UDACITY Logo

‹ Back to Deep Reinforcement Learning Nanodegree

Collaboration and Competition

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Brilliant Udacity Learner,

Congratulations! 🎉
The structure of this project is impressive and you've successfully trained your own agent to solve the environment. Going through the work, I could appreciate the time and effort put into this submission both with the code and the writeup, and it is really commendable. The accuracy obtained from your model clearly proves how excellent your implementation was. 👍

You should be proud of the work done as you seem to have a good hold on python and Deep Reinforcement Learning. Please continue with this same spirit of hard work in the projects ahead. It was my pleasure reviewing this wonderful project.

Training Code

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

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

Suggestions and Comments

You might want to consider how to best leverage Python's features to create clean, effective code:

- PEP 8 -- Style Guide for Python Code
- Python: Structuring Your Project
- Python Fundamentals Tutorial: Code Organization

The code is written in PyTorch and Python 3.

Nice work! The code is written in pyTorch and Python 3. ⭐

More In-depth Knowledge

You might want to check the following links below to understand the differences between pytorch and tensorflow:

- Tensorflow or PyTorch : The force is strong with which one?
- PyTorch vs TensorFlow—spotting the difference
- Should I go for TensorFlow or PyTorch?
- TensorFlow Vs PyTorch: Which Framework Is Better For Implementing Deep Learning Models?

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

Nice work saving the model weights of the successful agent. The submission indeed includes all the saved model weights. 👏

Suggestions and Comments

To know more about how to save model weights using pytorch, please check some resources below:

- Saving and loading a model in Pytorch
- SAVING AND LOADING MODELS
- Best way to save a trained model in PyTorch

README

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

An impressive README has been provided in this submission and it was included in the root of the GITHUB repository. This was elaborately done by including some details about the algorithm. Keep it up!👍

Suggestions and Comments

You might be interested to know more about READMEs and how to include a README.md file in the root of the Github repository:

- About READMEs
- How do I give my GitHub repository a README after the repo is already created?
- Add images to README.md on GitHub

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

Excellent job! The state and action spaces and when the environment is considered solved has been described well in the README file. 👍

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

Well done! The `README` has provided instructions for installing dependencies or downloading needed files for this project. It's good to see that you have also provided an additional instruction on training the agent using AWS. 👏

You might want to check this link to know How to write good README and why should you care, because your project's README tells a lot about your project.

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.

The `README` provided a simple instruction guidelines on how to run the project in the repository to train the agent.

Suggestions and Comments

- I would suggest you to include a more detailed steps on how to run this project in the jupyter notebook.
- An example could include some steps on how to execute the ipynb file to your jupyternotebook and making sure that all files are in the same folder. You could also include in the steps if it is necessary to change the kernel to match the `drlnd` environment by using the drop-down Kernel menu. A screenshot would be great too.

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.

Excellent! The submission includes a `REPORT.md` file in the root of the GitHub repository and it provides a description of the implementation. Learning Algorithm, Plot of Rewards, and Ideas for Future Work was also described well in this section. 👍

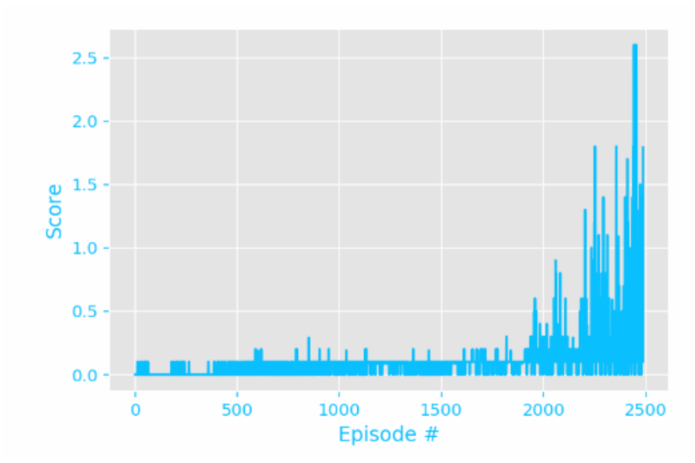
The report clearly describes the learning algorithm, along with the chosen hyperparameters. It also describes the model architectures for any neural networks.

Commendable work! The report clearly describes the learning algorithm. Details about the hyperparameters used was also provided along with the network architecture. Keep it up! 👍

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.

Good result! Plot of rewards per episode was properly implemented. The illustration below shows that the agents get an average score of +0.5 while the environment is solved on the 2487th episodes. 🎉



Suggestions and Comments

You may consider the following parameters below to achieved better results:

- Try increasing the `batch_size` to 1024.
- Try calling the method less frequently (say 5 times every 5-10 timesteps so that more data is collected)
- Try decreasing your hidden units to 128 and 64 with reLU activation on both layers.
- Try to set your Discount factor to `GAMMA = 0.99`

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

Intuitively done ! You have indeed provided a very good future ideas on how to improve the agent's performance. I would suggest you to try them out and see how well would it perform. 😃

DOWNLOAD PROJECT

RETURN TO PATH