CS 285 Homework 2

Jeffrey Cheng

September 26, 2022

Experiment 1 (CartPole)

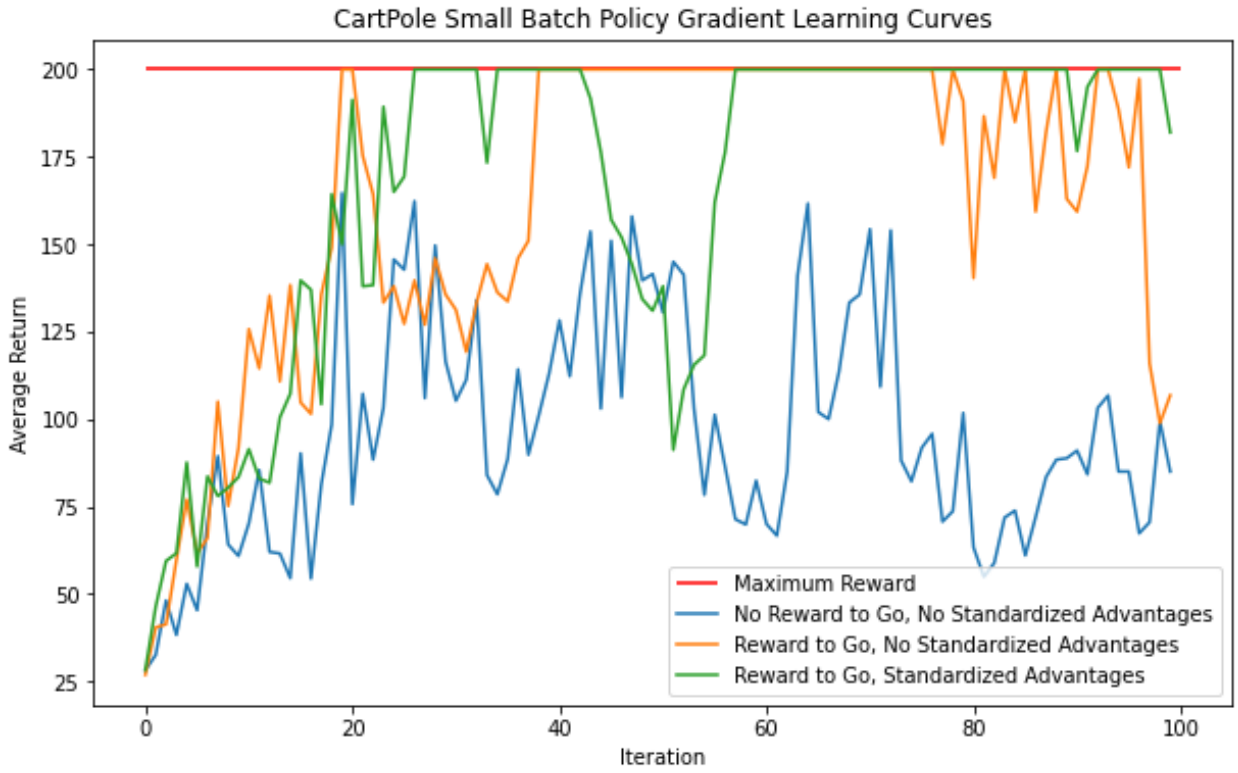

Figure 1: Learning curves for various policy gradient algorithms in the CartPole environment, using small batches.

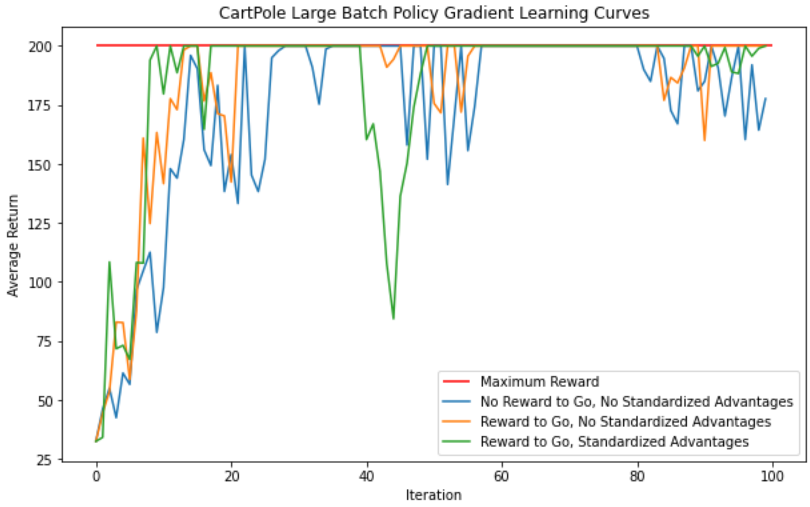

Figure 2: Learning curves for various policy gradient algorithms in the CartPole environment, using large batches.

Reward-to-go appears to be the better value estimator for both batch sizes. Standardizing the advantages seems to reach the maximum reward faster as well as stabilize better towards the end of the iterations. Larger batches seem to improve the speed of learning as well as improve stability across all algorithms.

The following commands were used in this experiment:

```
python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
-dsa --exp_name q1_sb_no_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
-rtg -dsa --exp_name q1_sb_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
-rtg --exp_name q1_sb_rtg_na

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
-dsa --exp_name q1_lb_no_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
-rtg -dsa --exp_name q1_lb_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
-rtg --exp_name q1_lb_rtg_na
```

Experiment 2 (InvertedPendulum)

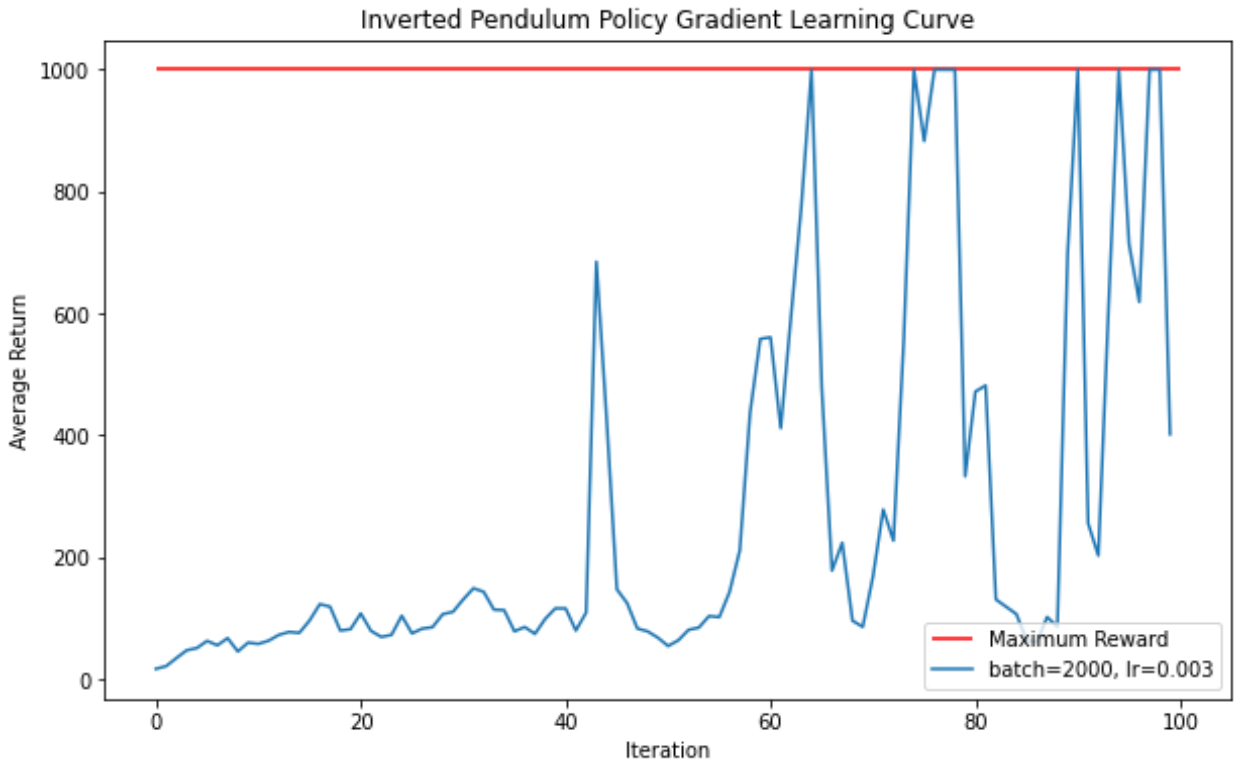

Figure 3: Learning curve for configuration where optimum is reached in 100 iterations in the InvertedPendulum environment. Batch size 2000, learning rate 7e-3.

The following command was used in this experiment:

```
python cs285/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
--ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b 2000 -lr 7e-3 -rtg \
--exp_name q2_b2000_r7e-3
```

Experiment 3 (LunarLander)

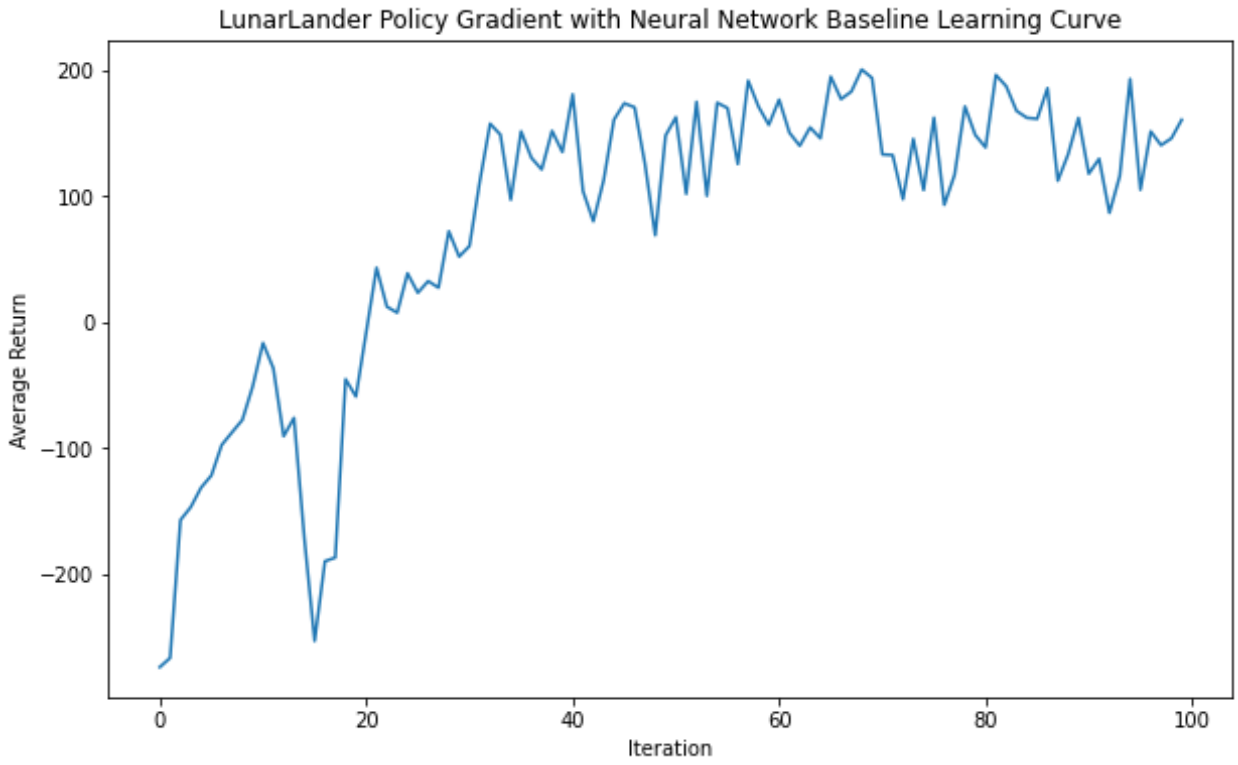

Figure 4: Learning curve for policy gradient with neural network baseline and reward-to-go in InvertedPendulum Environment

The following command was used in this experiment:

```
python cs285/scripts/run_hw2.py \
--env_name LunarLanderContinuous-v2 --ep_len 1000 \
--discount 0.99 -n 100 -l 2 -s 64 -b 40000 -lr 0.005 \
--reward_to_go --nn_baseline --exp_name q3_b40000_r0.005
```

Experiment 4 (HalfCheetah)

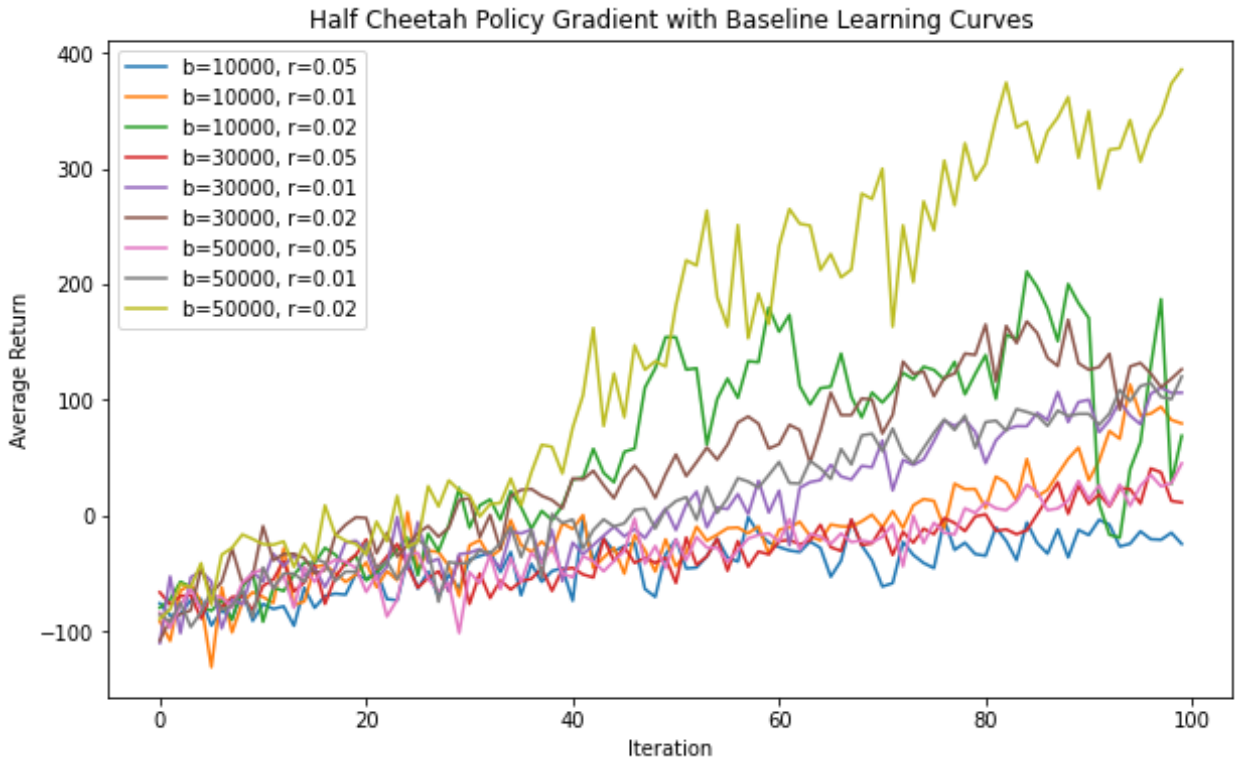

Figure 5: Learning curves for various batch sizes and learning rates in the HalfCheetah Environment

Low learning rates tend to correlate with slow, but stable learning. Larger batch sizes tend to correlate with faster learning. The largest batch size and learning rate configuration experienced the best performance at the end of 100 iterations.

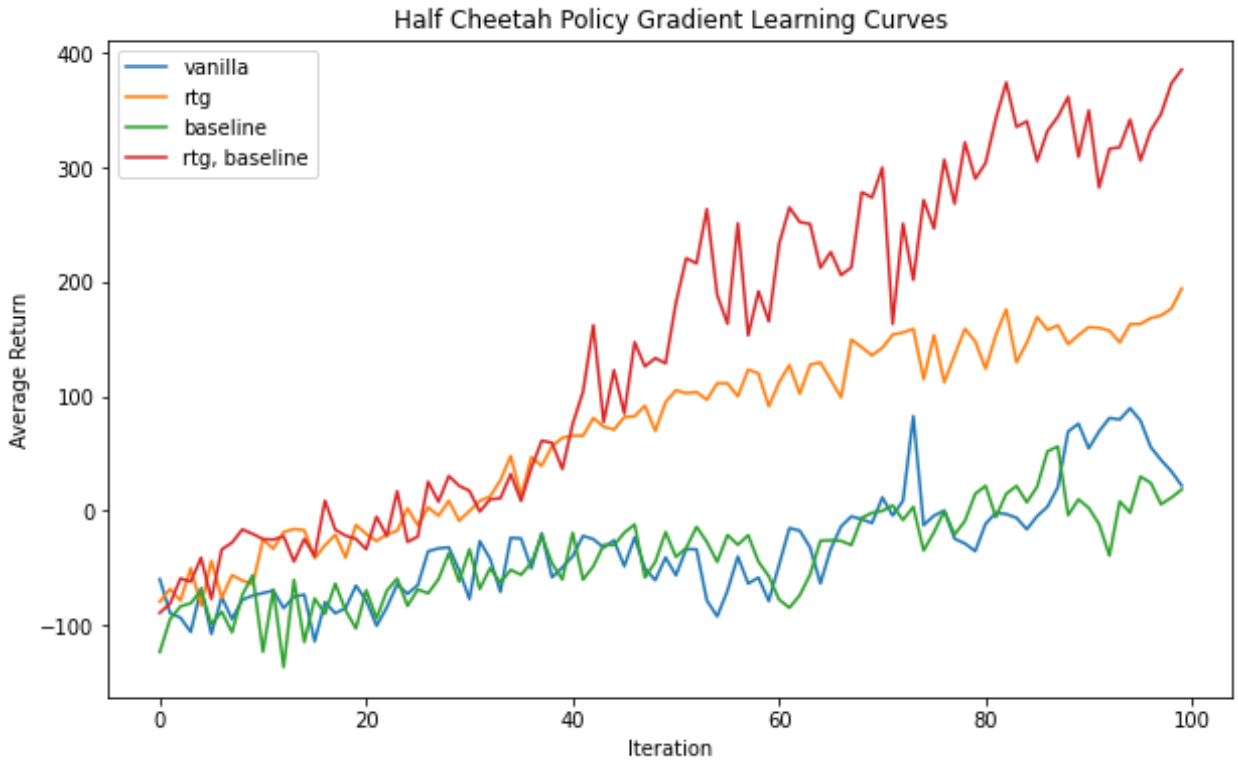

Figure 6: Learning curves for various policy gradient modifications in the HalfCheetah Environment. Batch size 50000, learning rate 0.02.

The policy gradient with reward-to-go and neural network baseline performed the best.

The following commands were used in this experiment:

```
python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.005 -rtg --nn_baseline \
--exp_name q4_search_b10000_lr0.005_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.01 -rtg --nn_baseline \
--exp_name q4_search_b10000_lr0.01_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.02 -rtg --nn_baseline \
--exp_name q4_search_b10000_lr0.02_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.005 -rtg --nn_baseline \
--exp_name q4_search_b30000_lr0.005_rtg_nnbaseline
```

```
python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.01 -rtg --nn_baseline \
--exp_name q4_search_b30000_lr0.01_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.02 -rtg --nn_baseline \
--exp_name q4_search_b30000_lr0.02_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 50000 -lr 0.005 -rtg --nn_baseline \
--exp_name q4_search_b50000_lr0.005_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 50000 -lr 0.01 -rtg --nn_baseline \
--exp_name q4_search_b50000_lr0.01_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b 50000 -lr 0.02 -rtg --nn_baseline \
--exp_name q4_search_b50000_lr0.02_rtg_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b <b*> -lr <r*> \
--exp_name q4_b50000_r0.02

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b <b*> -lr <r*> -rtg \
--exp_name q4_b50000_r0.02_rtg

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b <b*> -lr <r*> --nn_baseline \
--exp_name q4_b50000_r0.02_nnbaseline

python cs285/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
--discount 0.95 -n 100 -l 2 -s 32 -b <b*> -lr <r*> -rtg --nn_baseline \
--exp_name q4_b50000_r0.02_rtg_nnbaseline
```

Experiment 5 (HopperV4)

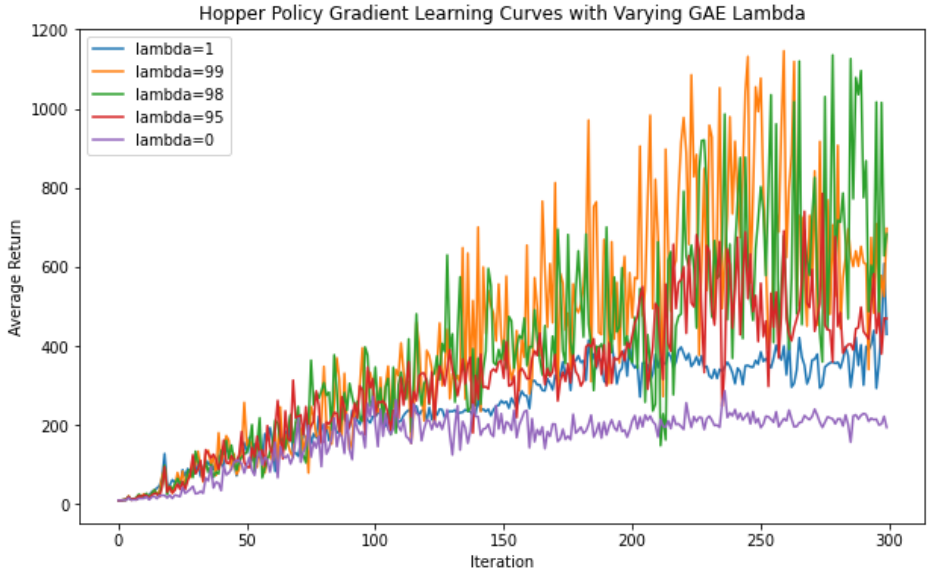

Figure 7: Learning curves for varying GAE lambda in the HopperV4 environment.

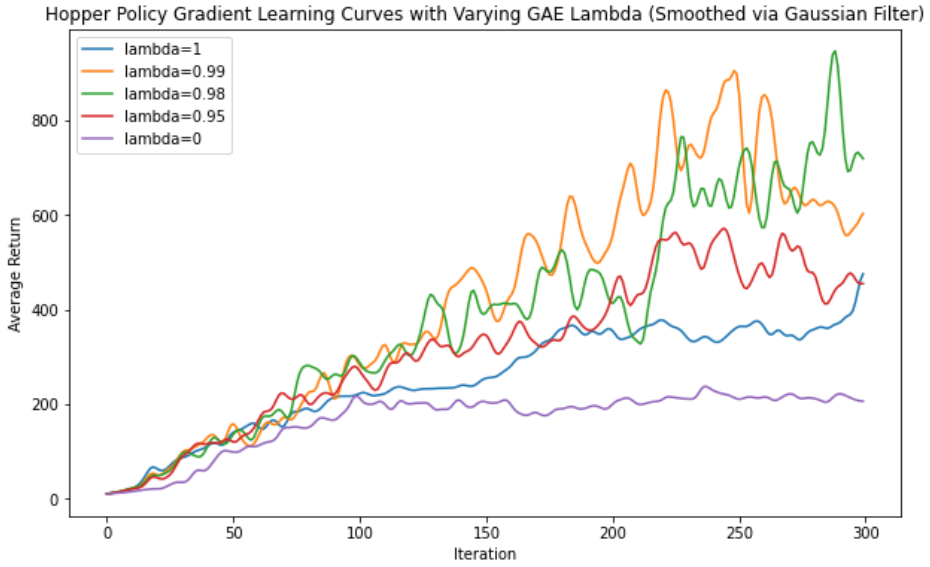

Figure 8: Learning curves for varying GAE lambda in the HopperV4 environment, smoothed via Gaussian filter with kernal standard deviation of 2.

Although GAE lambda values of 0.99, 0.98, and 0.95 improved performance, they seemed to induce a large amount of instability. A lambda of 0 had the worst peroformance, presumably because the next advantage is completely ignored.

The following commands were used in this experiment:

```
python cs285/scripts/run_hw2.py \
--env_name Hopper-v4 --ep_len 1000
--discount 0.99 -n 300 -l 2 -s 32 -b 2000 -lr 0.001 \
--reward_to_go --nn_baseline --action_noise_std 0.5 --gae_lambda 1 \
--exp_name q5_b2000_r0.001_lambda1

python cs285/scripts/run_hw2.py \
--env_name Hopper-v4 --ep_len 1000
--discount 0.99 -n 300 -l 2 -s 32 -b 2000 -lr 0.001 \
--reward_to_go --nn_baseline --action_noise_std 0.5 --gae_lambda 0.99 \
--exp_name q5_b2000_r0.001_lambda0.99

python cs285/scripts/run_hw2.py \
--env_name Hopper-v4 --ep_len 1000
--discount 0.99 -n 300 -l 2 -s 32 -b 2000 -lr 0.001 \
--reward_to_go --nn_baseline --action_noise_std 0.5 --gae_lambda 0.98 \
--exp_name q5_b2000_r0.001_lambda0.98

python cs285/scripts/run_hw2.py \
--env_name Hopper-v4 --ep_len 1000
--discount 0.99 -n 300 -l 2 -s 32 -b 2000 -lr 0.001 \
--reward_to_go --nn_baseline --action_noise_std 0.5 --gae_lambda 0.95 \
--exp_name q5_b2000_r0.001_lambda0.95

python cs285/scripts/run_hw2.py \
--env_name Hopper-v4 --ep_len 1000
--discount 0.99 -n 300 -l 2 -s 32 -b 2000 -lr 0.001 \
--reward_to_go --nn_baseline --action_noise_std 0.5 --gae_lambda 0 \
--exp_name q5_b2000_r0.001_lambda0
```