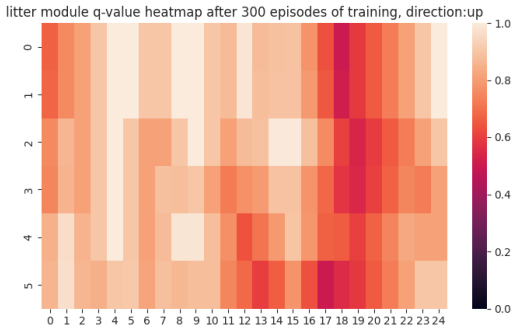


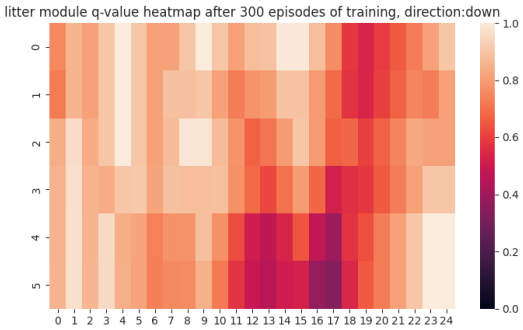
5 LITTER MODULE

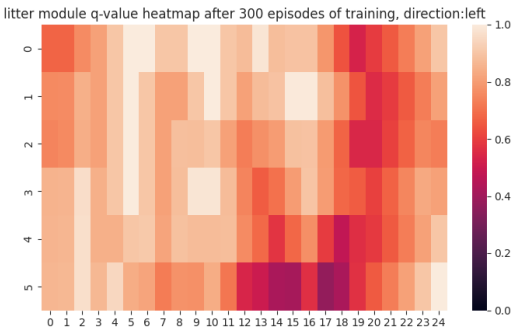
According to the problem statement, I randomly picked some grids to be littering plots by setting the reward value on that grid to be value 5 and the rest of the map is 0. Since the training for one episode will not terminate automatically, I set the maximum step length to be 500 and after 300 episodes of training, we can clearly see that agent is picking up litters and gain high rewards.

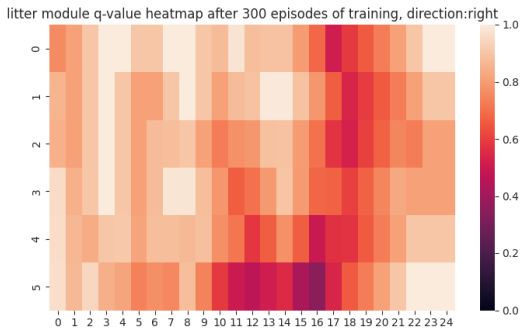






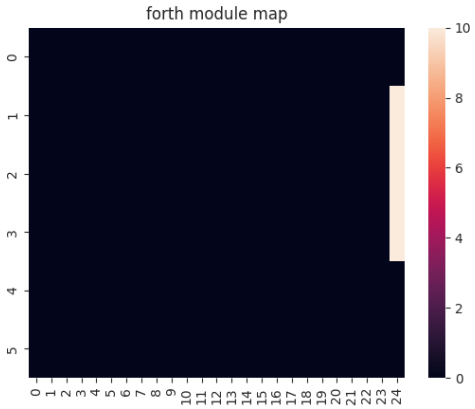


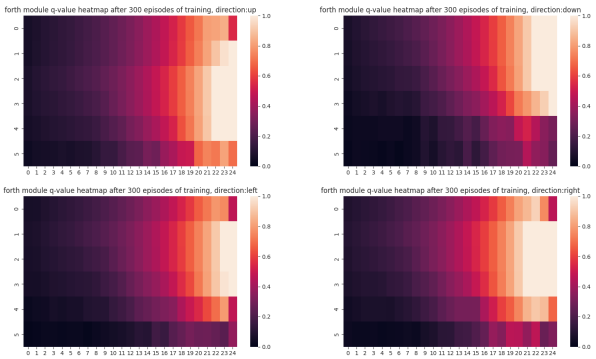


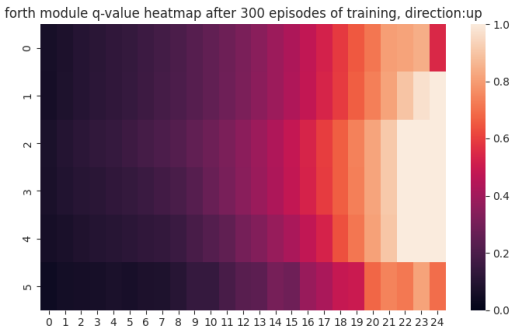


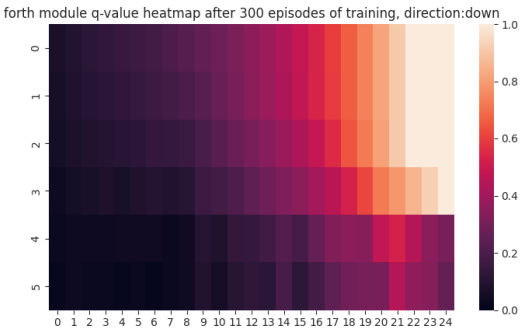
6 FORTH MODULE

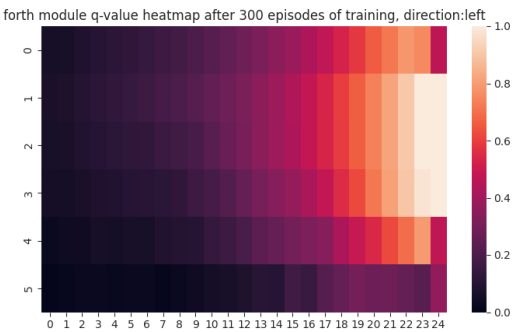
As the way the lab assignment is asked to be, I picked the right most grids to be final goal of the grid world by setting the reward value on that grid to be value 10 and the rest of the map is 0. The training will automatically complete and reset the position of agent once it hits the final, we can see how Q value is propagated and as the agent become more and more experienced, it will take less and less time for the agent to hit the goals.

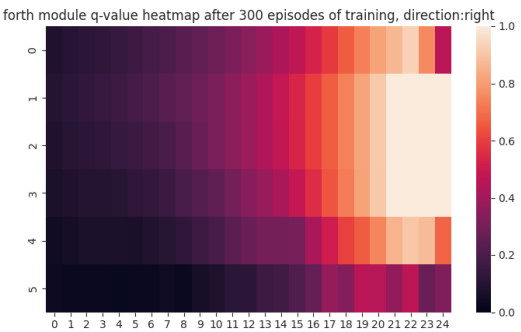






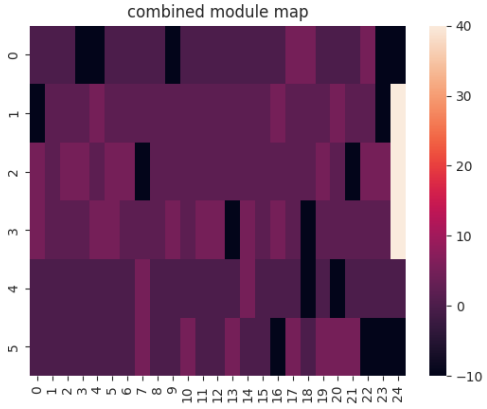


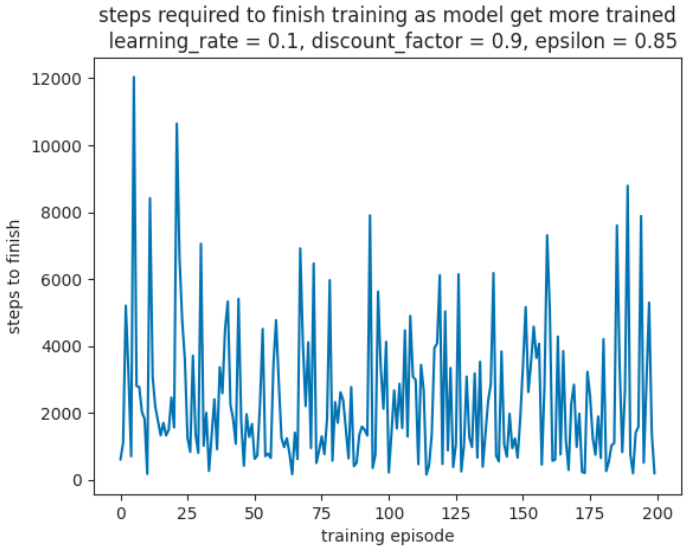


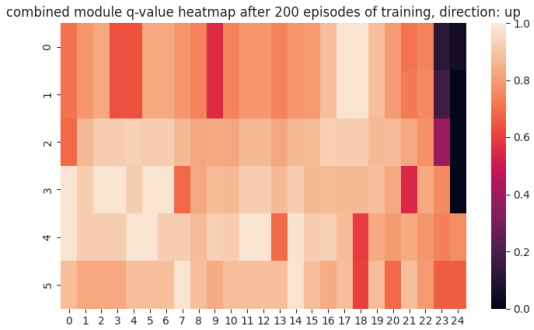


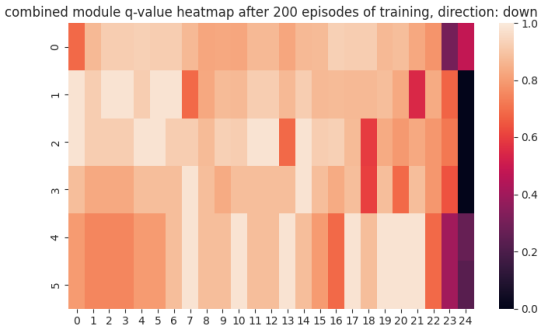
7 COMBINED MODULE

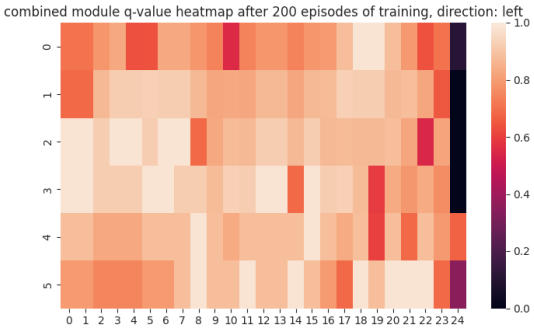
Finally, I combined all the module and plot a world with all the objects: obstacles (-10), litter(5), sidewalk(2), final(20). As the situation become more complex, it takes much longer time for the agent to complete training. since the agent is very experienced now and there are more objects providing good rewards, agent can reach relatively higher reward value in this environment.

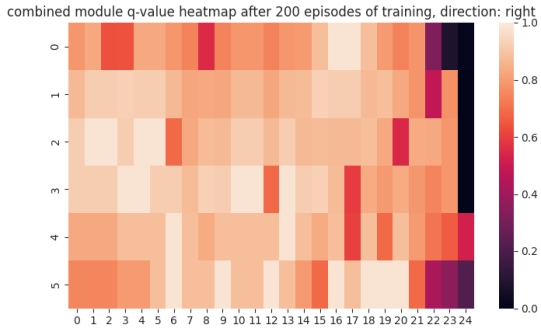












8 SUMMARY

Artificial intelligence and machine learning are big concepts, and there are three major categories of machine learning, supervised learning, unsupervised learning and reinforcement learning. Reinforcement learning can be divided into three types of algorithms, value base algorithm, policy base algorithm and model base algorithm. Q-learning we are going to talk about today is a value base algorithm.

In general, Q-learning is a good algorithm for solving a model-free RL issue. When the agent has a limited amount of actions to take in a given environment, it functions effectively. Furthermore, it is effective in segmenting the problem into distinct modules that may be fine-tuned and linearly merged.