Tasks

Project Report

You will be required to submit a project report along with your modified agent code as part of your submission. As you complete the tasks below, include thorough, detailed answers to each question*provided in italics*.

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?*

ANSWER: *By taking random actions in each step, the basic driving agent will basically perform a random walk on the grid until it finally reaches the destination in each trial. Most of the time the agent did not reach the destination. It is just a luck to be on the goal. As for any other observations, the agent only reached the destination when it actually started at or very close to it. It was just a matter of chance.*

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?*

ANSWER: *The agent have to obey traffic rules and follow the direction given by route planner. Total 4 features have been chosen to represent a state:*

1. Next waypoint:
The agent has no info of destination and/or doesn’t know which way to go next. So, it needs to be taught the route from route planner how to reach the destination.

2. Traffic light:
The agent doesn’t know any traffic rules. All actions of agent such as going left, forward etc. need to be taken with consideration of the traffic light. For example, going forward action takes priority over all other agent’s action if traffic light is green, it means the agent can decide whether it may go forward or not only by the state of traffic light.

3. Boolean for turn left:
In the traffic rules for this project, the agent can turn left when traffic light is green and there is no oncoming traffic making a right turn or coming straight through the intersection. So, I defined the variable that shows whether the agent can turn left or not.

4. Boolean for turn right
The agent can turn right when traffic light is green. Also, when traffic light is red, a right turn is permitted if no oncoming traffic is approaching from agent's left through the intersection.

I believe these sates are enough and appropriate because through these features the agent can decide to move in which direction and all actions are represented.
(Deadline wouldn't have made a difference since the action of agent would be the same irrespective of the number of moves left.)

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?*

ANSWER: *After implementing the Q learning and taking optimal actions in each step using the resulting Q table, it is observed that:*
1. The agent takes proper actions at each traffic light and tries to obey the traffic rules to the extent it is rewarded differently based on legal and illegal actions.
2. The agent will learn to reach the destination to get the big 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?*

ANSWER: *QLearning algorithm was implemented to help agent to choose best action depends on its current state and behavior was significantly improved. I used equation below to update Q value for each agent state and action.*

$$Q(s, a) = (1 - \alpha) \bullet Q(s, a) + \alpha \, [r + \gamma Q(s', a')]$$

I made some improvements to the Q learning process by making the following adjustments:

a(Alpha): For a , I found that higher value results in high rate of reaching destination. High a value means the agent update its QLearning value a lot from Q-Learning value from next state and action. Especially, it is important to learn Q-Learning value fast at the beginning of trials, so the higher ‘a’ resulted into high accuracy.

Gamma (discount factor): For gamma , I can see that higher gamma value tend to get high average reward. Although the agent with lower gamma value reaches destination as similar rate as higher gamma value, it seems to drive inefficiently to get to destination.

Here is a table displaying the number of times agent reached destination per 100 times of trials for different values of alpha and gamma:

	a=0.5	a=0.65	a=0.8
Gamma=0.2	91/100	95/100	97/100
Gamma=0.5	95/100	92/100	94/100
Gamma=0.9	92/100	96/100	98/100

Here is a table of average rewards for the different values alpha and gamma:

	a=0.5	a=0.65	a=0.8
Gamma=0.2	1.7897	1.6425	1.7576
Gamma=0.5	1.7961	1.9438	1.9296
Gamma=0.9	1.8137	2.100	2.0176

Finally, I chose a = 0.5 and gamma = 0.9 to the best parameters for QLearning method.

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?*

ANSWER: *By making the abovementioned adjustments, the agent is able to learn a feasible policy i.e. reach the destination within the allotted time, with net reward remaining positive. However, it does not necessarily reach the destination in the minimum possible time without any penalties. I believe in order to achieve that, changing of the reward is required in this project mainly because if we are trying to optimize total reward (as the goal of Q learning), the agent wouldn’t necessarily want to reach the destination in the minimum amount of time if, as an alternative, it can just collect rewards by driving around on the streets and still reach the destination in the allotted time.*
