CS 285 Deep Reinforcement Learning
Fall 2022 Rahul Shah

Homework 1

Report

1. **Behavioral Cloning**

Results:

Q1.1) Done.

Q1.2)

Environment	Eval_AverageReturn	Eval_StdReturn
Ant	4669.8564453125	117.36156463623047
Walker2d	660.2849731445312	517.1852416992188

Table 1: We can see drastically different results for the following two environment, with the models being structured similarly in both: both are MLP policies with a depth of $n = 2$ hidden layers, a width of 64, a learning rate of $lr = 5e - 3$, a `eval_batch_size=3000` (to make the standard deviation greater than 0) and finally only one iteration (Iteration 0) was executed, with 1000 gradient steps for training the agent, all for both.

Q1.3)

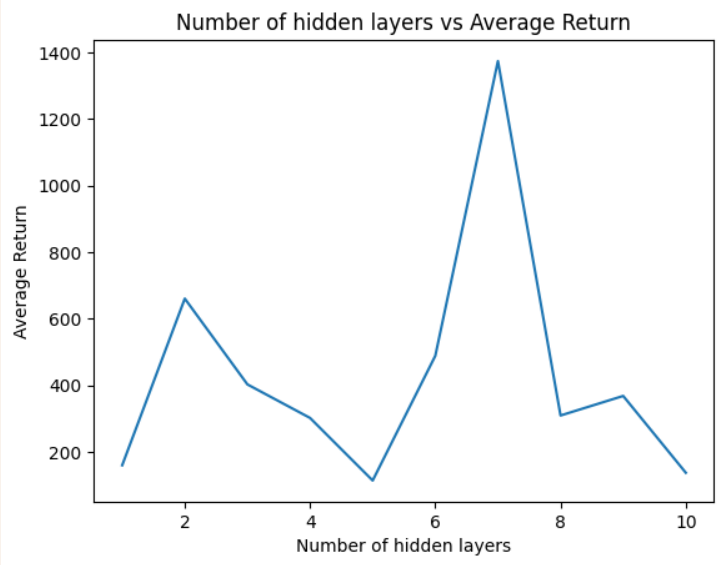

Figure 1: Here I tried varying the number of hidden layers in the model. We can note that increasing the number of layers from the original depth of 2 helps as the MLP can learn more complex behavior but eventually it overfits and fails to generalize when it has too many hidden layers. I chose this as I figured we could get better average returns with a higher depth, while not being too high.

2. DAgger

Run DAgger and report results on the two tasks you tested previously with behavioral cloning (i.e., Ant + another environment). Report your results in the form of a learning curve, plotting the number of DAgger iterations vs. the policy's mean return, with error bars to show the standard deviation. Include the performance of the expert policy and the behavioral cloning agent on the same plot (as horizontal lines that go across the plot). In the caption, state which task you used, and any details regarding network architecture, amount of data, etc. (as in the previous section).

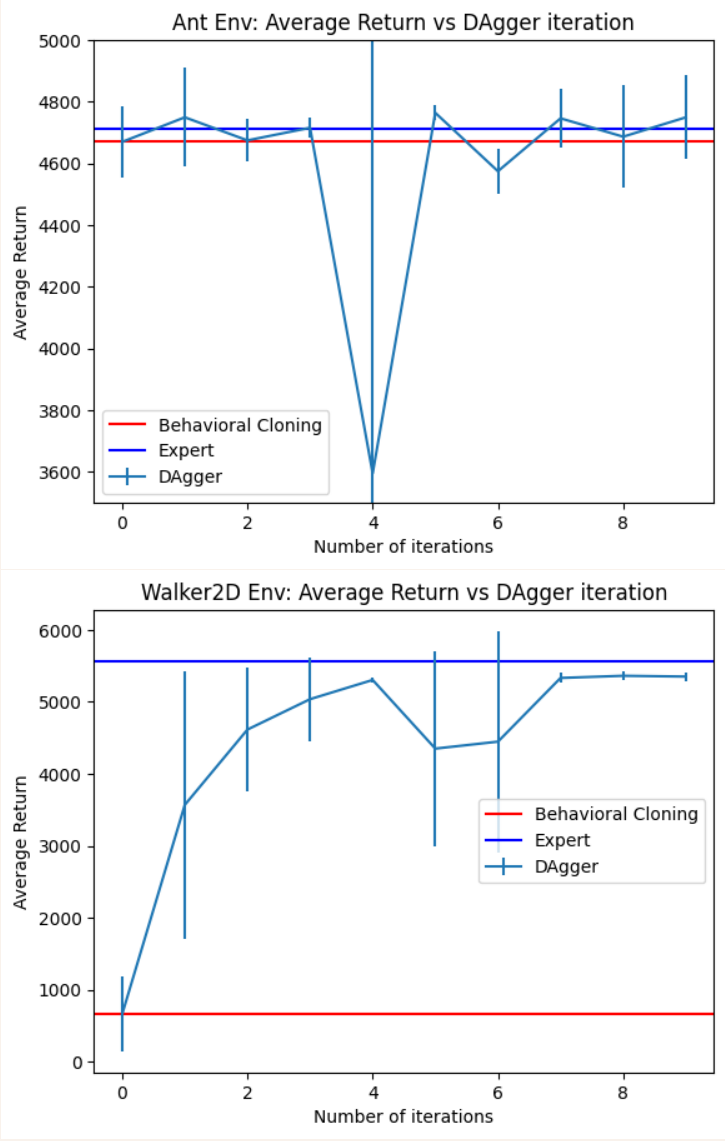

Figure 2: Here I ran Ant and Walker2D with `eval_batch_size=3000` and the rest as default: both are MLP policies with a depth of $n = 2$ hidden layers, a width of 64, a learning rate of $lr = 5e - 3$, each with 10 iterations, and with 1000 gradient steps for training the agent for both.