SmartCab

Implement a Basic Driving Agent

- *Observe what you see with the agent's behavior as it takes random actions. Does the smartcab eventually make it to the destination? Are there any other interesting observations to note?*

  Yes, it reaches the destination, but not all the times within the hardline of 100.

Inform the Driving Agent

- *What states have you identified that are appropriate for modeling the smartcab and environment? Why do you believe each of these states to be appropriate for this problem?*

  I have identified four states:
  - green: in a green light the agent is free to move in any direction.
  - restricted-green: in a green light the agent can't take a left turn.
  - red: in a red light the agent can't move in any direction.
  - expanded-red: in a red light the agent can turn only to the right

  These states are appropriate because they cover all the possible situations for the agent, when it can or can't move and what directions are not allowed.

- *How many states in total exist for the smartcab in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?*

  There are 128 possible states. It doesn't seem reasonable because many of those states are redundant and don't provide any additional information about what actions the agent can or can't take.

Implement a Q-Learning Driving Agent

- *What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?*

  The success rates improve from almost zero to 8%-9%, this is basically because information is being recorded and the model can said to be learning.

Improve the Q-Learning Driving Agent

- *Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?*

  By decaying Epsilon at higher rates -around 75%- and alpha at mid-lower values -around 40%- the success rate is around 25% with a Gamma value of 0.5 this is the set of parameters that perform best, when values move far form those values success rates diminish, more significantly with Gamma values close to 1. Very low epsilon values also reduce the time the model gets to success rates over 30%.

- *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? How would you describe an optimal policy for this problem?*

  My model doesn't gets close to find an optimal policy within the enforced deadline, but it'll eventually find it if enough tries are allowed.

  An optimal policy would be, take a random action when the green light is present with no cars present in any other direction unless a significantly high value is assigned to any specific direction, if there's a car onward going straight or right avoid making a left turn. Do nothing on red, unless there's no car coming from the left and there's a big Q value on the right.