UDACITY

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

DISCUSS ON STUDENT HUB

Teach a Quadcopter How to Fly

REVIEW

CODE REVIEW 5

HISTORY

Meets Specifications

Hello,

You've done a great work by implementing the DDPG and to teach quadcopter how to fly. It is very rare to see such a good performance of the Quadcopter in this challenging project. You have really gone far with changing the network architecture and that was the key for the good performance. I hope this project helped you in building your intuition towards reward function, hyperparameters and network architecture. It was great to see that you have provided every answer in so much detail.

For the next step, I would highly recommend you to solve Bi-Pedal-Walker environment of OpenAi. This will surely take you to one more level in Reinforcement Learning.

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

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

Awesome

- Successful implementation of the DDPG algorithm.
- Actor and critic network are correctly implemented.

- It was good to use a target network for both actor and critic network as suggested by the author of DDPG Paper.
- Further using soft updates for the target network was a good choice.

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.

Awesome

- Task and Reward function are explained properly.

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

Awesome

- Good job by choosing the DDPG algorithm for continuous state space problem.
- Good job by including hyperparameters and network architecture in the notebook.

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

- It is quite common to see agent not learning ideally in `Quadcopter` project since it is not an easy project. And you have done a good job so far.
- Performance of the agent is very impressive. The agent is able to learn the task quite well.

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

DOWNLOAD PROJECT

5 CODE REVIEW COMMENTS ›

RETURN TO PATH