

[< Back to Machine Learning Engineer Nanodegree](#)

# Teach a Quadcopter How to Fly

## REVIEW

### CODE REVIEW 1

### HISTORY

## Meets Specifications

It would have been if you plot the reward for ~500 episodes. It would help you to better track the performance of the agent.

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



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

### Awesome

- Effectively actualized both the Actor and Critic network for DDPG
- The implementation is highly modular, making it simple to troubleshoot and effortlessly scalable.



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

### Awesome

- The code to train the agent is attached.

## Plot the Rewards



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

### Awesome

- The plot of rewards is attached.

## 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

- Great job with the task.
- The whole task was quite well written with all the different functions and conditions defined clearly.
- The reward function was also quite well explained in the Jupyter notebook.



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

### Awesome

- The choice of parameters was good. The choice of using DDPG for the agent's continuous action space was a good choice.
- It was good to see that you didn't use the default set of hyperparameters.
- Manipulation of hyperparameters is a great learning exercise and helps to understand its impact on the performance of the agent.



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

- The agent performs quite well and is able to learn quickly.
- You have a keen eye on the problem domain and parameter for the undertaken task, which is a great asset in RL.



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

## Awesome

- You did a great job on this project.
- The code was quite modular which made it easy to read and debug.
- You have a keen eye on the domain knowledge and required parameters to get the job done.
- All the points in the Jupyter notebook were quite well articulated.

 [DOWNLOAD PROJECT](#)



[CODE REVIEW COMMENTS](#)



[RETURN TO PATH](#)