



[< Back to Deep Reinforcement Learning Nanodegree](#)

Collaboration and Competition

REVIEW

CODE REVIEW

HISTORY

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

Awesome Work!

Overall its a nicely done project. your results are promising! Good work! Keep digging! 🙌 🍷

Training Code

The repository includes functional, well-documented, and organized code for training the agent.

Awesome

The code for training the agent is functional, well-documented and provides great results!

The code is written in PyTorch and Python 3.

Python3 and PyTorch have been used.

The submission includes the saved model weights of the successful agent.

Model weights have been included.

README

The GitHub submission includes a `README.md` file in the root of the repository.

README is included in the repository.

The README describes the the project environment details (i.e., the state and action spaces, and when the environment is considered solved).

README is informative, all the details about the environment are covered

The README has instructions for installing dependencies or downloading needed files.

Instructions to install the dependencies and downloading the required files have been provided.

The README describes how to run the code in the repository, to train the agent. For additional resources on creating READMEs or using Markdown, see [here](#) and [here](#).

README describes how to run the code in the repository, to train the agent! 🙌

Report

The submission includes a file in the root of the GitHub repository (one of `Report.md`, `Report.ipynb`, or `Report.pdf`) that provides a description of the implementation.

The report is included in the repository as a markdown file.

The report clearly describes the learning algorithm, along with the chosen hyperparameters. It also describes the model architectures for any neural networks.

Report clearly defines learning algorithm, chosen hyper parameters. Nice work there! 🙌

A plot of rewards per episode is included to illustrate that the agents get an average score of +0.5 (over 100 consecutive episodes, after taking the maximum over both agents).

The submission reports the number of episodes needed to solve the environment.

A plot of rewards is included and satisfied the performance requirements of the project - Well done! 🙌

Also, report the number of episodes needed to solve the environment as required by Rubric. Just add how many episodes were needed to solve the environment.

The submission has concrete future ideas for improving the agent's performance.

Future ideas are added to improve agent's performance. Good work! Since we are interested in improving agent's performance, there few more things we can try.

More ideas

1. Use parameter space noise rather than noise on action. <https://vimeo.com/252185862>
2. Current our replay buffer is dumb. We can use prioritised experience buffer.
<https://github.com/Damcy/prioritized-experience-replay>
3. Different replay buffer for actor/critic
4. Try adding dropouts in critic network

👍 RESUBMIT

📄 DOWNLOAD PROJECT



Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

[Watch Video](#) (3:01)

RETURN TO PATH

Rate this review