HW2 - Policy Gradients

Ryan Yang

September 17, 2018

Problem 1.1. Please show that: $\sum_{t=1}^{T} \mathbb{E}_{\tau \sim p_\theta(\tau)}[\nabla_\theta log \pi_\theta(a_t|s_t)(b(s_t))] = 0$.

Solution.

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}_{\tau \sim p_\theta(\tau)}[\nabla_\theta log \pi_\theta(a_t|s_t)(b(s_t))] &= \sum_{\tau} p_\theta(\tau) \nabla_\theta log \pi_\theta(a_t|s_t)(b(s_t)) \\
&= \sum_{\tau} p_\theta(s_t, a_t) p_\theta(\frac{\tau}{s_t, a_t}|s_t, a_t) \nabla_\theta log \pi_\theta(a_t|s_t)(b(s_t)) \\
&= \sum_{\tau} p_\theta(\frac{\tau}{s_t, a_t}|s_t, a_t) p_\theta(s_t) p_\theta(a_t|s_t) \nabla_\theta log \pi_\theta(a_t|s_t)(b(s_t)) \\
&= \sum_{\tau} p_\theta(\frac{\tau}{s_t, a_t}|s_t, a_t) p_\theta(s_t) \nabla_\theta p_\theta(s_t|a_t) b(s_t) \\
&= \sum_{s_1} \sum_{a_1} ... \sum_{s_t} b(s_t) p_\theta(s_t) \sum_{a_t} p_\theta(\frac{\tau}{s_t, a_t}|s_t, a_t) \nabla_\theta p_\theta(a_t|s_t) \qquad (1) \\
&= \sum_{s_t} b(s_t) p_\theta(s_t) \sum_{a_t} \nabla_\theta p_\theta(a_t|s_t) \sum_{\frac{\tau}{s_t, a_t}} p_\theta(\frac{\tau}{s_t, a_t}|s_t, a_t) \\
&= \sum_{s_t} b(s_t) p_\theta(s_t) \sum_{a_t} \nabla_\theta p_\theta(a_t|s_t) \\
&= \sum_{s_t} b(s_t) p_\theta(s_t) \nabla_\theta 1 \\
&= 0
\end{aligned}$$

□

Problem 1.2.1. Explain why, for the inner expectation, conditioning on $(s_1, a_1, ..., a_{t^*} - 1, s_{t^*})$ is equivalent to conditioning only on s_{t^*}.

Solution. The inner expectation consists of the following expectation: $\mathbb{E}_{s_{t^*+1}:s_T, a_{t^*}:a_T}[\nabla_\theta log \pi_\theta(a_t|s_{t^*}) b(s_{t^*})|(s_1, a_1, ..., a_{t^*} - 1, s_{t^*})]$. But, the term $log \pi_\theta(a_t|s_{t^*}) b(s_{t^*})$ only depends on s_{t^*}, so by the Markov property, we can reduce the inner term to only being conditioned on s_t*, since s_t* is independent of all previous states and actions. □

Problem 1.2.2. Using the iterated expectation described above, show that: $\nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta(\tau)}[b(s_{t^*})] = 0$

Solution. From question 1.2.1 and citing the Law of Iterated Expectation, we are able to write the entire expectation as $\mathbb{E}_{s_0:s_{t^*}, a_0:a_{t^*-1}}[\mathbb{E}_{s_{t^*+1}:s_T, a_{t^*}:a_T}[\nabla_\theta log \pi_\theta(a_t|s_{t^*}) b(s_{t^*})|s_{t^*}]]$. Since the inner expectation is not over s_{t^*}, we can pull that term out of the inner expectation to get: $\mathbb{E}_{s_0:s_{t^*}, a_0:a_{t^*-1}}[b(s_{t^*}) \mathbb{E}_{s_{t^*+1}:s_T, a_{t^*}:a_T}[\nabla_\theta log \pi_\theta(a_t|s_{t^*})|s_{t^*}]]$.

The inner expectation can now be simplified as follows:

$$\begin{aligned}
\mathbb{E}_{s_{t^*+1}:s_T, a_{t^*}:a_T}[\nabla_\theta log\pi_\theta(a_t|s_{t^*})|s_{t^*}] &= \sum_{a_{t^*}} \sum_{s_{t^*+1}} ... \sum_{s_T} \pi_\theta(a_{t^*}|s_{t^*}) p(s_{t^*+1}|s_{t^*}, a_{t^*}) ... p(s_T|s_{T-1}, a_{T-1})(\nabla_\theta log\pi_\theta(a_{t^*}|s_{t^*})) \\
&= \sum_{a_{t^*}} \pi_\theta(a_{t^*}|s_{t^*}) \nabla_\theta log\pi_\theta(a_{t^*}|s_{t^*}) \sum_{s_{t^*+1}} p(s_{t^*+1}|s_{t^*}, a_{t^*}) \sum_{a_{t^*+1}} ... \sum_{s_T} p(s_T|s_{T-1}, a_{T-1}) \\
&= \sum_{a_{t^*}} \pi_\theta(a_{t^*}|s_{t^*}) \nabla_\theta log\pi_\theta(a_{t^*}|s_{t^*}) \\
&= \mathbb{E}_{a_{t^*}}[\nabla_\theta log\pi_\theta(a_{t^*}|s_{t^*})] \\
&= \int \frac{\nabla_\theta \pi_\theta(a_{t^*}|s_{t^*})}{\pi_\theta(a_{t^*}|s_{t^*})} \pi_\theta(a_{t^*}|s_{t^*}) da_{t^*} \\
&= \nabla_\theta \int \pi_\theta(a_{t^*}|s_{t^*}) da_{t^*} \\
&= \nabla_\theta 1 = 0
\end{aligned} \tag{2}$$

The above is true since $\sum_{s_{t^*+1}} p(s_{t^*+1}|s_{t^*}, a_{t^*}) \sum_{a_{t^*+1}} ... \sum_{s_T} p(s_T|s_{T-1}, a_{T-1}) = 1$. Now, we can write the entire expectation as: $\mathbb{E}_{s_0:s_{t^*}, a_0:a_{t^*-1}}[b(s_{t^*}) \cdot 0]$. This just equals 0. $\square$

CS294-112 Homework 2: Policy Gradients

Problem 4:

Figure 1: Average returns vs. Number of iterations for SB

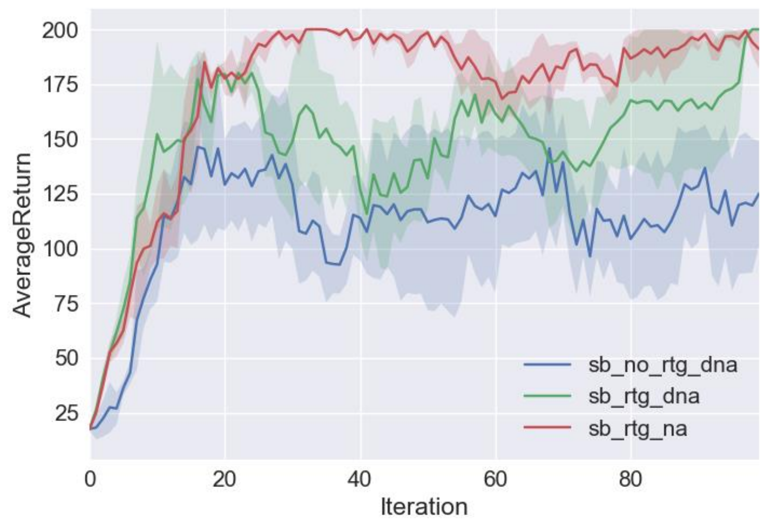
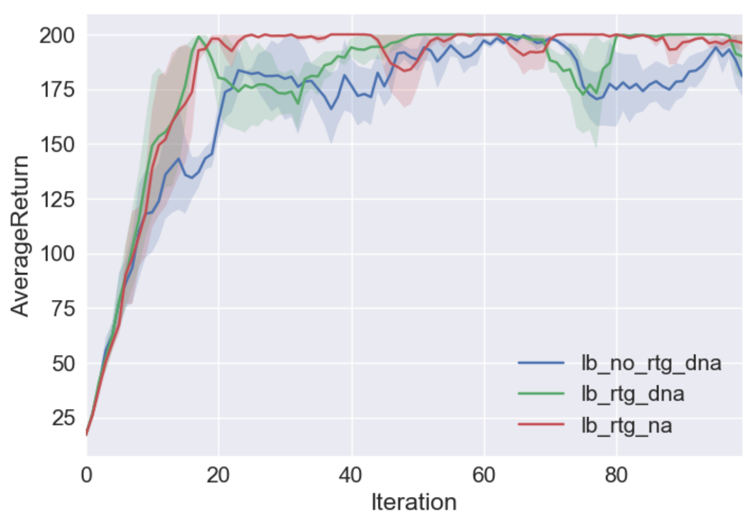


Figure 2: Average returns vs. Number of iterations for LB



Short Answers:

Which gradient estimator has better performance without advantage centering – the trajectory centric one, or the one using reward to go?

The one using reward to go has better performance without advantage centering.

Did advantage centering help?

Yes

Did the batch size make an impact?

Yes. A larger batch size allowed for a faster convergence rate.

Command Line Expressions:

python train_pg_f18.py CartPole-v0 -n 100 -b 1000 -e 3 -dna --exp_name sb_no_rtg_dna

python train_pg_f18.py CartPole-v0 -n 100 -b 1000 -e 3 -rtg -dna --exp_name sb_rtg_dna

python train_pg_f18.py CartPole-v0 -n 100 -b 1000 -e 3 -rtg --exp_name sb_rtg_na

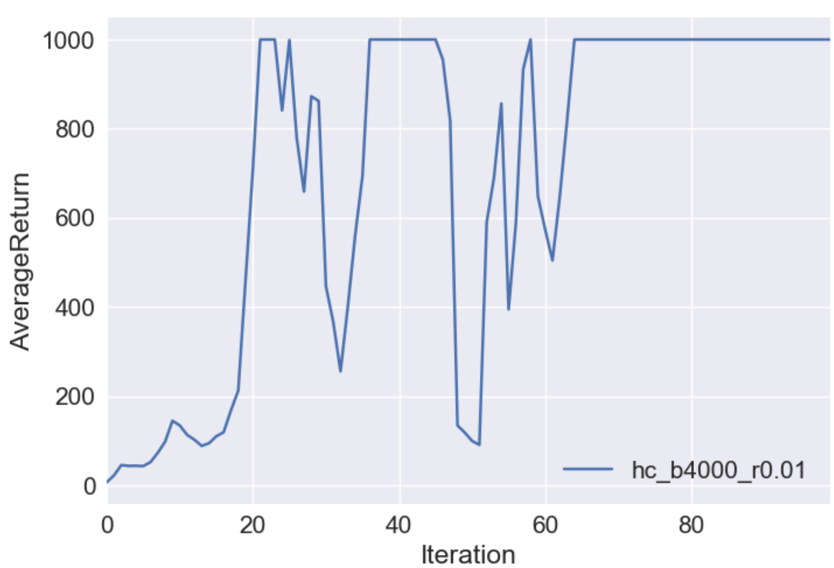
python train_pg_f18.py CartPole-v0 -n 100 -b 5000 -e 3 -dna --exp_name lb_no_rtg_dna

python train_pg_f18.py CartPole-v0 -n 100 -b 5000 -e 3 -rtg -dna --exp_name lb_rtg_dna

python train_pg_f18.py CartPole-v0 -n 100 -b 5000 -e 3 -rtg --exp_name lb_rtg_na

Problem 5:

Figure 3: Average returns vs. Number of iterations for InvertedPendulum. I used a batch size of 4000 and a learning rate of 0.01.

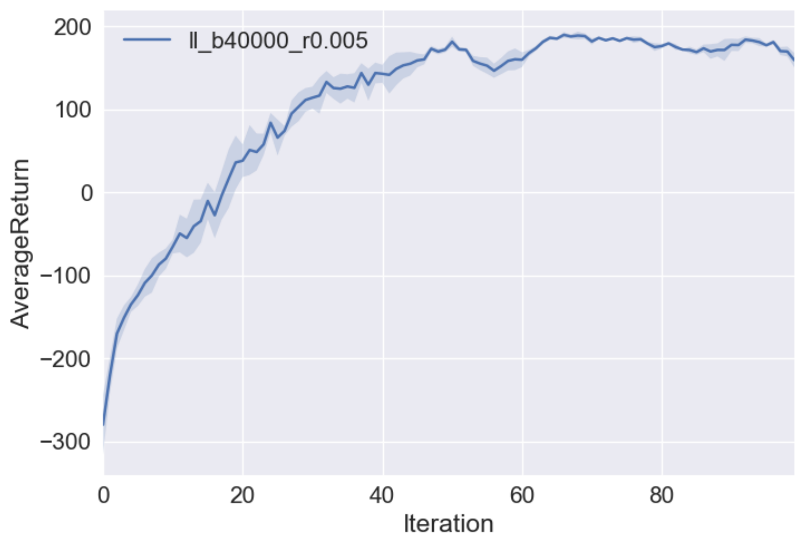


Command Line Expressions:

python train_pg_f18.py InvertedPendulum-v2 -ep 1000 --discount 0.9 -n 100 -e 3 -l 2 -s 64 -b 4000 -lr 0.01 -rtg --exp _name hc_b4000_r0.01

Problem 7:

Figure 4: Average returns vs. Number of iterations for LunarLander

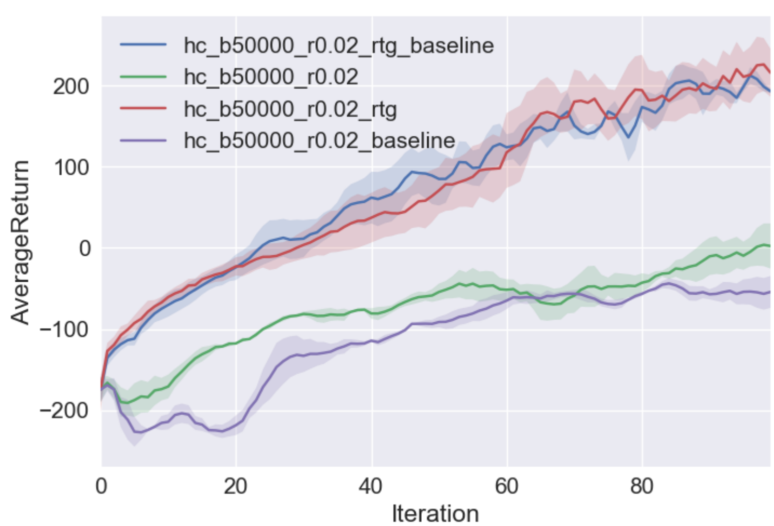


Command Line Expressions:

python train_pg_f18.py LunarLanderContinuous-v2 -ep 1000 --discount 0.99 -n 100 -e 3 -l 2 -s 64 -b 40000 -lr 0.005 -rtg --nn_baseline --exp_name ll_b40000_r0.005

Problem 8:

Figure 5: Average returns vs. Number of Iterations for HalfCheetah



Short Answers:

How did the batch size and learning rate affect the performance?

In general, an increased batch size and lower learning rate correlated with a better performance. A batch size of 50000 and learning rate of 0.02 seemed to give optimal performance.

Command Line Expressions:

python train_pg_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 50000 -lr 0.02 --exp_name hc_b50000_r0.02

python train_pg_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 50000 -lr 0.02 -rtg --exp_name hc_b50000_r0.02_rtg

python train_pg_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 50000 -lr 0.02 --nn_baseline --exp_name hc_b50000_r0.02_baseline

python train_pg_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 50000 -lr 0.02 -rtg --nn_baseline --exp_name hc_b50000_r0.02_rtg_baseline