Exp1:

Small-scale:

Learning\_rate\_0.005, n\_layers:2, size: 64

Chart, histogram

Description automatically generated

Small Batch Result

Chart, histogram

Description automatically generated

Large Batch Result

Questions:

1. Which value estimator has better performance without advantage-standardization: the trajectory- centric one, or the one using reward-to-go?

Value estimator has better performance by the one using reward-to-go. Training procedure has less variance times and faster to converge.

1. Did advantage standardization help?

Yes. Advantage standardization helps the estimator being more stable. Once the estimator reaches to the maximum, has lower variance to go back and forth.

1. Did the batch size make an impact?

Observed from those two figures, seems small batches are more difficult to train to converge. For all three different conditions.

Exp2:

Min\_batch\_size: 900

Max\_learning\_rate: 0.05

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

--ep\_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b <900> -lr <0.05> -rtg \

--exp\_name q2\_b<900>\_r<0.05>\_50\_iter

Chart, histogram

Description automatically generated

Exp3:

Command: 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

Chart, line chart

Description automatically generated

Exp4:

For 4a)

As the picture shows the best performance happened where b = 30000 and lr = 0.02, the purple line.

So for learning rate, small learning rate performs worse in this experiment, lr = 0.005 is the worst regardless of the batch size.

And for batch\_size, there is no guarantee that for larger batch size the performance will be better, but generally speaking for this experiment, batch size with 30000 and 50000 will performs better than 10000

Chart

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Figure for q4\_b

Exp5:

Chart, histogram

Description automatically generated

Figure with lambda setting

When lambda = 0.95, the network training process has more variance than using lambda=0.99/1. When lambda = 0, which leads the A\_pi(st+1,at+1) times 0; so at that point the network are only using the current step of reward and estimator. And the result is not good.