UDACITY

‹ Return to "Machine Learning Engineer Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

Teach a Quadcopter How to Fly

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations! Your submission passed all requirements and I believe you showed well how you dedicated yourself for this project!

Great future projects for you and I hope some of my comments help you in your goals. :)

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

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

`agent.py` was renamed as `DDPG.py`, and it contains a functional implementation of RL. Great job!

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

The provided code runs, excellent work you now have a state-of-the-art RL algorithm!

Plot the Rewards

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

You provided a extensive discussion about the learning process of your model, and some of your thoughts are interesting conclusions that you achieved.

About `To try stabilize the movements I added penalties for angular velocities and angular positions.`, the penalty is a technique apt to situations like this, for example, you don't want your drone to crash as you will break it, but does it really "stabilize" it?

For example, you are using all three dimensions on your reward functions, are all required? Wouldn't it be more stabilize driven if you penalize horizontal movements and apply distance rewards only considering on vertical movement?

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.

Your reward function `np.tanh(3. - 0.003*(abs(self.sim.pose[:3] - self.target_pos)).sum())` is interesting as it's normalized by tanh and `3` and `0.003` how did you reached this values?

Removing my question, it's clear that your reward function could drive the learning required for the task.

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

Your description is not through but I really like your part where "What apparently did help was: " It's very important to notice which techniques provide more enhancement, and learn how to apply techniques to the different tasks is a great knowledge!

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.

You are correct, normally DDPG has this kind of behavior of suddenly discovering the expected path and it will use that more often with some random search steps sometimes.

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

When you say "I don't complain about challenging projects as long as I have an appropiate knowledge base to solve them." I know you are in a course, and normally in courses, all knowledge is given to you directly, but I believe Udacity has a little bit different approach. Project need to be challenging and need to drive you to look for extra information, to learn how to find answers in a great part o a Machine Learning Engineer (I can say that I spend daily a couple of hours looking for and reading papers).

I'm happy that you took this challenge an created a great project and showed that you learned a lot about it! Congratulations!

DOWNLOAD PROJECT

RETURN TO PATH