[SOLUTION TEMPLATE] Assignment 2: Policy Gradients

Due September 25, 11:59 pm

4 Policy Gradients

- Create two graphs:
  - In the first graph, compare the learning curves (average return vs. number of environment steps) for the experiments prefixed with `cartpole`. (The small batch experiments.)
  - In the second graph, compare the learning curves for the experiments prefixed with `cartpole_lb`. (The large batch experiments.)

1. The learning curves for `cartpole_lb` small batch experments are as following:

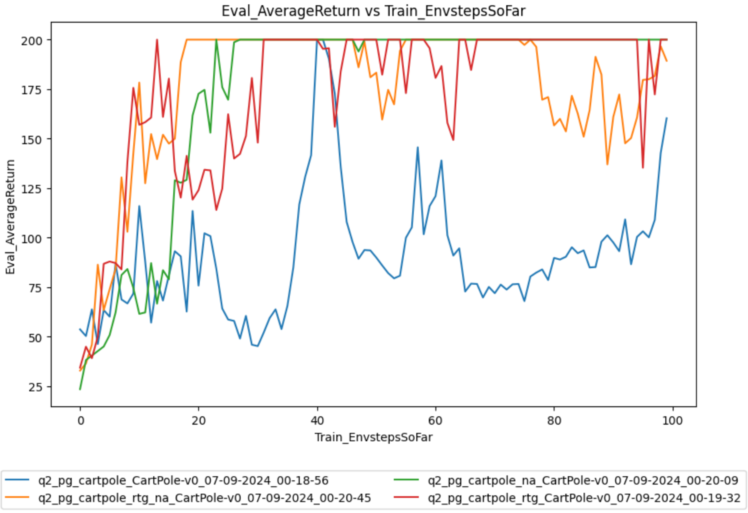


Figure 1: Learning Curve : CartPole-small

2. The learning curves for `cartpole_lb` large batch experments are as following:

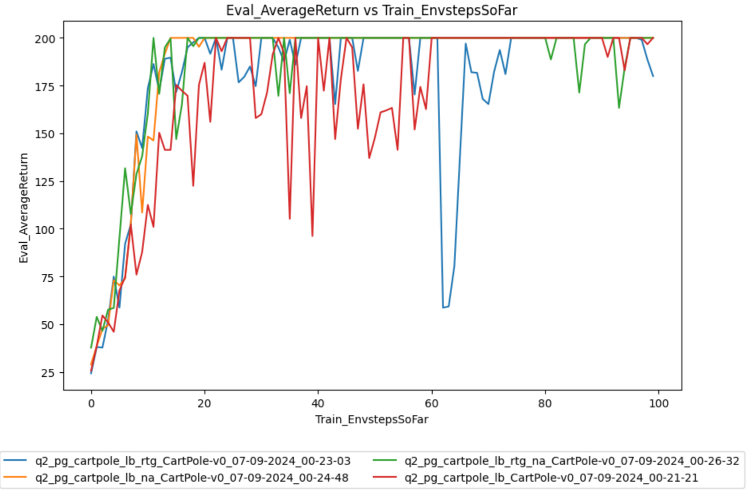


Figure 2: Learning Curve : CartPole-large

For all plots in this assignment, the x-axis should be number of environment steps, logged as `Train_EnvstepsSoFar` (*not* number of policy gradient iterations).

- Answer the following questions briefly:
  - Which value estimator has better performance without advantage normalization: the trajectory-centric one, or the one using reward-to-go?

    Solution: *The Reward-to-Go value estimator performs better without normalization.*
  - Did advantage normalization help?

    Solution: *Yes, normalization process helped. For trajectory-centric one, normalized one learned faster and reduced variance. For reward-to-go, it improves learning speed.*
  - Did the batch size make an impact?

    Solution: *Yes, larger batch size generally better, while normalized reward-to-go performed roughly same.*
- Provide the exact command line configurations (or `#@params` settings in Colab) you used to run your experiments, including any parameters changed from their defaults.

5 Neural Network Baseline

- Plot a learning curve for the baseline loss.

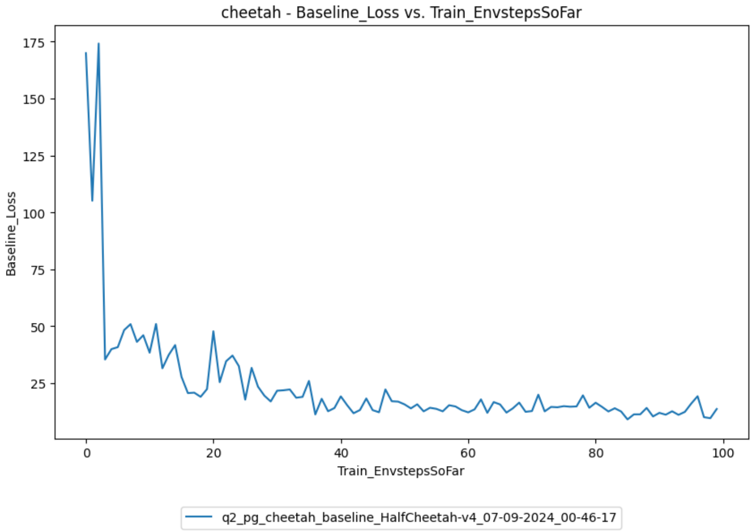
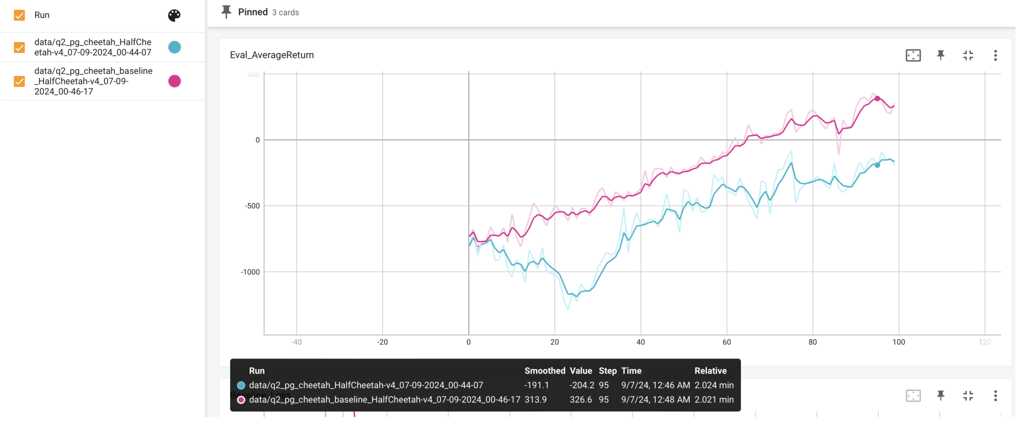


Figure 3: Learning Curve : HalfCheetah-Baseline loss

- Plot a learning curve for the eval return. You should expect to achieve an average return over 300 for the baseline version.



☑ Run
☑ data/q2_pg_cheetah_HalfCheetah-v4_07-09-2024_00-44-07 ●
☑ data/q2_pg_cheetah_baseline_HalfCheetah-v4_07-09-2024_00-46-17 ●

Run	Smoothed	Value	Step	Time	Relative
● data/q2_pg_cheetah_HalfCheetah-v4_07-09-2024_00-44-07	-191.1	-204.2	95	9/7/24, 12:46 AM	2.024 min
○ data/q2_pg_cheetah_baseline_HalfCheetah-v4_07-09-2024_00-46-17	313.9	326.6	95	9/7/24, 12:48 AM	2.021 min

Figure 4: Learning Curve : HalfCheetah-Learning Curve

- Run another experiment with a decreased number of baseline gradient steps (`-bgs`) and/or baseline learning rate (`-blr`). How does this affect (a) the baseline learning curve and (b) the performance of the policy?

  Solution:

  - *The baseline loss in both decreased number of baseline gradient steps and baseline learning rate is performs higher loss, but losses converge to roughly same level to non-decreased one after around* 40 *steps.*
  - *While decreased number of baseline gradient steps and baseline learning rate returned poorer performance in learning curve.*

- **Optional:** Add `-na` back to see how much it improves things. Also, set `video_log_freq 10`, then open TensorBoard and go to the "Images" tab to see some videos of your HalfCheetah walking along!

  Solution:

- *The normalization helps with improving learning performance, while baseline loss keeps at the same level while becomes more stable (i.e. smoother).*

6 Generalized Advantage Estimation

- Provide a single plot with the learning curves for the `LunarLander-v2` experiments that you tried. Describe in words how λ affected task performance. The run with the best performance should achieve an average score close to 200 (180+).

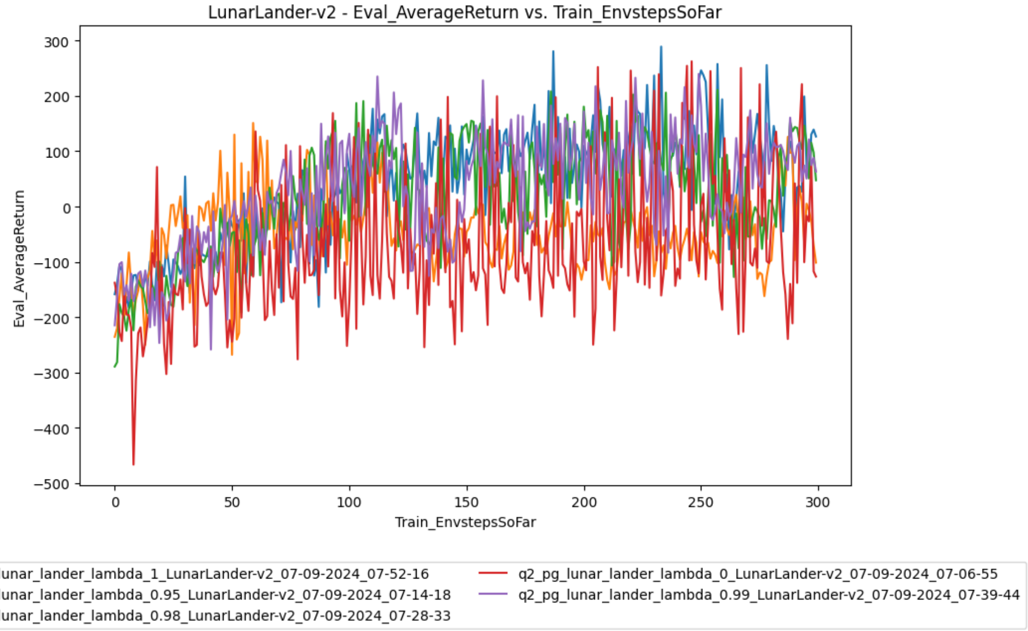


Figure 5: LunarLander: Learning Curve

- Consider the parameter λ. What does $\lambda = 0$ correspond to? What about $\lambda = 1$? Relate this to the task performance in `LunarLander-v2` in one or two sentences.
  - When $\lambda = 0$, GAE becomes $A^{\pi}_{GAE}(s_t, a_t) = \delta_t(s_t, a_t) = r(s_t, a_t) + \gamma V^{\pi}_{\phi}(s_{t+1}) - V^{\pi}_{\phi}(s_t)$, which is equivalent of single-step advantage estimator, with low-variance and high-bias.
  - When $\lambda = 1$, GAE becomes $A^{\pi}_{GAE}(s_t, a_t) = \sum_{t'=t}^{T-1} \gamma^{t'-t} \delta_{t'}$, which is the multi-step actor critic method, with high variance and low bias.

7 Hyperparameter Tuning

1. Provide a set of hyperparameters that achieve high return on `InvertedPendulum-v4` in as few environment steps as possible.
2. Show learning curves for the average returns with your hyperparameters and with the default settings, with environment steps on the x-axis. Returns should be averaged over 5 seeds.

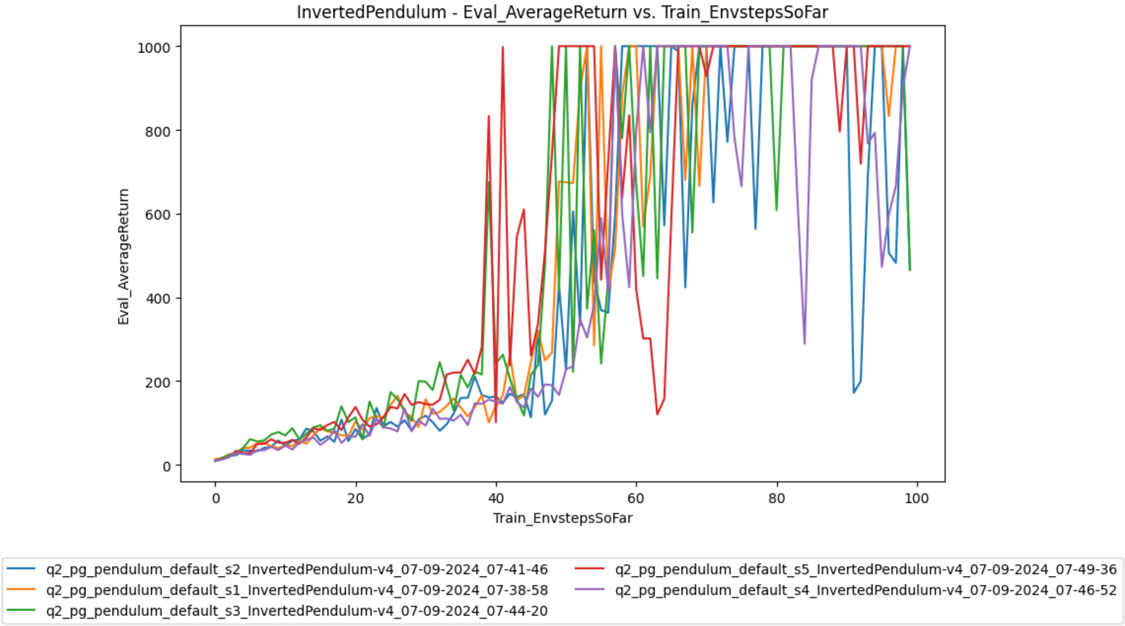


Figure 6: InvertedPendulum: Learning Curve

8 (Extra Credit) Humanoid

1. Plot a learning curve for the Humanoid-v4 environment. You should expect to achieve an average return of at least 600 by the end of training. Discuss what changes, if any, you made to complete this problem (for example: optimizations to the original code, hyperparameter changes, algorithmic changes).

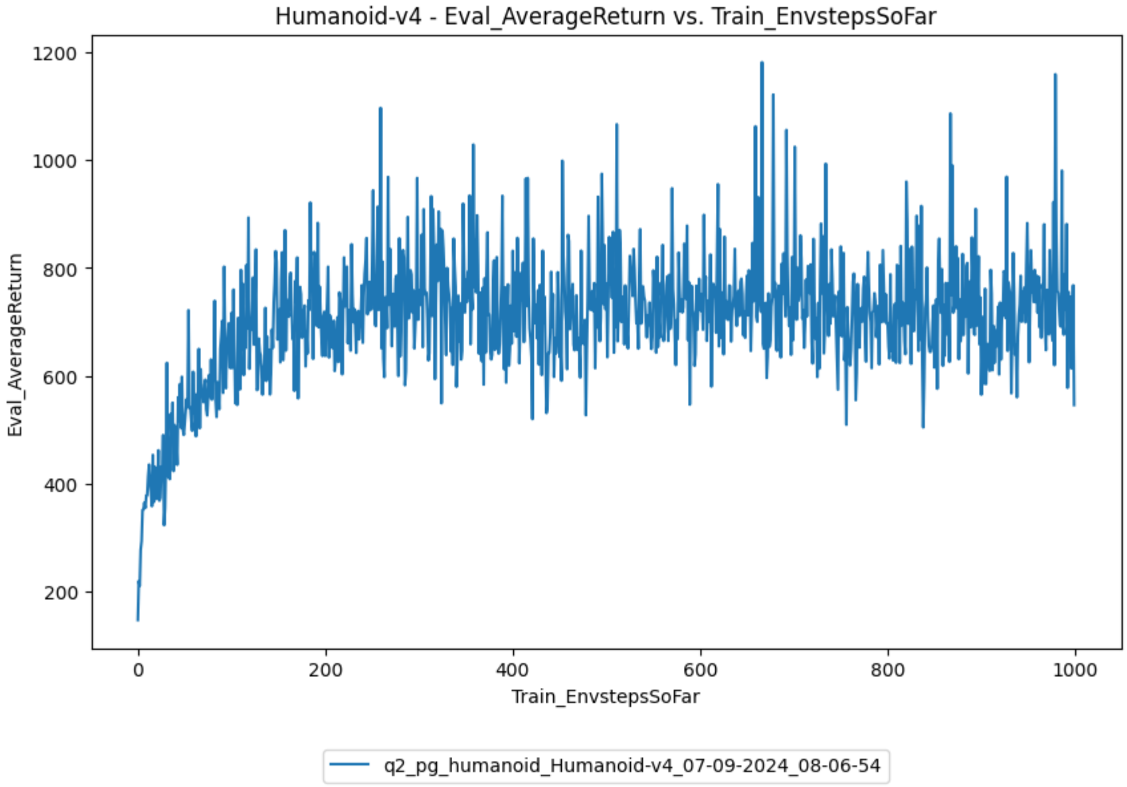


Figure 7: Humanoid: Learning Curve

9 Analysis

Consider the following infinite-horizon MDP:

$$a_1 \circlearrowright s_1 \xrightarrow{a_2} s_F$$

At each step, the agent stays in state s_1 and receives reward 1 if it takes action a_1, and receives reward 0 and terminates the episode otherwise. Parametrize the policy as stationary (not dependent on time) with a single parameter:

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

1. Applying policy gradients

   (a) Use policy gradients to compute the gradient of the expected return $R(\tau)$ with respect to the parameter θ. **Do not use discounting.**

   Hint: to compute $\sum_{k=1}^{\infty} k\alpha^{k-1}$, you can write:

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

   Solution: In this setting, each episode can continue indefinitely (since we are not using discounting), but at each step, there's a chance the agent terminates the episode. Due to π_θ is parameterized by θ, and $\pi_\theta(a_1|s_1) = \theta$, $\pi_\theta(a_2|s_1) = 1 - \theta$, then:

   $$\begin{aligned} J(\theta) &= \sum_{k=1}^{\infty} \Pr(\text{survive for k steps}) \times 1 \\ &= \sum_{k=1}^{\infty} \theta^k \\ &= \frac{\theta}{1-\theta} \end{aligned}$$

   Take the gradient with respect to θ:

   $$\nabla_\theta J(\theta) = \frac{1}{(1-\theta)^2}$$

   (b) Compute the expected return of the policy $\mathbb{E}_{\tau\sim\pi_\theta} R(\tau)$ directly. Compute the gradient of this expression with respect to θ and verify that this matches the policy gradient.

   Solution:

   $$\begin{aligned} \mathbb{E}_{\tau\sim\pi_\theta} R(\tau) &= \sum_{k=1}^{\infty} (\pi_\theta(a_1|s_1) \times 1 + \pi_\theta(a_2|s_1) \times 0) \\ &= \sum_{k=1}^{\infty} (\theta \times 1) \\ &= \frac{\theta}{1-\theta} \\ \nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta} R(\tau) &= \frac{1}{(1-\theta)^2} \\ &= \nabla_\theta J(\theta) \end{aligned}$$

2. Compute the variance of the policy gradient in closed form and describe the properties of the variance with respect to θ. For what value(s) of θ is variance minimal? Maximal? (Once you have an exact expression for the variance you can eyeball the min/max).

   Hint: Once you have it expressed as a sum of terms $P(\theta)/Q(\theta)$ where P and Q are polynomials, you can use a symbolic computing program (Mathematica, SymPy, etc) to simplify to a single rational expression.

   Solution:

   To compute the variance of the policy gradient in closed form, we first need to express the policy gradient and its associated variance mathematically.

   Step 1: Recall the policy gradient

   The expected return is given by:

$$J(\theta) = \frac{\theta}{1-\theta}$$

   The policy gradient is:

$$\nabla_\theta J(\theta) = \frac{1}{(1-\theta)^2}$$

   Step 2: Variance of the policy gradient

   The variance of the policy gradient refers to the variability in the gradient estimates over different episodes. For simplicity, let's compute the variance of a simple case where each episode either continues with probability θ or terminates with probability $1-\theta$.

   The reward R in each episode is the number of steps the agent survives (each survival step gives a reward of 1). The probability of surviving for k steps and then terminating is:

$$\Pr(\text{survive } k \text{ steps}) = (1-\theta)\theta^{k-1}$$

   Thus, the expected return is the sum over all possible episode lengths:

$$J(\theta) = \sum_{k=1}^{\infty} k(1-\theta)\theta^{k-1} = \frac{\theta}{(1-\theta)^2}$$

   To compute the variance, we need the second moment of the return, $\mathbb{E}[R^2]$, where R is the episode length.

   The probability of surviving for k steps is $(1-\theta)\theta^{k-1}$, so the second moment is:

$$\mathbb{E}[R^2] = \sum_{k=1}^{\infty} k^2(1-\theta)\theta^{k-1}$$

   This can be computed using a known formula for sums of this type:

$$\sum_{k=1}^{\infty} k^2\alpha^k = \frac{\alpha(1+\alpha)}{(1-\alpha)^3}, \quad \text{for } |\alpha| < 1$$

   Using $\alpha = \theta$, the second moment is:

$$\mathbb{E}[R^2] = \frac{\theta(1+\theta)}{(1-\theta)^3}$$

Step 3: Compute the variance

The variance of R, $\text{Var}(R)$, is given by:

$$\text{Var}(R) = \mathbb{E}[R^2] - (\mathbb{E}[R])^2$$

Substituting the expressions for $\mathbb{E}[R^2]$ and $\mathbb{E}[R]$:

$$\text{Var}(R) = \frac{\theta(1+\theta)}{(1-\theta)^3} - \left(\frac{\theta}{(1-\theta)^2}\right)^2$$

Simplifying the right-hand side:

$$\text{Var}(R) = \frac{\theta(1+\theta)}{(1-\theta)^3} - \frac{\theta^2}{(1-\theta)^4}$$

Step 4: Find the minimum and maximum of the variance

To find the values of θ where the variance is minimized and maximized, we analyze the expression:

$$\text{Var}(R) = \frac{\theta(1+\theta)}{(1-\theta)^3} - \frac{\theta^2}{(1-\theta)^4}$$

1.At $\theta = 0$:

$$\text{Var}(R) = 0$$

This makes sense because if $\theta = 0$, the agent always terminates after 1 step, so there is no variance in the returns.

2. At $\theta = 1$:

$$\text{Var}(R) = \infty$$

This also makes sense because if $\theta = 1$, the agent never terminates, and the episode length could become infinitely large, resulting in infinite variance.

3. For intermediate values of θ: The variance increases as θ increases from 0 to 1, and the variance becomes maximal near $\theta = 1$.

Conclusion:

- The minimal variance occurs at $\theta = 0$, where the policy deterministically terminates the episode after 1 step. - The maximal variance occurs as θ approaches 1, where the agent almost never terminates, leading to highly variable episode lengths.

3. Apply return-to-go as an advantage estimator.

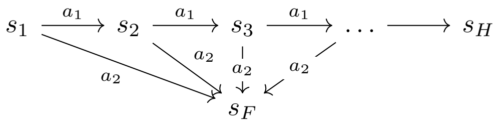
   (a) Write the modified policy gradient and confirm that it is unbiased.

   Solution:

   (b) Compute the variance of the return-to-go policy gradient and plot it on $[0, 1]$ alongside the variance of the original estimator.

   Solution:

4. Consider a finite-horizon H-step MDP with sparse reward:



The agent receives reward $R_{\max}$ if it arrives at s_H and reward 0 if it arrives at s_F (a terminal state). In other words, the return for a trajectory τ is given by:

$$R(\tau) = \begin{cases} 1 & \tau \text{ ends at } s_H \\ 0 & \tau \text{ ends at } s_F \end{cases}$$

Using the same policy parametrization as above, consider off-policy policy gradients via importance sampling. Assume we want to compute policy gradients for a policy π_θ with samples drawn from $\pi_{\theta'}$.

(a) Write the policy gradient with importance sampling.

(b) Compute its variance.

(c) **Solution:**

(a) Closed-form expression for the policy gradient with importance sampling

We want to compute the policy gradient for a policy π_θ, but we're using samples from another policy $\pi_{\theta'}$. The policy gradient using importance sampling is:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta'}}\left[\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)}\nabla_\theta \log \pi_\theta(\tau) R(\tau)\right]$$

For this specific MDP: - A trajectory τ either ends at s_H with reward $R(\tau) = R_{\max} = 1$ or ends at s_F with $R(\tau) = 0$. - The importance weight is given by the ratio of probabilities of the trajectory under the two policies:

$$\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)} = \prod_{t=1}^{H}\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)}$$

Since each action a_t can be a_1 (continuing to the next state) or a_2 (transitioning to the failure state s_F), the importance weight for a trajectory τ that reaches s_H (with all actions a_1) is:

$$\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)} = \prod_{t=1}^{H}\frac{\pi_\theta(a_1|s_t)}{\pi_{\theta'}(a_1|s_t)} = \left(\frac{\theta}{\theta'}\right)^H$$

because the probability of taking a_1 in each state under π_θ is θ, and under $\pi_{\theta'}$ is θ'.

Thus, the policy gradient becomes:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta'}}\left[\left(\frac{\theta}{\theta'}\right)^H \nabla_\theta \log \pi_\theta(\tau) R(\tau)\right]$$

For trajectories that end at s_H, $R(\tau) = 1$, and for those that end at s_F, $R(\tau) = 0$.

Since $\log \pi_\theta(\tau) = H \log \theta$ for successful trajectories (all a_1), the policy gradient simplifies to:

$$\nabla_\theta J(\theta) = H\left(\frac{\theta}{\theta'}\right)^H \frac{1}{\theta}$$

This expression holds for trajectories that succeed in reaching s_H (because $R(\tau) = 1$ for these trajectories).

(b) Closed-form expression for the variance

The variance of the policy gradient can be computed as:

$$\mathrm{Var}(\nabla_\theta J(\theta)) = \mathbb{E}_{\tau\sim\pi_{\theta'}}\left[\left(\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)}\nabla_\theta \log \pi_\theta(\tau) R(\tau)\right)^2\right] - \left(\mathbb{E}_{\tau\sim\pi_{\theta'}}\left[\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)}\nabla_\theta \log \pi_\theta(\tau) R(\tau)\right]\right)^2$$

Let's break this down: 1. Expected squared term: For successful trajectories τ, we have $\nabla_\theta \log \pi_\theta(\tau) = H/\theta$ and $R(\tau) = 1$, so the squared term is:

$$\left(\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)}\nabla_\theta \log \pi_\theta(\tau) R(\tau)\right)^2 = \left(\left(\frac{\theta}{\theta'}\right)^H \frac{H}{\theta}\right)^2 = \left(\frac{H\theta^{H-1}}{(\theta')^H}\right)^2$$

The expectation over $\pi_{\theta'}$ would consider the probability of success under $\pi_{\theta'}$, which is $(\theta')^H$. Thus, the expected value becomes:

$$\mathbb{E}_{\tau\sim\pi_{\theta'}}\left[\left(\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)}\nabla_\theta \log \pi_\theta(\tau) R(\tau)\right)^2\right] = \left(\frac{H\theta^{H-1}}{\theta'}\right)^2$$

2. Square of the expected term: From part (a), we know the expected policy gradient is:

$$\mathbb{E}_{\tau\sim\pi_{\theta'}}\left[\frac{\pi_\theta(\tau)}{\pi_{\theta'}(\tau)}\nabla_\theta \log \pi_\theta(\tau) R(\tau)\right] = H\left(\frac{\theta}{\theta'}\right)^H \frac{1}{\theta}$$

Squaring this term gives:

$$\left(H\left(\frac{\theta}{\theta'}\right)^H \frac{1}{\theta}\right)^2 = \left(\frac{H\theta^{H-1}}{\theta'}\right)^2$$

Thus, the variance is:

$$\mathrm{Var}(\nabla_\theta J(\theta)) = \left(\frac{H\theta^{H-1}}{\theta'}\right)^2 - \left(\frac{H\theta^{H-1}}{\theta'}\right)^2 = 0$$

In this simple case, the variance is zero because we are assuming a deterministic environment with a clear outcome based on the actions taken, leading to no stochasticity in the gradient estimates. However, in practice, if the environment or the policy had any stochasticity, the variance could be non-zero.

10 Survey

Please estimate, in minutes, for each problem, how much time you spent (a) writing code and (b) waiting for the results. This will help us calibrate the difficulty for future homeworks.

- **Policy Gradients:**
- **Neural Network Baseline:**
- **Generalized Advantage Estimation:**
- **Hyperparameters and Sample Efficiency:**
- **Humanoid:**
- **Humanoid:**
- **Analysis – applying policy gradients:**
- **Analysis – PG variance:**
- **Analysis – return-to-go:**
- **Analysis – importance sampling:**