Assignment 3: Reinforcement Learning Pipeline in Practice

CS260R 2023Fall: Reinforcement Learning. Department of Computer Science at University of California, Los Angeles. Course Instructor: Professor Bolei ZHOU. Assignment author: Zhenghao PENG.

> *NOTE:* ***We only grade your assignment based on the report PDF file and python files. You can do whatever you like in this notebook! Running all cells in this jupyter notebook does not generate all required figures in result.md***

(`README.md` and the intro of `assignment3.ipynb` are identical)

Welcome to the assignment 3 of our RL course!

Different from the previous assignment, this assignment requires you to finish codes mainly in python files instead of in a jupyter notebook. The provided jupyter notebook only walks you through the training and post-processing procedure based on those python files.

This assignment is highly flexible. You are free to do hyper-parameter searching or improve your implementation by introducing new components. You are encouraged to survey materials in the internet that explain the mechanisms of these classic algorithms. As a start point, Spinning Up (https://spinningup.openai.com/en/latest/) might be a good choice.

Tasks

We required you to finish the following tasks:

- Task 1: Implement TD3Trainer **(40/100 points)**
- Task 2: Implement PPOTrainer **(40/100 points)**
- Task 3: Conduct RL generalization experiments on one of the algorithms implemented **(20/100 points)**

Summary of Experiments

Group 1: Baselines

1. TD3 in Pendulum-v0
2. TD3 in MetaDrive-Tut-Hard-v0
3. PPO in CartPole-v0
4. PPO in MetaDrive-Tut-Hard-v0

Group 2: RL Generalization Experiments

Choose one of TD3, PPO or GAIL and conduct the generalization experiments (see "Training and test environments").

Deliverables

You don't need to generate the PDF file from the notebook!

Implement all files (see "File Structure") then run experiments to get figures to fill `result.md`. Then, your deliverables are:

1. The exported **PDF file** of the `result.md`: You should attach experiment result to the `result.md` and then generate a PDF file. Please visit the `result.md` for details.
2. Compress all files in the `assignment3` folder and prepare a **ZIP file**. No `.rar` or other format is allowed.

You need to submit **both the ZIP file and the PDF file** to bruinlearn. The PDF file goes to gradescope tab and the ZIP file goes to the assignment tab.

File structure

You need to pay attention to the files below:

TD3:

- `core/td3_trainer.py` - File which can be directly run to train TD3. **Please implement TD3 here.**

PPO:

- `train_ppo.py` - Train scripts for PPO in CartPole-v0 and MetaDrive. **Please implement TODO in this file.**
- `core/ppo_trainer.py` - PPO algorithm. **Please implement `compute_action` and `compute_loss`.**
- `core/buffer.py` - Supporting data structure for PPO (GAE is implemented here). **Please implement `compute_returns` for PPO.**

GAIL:

- `train_gail.py` - **Please implement TODO in this file.**
- `core/gail_trainer.py` - **Please implement GAIL here**

Common:

- `assignment3.ipynb` - Useful jupyter notebook to walk you through the whole assignment. Unlike previous work, you are not required to fill anything in this notebook (but you can use it to do whatever you like).
- `result.md` - A **template** file for your final submission. You need to attach images to it and then **generate a PDF file** that is submitted to the gradescope.
- `[train|eval]_[ppo|gail|td3].py` and `[ppo|gail|td3]_generalization_[eval|train].sh` - The reference files that can be used to debug quickly or conduct formal experiments in a batch.

Training and test environments

We prepared a set of pre-defined MetaDrive environments to train your agents.

The first is `MetaDrive-Tut-Easy-v0`, which only has one map and there is only a straight road in the map.

We also prepare a set of environments for RL generalization experiments.

Concretely, we will use a training environment to train the RL agent and use a separate test environment `MetaDrive-Tut-Test-v0` to run the RL agent. You can choose on of those training environments:

- `MetaDrive-Tut-1Env-v0`
- `MetaDrive-Tut-5Env-v0`
- `MetaDrive-Tut-10Env-v0`
- `MetaDrive-Tut-20Env-v0`
- `MetaDrive-Tut-50Env-v0`
- `MetaDrive-Tut-100Env-v0`

Those training environments contain [1, 5, 10, 20, 50, 100] unique traffic scenarios, respectively. And the test environment contains 20 unique traffic scenarios which are unique to those in the training environments.

By training a set of RL agents in different training environments and test them in the same test environment, we can examine the influence of the diversity of scenarios in the training environments.

We expect to see that when using a training environment with more diverse data, the test performance of the trained agent should be improved, showing that it has better generalization ability.

Please note that each MetaDrive can only run in single process. If you encounter any environmental error, please firstly try to restart the notebook and rerun the cell.

Notes

1. We use multi-processing to enable asynchronous data sampling. Therefore, in many places the tensors have shape `[num_steps, num_envs, num_feat]`. This means that `num_envs` environments are running concurrently and each of them samples a fragment of trajectory with length `num_steps`. There are totally `num_steps*num_envs` transitions are generated in each training iteration.

2. Each process can only have a single MetaDrive environment.

3. The jupyter notebook is used for tutorial and visualization. It is optional for you to use the notebook to train agents or visualize results.

4. By saying a generalization experiment, we expect you to train multiple agents with different training environments and plot the training-time or test-time performance of those agents in the figure.

Colab supporting

Though we use multiple files to implement algorithms, we can still leverage Colab for free computing resources.

- Step 1: Create a folder in your Google Drive root named `cs260r`
- Step 2: Upload the files in `assignment3` folder such as `train_ppo.py` and the folder `core` to the newly created `cs260r` folder in your Google Drive.
- Step 3: Run the last cell in Demo 1 (pay attention to the code we mount the colab to your google drive).