Report HW1 16831

Christopher Klammer

September 21, 2022

1 Behavior Cloning

1.1 Part 1: Comparing with Another Environment

Constants:

1. Seed: 6868
2. Train Batch Size: 100
3. Evaluation Batch Size: 10000
4. Gradient Steps per Iteration: 1000
5. Episode Length: 1000
6. Network Size: 64
7. Number of Layers: 2
8. Learning Rate: 5e-3
9. Iterations: 1

Environment Task	Expert Reward	Policy Return Mean	Policy Return Std
Ant	4739.1000*	4684.7061	143.7537
Walker 2D	5347.1900*	302.5458	302.9838

* Error values obtained from piazza post rather than from Initial DataCollection AverageReturn

Table 1: OpenAI Gym Ant vs. Walker 2D for Behavior Cloning

1.2 Part 2: Assessing Effect of Network Size on Policy Return

I chose to pick network size as my hyperparameter of choice because I was curious if the size of the hidden layers (number of nodes/features in the hidden layers) would allow the learned policy to potentially be more complex. I chose run this on the Walker2D environment because I think it is more complex. Likewise, a more challenging environment states may require additional features to learn difficult tasks. Something like walking on four legs obviously does not require as much balance as walking on two legs. I think part of the reason why the ant performed better than the walker was because it was a simpler environment. In turn, the Walker2D environment is likely more complicated in that it needs to recover when balance is temporarily lost, creating a wide array of states where specialized actions may be required. Ultimately, adding more nodes into the hidden layers at each level may allow us to capture more complex relationship.

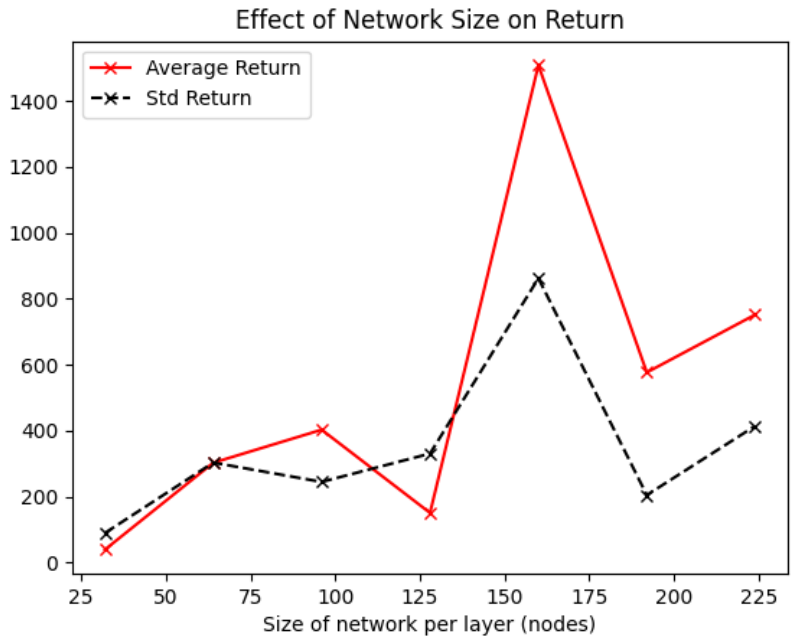

Figure 1: Variation of the network size hyperparameter for the Walker2D environment

After running the experiment, we can make some observations. In general, yes, the return increases as the size of the network increases with a peak at a network size of 160. However, we can also see the standard deviation increases as well in an almost linear fashion. Likely, there is one or two trajectories out

of the ten or so in evaluation that perform immensely better than the rest. Nonetheless, this is evidently not purely luck. The increase is consistent enough and the return is much higher on average, consistently outperforming when the network size was just 32. In future work, this could be combined with changing the amount of hidden layers to further optimize performance

2 DAgger

Constants:

1. Seed: 6868
2. Train Batch Size: 100
3. Evaluation Batch Size: 10000
4. Gradient Steps per Iteration: 1000
5. Episode Length: 1000
6. Network Size: 64
7. Number of Layers: 2
8. Learning Rate: 5e-3
9. Iterations: 10

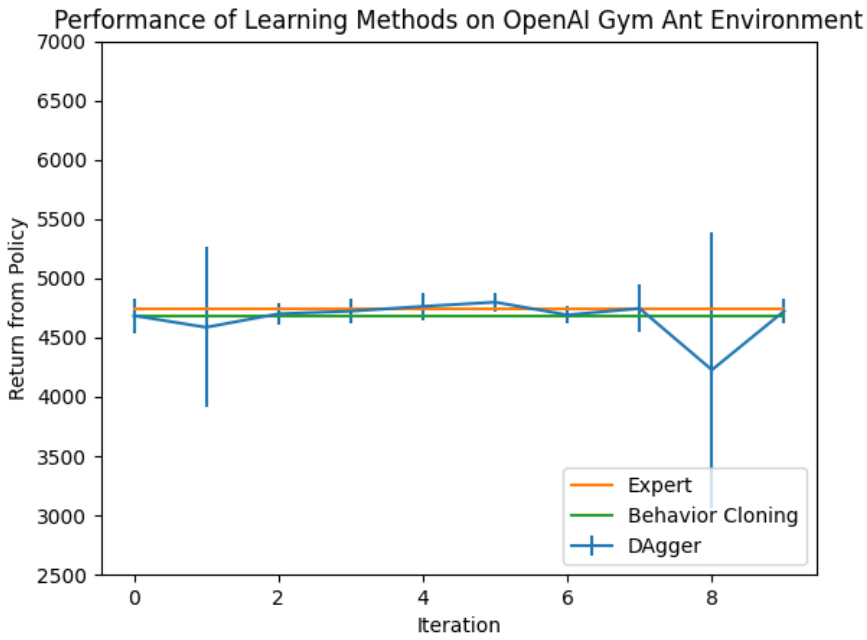

Figure 2: DAgger, Behavior Cloning, and expert performance on the Ant environment

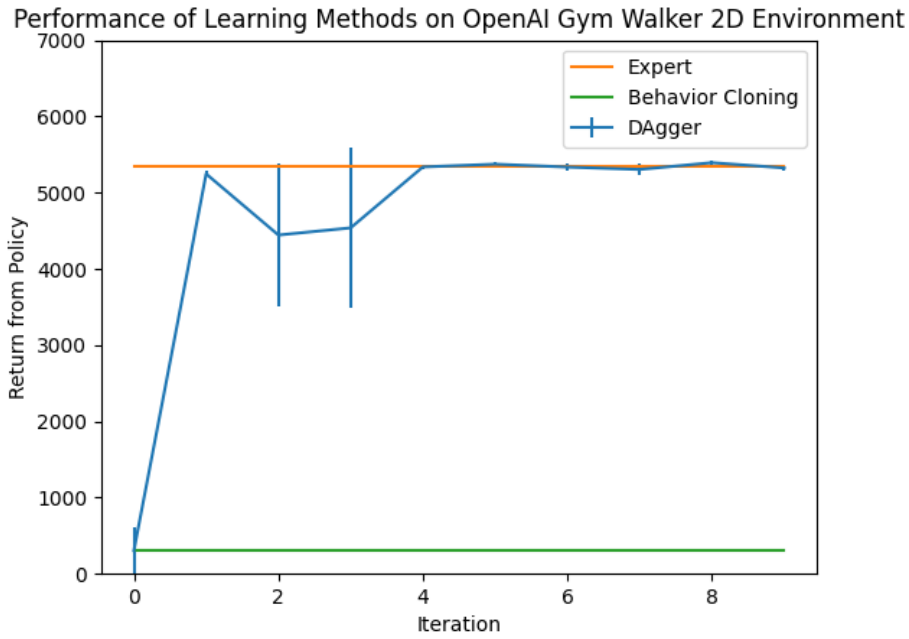

Figure 3: DAgger, Behavior Cloning, and expert performance on the Walker 2D environment

2.1 Analysis

The performance of the Ant environment was a tad underwhelming. Since behavior cloning performed extremely well initially, DAgger gave only a small nudge to performance. However, there was a high variance at a few points, making some points a bit skewed. Yet, on average, the return matches that of the policy by iteration 10 and is still an upgrade over behavior cloning.

The performance of the Walker2D environment was a bit more as expected. The behavior cloning in the first section did not perform as well on this environment. The Walker2D environment converged towards the expert as more iterations continued. Additionally, unlike the Ant environment, the standard deviation stabilized as well, meaning that the policy consistently performed well over all the different trajectories. Here, DAgger offered immense performance improvements over the baseline behavior cloning and by the end nearly matches the performance of the expert.

3 Commands

```
Part 1 BC Ant:

 --expert_policy_file hw1/rob831/policies/experts/Ant.pkl --env_name Ant-v4
 --exp_name bc_ant --n_iter 1
 --expert_data hw1/rob831/expert_data/expert_data_Ant-v4.pkl --video_log_freq -1
 --seed 6868 --eval_batch_size 10000


Part 1 BC Walker:

--expert_policy_file hw1/rob831/policies/experts/Walker2d.pkl --env_name Walker2d-v4
--exp_name bc_walker --n_iter 1
--expert_data hw1/rob831/expert_data/expert_data_Walker2d-v4.pkl --video_log_freq -1
--seed 6868 --eval_batch_size 10000

Part 2 DAgger Ant:

--expert_policy_file hw1/rob831/policies/experts/Ant.pkl --env_name Ant-v4
--exp_name dagger_ant --n_iter 10 --do_dagger
--expert_data hw1/rob831/expert_data/expert_data_Ant-v4.pkl --video_log_freq -1
--seed 6868 --eval_batch_size 10000

Part 2 DAgger Walker:

--expert_policy_file hw1/rob831/policies/experts/Walker2d.pkl --env_name Walker2d-v4
--exp_name bc_walker --n_iter 10 --do_dagger
--expert_data hw1/rob831/expert_data/expert_data_Walker2d-v4.pkl --video_log_freq -1
--seed 6868 --eval_batch_size 10000
```