UDACITY

PROJECT

Teach a Quadcopter How to Fly

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW 8

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Amazing job!

You presented an enjoyable submission, packed with information, thoughts and ideas.
I hope this project inspired you to keep researching this promising branch of Deep Learning, with innovations being published continuously.

Keep the good work!

Define the Task, Define the Agent, and Train Your Agent!

The `agent.py` file contains a functional implementation of a reinforcement learning algorithm.

Awesome

- Good work implementing **Deep Deterministic Policy Gradient** algorithm not only once but two times, one for testing on the `MountainCarContinuous-v0` and later for the `Custom_Task` (takeoff and hover)

The `Quadcopter_Project.ipynb` notebook includes code to train the agent.

Awesome

- The notebook code correctly trains the agent on the **Takeoff and hover** task 👍

Plot the Rewards

A plot of rewards per episode is used to illustrate how the agent learns over time.

Awesome

- Amazing work plotting not only the **Rewards**, but also the **Position** of the agent on different stages of the learning process 👏
- You also improved the readability of the plots tweaking the design, and properly labeling each one 👍

Reflections

The submission describes the task and reward function, and the description lines up with the implementation in `task.py`. It is clear how the reward function can be used to guide the agent to accomplish the task.

Awesome

- Outstanding work narrating in great detail, but also in a clear way, your thought process for designing the **reward function** 👏
- here are some Tips and Tricks on writing reward functions ⭐

The submission provides a detailed description of the agent in `agent.py`.

Awesome

- Again, incredible work **showing the different stages** of evolution on your agent, **citing your sources**, and **describing the results** 👏

The submission discusses the rewards plot. Ideally, the plot shows that the agent has learned (with episode rewards that are gradually increasing). If not, the submission describes in detail various attempted settings (hyperparameters and architectures, etc) that were tested to teach the agent.

Awesome

- Do not worry if you model did not learned to hover. Being able to go up without drifting in X, Y axis is already a great achievement taking into account the **dynamics** of the quadcopter, and how sensitive it is to the differences between rotor speeds.
- Will be great if you add information about the **Replay Buffer**. One interesting thing to try could be Prioritized Experienced Replay
    - From the description: *In this paper we develop a framework for prioritizing experience, so as to replay important transitions more frequently, and therefore learn more efficiently.*

A brief overall summary of the experience working on the project is provided, with ideas for further improving the project.

Awesome

- Thank you for sharing your thoughts on the project. You did an amazing job, even with the added difficulty of the undocumented Simulator! 👍

DOWNLOAD PROJECT

8 CODE REVIEW COMMENTS ›

RETURN TO PATH

Student FAQ