IFT6163 Assignment 1: Imitation Learning

Simon Chamorro

February 14, 2022

1 Behavioral Cloning

1.1 Behavior Cloning Evaluation

Environment	Expert Avg. Reward	Behavior Cloning Avg. Reward	Behavior Cloning Std. Reward	Behavior Cloning Exp. Performance
HalfCheetah	4205.78	3726.17	76.04	**88.6%**
Ant	4713.65	4094.08	96.17	**86.9%**
Hopper	3772.67	743.27	270.65	19.7%
Walker2d	5566.85	222.45	17.88	4.0%
Humanoid	10344.52	331.65	81.21	3.2%

Table 1: **Behavior Cloning Results**. This table presents the results obtained by behavior cloning agents on various environments. We can see that the agent reaches a performance above 30% of the expert performance in HalfCheetah and Ant (in bold). Qualitatively, we can also see that the agent learns useful locomotion policies only in these two environments. Defalut parameters were used for these experiments. For all experiments, 2000 samples of expert data are available. The replay buffer has a size of 100000. The train batch size is 100 and the policy is trained for 1000 steps with a learning rate of 0.005. The policy network has two layers, and a hidden dimension of 64. The maximum episode length was set to 1000 to match the length of the expert data training episodes. Each policy was evaluated over 10000 steps for a minimum of 10 runs.

Name: Simon Chamorro

1.2 Hyperparameter Analysis : Training Steps

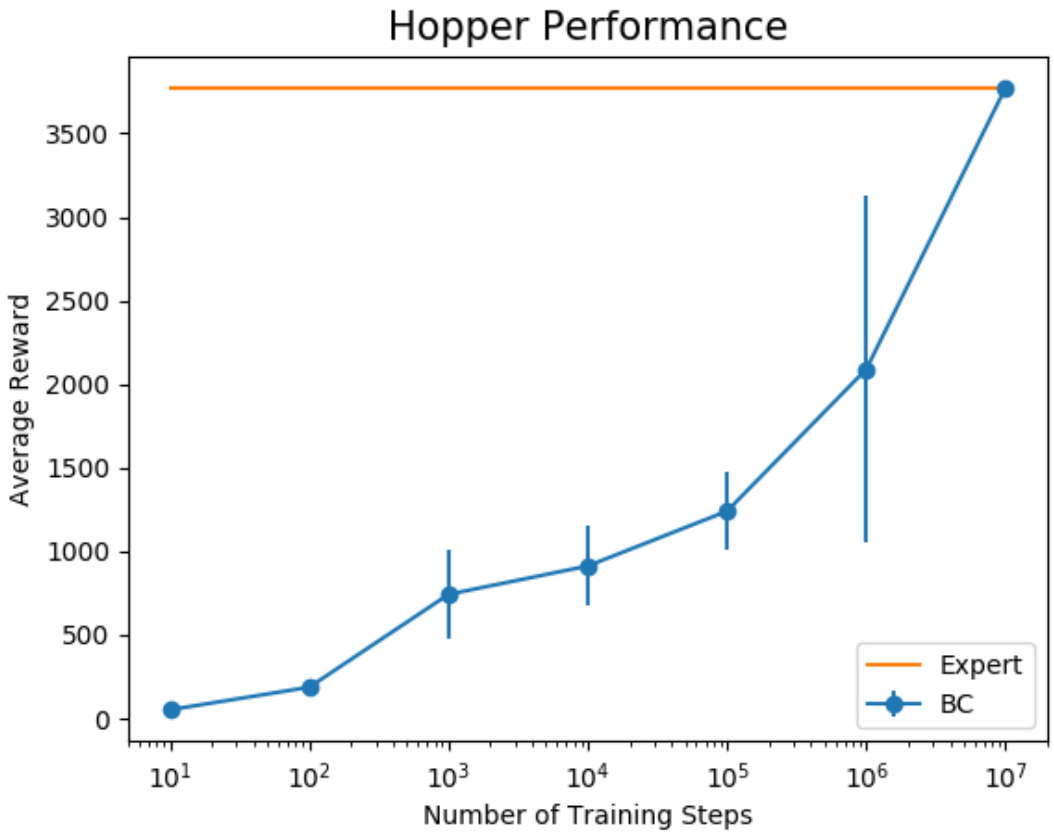
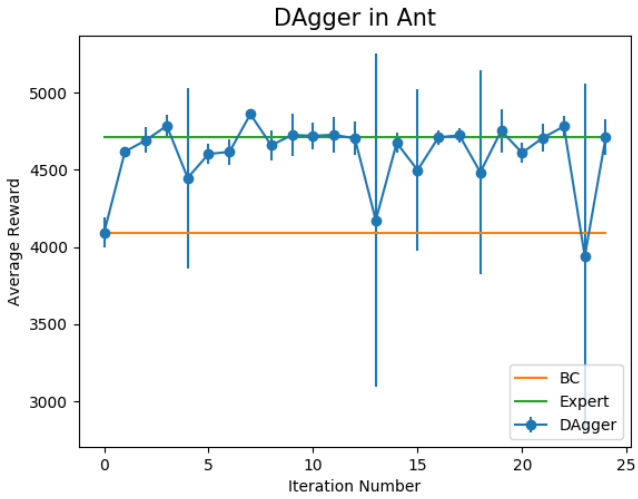


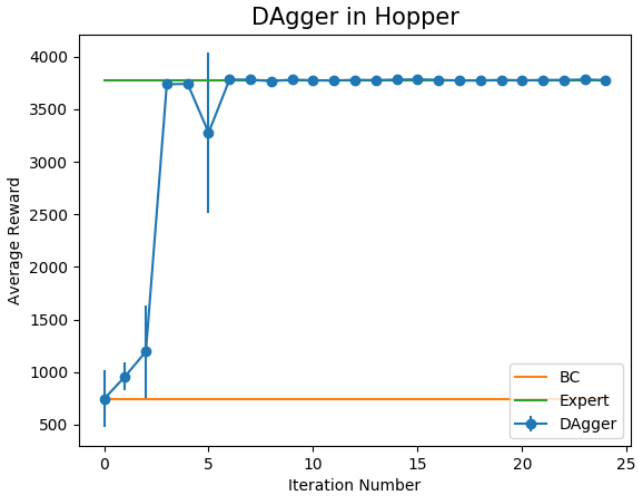
Figure 1: **Training Steps.** In this graph, we present the performance of behavior cloning in the Hopper environment with a varying amount of training steps. The parameters used are the following: A replay buffer of 100000 samples, 2000 samples of initial expert data, a train batch size of 100 and a learning rate of 0.005. The policy network has two layers, and a hidden dimension of 64. The maximum episode length is 1000. The policy was evaluated over 5000 steps for a minimum of 5 runs. The only varying parameter is the number of training steps. In fact, it is interesting to see that even if we do not add additional data, the performance increases with more training iterations and eventually reaches expert performance. This improvement, however, is very slow. After training for 1M steps, the performance is only around 70% expert performance, and it takes 10M steps to reach it. On the other hand, in the next section, we see that expert performance is reached after only 10 thousand training steps using DAgger. This experiment highlights the importance of the quality of training data. The on-policy data from DAgger is much more valuable and makes the learning process more data-efficient.

Name: Simon Chamorro

2 DAgger



(a) Ant



(b) Hopper

Figure 2: **DAgger Performance.** In this graph, we present the results of the DAgger algorithm in the Ant and Hopper environments. The parameters used are the following: For both experiments, the replay buffer has a size of 100000, 2000 samples of expert data are available and 1000 samples are collected in the environment per iteration, the train batch size is 100 and the policy is trained for 1000 steps with a learning rate of 0.005 at every iteration. The policy network has two layers, and a hidden dimension of 64. The maximum episode length was set to 1000 to match the length of the expert data training episodes. Each policy was evaluated over 10000 steps for a minimum of 10 runs at every iteration. Iteration 0 corresponds to the initial training on expert data, then DAgger is performed for 24 iterations.