CS 294-112 – Homework#3

Part 1: Q-Learning

Question 1: Basic Q-learning performance

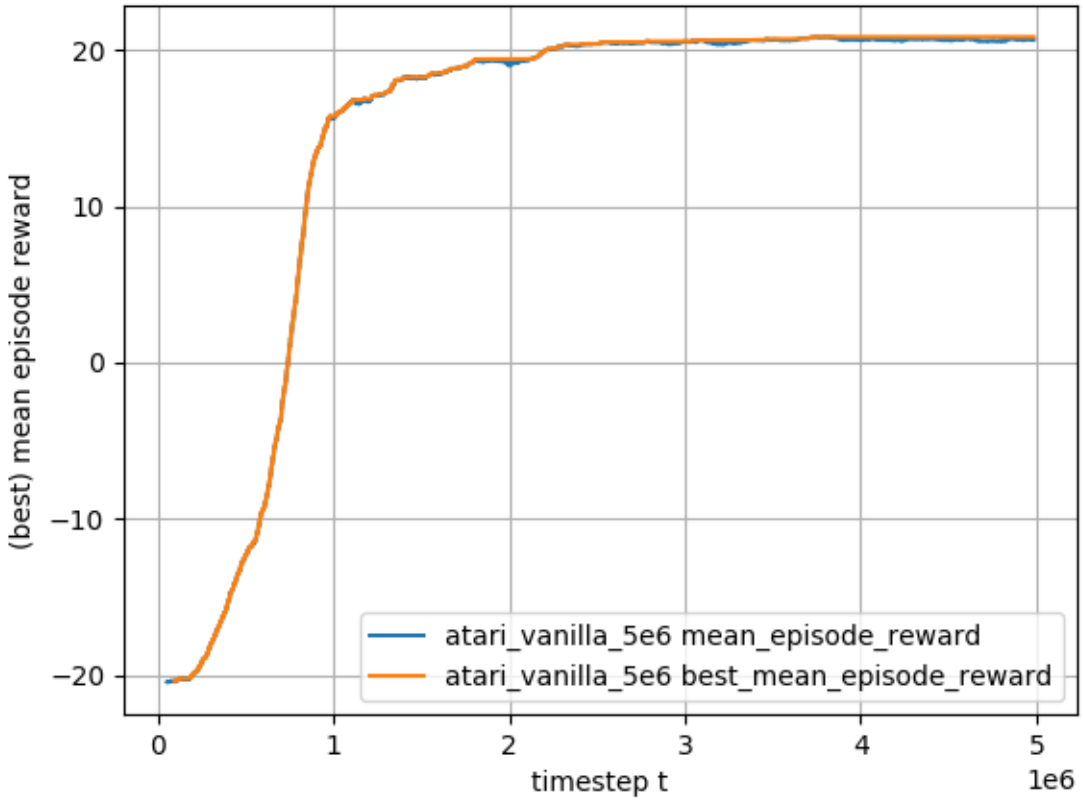

Figure 1.1. (Best) Mean Episode Reward vs Number of Timesteps / Number of Iterations. Pong game.

Question 2: Double Q-learning

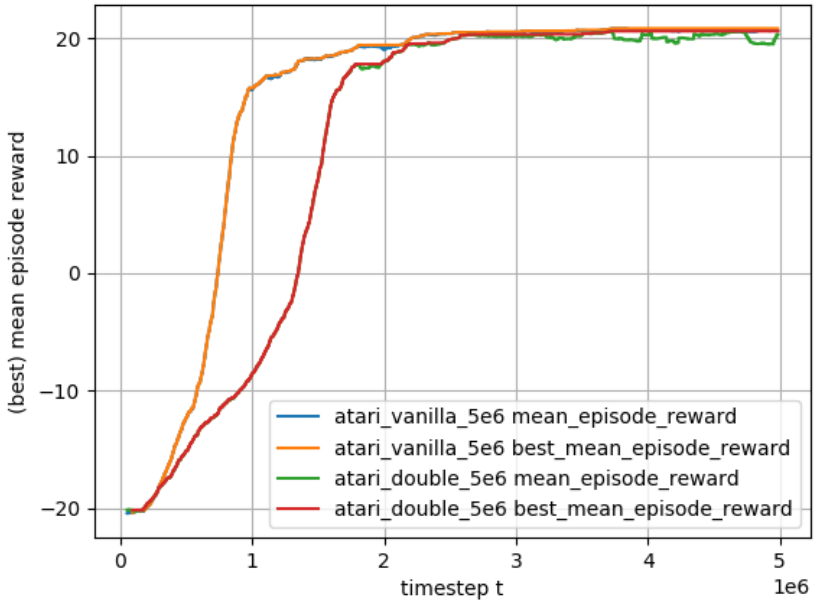

Figure 1.2a. (Best) Mean Episode Reward vs Number of Timesteps / Number of Iterations.Vanilla and double Q-learning. Pong game.

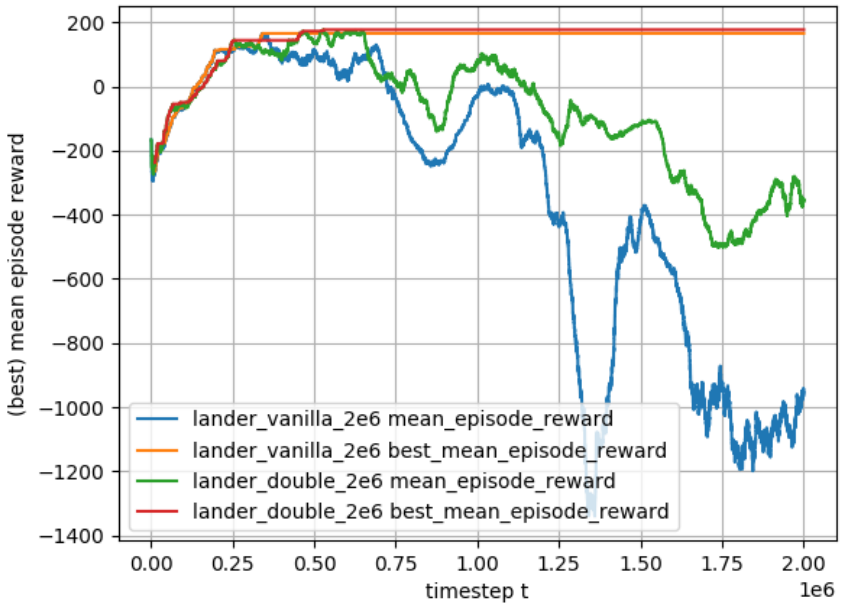

Figure 1.2b. (Best) Mean Episode Reward vs Number of Timesteps / Number of Iterations. Vanilla and double Q-learning. Lunar Lander game. Not stable after reaching the best performance.

Question 3: Experimenting with the hyperparameters

Selected the learning rate and experimented on lunar lander. The reason for selecting the learning rate is that, based on the figure 1.2.b, its performance is not after reaching the best performance so we suspect that it learned too much given the certain amount of data. Thus we reduce the learning rate. The plot is as follows:

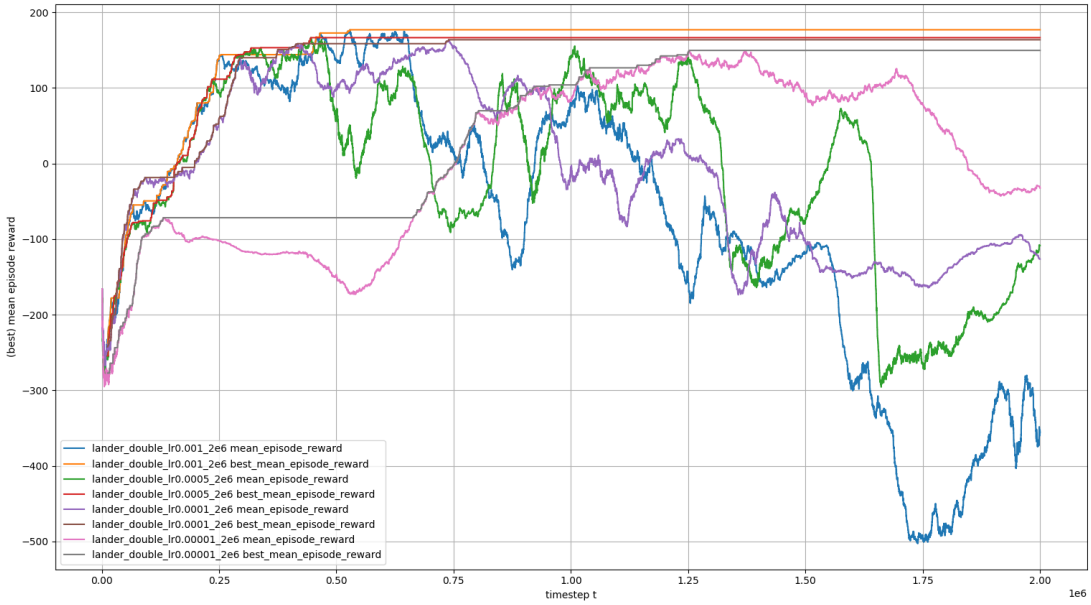

Figure 1.3. Double-Q learning. Lunar Lander. The original learning rate: 0.001 (blue and orange). Learning rate: 0.0005 (green and red). Learning rate: 0.0001 (violet and brown). Learning rate: 0.00001 (magenta and gray). The performance is slightly worse when the learning rate reduces but the drop is less, though the performance is still not stable.

Part 2: Actor-Critic

Question 1: Sanity check with Cartpole

Figure 2.1. Average Return vs Number of Iterations. Cartpole in 3 scenarios.

The best scenario is 10_10, i.e. 10 target updates x 10 gradient steps.

Explanation: Given the limited data, the model should be trained on bootstrapped data but if the target data are not updated, say case 1_100, the repeated training is over-fitting. If the target data are updated too frequent, we lose the advantages of reusing data for training. Thus, the balance is what is in between: train **n** times on the data and update the target data **m** times to renew them.

Question 2: Run Actor-Critic with more difficult tasks

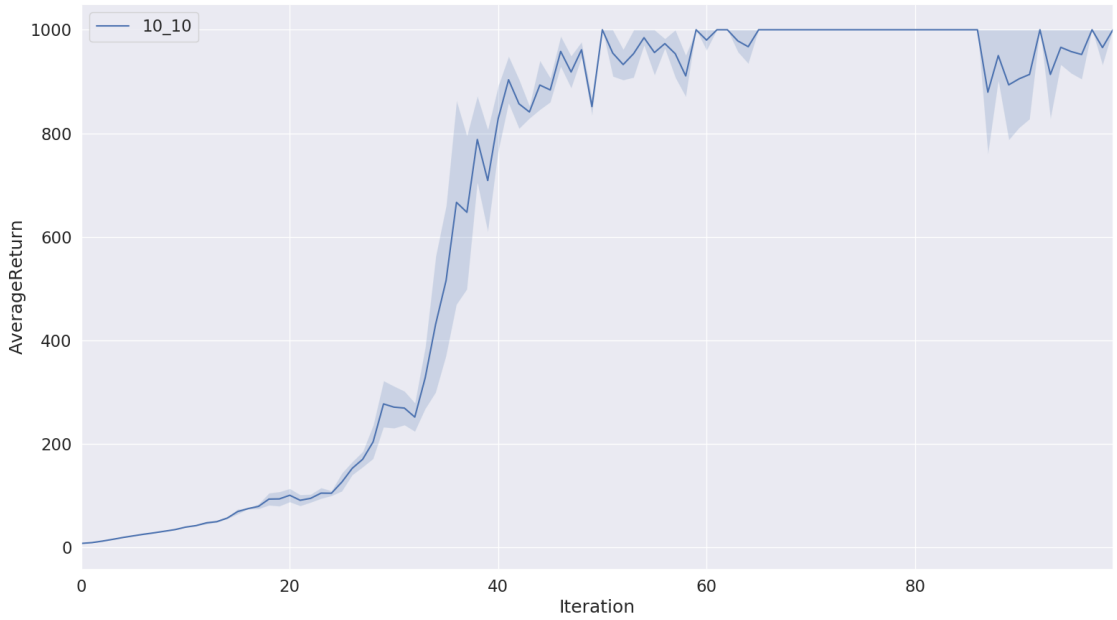

Figure 2.2a. Inverted Pendulum. 10_10

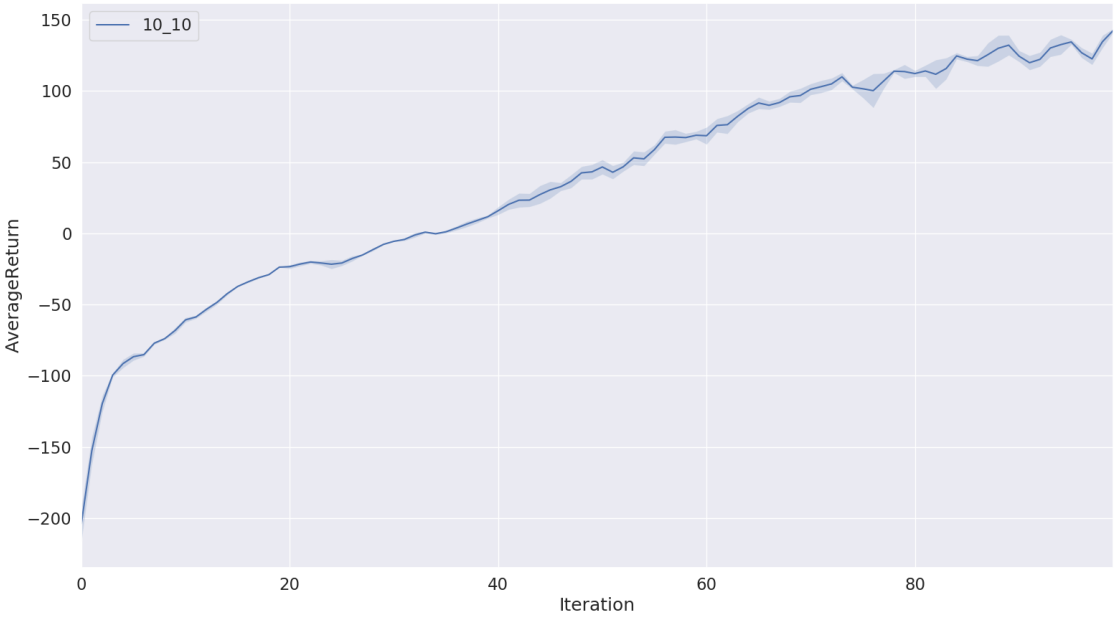

Figure 2.2b. Half Cheetah. 10_10

Bonus: Tried various architectures and learning rates including: --size 128, --nlayers 4, -ntu 20 -ngsptu 20, -ntu 40 -ngsptu 40, ratio lr_actor / lr_critic = 10 or 0.1, etc.
The one that works best is: --nlayers 3, -ntu 40 -ngsptu 40, ratio lr_actor/lr_critic = 10.

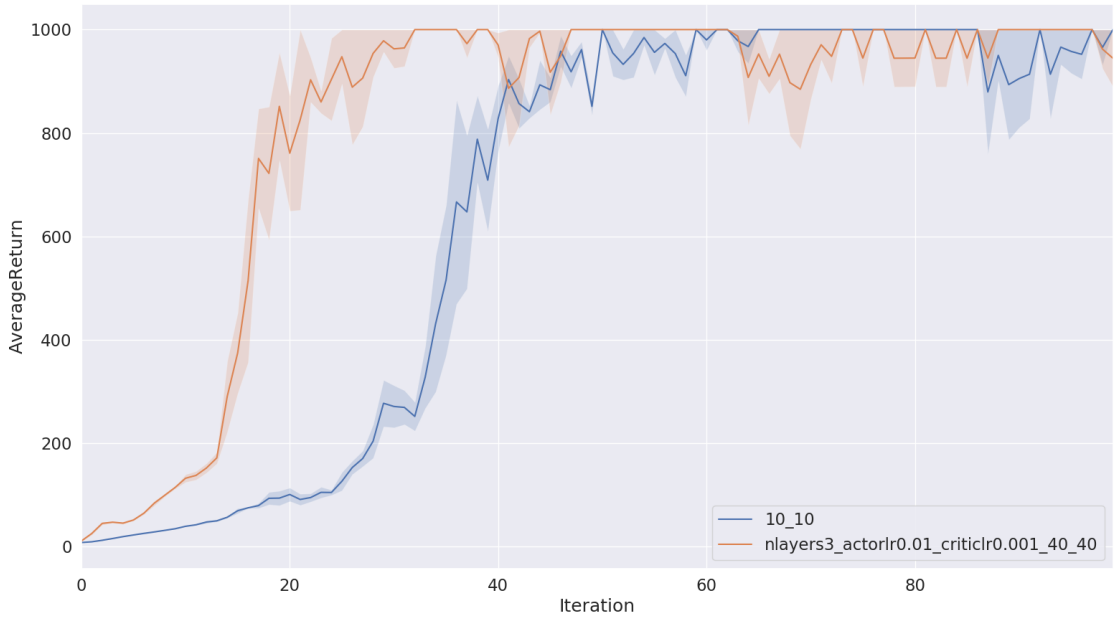

Figure B.1. Inverted Pendulum. 10_10 **vs** --nlayers 3, -ntu 40 -ngsptu 40, lr_actor 0.01, lr_critic 0.001

Figure B.2. Half Cheetah. 10_10 **vs** --nlayers 3, -ntu 40 -ngsptu 40, lr_actor 0.02, lr_critic 0.002