Report on Project 4: Train a Smartcab to Drive

1. *In your report, mention what you see in the agent's behavior. Does it eventually make it to the target location?*

   The agent picks a random action to move left, right, forward or none. The agent mostly roams randomly without any clear knowledge on how to act upon the environment. It tries to violate traffic light and bashes into oncoming traffic.

   It almost every cases it fails to reach the destination. But very few times it was able to reach the destination.

2. *Justify why you picked these set of states, and how they model the agent and its environment.*

   I've selected all the environmental variables available(lights, oncoming, right, left) and also next waypoint as the set of states.

   "Lights" tell us how the traffic light is changing, which is a very important factor for the agent to correctly sense the environment.

   "Oncoming", "Right", "Left" tells us about the traffic in the road and would help us to avoid collision. Though including this states complicates the status of the states and we might not visit enough scenarios for the agent to pick the right signals.

   "Next_waypoint" tells us about where should the agent be headed to make the right direction. So it is another very important variable for the state to move to give the agent more sense about where the destination is. Without this, the agent would learn to drive safely, but would have no sense of where it should go.

   I did not use deadline so that agent does not any incentive to bump into other cars or violate traffic rules to reach the destination quickly.

3. *What changes do you notice in the agent's behavior?*

   I see that instead of roaming randomly, the agent is now making decisions that makes more sense. In the start, the agent wanders without any clear sense of right direction, but as it learns over time, it tends to make more right decisions. It usually fails the first two or three iterations almost every time. But then it has enough knowledge base to direct itself

to the destination and reaches it.

4. *Report what changes you made to your basic implementation of Q-Learning to achieve the final version of the agent. How well does it perform?*

   Discount Function: I removed discount factor because I figured the environment for the agent is local. A redline or traffic encountered in previous state does not help it figure out the state in the next position. So I removed discount function.

   Epsilon: I have decided on .25 and I reduce it by 0.01 every time a random action is taken.

   Learning Rate: Since the environment is deterministic, i.e we need to develop a certain kind of action of a given set of conditions on the state, I choose a higher learning rate: 1

Epsilon	Learning Rate	Successful Trials (out of 100)
0.35	.7	98
0.35	1	98, 99
0.25	1	100, 98
0.15	1	100, 97

5. *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?*

   It seems the agent does get close, but it's not an optimal policy. Sometimes it waits to take forward direction while it could instead take a left or right based on its position and try to reach the destination faster. Sometimes it to avoid collision, it takes a circular path and rotates and then again try to reach the destination which is not always the most efficient way to do it.

   I also have read in the forum that the current planner does not always give the optimum path. So, I think a better planner would help the agent more.

We also could have considered how changing the learning rate and epsilon had an impact on the cumulative reward point earned.

References:

Numerous posts from the MLND discussion forum and MLND slack group.
Previous reviewer.