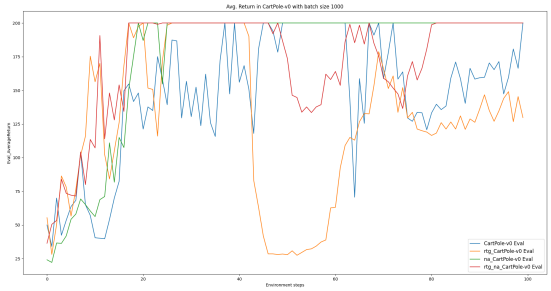
Assignment 2: Policy Gradients

Due September 25, 11:59 pm

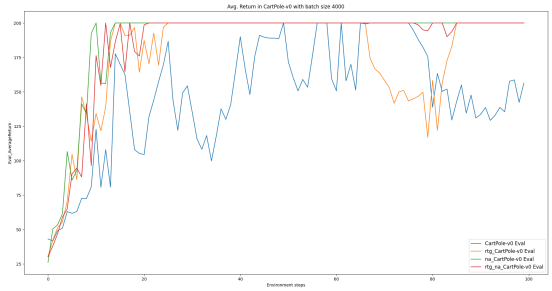
submitted by Chanjoon Park
Mar 18, 2024

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.)



(a) 1000 batch size



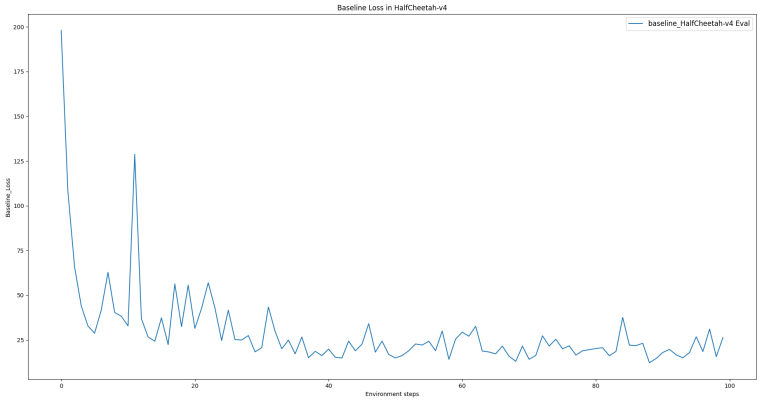
(b) 4000 batch size

- Answer the following questions briefly:
  - **Q1.** Which value estimator has better performance without advantage normalization: the trajectory-centric one, or the one using reward-to-go?
    A1. In small batch size, it is hard to tell the reward-to-go(Yellow) is definitely better than the trajectory-centric one(Blue). However, in large batch size, the reward-to-go shows better performance.
  - **Q2.** Did advantage normalization help?
    A2. Yes. Overall, the advantage normalization helps to stabilize the learning process.
  - **Q3.** Did the batch size make an impact?
    A3. Yes. In large batch size, the learning process is more stable and the performance is better.

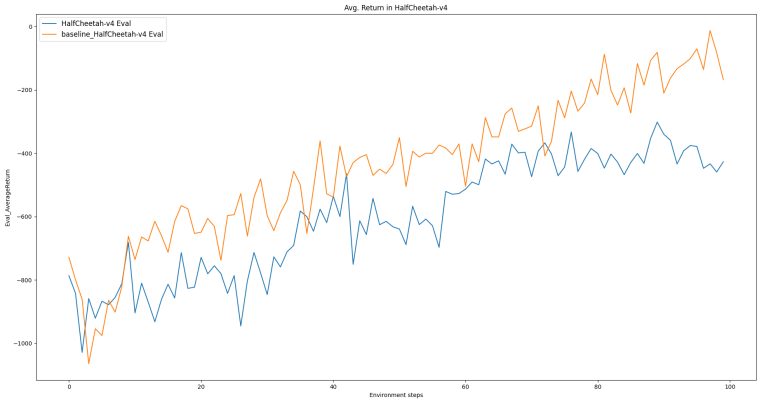
NOTE. The exact command line configurations for each experiment are provided in README.md.

5 Neural Network Baseline

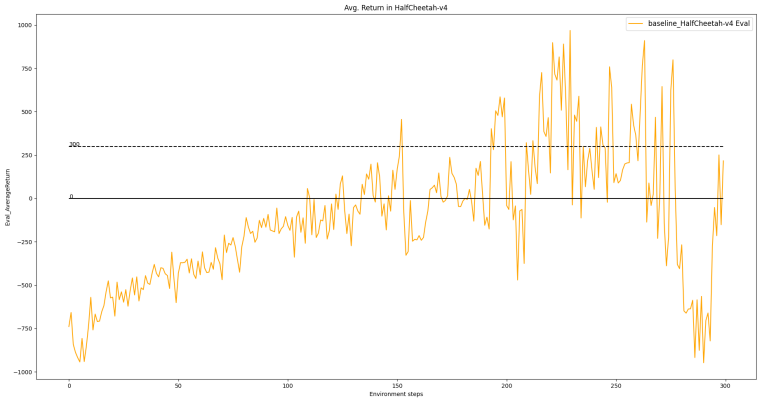
- Plot a learning curve for the baseline loss.



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



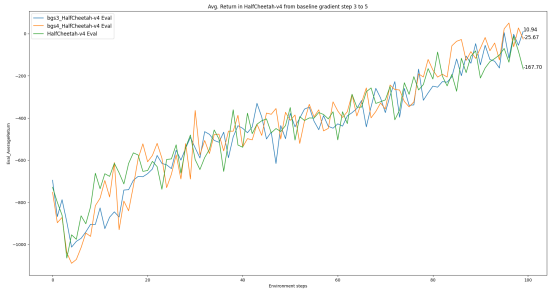
With the given command-line arguments, it failed to achieve the average return over 300.



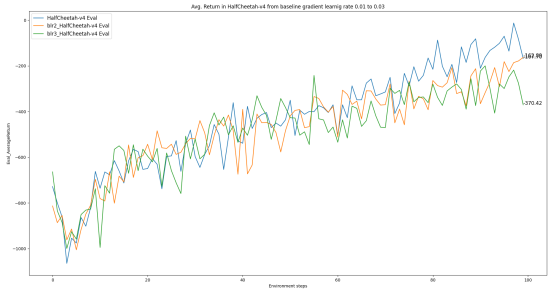
But as we can see, with 300 steps, it achieved the average return almost 300.

- 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?



(a) Baseline gradient step



(b) Baseline learning rate

Interestingly, the smaller baseline gradient step shows better performance than the larger one. And as we know, the smaller baseline learning rate shows better performance than the larger one.

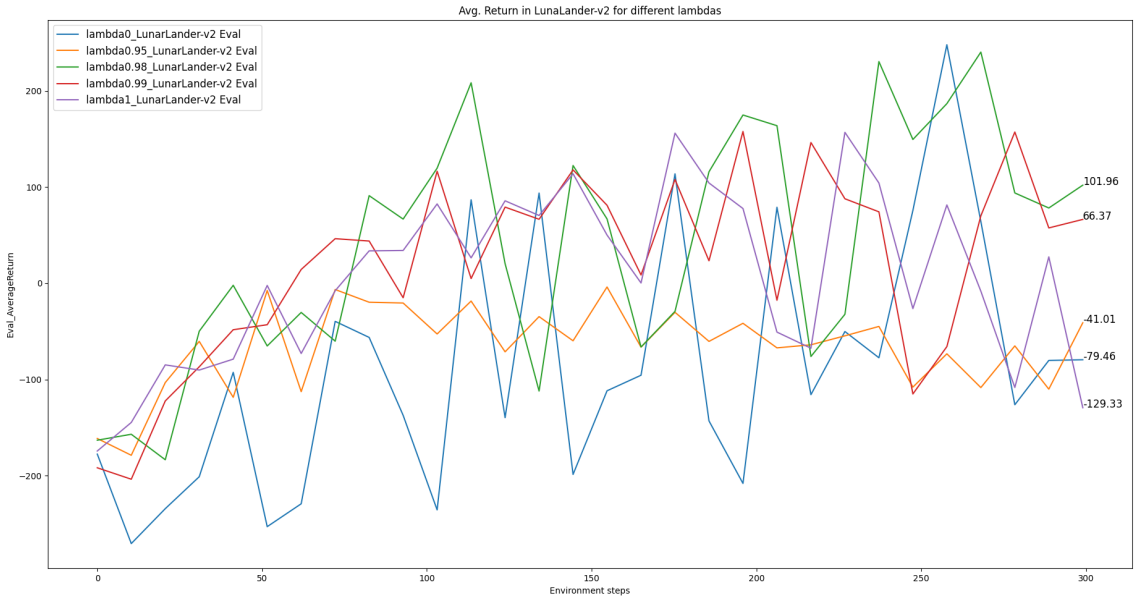
But this is just 3 experiments, so it is hard to tell the exact relationship between the baseline gradient step and the baseline learning rate.

- **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!

  Attached video files are in README.md. But as you can see, it is not walking expectedly.

6 Generalized Advantage Estimation

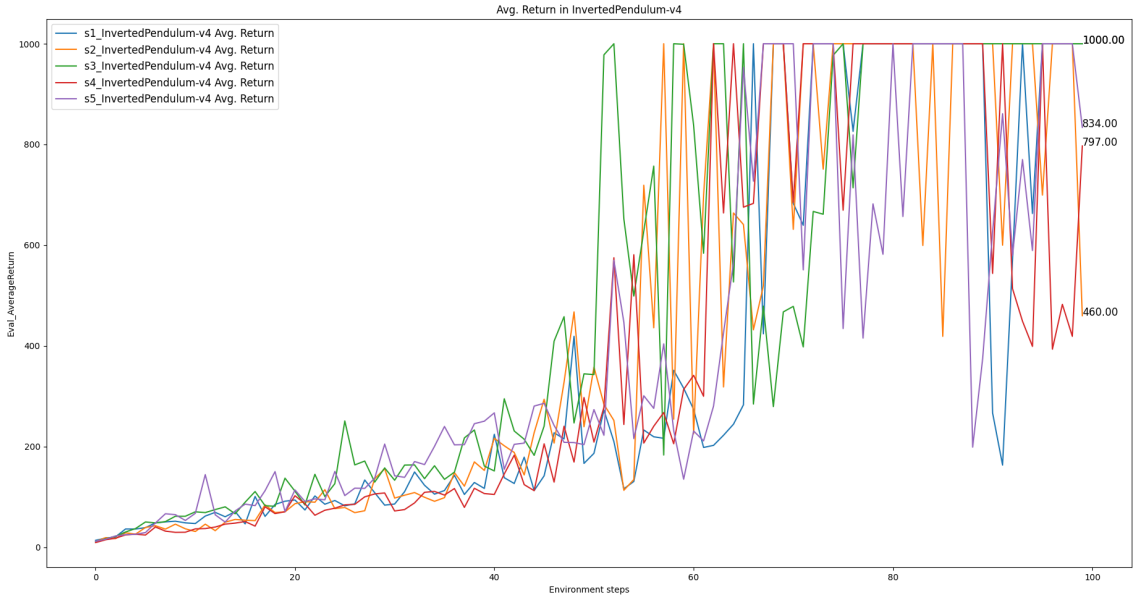
- Provide a single plot with the learning curves for the `LunarLander-v2` experiments that you tried. Describe in words how λ affected task performance. The run with the best performance should achieve an average score close to 200 (180+).
- Consider the parameter λ. What does $\lambda = 0$ correspond to? What about $\lambda = 1$? Relate this to the task performance in `LunarLander-v2` in one or two sentences.



Please note that this plot is smoothed with interpolation due to the large variance of the return. As we can see, the $\lambda = 0.98$ and 0.99 shows the best performance. The $\lambda = 0$ corresponds to the one-step TD error for estimating the advantage function, and the $\lambda = 1$ corresponds to the Monte Carlo estimate.

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.

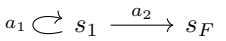


8 (Extra Credit) Humanoid

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).

9 Analysis

Consider the following infinite-horizon MDP:



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: The policy gradient $\nabla_\theta J(\theta)$ is given by $\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta} R(\tau) = \mathbb{E}_{\tau\sim\pi_\theta} \nabla_\theta \log \pi_\theta(\tau) R(\tau)$. In our case, $\nabla_\theta \log \pi_\theta(a_1|s_1) = \nabla_\theta \log \theta = \frac{1}{\theta}$.

   the expecte return $R(\tau)$ is given by $\sum_{k=1}^{\infty} k\theta^{k-1}(1-\theta)$ for $\tau = s_1, a_1, \ldots, s_1, a_2$ at step k. With the Hint provided,

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

   and

   $$R(\tau) = \sum_{k=1}^{\infty} k\theta^{k-1}(1-\theta) = \frac{1}{1-\theta}$$

   Now we can say the gradient of the expected return is given by

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

   (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: For $\tau = s_1, a_1, \ldots, s_1, a_2$ at step k, the expected return of the policy $J(\theta)$ is given by

   $$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta} R(\tau) = \frac{1}{1-\theta}$$

   The gradient of this expression with respect to θ is given by

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

   , which does not match the policy gradient computed in part 1 (a).

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:

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

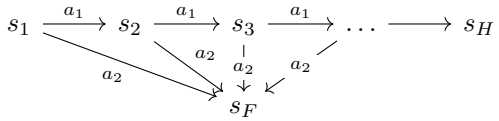
   (a) Write the modified policy gradient and confirm that it is unbiased.

   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.

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:**