Smart Cab – How 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?*

*Yes, the agent finally reached the destination. As it is the greedy approach thus it took a lot of time and does not care whether there is an oncoming vehicle or whether that vehicle is at its right or left and what is the state of red light. Which leads to take a lot of time.*

***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?*

*if(light=red):*

*next\_waypoint = [right,left(-),forward(-)]*

*if(next\_waypoint=forward)*

*action=[none(+) or right (-)]*

*else if (light=green)*

*next\_waypoint = [right(+), left, forward(+)]*

*if(oncoming=forward)*

*action=[none(+) or left (-)]*

*States as per US Laws*

1. *Light: red or green*
2. *Oncoming: forward, left, right or none*
3. *Left: Forward, left, right or none*
4. *Action: Take the next waypoint or none*
5. *Next waypoint: forward, left or right*

***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?*

***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?*

*The agent reaches the destination much faster. As while exploiting the earlier trials the agent was also exploring while collecting the positive and negative rewards. Thus, it helped to collect more rewards in the later trials and reach the destination quickly.*

***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?*

***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, obviously the agent reached to the optimal policy. In 100 iterations, it consistently reaches the destination, and minimally incurs penalties (especially in the beginning). After iterations, it seems to consistently take correct actions. One thing I would add is that the “learning” could probably be accelerated by creating more dummy agents (i.e. it would speed up convergence for states that were happening less often by making those events happen more frequently). By increasing the number of dummy agents, the car would probably interact more with them, and would accumulate penalties to avoid earlier than later. This would then translate to less “training time”, and would be a nice enhancement.