1 Behavioral Cloning

1.2

Training with the default settings set in `run_hw1.py`, except for `eval_batch_size=10000` and `ep_len=1000` (so approximately 10 trajectories), I arrived at the following results:

Environment	Expert Mean	BC Mean	BC std	% of Expert
Ant-v2	4713.653	4374.649	1108.576	92.8%
Humanoid-v2	10344.51	280.917	12.222	2.71%

The agent was trained with only trained for 1000 steps on a network with 2 hidden layers, each of size 64. From the table, we see that the Ant task achieved 92% of the expert performance, while the on the Humanoid task, the agent did not reach at least 30% of the expert's performance, instead reaching only 2.71% of the expert performance.

Here are the commands that could reproduce my results:

Listing 1: Ant task

```
python cs285\scripts\run_hw1.py --expert_policy_file cs285\policies\experts\Ant.
    pkl --env_name Ant-v2 --exp_name bc_ant --n_iter 1 --expert_data cs285\
    expert_data\expert_data_Ant-v2.pkl --video_log_freq -1 --ep_len 1000 --
    eval_batch_size 10000
```

Listing 2: Humanoid task

```
python cs285\scripts\run_hw1.py --expert_policy_file cs285\policies\experts\
    Humanoid.pkl --env_name Humanoid-v2 --exp_name bc_human --n_iter 1 --
    expert_data cs285\expert_data\expert_data_Humanoid-v2.pkl --video_log_freq -1
    --ep_len 1000 --eval_batch_size 10000
```

1.3

For this, I decided touse the Humanoid task and vary the amount learning rate of the MLP policy between 0.05 and 0.0001 on a log scale. Here's the command I used:

```
python cs285\scripts\run_hw1.py --expert_policy_file cs285\policies\experts\
    Humanoid.pkl --env_name Humanoid-v2 --exp_name bc_human --n_iter 1 --
    expert_data cs285\expert_data\expert_data_Humanoid-v2.pkl --video_log_freq -1
    --num_agent_train_steps_per_iter 50000 --lr <learning_rate>
```

where `<learning_rate>` is varied. Figure 1 below shows the results:

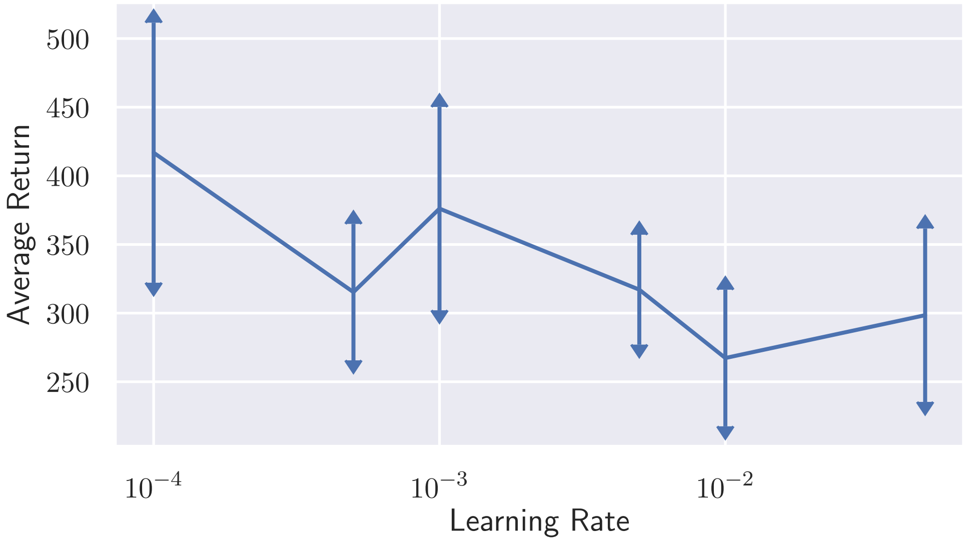

Figure 1: The agent was trained for 50,000 steps on the expert data. The policy network had 2 hidden layers, each of size 64. I chose to vary the learning rate because with deep neural networks, the learning rate is often one of the more important and impactful hyperparameters, so I thought there may be a similar case in behavioral cloning.