Implement a basic driving agent

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

In the beginning, the agents must take random actions since it has no experience. Since it cannot learn it doesn't reach the destination in a reasonable time.

Identify and update state

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

Each state consists of a tuple representing: light, traffic from oncoming, left, right direction next_waypoint, action_taken. There are 2 possibilities for light and 4 possibilities for each of the remaining thus there is 2 * 4 * 4 * 4 * 4 * 4 = 2048 different states. It would be possible to omit e.g. traffic from right at the cost of increasing the number of penalties the agent incurs. Including the deadline would increase the state space by a large amount. I chose this set of states to give the agent an ability to learn from past experiences. Each state represents a situation that can occur at the intersection.

Implement Q-Learning

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

After implementing Q-learning, the agent got the ability to use what it learned. It did not have to rely only on taking random actions. It could use the learned Q values to take into experience what it had seen before. Compared to the random agent, it reaches the final destination in most trials. The agent starts to obey traffic laws, the random agent did not at all. The agent seems to take the shortest path to the destination now although it is still limited by the size of state space and it has not explored all situations.

Enhance the driving agent

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 tried setting different values for alpha and gamma. I measured the performance by observing the ratio of destinations reached / trials and the ratio of penalties / number of moves. The log given below is for alpha = 0.7 and gamma = 0.1. Setting alpha = 0 and gamma = 0.1 shows the agent reaching the destination only in 25.5 % of trials with a penalty ratio of 81.55 %. Setting alpha = 1 and gamma = 0.1 shows the agent reaching the destination in 95 % of trial with a penalty ratio of 7.3 %.

The performance of the agent as measured by the reached / trials ratio varies from execution to execution due to random choices involved. After fixing the Q algorithm, I tried changing the parameters alpha and gamma and obtained the following results (the list shows the number of penalties incurred for trials):

Alpha = 0.1, gamma = 0.9

[5, 15, 9, 7, 7, 9, 8, 6, 7, 7, 5, 10, 15, 6, 11, 1, 2, 2, 2, 5, 6, 5, 3, 0, 2, 1, 1, 3, 2, 2, 2, 1, 0, 2, 0, 1, 1, 0, 2, 0, 2, 1, 0, 0, 0, 3, 1, 0, 1, 3, 0, 0, 0, 1, 2, 4, 3, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 2, 2, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 2, 1, 0, 0, 1, 0, 1, 0, 0, 0, 5, 0, 0, 0, 1, 1, 1, 0]

Penalties mean: 2.24242424242 var: 9.92022263451

Alpha = 0.5,gamma = 0.5

[9, 20, 15, 12, 2, 24, 12, 5, 4, 5, 0, 7, 7, 7, 1, 1, 6, 3, 0, 1, 2, 1, 0, 0, 1, 3, 0, 0, 1, 0, 1, 0, 1, 0, 8, 4, 0, 0, 4, 0, 0, 2, 1, 0, 0, 2, 3, 0, 4, 0, 2, 1, 2, 1, 2, 3, 3, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 3, 1, 1, 4, 2, 0, 0, 0, 2, 2, 1, 1, 0, 0, 2, 0, 2, 0, 0, 1, 1, 1, 0, 0, 2, 2, 2]

Penalties mean: 2.30303030303 var: 15.9072356215

Alpha= 0.1, gamma = 0.1

[1, 4, 0, 1, 0, 0, 2, 0, 4, 0, 2, 5, 1, 0, 5, 0, 1, 3, 0, 5, 5, 5, 0, 4, 0, 0, 4, 1, 1, 0, 5, 0, 1, 3, 0, 0, 6, 2, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 3, 1, 0, 0, 0, 1, 1, 0, 0, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 5, 0, 2, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 0, 4, 0, 0]

Penalties mean: 1.11111111111 var: 2.589569161

Based on this listing, I think setting the parameters to alpha = 0.1 and gamma = 0.1 is the best choice because it gives the smallest average number of penalties with smallest variance in penalties.

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, the agent gets close to finding an optimal policy. It did find the destination in each of last ten trials. It did incur some penalties since the agent is limited by time and by the number of elements in the state space and 100 trials is not enough to learn everything. For example, it had not yet encountered a situation with traffic incoming from all directions simultaneously e.g. input {'light': green, 'oncoming': 'forward', 'left': 'forward', 'right': 'forward'} with next_waypoint showing 'forward'.

Log of last 10 trials

Simulator.run(): Trial 90

Environment.reset(): Trial set up with start = (8, 4), destination = (1, 5), deadline = 40

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

LearningAgent.update(): deadline = 40, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = left, reward = -1.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = left, reward = -1.0

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

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = left, reward = 9.5

Simulator.run(): Trial 91

Environment.reset(): Trial set up with start = (2, 3), destination = (6, 3), deadline = 20

RoutePlanner.route_to(): destination = (6, 3)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = -0.5

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

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

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

reached 81/ current 90 = 90.0%

penalty 243/ num_moves 1389 = 17.49%

length of Q: 62

Simulator.run(): Trial 92

Environment.reset(): Trial set up with start = (5, 6), destination = (6, 2), deadline = 25

RoutePlanner.route_to(): destination = (6, 2)

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

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = -0.5

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'left'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

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

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

reached 82/ current 91 = 90.1%

penalty 245/ num_moves 1405 = 17.44%

length of Q: 62

Simulator.run(): Trial 93

Environment.reset(): Trial set up with start = (4, 5), destination = (8, 6), deadline = 25

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

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

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

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

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

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 12.0

reached 83/ current 92 = 90.2%

penalty 247/ num_moves 1420 = 17.39%

length of Q: 63

Simulator.run(): Trial 94

Environment.reset(): Trial set up with start = (6, 3), destination = (2, 4), deadline = 25

RoutePlanner.route_to(): destination = (2, 4)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = -0.5

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

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

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 12.0

reached 84/ current 93 = 90.3%

penalty 248/ num_moves 1429 = 17.35%

length of Q: 63

Simulator.run(): Trial 95

Environment.reset(): Trial set up with start = (4, 5), destination = (3, 1), deadline = 25

RoutePlanner.route_to(): destination = (3, 1)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

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

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

reached 85/ current 94 = 90.4%

penalty 248/ num_moves 1439 = 17.23%

length of Q: 63

Simulator.run(): Trial 96

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

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

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

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 9.5

Simulator.run(): Trial 97

Environment.reset(): Trial set up with start = (8, 1), destination = (6, 4), deadline = 25

RoutePlanner.route_to(): destination = (6, 4)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = -0.5

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

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

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

reached 86/ current 95 = 90.5%

penalty 249/ num_moves 1446 = 17.22%

length of Q: 63

Simulator.run(): Trial 98

Environment.reset(): Trial set up with start = (6, 2), destination = (2, 2), deadline = 20

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

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

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

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

reached 87/ current 96 = 90.6%

penalty 252/ num_moves 1456 = 17.31%

length of Q: 63

Simulator.run(): Trial 99

Environment.reset(): Trial set up with start = (1, 1), destination = (6, 3), deadline = 35

RoutePlanner.route_to(): destination = (6, 3)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2.0

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

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

reached 88/ current 97 = 90.7%

penalty 252/ num_moves 1464 = 17.21%

length of Q: 63