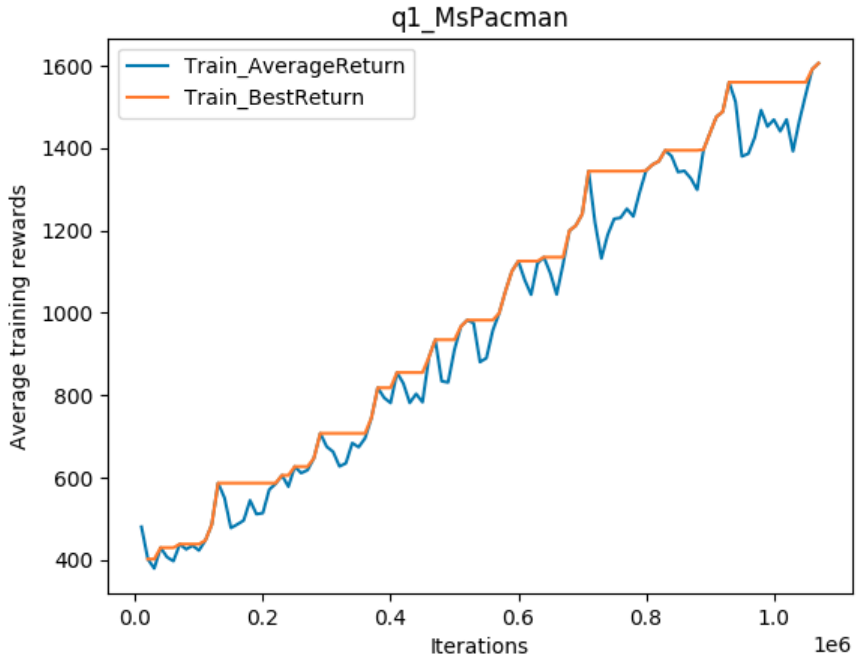
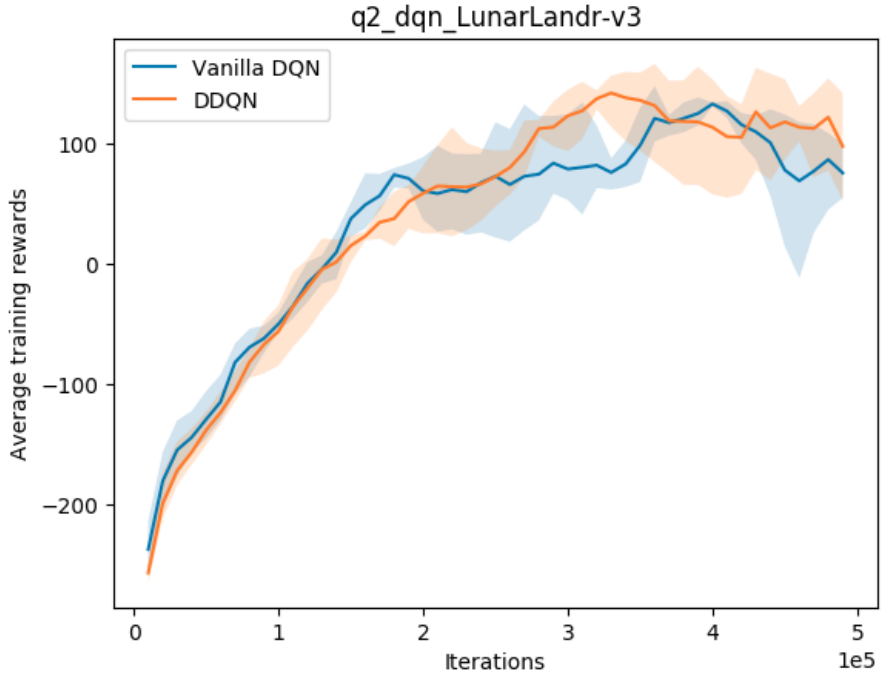
CS 285 HW 3

Q1



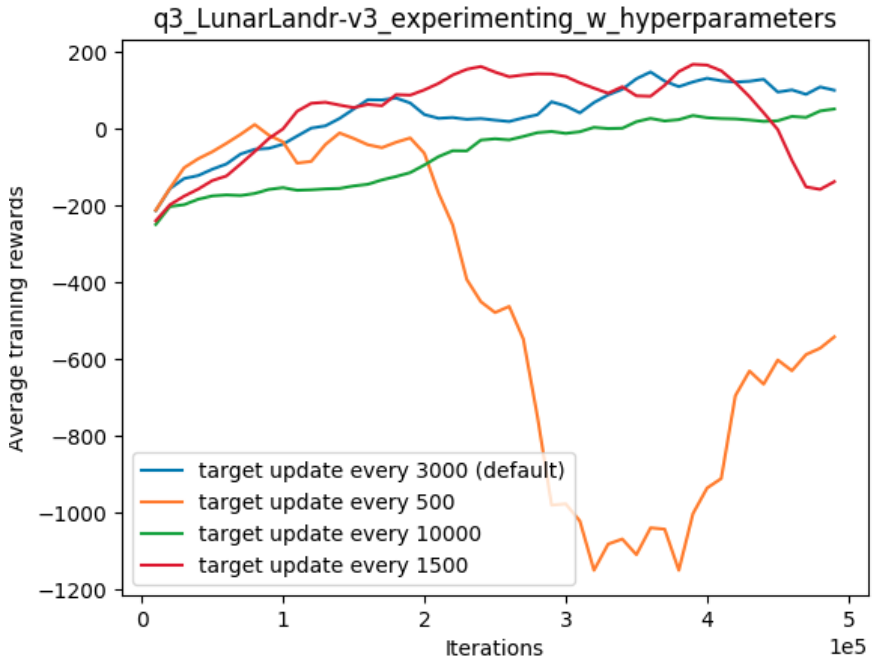
Q2



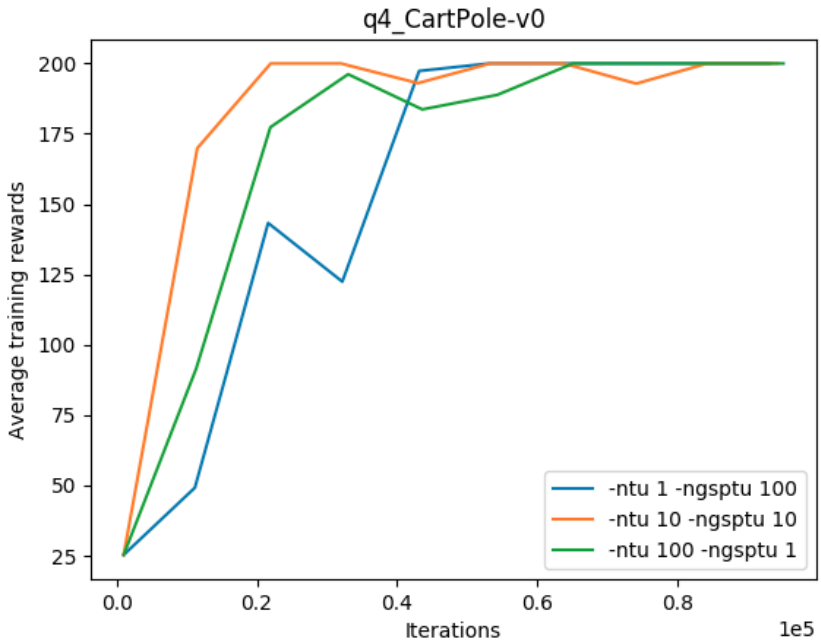
Q3

The hyperparameter I chose is the target update frequency. I set the target update frequency to be every 500, 1500, 3000 (default), 10000 time steps, respectively, as shown in the following plot. This parameter affects the performance of the DQN model by setting how frequently the learning value function approaches the target value function. A very small number (very high frequency) will make the target a 'moving target' so it becomes harder for the fitted Q function to approach the target. A very large number (very low frequency) is likely to make the learning slow since we are spending a lot of time steps approaching a fixed target before the target updates.

I find that a very small number (very high frequency) indeed makes the training much worse, as can be seen from the experiment with target updates every 500 steps. A very large number (very low frequency) indeed makes the training slow, as can be seen from the experiment with target updates every 10000 steps. Updating target at every 1500 steps seems to give the optimal training, which is smaller than the default value at 3000. It reaches 150 rewards much earlier than the default setting, though its rewards went down later on.

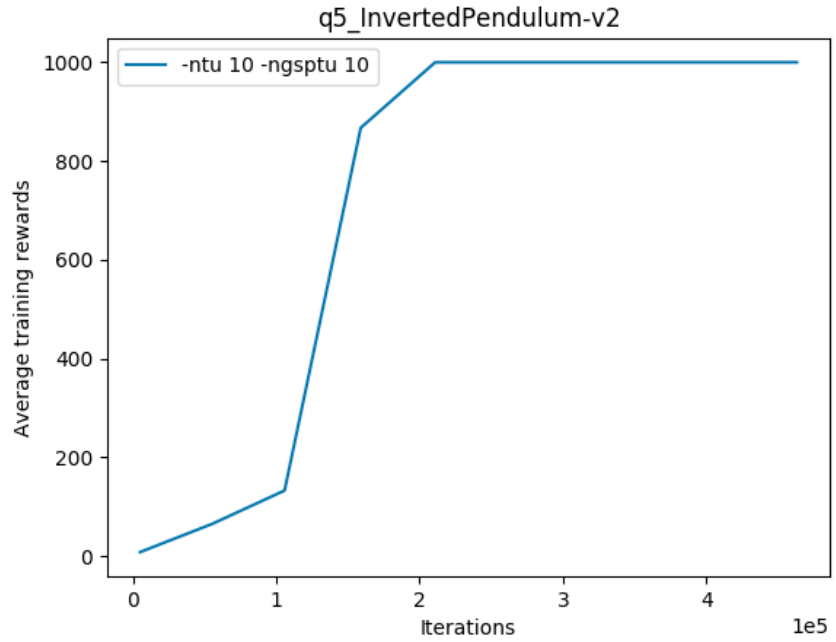


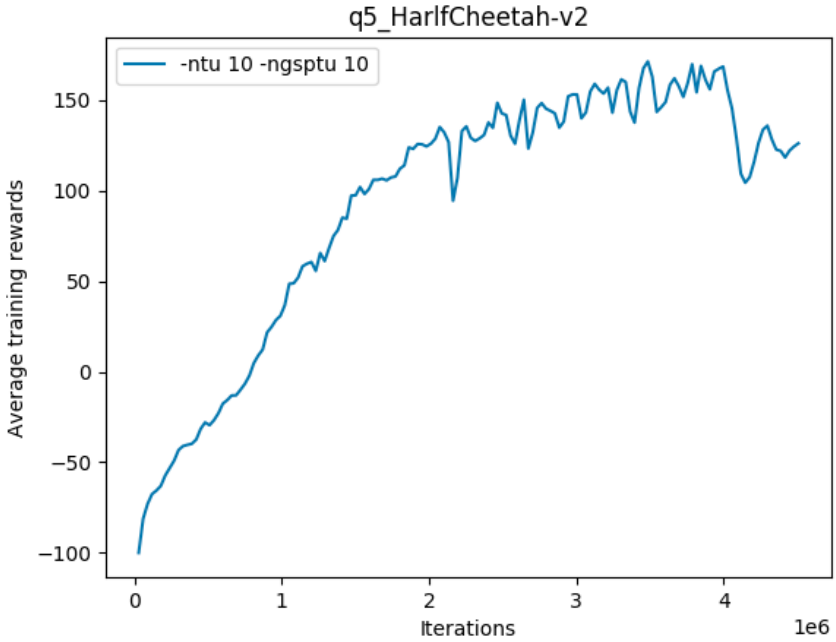
Q4



Caption: The best performance is with number of target updates ~= number of gradient steps per target update.

Q5


q5_InvertedPendulum-v2
-ntu 10 -ngsptu 10
Average training rewards
1000
800
600
400
200
0
0
1
2
3
4
Iterations
1e5


q5_HarlfCheetah-v2
-ntu 10 -ngsptu 10
Average training rewards
150
100
50
0
−50
−100
0
1
2
3
4
Iterations
1e6