Project Report

# Train A SmartCab to Drive

Asher Syed

*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: Yes, the smartcab is able to reach destination using random actions for approximately 25% of the trials. The smartcab is not able to move if the chosen random action is conflicted with the traffic lights.*

*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 have used these three variables for the state of smartcab*

1. Next point
2. State of traffic lights and state of other cars at the intersection
3. Remaining time

Each of the above variables are important in order to properly define smartcab state at a particular intersection. If we reduce the dimensionality of smartcab state, the Q values matrix will have less elements and may not model the agent behavior for all cases.

state = "{}".format(inputs)

# agent reaches destination only ~ 1% of the trials

state = "{}-{}".format(inputs, deadline)

# agent reaches destination only ~ 6% of the trials

state = "{}-{}-{}".format(inputs, deadline, self.next_waypoint)

# agent reaches destination only ~ 85% of the trials

*OPTIONAL: 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?*

*Answer: The number of all states is approximately given by*

*S = I * TL * M * T(rem)*

*Where*

*I = No. of Intersection*

*TL = State of Traffic Light at Each Intersection*

*M = Possible Moves from an Intersection*

*T(rem) = Remaining Time to Reach Destination*

Yes, this number is reasonable, as Q-learning has to map values to the possible states from a current state.

*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: The smartcab is selecting weighted actions based on q-values. This behavior is occurring due to weights given to actions based on their reward value at a particular intersection.*

*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 have tried with different sets of values of Alpha, Gamma and Epsilon*

| Alpha | Gamma | Epsilon |
|---|---|---|
| 0.2 | 0.6 | 0.0 |
| 0.4 | 0.6 | 0.0 |
| 0.6 | 0.6 | 0.0 |
| 0.8 | 0.6 | 0.0 |
| | | |

*I have used 0.0 for Epsilon as this param controls the randomness of selecting actions. Both Alpha and Gamma were chosen 0.8 and 0.6 respectively and the agent reaches destination ~ 85% percent of the time during a trial run of 100 episodes.*

*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: Yes, the agent does find an optimal policy in reaching the destination in the minimum possible time, however it does incur penalties in the course of action. As evident from the statistics over 100 run of 100 trials each, the smartcab reaches destination in approx. 85% of cases.*

*An optimal policy for this problem would be to use the shortest route to the destination.*