Machine Learning Engineer Nanodegree

# Smartcab

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

Random agent behavior has been implemented by following code:

```
# TODO: Update state
self.state = inputs['location']
self.heading=inputs['heading']
action = random.choice(Environment.valid_actions)
reward = self.env.act(self, action)
```

Agent is moving randomly. Rules of the environment (traffic light, other traffic) are not respected. After few attempts (5) the agent has not reached destination before the deadline in any of attempts. Actions of the agent are not in line with planner.

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

As a set of states I've picked the following parameters:

- self.next_waypoint: next action suggested by simple planner
- Inputs['light']: traffic light
- Inputs['oncoming']: oncoming traffic
- Inputs['left']: traffic from the left

First parameter allows me to find the direction toward the goal. 3 other parameters give the opportunity to pick actions available in this environment. So on each timestep I have understanding where is my goal and which actions are permitted.

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

The agent starts moving towards the goal and reaching the goal before the deadline.

*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?*

I made the following adjustments:
1) In order to eliminate cycles in car movements on the initial step, I've add 0.1 reward for planner recommendation action from the beginning.

2) I've picked initial action (proposed_action) randomly from valid actions in this environment. After that, I've picked the action with max Q.

No hard coding has been used.

*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?*

Yes: no movements of circles anymore and agent is staying on the same place when it is optimal. I've tried to vary alpha (from 0.1 to 0.9) but no major changes has been observed.

And more detailed observations regarding agent behavior, log of last 5 runs of the agent (self.alpha=0.5, self.gamma=0.5):

Simulator.run(): Trial 0

Environment.reset(): Trial set up with start = (5, 1), destination = (5, 5), deadline = 20

RoutePlanner.route_to(): destination = (5, 5)

LearningAgent.update(): deadline = 20, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (5, 1), 'heading': (0, -1), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 19, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 1), 'heading': (1, 0), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 18, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 2), 'heading': (0, 1), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 17, inputs = {'right': 'forward', 'light': 'green', 'oncoming': None, 'location': (5, 2), 'heading': (-1, 0), 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 16, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (5, 3), 'heading': (0, 1), 'left': None}, action = forward, reward = 2

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 15, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (5, 4), 'heading': (0, 1), 'left': None}, action = forward, reward = 12

Simulator.run(): Trial 1

Environment.reset(): Trial set up with start = (1, 5), destination = (5, 1), deadline = 40

RoutePlanner.route_to(): destination = (5, 1)

LearningAgent.update(): deadline = 40, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (1, 5), 'heading': (1, 0), 'left': None}, action = forward, reward = -1

LearningAgent.update(): deadline = 39, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (1, 5), 'heading': (1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 38, inputs = {'right': 'right', 'light': 'green', 'oncoming': None, 'location': (1, 6), 'heading': (0, 1), 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 37, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (2, 6), 'heading': (1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 36, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (3, 6), 'heading': (1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 35, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (3, 1), 'heading': (0, 1), 'left': None}, action = left, reward = -1

LearningAgent.update(): deadline = 34, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (3, 1), 'heading': (0, 1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 33, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (2, 1), 'heading': (-1, 0), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 32, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (2, 6), 'heading': (0, -1), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 31, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (3, 6), 'heading': (1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 30, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (3, 1), 'heading': (0, 1), 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 29, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (4, 1), 'heading': (1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 28, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (4, 2), 'heading': (0, 1), 'left': None}, action = left, reward = 2

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 27, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (5, 2), 'heading': (1, 0), 'left': None}, action = left, reward = 12

Simulator.run(): Trial 2

Environment.reset(): Trial set up with start = (8, 1), destination = (3, 2), deadline = 30

RoutePlanner.route_to(): destination = (3, 2)

LearningAgent.update(): deadline = 30, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (8, 1), 'heading': (0, 1), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 29, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (7, 1), 'heading': (-1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 28, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (6, 1), 'heading': (-1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 27, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 6), 'heading': (0, -1), 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 26, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (5, 6), 'heading': (-1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 25, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (5, 5), 'heading': (0, -1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 24, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 5), 'heading': (1, 0), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 23, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (6, 6), 'heading': (0, 1), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 22, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (5, 6), 'heading': (-1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 21, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (4, 6), 'heading': (-1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 20, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (3, 6), 'heading': (-1, 0), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 19, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (3, 5), 'heading': (0, -1), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 18, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (3, 4), 'heading': (0, -1), 'left': None}, action = forward, reward = 2

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 17, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (3, 3), 'heading': (0, -1), 'left': None}, action = forward, reward = 12

Simulator.run(): Trial 3

Environment.reset(): Trial set up with start = (6, 5), destination = (3, 3), deadline = 25

RoutePlanner.route_to(): destination = (3, 3)

LearningAgent.update(): deadline = 25, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 5), 'heading': (-1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 24, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (5, 5), 'heading': (-1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 23, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (5, 4), 'heading': (0, -1), 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 22, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (4, 4), 'heading': (-1, 0), 'left': None}, action = forward, reward = 2

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 21, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (3, 4), 'heading': (-1, 0), 'left': None}, action = right, reward = 12

Simulator.run(): Trial 4

Environment.reset(): Trial set up with start = (8, 1), destination = (8, 6), deadline = 25

RoutePlanner.route_to(): destination = (8, 6)

LearningAgent.update(): deadline = 25, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (8, 1), 'heading': (0, 1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 24, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (7, 1), 'heading': (-1, 0), 'left': 'left'}, action = right, reward = 2

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 23, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (7, 6), 'heading': (0, -1), 'left': None}, action = right, reward = 12

Simulator.run(): Trial 5

Environment.reset(): Trial set up with start = (1, 6), destination = (6, 1), deadline = 50

RoutePlanner.route_to(): destination = (6, 1)

LearningAgent.update(): deadline = 50, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (1, 6), 'heading': (0, 1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 49, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (8, 6), 'heading': (-1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 48, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (8, 5), 'heading': (0, -1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 47, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (1, 5), 'heading': (1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 46, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (2, 5), 'heading': (1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 45, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (2, 6), 'heading': (0, 1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 44, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (1, 6), 'heading': (-1, 0), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 43, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (1, 5), 'heading': (0, -1), 'left': 'left'}, action = right, reward = 2

LearningAgent.update(): deadline = 42, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (2, 5), 'heading': (1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 41, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (3, 5), 'heading': (1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 40, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (4, 5), 'heading': (1, 0), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 39, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (5, 5), 'heading': (1, 0), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 38, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (5, 6), 'heading': (0, 1), 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 37, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 6), 'heading': (1, 0), 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 36, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 5), 'heading': (0, -1), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 35, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (6, 4), 'heading': (0, -1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 34, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (7, 4), 'heading': (1, 0), 'left': 'forward'}, action = right, reward = 2

LearningAgent.update(): deadline = 33, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (7, 5), 'heading': (0, 1), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 32, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (6, 5), 'heading': (-1, 0), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 31, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 4), 'heading': (0, -1), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 30, inputs = {'right': None, 'light': 'green', 'oncoming': 'left', 'location': (6, 3), 'heading': (0, -1), 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 29, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (6, 2), 'heading': (0, -1), 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 28, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (7, 2), 'heading': (1, 0), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 27, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (7, 3), 'heading': (0, 1), 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 26, inputs = {'right': None, 'light': 'red', 'oncoming': None, 'location': (6, 3), 'heading': (-1, 0), 'left': None}, action = right, reward = 2

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 25, inputs = {'right': None, 'light': 'green', 'oncoming': None, 'location': (6, 2), 'heading': (0, -1), 'left': None}, action = forward, reward = 12

*For how many (latest) trials were the agent able to run without incurring penalty?*

So for last 3 trials the agent were able to reach destination without negative rewards (penalties). It means that the agent has educated on the 2 first trials.

*How big was the final reward scores of last few trials and how do they compare with earlier trials?*

Final reward score (sum of all rewards point) in the last 2 runs were higher in comparison with first runs (49 in the trial 5 compared to the 22 in the trial 1).

*Did the agent reach destination within allotted time first or following traffic rules correctly first?*

Yes - the agent has reached the destination within the allocated time in the first trial.