Train a Smartcab to Drive

By: Travis Miller

Special thanks to the Studywolf blog post for explaination of Q-learning and starting code.

https://studywolf.wordpress.com/2012/11/25/reinforcement-learning-q-learning-and-exploration/

Implement a basic driving agent

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

I implemented a random choice action (action = [None, 'forward', 'left', 'right'][random.randint(0,3)]). The smartcab then tried to execute the random action: if the action was a legal action (ex. Going forward on a green) the smartcab was rewarded, otherwise, it was punished for "trying" illegal actions and it was not allowed to move. More over, the smartcab was rewarded higher if it followed the route planner and the action was legal.

Making random moves took a very long time, but it did eventually make it to the destination as expected.

Identify and update state

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

I choose:

- "Next waypoint" because this is the action the smartcab should be trying to take to get to the destination.
- "Color of light" because if the light is red, the smartcab should not be going left or forward
- "Oncoming traffic" because even if the light is green, it should not be turing left when there is oncoming traffic.
- "Traffic from the left" because if the light is red, the smartcab may turn right if there is no traffic coming from the left.

I choose to ignore:

- "Traffic from the right" because there is no case where it matters to look at the right traffic. If the light is green, then traffic on the right needs to stop and doesnt matter. If the light is red, then the smartcab should only be (maybe) making right turns, and the right traffic won't matter
- "Duration" because "The smartcab only has an egocentric view of the intersection it is currently at". This means that even with the duration, the smartcab couldn't predict the next set of traffic lights. Also, we wouldn't want to encourage the smartcab to make illegal moves, so it should just be safe and follow the route planner at its own pace.

Implement Q-Learning

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

At the beginning, it just sat there (choose None) because it was getting rewarded for doing that. It would also sometimes get stuck in a right turn loop which was again because right turns are almost always valid and gave a reward. I changed the default value of actions to be something high (4.0) so that it would choose that action and explore the environment and possible rewards. After awhile, most default values had been filled in and it was making correct descisions.

Enhance the driving agent

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?
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?

I changed the default value of actions to be something high (100.0) so that it would choose that action and explore the environment and possible rewards. I also made it that after each trip, the value of unknown actions would be reduced by 1.0 so that the smartcab would explore less. This bottoms out after 100 trips (the max tolerance stated in the problem) at 0.0 which is less than doing nothing at 1.0. This means that it will avoid accidents in the future instead of being curious about them.

I also re-ordered the actions ([None, 'right', 'forward', 'left']) from most likely to be legal to the actions which may be not be allowed. This is again to try and avoid accidents.

I also made the alpha learning rate arbitrarily 0.5 since getting the large reward at the end of the journey (in this case) doesn't really matter because the smartcab is egocentric and can't really take it into account. As long as the value is above or equal to the default reward rate for “correct” actions (following the next waypoint), the smartcab will make it to the destination.

I find that after 100 trips, the smartcab is consistently making it to the destination and, when in doubt (new state encountered), it will favor doing “None” for a move (since this is the first action index) instead of making an accident.

This is close to the optimal policy of making it to the destination in the minimum possible time and not incurring any penalties. The optimal policy is to follow the “next waypoint” whenever traffic rules will allow it. It would not be optimal to break a traffic rule to reach the destination in time because the penalty could be much larger than the reward. My smartcab does follow the optimal policy, except in the occasion where it encounters a new car after 100 trips when it may do “None”; there are not enough cars on these roads to properly encounter enough of them during the training time.