ACML - Reinforcement Learning

Dimitrios Gagatsis(6258651), Christos Kaparakis(6258164)

November 24, 2020

1 Introduction

This report aims to inform the user of the implementation of a Reinforcement Learning algorithm on the Mountain Car problem.

2 How to use

Execute the python file **qmountaincar.py**.

It will automatically start training using the Q-Learning algorithm. Specific parameters, such as **number of episodes** to run the algorithm *(default 10.000)*, how frequently we want to **render** an episode *(default every 1000 episodes)*, the γ **discount factor** *(default 0.995)* can be changed. Furthermore, the default options for saving the Q-table, or showing graphs can be changed as well.

3 Reinforcement Learning Algorithm

We chose to use the Q-Learning algorithm because this seeks to find the best action to take given the current state.

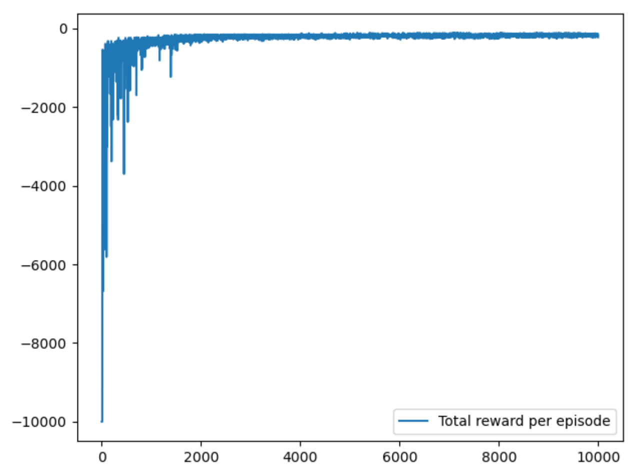
3.1 Q-Learning

We chose the Q-Learning algorithms because of it's simplicity which makes it easy to understand and because of it's use of the Q-table The Q-learning seeks to learn a policy that maximizes the total reward. The q-learning function learns from actions that are outside the current policy. More specifically, the q-learning function takes random actions from the Q-table we defined using the parameters of the environment. The parameters are the speed and position of the vehicle. The positions values are from -1.2 to 0.6 and the speed values are from -0.07 to 0.07.

3.2 Implementation

As explained in the previous section, Q-learning uses the Q-values (Q stands for quality) that are stored in a table (the Q-table). The Q-values represent how important an action is for the agent for every combination of our parameters. Since the range of the parameters are -1.2 to 0.6 for the position and -0.07 to 0.07 for the speed, we need to make the state space discrete. After experimentation, we decided to use 100 random values of the speed and the position to create the q-table with 100*100 dimensions (more specifically we will mention below).

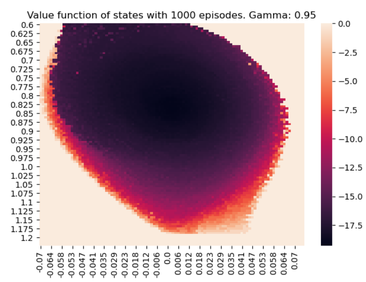
The discretization of the state space will play an important role for the visualization later.



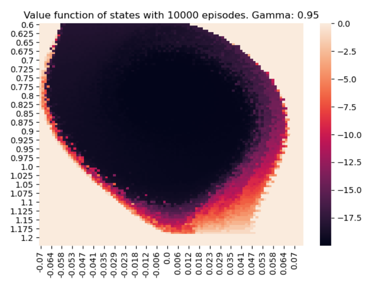
The above plot describes the learning performance of the algorithm as the episodes progress.

4 Visualization

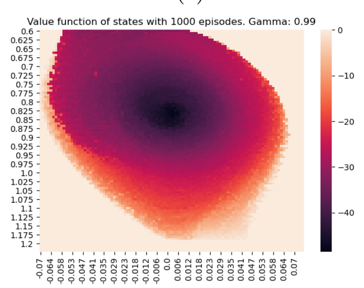
After a number of episodes in the car mountain game, we visualize the results of the Q-table in a heat map. The following graph illustrates the q-table values after a number of repetitions and different gamma values. In particular, the parameters we tuned to get a better visualization was the number of episodes and the dimensions of the q-table. We noticed that as the number of episodes increases, the values of the q-table are raising. Also, the dimensions of the q-table are responsible for the number of pixels in our visualization,thus the more they are the higher resolution the graph has.



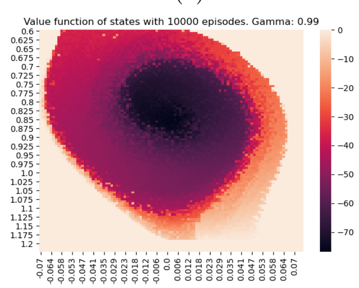
(a)



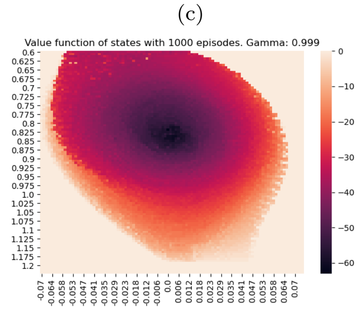
(b)



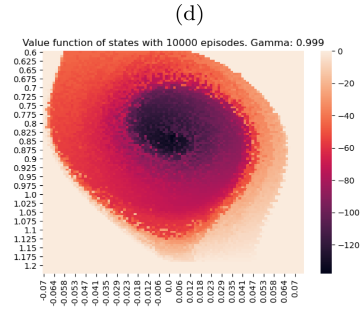
(c)



(d)



(e)



(f)