Assignment 3

Due Date 11:59pm, November 1, 2023.

Submission Instructions Please submit your response in two files via Canvas:

- a `pdf` file that includes your answers to all the problems, and
- a `zip` file that includes your completed `Jupyter` notebooks.

The completed `Jupyter` notebooks should include missing code as well as the code output.

Accompanying Code The assignment includes an accompanying `zip` file with starter code for the programming problems. To execute and complete this starter code you will need Python 3 and `Jupyter`. You can use either your local Python 3 run-time or the one provided by Google Colaboratory (available using your Rice NetID). If you have any questions about working with the code, please contact the teaching staff.

Collaboration Policy Collaborating on assignments is permitted, provided the submission lists the students who you collaborated with. As per the Rice Honor System Handbook, this means that students are allowed to develop answers to specific problems together and check answers with each other. However, collaboration does not confer the right for students to submit the exact same document— students must write down the answers themselves. While the "core" of the answer can be the same, the wording cannot be identical unless precise wording is necessary to answer the question. Students must be able to demonstrate that they worked together when developing responses and that one student did not copy off the other. This means that students are barred from dividing questions among themselves.

Citation Policy Students don't have an obligation to cite class slides, lectures, or class textbooks on assignments; these are considered common knowledge. An academic citation style is required for all other sources (such as research articles and blogs).

Late Policy Assignments should be turned on time via Canvas. Assignments handed in late will be marked off 10% per day. Assignments more than 3 days late will not be accepted. In turn, you can expect the teaching staff to grade your assignments and provide feedback in a timely manner.

Grading The assignment is worth 20% of your final grade.

Updates to the Assignment In case there are any updates to the assignment (e.g., additional clarifications, typo fixes, hints, etc.), they will be indicated via the following table.

Version	Date	Note
v1	October 18	Assignment released

1. Q Learning with Function Approximation

Starter code for this problem is provided in the accompanying `Jupyter` notebook. For each part, include the resulting plots with 2-4 sentence summary of these plots in your `pdf` submission.

1.1 30pts

Implement the general recipe for Q Learning algorithm with function approximation. Use this recipe to learn the optimal policy for the Acrobot environment from Gymnasium.

1.2 30pts

Conduct an ablation study to characterize the utility of *replay buffers* and *target networks* on the performance of the Q Learning algorithm. In particular implement the following three version of the algorithms:

(a) Q learning with neither target networks nor replay buffers,
(b) Q learning with replay buffers but without target networks,
(c) Q learning with target networks but without replay buffers, and

compare their performance with the general recipe of Problem 1.1.

Hint: All variants are special cases of the general recipe.

1.3 15pts

Select any one hyperparameter of the algorithm and study its effect on the agent's performance. You should try at least five different values of this hyperparameter and report on the observed performance.

1.4 10pts

While applying RL in real world, you may consider the use of off-the-shelf implementation of RL algorithms. The package Stable Baselines 3 aims to provide a set of reliable implementations of RL algorithms in PyTorch.

Familiarize yourself with this package and use its implementation of Deep Q Network to learn the optimal policy for the Acrobot environment.

2. Policy Gradient Theorem

2.1 3pts

Describe a class of reinforcement learning problems in which you cannot use the general recipe of Deep Q Learning implemented in Problem 1 but can use policy gradient methods.

2.2 12pts

Consider an Markov decision process (MDP) with

- a designated initial state s_0,
- discount factor γ, and
- policy π parameterized by θ, i.e., $\pi \equiv \pi(a|s;\theta)$.

For this MDP, we define $d_\pi(s)$ as the discounted weighting of encountering state s after starting at s_0 and then following the policy π:

$$d_\pi(s) \equiv \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|s_0, \pi)$$

Show that, in this case, the policy gradient corresponds to

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta v_\pi(s_0) \\ &\propto \sum_s d_\pi(s) \sum_a q_\pi(s,a) \nabla_\theta \pi(a|s).\end{aligned}$$