P4-Report

Mention what you see in the agent's behavior. Does it eventually make it to the target location?

Since the grid is finite and enforce_deadline is false, the agent eventually reaches it's target location, but it may reach it before the deadline or after the deadline(may be infinitely after the deadline), that's because the agent here takes random steps and these random steps may get it close or far from the target, it **d**oesn't take the optimal ones and it doesn't learn from the wrong ones(exploration only, rewards are not being used).

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

Initially,each state is a tuple of 3 things, inputs sensed from the current environment,the next waypoint of our agent and the deadline (I'll change this format during enhancing the algorithm),
At each intersection our agent state depends on the traffic light whether it's true or false as it should wait till the light is true and continue.
Our agent depends also on the other cars coming from any direction and whether our agent should turn left or right or move forward or wait at a given position
It also depends on the remaining deadline as our agent's behavior and actions may vary depending on whether there is much more or much less deadline for the same position.
The state is a tuple to be hashable, in other words it will be used next as a key to the Q table(with actions).

Implement Q-learning,What changes do you notice in the agent's behavior?

After implementing initial Q-learning algorithm, by initializing the Q table values to zeros, and choosing then the best actions among the possible actions each time, the agent gets stuck most of the times into local maximum, there are also too many states as i embedded the deadline as part of my state, so the agent sees too many various combinations of states which may be not a good thing.(very high complexity)

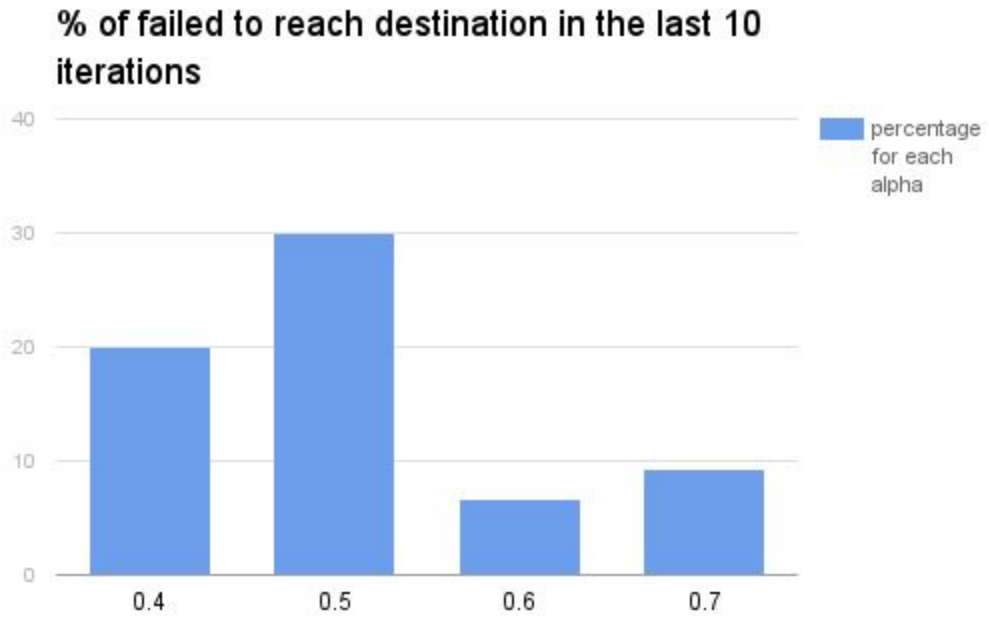
Report what changes you made to your basic implementation of Q-Learning to achieve the final version of the agent. How well does it perform?

I've reduced the complexity of the state and removed deadlines from it to reduce the total available number of states.
I've also added a new parameter epsilon, to make the trade off between exploration and exploitation, in other words the agent starts taking random action and epsilon decreases gradually till it reaches zero, this allows the agent to explore the environment first, then the

exploitation part starts and the agent instead of choosing randomly it chooses the best action available,
This method helps in decreasing the previous problem of being stuck in local maximum but it doesn't remove it at all as the agent continues to get stuck sometimes.

Epsilon decreases each iteration by 0.02, that's because it produced reasonable results in balancing between exploring and exploiting, i've tried decreasing by 0.1 and this leads to too less random steps and decreasing by 0.001 which leads to too more random steps

For tuning the learning rate, I've tried started with a very low alpha from the last submission (0.2), then after reconsidering,when decreasing alpha it means q values are never updated, then by increasing alpha and taking into consideration rewards and next state values this allows for more learning,
After trying different values of alphas with discount = 0.8 at first , and taking the average of more than one try for every alpha, i settled for alpha = 0.8 , then started balancing the discount factor a little, by decreasing it, at discount = 0.7 and trying different values of alphas starting from 0.2 up to 0.7 as shown in the figure below, turned out that alpha = 0.6 is good for this case and yields very good results in the last 10 iterations (from 0 to 1 times the agent couldn't reach the target), so i settled for alpha = 0.6 and gamma = 0.7.
(After alpha= 0.6 errors started to increase again so i stopped and took 0.6)



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 the enhancements above, the agent is able to reach the destination before reaching the deadline with total positive reward, and with an error percentage on average of 6.6% (on the final 10 iterations) that it would not reach the final destination before the deadline.
There is a minor problem while testing the agent at different values of alphas and gammas, sometimes even if the results of the agent were really good at the final 10 trials, after re-training it, it may fall into a local maxima and continue on choosing None as an action and not reaching the destination for several times, this happened sometimes while optimizing the parameters(even with parameters which gave very good results), this didn't show up with the current settings of alpha and gamma, but it may show as the agent probably may fall into a local maxima at some point.
Agent sometimes go in circles before reaching the destination that's because it sometimes continues to choose the same action for several times.
Other than that the agent behaves well and takes the shortest path to reach it's destination and reaches before deadline.