Reinforcement Learning Homework 04

Qiu Yihang

April 2023

1 Implementation of the DQN and Double DQN algorithms

The source code is under directory `code`. The results and analyses are as follows.

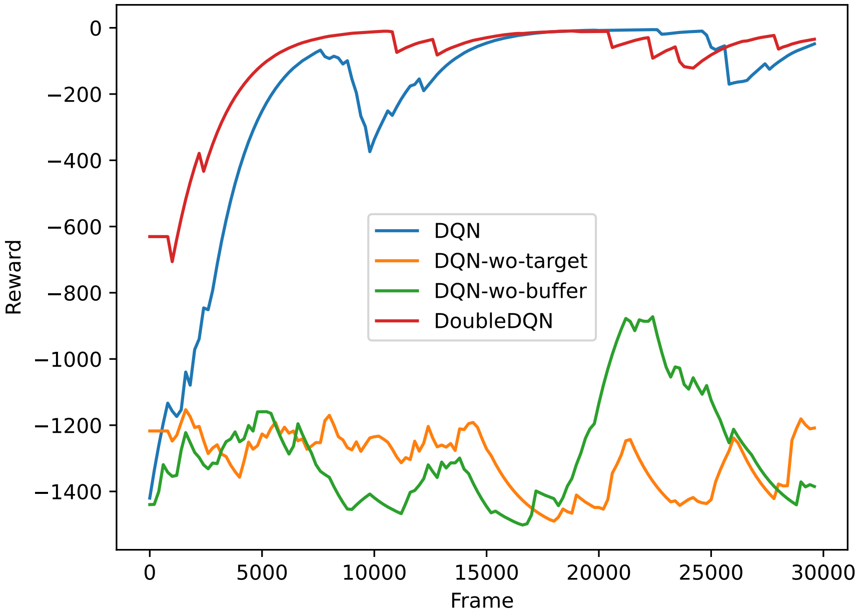


Figure 1: The comparison of the rewards of the DQN (with or without target network or buffer) and Double DQN algorithms on the `Pendulum-v0` environment during training.

1.1 The Target Network

DQN without target network **appears more unstable** and also fails to find a good policy to gain a higher reward.

The reason is that target network will be updated after several iterations of updating on the main network, help to stablize the expected Q value, which helps to stablize the training process.

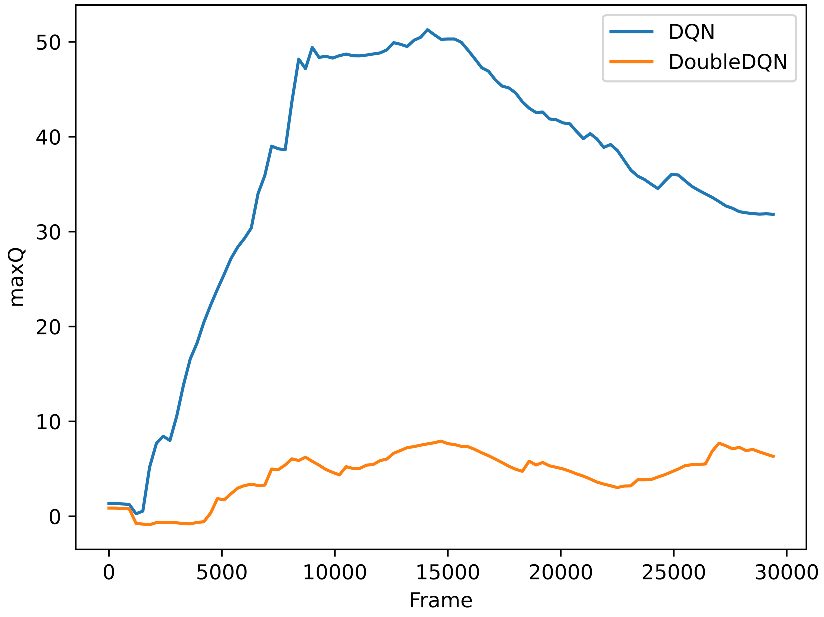
1.2 Buffer

DQN without buffer **appears more unstable** and also fails to find a good policy to gain a higher reward.

The reason is that the buffer uniformly samples the information collected in the past to replay and helps the agent to learn from the experience of the past, which prevents repetitive exploration and meaningless repetitions. This also helps stablize the training since the distribution of the samples is more uniform.

1.3 The Difference between Vanilla DQN and Double DQN

The estimated maximal Q-value of the two algorithms during training are shown as follows.



The results show that the Double DQN algorithm **converges faster** and **appears more stable** than the vanilla DQN algorithm. Also, from the perspective of the estimated Q value, the vanilla DQN tends to **overestimate** the Q value.

This is because the Double DQN algorithm uses the main network to select the action and the target network to evaluate the Q-value.

Actions selected by the main network are more precise, helping Double DQN converge faster. Value estimation by target network is more stable, which helps to stablize the training process and prevents overestimation.