In your report, mention what you see in the agent's behavior. Does it eventually make it to the target location?

When the action is set to pick from random the car moves around at random and would sometimes bump into the target location, but as soon as I changed the action to be the next waypoint, the agent would eventually make it to the target location.

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

I have included the stoplight of the intersection and the headings of all four sides of the intersection as the state of the agent. This takes in all the available sensor input information to build the model representing traffic. This way the state model is most robust against any traffic conditions and all types of traffic rules. The stoplight is needed because it's the basic foundation that dicates traffic flow. The next waypoint of the agent is included so the agent would know if it's getting closer to the target. When I excluded the next waypoint from the state it went around essentially at random and never got to the target. The oncoming, left side, and right side traffic are all included to let the agent properly learn the rules of the road. This also makes it more robust to the varying conditions of traffic. This state model considers all available inputs, and if any are omitted then the agent would be oblivious to the changes in that input and not act upon them.

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

Because I have initialized the Q table arbitrarily, the agent would sometimes miss the deadline early in the trials, but as the Q table builds up the agent would get to the target location without going over the deadline. Eventually the agent would decide between taking immediate negative reward but with a positive future reward, and sitting still at a stoplight. All of these contribute to getting the maximum total reward.

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?

I have added alpha the learning rate, gamma the discount rate, and epsilon the chance that the agent will explore to see if other actions would give more rewards. I then performed a manual grid search of low, medium, and high values for initial Q values, discount factor, and the learning rate. Epsilon was kept constant for all these runs.

I have decided that for this project getting to the target location within the deadline is the top priority, so I tried 27 combinations of the values to see which trip did the agent miss the deadlines. The following table shows the result of 100 trips for each initial Q values, discount factor, and learning rate, respectively. For example, LLL means the Q table values are initialized with a low value like 0.1, as well as discount factor and learning rate, whereas medium sets them at 0.5 and high at 0.9. The second column shows the trip number that the agent missed the deadline, with "all" being the agent rarely, if ever, made it within the deadline.

Values	Missed Deadlines
LLL	all

LLM	1, 4, 14, 15, 31, 37, 44, 50, 84, 100
LLH	2, 4, 9, 12, 15, 36, 58, 74, 83, 92, 98
LML	all
LMM	1, 3, 4, 8, 9, 13, 16, 18, 20, 25, 26, 27, 29, 30, 36, 39, 41, 42, 43, 44, 45, 50, 52, 53, 57, 62, 71, 73, 75, 76, 79, 83, 84, 86, 88, 92, 93, 95, 97, 98, 100
LMH	1, 2, 3, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 42, 45, 46, 64, 65, 81
LHL	4, 15, 18, 19, 27, 46, 57, 89
LHM	all
LHH	all
MLL	1, 2, 3, 5, 7, 8, 10
MLM	3,5
MLH	1, 2, 18
MML	2, 3, 4, 5, 8, 9, 11, 74
MMM	1, 2, 3, 4, 18, 46
MMH	none
MHL	all
MHM	all
MHH	all
HLL	all
HLM	1, 4, 5, 6, 7, 8, 10, 11, 14, 15, 16, 17, 24, 32
HLH	1, 2, 3, 4
HML	all
HMM	all
HMH	1, 2, 3, 5, 6, 7, 8, 9, 12, 13, 17, 20, 94
HHL	all
HHM	all
HHH	all

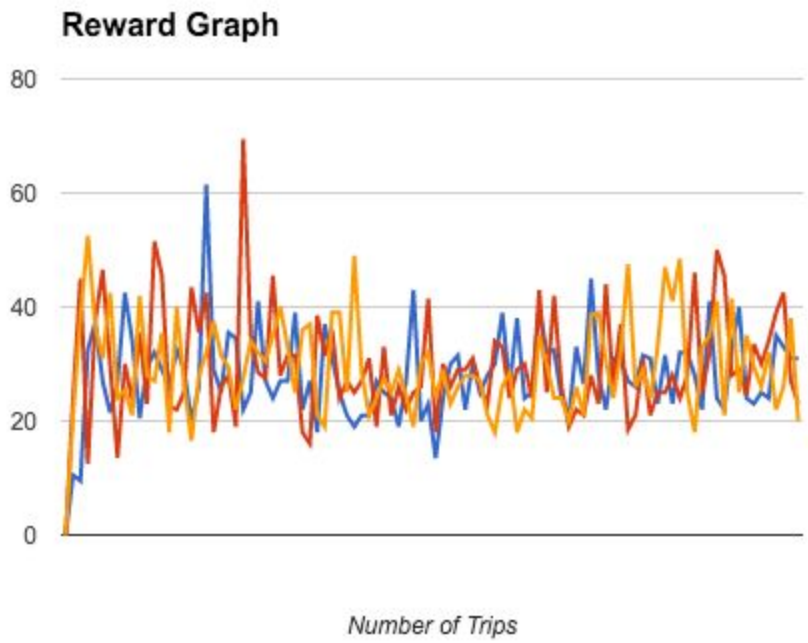
I then took the top performing combinations and did a second run with small variations of parameter values, such as high value for learning rate being 0.95 or 0.97, or medium for initial Q value being 3.14 or 10.1. The following table shows the results for the second group.

MMH	2, 4, 5, 40, 47, 72
	1, 3, 24, 37, 95
	1, 2, 5, 8
MLM	1

	1, 9, 13, 15, 30, 84
	1, 2, 4
	1, 2, 9, 14, 39
	1, 2, 4, 52, 71
MLH	1, 8, 97
	1, 2
HLH	1, 2, 3, 4, 5, 17, 84
MMM	1, 6, 13, 15, 41, 74, 79

As we can see the pattern for a better performing agent is to have a medium value to initialize the Q table, a low to medium value for the discount factor, and a medium to high value for the learning rate. The agent is still not perfect as it sometimes still misses the deadline, but it's better than what we started with.

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?



After I settled on my values for alpha, gamma, epsilon, as well as the value to initialize the Q table with, I ran the agent 3 time for 100 trips each. I tracked the total reward for each trip and plotted them on the graph above. Each line represent the reward values for 100 trips. As we can see the agent's behavior gets more consistent as it learns the environment, but nevertheless would still miss some deadlines. It would usually follow the waypoint, but would sit still sometimes. So at the end the agent was not operating at the most optimal policy, but it got close to finding the optimal policy.