Project: Smartcab

Q1) 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 in a few random cases does reach the destination.

A few interesting observations:
The reward for reaching the destination within the deadline is high and positive.
The reward for reaching the destination outside the deadline is low and in some cases negative.

Q2) 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?

Following states have been identified as being appropriate for modeling the “smartcab” and its environment:

- “light”
- “oncoming”
- “left”
- “right”
- “location”: self.next_waypoint

“Light” is necessary to give us an indication if we can make a move in the current time step.
“oncoming” is necessary to tell us if we can make a left/right turn.
“Left & Right” lights tell us if there is traffic coming from the left or right side.
“location” tells us where we are in the traffic grid space.

The input “Deadline” was left out of the state consideration because including the deadline would explode/expand the state space combinations into a very large number. This would cause the agent to spend a really long time to learn an optimal policy.

All the above inputs signals are required to safely and lawfully inform the agent to choose its next action.

Q3) 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 the Q-Learning algorithm, the smartcab is able to learn a policy that makes it possible for the smartcab to reach the destination within the deadline. The updated learned Q-Values for each State-Action pair obtained after multiple iterations is the reason the smartcab is able to reach its destination.

Q4) 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?

The following parameters were tuned in the basic implementation of Q-Learning:

- The exploration rate: Epsilon
- The learning rate : Alpha
- The discount rate : Gamma

The driving agent performs best with the combination of Epsilon: 0.1, Alpha: 0.6 and Gamma: 0.2

Epsilon	Alpha	Gamma	Fail List
0.1	0.6	0.4	[1, 8, 40, 42, 66]
0.1	0.6	0.6	[0, 1, 2, 4, 7, 11, 24, 25, 26, 44, 45, 58, 59, 62, 92]
0.1	0.6	0.8	[0, 5, 7, 17, 45, 47, 48, 60, 71, 87]
0.1	0.6	0.2	[0]
0.1	0.4	0.2	[9,38]
0.1	0.8	0.2	[0, 8, 56]
0.3	0.4	0.2	[0, 9, 25, 29]
0.5	0.6	0.2	[0, 1, 2, 6, 7, 11, 13, 14, 18, 26, 35, 37, 42, 84, 89, 99]

The final driving agent on an average is able to reach the destination within the allotted time in all cases except in a few initial ones where its learning the optimal Q-value.

Q5) 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?

The agent does find an optimal policy after the finding the optimal q-values. An optimal policy for a smart cab is one that does not break any traffic rules and reaches the destination by taking the best&fastest route possible within the given deadline.