## Introduction

This homework is designed to follow up on the lecture about policy gradient algorithms. For this assignment, you will need to know about the basics of the policy gradient algorithm we talked about in class, specifically REINFORCE and PPO. If you have not already, we propose you brush up on the lecture notes.

**You are ALLOWED to discuss this homework with your classmates at the level of general solution strategies or tips. However, any work that you submit must be entirely your own, which includes your own unique code and your own unique written answers.**

**You are NOT ALLOWED to use any large language models (LLMs) or other AI-assisted tools.**

If you find errors in the homework assignment or have public questions, please post in the Campuswire “#questions-hw” channel or ask during instructor/TA office hours.

## Points

| Q1 | REINFORCE plots |  | 10 pts |  |
| --- | --- | --- | --- | --- |
| Q2 | Understanding REINFORCE |  | 5 pts |  |
| Q3 | PPO implementation & plots |  | 10 pts |  |
| Q4 | Understanding PPO vs. REINFORCE |  | 5 pts |  |
|  |  |  | **30 pts total** |  |

## Code folder

Find the folder with the provided code in the following google drive folder: <https://drive.google.com/file/d/1cEQZFOdoJPv64xW5zQweAnYA2vQsP2VF/view?usp=sharing>

Download all the files in the same directory, and run the run.py file to run your code. You will have to complete the TODOs in ppo/ppo.py to complete this homework.

Thanks to Eric Yu for this code.

## Environment

The environment we will use in this homework is built upon the Pong, Space Invaders, and Breakout environment from OpenAI gym Atari environments (<https://gym.openai.com/envs/#atari>). In this homework, we will attempt to learn these agents from state observations. We already have a working implementation of REINFORCE in the code folder, which you can run as python run.py alg=reinforce seed=42 env=<env\_name>.

**Deliverables:**

* Within PDF write-up, any written answers and plots.
* Within the compressed ZIP folder, the code files you changed (e.g. ppo.py, run.py). Please name your submission ZIP folder as *<net\_ID>\_assignment4.zip*

## Questions

1. In the code folder, you will find already available code for running REINFORCE. Run this code on the following environments: Pendulum-v1, BipedalWalker-v3, and LunarLanderContinuous-v2. It is okay if REINFORCE does not perform as well in these environments. Generate the plot over training times for these 3 environments over three different seeds, and create three plots that show the average performance of REINFORCE on each environment.
2. Based on your plots, why do you think REINFORCE suffers in these environments?
3. Now, complete the PPO code found in ppo/ppo.py. You will find a few different TODOs for you. Follow the original PPO pseudocode if you need to. Once again, use the previous three environments and three different seeds to plot your training rewards. **Clearly show the comparison between REINFORCE and PPO in your plots.**  
     
   Your expected mean performance should be AT LEAST:  
   Pendulum: -400  
   BipedalWalker: 125  
   LunarLanderContinuous: 100  
     
   Submit your writeup, along with your ppo.py file as your submission. If you change any of the hyperparameters, include run.py as well.
4. How does the performance of PPO and REINFORCE compare in these environments? Does one outperform the other? What factors do you think contribute to this difference?