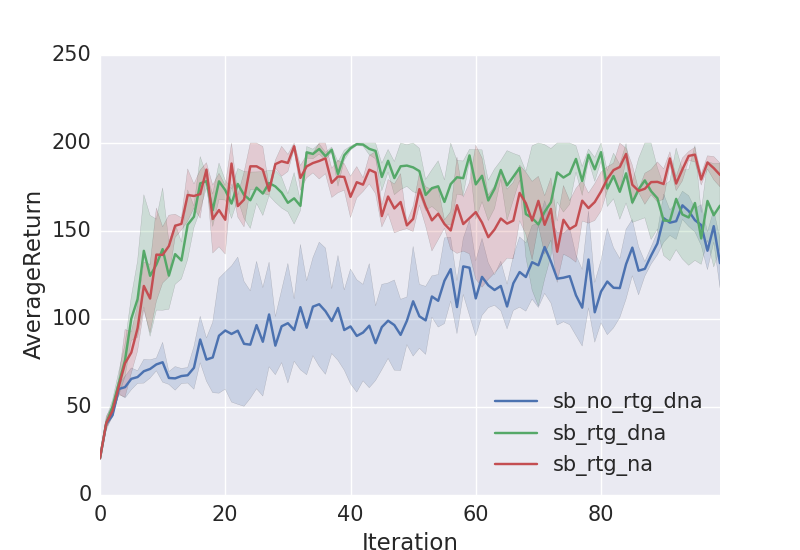
CS 294-112 HW2

Yujia Luod

**CartPole-v0**

Small Batch

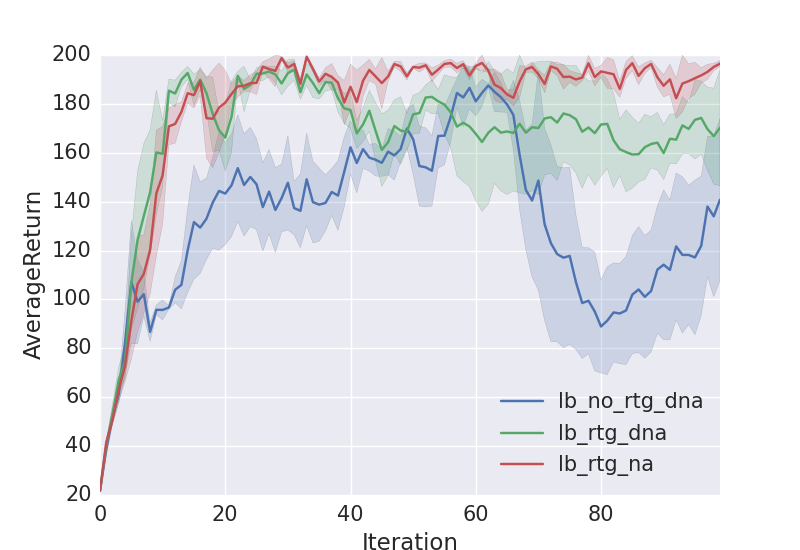


python train\_pg\_f18.py CartPole-v0 -n 100 -b 1000 -e 3 -dna --exp\_name sb\_no\_rtg\_dna

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

Big Batch



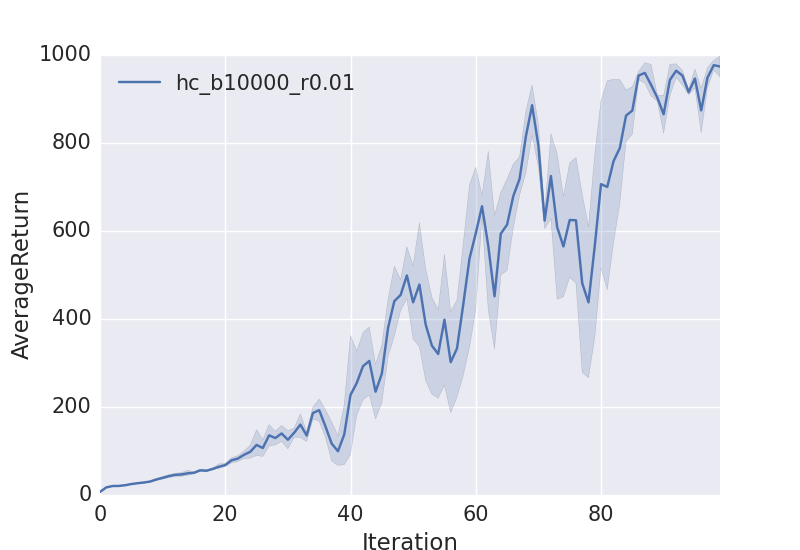
python train\_pg\_f18.py CartPole-v0 -n 100 -b 5000 -e 3 -dna --exp\_name lb\_no\_rtg\_dna

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

Comments: Reward-to-go gradient estimator converges faster. Advantage centering does not have an obvious effect on the learning curve. In terms of batch size, if using reward-to-go, bigger batch size results in more stable learning

**InvertedPendulum**



The smallest batch size is 10000, and biggest learning rate is 0.01

Commands:

python train\_pg\_f18.py InvertedPendulum-v2 -ep 1000 --discount 0.9 -n 100 -e 3 -l 2 -s 64 -b 1000 -lr 0.001 -rtg --exp\_name hc\_b1000\_r0.001

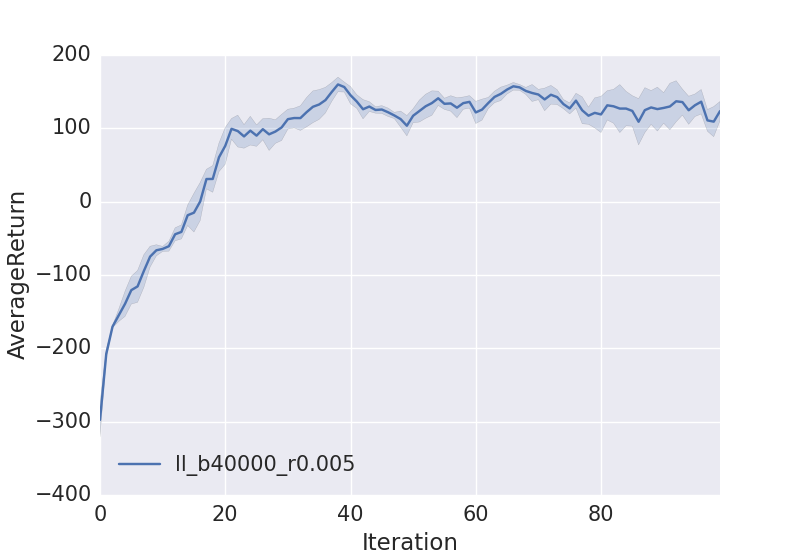
python train\_pg\_f18.py InvertedPendulum-v2 -ep 1000 --discount 0.9 -n 100 -e 3 -l 2 -s 64 -b 5000 -lr 0.001 -rtg --exp\_name hc\_b5000\_r0.001

python train\_pg\_f18.py InvertedPendulum-v2 -ep 1000 --discount 0.9 -n 100 -e 3 -l 2 -s 64 -b 10000 -lr 0.01 -rtg --exp\_name hc\_b10000\_r0.01

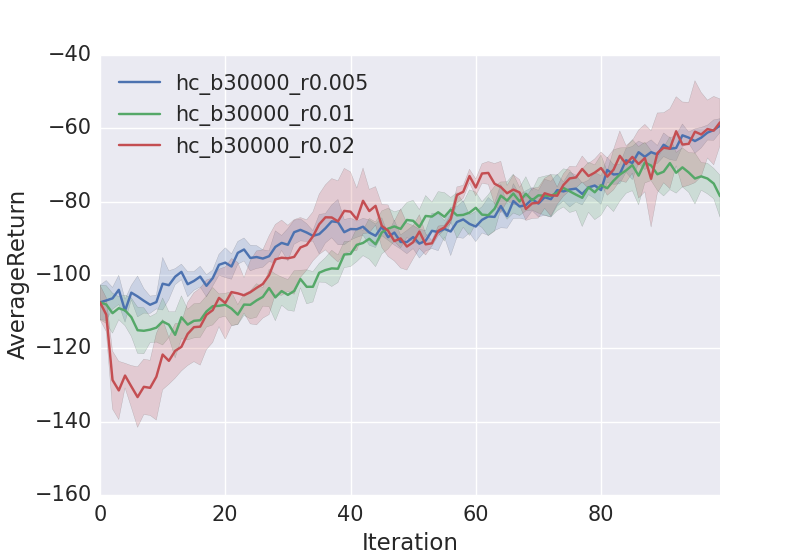
python train\_pg\_f18.py InvertedPendulum-v2 -ep 1000 --discount 0.9 -n 100 -e 3 -l 2 -s 64 -b 5000 -lr 0.01 -rtg --exp\_name hc\_b5000\_r0.01

python train\_pg\_f18.py InvertedPendulum-v2 -ep 1000 --discount 0.9 -n 100 -e 3 -l 2 -s 64 -b 5000 -lr 0.05 -rtg --exp\_name hc\_b5000\_r0.05

**LunarLander**

\

**HalfCheetah**

Effect of learning rate:

python train\_pg\_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 30000 -lr 0.005 --exp\_name hc\_b30000\_r0.005

python train\_pg\_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 30000 -lr 0.01 --exp\_name hc\_b30000\_r0.01

python train\_pg\_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 30000 -lr 0.02 --exp\_name hc\_b30000\_r0.02

Effect of batch size:

python train\_pg\_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 30000 -lr 0.02 --exp\_name hc\_b30000\_r0.02

python train\_pg\_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 10000 -lr 0.02 --exp\_name hc\_b10000\_r0.02

python train\_pg\_f18.py HalfCheetah-v2 -ep 150 --discount 0.9 -n 100 -e 3 -l 2 -s 32 -b 10000 -lr 0.02 --exp\_name hc\_b50000\_r0.02

How do batch size and learning curve affect

Reward to go and baseline method