Machine Learning Engineering Nanodegree
P4 - Train a SmartCab to Drive
Jose Augusto Montiel

Basic Driving Agent

Does it eventually make it to the target location?

Yes, although it can be considered as a mere act of coincidence, since our agent just takes random actions until it finally reaches its final destination, no learning is being done nor its performance improves on each iteration.

Justify why you picked these set of states, and how they model the agent and its environment.

This was the selected State model:

LearningAgentState(next_waypoint, light, green_oncoming_is_forward, red_left_is_forward)

next_waypoint: it allows the agent to keep awareness of its position relative to its final destination, under a real environment and product, this could be replaced by the geographical coordinates of the car obtained through a GPS. Possible values: *Left, Right, Forward.*

light: this is necessary to determine the proper action under any traffic light circumstance. It will prevent the car from advancing on a red light to prevent accidents. Possible values: *Red, Green.*

green_oncoming_is_forward: this variable helps the agent to learn the U.S. Right of Way's rules. It will help to keep track of whenever the light is green and there was or not an oncoming car, preventing the agent from turning left and causing a collision. Possible values: *True, False.*

red_left_is_forward: this variable helps the agent to learn the U.S. Right of Way's rules. It will help to keep track of whenever the light is red and there is or not a car coming from the left, preventing the agent from turning right and possibly causing a collision. Possible values: *True, False.*

Other available inputs were discarded or combined into composite elements due to performance and because they did not contribute much into the recreation of the environment and decision making process.

left, right, oncoming: this variable accounted for other agents at our agent's intersection at that given point in time. Possible values for these three variables were *Left, Right, Forward or None.* With 4 possible values each, implementing these inputs as-is would mean increasing the state-space by a factor of 64. However, by analyzing the U.S. Right of Way, we can conclude that we do not care about the *right* value at all, and so it is discarded. Now, for *left* we do take it into consideration when light is red and our agents wants to turn right. Same principle applies to *oncoming*, we only care for this value when the light is green and our agent wants to make a left turn. In order to extract both premises we composed two variables that store these conditions as either True or False, this way our agent can learn through its reward mechanism whether is right or not to take certain action under those given circumstances.

deadline: this variable lets us know how many units of time (steps) are left for the agent to arrive to its destination. After consideration, this variable seemed irrelevant to our state model. Our final state has the variable *next_waypoint* which points our agent towards the "*right direction"*, at the same time, our reward mechanism implements a *discount factor,* what this means is that our agent will learn that by following the directions toward its destination at early as possible, it will lead to larger rewards, hence, making the deadline variable pointless to our purpose.

Enhanced Driving Agent

What changes do you notice in the agent's behavior?

The behavior of the agent became less "erratic" as the trials advanced. On the second half of the training process, the agent started to move frequently towards its destination instead of moving further away from the waypoint, incurring into penalties like moving forward on a red light or even collisioning with another vehicle, it also arrived more frequently to its destination. All this thanks to the learning process (Q - Learning) to which the agent has gone through, where the agent has assessed multiple state-action pairs and calculated the overall utility of that path of action, this generates an optimal policy that our driving agent can follow at any state it may encounter to then take an appropriate action.

Report what changes you made to your basic implementation of Q-Learning to achieve the final version of the agent. How well does it perform?

In order to fine-tune the Driving Agent first we tried adding composed states to the Environment States, for example, we kept recording to each state whether the 'action' was taken toward the

next waypoint, however this proved ineffective for the learning process, the number of successful arrivals diminished dramatically.

Next, we fine-tuned the parameters of Q-Learning, Discount Factor (□) and Learning Rate (α) in the following manner and this are its results:

Discount Factor	Learning Rate	Destination Achieved / Trials
0.25	0.25	86/100
0.25	0.4	22/25
0.25	0.6	19/25
0.4	0.25	22/25
0.6	0.25	3/25

This results are discussed in more depth in the following section.

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?

We trained a Learning Driving Agent through the Q-Learning Algorithm, the Q - Table generated after training the Agent for 100 trials was then used as final Policy for a Trained Driving Agent, to whom we put under a 100 trial test in order to prove its effectiveness.

The results are as follow:

Discount Factor	Learning Rate	Destination Achieved / Trials	Actions Taken (Taken/Available)	Total Penalties Incurred
0.25	0.25	100/100	1182/2870	0
0.25	0.4	100/100	1252/3075	0
0.25	0.6	100/100	1468/2860	0
0.4	0.25	100/100	2160/2910	0
0.6	0.25	100/100	1380/3060	0

As result of this process we determined as the Optimal Policy the one obtained by using a Discount Factor (□) of 0.25 and a Learning Rate (α) of 0.4 . With this policy, our Driving Agent successfully arrived to its final destination within the given timeframe in 100 out of 100 opportunities. Our agent only had to take 2065 actions out of the 2895 actions allowed, a 59% breathing room. Besides, it did not incur into any kind of penalties (negative rewards).