Homework 2

Suchismita Padhy

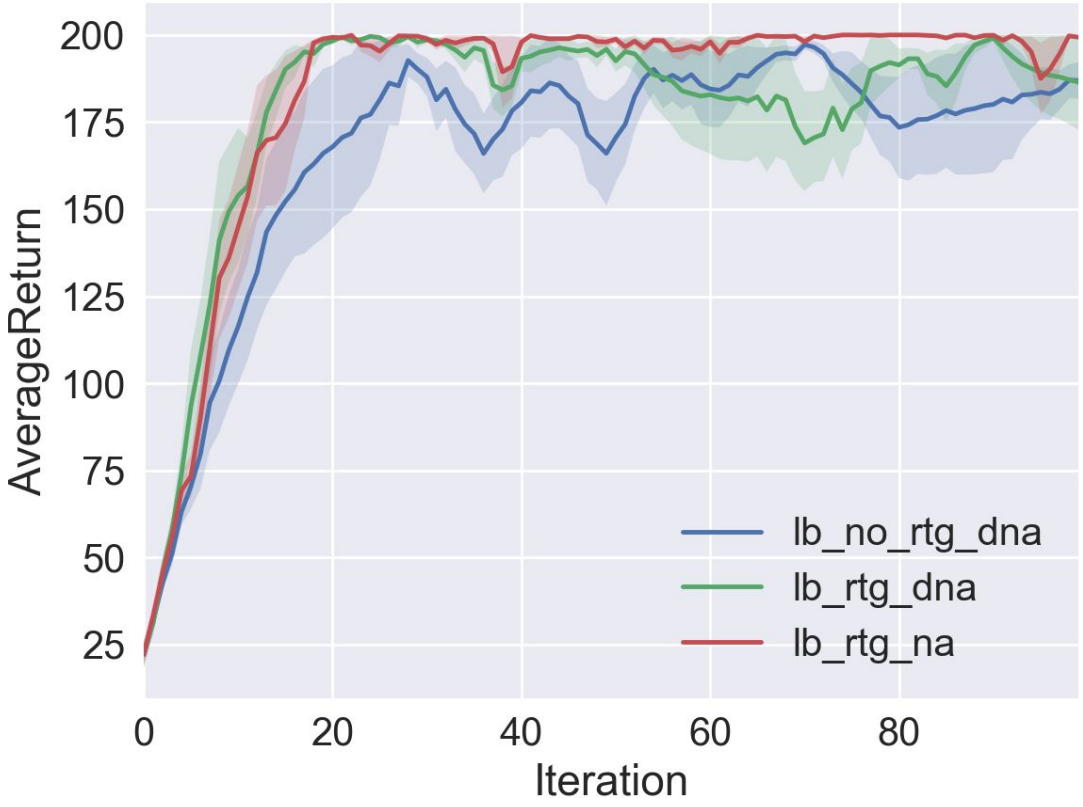
4. Cartpole :

– Which gradient estimator has better performance without advantage-centering :the trajectory-centric one, or the one using reward-to-go?
The reward-to-go seem to be having a better performance.

–- Did advantage centering help?
Yes. It made the learning more stable.

–- Describe what you expected from the math. do the empirical results match the theory?

Yes. Reward-to-go is supposed to reduce variance, and when we enable rtg, we see better convergence. Similarly, for advantage normalization we saw better performance as expected.



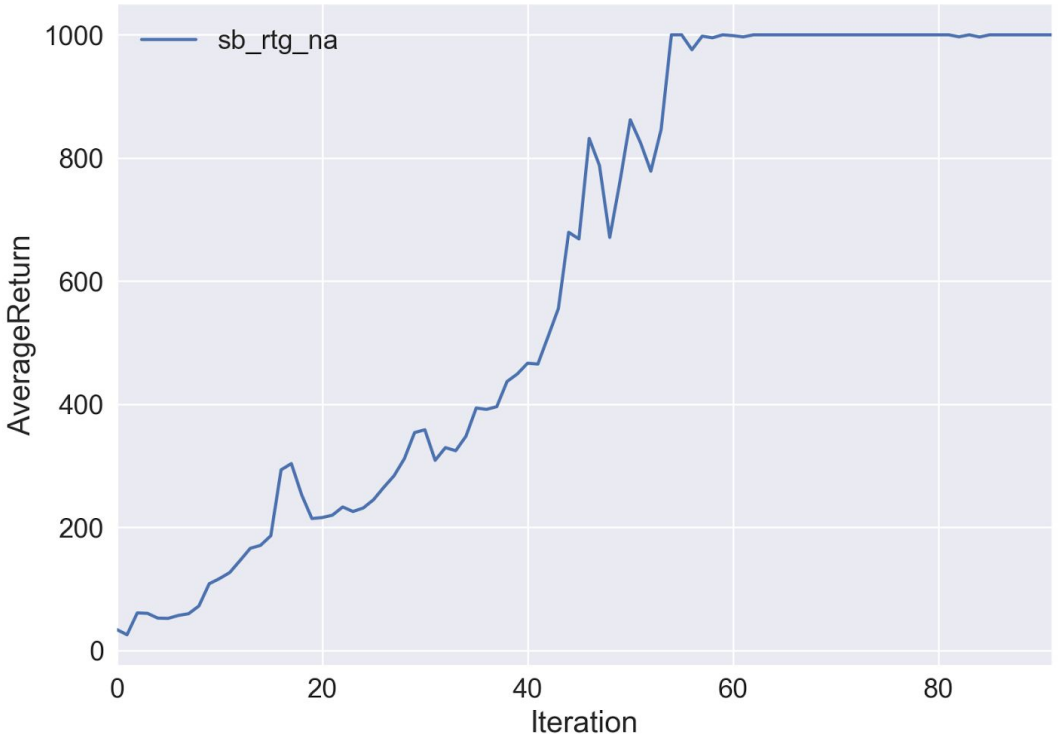
–- Did the batch size make an impact?
Larger batch sizes increased the robustness of the training. When a large enough batch was used the oscillations in reward was eliminated to a large degree. However, the training time increased with increase in batch size.

Commandline params used :

```
python train_pg.py CartPole-v0 -n 100 -b 1000 -e 5 -dna --exp_name sb_no_rtg_dna
python train_pg.py CartPole-v0 -n 100 -b 1000 -e 5 -rtg -dna --exp_name sb_rtg_dna
python train_pg.py CartPole-v0 -n 100 -b 1000 -e 5 -rtg --exp_name sb_rtg_na
```

```
python train_pg.py CartPole-v0 -n 100 -b 5000 -e 5 -dna --exp_name lb_no_rtg_dna
python train_pg.py CartPole-v0 -n 100 -b 5000 -e 5 -rtg -dna --exp_name lb_rtg_dna
python train_pg.py CartPole-v0 -n 100 -b 5000 -e 5 -rtg --exp_name lb_rtg_na
```

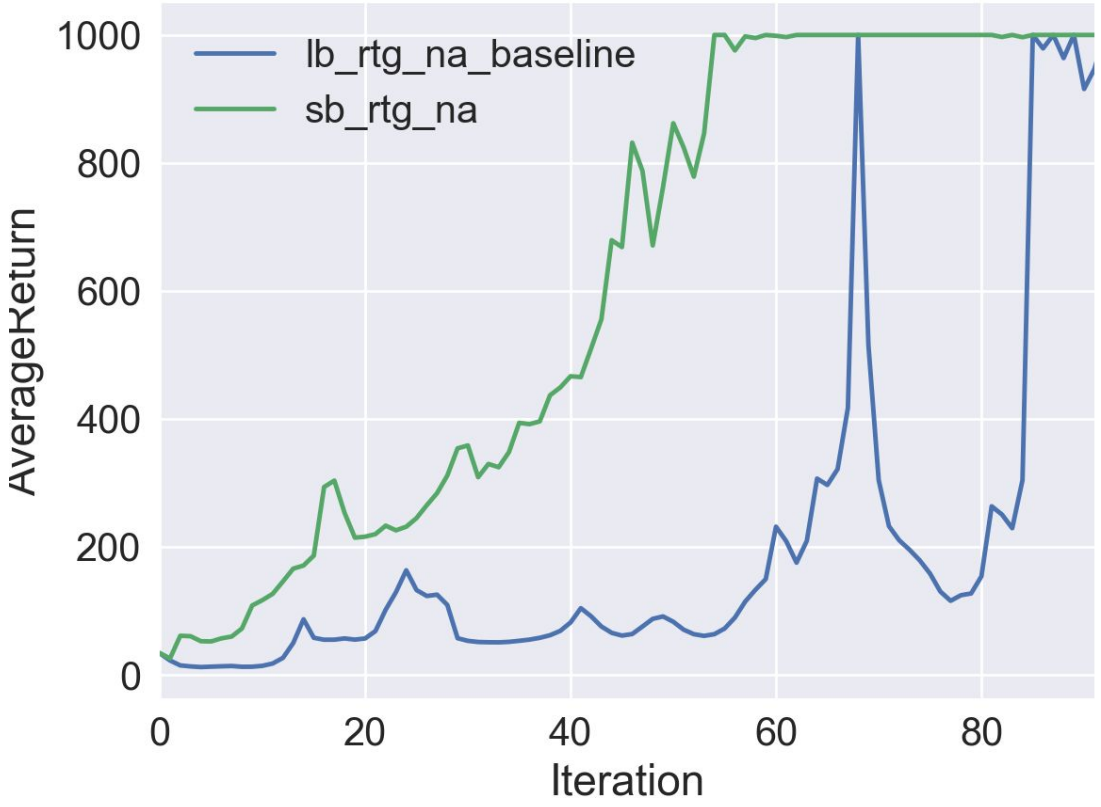
Inverted Pendulum :



Smallest batch size to make it work : 5000
I used rewards to go with advantage normalization.

```
python train_pg.py InvertedPendulum-v1  -lr 1e-3 -n 100 -b 5000 --size 200 -e 1 -rtg --exp_name sb_rtg_na
```

5. Neural Network Baseline



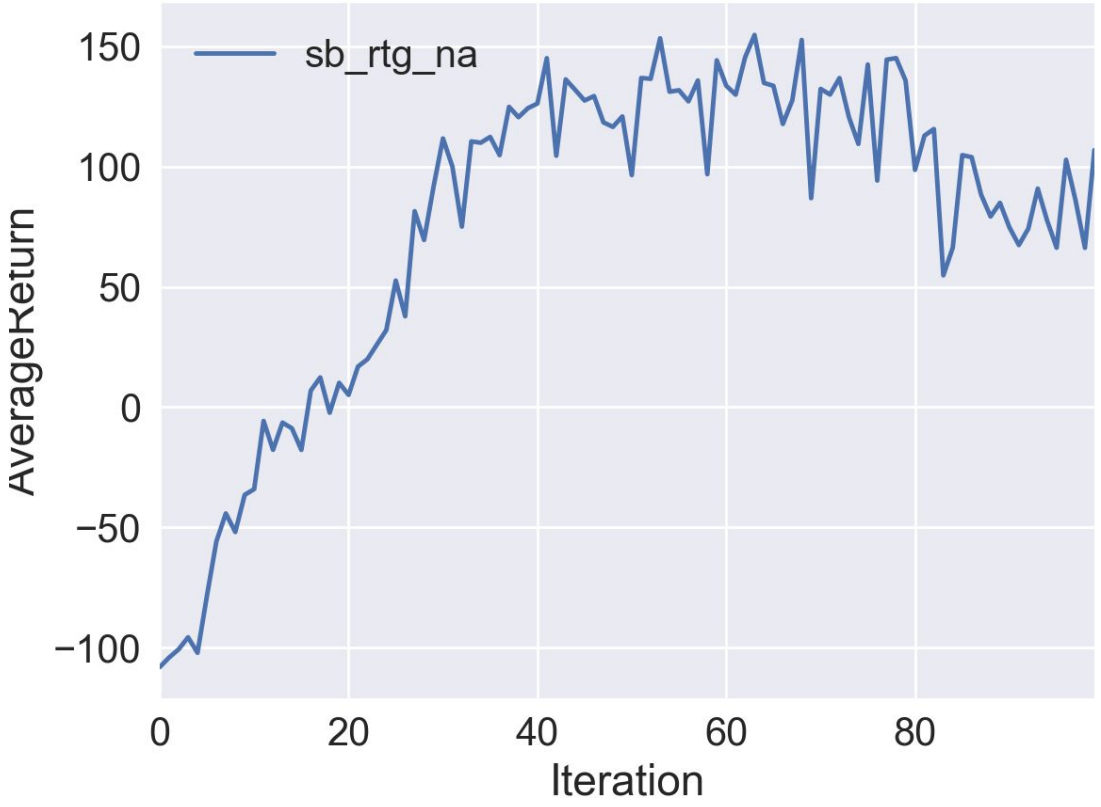
*batch size for both was 5000.

```
python train_pg.py InvertedPendulum-v1  -lr 1e-3 -n 100 -b 5000 --size 200 -bl -e 1 -rtg
--exp_name lb_rtg_na_baseline
```

6. Half Cheetah

Commandline options

```
python train_pg.py HalfCheetah-v1 -ep 150 --discount 0.9 -n 100 -b 5000 --size 256 -e 1 -rtg
--exp_name sb_rtg_na
```


sb_rtg_na
150
100
50
0
−50
−100
AverageReturn
0
20
40
60
80
Iteration