Ouassim Fari – Ouassim.fari@gmail.com

Project 4 Train a Smartcab to Drive

QUESTION: *Observe what you see with the agent's behavior as it takes random actions. Does the* **smartcab** *eventually make it to the destination? Are there any other interesting observations to note?*

After changing the action to “random”, the smartcab moves. But the motion is erratic and with no particular pattern. The smartcab did reach destination a couple of times but hit the “hard time limit” more often.

```
Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.
```

QUESTION: *What states have you identified that are appropriate for modeling the* **smartcab** *and environment? Why do you believe each of these states to be appropriate for this problem?*

I think the best stats are the inputs (traffic light, incoming car on three direction) and the next waypoints. I initially added the deadline but ended up removing it as it did not prove to be really beneficial. I think the input are very important to describe the state as it also defines the potential moves and thus the potential reward. Without knowing that the light is red, the smartcab wouldn’t be able to link the negative or positive reward to a state. I added the next waypoint as it gives also the information of where is located the next point, which also influences the action the smartcab should take.

OPTIONAL: *How many states in total exist for the* **smartcab** *in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?*

The inputs are binary and the amount of next potential waypoints is equal to the total number of intersections. Thus the total amount of states should be (2^4)*gridsize .

So : 768 states

This number seems to be reasonable as we have 100 trials and for each trial we have between 20 and 55 steps (deadlines). This leads to potentially visit between 15360 and 42240 states. Despites some states probably overlapping each other, we can assume that most of the 768 original states will be visited.

QUESTION: *What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?*

We can now notice that the agent is way more likely to reach destination after a few trial/iterations. The path followed make more sense. This behavior is occurring because we are now taking the action (forward left right) which tends to maximize the reward.

QUESTION: *Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?*

I wrote an optimization code to find the question. I iterate through many potential values of alpha and gamma and extract the ones which gives the best success rate. Multiple combination of alpha and gamma can give the best rate, so I then filter one more time on the one that gives the best rewards (which means less penalty and the fastest).

The best solution I found is for alpha : 0.9 and gamma: 0.7

With that configuration, I reached destination more than 99% of the time.

QUESTION: *Does your agent get close to finding an optimal policy, i.e. reach the destination in the minimum possible time, and not incur any penalties? How would you describe an optimal policy for this problem?*

Yes. An optimal policy can be defined as the policy which maximize the reward. As explained in the question above, my agent maximizes the success rate (reaching destination as well maximize the reward. The reward itself is the sum of the penalties and positive rewards which incurs during the travel.