CS 285 HW 2

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedExp 1

1. Which value estimator has better performance without advantage-standardization: the trajectory-

centric one, or the one using reward-to-go?

* Answer: The reward-to-go one has better performance, as most obviously shown in the right plot with a larger batch.

1. Did advantage standardization help?

* Answer: No. It makes the training worse.

1. Did the batch size make an impact?

* Answer: Yes. Large batch size gives more stable training and faster convergence.

Command line for Exp. 1:

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

A screenshot of a cell phone

Description automatically generatedExp 2

Command line for Exp. 2:

python cs285/scripts/run\_hw2.py --env\_name InvertedPendulum-v2 \

--ep\_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b 5000 -lr 0.008 -rtg \

--exp\_name q2\_b5000\_r0.008

A screenshot of a cell phone

Description automatically generatedExp 3

Command line for Exp. 3:

python cs285/scripts/run\_hw2.py --env\_name InvertedPendulum-v2 \

--ep\_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b 5000 -lr 0.008 -rtg \

--exp\_name q2\_b5000\_r0.008

Exp 4

A screenshot of a cell phone

Description automatically generated

* Describe in words how the batch size and learning rate affected task performance.

Answer: Larger batch size generally gives better return than smaller batch size, for example, both 30000 and 50000 batch outperform the 10000 batch size. A greater learning rate also gives better return, for example, with batch size 30000 batch, lr=0.02 outperforms lr=0.01 and etc.

As can be readily seen, the best batch size is 30000 and the best learning rate is 0.02. Plug them in for the remaining 4 runs:

A screenshot of a cell phone

Description automatically generated