

Summary: In this workshop you will gain some practical experience with deep reinforcement learning using value-based and policy-based methods. This workshop assumes that you have software dependencies installed, look at [appendix A](#) for installation instructions. First, you will train agents in a relatively simple but not trivial domain (Atari Breakout). For time constraints, work together with your peer on your left and/or right. Then, you will evaluate the performance of your agents and produce learning curves. Last, you will test pre-trained policies in the domain of SuperMarioBros.

Task 1: Train a value-based or policy-based agent

First of all, decide whether you want to train a value-based agent (e.g. DDQN) or a policy-based agent (e.g. PPO). If the latter, please boot your machine with Linux. Use either operating system otherwise. Ideally, you and your peer(s) want to choose between both types of methods (though not the same type) – to observe differences in performance on the same task.

To do this task, download the code of the workshop materials of this week from Blackboard. This is a zip file called pfrl.zip, which is an adapted version from the [PFRL library](#). Do not install this library (as suggested in the documentation) if you want to use the modified code. This will not only allow you to explore/study the code but also to extend it – if wished to do so. Instead, put the extracted folder in a path that you are aware of. If you need to install Atari support, look at the workshop of week 1.

On Windows, open a terminal and train a DDQN agent using the following commands:

- `cd $YOUR_PATH/pfrl`
- [open `./pfrl/pfrl/nn/atari_cnn.py` and `./pfrl/train_dqn_ale.py` and change the number of neurons of the MLP layers from 512 to 128, for the purpose of faster training – you should put those numbers back to 512 training+evaluation]
- `python train_dqn_ale.py --env BreakoutNoFrameskip-v4 --arch doubledqn --final-exploration-frames 300000 --steps 330000`
- rename the last folder under `$YOUR_PATH/pfrl/results` and add a run id (e.g. 20220419T092521.577582-run1) to identify the agent more easily later on

On Linux, open a terminal and train a PPO agent using the following commands:

- `cd $YOUR_PATH/pfrl/`
- [open `./train_ppo_ale.py` and change the number of neurons of the MLP layers from 512 to 128, for the purpose of faster training – after training+eval put them back to 512]
- `python train_ppo_ale.py --env BreakoutNoFrameskip-v4`
- rename the last folder under `$YOUR_PATH/pfrl/results` and add a run id (e.g. 20220419T092521.577582-run1) to identify it more easily later on

Task 2: Evaluate the performance of your trained agents

On your Windows terminal, type the following and wait for the mean average reward:

- [identify `$YOUR_PATH_TO_THE_MODEL`, which is a folder inside the folder that you renamed above. For example: 20220419T092521.577582-run1/330000_finish]
- `python train_dqn_ale.py --env BreakoutNoFrameskip-v4 --demo --load $YOUR_PATH_TO_THE_MODEL --eval-n-runs 100 --render`

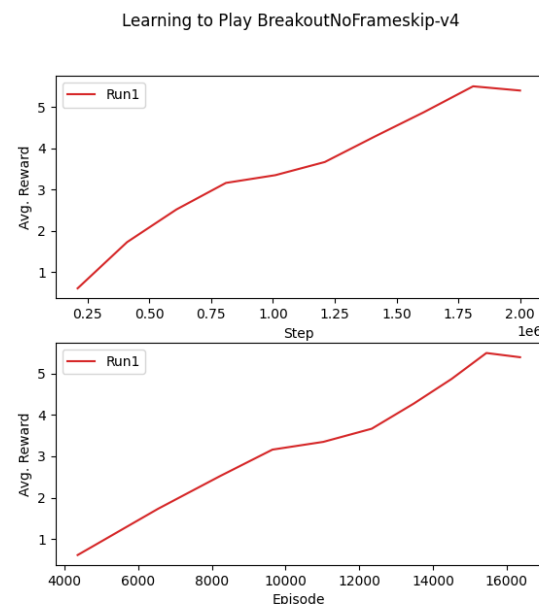
On your Linux terminal, type the following and identify the mean average reward:

- [identify \$YOUR_PATH_TO_THE_MODEL, which is a folder inside the folder that you renamed above. For example: 20220419T092521.577582-run1/330000_finish]
- `python train_ppo_ale.py --env BreakoutNoFrameskip-v4 --demo --load $YOUR_PATH_TO_THE_MODEL --eval-n-runs 100 --render`

Generate learning plots for your agents using the following commands:

- [copy the printed output during the training of your agent to a text file. For example: `./results-ppo-breakout/result-ppo-breakout- run1.txt`]
- [open the file `PlotLearningCurves.py` and change the folder to the one that hosts your logged files for each run, there should be at least one `.txt` file]
- `python PlotLearningCurves.py`

You should see something like this:



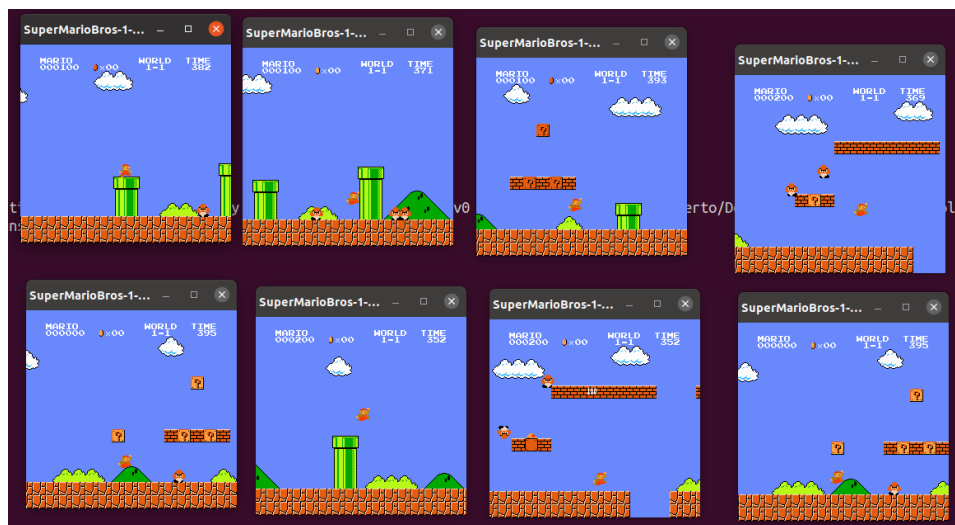
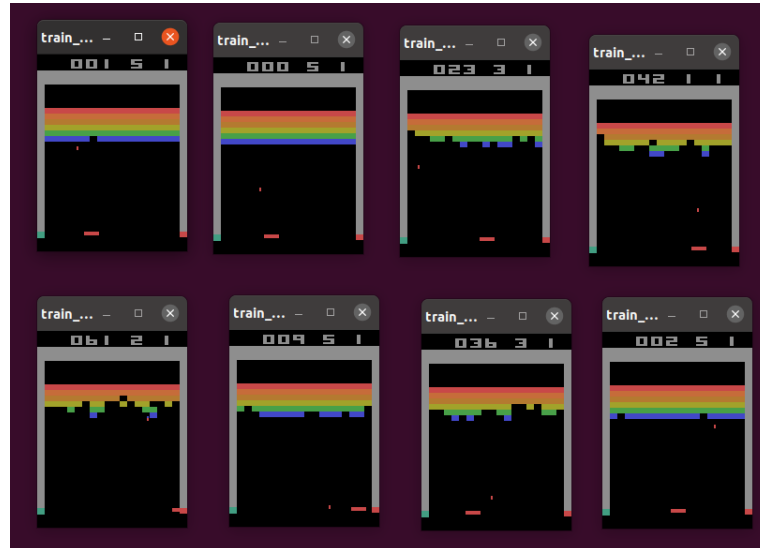
Task 3: Test pre-trained policies

To do this task you require to reboot your PC with Linux. Once you do that, download the workshop materials from Blackboard (pfrl.zip) and extract it in a path that you are aware of. Then open a terminal and run the following commands with the correct paths in your PC:

- `cd $YOUR_PATH/pfrl`
- `time python train_ppo_ale.py --env BreakoutNoFrameskip-v4 --demo --load /home/heriberto/Downloads/pfrl/results-breakout/20220419T102547.103901-run1/2010480_except --eval-n-runs 100 --render`
- `time python train_ppo_ale.py --env SuperMarioBros-1-1-v0 --demo --load /home/heriberto/Downloads/pfrl/results-ppo/20220416T153806.685398-run3/4415616_except --eval-n-runs 100 --render`

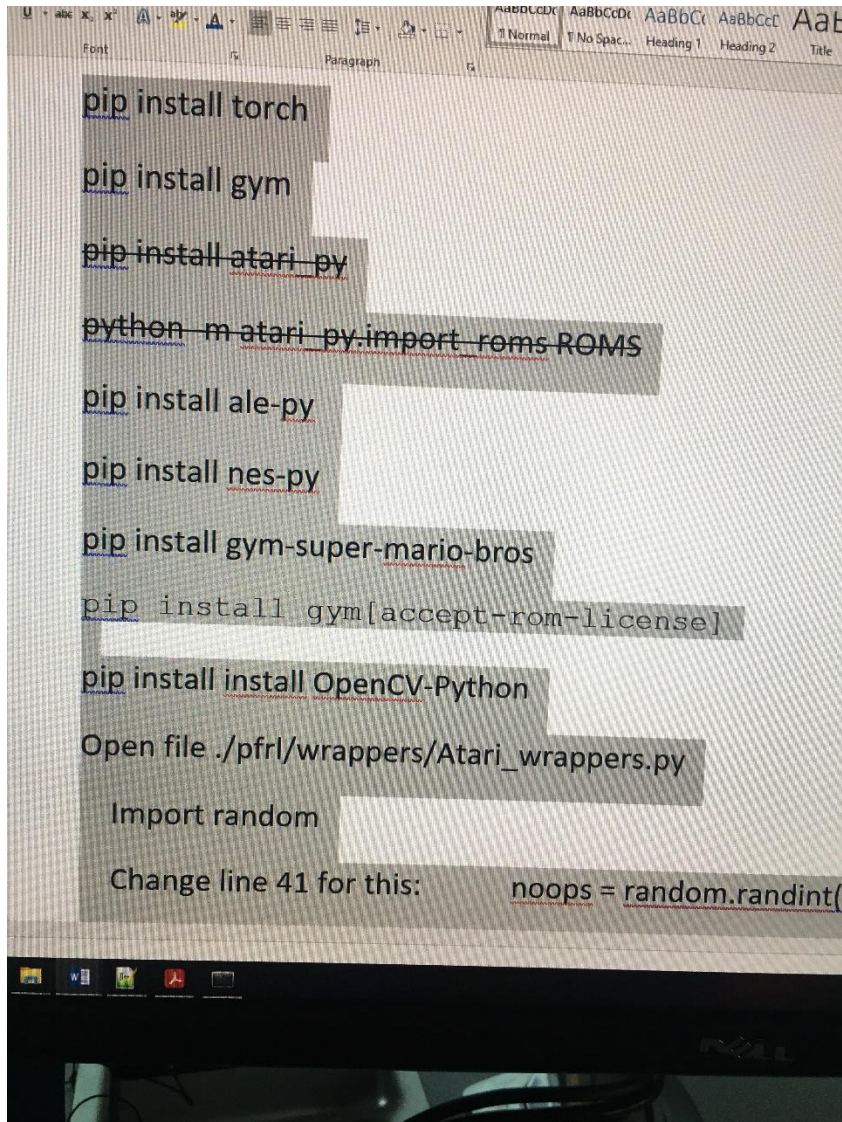
N.B. The pretrained Breakout agent was trained with MLP layers of 128 neurons and the pretrained SuperMario agent was trained with MLP layers of 512 neurons.

You should be able to see your pre-trained agents playing the game (not same layout).



In your own time, after the workshop, try to train agents with different configurations to increase your understanding in this topic.

Appendix A: Installation of Software Dependencies



```
pip install torch
pip install gym
pip install atari-py
python m_atari.py import roms ROMS
pip install ale-py
pip install nes-py
pip install gym-super-mario-bros
pip install gym[accept-rom-license]
pip install install OpenCV-Python
Open file ./pfrl/wrappers/Atari_wrappers.py
Import random
Change line 41 for this: noops = random.randint(
```