CS 277 (W24): Control and Reinforcement Learning
Exercise 1

Due date: Monday, January 22, 2024 (Pacific Time)

Roy Fox
https://royf.org/crs/CS277/W24

In the following questions, a formal proof is not needed (unless specified otherwise). Instead, briefly explain informally the reasoning behind your answers.

Part 1 Relations between horizon settings (25 points)

In this part we will develop some intuition about the discounted return.

Question 1.1 (5 points) Suppose that, at some time t, the distribution of the state s_t is $p(s_t)$. Also suppose that the agent's policy is $\pi(a_t|s_t)$ and that the world's dynamics is $p(s_{t+1}|s_t, a_t)$. Write an expression for the *marginal* (i.e. not joint with or conditional on another variable) distribution of the successor state $p(s_{t+1})$ at time $t+1$.

Definition (Stationary distribution)**.** *A* stationary distribution $\bar{p}$ *is any state distribution such that, if the distribution of the state s_t at any time t is $\bar{p}$, then the distribution of s_{t+1} at time $t+1$ is also $\bar{p}$.*

Question 1.2 (5 points) Assume that, in a particular MDP and with a particular agent policy π, the initial distribution $\bar{p}(s_0)$ is a stationary distribution for that process. Write an expression for $\mathbb{E}_{\xi\sim p_\pi}[R^T(\xi)]$, the expected T-step finite-horizon return that only involves T, $\bar{p}$, π, and the reward function $r(s,a)$.

Question 1.3 (5 points) Again assuming a stationary distribution, write an expression for $\mathbb{E}_{\xi\sim p_\pi}[R^\gamma(\xi)]$, the expected discounted-horizon return with discount γ that only involves γ, $\bar{p}$, π, and r.

Question 1.4 (5 points) For a given discount γ, what is the "effective finite horizon" of the discounted horizon with discount γ, i.e. the finite horizon T that would gives the same expression in both previous questions?

Question 1.5 (5 points) The purpose of defining the return R is to summarize a sequence of rewards into one real number that we can maximize. Suppose that instead we select just a single one of the rewards, $r_t = r(s_t, a_t)$, by drawing the variable t geometrically at random with parameter $1-\gamma$, and then we maximize the expectation $\mathbb{E}_{t\sim \mathrm{G}(1-\gamma)}[r_t]$. Show that $\mathbb{E}[r_t] = c\,\mathbb{E}[R^\gamma]$,[1] where the constant c does not depend on p, π, or r. In this question do not assume a stationary distribution.

[1] To clarify, both these expectations are w.r.t. $\xi \sim p_\pi$, and the former is also w.r.t. $t \sim \mathrm{G}(1-\gamma)$

Part 2 Value function (25 points)

In this part we will see some very useful properties of the expected return, called the *value function*.

Question 2.1 (8 points) Show by induction that, for given dynamics p, policy π, and any $t \geq t_0 \geq 0$, the conditional distribution $p(s_t|s_{t_0})$ of s_t given s_{t_0} depends only on $t - t_0$ (and not on t_0). Hint: the induction step involves marginalization, somewhat like Question 1.1.

Question 2.2 (8 points) In the discounted horizon, the future return from time $t_0 \geq 0$ is defined as

$$R^{\gamma}_{\geq t_0} = \sum_{t \geq t_0} \gamma^{t-t_0} r(s_t, a_t).$$

For any time step $t_0 \geq 0$, we define the value function to be the expected future return from time t_0, given the state s_{t_0} at that time:

$$V^{\pi,\gamma}_{t_0}(s) = \mathbb{E}_{\xi \sim p_\pi}[R^{\gamma}_{\geq t_0} | s_{t_0} = s]. \tag{1}$$

Using the claim in Question 2.1, show that this value function is time-invariant, i.e. it doesn't depend on t_0 after all.

Question 2.3 (9 points) In the T-step finite horizon, the future return from time $t_0 \geq 0$ is defined as

$$R^{T}_{\geq t_0} = \sum_{t=t_0}^{T-1} r(s_t, a_t).$$

With $V^{\pi,T}_{t_0}$ now defined as in (1) but for $R^{T}_{\geq t_0}$, show a very simple MDP and policy, such that in the 2-step finite horizon $V^{\pi,T=2}_0 \neq V^{\pi,T=2}_1$. In other words, give a counterexample that shows that what Question 2.2 claims for the discounted horizon doesn't hold in the finite horizon.

Part 3 Behavior Cloning (50 points)

In this part, you will install a Deep RL framework, RLlib, and use it to evaluate the Behavior Cloning algorithm. Useful scripts are provided here: `https://royf.org/crs/CS277/W24/CS277E1.zip`. Except for the questions that ask you to report results, no answer is needed, just to successfully follow the instructions.

Setup

Question 3.1 (5 points) Follow the HPC3 Setup and Usage Guide to learn how to use HPC3, UCI's compute cluster.

Question 3.2 (5 points) In the conda environment you create for this course, install the latest RLlib and its supporting Deep Learning frameworks, TensorFlow and PyTorch (only one of these is needed).

```
pip install "ray[rllib]==2.9.0" tensorflow torch
```

We will also need a library called Gymnasium (gym) that provides an API for RL environments.

```
pip install gymnasium
```

Reinforcement Learning

Question 3.3 (5 points) Use RLlib to train an agent to perform the Cart Pole task, a gym environment called `CartPole-v1`. The algorithm we will use is called PPO (we will learn about it in a later lecture). Run the algorithm for 1000 "iterations" (what iteration means in RLlib is algorithm-dependent) and create a checkpoint (a save of the agent's trained parameters) every 10 iterations.

```
rllib train
  --run PPO
  --env CartPole-v1
  --checkpoint-freq 10
  --stop '{"training_iteration": 1000}'
```

Report the mean and standard deviation of the returns of 5 episodes.

Question 3.4 (5 points) Roll out the agent that you trained to see how well it performed. The agent checkpoints are by default in a path named

~/ray_results/default/PPO_CartPole-v1_<experiment_id>,

where `experiment_id` is an automatically assigned identifier of the experiment you ran in the previous step, ending with the date and time. Using the provided evaluation script (adapted from RLlib code), roll out for 5000 steps of agent–environment interaction, which should be about 10 episodes.

```
python evaluate.py
  ~/ray_results/default/PPO_CartPole-v1_<experiment_id>/checkpoint_000099/
  --run PPO
  --env CartPole-v1
  --steps 5000
```

You can also roll out an earlier checkpoint to see the difference in performance. The script will output the return of each episode, which should be 500 for successful episodes. If you are not seeing consistently good episodes, rerun the training in the previous question.

Imitation Learning

Question 3.5 (5 points) Use the trained agent to generate demonstrations for an imitation learning agent. Roll out for 250000 steps. The demonstrations will be saved in a file named `rollouts.pkl`.

```
python evaluate.py
  ~/ray_results/default/PPO_CartPole-v1_<experiment_id>/checkpoint_000099/
  --run PPO
  --env CartPole-v1
  --steps 250000
  --out rollouts.pkl
```

Question 3.6 (5 points) Using the provided conversion script, convert the demonstrations into the format used by RLlib for training from offline data. The results will be saved in a directory named `CartPole-v1`.

Question 3.7 (5 points) Train a Behavior Cloning agent. The agent will be trained on input from the data generated in the previous step, and evaluated in new simulations of the environment.

```
rllib train
  --run BC
  --env CartPole-v1
  --checkpoint-freq 1000
  --stop '{"training_iteration": 10000}'
  --config='{"offline_data": "./CartPole-v1/output-<evaluation_id>.json"}'
```

Question 3.8 (5 points) Repeat Question 3.4 for the trained BC agent, and **report the mean and standard deviation of the returns of 5 episodes.**

Question 3.9 (10 points) The returns reported in the previous question are likely below the 500 achievable by an optimal policy. Considering some of the things that can go wrong in Behavior Cloning (see the Lecture 2, slide 28), which may have gone wrong here? Briefly justify your intuition.