CS 175: Project in Artificial Intelligence

Winter 2020

Assignment 2

due Tuesday, February 4 2020, 11pm

In this assignment, you will install a Deep RL framework, and use it to evaluate and compare two Deep RL algorithms. The OS we use in this assignment is Linux / MacOS. If you're using Windows or another OS, please use a local or remote virtual machine.

The framework we use in this assignment is RLlib (`https://ray.readthedocs.io/en/latest/rllib.html`), although in the project you are free to use any implementation you'd like (OpenAI baselines, Spinning Up, or any of the many others out there, or even your own implementation).

1 Install dependencies

Make sure you have a recent version of Python installed, such as the latest Python 3.7; but **not Python 3.8**, as RLlib doesn't seem to support it at this time. It is recommended that you create a new `conda` environment for this assignment, please see instructions here: `https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html`.

RLlib can use any Deep Learning library. Most of the algorithms already implemented in RLlib support TensorFlow, and some support PyTorch. The following instructions will use TensorFlow for CPU, but if you intend to work with PyTorch you can install that library instead, and run experiments with `--use_pytorch`. It is also possible to install TensorFlow for GPU, and this would allow much faster execution, but is not needed in this assignment.

Install TensorFlow and some other useful dependencies:

```
pip install tensorflow psutil requests setproctitle
```

2 Install RLlib

RLlib is implemented on top of the Ray distributed execution library.

Install RLlib:

```
pip install ray[rllib]
```

3 Run experiments

Let's train policies for the CartPole environment (`https://gym.openai.com/envs/CartPole-v1/`). We'll use two Policy Gradient algorithms, one plain (PG) and one fancy (PPO).

Run the two algorithms:

```
rllib train --run=PG --env=CartPole-v1 --stop='{"timesteps_total":  200000}'
rllib train --run=PPO --env=CartPole-v1 --stop='{"timesteps_total":  200000}'
```

Note the stopping criterion we specified: after 200K timesteps of interaction with the CartPole simulator. You can specify many different criteria, or none (for training until the process is killed).

Take note of the "Result logdir" that RLlib prints after each evaluation. When running the algorithm, you can specify a different logdir, or just use the default like we did here.

4 Visualize results

TensorFlow comes with a utility for visualizing training results, called TensorBoard. If you installed PyTorch and not TensorFlow, you'll need to install TensorBoard separately now.

Run a TensorBoard web server:

```
tensorboard --logdir <the result logdir from the previous section>
```

Take note of the URL in which TensorBoard is now serving (likely `http://localhost:6006/`).

5 Evaluate results

Open a browser at the URL from the previous section. Take some time to make yourself familiar with the TensorBoard interface.

You can see the RLlib runs on the bottom left, with a color legend. If you happened to execute more runs than the two detailed in Section 3, uncheck all the other runs.

Find the plot tagged "tune/episode_reward_mean". You can find it manually, or use the "Filter tags" box at the top. Enlarge the plot using the left of 3 buttons at the bottom.

On the left you'll find some useful options. Uncheck "Ignore outliers in chart scaling" and note the effect on the plot.

Unfortunately, there's currently no good way to save the plot as an image, so just take a screenshot. This will be the first part of the submission for this assignment.

6 Analyze results

In your submission, include short answers to the following questions:

1. Which algorithm seem to perform better according to the plot you got? Why?
2. You may notice some jitter in the curves, which is likely the result of the stochasticity of the environment and of the algorithm. How could your analysis be more robust to these fluctuations and better compare the average performance of the algorithms?