Robot Learning	Assignment 3
CS 4756 Spring 2024	Due 11:59pm Thursday, March 14

(1) Policy Gradients (10 points)

Recap: Recall that the goal of RL is to learn some θ^* that maximizes the objective function:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)}\big[r(\tau)\big] \tag{1}$$

where each τ is a rollout of length T and $r(\tau) = \sum_{t=0}^{T-1} r(s_t, a_t)$ is the reward for that rollout. $\pi_\theta(\tau)$ is the probability of the rollout under policy π_θ, i.e. $\pi_\theta(\tau) = \Pr[s_0]\pi_\theta(a_0|s_0)\prod_{t=1}^{T-1}\Pr[s_t|s_{t-1}, a_{t-1}]\pi_\theta(a_t|s_t)$.

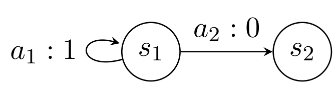
The policy gradient approach requires that we take the gradient of this objective as follows:

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \int \pi_\theta(\tau) r(\tau)\ d\tau = \int \pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau) r(\tau)\ d\tau & (2) \\ &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\big[\nabla_\theta \log \pi_\theta(\tau) r(\tau)\big] & (3) \end{aligned}$$

The gradient can further be refined by noting that future actions do not affect past rewards, resulting in the following "reward-to-go" formulation:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_{t=0}^{T}\left(\nabla_\theta \log \pi_\theta(a_t|s_t) \cdot \sum_{t'=t}^{T} r(s_{t'}, a_{t'})\right)\right] \tag{4}$$

In this question, we consider a toy MDP and get familiar with computing policy gradients.



Consider the following infinite-horizon MDP. The initial state is always s_1, and the episode terminates when s_2 is reached. The agent receives reward 1 for taking action a_1 and reward 0 for taking action a_2. In this case, we can define the policy with a single parameter θ:

$$\pi_\theta(a_1|s_1) = \theta, \quad \pi_\theta(a_2|s_1) = 1 - \theta$$

(a) Use policy gradients to compute the gradient of the expected return of π_θ with respect to the parameter θ (Eq. 3). Do not use discounting.
You may find this fact useful:

$$\sum_{k=1}^{\infty} k\alpha^{k-1} = \frac{d}{d\alpha}\sum_{k=1}^{\infty}\alpha^k$$

(b) Compute the expected return of the policy π_θ directly (Eq. 1). Compute the gradient of this expression and verify that it matches your result in **(a)**.

(c) Reward-to-go can be helpful and improve the statistical qualities of our policy gradient. Apply reward-to-go as an advantage estimator. Write the new policy gradient (Eq. 4), and verify that it is unbiased.

(2) Bounding Error in Approximate Policy Iteration (10 points)

In this problem, we look at how errors in approximating the value function affect the performance of a policy during policy iteration.

Let's assume we are in an infinite horizon MDP with discount factor γ. We have a reference policy π whose true value function is $V^\pi(s)$. We collect rollouts with π, and fit a neural network to approximate this value function, where $\hat{V}(s) \approx V^\pi(s)$.

Let's assume we did a really good job and can guarantee that the error from the fit is at most ϵ. More formally, let $||V^\pi - \hat{V}||_\infty \leq \epsilon$.[1]

We now choose a greedy policy to improve upon policy $\hat{\pi}$:

$$\hat{\pi}(s) = \underset{a}{\operatorname{argmax}} \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a)\hat{V}(s') \right]$$

Note that this is exactly the policy improvement step, except the value function is substituted with our approximate value function. We want to know how the greedy policy $\hat{\pi}(s)$ performs with respect to $\pi(s)$.

Let $V^{\hat{\pi}}(s)$ be the value of the greedy policy $\hat{\pi}(s)$. Prove the following:

$$|V^\pi(s) - V^{\hat{\pi}}(s)| \leq \frac{2\gamma\epsilon}{1-\gamma} \quad \text{, for all } s$$

In other words, $\hat{\pi}$ can end up doing much worse than π. Additionally, even though the error from fitting the value was ϵ, the performance error scales up by a factor of $\frac{1}{1-\gamma}$.

Hint: One way to approach the question would be:

1. For any policy we have,

$$V^\pi(s) = R(s,\pi(s)) + \gamma \sum_{s'} P(s'|s,\pi(s))V^\pi(s')$$

   Use this substitution to expand $V^\pi(s) - V^{\hat{\pi}}(s)$.

2. Next, note that you need to establish a relationship between $\pi(s)$ and $\hat{\pi}(s)$. Exploit the following observation: $\hat{\pi}(s) = \operatorname{argmax}_a f(s,a)$ must imply $f(s,\hat{\pi}(s)) \geq f(s,\pi(s))$ for any policy π.

3. Use these facts to obtain the following intermediate result. For any s,

$$V^\pi(s) - V^{\hat{\pi}}(s) \leq 2\gamma\epsilon + \gamma \sum_{s' \in \mathcal{S}} \Pr[s'|s,\hat{\pi}(s)](V^\pi(s') - V^{\hat{\pi}}(s'))$$

   From the above, you should be able to prove the final result for all s.

[1]Note: $||x||_\infty$ is the L-infinity norm

(3) Extra Credit (Mandatory for 5756): Reward Shaping with an Approximate Value Function (5 points)

Previously, we saw how acting greedily with an approximate value function can result in a *worse* policy. Instead, what if we use the approximate value function to *shape the reward*?

Let's define a *reward bonus* using the approximate value function from Q2.

$$F(s, s') = \gamma \hat{V}(s') - \hat{V}(s)$$

This extra reward is gained whenever we transition from state s to state s'. Informally, we are giving a small intermediate reward for moving toward states of higher value. Adding these intermediate rewards helps in speeding up a policy's convergence in environments with sparse rewards.
At each step i of an episode, the shaped reward R_i is then defined as

$$R_i = r_i + F(s_i, s_{i+1})$$

where r_i is the base reward $r(s_i, a_i)$ received for step i. We continue to use a *discount factor* of γ in computing total reward.

(a) Consider a given episode of potentially infinite-length, of visited states $s_0, s_1, ...$
Write out the total reward received in the shaped environment, expressed in terms of the total reward that would have been accrued in the unshaped environment. What is noticeable about this relationship?

(b) Show that the performance of the optimal policy $\hat{\pi}$ computed with the shaped rewards is the same as the performance of the optimal policy π^* without reward shaping. Formally, prove that:

$$||V^{\pi^*} - V^{\hat{\pi}}||_\infty = 0$$