The Effect of Multi-step Methods on Overestimation in Deep Reinforcement Learning

Lingheng Meng, Rob Gorbet, Dana Kulić

Abstract

Multi-step (also called n-step) methods in Reinforcement Learning (RL), with tabular representation of the value-function, have been shown to be more efficient than the 1-step method due to faster propagation of the reward signal, both theoretically and empirically, in tasks exploiting tabular representation of the value-function. Recently, research in Deep Reinforcement Learning (DRL) also shows that multi-step methods improve learning speed and final performance in applications where the value-function and policy are represented with deep neural networks. However, there is a lack of understanding about what is contributing to the boost of performance. In this work, we analyze the effect of multi-step methods on alleviating the overestimation problem in DRL, where multi-step methods improve learning speed and final performance in tasks exploiting tabular representation of the value-function. Recently, research in Deep Reinforcement Learning (DRL) also shows that multi-step methods improve learning speed and final performance in applications where the value-function and policy are represented with deep neural networks. However, there is a lack of understanding about what is contributing to the boost of performance. In this work, we analyze the effect of multi-step methods on alleviating the overestimation problem in DRL, where multi-step methods improve learning speed and final performance.

Motivation

- Multi-step (also called n-step) methods in Reinforcement Learning (RL), with tabular representation of the value-function, have been shown to be more efficient than the 1-step method due to faster propagation of the reward signal.
- However, there is a lack of understanding about what is contributing to the boost of performance of multi-step methods in DRL.

Background

Overestimation Problem

Assume $Q^{true}(s', a')$ is represented by a function approximator $Q^{approx}(s', a')$ with noise $\epsilon(s', a')$.

Then, for Q-Learning technique

$$Q^{approx}(s, a) = r(s, a) + \max_a Q^{approx}(s', a')$$

where zero-mean noise may easily result in overestimation problem because

$$\max_{a'} Q^{approx}(s', a') > \max_a Q^{true}(s', a')$$

E.g., if $Q^{true}(s', a') = 0$ and $E[\epsilon(s', a')] = 0$ then

$$\max_{a'} Q^{approx}(s', a') = \max_a [Q^{true}(s', a') + E[\epsilon(s', a')]]$$

while

$$\max_a Q^{true}(s', a') = 0$$

Deep Deterministic Policy Gradient (DDPG)

[2]

Critic, i.e. Q-value, is optimized by minimizing

$$L_Q = E_{\{(s_t, a_t, r_t, s_{t+1})\} \sim D}(Q_{\phi_1} - Q_{\phi_2}(s_t, a_t))^2$$

where $Q_{\phi_1}$ is target critic, and $\mu_{\phi_2} = \mu_{\phi_1}$ is target actor representing the optimal policy.

Actor, i.e. policy, is optimized by maximizing

$$J_{\theta} = E_{\tau_t \sim \tau} (Q_{\phi_2}(s_t, \mu_{\phi_1}(s_t)))$$

where $Q_{\phi_2}$ and $\mu_{\phi_1}$ are online critic and actor.

Proposed Methods

Multi-step DDPG (MDDPG)

Bootstrapped target Q is based on multi-step immediate rewards

$$Q_{(n)}^{target} = \frac{1}{n} \sum_{i=1}^{n} Q(t_i)^{(1)}$$

• The minimum of a set of target Q-values

$$Q_{(n)}^{target} = \min_i Q_i^{(1)}$$

• An average over target Q-values with step size from 2 to n, considering n = 1 is the most prone to overestimation:

$$Q_{(n)}^{target-exp-1} = \frac{1}{n-1} \sum_{i=2}^{n} Q_i^{(1)}$$

Mixed Multi-step DDPG (MMDDPG)

An average over target Q-values with step size from 1 to n

$$Q_{(n)}^{target-exp} = \frac{1}{n} \sum_{i=1}^{n} Q_i^{(1)}$$

Discussion

Experimental Evidence of Multi-step Methods’ Effect on Alleviating Overestimation

From Fig. 1:

• Almost all MDDPG(n) with n = 1 outperform DDPG.
• Bad performance of DDPG corresponds to an extremely large Q-value.

Experimental Results

Performance Comparison

3 ways to calculate $Q$ depending on how the experiences are acquired: (1) offline, (2) online, (3) model-based expansion.

$$Q_{(n)}^{(i)} = \begin{cases} r_t + \gamma \max_{a'} Q_{(n)}^{approx}(s_{t+1}, a') & \text{offline} \\ r_t + \gamma \sum_{i=1}^{n} \gamma^{i-1} r_{t+i} & \text{online} \\ r_t + \gamma \max_{a'} Q_{(n)}^{approx}(s_{t+1}, a') & \text{model-based expansion} \end{cases}$$

References