P4: Smartcab System via Reinforcement Learning

1. Implement a Basic Driving Agent

*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?*

When the smartcab's behavior takes random actions, it is observed that it does eventually make it to the destination as long as:

- It is within the deadline as defined by the Environment.hard_time_limit variable, or
- If the enforce deadline toggle is set to True and its current deadline is greater than 0.

Some observations:

- Since actions are generated randomly, it is highly likely that the smartcab would perform invalid moves (denoted as reward == -1.0) or a move that does not correspond with the next_waypoint (denoted as reward == -0.5).
- When the smartcab reaches the destination, its next_waypoint value is None and the current simulation trial ends.
- Other cars are represented as DummyAgent objects and are strictly following the traffic rules defined in DummyAgent.update() method.

2. Exploring the Data

*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?*

We identified the states consist of a combination of the following state variables while the simulation trial is still active:

State variable	Values	Description
light	['red', 'green']	• Indicates the current state of the traffic light at the intersection. • It is factored into what next action (i.e. whether to move or not) to take.
oncoming	[None, 'forward', 'left', 'right']	• Indicates if there is oncoming traffic at the intersection. • If the next waypoint is to make a left at the intersection and the light is green, this state variable is taken into consideration on whether to make a left on the next action or not.
left	[None, 'forward', 'left', 'right']	• Indicates if there is oncoming traffic from the smartcab's left at the intersection. • If the next waypoint is to make a right and the light is red, this state variable is taken into consideration on whether to make a right on red on the next action or not.
next_waypoint	['forward', 'left', 'right']	• Indicates the direction of the next intersection waypoint.

*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?*

Using the above defined state variables would yield 2 x 4 x 4 x 3 = 96 states and seems to provide a reasonable number for Q-Learning since the first three state variables (light, oncoming, light) are from the environment and are taken into consideration in the U.S. Right-of-Way rules, i.e.:

- On a green light, a left turn is permitted if there is no oncoming traffic making a right turn or coming straight through the intersection.
- On a red light, a right turn is permitted if no oncoming traffic is approaching from your left through the intersection.

Also, the next_waypoint state variable that indicates the direction of the next intersection is considered while the current simulation trial is active; even though None is also a valid next_waypoint value, it is not taken into consideration during an active simulation trial since it indicates that the destination has been reached (i.e. the current simulation trial ends).

Note that the 'right' state variable (consists of possible values: None, 'forward', 'left', 'right') from the environment was not included since the other cars are represented as DummyAgent objects, where it is assumed that they are strictly following the traffic rules defined in DummyAgent.update() method. Otherwise, this state variable would have been included and the number of states would have been 96 x 4 = 384 states.

Also note that the 'deadline' state variable was not included since, unlike the other state variables where they consist of between two to four possible values, this value is dependent on the distance between the start and destination points that are randomly generated for each simulation trial. If this state variable was included, this could potentially cause the number of possible states to grow exponentially. Furthermore, the accumulated rewards already take the deadline into account, namely when the agent has reached its final destination before the deadline has expired.

3. Preparing the Data

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?

When using the best action only for each step instead of using a random action, the following observations are made:

- The smartcab did not move despite having the right-of-way.
- One possible explanation to the behavior mentioned above involves the action being determined based on the maximum Q value in the Q-table for a given state, where the (state, action) pair in the Q-table is updated and is referenced whenever this pair is encountered again: all values in the Q-table are initialized at 0 and because the reward for doing nothing is 0, updating the Q-table via the Q-learning equation will not change anything.

4. Training and Evaluating Models

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?

20 simulations per (alpha, gamma, epsilon) combination were run and, as illustrated in the following tables where each represent the average number of times that the smartcab have successfully reached its destination within the deadline for the last 10 trials:

- The average deteriorates as epsilon increases.
- The average also tends to deteriorate if the alpha or gamma values are greater than approximately 0.8.
- Overall, alpha and gamma values less than 0.5 and epsilon value approximately equal to 0.1 tend to perform the best overall average as they yielded at least 9 successful trips out of the last 10 trials.

Epsilon	0.1	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	9.6	9.55	9.55	9.4	9.7	9.75	9.75	9.75	8.9	9.45
	0.2	9.7	9.8	9.8	9.5	9.4	9.55	9.25	8.85	8.7	9.15
	0.3	9.7	9.6	9.5	9.65	9.45	8.6	8.55	8.6	7.8	8
	0.4	9.85	9.65	9.65	9.15	9.1	9.25	8.9	8.55	8.05	6.35
	0.5	9.7	9.6	9.65	9.5	9.55	8.9	9.15	8.45	8.2	6.75
	0.6	9.6	9.7	9.45	9.55	9.45	9.35	8.9	8.5	7.55	5.4
	0.7	9.55	9.55	9.6	9.7	9.45	8.8	9.05	8	6.4	4.7
	0.8	9.75	9.75	9.7	9.3	9.1	8.4	9.2	8.05	6.65	3.65
	0.9	9.6	9.6	9.55	9.55	9.15	8.85	8.9	7.8	6.5	4.05
	1.0	9.4	9.65	9.5	9.35	8.85	8.7	8.05	7.1	5.85	3.9

Epsilon	0.2	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	9.25	9.25	9.45	9.5	9.1	9.05	8.95	8.95	9	8.65
	0.2	9.25	9	9.5	8.9	8.8	8.75	8.5	8.75	8.25	7.4
	0.3	8.9	9.4	9.1	9.05	8.85	8.75	8	8.65	7.65	7.55
	0.4	9.6	9.4	8.95	9.15	9.1	8.7	8.25	8.35	7.9	5.95
	0.5	9.35	9.4	8.95	9.3	9.2	8.15	8.9	8.25	7.45	5.8
	0.6	9.2	9.1	8.65	8.7	8.95	9.1	8.6	8.2	6.65	4.05
	0.7	9.8	9.4	9	8.85	8.65	8.85	7.9	8.35	6.85	4.3
	0.8	9.35	9.4	9.35	8.9	8.7	8.8	8.65	7.95	6.4	3.8
	0.9	9.4	9.35	8.7	8.75	9.15	8.55	7.15	7.1	5.9	3.55
	1.0	9.4	9.4	9.2	8.7	8.85	8.3	7.55	6.6	5.05	3.2

Epsilon	0.3	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	8.65	9	8.85	8.95	8.45	8.3	8.45	7.85	8.2	8.5
	0.2	8.55	8.8	8.95	8.6	8.8	7.7	7.6	7.8	7.5	7.55
	0.3	8.8	8.7	8.6	8.45	8.35	7.45	7.85	7.9	7.6	6.35
	0.4	9.25	9.2	8.7	7.95	8.4	7.9	7.5	7.15	6.95	5.45
	0.5	9.05	8.55	8.75	7.8	8.7	8.3	7.15	7.65	6.55	4.7
	0.6	8.45	8.75	8.35	8.45	8.4	8	7.85	7.3	6.6	4.6
	0.7	8.8	9.35	8.85	8.5	8.05	8.05	8.1	6.8	5.5	4.4
	0.8	9.1	8.7	8.75	7.8	8.5	7.75	8	6.45	5.65	4.35
	0.9	8.85	8.9	9.05	8.2	8.25	8.1	6.95	6.95	4.85	2.95
	1.0	8.85	8.85	8.2	8.2	8	7.55	5.65	5.75	4.55	3.4

Epsilon	0.4	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	8.35	7.55	8.35	8.05	7.75	7.6	7.55	7.05	7.5	7.35
	0.2	8.2	7.7	7.45	7.75	7.7	7.1	7.15	6.6	6.45	6.35
	0.3	7.95	8.05	7.95	7.6	7.4	7.35	7.35	7	7.25	5.7
	0.4	8	8.5	8.05	8.35	7.35	7.35	7.05	6.55	5.85	4.85
	0.5	7.8	7.95	8	7.8	7.7	6.95	7.8	6.7	6.35	4.45
	0.6	7.75	7.75	8.2	7.45	7.3	7.2	6.4	6.6	5.55	4.6
	0.7	7.65	8.2	7.85	7.45	7.7	7.35	6.75	6.2	5.55	3.7
	0.8	8.3	7.65	7.9	7.7	7.65	7.15	6.35	6	4.75	3.7
	0.9	7.85	7.8	7.75	7.25	7.2	6.9	5.95	5.55	4.8	3.25
	1.0	7.65	7.4	7.15	7.6	7.55	6.6	5.9	4.8	3.6	2.5

Epsilon	0.5	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	6.9	6.9	6.55	7.05	7.35	6.75	6.9	6.2	6.1	5.65
	0.2	6.55	7.15	7.4	6.8	6.85	6.95	6.45	6.3	6	5.35
	0.3	6.5	7.75	6.8	6.95	7.45	6.55	6.5	5.9	5.8	5.25
	0.4	6.75	7.15	6.65	7.25	6.9	6.65	6.2	5.6	5.85	4.5
	0.5	8	7	7.05	6.15	7.1	6.25	6.35	5.8	5.55	4.4
	0.6	7.55	6.85	6.8	7.45	6.55	6.1	6.2	5.55	4.75	4.05
	0.7	6.65	7.55	6.65	6.9	6.65	6.05	5.7	5.85	4.75	3.75
	0.8	7	7.35	7	6.35	6.85	5.95	5.55	5.35	4.85	3.8
	0.9	6.8	6.75	6.75	7.1	6	6.4	5.15	4.7	3.85	3.15
	1.0	7.5	6.55	6.35	6.4	6.3	5.8	5.05	4.7	3.45	2.6

Epsilon	0.6	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	6	5.45	5.65	5.9	6.15	5.15	5.05	5.05	4.95	5.45
	0.2	6.05	5.9	6.05	6.05	5.35	5.75	5.15	4.5	4.95	5.05
	0.3	6.1	5.7	6.25	5.85	6.2	5.6	5.2	4.9	5	4.8
	0.4	5.7	6.2	5.75	5.8	5.5	5.85	5.4	5.35	4.15	4.1
	0.5	5.85	6.3	5.7	5.8	6.15	6	5.35	5.35	4.95	3.7
	0.6	6.1	6.1	5.85	5.9	5.95	5.5	5.15	4	4.5	4.4
	0.7	6.7	5.95	5.75	5.65	5.75	5.8	5.05	4.85	4.55	4.15
	0.8	5.7	6.75	5.4	5.2	5.45	5.25	4.7	4.95	3.55	3.1
	0.9	6.15	5.8	6.3	5.35	5.5	5.55	5.2	4.55	3.65	3.05
	1.0	5.8	6.1	5.7	5.5	5	5.1	4.25	3.95	2.85	2.1

Epsilon	0.7	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	4.85	4.75	5	5.15	4.1	4.85	4.4	5.25	3.55	3.7
	0.2	5.5	4.1	4.05	4.55	4.4	4.75	3.95	4.8	4.9	3.7
	0.3	4.75	5	5.15	5.15	4.35	4.15	4.65	4.35	3.7	4.15
	0.4	5.05	5.65	4.5	4.85	5	4.55	4.4	4.5	4.55	3.2
	0.5	5.05	5	4.75	4.4	4.75	4.65	4.5	4.45	4.55	3.8
	0.6	4.15	4.85	4.4	4.6	4.5	4.2	4.15	4.5	3.45	3.85
	0.7	4.25	4.9	4.7	4.6	4.75	4.5	4.15	4.25	3.7	2.95
	0.8	4.95	5.25	4.65	4.45	4.95	4.9	4.9	3.95	3.25	3.05
	0.9	4.9	4.9	4.95	4.8	4.8	4.25	3.55	3.5	3.9	2.3
	1.0	4.9	4.55	4	4.6	4.8	4.45	4.25	3.4	3.15	2.2

Epsilon	0.8	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	3.45	3.75	3.8	3.95	3.4	3.4	4.35	3.85	3.95	4.35
	0.2	3.9	4.5	4.15	4.3	4.05	3.65	3.75	4.15	3.5	3.6
	0.3	4.15	3.85	3.85	4.1	3.55	3.9	3.5	3.6	4	3.7
	0.4	3.95	4.3	3.65	3.35	4.15	3.85	3.8	3.55	3.15	4.25
	0.5	3.15	3.5	3.7	3.6	3.3	3.45	3.9	3.55	3.5	3.6
	0.6	4.25	3.05	3.75	3.7	3.4	4.6	3.65	3.45	3.5	3.75
	0.7	3.9	3.35	3.45	3.95	3.35	3.6	3.75	4.1	2.8	2.65
	0.8	3.1	3.2	3.9	3.1	3.6	3.4	3.25	3.45	2.75	3.4
	0.9	3.75	3.6	3.4	3.95	3.4	3.75	3.3	3.35	2.9	2.65
	1.0	3.9	3.4	4	3	2.75	2.85	2.75	2.85	2.35	2

Epsilon	0.9	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	2.85	2.75	3.15	3.35	2.5	2.7	2.75	2.35	2.7	3.05
	0.2	2.65	3	2.55	2.9	3	3.85	2.55	2.9	2.5	2.85
	0.3	2.4	2.75	2.95	2.6	3	2.7	2.25	2.5	2.6	2.55
	0.4	2.35	2.85	2.95	2.5	3.25	3	3.35	3.15	2.7	2.6
	0.5	3	3.3	2.9	2.85	2.9	2.55	2.55	2.4	2.6	2.3
	0.6	2.85	2.95	2.5	2.85	3	3.05	2.8	2.55	2.75	2.8
	0.7	3.25	2.85	2.85	3.3	3.35	2.65	2.6	2.75	2.5	2.55
	0.8	3.3	2.45	2.9	2.9	2.75	2.6	2.65	2.75	3	2.55
	0.9	2.6	2.8	2.5	2.75	2.5	2.85	2.4	2.8	2.8	2.35
	1.0	3.1	2.95	3.1	2.65	3.15	2.8	2.7	2.15	2.2	2.35

Epsilon	1.0	Gamma									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Alpha	0.1	2.1	2.3	2.1	2.05	2.15	1.95	1.55	1.8	1.65	2.5
	0.2	2.35	2.3	2.5	2.2	2.15	2.05	1.9	2.1	2.25	2.25
	0.3	2.8	1.5	2.2	2.35	1.95	2.25	1.85	2.1	1.75	1.95
	0.4	1.65	1.7	1.45	1.85	2.05	2.15	2.65	2.45	2.35	2.15
	0.5	1.95	2.2	2.75	1.65	2.45	2.15	2.05	1.6	2.45	2.3
	0.6	2.3	1.9	2	2.6	1.95	1.85	2.4	2.15	1.9	1.95
	0.7	2.25	2	2.05	2.3	2.7	2.6	1.95	2.1	2	2.15
	0.8	2.1	1.9	1.65	2.25	1.85	2.35	1.65	2.3	2.65	2
	0.9	2.15	2.05	1.75	2.1	2.15	2.2	2.2	1.9	1.95	2.3
	1.0	1.8	2.25	2.7	2.55	2.15	2.2	2.2	2.35	2.1	2.2

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?

Ideally, characteristics of the optimal policy for this problem would include the ability for the agent to reach its destination within the deadline without incurring penalties (ex: obey the traffic laws, follow the next waypoints). Upon inspection of the obtained results where alpha and gamma values are less than 0.5 and epsilon approximately equal to 0.1, it is observed that for the majority of the last 10 simulation trials the agent was close to finding an optimal policy as the smartcab tried to always go the shortest path to the destination path while incurring few penalties.

Given the smartcab reaching its destination within the deadline in the last 10 trials, the following tables represent:

- An example of the few instances where penalties occurred (possibly due to the random action taken with epsilon probability during the simulation).
- The Q-table values of the corresponding learning agent state before its trial run.
- Please note that results may vary due to the random nature of each simulation trial run.

Index	Trial	Input	action	next_waypoint	reward
1	90	{'light': 'green', 'oncoming': None, 'right': None, 'left': None}	left	right	-0.5
2	91	{'light': 'red', 'oncoming': None, 'right': None, 'left': None}	right	forward	-0.5
3	91	{'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}	left	forward	-0.5
4	93	{'light': 'red', 'oncoming': None, 'right': None, 'left': None}	left	forward	-1.0
5	96	{'light': 'green', 'oncoming': None, 'right': None, 'left': None}	left	forward	-0.5
6	96	{'light': 'green', 'oncoming': None, 'right': None, 'left': None}	left	forward	-0.5
7	97	{'light': 'green', 'oncoming': None, 'right': None, 'left': None}	right	forward	-0.5
8	97	{'light': 'red', 'oncoming': None, 'right': None, 'left': None}	right	forward	-0.5
9	98	{'light': 'red', 'oncoming': None, 'right': None, 'left': None}	forward	forward	-1.0
10	98	{'light': 'green', 'oncoming': None, 'right': None, 'left': None}	left	forward	-0.5
11	98	{'light': 'green', 'oncoming': None, 'right': None, 'left': None}	right	forward	-0.5
12	98	{'light': 'green', 'oncoming': None, 'right': None, 'left': None}	forward	left	-0.5

Index	State	Actions			
		None	forward	left	right
1	{light:'green', oncoming:None, left:None, next_waypoint:'right'}	0.18672	0.21712	0.12913	2.27574
2	{light:'red', oncoming:None, left:None, next_waypoint:'forward'}	0.26494	0.82340	0.80629	-0.34261
3	{light:'green', oncoming:None, left:None, next_waypoint:'forward'}	0.39079	2.15809	0.15863	-0.16603
4	{light:'red', oncoming:None, left:None, next_waypoint:'forward'}	0.20577	0.82340	0.80629	-0.28861
5	{light:'green', oncoming:None, left:None, next_waypoint:'forward'}	0.40280	2.27343	0.13576	-0.16603
6	{light:'green', oncoming:None, left:None, next_waypoint:'forward'}	0.40280	2.27343	0.13576	-0.16603
7	{light:'green', oncoming:None, left:None, next_waypoint:'forward'}	0.40280	2.23223	0.12096	-0.16603
8	{light:'red', oncoming:None, left:None, next_waypoint:'forward'}	0.26984	0.82340	0.83680	-0.28861
9	{light:'red', oncoming:None, left:None, next_waypoint:'forward'}	0.20707	0.82340	0.83680	-0.24445
10	{light:'green', oncoming:None, left:None, next_waypoint:'forward'}	0.40280	2.15944	0.12096	-0.22906
11	{light:'green', oncoming:None, left:None, next_waypoint:'forward'}	0.40280	2.15944	0.12096	-0.22906
12	{light:'green', oncoming:None, left:None, next_waypoint:'left'}	0.24760	-0.14741	2.25900	-0.03586