Project: Train a Smartcab to Drive

Implementing a basic driving agent

A basic driving agent was implemented which randomly chooses the direction it has to go. The basic agent was first run by not enforcing the deadline. This way, the car does not care about deadline, destination or penalty. Given enough moves, the car will reach the destination. The basic agent was then run by enforcing the deadline. This was done to find out the number of times the car reaches the destination within the stipulated time. The basic agent was run 100 times and the car reached the destination 17 times. This agent also does not care about the reward and penalties of each move. Our aim is to implement a policy following which the car consistently reaches the destination with less number of penalties.

Identify and update states

There are several inputs that can be considered to be included in the set of states required for modeling the driving agent. There are:

1. Next waypoint: next intersection where a car needs to go to reach destination
2. Deadline: The stipulated number of steps before which the car must reach destination for the trial to be marked successful.
3. Traffic light: red or green
4. Traffic: information about the direction of traffic coming from oncoming, left and right.

Next waypoint should be included in the set of states as it is responsible for obtaining the optimum policy to reach the destination. If next waypoint is not included, the car will not be able to consistently reach the destination within deadline. Deadline may be included in the set of states, but I have decided not to include it as including deadline may increase penalties. Information about traffic light should be included in the set of states. Not following traffic lights is one of the main reasons for getting penalties and causing agent. So, the driving agent must learn to wait at the red lights.

Information about oncoming and left traffic is required in avoiding penalties while turning left or going straight. So, they are included in the set of states. Information about traffic on the right is not important in avoiding penalties, hence it is not included. The relevant states included are:

1. Next waypoint
2. Traffic light
3. Traffic: left and oncoming

Implement Q-Learning

Q-Learning algorithm is implemented using the set of states identified in the previous section. In the first few trials, when the agent is still learning, there are few penalties. The car reaches the destination in few of them. After the q-table is updated of the negative reward received by not following traffic rules, the agent starts to learn to follow traffic rules, like to wait at the traffic light. When high reward is received in reaching the destination before deadline, the agent gets a higher incentive to reach the destination. After a few trials, the agent is able to reach the destination on time with fewer penalties.

At the learning rate and discount rate of 0.5 and epsilon value of 0.1, the car reaches the destination on an average 85 times out of 100 with a penalty rate of 0.024 per move.

Enhance the driving agent

The Q-Learning driving agent implemented in the previous section reaches the destination in a reasonable number of times with low penalty rate. This agent can be enhanced by optimizing the parameters: learning rate (alpha), discount factor and epsilon.

The alpha value is held constant at 0.9 and epsilon at 0.1, the number of successes and penalty rate (number of penalties per move) is calculated for different values of discount factor. These values are given in Figure 1. Discount factor encourages learning agent to seek out reward sooner rather than later. As the discount factor is increasing, the number of successes is decreasing as the agent is not seeking the higher reward of reaching the destination. The same cannot be said about penalty rate as it is also governed by the epsilon value. As the epsilon value is held constant in the run of 100 trials, the agent is as likely of taking a random move in the beginning as it is at the end. So, as the agent learns to avoid penalty, some penalties still occur due to epsilon value. As the epsilon value is increased, the rate of penalties increases.

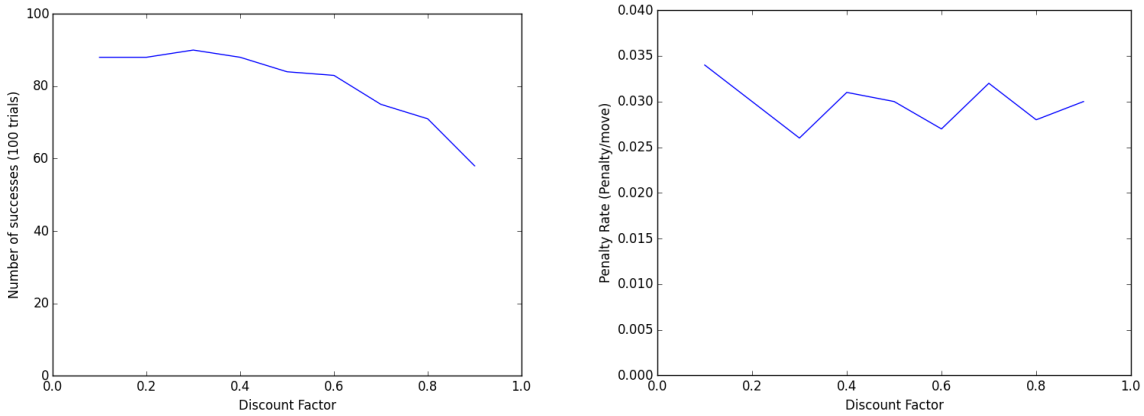

Figure 1: Change in number of success and Penalty rate with discount factor (100 trials)

A discount factor of 0.3 is chosen as a result of this experiment. A learning agent using these values reaches the destination at an average 90 times of 100 with a penalty rate of 0.026. The agent also gets to the destination in lower net reward i.e. the agent is reaching the destination in the shortest possible time. Hence, the Q-Learning agent has learnt the optimum policy of reaching the destination in shortest time with less number of penalties.