[SOLUTION TEMPLATE] Assignment 2: Policy Gradients

Due September 25, 11:59 pm

Student:Lu Hongyu

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.)

  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? **answer: reward-to-go**
  - Did advantage normalization help? **answer: yes**
  - Did the batch size make an impact? **answer: converge much faster and more stable**
- Provide the exact command line configurations (or `#@params` settings in Colab) you used to run your experiments, including any parameters changed from their defaults.

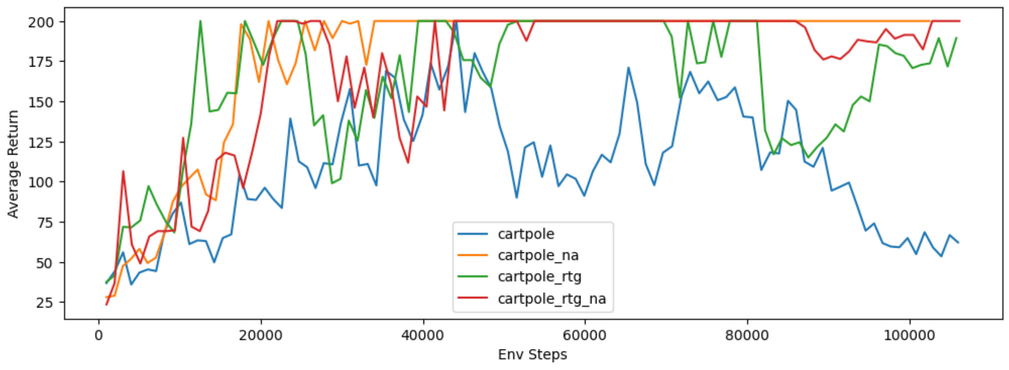

Figure 1: Cartpole

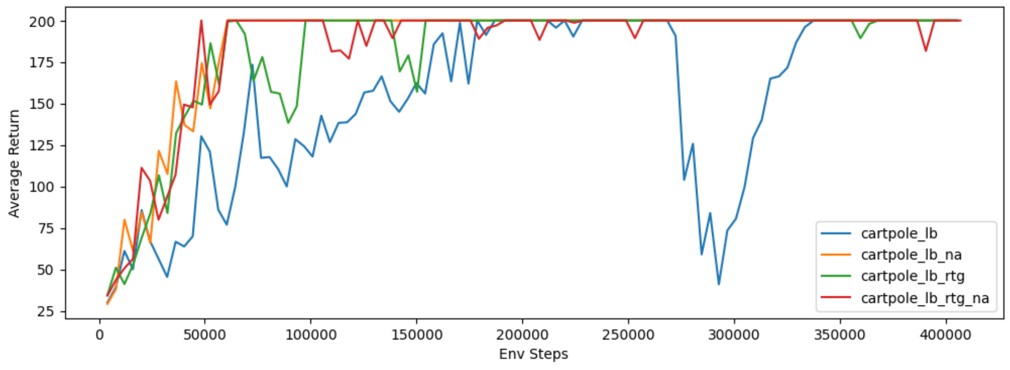

Figure 2: Cartpole_lb

5 Neural Network Baseline

- Plot a learning curve for the baseline loss.

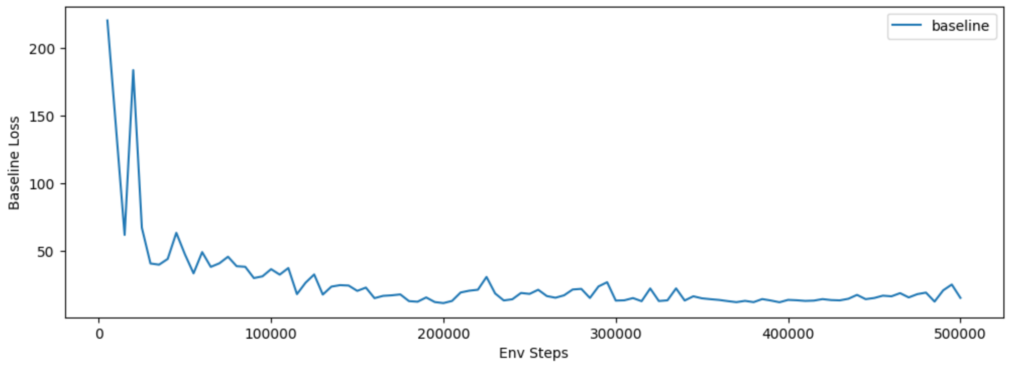

Figure 3: baseline_loss

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

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+).
- 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.
  Answer:
  When $\lambda = 0$, $A(s_t, a_t) = r(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t)$, which is biased.
  When $\lambda = 1$, $A(s_t, a_t) = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k) - V(s_t)$, which is unbiased but has high varience.

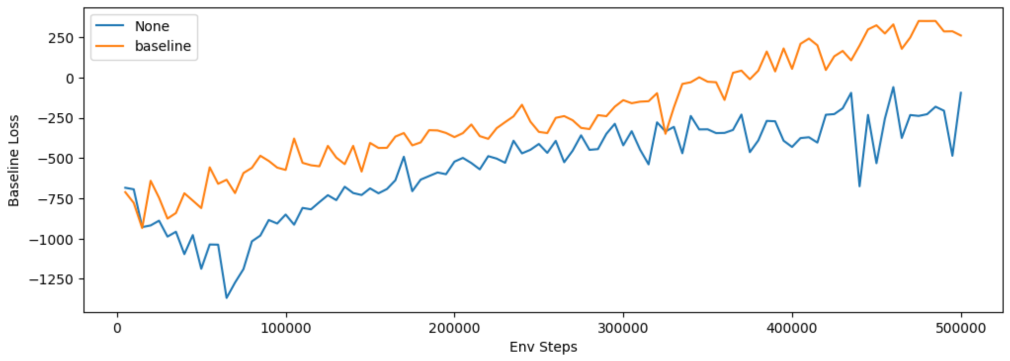

Figure 4: Cheetah

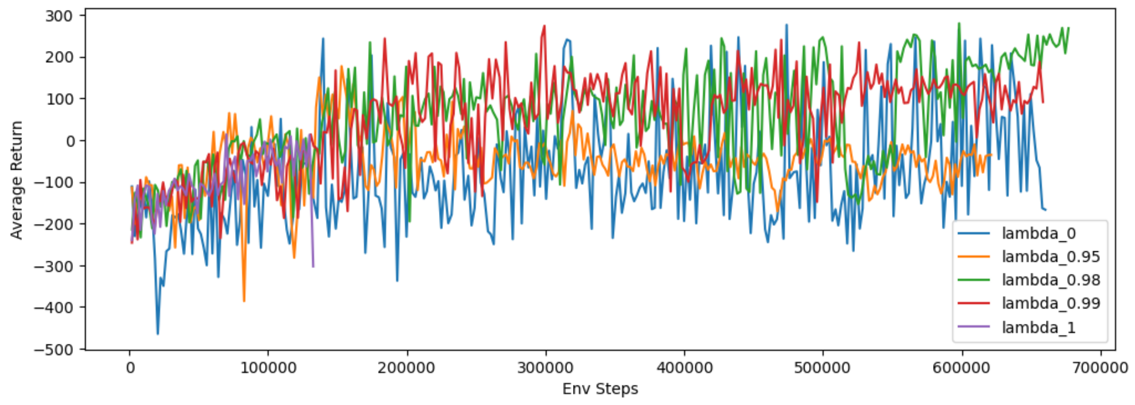

Figure 5: lunar_gae

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.
   Answer:
   As you can see from Figure 6 the one with learning rate of 0.02 learns fastest.

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).
   This experiment just takes too much time..

9 Analysis

Consider the following infinite-horizon MDP:

$$\begin{tikzcd} s_1 \arrow[loop left, "a_1"] \arrow[r, "a_2"] & s_F \end{tikzcd}$$

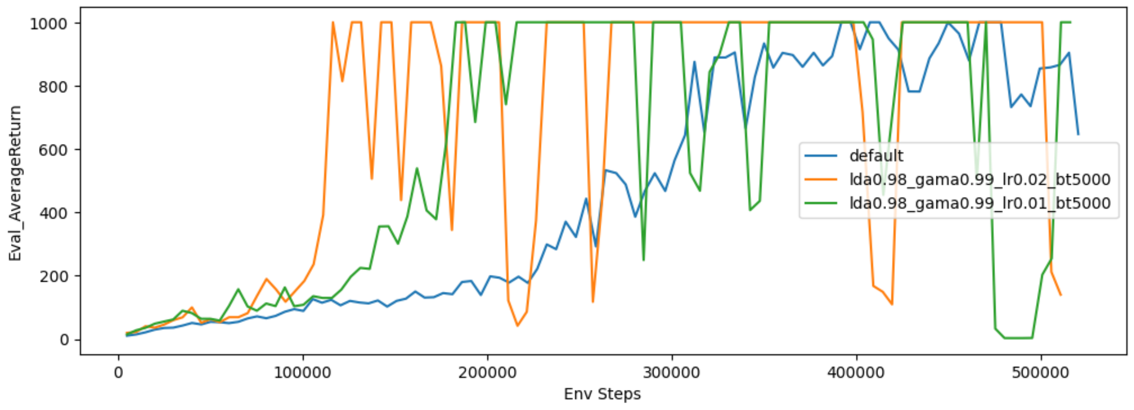

Figure 6: Pendulum

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: Assume that the steps of $R(\tau)$ is k. So:

$$\begin{aligned}\nabla J(\theta) =& \sum_{k=1}^{\infty} R(\tau_k)P_\theta(\tau_k)\nabla log(P_\theta(\tau_k)) \\ =& \sum_{k=1}^{\infty}(k-1)\theta^{k-1}(1-\theta)(\frac{k-1}{\theta} + \frac{1}{\theta - 1}) \\ =& \sum_{k=1}^{\infty} k\theta^k(1-\theta)(\frac{k}{\theta} - \frac{1}{1-\theta}) \\ =&(1-\theta)\sum_{k=1}^{\infty} k^2\theta^{k-1} - \sum_{k=1}^{\infty} k\theta^k \\ =&(1-\theta)\nabla\sum_{k=1}^{\infty} k\theta^k - \sum_{k=1}^{\infty} k\theta^k \\ =&\frac{1}{(1-\theta)^2}\end{aligned}$$

   (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:

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta} R(\tau) &= \sum_{k=1}^{\infty} R(\tau_k)\nabla P_\theta(\tau_k) \\
&= \sum_{k=1}^{\infty}(k-1)\nabla(\theta^{k-1}(1-\theta)) \\
&= \nabla(1-\theta)\sum_{k=1}^{\infty} k\theta^k \\
&= \frac{1}{(1-\theta)^2}
\end{aligned}$$

So that this matches the result of policy gradient.

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:

$$\begin{aligned}
Var(\nabla_\theta J(\theta)) &= E(J(\theta)^2) - E(J(\theta))^2 \\
&= \sum_{k=1}^{\infty} R(\tau_k)^2(\nabla log P_\theta(\tau_k))^2 P_\theta(\tau_k) - \frac{1}{(1-\theta)^4} \\
&= \frac{4\theta^2+8\theta+1}{\theta(1-\theta)^4}
\end{aligned}$$

   When θ=1 or 0,the varience goes to ∞. When $\theta \approx 0.11$,the varience arrive at the minimal.

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

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

   Solution:

$$\begin{aligned}
\nabla_\theta J(\theta) &= E(\sum_{t=1}^{k} \nabla log\pi_\theta(a_1|s_1)\sum_{t'=t}^{k} r(s_1)) \\
&= \sum_{k=1}^{\infty}(\sum_{t=1}^{k}(\nabla log\pi_\theta(a_1|s_1)\sum_{t'=t}^{k} r(s_1)))\theta^k(1-\theta) \\
&= \frac{1}{(1-\theta)^2}
\end{aligned}$$

   (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:

$$
\begin{aligned}
Var\nabla_\theta J(\theta) =& E([\sum_{t=1}^{k} \nabla log\pi_\theta(a_1|s_1) \sum_{t'=t}^{k} r(s_1)]^2) - E(\sum_{t=1}^{k} \nabla log\pi_\theta(a_1|s_1) \sum_{t'=t}^{k} r(s_1))^2 \\
=& \sum_{k=1}^{\infty} (\sum_{t=1}^{k} (\nabla log\pi_\theta(a_1|s_1) \sum_{t'=t}^{k} r(s_1)))^2 \theta^k (1-\theta) - \frac{1}{(1-a)^4} \\
=& \sum_{k=1}^{\infty} \frac{1}{4} k^2 (k+1)^2 \theta^{k-2} (1-\theta) - \frac{1}{(1-a)^4} \\
=& \frac{\theta^2 + 3\theta + 1}{\theta(1-\theta)^4}
\end{aligned}
$$

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

$$\begin{tikzcd}
s_1 \arrow[r, "a_1"] \arrow[rrd, "a_2"'] & s_2 \arrow[r, "a_1"] \arrow[rd, "a_2"] & s_3 \arrow[r, "a_1"] \arrow[d, dashed, "a_2"] & \cdots \arrow[r] \arrow[ld, dashed, "a_2"'] & s_H \\
 & & s_F & &
\end{tikzcd}$$

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. **Solution:**

$$
\begin{aligned}
\nabla_\theta J(\theta) =& E_{\tau\sim P_{\theta'}(\tau)}[\frac{P_\theta(\tau)}{P_{\theta'}(\tau)} \nabla_\theta \log \pi_\theta(\tau) r(\tau)] \\
=& (H-1)\theta^{H-2}
\end{aligned}
$$

(b) Compute its variance. **solution:**

$$
\begin{aligned}
Var\nabla_\theta J(\theta) =& E_{\tau\sim P_{\theta'}(\tau)}[(\frac{P_\theta(\tau)}{P_{\theta'}(\tau)} \nabla_\theta \log \pi_\theta(\tau) r(\tau))^2] - E_{\tau\sim P_{\theta'}(\tau)}[\frac{P_\theta(\tau)}{P_{\theta'}(\tau)} \nabla_\theta \log \pi_\theta(\tau) r(\tau)]^2 \\
=& \frac{(H-1)^2 \theta^{2H-4}}{\theta'^{H-1}} - (H-1)^2 \theta^{2H-4} \\
=& (\frac{1}{\theta'^{H-1}} - 1)(H-1)^2 \theta^{2H-4}
\end{aligned}
$$

The importance sampling is unbiased but it has large varience when the high-reward trajectories have low probability in the sample distribution.

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:**