**Deep RL Assignment 1: Imitation Learning**

**Kourosh Hakhamaneshi**

2. Behavioral Cloning

2.2)

The following settings were used during training for all tasks:

NN\_model: 2 hidden layers with 128 and 64 neurons each, respectively.

**Input data normalization** was also applied

Learning rate: Decaying learning rate function: **0.005\*(0.5)^(global\_step/10000) with staircase=True.**

For training+validation set **20 rollouts** of the expert policy has been executed (20000 observation, action pairs) and **200 epochs** with a **batch size of 256** were the settings for training.

Validation set size was 0.1 times the size of the training set.

Surprisingly the bc agent has a comparable result to the expert on all tasks. Here is a table of comparison for 20 rollouts on each task.

Table1.1 Comparison of Expert vs. BC agent on all the tasks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Ant-v2 | HalfCheetah-v2 | Hopper-v2 | Humanoid-v2 | Reacher-v2 | Walker2d-v2 |
| Expert | avg | 4867 | 4162 | 3778 | 10391 | -4.59 | 5521 |
| std | 114.8 | 99 | 3.6 | 57 | 1.66 | 70.6 |
| BC | avg | 4801 | 4127.1 | 3778 | 10399 | -4 | 5508 |
| std | 111 | 80.14 | 2.7 | 49.26 | 1.74 | 80 |

2.3)

The same settings as above were used, except for the fact that num\_epochs were swept.

**num\_epochs** was chosen as a variable here, because it basically determines how good the nn is getting trained, especially when we have decaying learning rate, where the learning rate decreases stepwise.

The take away is that if not trained enough the standard deviation of bc policy is going to be large. But with larger number of training steps, the standard deviation gets smaller. This could actually be seen when we train on a very large number of epoch (i.e. 200)

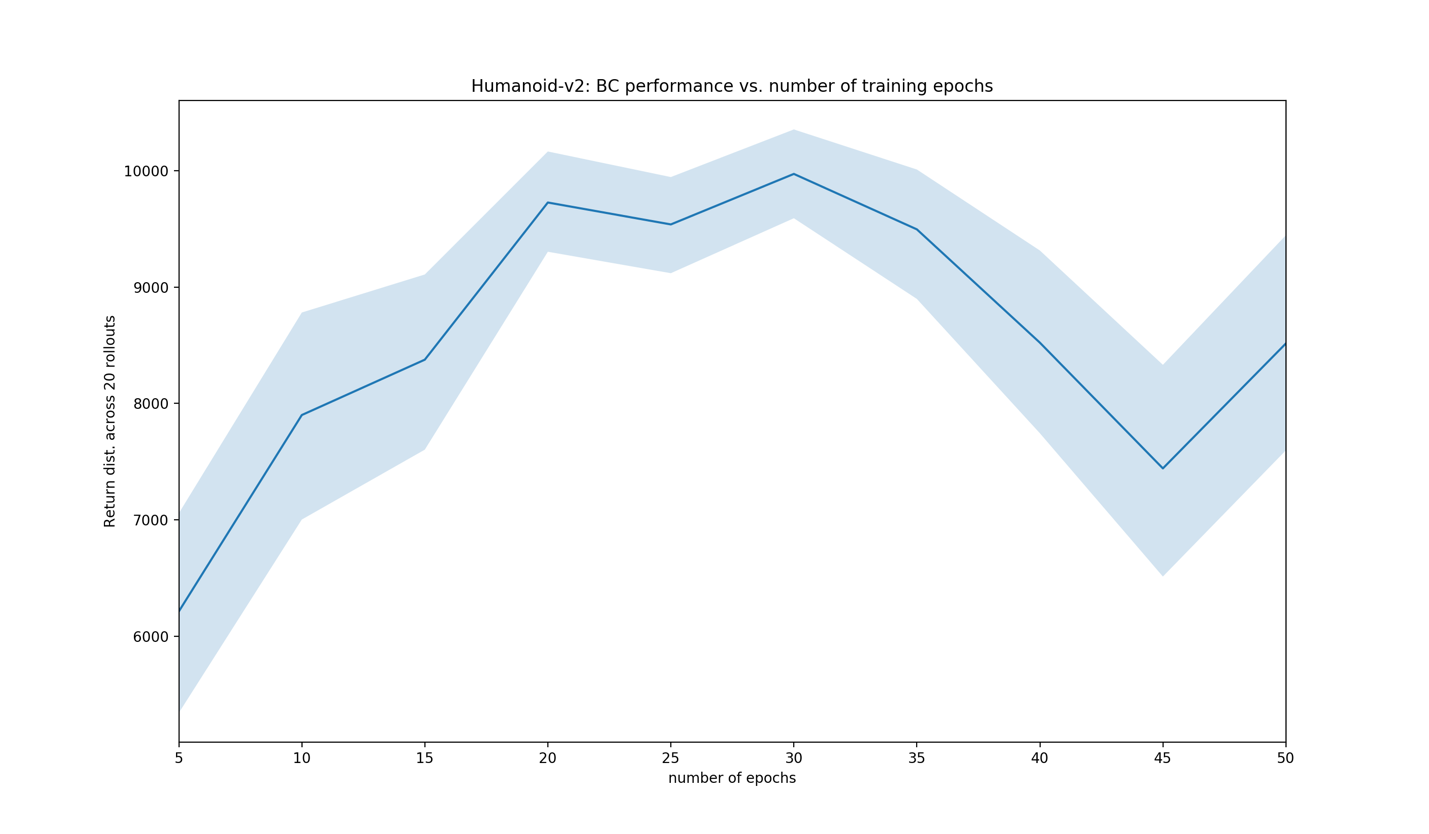


Figure 2.**Error! No text of specified style in document.**‑ Epochs=range(5,50,step=5)

3.2) The comparison on Humanoid task was made, The setting for the nn is the same as previous parts.

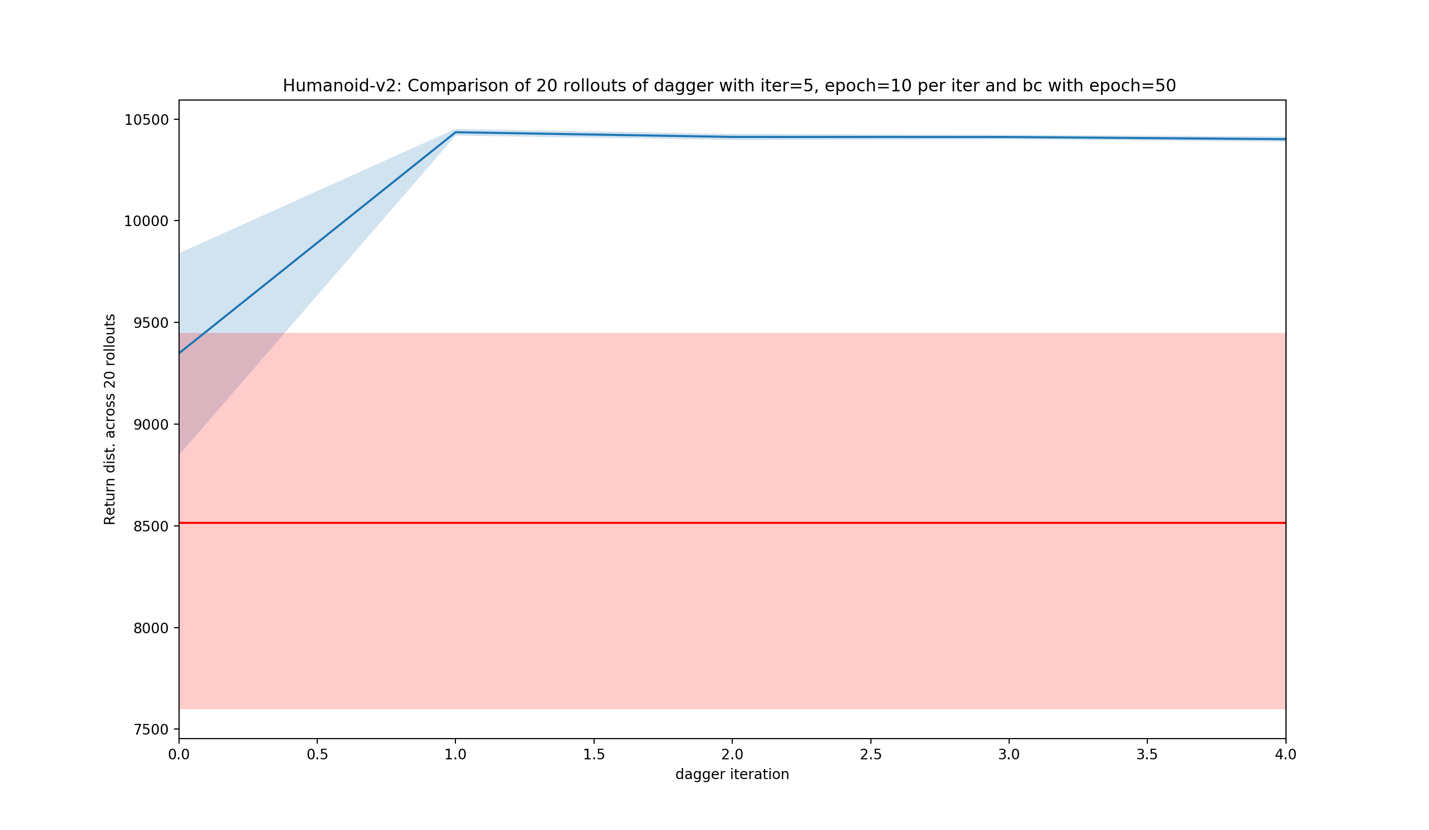


Figure 2. Blue is dagger performance accross each iteration/aggregation and the red curve is the bc policy performance obtained from 2.3 with epoch=50