CS 285 Set 3

Erich Liang

Due: 10/20/21

1 Question 1

Figure 1: Result of DQN agent ran on MsPacMan game.

2 Question 2

Figure 2: Result of DQN agent and DDQN agent ran on LunarLander. Each result was generated from averaging three random seeds.

3 Question 3

The hyperparameter varied was the lander exploration schedule. The default exploration schedule is as follows: the exploration schedule starts with $\epsilon = 1$, and this value of epsilon is decreased at a constant rate to reach $\epsilon = 0.02$ after 0.1 of the iterations have occurred; this is denoted by num_timesteps * 0.1 within the lander_exploration_schedule function. When this 0.1 multiplier is changed to a larger value, it will have the effect of extending the period of time that the agent explores.

- For 'q3_hparam0', default values were used (multiplier is 0.1 by default)
- For 'q3_hparam1', the multiplier for 'num_timesteps' was changed to 0.2
- For 'q3_hparam2', the multiplier for 'num_timesteps' was changed to 0.3
- For 'q3_hparam3', the multiplier for 'num_timesteps' was changed to 0.4

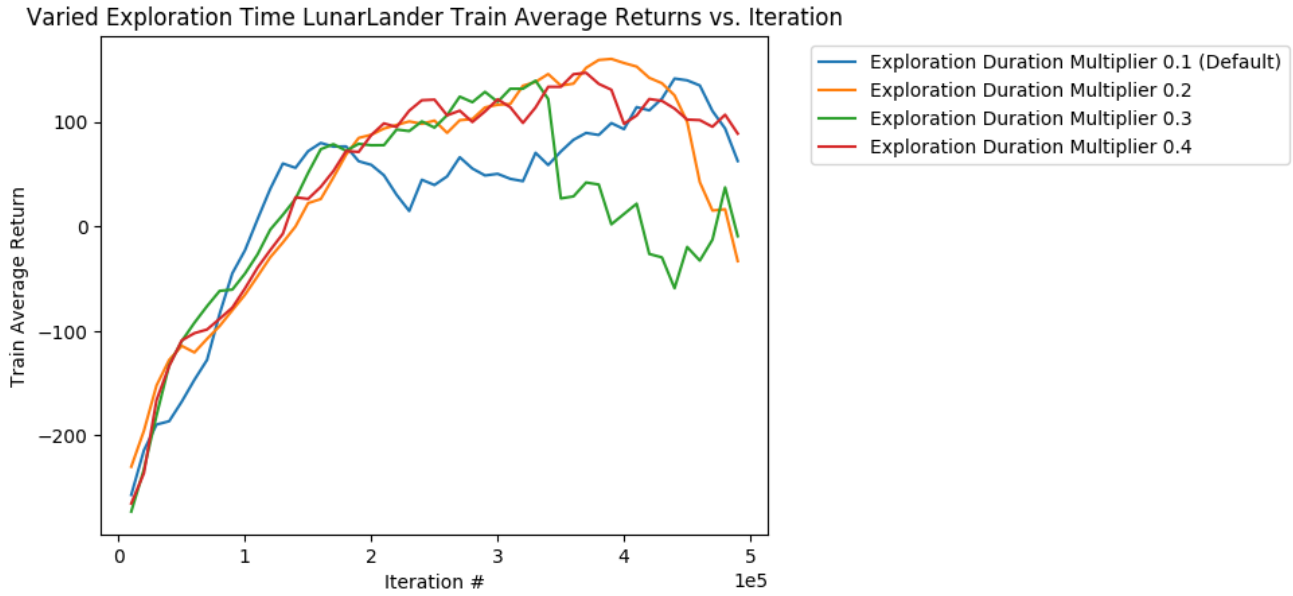

Figure 3: An analysis of how sensitive Q-learning on LunarLander is to changes to the exploration schedule that controls the ϵ value in the epsilon-greedy policy. From the graph above, we can see that as the amount of iterations the agent is allowed to explore increases, the agent takes more iterations to hit an "initial spike" in return; this is to be expected, as the default parameter run that has the least amount of exploration time resorts to more exploitation methods earlier than the other runs. However, after more iterations pass, the agents that given allowed more time to perform exploration tended to converge to the highest observed return value at a much earlier time compared to the default algorithm. In some cases, the highest return value achieved by algorithms given more time to explore was also higher than the highest return value achieved by the default algorithm.

4 Question 4

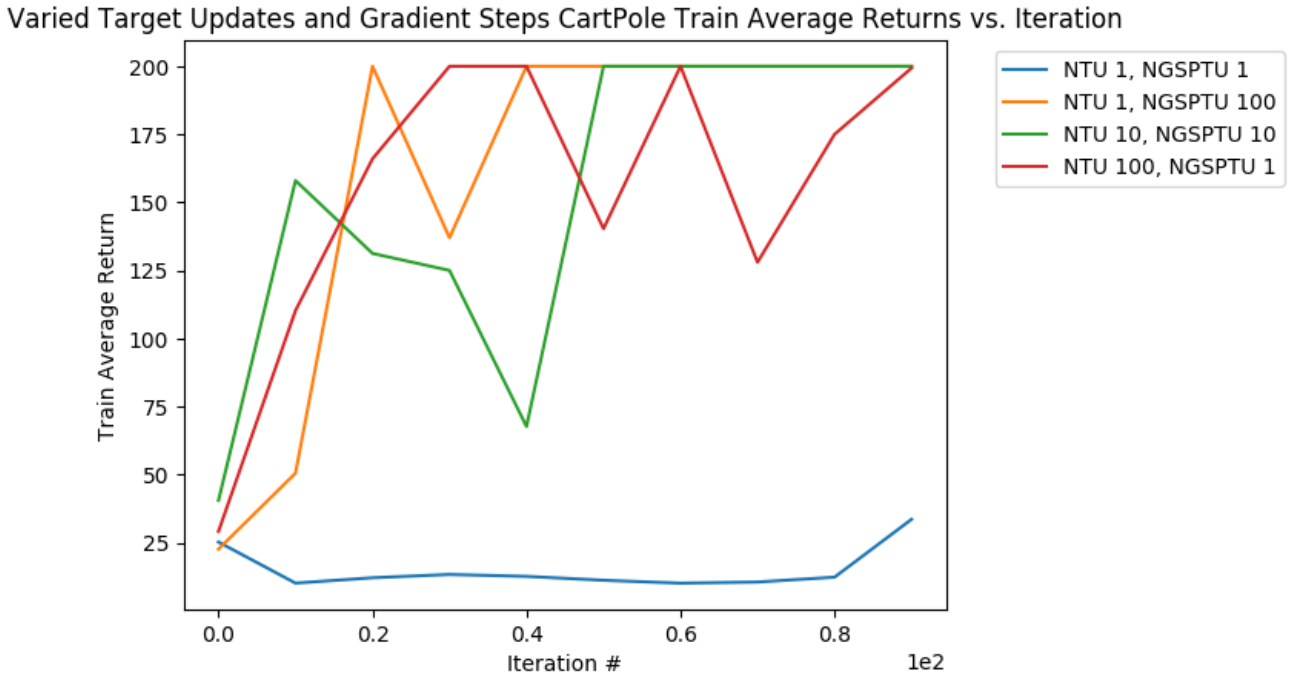

Figure 4: The training average returns for the CartPole task over iterations, where the number of target updates and number of gradient steps per target update were varied. Compared to the default algorithm that uses only one target update and one gradient step per target update within an iteration, all other algorithms that used more than one target update or gradient step per target update resulted in much higher train average returns. All non-default algorithms eventually converged to the maximal return of 200. Among these non-default algorithms, 100 target updates and 1 gradient step per target update resulted in the most unstable results. Between NTU 1 NGSPTU 100 and NTU 10 NGSPTU 10, NTU 1 and NGSPTU 100 achieved the optimal return at earlier iterations; hence, NTU 1 and NGSPTU 100 was the best set of parameters for CartPole task.

5 Question 5

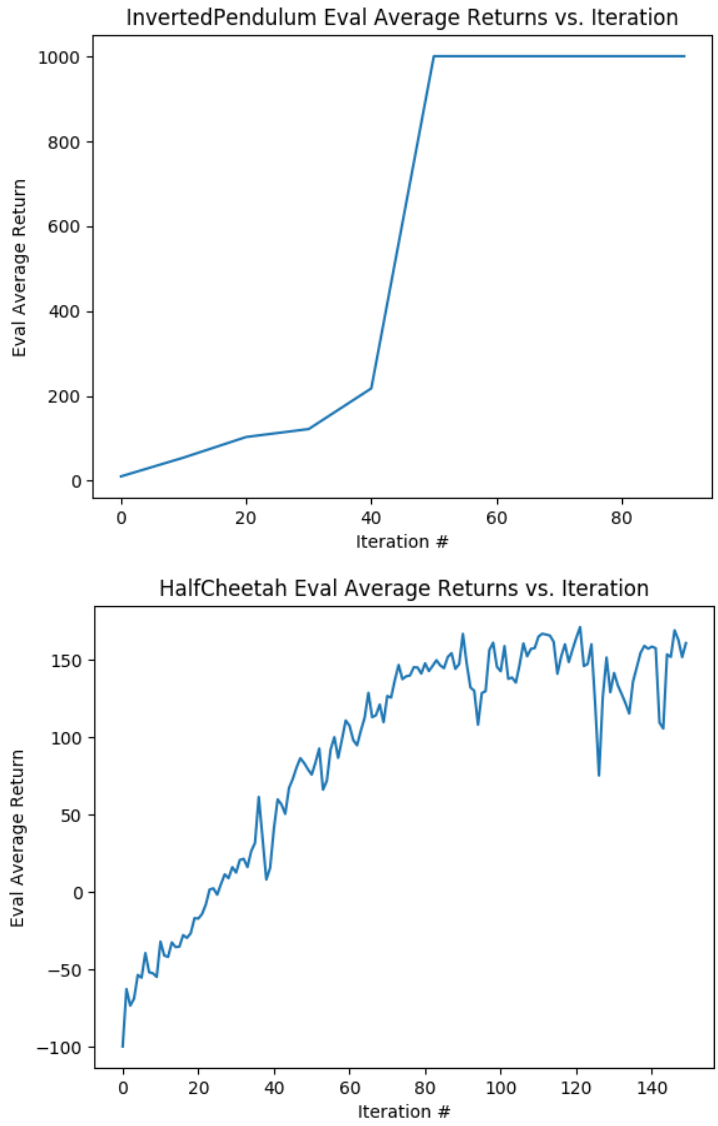

Figure 5: Evaluation average returns for InvertedPendulum and HalfCheetah using actor critic. The best eval average return achieved for InvertedPendulum was 1000, and the best eval average return achieved for HalfCheetah was 171.