Reinforcement Learning 2022
Practical Assignment 3: Policy-based RL

1 Introduction

In the previous assignment, you have worked with deep Q-learning, which aims to learn the values of actions in various states. Such a method falls under the umbrella of *value-based* reinforcement learning. In this assignment, in contrast, we are going to investigate the *policy-based* approach to reinforcement learning. In this approach, the policy is optimized directly, without the need to learn the values of actions in states.

The environment that we will use in this environment is Cartpole as shown in Figure 1. In this environment, there is a cart (black box) that moves along a horizontal axis. The goal is to move the cart in such a way that the pole is upright. Note that you do not have to implement this environment, as OpenAI has already implemented it in Gym (see source below figure, or click on highlighted Cartpole text).

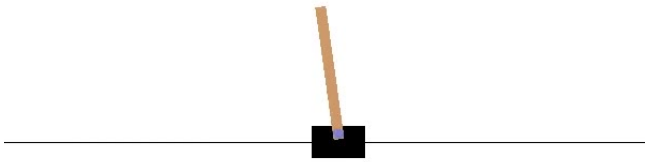

Figure 1: Illustration of the Cartpole environment. Source: [1].

You will implement the following algorithms **from scratch** (it is not allowed to use existing libraries that implement these algorithms such as Stable baselines):

- REINFORCE
- Actor-critic with:
  - Bootstrapping
  - Baseline subtraction
  - Bootstrapping + baseline subtraction

Detailed descriptions of all these techniques can be found in the lecture notes. For the implementation, you can use python with Tensorflow or Pytorch.

2 Goal

The goal of the assignment is to implement and investigate the 4 techniques. Questions that you should aim to answer are:

- Do the policy gradients used by REINFORCE indeed suffer from high-variance?
- What is the effect of bootstrapping and baseline subtraction, and their combination, within the actor-critic framework on on the variance of the policy gradients?
- How do these techniques compare in terms of performance?

As always, make sure to think carefully about the experimental design (what network architecture, what hyperparameters do you use, how to account for randomness, etc.) that you use to answer these questions.

3 Bonus (optional)

In case you are up for an additional challenge, you can consider:

- Implementing a genetic algorithm such as CMA-ES for policy optimization
- Investigate different exploration strategies such as entropy regularization

4 Submission

Make sure to nicely document everything that you do. Your final submission consists of:

- Source code with instructions (e.g., README) that allows us to **easily** (single command per experiment / sub task) rerun your experiments on a university machine booted into Linux (DSLab or computer lab).
- A self-contained scientific pdf report (using the ICML template) of at **at most 8 pages** including figures and references using the provided template. You are not allowed to deviate from this page limit or template. Every page over the limit will result in a loss of 1 point. This report contains an explanation of the techniques, your experimental design, results (performance statistics, other measurements,...), and overall conclusions, in which you briefly summarize the goal of your experiments, what you have done, and what you have observed/learned.

If you have any questions about this assignment, please visit our lab sessions where we can help you out. In case you cannot make it, you can post questions about the contents of the course on the Brightspace discussion forums, where other students can also read and reply to your questions. Personal questions (not about the content of the course!) can be sent to our email address: rl@liacs.leidenuniv.nl.

The deadline for this assignment is the 3rd of April 2022 at 23:59 CET. For each full 24 hours late, one full point will be deducted (e.g., if your work is graded with a 7, but you are two days too late, you get a 5).

The deadline for this assignment is the **8th of May 2021 at 23:59 CET**. For each full 24 hours late, one full point will be deducted (e.g., if your work is graded with a 7, but you are two days too late, you get a 5).

Good luck and have fun! :-)

5 Useful resources

For this assignment you may find the following resources useful:

- Lecture notes by Thomas Moerland

- Definitely go to OpenAIs Spinning Up, and the code repo, which is here.
- Baseline implementations of PPO, and many other algorithms are here. Don't copy code from here, only use it as a reference.