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. It does run some red lights and it doesn't give the right of way all the time but in general it does. In red lights it tends to turn right and in many it stays put in red lights, it may be because it doesn't get a penalty because of that.

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 states based only on the light, left, oncoming and next. These states are appropriate because they cover all the possible restrictions to traffic so I consider them to be significantly more relevant than all the others.

- *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, all the possible combinations of the selected inputs (light, left, oncoming and next). It seems reasonable because it significantly reduces the number of states that would result if all the inputs were considered.

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, it seems to obey traffic rules and seems to move toward the destination.

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 25%- and alpha at mid-lower values -around 40%- the success rate averages around 55% with a Gamma value of 0.75 this is the set of parameters that perform best, when values move far from 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 around 45%.

- *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 too 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 mostly obeying traffic rules and move in the direction of the destination. My model even though it reaches the goal a majority of the times it usually commits some infractions or move in the wrong the direction.