



[◀ Return to "Machine Learning Engineer Nanodegree" in the classroom](#)

Teach a Quadcopter How to Fly

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Everything looks good. Unfortunately, there are not any direct solutions to these kind of problems. You have to reach a solution through your intuition and lots of trial and error.

Since you have completed this project, Now if you want to know how to apply these techniques to real-world problems. For that, I would advise you to go through [Deep Reinforcement Learning for Self Driving Car by MIT](#). So that you can know more about reinforcement learning algorithms in broader and real-world perspective.

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

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

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

Plot the Rewards

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

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.

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

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.

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

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)