

**Homework #4. Distributed Deep Q-Learning**

Yongsung Cho

CHI CHIEH WENG

Computer Science, Oregon State University

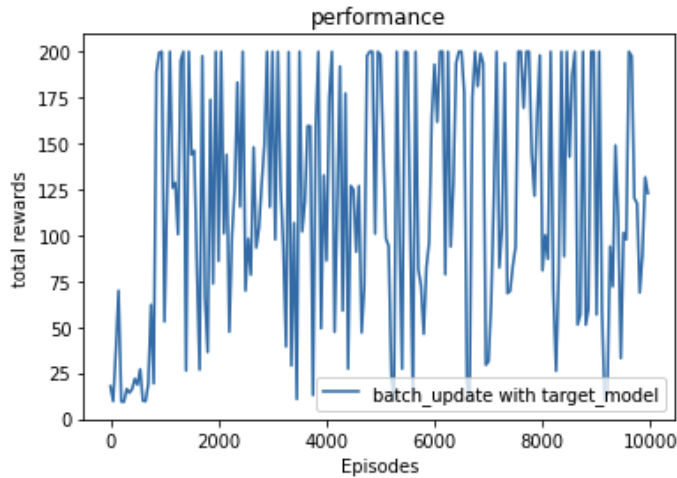
AI 533: Intelligent Agents & Decision

Prof. Alan Fern

Due Dec 6, 2021

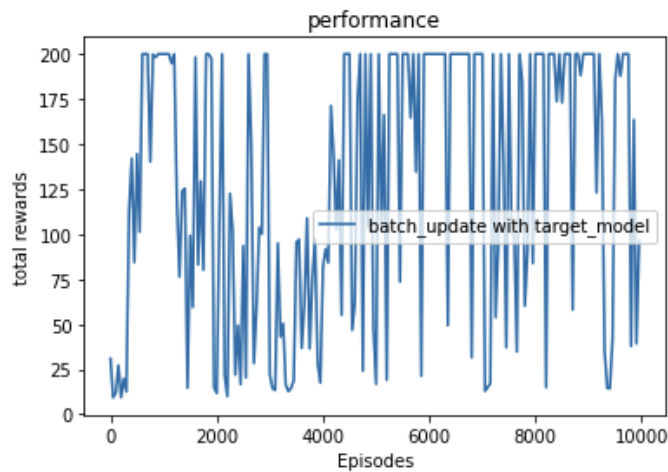
## Part 1: Non-distributed DQN

1. **DQN without a replay buffer and without a target network.** (memory\_size = 1, update\_steps = 1, batch\_size = 1, use\_target\_model = False)

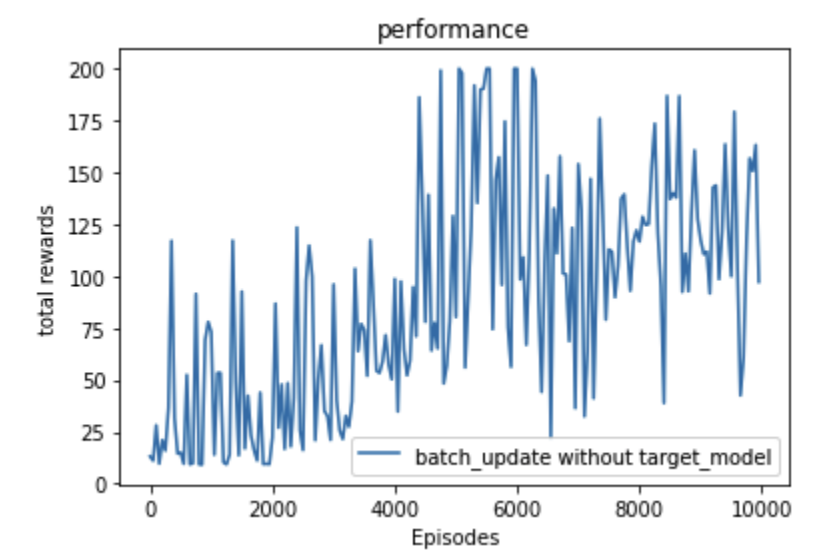


Slow

2. **DQN without a replay buffer (but including the target network).**  
(memory\_size = 1, update\_steps = 1, batch\_size = 1, use\_target\_model = True)

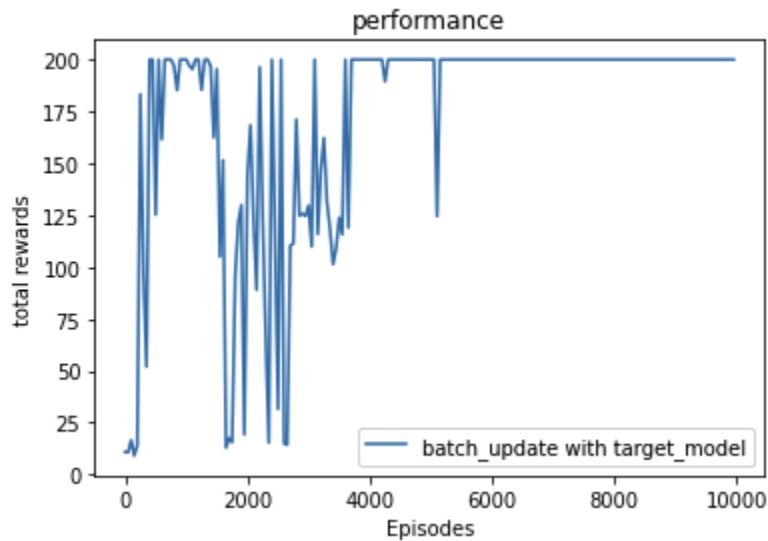


3. **DQN with a replay buffer, but without a target network.**



Fast but.... result

4. **Full DQN**



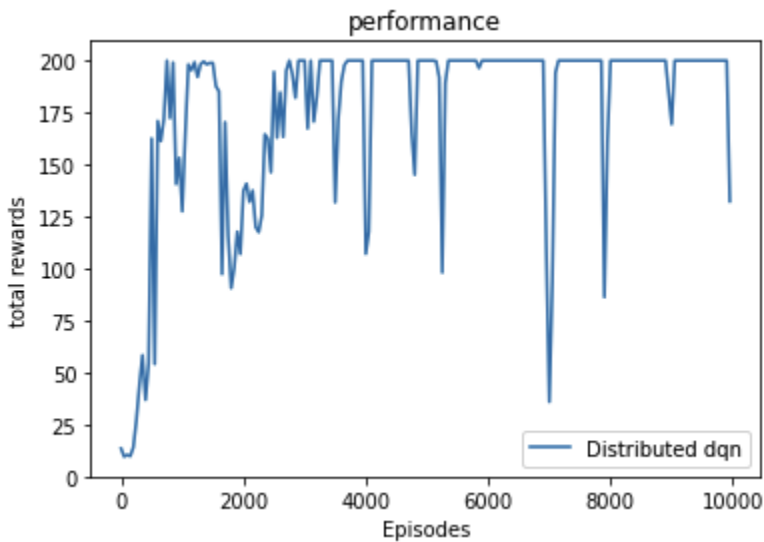
**Observations:**

Comparing the realized runtimes of these four DQNs, we find that a DQN without a replay buffer will take longer than a DQN with a replay buffer. In addition, the DQN with replay buffers has better performance than the DQN without replay buffers, and it can be easily seen from the graph that the reward increases gradually for the DQN with replay buffers, but

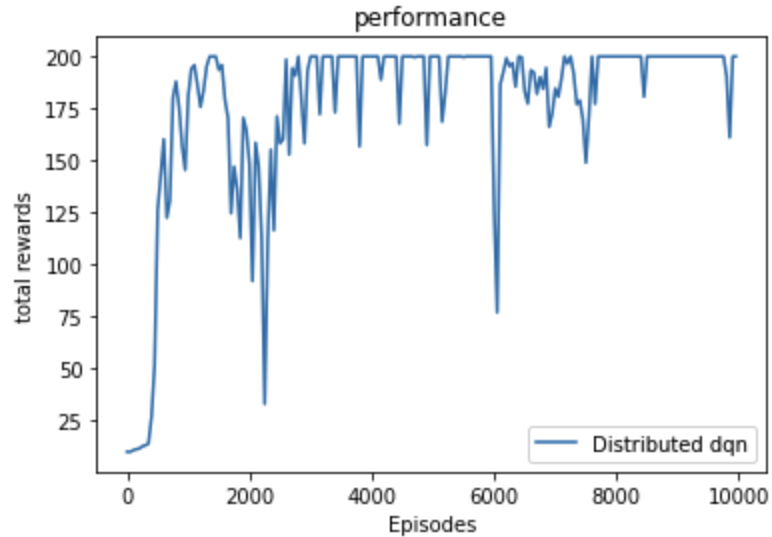
no progress is made for the DQN without replay buffers. On the other hand, the full DQN with replay buffers and target network has the best performance and tends to get the maximum reward after running enough episodes. However, the DQN with only the target network performs poorly, It requires a replay buffer and a target network to achieve a suitable learning curve.

## Part 2: Distributed DQN(training\_episodes = 10000, test\_interval = 50, trials = 30)

collector workers = 4, evaluator workers = 4, total time = 464 (second)



collector workers = 12, evaluator workers = 4, total time = 439 (second)



### Observations:

First, we tried two different sets of distributed DQNs with a large difference in the number of workers between the two sets, as this clearly shows the difference in the results. As we expected, the runtime with 12 collector workers was the fastest, followed by 4 collector workers. In addition, we can clearly observe that the more collector workers will have the faster training speed, because each collector will perform its tasks in parallel. (In this part, there are some bugs in our program that cannot display the resulting graph on the exclusive node. Therefore, we collect the running results on the exclusive node, and then run the graph on the Jupyter notebook, so it does not affect the results.)