What you see in the agent’s behavior. Does it eventually make it to the target location?

The Basic Agent is an agent with completely random moves which does not obey the traffic rules. This agent, as expected, behaved erratically and rarely reached to its destination.

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

For modeling an agent which can behave in a more intelligent manner, a set of states must be defined. Ideally, these states should contain all of the necessary information about the environment that is needed for choosing the correct action that eventually learn the environment and reaches its destination without acid end and on time. The states are as follow:

1. The state of the traffic light (either 'Green' or 'Red') - Green = NS open, Red = EW open
2. The state of oncoming traffic.
3. The state of traffic om the right.
4. The state of traffic om the left.
5. The direction recommended by the route planner

The first four states give enough information to the smart cab to learn the traffic laws. The cab learns If the light is green or red, and if it is green, the cab can freely go straight through the intersection or make a right turn. However, the left turn on green depends on the state of the on-coming cars. If the light is red, the cab can make a right turn, if there are no cars on the left, on the right, and on-coming cars.

Adding the next\_waypoint state allows the smart cab to make the shortest route towards the destination.

For example, one of the states my agent can experience is ('green', None, None, 'forward', 'forward').

**Note:** I originally picked “deadline” as one of the states. The size of state space with 5 states above plus the deadline; is 10 times larger than just using the 5 states mentioned above. Also deadline in the state set (e.g. 'green', None, None, 'forward', 24, 'forward') does not provide any useful information needed for the agent to take the right actions. Q-matrix also does not get enough Q\_table entries with 100 trials.

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

After implementing a simple version of Q-Learning (and an action selection method that gradually becomes less explorative as the number of iterations approaches 100), the behavior of the cab greatly improved:

* It started to obey the traffic lights
* Followed the directions provided by the route planner

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?

Applied large number of parameter combinations for Q-learning. The parameters were alpha, gamma, and epsilon.  With initial value of 0.

hyper\_params={"init\_values":[0],"epsilons":[.1,.2,.4],"alphas":[.3,.7,.9],"gammas":[.4,.7,.9]}

The program generates 27 combination of parameters. Ran the program with this combination of parameters and achieved the best performance with the following parameters:

init\_value = 0, epsilon=0.2, alpha=0.7, gamma=0.7.

The smartcab reached the destination about 75 times in 100 trials with this parameter. This parameters scored the highest accumulated rewards in 100 trials (3359). There are also other parameters that helps the agent arrives at the destination more than 75 time e.g. 90 times, however the total rewards are less than the best parameter total rewards.

High total rewards indicate that the agent has learned better on how to follow the traffic laws and planner’s suggested moves.

The list below shows the top 5 performance of the smartcab reaching the final destination on time using different set of parameters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| epsilon | alpha | gamma | Total Steps | Reached Dest. | Total Reward |
| 0.2 | **0.7** | **0.7** | 28982 | 74 | 3359 |
| 0.1 | 0.7 | 0.9 | 36266 | 55 | 3261 |
| 0.1 | 0.3 | 0.4 | 22589 | 82 | 3245 |
| 0.2 | 0.3 | 0.4 | 19505 | 82 | 3174.5 |
| 0.1 | 0.7 | 0.7 | 33482 | 57 | 3171.5 |

The full list of parameter results can be found in *Smartcab Perfrmance Report.csv* file*.*

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?

The Q\_Table output below shows the environment states and smartcab actions with 100 trials and with the best performed parameters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | State:  light, oncoming, left, right, next waypoint | Action: None | Action:  Forward | Action:  Left | Action:  Right |
| 1 | ('green', None, 'forward', None, 'forward') | 0 | 5.872452 | 4.014144 | 0 |
| 2 | ('green', None, None, 'right', 'forward') | 0 | 6.818867 | 0 | 0 |
| 3 | ('green', None, None, 'left', 'right') | 0 | 0 | 0 | 4.29575 |
| 4 | ('green', None, None, 'forward', 'left') | 0 | 3.849917 | 0 | 0 |
| 5 | ('green', None, None, 'right', 'right') | 0 | 0 | 0 | 0 |
| 6 | ('red', 'forward', None, None, 'forward') | 2.693203 | 0.103447 | -0.7 | 0 |
| 7 | ('green', None, None, None, 'left') | 3.910098 | 3.54343 | 5.188305 | 3.63423 |
| 8 | ('red', 'left', None, None, 'forward') | 2.034997 | -0.7 | 0 | 0 |
| 9 | ('red', None, 'left', None, 'right') | 0 | 0 | 0 | 0 |
| 10 | ('red', None, None, 'left', 'forward') | 1.253 | -0.7 | -0.7 | 0 |
| 11 | ('red', 'left', None, None, 'right') | 0 | -0.7 | 0 | 5.269926 |
| 12 | ('green', None, None, 'forward', None) | 0 | 0 | 0 | 0 |
| 13 | ('red', None, None, 'forward', 'left') | 0.7 | -0.7 | 0 | 0 |
| 14 | ('red', None, None, 'right', 'left') | 1.253 | 0 | 0 | 0 |
| 15 | ('red', None, 'right', None, 'left') | 1.68987 | 0 | 0 | 0 |
| 16 | ('green', 'right', None, None, 'forward') | 0 | 0 | 0 | 2.688991 |
| 17 | ('red', 'left', None, None, None) | 0 | 0 | 0 | 0 |
| 18 | ('red', 'right', None, None, 'left') | 0.7 | 0 | 0 | 0 |
| 19 | ('red', None, None, None, 'left') | 3.525956 | 1.301237 | 1.633093 | 4.386814 |
| 20 | ('red', None, None, 'forward', 'forward') | 0 | 0 | 0 | 0 |
| 21 | ('green', None, None, None, 'forward') | 4.474649 | 5.319602 | 4.475034 | 3.705748 |
| 22 | ('green', None, None, 'forward', 'forward') | 0 | 4.543597 | 0 | 2.819763 |
| 23 | ('green', None, 'left', None, None) | 0 | 0 | 0 | 0 |
| 24 | ('red', None, None, 'right', 'right') | 0.7 | 0 | 0 | 0 |
| 25 | ('red', None, 'forward', None, 'forward') | 0 | -0.7 | 0 | 0 |
| 26 | ('green', None, None, None, None) | 0 | 0 | 0 | 0 |
| 27 | ('green', 'right', None, None, 'right') | 0 | 0 | 0 | 0 |
| 28 | ('red', 'forward', None, None, 'left') | 0 | 0 | 0.404902 | 3.693613 |
| 29 | ('green', 'forward', None, None, 'forward') | 0 | 0 | 0 | 0.35 |
| 30 | ('red', None, None, None, 'right') | 5.161774 | 2.488723 | 3.707366 | 5.240708 |
| 31 | ('red', None, 'left', None, 'left') | 0.7 | 0 | 0 | 0 |
| 32 | ('green', None, 'right', None, 'left') | 0.7 | 0 | 0 | 0 |
| 33 | ('green', None, 'left', None, 'right') | 2.669737 | 4.01031 | 2.8659 | 3.053049 |
| 34 | ('red', 'forward', None, None, 'right') | 2.034997 | -0.7 | 0 | 0 |
| 35 | ('green', None, 'left', None, 'left') | 2.021528 | 0 | 0 | 2.696996 |
| 36 | ('green', None, None, 'forward', 'right') | 0 | 1.983759 | 0 | 0 |
| 37 | ('green', None, 'right', None, 'right') | 0 | 0 | 3.120744 | 0 |
| 38 | ('green', None, 'left', None, 'forward') | 0 | 0 | 0.35 | 4.545669 |
| 39 | ('red', None, None, None, None) | 0 | 0 | 0 | 0 |
| 40 | ('green', None, None, 'left', 'left') | 0 | 2.542082 | 0 | 0 |
| 41 | ('green', None, None, 'left', 'forward') | 0 | 5.980396 | 0 | 0 |
| 42 | ('red', None, 'right', None, 'right') | 0 | -0.7 | 0 | 0 |
| 43 | ('green', 'left', None, None, 'forward') | 3.970959 | 0 | 4.433432 | 0 |
| 44 | ('green', 'right', 'forward', None, 'left') | 0 | 0 | 0 | 0 |
| 45 | ('green', None, None, 'right', 'left') | 0 | 0 | 3.998438 | 0 |
| 46 | ('red', None, None, None, 'forward') | 3.374262 | 1.339678 | 1.506678 | 3.721498 |
| 47 | ('red', 'right', None, None, 'forward') | 0 | 0 | 0 | 3.349343 |
| 48 | ('red', None, 'left', None, 'forward') | 1.253 | 0 | 0 | 0 |
| 49 | ('red', None, None, 'left', 'right') | 0.7 | 0 | 0 | 0 |
| 50 | ('green', 'forward', None, None, 'left') | 0 | 0 | 0 | 0 |
| 51 | ('green', None, 'forward', None, 'left') | 0 | 0 | 7.502148 | 0 |
| 52 | ('green', 'right', None, None, 'left') | 0 | 5.462706 | 0 | 0 |
| 53 | ('red', None, 'forward', None, 'right') | 0.7 | 0 | 0 | 0 |
| 54 | ('red', None, 'right', None, 'forward') | 0 | 0 | 0 | 0 |
| 55 | ('red', 'right', None, None, 'right') | 0.7 | -0.7 | -0.7 | 0 |
| 56 | ('green', None, 'forward', None, 'right') | 0 | 2.570357 | 0 | 0 |
| 57 | ('green', None, None, None, 'right') | 4.334538 | 3.88283 | 3.58081 | 5.513896 |
| 58 | ('green', 'forward', None, None, 'right') | 0 | 2.710206 | 0 | 0 |
| 59 | ('green', 'left', None, None, 'right') | 0 | 4.123781 | 0 | 0 |
| 60 | ('green', None, 'right', None, 'forward') | 0 | 0 | 0 | 3.810757 |

The Q\_table only has 60 rows. With all combination of the states (light, oncoming, left, right, next waypoint), we should have 2x4x4x4x3 or 384 states. Even though the smartcab reaches the destination on time, it is still not fully follows the traffic lights and planner’s next suggested move. For example, at row #58, the smartcab moves forward (the highest score in the action functions 2.710) even though the next waypoint is ‘right’. Another example is the row #47 where the traffic light is red but the smartcab moves forward (the highest score in the action functions 3.349).

By running more trials, we can achieve a more complete Q\_table and eventually a converge. The agent occasionally makes mistakes when encountering other cars, even after 100 iterations.

The chart below shows the statistics of the last 10 trials.

The agent tries to follow the rules of the road and have a few -1 rewards.

The agent tries to follow planner’s suggested moves to get 2 rewards.

The agent tries to follow traffic laws and take an action to get .5 rewards.

The agent tries to follow traffic lights and does not take any actions to get 1 reward.

The table below also shows that the agent tries to take the shortest route (more 2 rewards than .5 rewards on each trial) to the destination.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trial # | Number of Reward = -1 | Number of Reward = .5 | Number of Reward = 1 | Number of Reward = 2 | Number of Reward = 12 |
| Trial 90 | 0 | 0 | 1 | 4 | 1 |
| Trial 91 | 2 | 7 | 13 | 9 | 0 |
| Trial 92 | 2 | 1 | 1 | 8 | 1 |
| Trial 93 | 1 | 4 | 1 | 4 | 1 |
| Trial 94 | 0 | 7 | 2 | 8 | 1 |
| Trial 95 | 0 | 4 | 0 | 5 | 1 |
| Trial 96 | 0 | 8 | 1 | 14 | 1 |
| Trial 97 | 0 | 15 | 1 | 10 | 0 |
| Trial 98 | 3 | 2 | 6 | 10 | 0 |
| Trial 99 | 1 | 3 | 6 | 4 | 1 |
| Total | 9 | 51 | 32 | 76 | 7 |

Simulator.run(): Trial 90

Environment.reset(): Trial set up with start = (5, 2), destination = (1, 3), deadline = 25

RoutePlanner.route\_to(): destination = (1, 3)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

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

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 12

Simulator.run(): Trial 91

Environment.reset(): Trial set up with start = (1, 3), destination = (7, 3), deadline = 30

RoutePlanner.route\_to(): destination = (7, 3)

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

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

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

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

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

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': 'right', 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': 'right', 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', '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

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'left'}, action = left, reward = 0.5

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

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

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

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

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

LearningAgent.update(): deadline = 1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 0.5

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

Environment.reset(): Primary agent could not reach destination within deadline!

Simulator.run(): Trial 92

Environment.reset(): Trial set up with start = (7, 5), destination = (4, 1), deadline = 35

RoutePlanner.route\_to(): destination = (4, 1)

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = -1

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

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

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12

Simulator.run(): Trial 93

Environment.reset(): Trial set up with start = (4, 3), destination = (7, 5), deadline = 25

RoutePlanner.route\_to(): destination = (7, 5)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

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': 'left'}, action = forward, reward = 0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1

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

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

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

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

Simulator.run(): Trial 94

Environment.reset(): Trial set up with start = (2, 3), destination = (5, 6), deadline = 30

RoutePlanner.route\_to(): destination = (5, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

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

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

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

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

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

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

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

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 12

Simulator.run(): Trial 95

Environment.reset(): Trial set up with start = (5, 6), destination = (1, 4), deadline = 30

RoutePlanner.route\_to(): destination = (1, 4)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'right'}, action = right, reward = 0.5

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

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

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12

Simulator.run(): Trial 96

Environment.reset(): Trial set up with start = (3, 4), destination = (7, 3), deadline = 25

RoutePlanner.route\_to(): destination = (7, 3)

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

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

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 = 0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

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

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

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

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

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

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

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

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

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 12

Simulator.run(): Trial 97

Environment.reset(): Trial set up with start = (2, 1), destination = (6, 2), deadline = 25

RoutePlanner.route\_to(): destination = (6, 2)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

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

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = 0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

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

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

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

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'left'}, action = forward, reward = 0.5

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 0.5

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

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

Environment.reset(): Primary agent could not reach destination within deadline!

Simulator.run(): Trial 98

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

RoutePlanner.route\_to(): destination = (2, 2)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

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

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

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

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

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

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

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

Environment.reset(): Primary agent could not reach destination within deadline!

Simulator.run(): Trial 99

Environment.reset(): Trial set up with start = (6, 3), destination = (2, 1), deadline = 30

RoutePlanner.route\_to(): destination = (2, 1)

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

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1

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

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 2

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 2

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

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

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = None, reward = 1

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

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

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

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = right, reward = 0.5

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

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = left, reward = 12

number of successful customer drop offs: 73

Total Steps: 25182 in 100 trial