Project Report: 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?*

The smartcab eventually makes it to the destination somehow after sufficient time, even though it follows random choices of direction. As far as I observe, the cab doesn't obey traffic rules, either yields to incoming traffic or follows traffic lights.

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 added **light**, **oncoming**, **left**, **right and next_waypoint** into **self.state**, then I noticed that the cab began to obey traffic rules after I update inputs of state. First of all, **light** allows agent to learn how to get through intersection without running the red light. Then, **oncoming, left** and **right** give the agent idea where other drivers are to obey basic traffic rules. Also, **next_waypoint** provides relative direction to destination when agent exploits the city.

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?*

After implementing Q-Learning, the behavior of smartcab improves significantly compared to random walking. It seems like that it gradually understands what's going on in the environment and consistently obeys to traffic rule, and quickly finds target destination. Clearly, this is a result of choosing actions leading to

target destination that maximizes Q values. I checked the last 10 trials and noticed that the smartcab was able to reach destination every time, which indicates Q-Learning algorithm works very well.

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?*

Let's focus on tuning **alpha** ranging from 0 to 0.6. Interestingly, success rate was 30% when alpha equals 0, then it increased to 100% when alpha was incremented by 0.1. And result remains 100% success rate while alpha adds from 0.1 to 0.6.
So, I intentionally change **n_trials** = 50000 and success rate is shown below.

alpha	success rate
0	20%
0.1	99.876%
0.2	99.913%
0.3	99.861%
0.4	99.859%
0.5	99.866%
0.6	99.850%

In this stage, we can say smartcab perform the best when alpha is set to 0.2. However, all success rates are above 99% except alpha equals 0, indicating the agent is able to find destination nearly every 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, I think the agent is close to optimal policy because of over-99% success rate shown above. It is highly skilled and knowledgeable to explore environment and finally reach the final destination. One thing makes me doubt is that agent still get penalty sometimes, which make me think that policy should be still improved even though it is very good. In real life situation, we want to train a smartcab to drive flawlessly without breaking traffic rules or hitting someone else. So, I think the optimal policy for this problem is to reach destination within minimal steps without any penalty.