Paper Reading No.16

Deep Reinforcement Learning with Double Q-Learning

Sheng Lian

October 2019

1 Brief Paper Intro

- ***Paper ref:*** AAAI 2016, https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/viewPaper/12389
- ***Authors:*** Hado van Hasselt , Arthur Guez, and David Silver from Google DeepMind,
- ***Paper summary:*** The popular Q-learning algorithm is known to overestimate action values under certain conditions. In this paper, authors prove that DQN, which combines the Q-learning and deep neural network, also suffers from substantial overestimations. And this overestimation, both in Q-learning and DQN, can harm the RL's performance. So authors proposed Double DQN (DDQN), which is extended from Double Q-learning, to solve this problem.
- ***Reading motivation:*** When reading DQN papers, you cannot skip DDQN. This paper optimize DQN from it's essence and obtain promising performance.

2 The problem of Q-learning and DQN

In RL, under a given policy π, the true value of an action a in a state s is

$$Q_\pi(s,a) \equiv \mathbb{E}\left[R_1 + \gamma R_2 + \ldots | S_0 = s, A_0 = a, \pi\right] \tag{1}$$

The optimal value is then $Q_*(s,a) = \max_\pi Q_\pi(s,a)$. However, how to choose the 'max' is tricky. **Q-learning** algorithm can be used for estimating the optimal action values by learning a parameterized value function $Q\left(s,a;\boldsymbol{\theta}_t\right)$. The parameter θ can be updated by

$$\theta_{t+1} = \theta_t + \alpha\left(Y_t^{\mathrm{Q}} - Q\left(S_t, A_t; \boldsymbol{\theta}_t\right)\right)\nabla_{\boldsymbol{\theta}_t} Q\left(S_t, A_t; \boldsymbol{\theta}_t\right) \tag{2}$$

and the Y_t^Q is defined as

$$Y_t^{\mathrm{Q}} \equiv R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a; \boldsymbol{\theta}_t\right) \tag{3}$$

In **DQN** situation, the process of learning θ is accompanied by deep neural network. For optimizing the performance, DQN introduced target network and experience replay to training, where target network's parameter θ^- are copied from online network's θ. The formulation goes as

$$Y_t^{\mathrm{DQN}} \equiv R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a; \boldsymbol{\theta}_t^-\right) \tag{4}$$

3 Doubel Q-learning

After introducing Q-learning and DQN, here is the problem: The max operator in standard Q-learning and DQN, uses the same values both to select and to evaluate an action. This makes it more likely to select overestimated values, resulting in overoptimistic value estimates. In Double Q-learning [1], two value functions are learned to update value functions. θ is used for

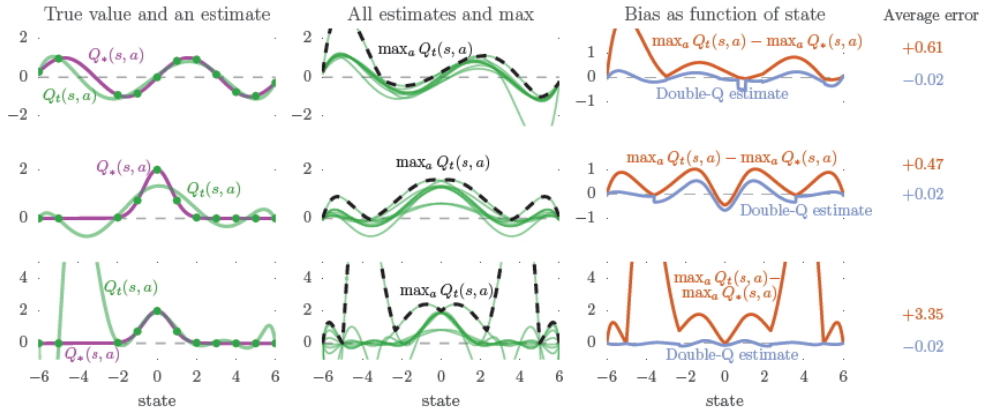

Figure 1: Illustration of overestimations during learning.

choosing the greedy policy and the θ' is used for determining the value. The Double Q-learning's error goes as

$$Y_t^{\text{DoubleQ}} \equiv R_{t+1} + \gamma Q\left(S_{t+1}, \operatorname{argmax} Q\left(S_{t+1}, a; \boldsymbol{\theta}_t\right); \boldsymbol{\theta}'_t\right) \tag{5}$$

After that, authors prove that even if the value estimates are on average correct, estimation errors of any source can drive the estimates up and away from the true optimal values. And, Q-learning's overestimations increase with the number of actions, while Double Q-learning is unbiased. Fig 1 shows this adventure of Double Q-learning.

4 Double DQN

Like DQN to Q-learning, authors propose Double DQN as a optimization of Double Q-learning. As introduced above, DQN naturally have a second value function, so DDQN don't need to introduce additional network like Double Q-learning do. Therefore, authors propose to choose the greedy policy according to the online network, but using the target network to estimate its

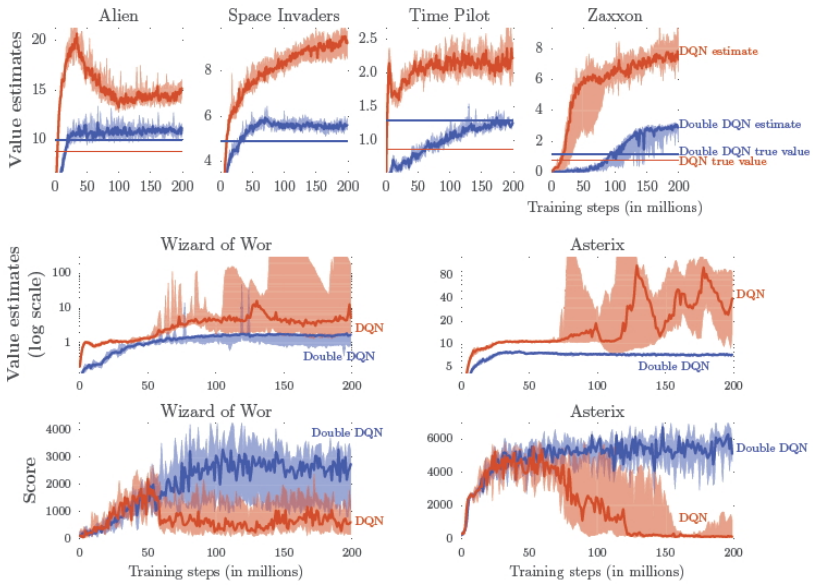

Figure 2: The value and score estimates by DQN (orange) and Double DQN (blue) on six Atari games

value. So the target Y is replaced by

$$Y_t^{\text{DoubleDON}} \equiv R_{t+1} + \gamma Q\left(S_{t+1}, \underset{a}{\text{argmax}} Q\left(S_{t+1}, a; \boldsymbol{\theta}_t\right), \boldsymbol{\theta}_t^-\right) \tag{6}$$

Here, the target network θ_t^- for the evaluation of the current greedy policy. The update to the target network stays unchanged from DQN, and remains a periodic copy of the online network.

Authors evaluate the proposed method on many games, e.g. in Fig 2, the DQN and DDQN is tested on 6 Atari games. Besides, many other experiments show that DDQN is a simple but effective modification towards DQN.

References

[1] Hado V Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.