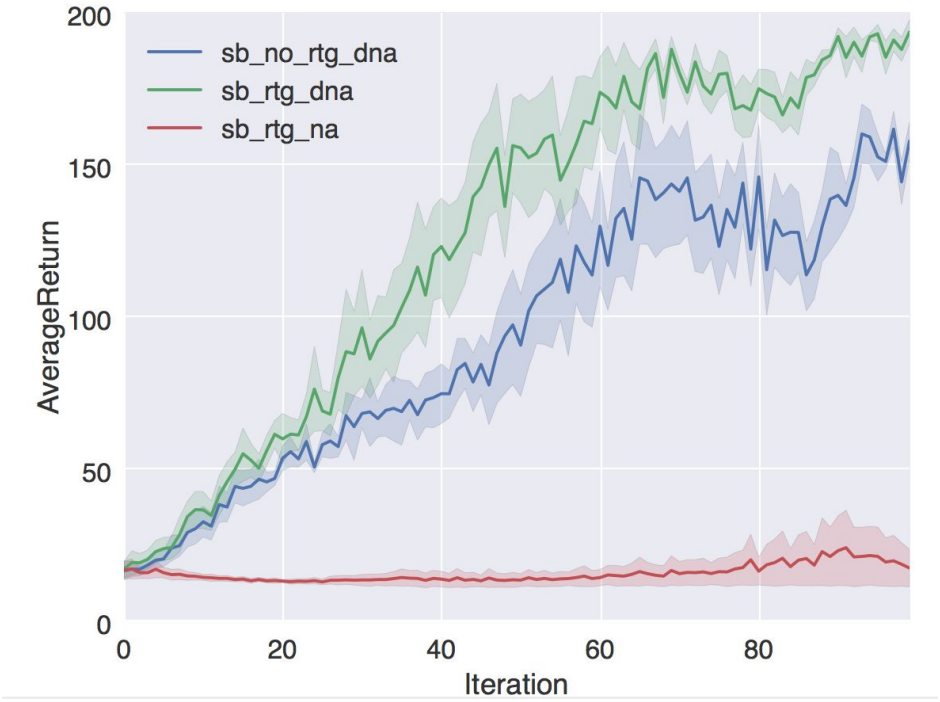
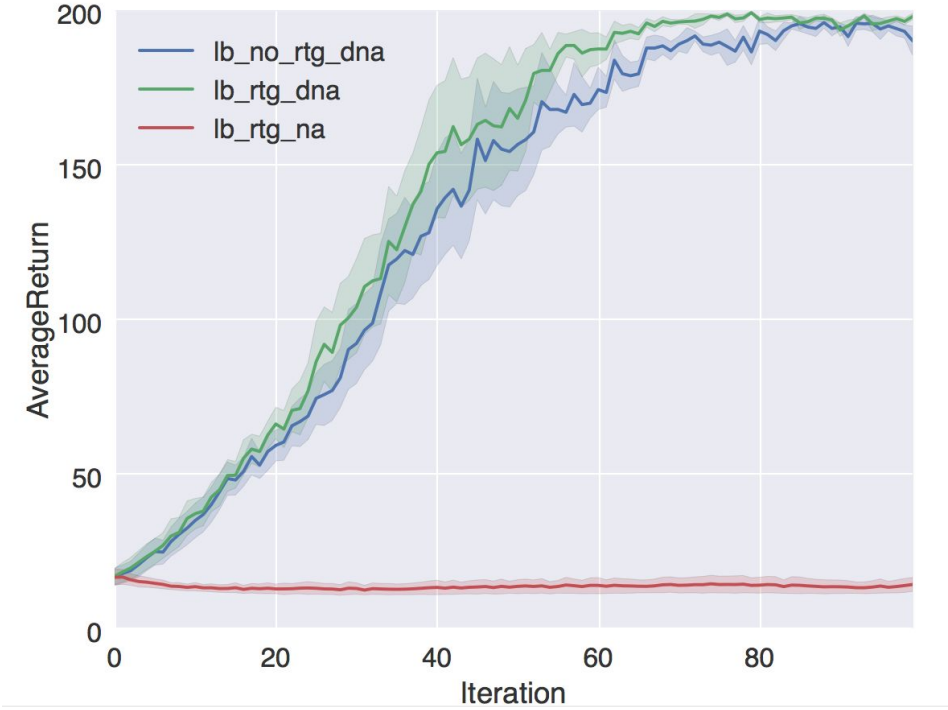
Deep Reinforcement Learning: Homework 2 Report
9/20/2017
Yifat Amir

CARTPOLE EXPERIMENTS

1. Learning curves for small batch experiments



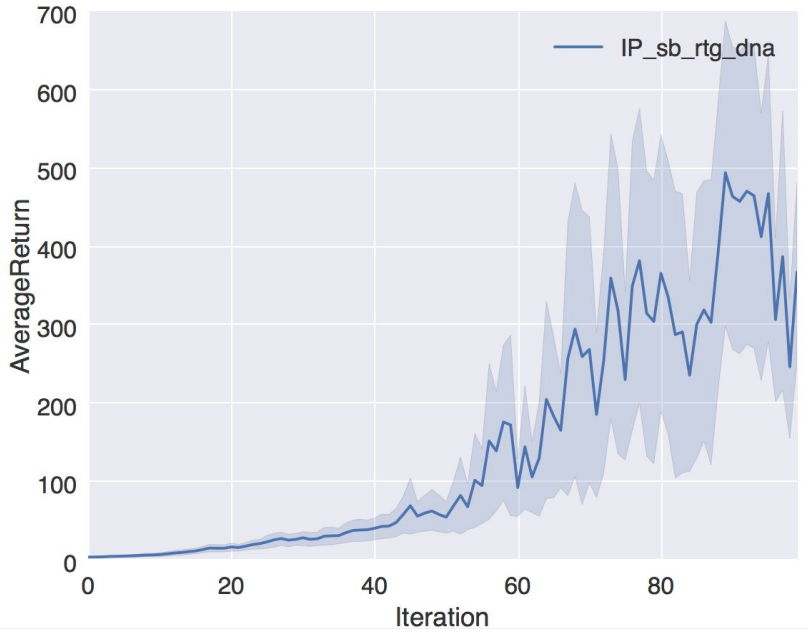
2. Learning curves for large batch experiments



3. **Answers to questions:**
    a. Reward-to-go is the gradient estimator that has better performance without advantage-centering in comparison to the trajectory-centric one.
    b. Advantage centering did not help-- in fact it performed awfully.
    c. The empirical results match the theory since I expected reward-to-go to perform better since it captures more information than the trajectory-centric reward calculation.
    d. Higher batch size made convergence at an average reward of 200 happen faster.
    e. My exact command line configurations used to run experiments are below:
        i. `python train_pg.py CartPole-v0 -n 100 -b 1000 -e 5 -dna --exp_name sb_no_rtg_dna`
        ii. `python train_pg.py CartPole-v0 -n 100 -b 1000 -e 5 -rtg -dna --exp_name sb_rtg_dna`
        iii. `python train_pg.py CartPole-v0 -n 100 -b 1000 -e 5 -rtg --exp_name sb_rtg_na`
        iv. `python train_pg.py CartPole-v0 -n 100 -b 5000 -e 5 -dna --exp_name lb_no_rtg_dna`
        v. `python train_pg.py CartPole-v0 -n 100 -b 5000 -e 5 -rtg -dna --exp_name lb_rtg_dna 3`
        vi. `python train_pg.py CartPole-v0 -n 100 -b 5000 -e 5 -rtg --exp_name lb_rtg_na`

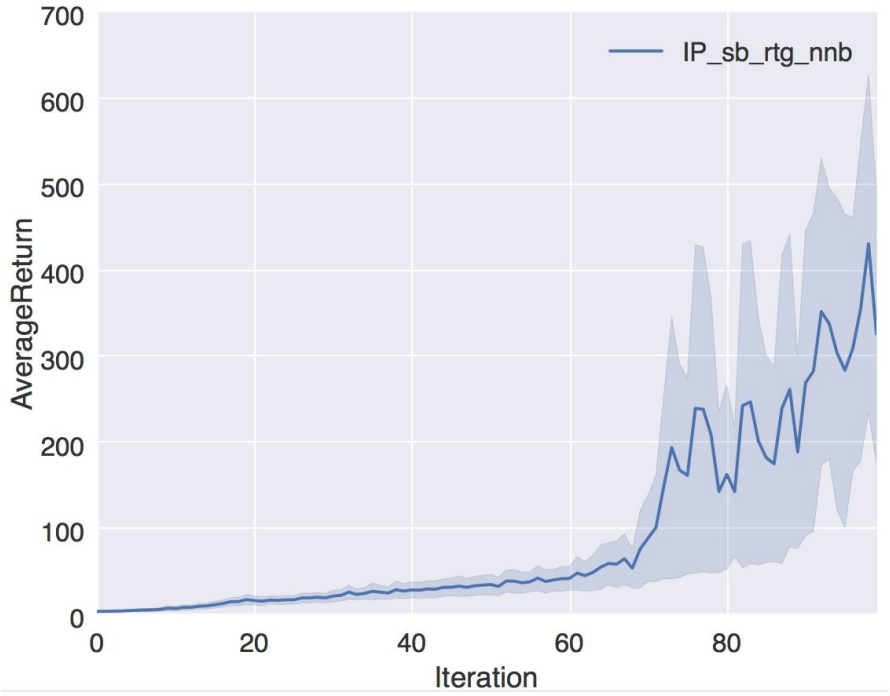
INVERTED PENDULUM EXPERIMENTS

1. Provide a learning curve where the policy gets to optimum (maximum score of 1000) in less than 100 iterations. (This may be for a single random seed, or averaged over multiple.) (Also, your policy performance may fluctuate around 1000—this is fine.)
    a. `python train_pg.py InvertedPendulum-v1 -n 100 -b 1000 -e 5 -rtg -dna --exp_name IP_sb_rtg_dna`



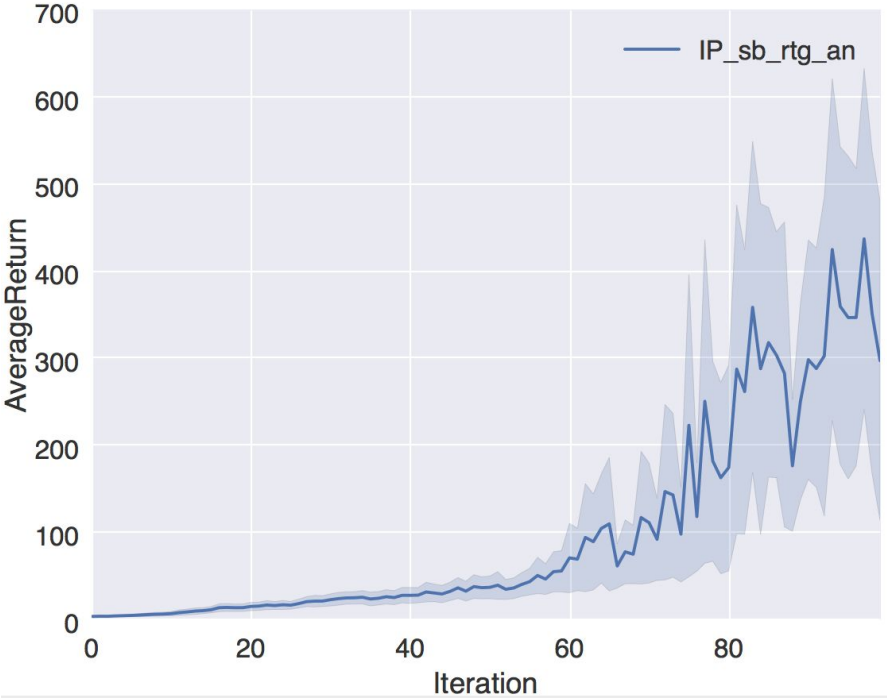
2. Learning curve with both the NN baseline function and advantage normalization

```
python train_pg.py InvertedPendulum-v1 -n 100 -b 1000 -e 5 -rtg
--nn_baseline --exp_name IP_sb_rtg_nnb
```



VS. learning curve without NN baseline but with advantage normalization (not very different)

```
python train_pg.py InvertedPendulum-v1 -n 100 -b 1000 -e 5 -rtg
--exp_name IP_sb_rtg_an
```



HALF CHEETAH EXPERIMENTS

1. Find any settings which result in the agent attaining an average score of 150 or more at the end of 100 iterations, and provide a learning curve.

```
python train_pg.py HalfCheetah-v1 -ep 150 --discount 0.9 -rtg
--nn_baseline -b 50000 -n 100 -e 3
```

[As of now I have not gotten to an average score of 150. I will submit this anyways since I've been trying.]