UDACITY

PROJECT

Train a Smartcab to Drive

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Overall, well done!. You demonstrate a good understanding of Reinforcement Learning concepts and your report is written in explanatory terms allowing your audience to understand the work done and results.

Congratulations!

Implement a basic driving agent

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

Great!

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

The agent produces all of the valid outputs!

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

okay for the agent to make mistakes.

No errors when running the code!

Identify and update state

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

Well done recognizing valid inputs for your agent' state, including reasons for them as well as reasons for not including other inputs like deadline.

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.

The states update in the pygame window.

Implement Q-Learning

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

Correct implementation of the Q-Learning algorithm.

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

Great!

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.

Excellent!

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.

Very clear ideas here and great work building your metrics to benchmark your algorithm with the different tuning parameters , awesome!

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.

Well done recognizing how the optimal policy looks like and discussing how close your agent is.

DOWNLOAD PROJECT

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

RETURN TO PATH