

Basic Driving Agent

The basic driving agent picks a random action from all possible actions available. It does not seem to reach the goal in a consistent manner. The file BasicAgent.txt has sample result of running the basic agent. The table below describes summary of running the basic smart cab 3 times

Trials	Last Deadline	Reached Goal?
1	-225	No
2	-206	Yes
3	-149	yes

Choice of States

I chose my state as a tuple with information about light, oncoming, left and next_waypoint. I choose these states because state needs to contain all the information necessary to predict the effects of an action and to determine if it is a goal state. The smart car needs to know whether the light is green or red and if oncoming and left traffic is coming to decide on a which way it can go. The traffic rules are not particular about traffic from right and the smart cab drives in no deadline mode, so we can ignore right and deadline from state.

Q-Learning

Using Q-learning the smart cab seems to make better decision than basic driving agent. It is able to reach the goal. The table below describes the summary of running the smart cab using q-learning with alpha 0.8, Gamma 0.2 and epsilon 0.1

Trials	Last Deadline	Reached Goal?
1	-154	Yes
2	1	Yes
3	3	Yes

After setting alpha 0.9, Gamma 0.3 and epsilon 0.1 the result summary is as follows

Trials	Last Deadline	Reached Goal?
1	-40	Yes
2	-7	Yes
3	25	Yes

It seems increasing alpha and gamma values the smart cab is able to learn faster.

Improving Q-Learning

Smart cab using q-learning is able to learn to get to goal faster over time. But it is still not using the shortest path to reach goal after 2 or 3 trials. Using continuous state space could help improve q-learning. Another option is to use neural network instead of q-learning to program the smart cab to learn to drive.