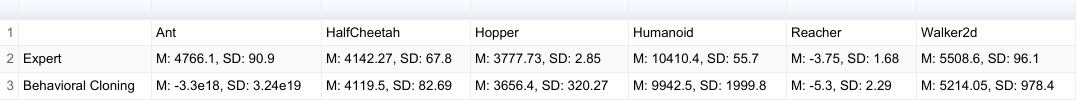
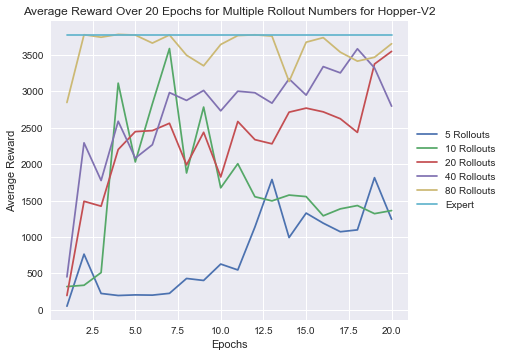
CS294-112 Homework 1: Imitation Learning

Note that for all Neural Networks, the x values are all normalized with the mean and standard deviation.

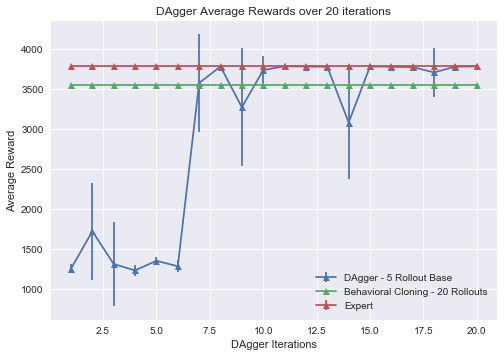
**Table 1:** This is the table for question 2.2. I used all the tasks. Ant-V2 had significantly worse results than the other tasks. These results were generated with a 3-layer NN created using Tensorflow with a dense hidden layer of size 64 units using the RELU activation function, an input layer, and an output layer. The results were gathered with 80 rollouts from the expert policy. I used a batch size of 64 and trained over 20 epochs. The x values were normalized with the mean and standard deviation before training and evaluating.



**Figure 1:** This is the graph for question 2.3. I used the Hopper-V2 task/environment. The hyperparameter I experimented with was the number of rollouts used to train the NN. I chose this hyperparameter since, during my implementation of the neural network, I tried using 5 rollouts and 10 rollouts and noticed drastically different results. I wanted to explore this further and improve my results. Also, since the amount of data is positively correlated with the accuracy of a deep learning network, I expected my results to improve as I increased the amount of data. I wanted to verify this hypothesis and confirm what Professor Levine stated in lecture.



**Figure 2:** This is the graph for question 3.2. I used the Hopper-V2 task with 10 epochs per iteration for 20 iterations of DAgger. The original data I used was the Hopper-V2 data with 5 rollouts. I used a 3-layer NN with a 64-unit dense hidden layer with a batch size of 64, an input layer, and an output layer. The x values were normalized with the mean and standard deviation before training and evaluating. The behavioral cloning data is for 20 rollouts. I used this because at the end of the 20 iterations of DAgger, the amount of data is closest to the 20 rollout behavioral cloning data (Around 20,000 observation-action pairs).



**Figure 3:** This is the graph for question 4.1. I used the Hopper-V2 task with 20 epochs per iteration and 20 rollouts from the expert policy. I used a 6-layer NN with an input layer, 2 dense 1024 unit layers, 2 dropout layers with p = 0.2, and an output layer. I used a batch size of 64. I expected this NN to drastically overfit the training data and not give very good results. Compared to the simple NN I used for question 1, this NN did significantly worse. This is mainly due to the fact that the NN overfit the observations in the training set. Therefore, if an observation comes that is not in the training set, the NN will have trouble determining the correct action to take.

