




 Alrick11 / **DQN-and-PPO-implementation-of-cartpole** Private<> **Code** Issues Pull requests Actions Projects Security master ▾**DQN-and-PPO-implementation-of-cartpole** / **Generative-Models** /

...



Alrick11 Update README.md ...

25 minutes ago  **History**

..



README.md

25 minutes ago

README.md



README

You can find the Trajectory plots in `mushr_push_sim/TrajVis/` and tensorboard training and test plots in `mushr_push_sim/train/TensorboardVisuals/`

Setting up Conda environment and packages

Create a new conda environment and go to the home directory (`mushr_push_sim: $HOME`)

```
conda create -n my_env python=3.7
conda activate my_env
cd $HOME
pip install -r requirements.txt
```

Data Generation

Its better to avoid this step. The data is already present in `$HOME/datacol/json_files`, as you would require MuJoCo to collect data. In any case the way to create data is as follows.

```
rm -f $HOME/datacol/json_files/*
```

```
python collect_data.py --cpu_count 3
```

This command will generate train.csv.gz, test_seen.csv.gz and test_unseen.csv.gz.

Training Model and Visualization

Training is simple however, you might want to train the model on a server for training quickly (also I haven't tested the code on cpu, but it should work.). Before training you need to set the `data_dir`, `home_dir` and `traj_save_addr` in `$HOME/train/main.py`. You can set these values by passing them as arguments or change it in the code itself.

`data_dir` is the parent folder of data stored (`$HOME/datacol/json_files`).

`home_dir` is the parent folder for train (`$HOME/train/main.py`) `traj_save_addr` is the parent folder where you want to save your visual trajectories.

Now run the training code as follows:

```
cd $HOME/train/  
python main.py --data_dir '{data_dir}' --home_dir '{home_dir}' --traj_sæ
```

If you made the changes in the argument parser of the code simply run,

```
cd $HOME/train/  
python main.py
```

Visualizations

There are two visuals, the trajectories, red are predicted positions and blue the actual positions of the block stored in `{traj_save_addr}`, and tensorboard visualizations of the train and test loss plots in `$HOME/train/TensorboardVisuals/`. To run the tensorboard plots, simple run in terminal

```
cd $HOME/train/  
tensorboard --logdir=TensorboardVisuals/ --bind_all
```

To visualize the model,

```
cd $HOME/train/  
python -c "import netron; netron.start('model.onnx');"
```