Homework 1
CSCI-5930: Machine Learning (Fall'25), Grad version.

During the 1st and part of the 2nd weeks of class you may have received an orientation to the famous "Agent-environment loop" that fuels the training of Reinforcement Learning based agents (Fig. 1).

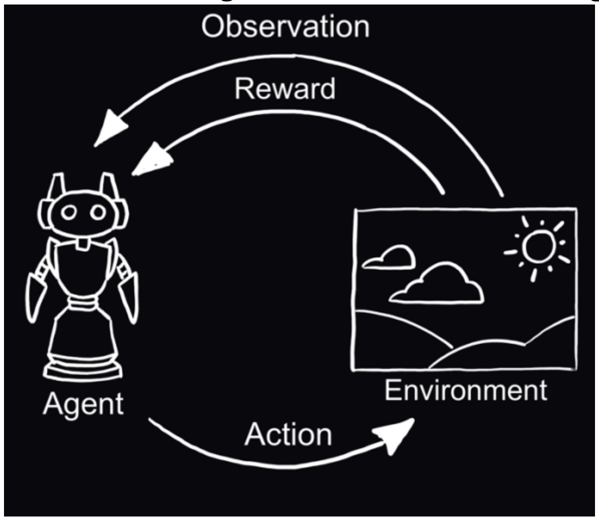

Figure 1. The agent-environment loop

More specifically, in Fig. 2 you can see that based on the current state, S_t (at time step, t), the learning agent chooses an action, A_t. It then receives a reward, R_{t+1} and new observation from the environment, S_{t+1}.

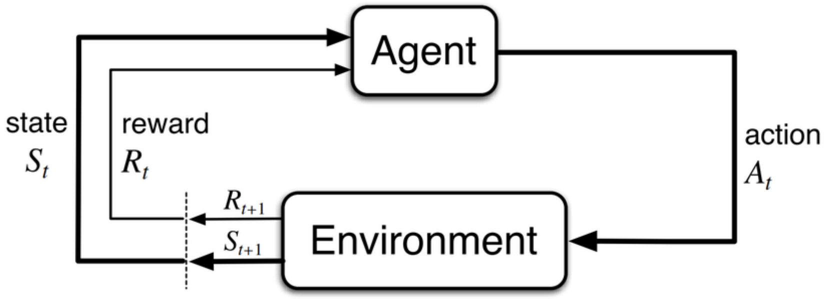

Figure 2. The agent-environment loop in terms of time references

In this homework, you are encouraged to work on the "RL-02-c--CartPole-v0-code3.py" code presented to you (also in Canvas module 1 & 2), where you will find a "CartPole-v1" agent acts based on uniform random policy (i.e., each time step agent is pushing the cart left or right randomly).

Pre-requisite

- Install Python 3.10+ in your system
- Setup workspace either in Microsoft Visual Studio Code, or PyCharm.
- Create a virtual environment, and make sure you install the "**gymnasium**" package among other necessary packages you would find in the RL-02-c--CartPole-v0-code3.py script posted in Canvas. (https://github.com/Farama-Foundation/Gymnasium?tab=readme-ov-file#installation)
- Get familiar with the "**CartPole-v1**" gymnasium environment to understand the Action Space, Rewards, etc. [https://gymnasium.farama.org/environments/classic_control/cart_pole/]

Tasks

1. (**Task1.py**): Experiment with the “RL-02-c--CartPole-v0-code3.py”: on line #17 you will see the parameter, $n = 500$, which denotes number of timesteps the code is going to stay within the “agent-environment interaction loop”. Please configure the code in a way so that it runs 100 episodes (not timesteps), and report maximum, average and standard deviation of the total rewards received per episode.
    a. For reporting, you can prepare a separate pdf file, named “**HW1-report.pdf**” containing the investigative results from Task #1.
2. (**Task2.py**): Repeat task 1 with this deterministic policy instead of random: “your agent begins with pushing the cart right, and each time-step agent toggles the movement until the pole is dropped, i.e., the episode gets ended”.
    a. Report the results in the same report file add another section to report this task #2 “HW1-report.pdf”.
3. (**Task3a.py** and **Task3b.py**): You see in the provided code, Dr. B struggled to capture the agent training into a “beautiful” gif file for the duration of 500 timesteps, which may contain multiple episodes. Huh! He did that by saving the environment in each step as an image file and later concatenate those into the gif file. However, in the “gymnasium” library, you will find a much easier way to record the same, with “**RecordVideo**” and “**RecordEpisodeStatistics**” wrapper functions. So, please revise the codes in **Task #1** and **Task #2** above to record agent during training and add the trigger to record an mp4 per episode. Please do not change the path where generated video files will be saved, i.e., use the existing “video” directory.
4. (**Task4a.py** and **Task4b.py**) Among over 900 environments, try put an agent into a different agent-environment loop (i.e., other than the CartPole-v1. Make two versions: one is driven by uniform random policy, and for other pick a deterministic policy of your choosing. Try best to intuitively pick the deterministic policy (guessing only; no training is necessary) in a way so that it’s more rewarding/longer episodes.
5. **Reporting about an application**: Please try to outline at the end of “**HW1-report.pdf”** a real-world decision task in terms of the perspective you learned in this assignment.

Deliverables

In Canvas (before deadline), submit the following:

1. Task1.py
2. Task2.py
3. Task3a.py
4. Task3b.py
5. Task4a.py
6. Task4b.py
7. HW1-report.pdf