Homework 3

Author:
Reece D. Huff

CS 285: Deep Reinforcement Learning

Dr. Sergey Levine

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

Question 1

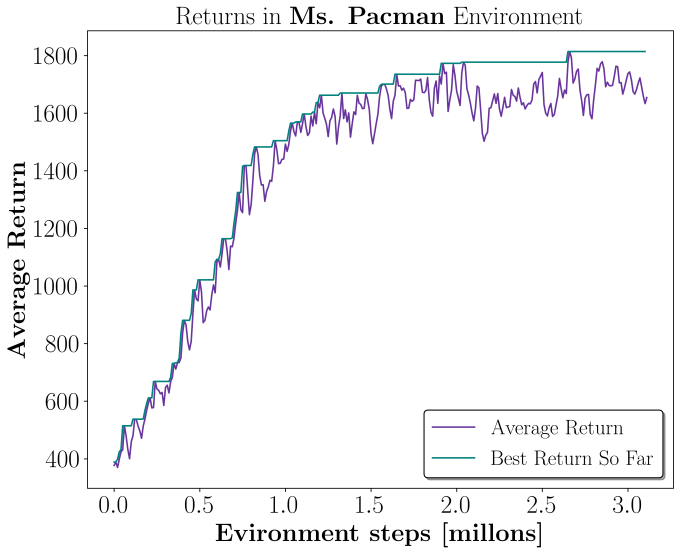


Figure 1: Returns from the `Ms. Pacman` environment.

`experiment1.sh`

```
python cs285/scripts/run_hw3_dqn.py --env_name MsPacman-v0 --exp_name q1
```

Question 2

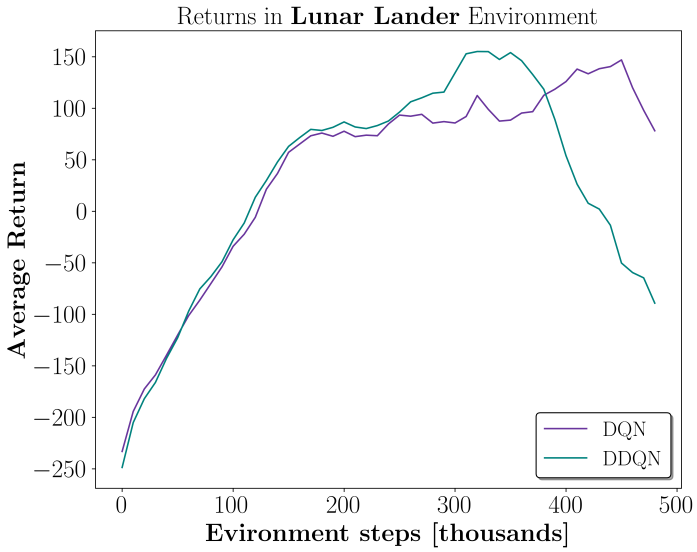


Figure 2: Average return in the `LunarLander` environment for both DQN and DDQN. Note that the plots are averaged acrossed three random seeds. Interestingly, DDQN reachs a higher average return but drops at the end, though I imagine this would be fixed if I ran it on more random seeds.

`experiment2.sh`

```
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name q2_dqn_1 --
    seed 1
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name q2_dqn_2 --
    seed 2
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name q2_dqn_3 --
    seed 3

python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name
    q2_doubledqn_1 --double_q --seed 1
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name
    q2_doubledqn_2 --double_q --seed 2
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name
    q2_doubledqn_3 --double_q --seed 3
```

Question 3

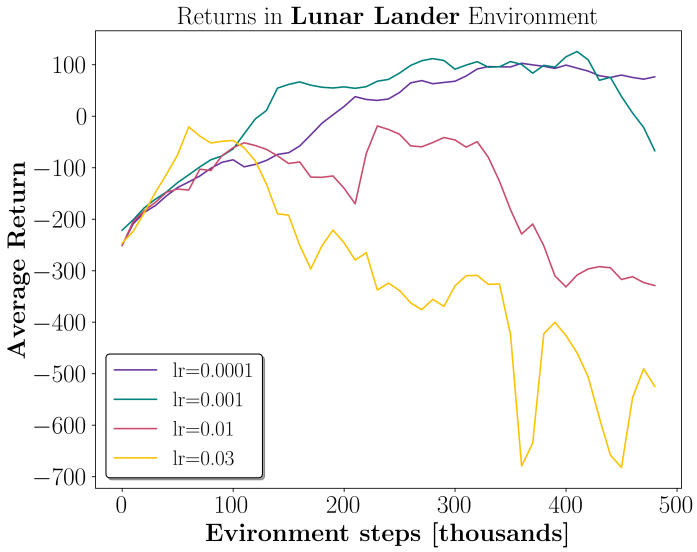


Figure 3: Average returns in the `LunarLander` environment as the `lr` rate is varied. It is interesting to note that for DQN, the performance can suffer if the learning rate is too large. We find that the small learning rates are far more stable.

`experiment3.sh`

```
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name q3_hparam1 --
    lunar_lr 1e-4
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name q3_hparam2 --
    lunar_lr 1e-3
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name q3_hparam3 --
    lunar_lr 1e-2
python cs285/scripts/run_hw3_dqn.py --env_name LunarLander-v3 --exp_name q3_hparam4 --
    lunar_lr 3e-2
```

Question 4

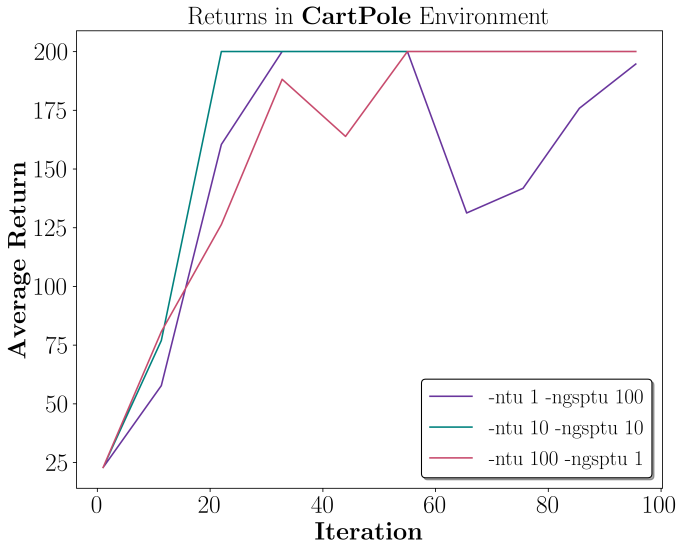
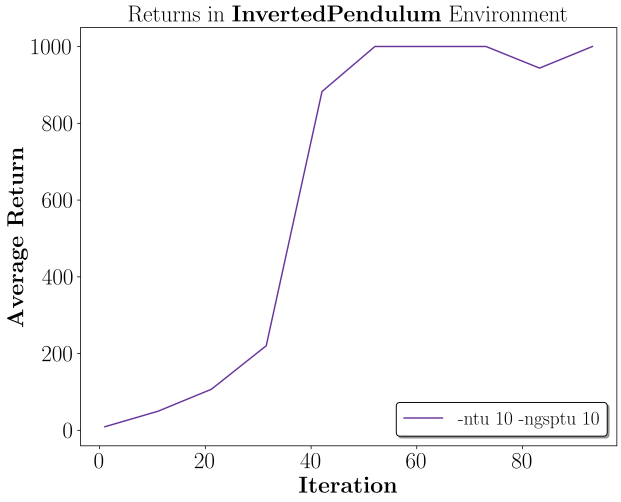


Figure 4: Average returns in the `CartPole` environment as the number of target updates (`-ntu`) and the number of gradient updates per target update (`-ngsptu`) is varied. The optimal set of hyperparameters is 10 and 10 for `-ntu` and `-ngsptu` because we see that it reaching 200 and stays at 200.

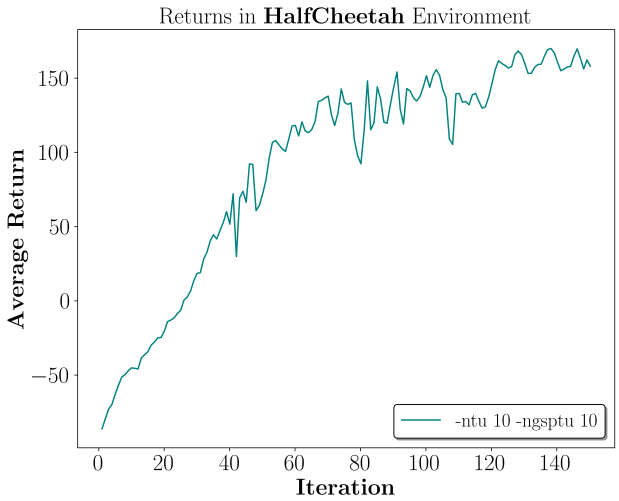
`experiment4.sh`

```
python cs285/scripts/run_hw3_actor_critic.py --env_name CartPole-v0 -n 100 -b 1000 --
    exp_name q4_100_1 -ntu 100 -ngsptu 1
python cs285/scripts/run_hw3_actor_critic.py --env_name CartPole-v0 -n 100 -b 1000 --
    exp_name q4_1_100 -ntu 1 -ngsptu 100
python cs285/scripts/run_hw3_actor_critic.py --env_name CartPole-v0 -n 100 -b 1000 --
    exp_name q4_10_10 -ntu 10 -ngsptu 10
```

Question 5



(a) Returns in the `InvertedPendulum` environment with optimal hyperparameters (`-ntu 10 & -ngsptu 10`).



(b) Returns in the `HalfCheetah` environment with optimal hyperparameters (`-ntu 10 & -ngsptu 10`).

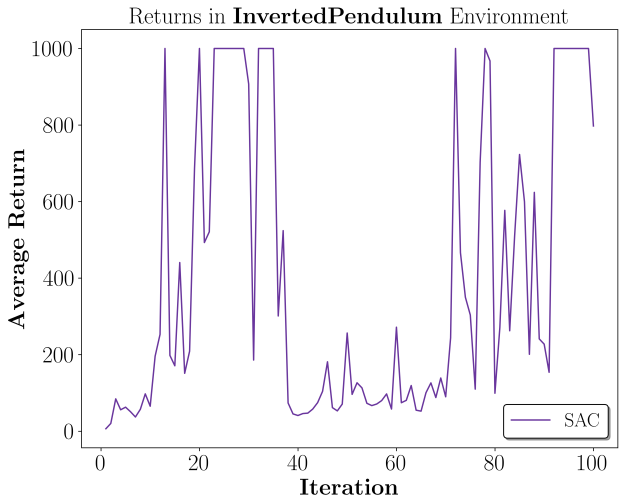
Figure 5: All of the results for Question 5.

`experiment5.sh`

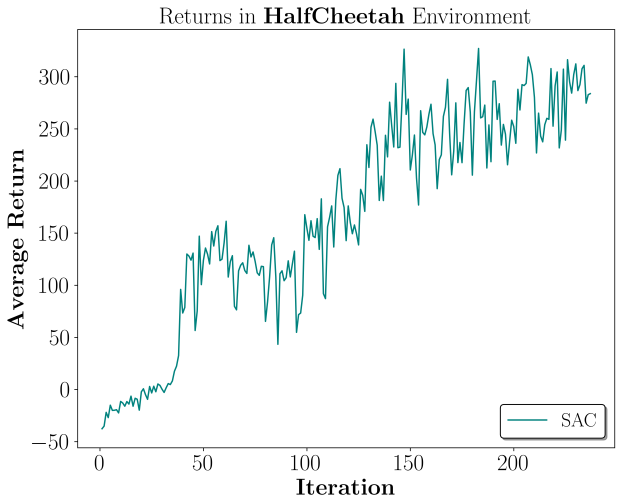
```
python cs285/scripts/run_hw3_actor_critic.py --env_name InvertedPendulum-v4 --ep_len
    1000 --discount 0.95 -n 100 -l 2 -s 64 -b 5000 -lr 0.01 --exp_name q5_10_10 -ntu
    10 -ngsptu 10

python cs285/scripts/run_hw3_actor_critic.py --env_name HalfCheetah-v4 --ep_len 150 --
    discount 0.90 --scalar_log_freq 1 -n 150 -l 2 -s 32 -b 30000 -eb 1500 -lr 0.02 --
    exp_name q5_10_10 -ntu 10 -ngsptu 10
```

Question 6



(a) Returns in the `InvertedPendulum` environment with optimal hyperparameters (`--actor_update_frequency 10`).



(b) Returns in the `HalfCheetah` environment with optimal hyperparameters (`--actor_update_frequency 10 --lr 0.00001`).

Figure 6: All of the results for Question 6. Notice that the `--actor_update_frequency` was increased to 10. I noticed that this helped performance as it was discussed on Edstem. Also, I had to lower the `lr` in the `HalfCheetah` environment to get good performance.

NOTE: I ended the `HalfCheetah` environment run early because it was taking a long time and it was close to deadline. The reward was already very high (350), so I figured it was fine and the passed the GradeScope autograder. Please let me know if you would like for me to run it for longer and I can resubmit.

`experiment6.sh`

```
python cs285/scripts/run_hw3_sac.py --env_name InvertedPendulum-v4 --ep_len 1000 --
    discount 0.99 --scalar_log_freq 1000 -n 100000 -l 2 -s 10 -b 1000 -eb 2000 -lr
    0.0003 --init_temperature 0.1 --exp_name q6a_sac_InvertedPendulum --seed 1 --
    actor_update_frequency 10

python cs285/scripts/run_hw3_sac.py --env_name HalfCheetah-v4 --ep_len 150 --discount
    0.99 --scalar_log_freq 1500 -n 2000000 -l 2 -s 256 -b 1500 -eb 1500 -lr 0.00001 --
    init_temperature 0.1 --exp_name q6b_sac_HalfCheetah --seed 1 --
    actor_update_frequency 10
```