

Arturo Polanco Lozano's Smartcab trainig

QUESTION: 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?

Indeed it does, the red car (our agent) at the begining perform it's actions randomly (None, Forward, Left, Right) across the map and barely perform under the estimated amount of time, after a while it reach the goal of moving to the right location. The confirmation of the Q-Learning algorithm working once adding more trials, measuring it's performance in order to find an optimal behavior it end up with an average score of 19/100 after 100 tries and 5 testing episodes.

QUESTION: 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?

In order to implement the Q-learning agent it's necessary to identify the possible states, once we do it is possible the select the ones that gives us a less complex model without sacrificing the performance. The possible percept variables are: left (None, Forward, Left, Right). Right (None, Forward, Left, Right), Oncoming (None, Forward, Left, Right), Waypoint (None, Forward, Left, Right) and Light (Red, Green). The final number of states can be calculated taking in count the partial states variable, looking behind the we get the following operation $4 \times 4 \times 4 \times 4 \times 2 = 512$, since there are 4 actions available the size of the Q Matrix is 512×4 that would be filled by the agent while the program is running; the process ended up being so sparse and could not converge the Q values with the first tries and in order to achive it it was necessary to add more tries. We need to reduce the number of space taking in count less variables like traffic light, that would make our Q-learning matrix smaller and reduce the negative reward it gets when when that condition is

not accomplish that task.

OPTIONAL: 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?

Exist 512 x 4 states in total, as was mentioned before the final Q matrix is extremely big and that caused a lot of problems. I consider that is important to work with less states.

QUESTION: 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?

Just after a few iterations over the course of the trials the agent performs very well, moving to the destination point quicker than before, taking less steps to achieve it, cumulating positive reward since it can also take negative reward steps and without violating many traffic light rules.

When the position and destination of other cars were not taken into account, instead just using light and waypoint as states the result was an improvement in the performance of 86/100 in 5 episodes of 100 trials and the mode was 93/100. The increase of the performance is related to the reduction of the size of the Q matrix, (8x4 this time).

The input variables are:

The input ['light'] which lets the agent know if the possible movements may be executed (Forward, Left, Right, None) using two variables, green and red in order to show that is allowed or not.

The input variable ['waypoint'] contains the possible movement by the agent, The input ['Forward'] and ['Right'] if the traffic light is 'Green' and there is

no car proceeding straight from those positions, the the movement is allowed. The input ['Left'] variable has to be included in the state because it is legal to turn right on a red light, but only if there is no car proceeding straight from the left.

The input ['oncoming'] wich informs if there is another car coming to our way and lets know if movement the agent is trying to do is possible.

The input ['None'] if none of the previous movements are possible.

The ignored variables are:

Destination because it's changing after each interval, that would defeat the purpose of the agent and wouldn't let it learn.

Location since the changing nature of this variable would create plenty of more states for values to converge and would require more than 100 trials to make it work properly.

When the default reward was not 0 there were importants iprovements in the average, for instance I could set the reward to 30 and get an improvement of 96/100 but with a penalty rate of 0.04 and 2 state variabes. I can conclude that it is more possible to reach the final location but does not do very good with the traffic lights, there are some problems about the omission of it when the agents turn to the right.

It is important to mention that this version doesn't know about nearby traffic at all, which means it has no chance of learning how to handle certain situations correctly, thus it is bound to make mistakes, similarly to an agent with a bigger state space (except the latter can eventually learn how to avoid those mistakes).

QUESTION: 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?

What if the agent doesn't care about the cars?

I fount surprissingly that the parameters values are not so different, the

penalty rate stayed the same as 0.03. When it only uses light and waypoint as states the performance improved to 86/100 in 5 episodes of 100 trials, and the mode to 93/100. This due to a decrease in the size in the Q matrix 8×4 and with more trials it's easier to get a better performance. This confirms that when a less complex model is used (using 2 variables instead of 5) is better for the Q-learning algorithm.

QUESTION: 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?

When is time to compare the q learning agent and the optimal agent taking in count: reaching the final destination in the less time possible and doing it with an accumulative positive reward; is possible to appreciate in the graphic that while more trials were executed the cab could get to the final destination using less steps and improving the time and getting a good accumulative reward. Finally it is possible to say that the optimal policy was learned not only executing the right moves but also taking the shortest route without doing illegal moves.

Over the last 10 trials we can see a tendency of improvement in the fewer number of steps taken for the agent in order to get to the final destination and also reducing the number of violations of the traffic light rules, there is room of improvement if more tries are added.

Finally there is how I tuned the parameters in order to find an optimal behavior, the values where `learning_rate = 0.9`, `discount_factor = 0.33`, and `epsilon = 0.1`.

