

CS 533

Assignment 4

Distributed Deep Q-Learning

Name:

Jui-Hung Lu (933-293-709)

Chi Wen (933-276-677)

Part 1: Non-distributed DQN

For each experiment, record the parameters that you used, plot the resulting learning curves, and give a summary of your observations regarding the differences you observed.

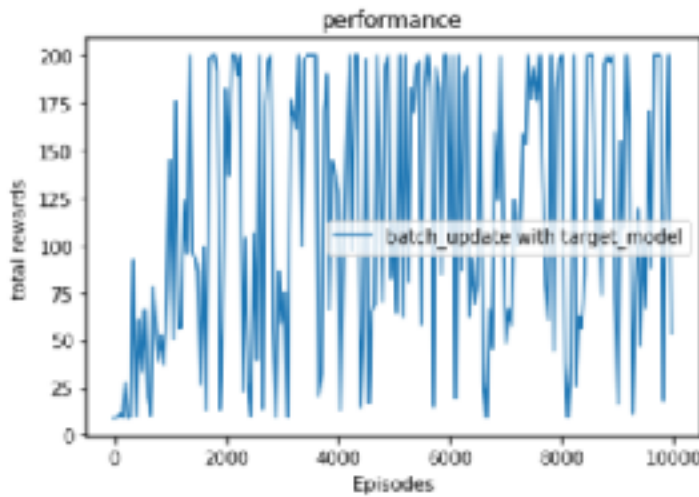


Figure 1. DQN without a replay buffer and without a target network

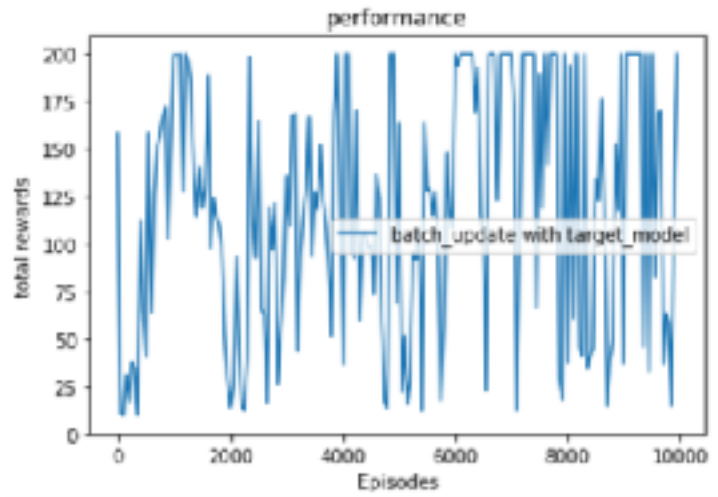


Figure 2. DQN without a replay buffer (but including the target network)

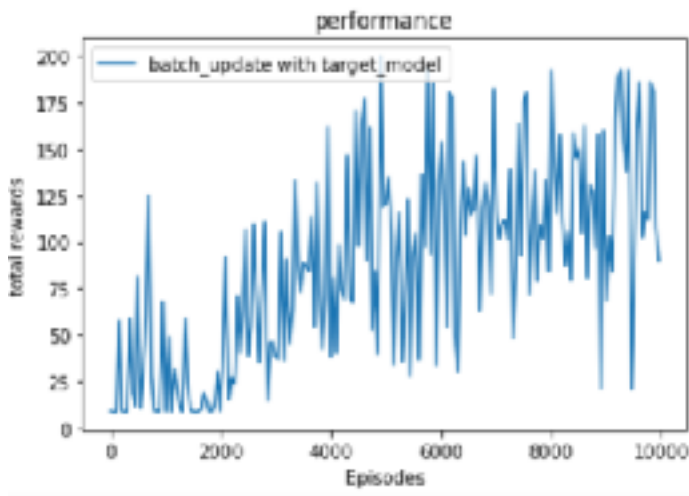


Figure 3. DQN with a replay buffer, but without a target network

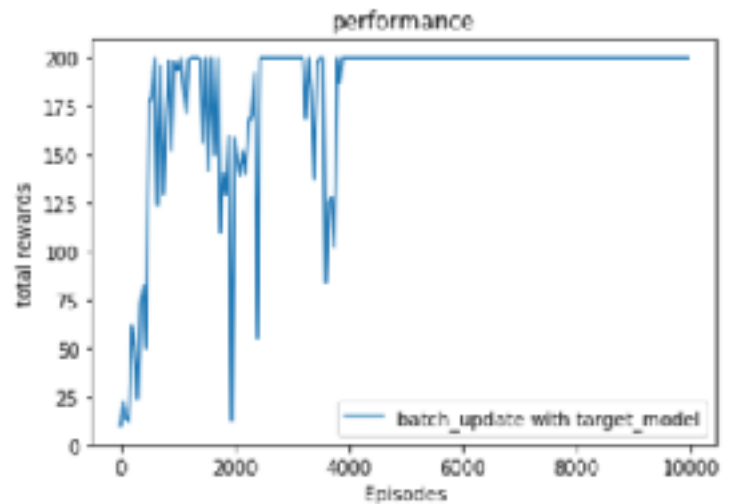


Figure 4. Full DQN

Parameters:

Figure 4: epsilon_decay_steps=100000, final_epsilon=0.1, batch_size=32, update_steps=10, memory_size=2000, beta=0.99, model_replace_freq=2000, learning_rate=0.0003, use_target_model=True

Figure 3: Set use_target_model = False and other parameters same as figure 4.

Figure 1: Set memory_size=1, update_steps=1, batch_size=1 and other parameters same as figure 3.

Figure 2: Set use_target_model = True and other parameters same as figure 1.

Summary:

From figure 1 to figure 4, it is clear that full DQN (figure 4.), which has a target network and a replay buffer, has great overall performance. Comparing those three other images, figure 3, which is the DQN with the replay buffer but without a target network stands out, that shows that having an experiment replay makes greater impacts on both time and performance perspective. From figure 1 and figure 2, we can observe the target network do has effect on the performance, but not as much as the replay buffer did.

In conclusion, experience replay has the largest performance improvement, on the other hand, target network improvement is significant but not as critical as the replay buffer.

Part 2: Distributed DQN

For this part of the homework you need to submit your code for distributed DQN and run experiments that vary the number of workers involved. Produce some learning curves and timing results and discuss your observations.

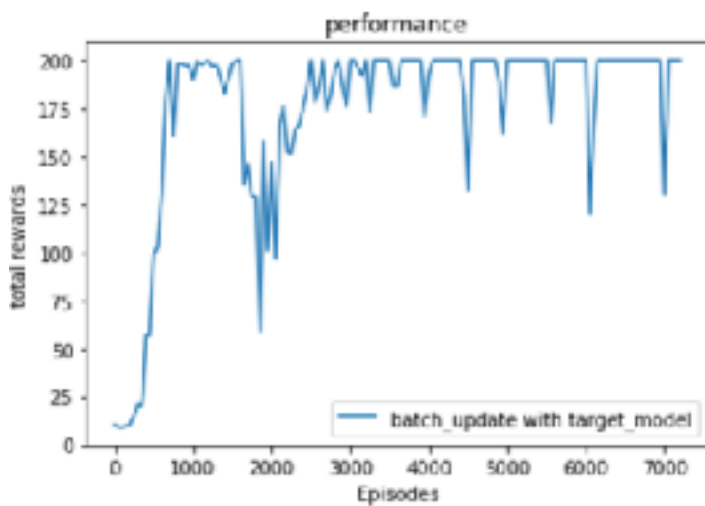


Figure 5. 4 collectors, 4 evaluators

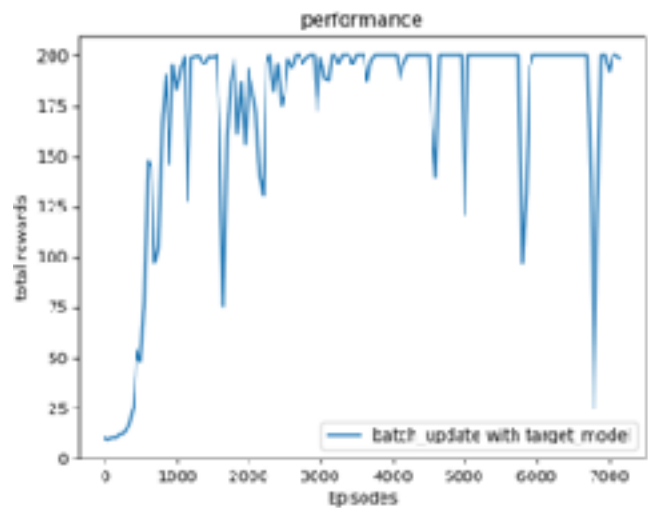


Figure 6. 8 collectors, 4 evaluators

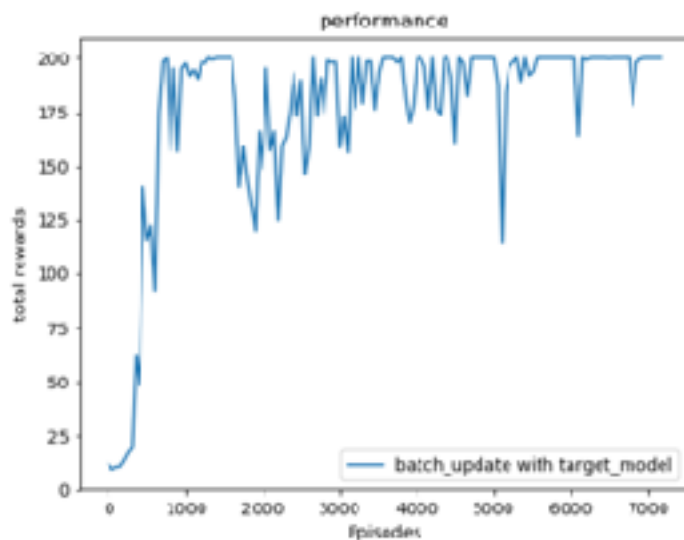
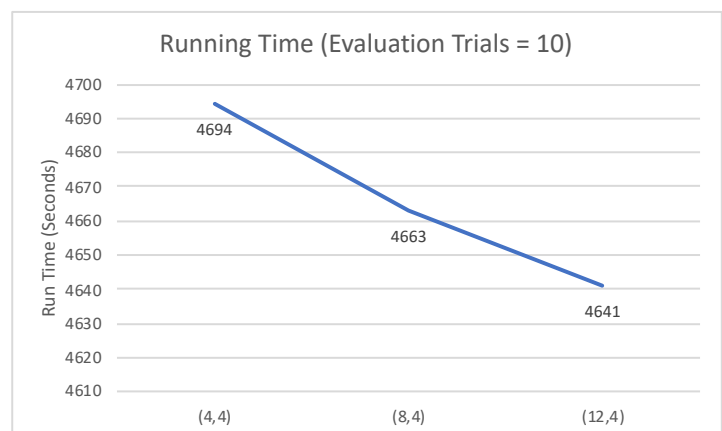


Figure 7. 12 collectors, 4 evaluators



Observation:

The distributed DQN with 12 collectors and 4 evaluators is the fastest compared to others. In the case of converging, because collectors spend time on communicating with the server more than evaluators does, collectors have more influence on the overall running time than evaluators does. Moreover, if the amount of the collector increases, the amount of completed episode will also increases at the same time, which could explain the reason of 12 collectors and 4 evaluators being the fastest of all.