ACML-4

Balaganesh Mohan

November 2019

1 ReadMe

To run this program, open "Mountaincar.py" in a python compiler. Make sure to have all dependencies installed.

2 Implementation

Q learning for non-deterministic environment was tried here. For the value at index 0, we can see the high value is 0.6, the low is -1.2, and then for the value at index 1, the high is 0.07, and the low is -0.07. Also we can infer, every value has 8 decimal places. The size of the Q table is unfathomably huge and useless. What we can do here is convert these continuous values into discrete values. We can do that by grouping the ranges into smaller number say 20, this can be tweaked later to better results. This gives a 20 x 20 x 3 table. The 20 x 20 x 3 is every combination of the groups of all possible states. 3 is every possible action, i.e. left,right and none. The Q table gives every possible state in the environment. Each step has a reward of say -1 and the flag is 0 reward. This reward can be user defined based on the approach of a problem. The Q table is the reference and will be used to determine all the moves that can be done. During "Exploitation" we can choose to go with the action with highest Q value. Next, we must induce the learning formula from the lecture or also called "Exploration".

2.1 findings

Some parameters like gamma value(0.9) and actions(3) were constant throughout the experiment. The first set of parameters I experimented with greedy epsilon = 0.1 and each observation discretized into 120 units and alpha = 0.1. This parameters took a while to converge about 305,000 episodes. Figure 1 shows the output heatmap for this parameters set. These parameters although slow, they give nice outputs eventually. Some of them move occasionally, primarily because the agent has an exploration probability 0.1.

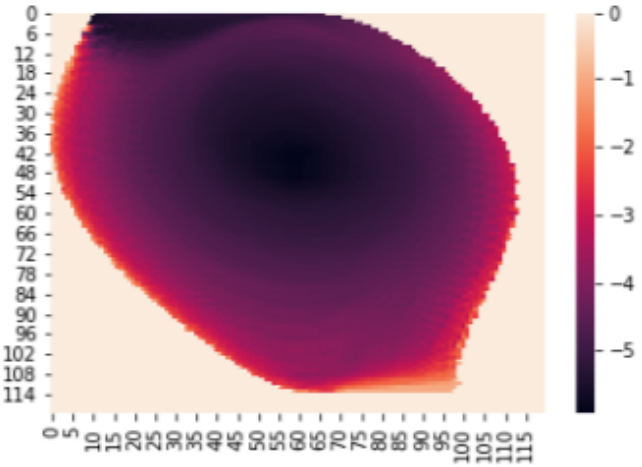

Figure 1: Q-learning greedy

The best parameters for this experiment were when, alpha = 0.1, epsilon = 0 and state_num = 20. The algorithm converged after 10000 episodes. Figure 2 shows the heatmap with the parameters specified.

The heatmap for both the experiments were constructed for each state(position,speed) by selecting the actions that caused the maximum value in a state.

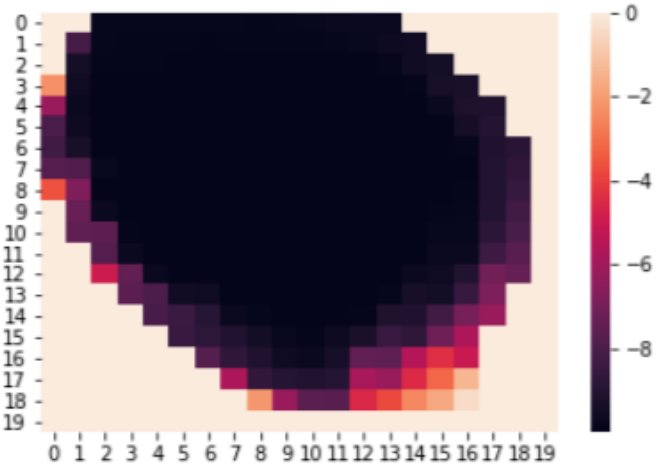

Figure 2: Q-learning