

Project 4: Reinforcement Learning

Train a Smartcab How to Drive

Reinforcement Learning Project for Udacity's Machine Learning Nanodegree.

Install

This project requires **Python 2.7** with the pygame library installed.

Code

Template code is provided in the `smartcab/agent.py` python file. Additional supporting python code can be found in `smartcab/enviroment.py`, `smartcab/planner.py`, and `smartcab/simulator.py`. Supporting images for the graphical user interface can be found in the `images` folder. While some code has already been implemented to get you started, you will need to implement additional functionality for the `LearningAgent` class in `agent.py` when requested to successfully complete the project.

Run

In a terminal or command window, navigate to the top-level project directory `smartcab/` (that contains this README) and run one of the following commands:

```
python smartcab/agent.py
python -m smartcab.agent
```

This will run the `agent.py` file and execute your agent code.

Task 1: Implement a Basic Driving Agent

To begin, your only task is to get the smartcab to move around in the environment. At this point, you will not be concerned with any sort of optimal driving policy. Note that the driving agent is given the following information at each intersection:

The next waypoint location relative to its current location and heading. The state of the traffic light at the intersection and the presence of oncoming vehicles from other directions. The current time left

from the allotted deadline. To complete this task, simply have your driving agent choose a random action from the set of possible actions (None, 'forward', 'left', 'right') at each intersection, disregarding the input information above. Set the simulation deadline enforcement, `enforce_deadline` to False and observe how it performs.

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?

I have decided to implement two basic driving agents. One does not take into account the U.S. Right-of-Way rules

```
action = self.selectAction(current_env_state, True)
```

and another one that does take into account the following

- On a green light, a left turn is permitted if there is no oncoming traffic making a right turn or coming straight through the intersection.
- On a red light, a right turn is permitted if no oncoming traffic is approaching from your left through the intersection.

```
action = self.selectAction(current_env_state)
```

This latter option only performs legal action. Therefore it does not get less than -0.5 reward.

To test these basic agents, I have executed it 100 times (trials). In terms of performance metric, I have selected the speed with which the cab can deliver its passenger to its destination.

The results are not very encouraging: most of the times the agent does not reach the final destination, and for those cases that it managed to reach the final destination, it took more than needed.