UDACITY

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

Teach a Quadcopter How to Fly

REVIEW

CODE REVIEW 8

HISTORY

Meets Specifications

Dear Udacian, you have done a tremendous job completing the project. 🌟

The project is a difficult one and you have done a good job combining various concepts such as Actor,Critic, ReplayBuffer, OUNoise and DDPG Agent in making your agent learn and perform the task you have defined successfully. 👍🏽

The algorithm you have implemented is Deep Deterministic Policy Gradients algorithm with experiene replay. Able to run the code and train the agent successfully. The task you have implemented is takeoff and it is very neatly presented in the notebook. ⭐

Good job in providing various visualisations showing the behaviour of agent after training gets completed. 👍🏽

I am happy to say that I am marking your submission successful as the agent successfully learns and performs the task you have defined for it.

I suggest you to check this blog post. It's very good and should help you with experimenting more.

All the best for the future projects! 🍀

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

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

Awesome

- A functional implementation a reinforcement learning algorithm is found in the file agent.py
- Deep Deterministic Policy Gradients, a reinforcement learning algorithm which is found to be good for continuous state space and action space is used. Good job here!
- Experience replay is used in the algorithm. Good job here!!

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

Awesome

- The task implemented is takeoff. Good job in choosing simple task in the beginning. It helps to understand the concepts faster.
- It's been very neatly presented in the notebook Quadcopter_Project.ipynb.
- Able to successfully run the code and train the agent.

Plot the Rewards

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

Awesome

- Thanks for providing the plot of rewards per episode which shows how the agent learned over time.
- The rewards obtained were low in the beginning and went high and then becomes stable indicating how the agent improved.

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

- The task implemented in the project is takeoff.
- Description of the task and the formulated reward function is found in the submission.
- The reward function used makes sense and is good enough to make the agent learn the desired task. Great job here!

Great job here!

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

Awesome

- Good job providing the details of the implemented agent.
- The answer gives the details of the algorithm used, hyper-parameters and architectural information of the model.

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 reward plot obtained shows improvement with the number of episodes.
- Episode rewards were low in the beginning and then improved showing how the agent has learnt over number of episodes.

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

Awesome

- Thanks for providing the details.
- Lots of learning is required by the project and it also requires combining of a lot of concepts.
- You have done a great job doing the project.
- You have summarized your experience well here.

DOWNLOAD PROJECT

8 CODE REVIEW COMMENTS ›

RETURN TO PATH