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 by not enforcing a deadline, we are making the process infinitely long  and the Basic Agent under this circumstances can learn an infinite amount of states since these are time-based, defined by the deadline value.

## 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, oncoming, right, left, deadline)

**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.

**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.

**oncoming, right, left**: this values allows the Agent to be aware of other cars on the road, letting the Agent determine what actions to take to avoid collisions with other vehicles.

**deadline**: A deadline allows us to confront the Exploration vs Exploitation dilemma, by making the process finite we teach the Agent not to wander off infinitely collecting rewards, by not setting a deadline we are allowing there to be an infinite amount of valid states for the Agent to explore, making the learning process incredibly long and expensive.

# 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 ($\alpha$) in the following manner and this are its results:

| Discount Factor | Learning Rate | Destination Achieved / Trials |
|---|---|---|
| 0.25 | 0.25 | 18/25 |
| 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 25 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 | 18/25 | 571/735 | 0 |
| 0.25 | 0.4 | 22/25 | 565/700 | 0 |
| 0.25 | 0.6 | 19/25 | 512/645 | 0 |
| 0.4 | 0.25 | 22/25 | 655/735 | 0 |
| 0.6 | 0.25 | 3/25 | 645/720 | 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 ($\alpha$) of 0.4 . With this policy, our Driving Agent successfully arrived to its final destination within the given timeframe in 22 out of 25 opportunities, which represents an effectiveness of 88%. Our agent only had to take 565 actions out of the 700 actions allowed, a 20% breathing room. Besides, it did not incur into any kind of penalties (negative rewards).