Implement a Basic Driving Agent

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: Smartcab eventually reaches the destination in some of the trials within hardline. When the cab is approaching an intersection it does not check if there is already a car at the point to avoid hitting it.

Inform the Driving Agent

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: I chose the following variables as state for the model

Light: smartcab needs to make decisions based on light, it needs to abide by the traffic rules ensuring safety.

Oncoming: Oncoming traffic would affect smartcab's decision to take turns

Left: If Light is red, traffic coming from left would be moving and we cannot take turns

Right: Traffic coming from right does not affect the smartcab's decision in any way. When the light is red, we cannot turn left and turning right is not affected by whether there is traffic on right side.

Waypoint: This variable keeps track of which way we need to head and is necessary or smartcab to make decision.

Deadline: Deadline can be used as a state variable because it can let the agent make risky decisions to reach destination faster maybe run a red light but including it would increase the state space by 100 times, so I did not include the variable.

Implement a Q-Learning Driving Agent

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?

In Q learning, we are making informed decision. Initially the smartcab makes random actions, once sufficient trial runs are passed, it starts to make specific actions for specific states based on the maximum q value. The agent needs to explore the state sufficient no of times to arrive at a proper action.

With Q learning agent, the smartcab is able to reach the destination within the deadline for 96 out

of 100 trials.

Our agent starts to obey the traffic rules as we see from below Qtable when the light is red, the value for moving forward decreasing when the next_waypoint is forward

Trial	Light	Oncoming	Left	Waypoint	Action	Qvalue
1	red	None	None	forward	forward	-0.2
20	red	None	None	forward	forward	-0.324434
32	red	None	None	forward	forward	-0.406461
100	red	None	None	forward	forward	-0.472794

Whereas for the action 'right' (permissible action) increases

Trial	Light	Oncoming	Left	Waypoint	Action	Qvalue
1	red	None	None	forward	right	-0.068
25	red	None	None	forward	right	0.691089
50	red	None	None	forward	right	1.153812
75	red	None	None	forward	right	0.826035
100	red	None	None	forward	right	0.711466

Q learning is an off policy learning. We choose the action resulting in better rewards in future captured by the Q value for state action pair, this is exploitation. We also randomize the actions to some extent (alpha) which makes sure that we are exploring all the actions which could result in better rewards, this is exploration. Thus a perfect combination of exploration and exploitation lets the smartcab to make better decisions.

Below table shows for a specific state, how Q value helps in choosing the right action (increasing Q value)

Trial	Light	Oncoming	Left	Next_waypoint	Action	Qvalue
1	green	None	None	left	left	0.4
2	green	None	None	left	left	0.7776
6	green	None	None	left	left	1.788907
9	green	None	None	left	left	2.27175
12	green	None	None	left	left	3.739136

Improve the Q-Learning Driving Agent

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:

I ran 1000 trials for each set of parameters

Performance evaluation is based on two values:
1. % success: no of times agent reached destination / 1000
2. Reward/Step: Total reward for 1000 trails / no of steps

Alpha	Gamma	Epsilon	% success	Reward/step
0.2	0.7	0.2	78.6	1.49
0.2	0.7	0.3	74.3	1.29
0.2	0.7	0.5	51	0.8
0.2	0.7	0.1	90	1.73
0.2	0.9	0.1	78	1.65
0.2	0.9	0.5	51	0.8
0.2	0.5	0.1	88	1.72
0.2	0.5	0.5	51	0.82
0.2	0.3	0.1	93	1.7
0.2	0.1	0.1	97.5	1.6
0.2	0.4	0.05	93	1.85
0.2	0.6	0.05	92	1.83
0.4	0.7	0.1	87.5	1.74
0.4	0.9	0.1	81	1.5
0.4	0.5	0.1	87	1.7
0.4	0.7	0.5	54	0.8
0.6	0.7	0.1	85	1.64
0.6	0.9	0.1	77	1.4
0.6	0.5	0.1	91	1.75
0.6	0.3	0.1	96	1.63
0.6	0.1	0.1	97	1.57
0.8	0.7	0.1	93	1.57
0.8	0.9	0.1	76	1.33
0.8	0.5	0.1	91	1.65
0.8	0.3	0.1	96	1.6
0.8	0.5	0.5	53	0.8

I tried to optimize the epsilon parameter, so for same alpha and gamma I varied epsilon and saw the success rate clearly deteriorating with increase in epsilon.

I tried different gamma values for same alpha and epsilon, the success rate increasing with decrease in gamma when gamma is less than 0.5 but average reward per step is decreasing because

having lower gamma means it relies on short term rewards and success rate is decreasing with gamma beyond 0.5.

For the same gamma and epsilon values, I tried different alpha. I find with increasing alpha, reward per step decreasing with success rate decreasing from 0.2 to 0.4 and increasing afterwards.

So I chose alpha = 0.2, gamma = 0.4, epsilon = 0.05 which gave best success rate of 93 % and reward/step of 1.85

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:

Trained agent is close to finding the optimal policy as it reaches destination within deadline. The optimal policy would be for the agent to reach the destination within deadline, follow traffic rules and take quickest route to destination.

Trained smartcab with parameters chosen reaches the destination 93 % of the trials within deadline.

Trained smartcab follows the traffic rules as mentioned in answer in implementing Q learning section, we clearly saw how the action that follows traffic rules had Q value increased as the trial runs pass.

In some of the states, we see the Q value for action deviating from next_waypoint being more suggesting that the smartcab chooses to accumulate reward rather than to reach destination quicker.

Light	Oncoming	Left	Next_waypoint	Action	Qvalue
green	forward	None	forward	forward	6.812005
green	forward	None	left	forward	0.17768
green	forward	None	right	forward	0.329387
green	left	left	forward	forward	0.730175
green	left	None	forward	forward	3.150144
green	left	None	left	right	0.818136
green	left	None	right	right	3.303626
green	None	forward	forward	forward	2.98064
green	None	forward	left	forward	0.31404
green	None	forward	right	right	4.557041
green	None	left	forward	forward	7.074043
green	None	left	left	left	3.798281
green	None	left	right	forward	0.862925
green	None	None	forward	forward	4.346719
green	None	None	left	left	4.177976

green	None	None	right	right	3.020459
green	None	right	forward	left	0.629611
green	None	right	left	forward	0.152319
green	None	right	right	right	2.541932
green	right	None	forward	forward	6.153738
green	right	None	left	right	0
green	right	None	right	forward	0.326471
red	forward	None	forward	None	0
red	forward	None	left	right	0.675984
red	forward	None	right	right	3.292329
red	left	left	forward	right	0
red	left	left	right	right	0
red	left	None	forward	right	0.423441
red	left	None	left	right	0.807826
red	left	None	right	right	3.452124
red	None	forward	forward	None	0
red	None	forward	left	None	0
red	None	forward	right	None	0
red	None	left	forward	None	0.001293
red	None	left	left	right	0.125427
red	None	left	right	right	2.832388
red	None	None	forward	right	0.783541
red	None	None	left	right	0.960267
red	None	None	right	right	3.933949
red	None	right	forward	right	0
red	None	right	left	right	0.28299
red	None	right	right	right	2.534816
red	right	forward	forward	None	0
red	right	None	forward	None	0
red	right	None	left	None	0
red	right	None	right	right	1.669023