Addressing Function Approximation Error in Actor-Critic Methods

Mahdin Rohmatillah
Department of Electrical Engineering and Computer Science
National Chiao Tung University
mahdin.eed08g@nctu.edu.tw

1 Introduction

This paper mainly addresses the issue of overestimation in the high-dimensional, continuous action spaces which results in both of bias and variance, leading to sub-optimal solution. The proposed solution in this paper is called Twin Delayed Deep Deterministic Policy Gradient (TD3) , which is an extended version of Deep Deterministic Policy Gradient (DDPG) paper, Lillicrap et al. [2015]. DDPG introduces the combination of Deep Q Network (DQN), Mnih et al. [2013] with the Deterministic Policy Gradient (DPG), Silver et al. [2014], the first method handling continuous action, to form model-free off-policy actor-critic algorithm.

The main contribution of this paper is introducing three main important solutions which will clearly differs TD3 to the DDPG in tackling the overestimation issue, the first is introducing Clipped Double Q-Learning which will take the minimum value between two different critic networks. The second is delayed target network and policy update, and the last is the target policy smoothing regularization with the help of clipped noise. All of them will be explained in detail in part 3.

In my opinion, the idea of the proposed method is quite simple which mainly used ideas of the existed algorithms like DQN, DDQN and the extended version of DDPG. However, the depicted result is very promising, even there is an confusing part in the case of comparison to the Soft Actor Critic (SAC), Haarnoja et al. [2018], which will be shown in the part 3.

2 Problem Formulation

The notations used in this report are:

- $s \in \mathcal{S}$ defines the states
- $a \in \mathcal{A}$ defines the actions
- $\pi : \mathcal{S} \rightarrow \mathcal{A}$ defines the policy
- γ represents the discount factor
- $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(S_i, a_i)$ is the discounted sum of rewards
- π_ϕ is the policy in the continuous action space with parameters ϕ
- $J(\phi) = \mathbb{E}_{s_i \sim p_\pi, a_i \sim \pi}[R_0]$ is the objective function
- $\nabla_\phi J(\phi) = \mathbb{E}_{s \sim p_\pi}[\nabla_a Q^\pi(s,a)|_{a=\pi(s)} \nabla_\phi \pi_\phi(s)]$ defines the deterministic policy gradient method
- $Q^\pi(s,a) = \mathbb{E}_{s_i \sim p_\pi, a_i \sim \pi}[R_t|s,a]$ defines the action-value function following policy π
- $\rho^{\pi(s)}$ represents the state distribution, following policy π

Preprint. Under review.

Initially, the overestimation problem is found in the Q learning, Watkins and Dayan [1992] method where it apply max operator to determine the target as shown in the following equation:

$$Q_{t+1}(S_t, a_t) = (1 - \alpha_t(s_t, a_t))Q_t(s_t, a_t) + \alpha_t(s_t, a_t)[r_t + \gamma \max_b(Q_t(s_{t+1}, b))] \tag{1}$$

In the case of tabular setting, the positive bias has been overcome using Double Q-Learning, Hasselt [2010] that applies double estimator to separate the action selection and action evaluation that will replace target in Q-Learning to the form of:

$$Y_t^{DoubleQ} \equiv R_{t+1} + \gamma Q(S_{t+1}, argmax_a Q(S_{t+1}, a; \boldsymbol{\theta_t}); \boldsymbol{\theta_t'}) \tag{2}$$

While the target in Q-Learning can be written as:

$$Y_t^{Q} \equiv R_{t+1} + \gamma Q(S_{t+1}, argmax_a Q(S_{t+1}, a; \boldsymbol{\theta_t}); \boldsymbol{\theta_t}) \tag{3}$$

The success of Double Q-Learning is extended to the large-scale function approximation, called Double Deep Q Network (DDQN)Van Hasselt et al. [2016] which compared to the standard DQN, the target can be written as:

$$Y_t^{DoubleDQN} \equiv R_{t+1} + \gamma Q(S_{t+1}, argmax_a Q(S_{t+1}, a; \boldsymbol{\theta_t}); \boldsymbol{\theta_t^-}) \tag{4}$$

$$Y_t^{DQN} \equiv R_{t+1} + \gamma Q(S_{t+1}, \max_a Q(S_{t+1}, a; \boldsymbol{\theta_t^-})) \tag{5}$$

In the case of high-dimensional and continuous action spaces, the positive bias is found in the Deterministic Policy Gradient (DPG) and DDPG which can be shown as follow, assume that $Q_\theta(s, \pi_\phi) = Q^\pi(s.\pi_\theta)$ (the approximate value function is equal to the true value function), in expectation over the steady-state distribution, we can obtain the following equation in the direction of the true policy update:

$$\begin{aligned} &\mathbb{E}_{s\ \pi}[Q_\theta(s, \pi_{new}(s))] = \mathbb{E}_{s\ \pi}[Q^\pi(s, \pi_{new}(s))] \\ &\forall \phi_{new} \in [\phi, \phi + \beta(\pi_{true} - \pi)] \text{ such that } \beta > 0 \end{aligned} \tag{6}$$

Since ϕ_{true} maximizes the rate of change of true value, $\Delta^\pi_{true} \geq \Delta^\pi_{approx}$ and $\Delta^\theta_{approx} \geq \Delta^\pi_{true}$ and given the equation above, then the overestimation is clearly stated as $Q_\theta(s, \pi_{approx}(s)) \geq Q^\pi(s, \pi_{true}(s)) \geq Q^\pi(s, \pi_{approx}(s))$

Beside the positive bias problem, another important issue addressed in this problem is the high variance estimates resulting in a noisy gradient for the policy update. It emerges as the Bellman equation is never exactly satisfied in the function approximation setting that each update of the Q-value can be expressed as

$$Q_\theta(s, a) = r + \gamma \mathbb{E}[Q_\theta(s', a')] - \delta(s, a) \tag{7}$$

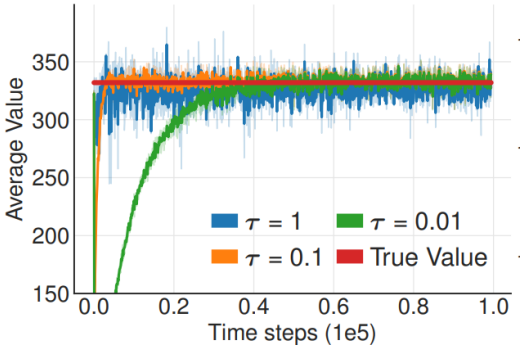

Figure 1: Average estimated value of randomly selected Hopper-v1 under fixed policy

The variance of the estimated value will be proportional to the variance of the future reward and the

estimation error (TD-error) $\delta(s, a)$. Therefore, if the error of each update is not reduced, the error will be accumulated making variance can grow rapidly given large γ. In this paper, the practical proof of the emergence of variance is shown by comparing soft target network update, $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$ with the standard update like in the DQN, $\theta' \leftarrow \theta$.The result shows that without soft target update, the average estimated value will be very volatile under fixed policy as shown in the Figure 1.

3 Theoretical Analysis

In this section, The main contribution of TD3 comprising of Clipped Double Q-Learning, delayed target and policy network update and target policy smoothing regularization is explained. Next, the proof of the convergence of Clipped Double Q-Learning which is one of the foundation of the TD3. Furthermore, The result comparison of the proposed algorithm is shown which slightly different with the result shown in the paper.

Clipped Double Q-Learning for Actor-Critic

The implementation of Clipped Double Q-Learning is highly motivated to overcome over-estimation bias in the critic network. Therefore, the target is modified to the form of

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \pi_{\phi'}) \tag{8}$$

by taking the minimum value, the value target may induce an underestimation bias, which more preferable, instead of overestimation bias. The proof of convergence of Clipped Double Q-Learning is highly motivated by a proof introduced in Singh et al. [2000] for showing the convergence of SARSA(0) Rummery and Niranjan [1994] (This proof is also used in the Double Q-Learning convergence proof).

Lemma 1. Consider a stochastic process $(\alpha_t, \Delta_t, F_t, t \geq 0)$ where $\alpha_t, \Delta_t, F_t : X \rightarrow \mathbb{R}$ satisfy the equation:

$$\Delta_{t+1}(x_t) = (1 - \alpha_t(x_t))\Delta_t(x_t) + \alpha_t(x_t)F_t(x_t) \tag{9}$$

where $x_t \in X$ and $t = 0, 1, 2, ...$ Let P_t be a sequence of increasing σ-fields such that α_0 and Δ_0 and P_0-measurable and α_t, Δ_t, and F_{t-1} are P_t-measurable, $t = 1, 2, ...$ Assume that the following hold:

1. The set X is finite
2. $\alpha_t(x_t) \in [0, 1], \sum_t \alpha_t(x_t) = \infty, \sum_t(\alpha_t(x_t))^2 < \infty$ with probability 1 and $\forall x \neq x_t$: $\alpha(x) = 0$
3. $\| \mathbb{E}[F_t \mid P_t] \| \leq \kappa \| \Delta_t \| + c_t$ where $\kappa \in [0, 1)$ and c_t converges to 0 with probability 1
4. Var $[F_t(x_t \mid P_t] \leq K(1 + \kappa \| \Delta_t \|)^2$, where K is some constant

where $\| \cdot \|$ denotes the maximum norm. Then Δ_t converges to 0 with probability 1.

Theorem 1. Given the following conditions:

1. Each state action pair is sampled an infinite number of times.
2. The MDP is finite
3. $\gamma \in [0, 1)$.
4. Q values are stored in a lookup table
5. Both Q^A and Q^B receive an infinite number of updates
6. The learning rates satisfy $\alpha_t(s, a) \in [0, 1], \sum_t \alpha_t(s, a) = \infty, \sum_t(\alpha_t(s, a))^2 < \infty$with probability 1 and $\alpha_t(s, a) = 0, \forall(s, a) \neq (s_t, a_t)$.

Then Clipped Double Q-Learning will converge to the optimal value function Q^*, as defined by the Bellman optimality equation, with probability 1.

The form of Lemma 1 is actually follows Q-learning‘s update rule, equation (2), but it puts attention to the difference between estimated Q value with the optimal Q value.

Actually, this kind of way of proving the convergence of the proposed algorithm has been also introduced in the Double Q-Learning paper. However, since in this paper clipped Double Q-Learning is used, the proof of Theorem 1 must follows the rule of clipped version of Double Q-Learning, that is shown below.

Proof of Theorem 1. Let $P_t = Q_0^A, Q_0^B, s_0, a_0, \alpha_0, r_1, s_1, ..., s_t, a_t, X = S \times A, \Delta_t = Q_t^A - Q^*$

Defining a^* =argmax$_a$Q$^A(s_{t+1}, a)$ and using Clipped Double Q-Learning equation, we can form:

$$\begin{aligned}\Delta_{t+1}(s_t, a_t) &= (1 - \alpha_t(s_t, a_t))(Q_t^A(s_t, a_t) - Q^*(s_t, a_t)) \\ &\quad + \alpha_t(s_t, a_t)(r_t + \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - Q^*(s_t, a_t)) \\ &= (1 - \alpha_t(s_t, a_t))\Delta_t(s_t, a_t) + \alpha_t(s_t, a_t)F_t(s_t, a_t),\end{aligned} \tag{10}$$

where we have defined $F_t(s_t, a_t)$ as:

$$\begin{aligned}F_t(s_t, a_t) &= r_t + \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - Q_t^*(s_t, a_t) \\ &= r_t + \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - Q_t^*(s_t, a_t) \\ &\quad + \gamma Q_t^A(s_{t+1}, a^*) - \gamma Q_t^A(s_{t+1}, a^*) \\ &= F_t^Q(s_t, a_t) + c_t\end{aligned} \tag{11}$$

where $F_t^Q = r_t + \gamma Q_t^A(s_{t+1}, a^*) - Q_t^*(s_t, a_t)$ denotes the value of F_t under standard Q-Learning (and also corresponding F_t in Lemma 1.) and $c_t = \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - \gamma Q_t^A(s_{t+1}, a^*)$. From condition 2 and 6 in Lemma 1 and Theorem 1 respectively, we can define $F_t(s, a) = F_t^Q(s, a) = C_t(s, a) = 0$ if $(s, a) \neq (S_t, a_t)$. It is also well-known from Q-Learning algorithm, $\| \mathbb{E}[F_t^Q(\cdot, \cdot) \mid P_t] \| \leq \gamma \| \Delta_t \|$ for all t where $\| \cdot \|$ is the maximum norm. Then the condition 3 in the Lemma 1 holds if it can be shown that c_t converges to 0 with probability 1.

Let $y_t = r_t + \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*))$ and $\Delta_t^{BA}(s_t, a_t) = Q_t^B(s_t, a_t) - Q_t^A(s_t, a_t)$ where c_t converges to 0 if Δ^{BA} converges to 0. the update of Δ_t^{BA} at time t is the sum of updates of Q^A and Q^B :

$$\begin{aligned}\Delta_{t+1}^{BA}(s_t, a_t) &= \Delta_t^{BA}(s_t, a_t) + \alpha(s_t, a_t)(y - Q_t^B(s_t, at) - (y - Q_t^A(s_t, at))) \\ &= \Delta_t^{BA}(s_t, a_t) + \alpha(s_t, a_t)(Q_t^B(s_t, at) - Q_t^A(s_t, at)) \\ &= (1 - \alpha_t(s_t, a_t))\Delta_t^{BA}(s_t, a_t)\end{aligned} \tag{12}$$

Clearly $\Delta_t^{BA}(s_t, a_t)$ will converge to 0, which implies that condition 3 of Lemma 1 is satisfied, by set $Q^A(s_t, a_t)$ converges to the optimal action value, $Q^*(s_t, a_t)$. Using similar way, we can proof the convergence of $Q^B(s_t, a_t)$ by defining $\Delta_t = Q_t^B - Q^*$

Delayed Target Network and Policy update

The idea of delay update both in the policy and the target networks come from the interplay between the high variance estimate and the policy performance in actor critic setting. The value estimates will be fluctuated badly as the policy is poor, and the policy becomes poor as the value estimates is inaccurate. In order to handle this issue, TD3 extend the idea of soft target update in DDPG to the delayed update which obtained from DQN. The delay update will reduce the likelihood of repeating updates with respect to an unchanged critic. Furthermore, the delayed update can be said to let the target to provide small value estimates error first before being updated.

Target Policy Smoothing Regularization

In the deterministic policy gradient algorithm, the policy is updated following the gradient of Q which in detail, $\phi^{k+1} = \phi^k + \mathbb{E}_{s \sim \rho^{\pi^k}}[\nabla_\theta Q^{\pi^k}(s, \pi_\phi(s))]$. Therefore, it is likely to overfit to the narrow peaks in the value estimate. In order to solve this problem, the target policy smoothing is proposed by adding a clipped noise in the target action.

$$y = r + \gamma Q_{\theta'}(s', \pi_{\phi'}(s') + \epsilon),$$
$$\epsilon \sim clip(\mathcal{N}(0, \sigma), -\mathrm{c}, \mathrm{c}) \tag{13}$$

The intuition of introducing clipped noise in the target is that similar action should have similar value.

The Overall Algorithm of TD3

Algorithm 1 TD3

Initialize critic networks $Q_{\theta_1}, Q_{\theta_2}$ and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $Q'_{\theta_1} \leftarrow Q_{\theta_1}, Q'_{\theta_2} \leftarrow Q_{\theta_2}, \phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ ***to*** T **do**
  select action with exploration noise $a \sim \pi_\phi(s) + \epsilon, \epsilon \sim (\mathcal{N}(0, \sigma)$ and observe reward r and new state s'
  store transition tuple (s, a, r, s') in $\mathcal{B}$
  sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s') + \epsilon, \epsilon \sim clip(\mathcal{N}(0, \sigma), -\mathrm{c}, \mathrm{c})$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
  update critics $\theta_i \leftarrow \mathrm{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  if t *mod* d **then**
    update ϕ by deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum (\nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    update target networks:
    $\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$
    $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
  end
end

The Experiment Result

The result shown in this result is limited only in the HalfCheetah-v2 environment in the MuJoCo continuous control tasks . The hyperparameters and network configuration are set to be identical as in the provided code on the github. Meanwhile, the Soft Actor Critic (SAC) algorithm is set to be identical to the paper setting with 1 iteration training per time step. From the Figure 2, we can see that the performance of SAC outperforms both of TD3 and DDPG. This ressult is clearly different with result provided in the paper, where the performance of SAC much worse compared to the TD3. My assumption related to this result is that the underestimation suffered by TD3 affects the learning process, leading to sub-optimal solution.

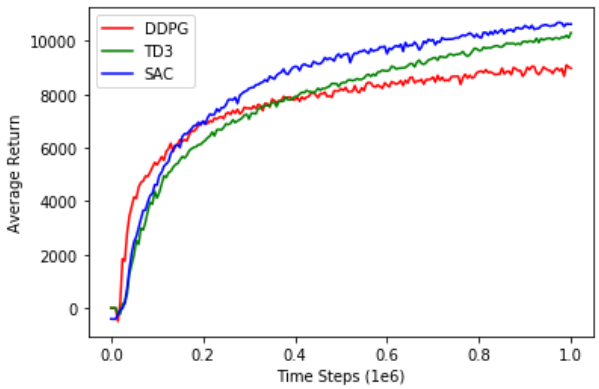

Figure 2: Learning Curve of the HalfCheetah-v2

4 Conclusion

This paper provides a solution in tackling overestimation in the high-dimensional continuous action control space by using a method called TD3 (Twin Delayed Deep Deterministic Policy Gradient) algorithm. It mainly consists of three steps, Clipped Double Q-Learning, delayed target and policy network update, and target policy smoothing regularization. Although the paper shows very promising result compared to DDPG and SAC, in my experiment it is found that the SAC can outperform TD3 in the HalfCheetah-v2 environment.It indicates that the possibility of underestimation affects the learning process in TD3 algorithm.

Therefore, providing clear evidence of underestimation in TD3 accompanied by a rigid solution must be conducted in the future. Moreover, this algorithm can also be extended into model-based algorithm for tackling hard task like in the Ant and humanoid environment, which have very huge state and action dimension.

References

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. 2014.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Hado V Hasselt. Double q-learning. In *Advances in neural information processing systems*, pages 2613–2621, 2010.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3):287–308, 2000.

Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.