UDACITY

PROJECT

Train a Smartcab to Drive

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

Implement a basic driving agent

Student is able to implement the desired interface to the agent that accepts specified inputs.

Awesome!

The driving agent produces a valid output (one of None, 'forward', 'left', 'right') in response to the inputs.

Great work!

The driving agent runs in the simulator without errors. Rewards and penalties do not matter - it's okay for the agent to make mistakes.

Excellent! The code runs with no error.

Identify and update state

Student has identified states that model the driving agent and environment, along with a sound justification.

Good analysis so far! Although deadline does not need to be included into the state space, a discussion on why it wasn't included is needed here. Also please talk about what would happen to the agent if any of these inputs were omitted, would it still be able to learn optimally? This goes along the lines of what states receives rewards.

The driving agent updates its state when running, based on current input. The exact state does not matter, and need not be correlated with inputs, but it should change during a run.

Great work here!

Implement Q-Learning

The driving agent updates a table/mapping of Q-values correctly, implementing the Q-Learning algorithm.

Awesome implementation of the Q-Learning algorithm.

Given the current set of Q-values for a state, it picks the best available action.

Student has reported the changes in behavior observed, and provided a reasonable explanation for them.

Enhance the driving agent

The driving agent is able to consistently reach the destination within allotted time, with net reward remaining positive.

Great Work! The agent consistently reaches its destination.

Specific improvements made by the student beyond the basic Q-Learning implementation have been reported, including at least one parameter that was tuned along with the values tested. The corresponding results for each value are also reported.

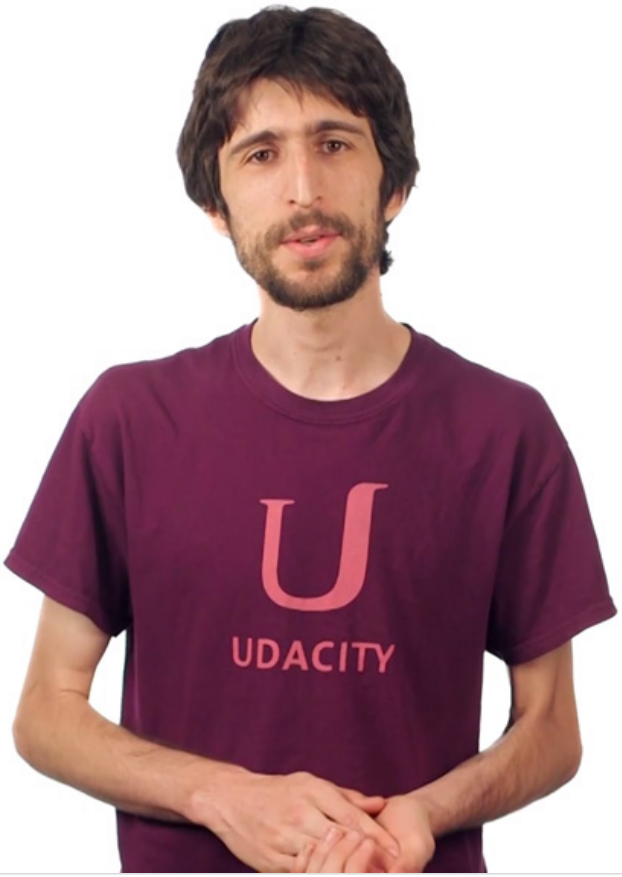
Awesome table here, just include in your analysis and table the utilities/rewards. This way it can demonstrate if the agent is following the necessary traffic laws while reaching the its destination within the allotted time.

A description is provided of what an ideal or optimal policy would be. The performance of the final driving agent is discussed and compared to how close it is to learning the stated optimal policy.

Great job providing what an optimal policy would look like. This is being marked off, because what's also needs to be discussed is whether the agent is acting in a sub-optimal way. Does the agent break laws under certain circumstances? Does the agent go non ideal routes to accumulate more reward?

RESUBMIT

DOWNLOAD PROJECT



Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Rate this review