CS224R Spring 2023 Homework 1
Imitation Learning

Due 4/19/2023

SUNet ID: **ishancs** (**06681175**)
Name: **Ishan Sabane**
Collaborators: None

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1: Behavior Cloning

1. Run behavioral cloning (BC) and report results on two tasks: the Ant environment, where a behavioral cloning agent should achieve at least 30% of the performance of the expert, and one environment of your choosing where it does not. A policy that achieves greater than 30% of the expert on the Ant task will receive full credit on the autograder. Here is how you can run the Ant task:

```
python cs224r/scripts/run_hw1.py \
    --expert_policy_file cs224r/policies/experts/Ant.pkl \
    --env_name Ant-v4 --exp_name bc_ant --n_iter 1 \
    --expert_data cs224r/expert_data/expert_data_Ant-v4.pkl \
    --video_log_freq -1
```

   When providing results, report the mean and standard deviation of your policy's return over multiple rollouts in a table, and state which task was used. When comparing one that is working versus one that is not working, be sure to set up a fair comparison in terms of network size, amount of data, and number of training iterations. Provide these details (and any others you feel are appropriate) in the table caption.

Environment	eval_batch_size	Mean Return	Std Return	Network Size (layers)	Layer Size
Ant	5000	4190.195	62.663	2	64
Walker2d	5000	377.17	449.91	2	64

Table 1: The above comparison between the Ant and the Walker2d environment uses the default training settings. Both environments have the same amount of expert data available for training. The neural network is made up of two hidden layers each with 64 nodes. The learning rate is 5e-3 for training the BC agent. The evaluation batch size is increased as mentioned in the handout to 5000 to get average return across multiple rollouts.

Why does Behavior Cloning fail in the Walker2d environment?

According to me, the reason behind the BC agent not performing well in the Walker2d environment is because of the complex dynamics of the environment itself. Since BC uses just the expert data, during evaluation phase, it might not be able to understand observations which are not seen in the training dataset. This leads to suboptimal actions which affect the next state and lead to errors in the next state.

Such states can easily drift away from the expert data distribution. Moreover, sequential sub-optimal actions lead to compounding of errors. This leads to "co-variate shift" where the state visited by the learned policy are not similar to expert policy thereby reducing the returns. Hence the BC agent gives a poor performance in the Walker2d environment.

2. Experiment with one set of hyper-parameters that affects the performance of the behavioral cloning agent, such as the amount of training steps, the amount of expert data provided, or something that you come up with yourself. For one of the tasks used in the previous question, show a graph of how the BC agent's performance varies with the value of this hyperparameter. State the hyperparameter and a brief rationale for why you chose it.

   Hyper-parameter Chosen: Number of Hidden layers of the Neural network.

   (Note: Each Additional Layer corrresponds to a Linear Layer with 64 Nodes followed by the default activation function (tanh))

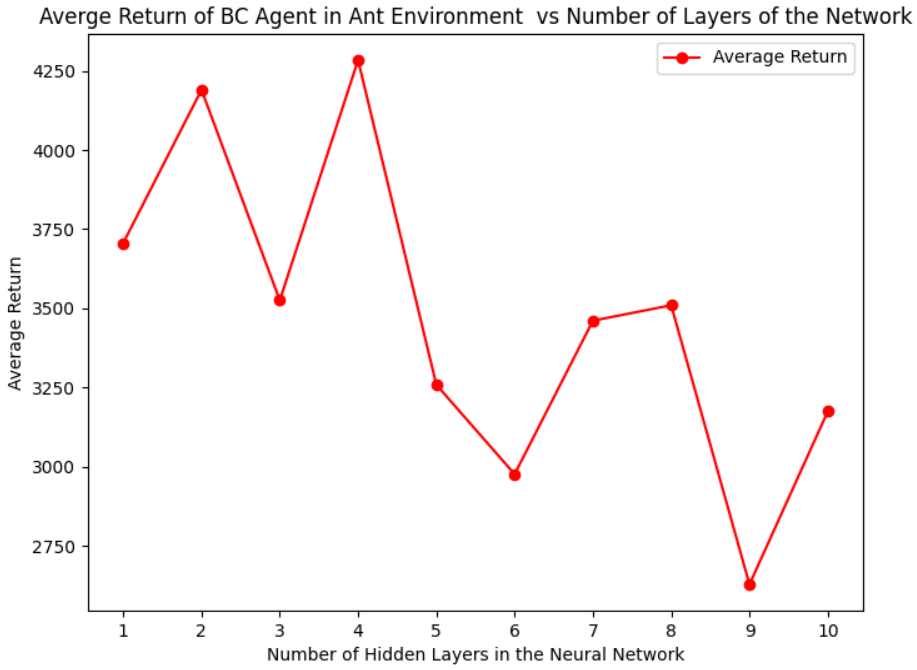

Figure 1: Variation of BC Agent Performance in Ant environment with respect to the number of hidden layers in the neural network each with 64 nodes.

   Rationale The Behaviour Cloning Agent is trained on the expert dataset which is limited in size, and hence in an effort to improve the performance of the model, we want to optimze the complexity of the neural network such that it does not under or overfit the data.

   Observation While increasing the number of layers will eventually over-fit the training data as the model complexity increases, we are able to see that the model gives the best performance with 4 layers. This is useful for hyper-paramter tuning.

Problem 2: DAgger

1. Once you've filled in all of the TODO commands, you should be able to run DAgger.

```
python cs224r/scripts/run_hw1.py \
  --expert_policy_file cs224r/policies/experts/Ant.pkl \
  --env_name Ant-v4 --exp_name dagger_ant --n_iter 10 \
  --do_dagger \
  --expert_data cs224r/expert_data/expert_data_Ant-v4.pkl \
  --video_log_freq -1
```

2. 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).
   Ant Environment

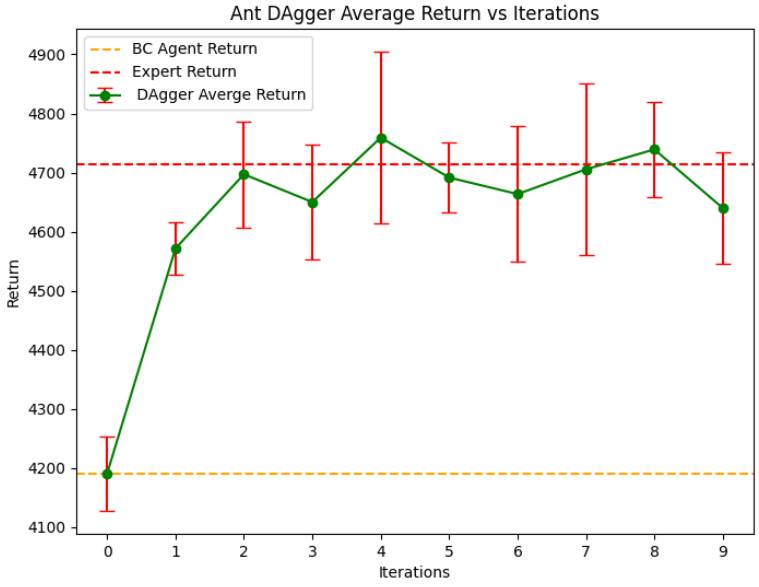

Figure 2: Average Return of DAgger on the **Ant Environment** as a function of the number of iterations. The above results are generated using the default training settings. The network consists of two hidden layers each having 64 nodes. The horizontal line on the top denotes the return from the expert policy while the line on the bottom denotes the return using Behaviour Cloning agent from question 1

Walker2d Environment

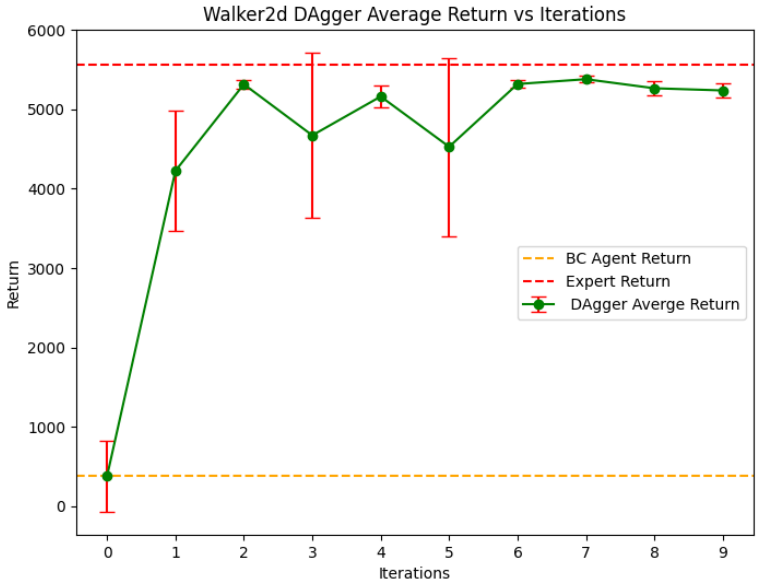

Figure 3: Return of DAgger on the **Walker2d environment** as a function of the number of iterations. The above results are generated using the default training settings. The network consists of two hidden layers each having 64 nodes. The horizontal line on the top denotes the return from the expert policy while the line on the bottom denotes the return using Behaviour Cloning agent from question 1

Observation
The performance of DAgger greatly improves over the performance of the BC agent trained on the Walker2d enivronment!