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. The smartcab uses a random path, and it doesn't move sometimes due to 'None' action being selected.

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?

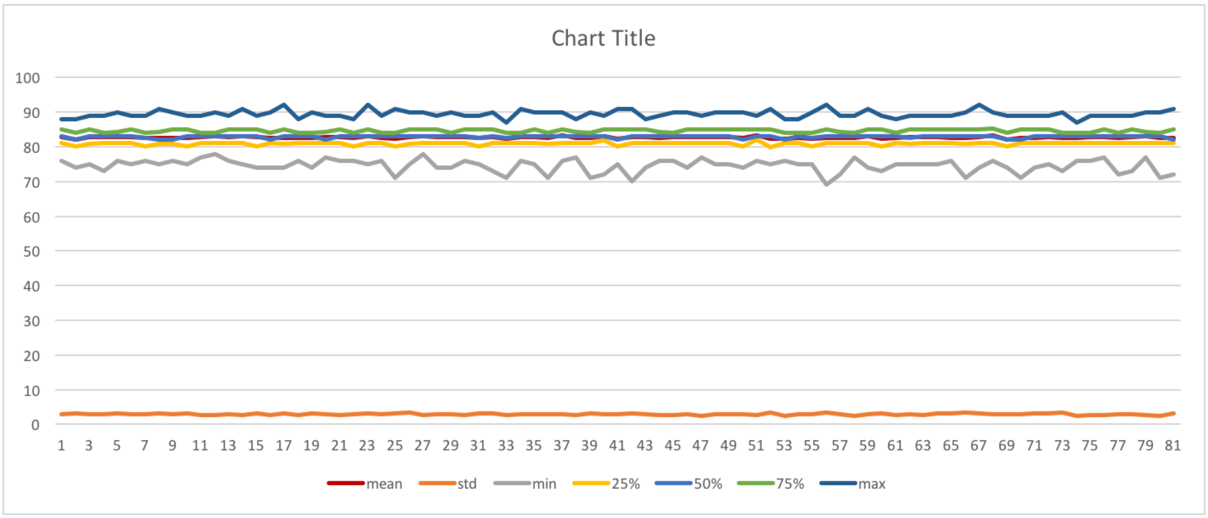
Answer: The smartcab is selecting actions based on largest q-value. This behavior is occurring as the q-values will have higher values for actions leading to final goal.

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

I have used 0.0 for Epsilon as this parameter controls the randomness of selecting actions. Both Alpha and Gamma were chosen 0.8 and 0.6 respectively and the agent reaches destination ~ 82% percent of the time during a trial run of 100 episodes.

Alpha and Gamma values do not affect the agent’s learning rate as shown by the following chart



The above chart is generated by running 100 trials of 100 episodes each with varying alpha and gamma values. The alpha and gamma were varied from 0.1 to 0.9. The runner.sh bash script is used to generated the resultset. And resulting result.csv file was imported into excel. All files are added to the repository

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: No, the agent does not find an optimal policy in reaching the destination in the minimum possible time, as it incurs 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 using minimum stops due to traffic lights.*

An example from real-life suggests that traffic lights follow same pattern on a route so if a traffic light is red at an intersection, it would also be red at the next intersection. Also traffic lights could have added time before state change so as to better train the smartcab agent but that would require some environment changes.

The smartcab agent can have hardcoded logic to always follow some predefined rules in case of taking random actions. These rules include

1. *Taking ‘None’ actions at a red traffic light*
2. *Taking alternate shortest route to destination on red traffic lights*
3. *Avoiding red traffic lights.*