控制与决策
Control and Decision

基于组合网络优化的延迟深度确定性策略梯度

程玉虎, 安冰清, 孔毅

引用本文:
程玉虎, 安冰清, 孔毅. 基于组合网络优化的延迟深度确定性策略梯度[J]. 控制与决策, 2025, 40(3): 1015-1023.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2024.0147

基于组合网络优化的延迟深度确定性策略梯度

程玉虎，安冰清，孔 毅†

(中国矿业大学 信息与控制工程学院，江苏 徐州 221116)

摘 要: 值函数估计偏差修正已成为深度强化学习领域的一个重要研究方向. 现有大多数研究工作均聚焦于如何缓解高估偏差, 却忽略了缓解高估偏差过程中引入的低估偏差问题. 为此, 通过在 Actor-Critic 框架中灵活设置多个 Actor 和 Critic 网络来缓解值函数低估偏差, 提出一种基于组合网络优化的延迟深度确定性策略梯度 (D3PG-CNO). D3PG-CNO 的主要思路为: 在经验收集阶段用一个 Critic 网络对多个 Actor 网络的输出动作进行评估, 并选择最优的动作存入经验池. 在经验训练阶段, 从多个 Critic 网络中选出在当前状态-动作对下估计结果最小的 Critic 网络, 并用其对多个 Actor 网络的输出动作进行评估, 选择评估最大值进行目标值的计算. MuJoCo 平台上的实验结果显示, 相比于现有的确定性策略梯度算法, D3PG-CNO 显著降低了估计偏差, 提高了算法的稳定性和收敛速度, 并在多个任务中表现出更好的性能.

关键词: 深度强化学习；低估偏差；确定性策略梯度；Actor-Critic 框架；值函数

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.0147

引用格式: 程玉虎, 安冰清, 孔毅. 基于组合网络优化的延迟深度确定性策略梯度 [J]. 控制与决策, 2025, 40(3): 1015-1023.

Delayed deep deterministic policy gradient based on combinatorial network optimization

CHENG Yu-hu，*AN Bing-qing*，*KONG Yi*†

(School of Information and Control Engineering， China University of Mining and Technology， Xuzhou 221116， China)

Abstract: In recent years, value function estimation bias correction has become an important research direction in the field of deep reinforcement learning. Most existing research work focuses on how to alleviate overestimation bias, but ignores the problem of underestimation bias introduced in the process of mitigating overestimation bias. To this end, this paper flexibly sets up multiple Actor and Critic networks in the Actor-Critic framework to alleviate the value function underestimation bias, and proposes a delayed depth deterministic policy gradient based on combined network optimization (D3PG-CNO). The main idea of the D3PG-CNO is to use a Critic network to evaluate the output actions of multiple Actor networks in the experience collection phase, and to select the optimal actions to store in the experience pool. In the experience training stage, the Critic network with the smallest estimated result under the current state-action pair is selected from multiple Critic networks and used to evaluate the output actions of multiple Actor networks, and the maximum evaluation value is selected to calculate the target value. Experimental results on the MuJoCo platform show that the D3PG-CNO significantly reduces estimation bias compared to existing deterministic policy gradient algorithms, improves the stability and convergence speed of the algorithm, and shows better performance in multiple tasks.

Keywords: deep reinforcement learning； underestimation bias； deterministic policy gradient； Actor-Critic framework；value function

0 引 言

近年来, 随着深度学习 (deep learning, DL)[1] 技术的不断发展, 强化学习[2-3] 与深度学习相结合产生了深度强化学习 (deep reinforcement learning, DRL)[4]. 相较于传统强化学习, 深度强化学习运用神经网络[5] 使智能体在处理复杂任务时能够表现出更强的学习能力和泛化能力[6-7], 在自动驾驶[8]、棋牌博弈[9]、机器人控制[10-11] 等领域均取得了显著成效.

收稿日期: 2024-02-06；**录用日期:** 2024-07-15.
基金项目: 国家自然科学基金项目 (62176259, 62006232).
责任编委: 易建强.
†通信作者. E-mail: kongyicumt@163.com.

虽然深度强化学习在诸多领域取得了巨大突破, 但由于应用场景的复杂性和多样性, 深度强化学习研究中仍然存在许多亟待解决的问题, 其中估计偏差问题[12] 是当下的热门研究方向之一. 深度强化学习中的估计偏差通常分为低估和高估两类. 高估偏差在许多传统的 Q 学习算法中普遍存在, 如深度 Q 网络 (deep Q network, DQN)[13] 和深度确定性策略梯度算法 (deep deterministic policy gradient, DDPG)[14]. 高估偏差主要产生的原因是 Q 学习在进行策略优化时, 每次都会贪婪地选取动作以得到更高的回报, 且该偏差将随着策略的更新而不断累积. 现有研究表明, 高估偏差将会导致智能体在决策时做出次优甚至失败的决策, 这对策略的优化带来了巨大的挑战. 因此, 解决估计偏差问题在深度强化学习中至关重要. 为了解决高估偏差问题, van Hasselt 等[15] 提出了基于深度强化学习的 double Q-learning (deep reinforcement learning with double Q-learning, DDQN), 将 double Q-learning[16] 与深度神经网络相结合, 使用两个独立的值估计器来获取 Q 值的无偏估计, 使 DDQN 既能减少高估偏差, 又能处理高维、复杂状态空间的任务; Duan 等[17] 提出了分布式软 Actor-Critic 算法, 该算法在最大熵强化学习中嵌入状态-动作回报的分布函数, 通过学习回报分布减少了估计偏差; Fujimoto 等[18] 提出了双延迟深度确定性策略梯度算法 (twin delayed deep deterministic policy gradient, TD3), 通过对两个值函数逼近器取最小值、延迟更新策略网络和目标策略平滑正则化等方式来解决高估偏差问题; Lan 等[19] 提出了使用多个值估计器对 double Q-learning 进行优化的 maxmin Q-learning, 该算法提供了一个参数灵活调控误差, 并从理论上证明了存在一种可以最大程度减轻估计偏差的参数选择.

尽管许多算法在解决高估偏差上取得了显著的成功, 但却带来了低估偏差问题, 即估计动作值函数时容易使估计值低于真实值. 如 TD3 可以抑制累积误差, 但对两个值函数逼近器取最小值的操作会导致对 Q 值估计的低估偏差, 这也会导致不稳定的训练过程、算法性能的下降和次优策略的选择. 为了缓解低估偏差, He 等[20] 提出了加权深度确定性策略梯度 (weighted delayed deep deterministic policy gradient, WD3) 算法, 通过修改 TD3 中 Q 网络的更新规则来缓解低估偏差; Wu 等[21] 提出了三平均深度确定性策略梯度算法, 通过将 Q 网络的数量增加到 3 个, 进一步扩展了 WD3; Wei 等[22] 提出了准中值算法, 通过选择 Q 网络的中位数代替 TD3 中的两个值函数逼近器以解决低估问题.

本文尝试将 maxmin Q-learning 中多个 Q 网络灵活控制的思想与 Actor-Critic 框架相结合, 以此来缓解 TD3 存在的值函数低估问题, 并提高算法的收敛速度、稳定性和性能水平. 主要工作如下:

1) 提出一种基于组合网络优化的延迟深度确定性策略梯度 (D3PG-CNO), 通过在 Actor-Critic 框架中设置多个 Actor 和 Critic 网络缓解值函数低估偏差. 从 N 个 Critic 网络中选出在当前状态-动作对 (s,a) 下估计结果最小的 Critic 网络, 并用其对 M 个 Actor 网络的输出动作进行最大值选择, 对经验训练阶段的目标值计算进行优化, 从而确保目标值的准确性.

2) 为提高经验池中数据的质量, D3PG-CNO 在经验收集阶段采用一种基于多 Actor-单 Critic 网络的评估机制, 利用单个 Critic 网络对 M 个 Actor 网络的输出动作进行评估, 并将评估结果最优的动作存入经验池. 同时, 为增强智能体探索能力, 在动作选择过程中引入噪声机制.

3) 在 MuJoCo 平台上对所提算法开展机器人仿真控制任务的实验, 结果表明: D3PG-CNO 能够显著降低值函数低估偏差, 并提高算法稳定性和收敛性. 与现有的深度确定性策略梯度算法相比, D3PG-CNO 能获得更高的回报. 此外, 通过灵活调节 Actor 和 Critic 网络数量, 进一步增强了算法性能.

1 背 景

强化学习通常可以用马尔可夫决策过程 (Markov decision process, MDP) 来描述, MDP 由五元组 $(\mathcal{S},\mathcal{A},P,\gamma,\mathcal{R})$ 构成. 其中: $\mathcal{S}$ 是状态空间, $\mathcal{A}$ 是动作空间, $P(s'|s,a)$ 是状态转移概率, γ 是折扣因子, $\mathcal{R}:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ 是奖励函数. 在每个时间步 t 上, 给定状态 $s\in\mathcal{S}$, 智能体会根据策略 $\pi:\mathcal{S}\to\mathcal{A}$ 来选择动作 $a\in\mathcal{A}$, 同时获得该时间步的奖励 r 并根据状态转移概率得到下一状态 s'. 回报定义为累计奖励的折扣总和: $R_t=\sum\limits_{i=t}^{T}\gamma^{i-t}r(s_i,a_i)$, 其中 T 表示终止时刻. 在策略为 π, 状态为 s, 采取动作 a 时的动作值函数定义为 $Q^{\pi}(s,a)=\mathbb{E}_{s_i\sim p_\pi,a_i\sim\pi}[R_t|s,a]$, 用于评估 π 的好坏. 在 Q 学习中, 通过贝尔曼方程递归地估计动作值函数:

$$Q^{\pi}(s,a)=r+\gamma\mathbb{E}_{s',a'\sim\pi(s')}[Q^{\pi}(s',a')]. \tag{1}$$

强化学习的目的是通过最大化回报 $J(\phi)=\mathbb{E}_{s_i\sim p_\pi,a_i\sim\pi}[R_0]$ 来获取最佳策略 π^*. 连续控制的强化学习算法在策略函数更新时, 可以通过对值函数采

取梯度上升算法进行更新, 这样的更新方式被称为确定性策略梯度算法, 其回报梯度的表达式为

$$\nabla_\phi J(\phi)=\mathbb{E}_{s\sim p_\pi}[\nabla_a Q^\pi(s,a)|_{a=\pi_\phi(s)}\nabla_\phi\pi_\phi(s)], \quad (2)$$

其中ϕ为策略π_ϕ的参数.

在高维连续状态空间中, 可以用函数逼近器$Q_\theta(s,a)$来估计动作值. Q学习[13]引入时序差分算法 (temporal difference, TD), 结合动态规划和蒙特卡罗方法的思想, 通过观察序列状态和奖励来逐步更新动作值函数. 函数逼近器的目标值如下:

$$y=r+\gamma Q_{\bar\theta}(s',a'),\ a'\sim\pi_{\bar\phi}(s'). \quad (3)$$

其中: 动作 a' 从目标策略网络 $\pi_{\bar\phi}$ 中选取. $Q_{\bar\theta}$为目标Q网络, 其参数 $\bar\theta$ 以一定的比例 η 进行更新: $\bar\theta\leftarrow\eta\theta+(1-\eta)\bar\theta$. 该更新方法可以用于异策略算法, 通过从经验池 $\mathcal{D}$ 中进行小批量采样后最小化损失函数来更新动作值网络参数. 损失函数定义为

$$L(\theta)=\mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[(y-Q_\theta(s,a))^2]. \quad (4)$$

2　估计偏差

在 Actor-Critic 框架中, TD3[18]为缓解高估问题构造了两个独立的值函数逼近器: Q_{θ_1}和Q_{θ_2}. 在进行Q网络更新时, 使用下一状态下两个值函数逼近器之间的最小值来计算目标值

$$y=r+\gamma\min_{i=1,2}Q_{\bar\theta_i}(s',\pi_{\bar\phi}(s')). \quad (5)$$

在两个值函数逼近器逐步逼近假设的真实值Q^π的过程中, 由函数近似引起的估计偏差为

$$e^i_{s',a'}=Q_{\bar\theta_i}(s',\pi_{\bar\phi}(s'))-Q^\pi(s',\pi_{\bar\phi}(s')). \quad (6)$$

将上述偏差建模为独立的高斯分布$\mathcal{N}(\varepsilon_i,\sigma_i)$. 对两个估计器$Q$值取最小值后的误差为

$$Z=(r+\gamma\min_{i=1,2}Q_{\bar\theta_i}(s',\pi_{\bar\phi}(s')))-(r+\gamma Q^\pi(s',\pi_{\bar\phi}(s')))=\gamma\min(e^1_{s',a'},e^2_{s',a'}), \quad (7)$$

其中$e^1_{s',a'}$和$e^2_{s',a'}$分别为两个估计器的估计偏差.

由于策略网络是延迟更新的, 可以将值函数逼近器与策略网络视为解耦, 所以两估计误差的均值相近, 即$\varepsilon_1-\varepsilon_2\approx0$. 那么在两个相关高斯随机变量的最小值一阶矩处, 逼近误差变为

$$\mathbb{E}[Z]=\gamma\Big(\frac{\varepsilon_1+\varepsilon_2}{2}-\frac{\zeta}{\sqrt{2\pi}}\Big). \quad (8)$$

其中: $\zeta=\sqrt{\sigma_1^2+\sigma_2^2-2\rho\sigma_1\sigma_2}$, ρ为两个误差分布之间的相关系数. 因为两个函数逼近器在学习过程中使用相同的经验池, 所以两个误差分布是相关的. 当$\sigma_1,\sigma_2>(\pi/(1-\rho))^{\frac{1}{2}}\varepsilon_1$时, 逼近误差将会小于零, 这进一步导致了低估现象.

虽然延迟策略更新和值估计的最小化操作是为了减少误差增长, 但是由于值函数估计的方差与未来奖励估计误差的方差成正比, 值函数估计的方差并没有被消除. 随着智能体与环境的交互, 值函数估计的方差会不断增长. 假设两个值函数逼近器估计误差的期望都是常数, 过大的方差必然会产生低估误差. 在 Actor-Critic 框架中, 虽然消除累计误差的低估偏差比高估偏差更可取, 但是低估可能会导致智能体在学习过程中较为保守, 从而影响对最优策略的探索.

3　D3PG-CNO

针对估计偏差问题, maxmin Q-learning[19]设置N个值函数估计器Q_{θ_i}, 在目标更新时, 首先针对当前状态下的每个状态-动作对, 在N个值函数估计器中选出对应最小状态-动作值的值函数估计器, 再对该值函数估计器取其对应动作的最大值:

$$\max_{a'}[\min_{i\in\{1,\dots N\}}Q_{\theta_i}(s',a')].$$

该算法证明存在一种可以实现无偏估计的参数选择, 使估计偏差和方差小于 Q-learning. 实验表明, 该算法能够较好地控制估计偏差, 同时具备良好的性能.

为减小 Actor-Critic 框架的估计偏差, 本文结合 maxmin Q-learning 的思想提出了 D3PG-CNO. 算法流程如图 1 所示, 伪代码如下所示.

D3PG-CNO算法

初始化 Critic网络、Actor网络、目标网络、经验池

循环 每个回合:

　初始化s

　循环 时间步$t=0\rightarrow T$:

　观察当前状态s_t, 进行动作选择:

$$a_t=\arg\max_{j\in\{1,\dots,M\}}[Q_{\theta_1}(s_t,\pi_{\phi_j}(s_t))]+\varepsilon;$$

　观察下一状态s_{t+1}和奖励r_t, 存入经验池$\mathcal{D}$;

　最小化损失函数更新Critic网络参数θ:

$$Q_{\min}\leftarrow\min_{i\in\{1,\dots,N\}}Q_{\bar\theta_i}(s_t,a_t),$$

$$y_t\leftarrow r_t+\gamma Q_{\min}(s_{t+1},\tilde a),$$

$$\tilde{\mathcal{L}}_{\text{D3PG-CNO}}(\theta_i)=\mathbb{E}_{\{s_t,a_t,r_t,s_{t+1}\}\sim\mathcal{D}}[(y_t-Q_{\theta_i}(s_t,a_t))^2]$$

　如果 t mod $d=0$, 延迟更新Actor网络:

　用确定性策略梯度方法更新参数ϕ:

$$\nabla_{\phi_j}J(\phi_j)=\mathbb{E}_{s_t\sim\mathcal{D}}[\nabla_a Q_{\min}(s_t,a)|_{a=\pi_{\phi_j}(s_t)}\nabla_{\phi_j}\pi_{\phi_j}(s_t)];$$

　更新目标网络:

$$\bar\theta_i\leftarrow\eta\theta_i+(1-\eta)\bar\theta_i,$$

$$\bar\phi_j\leftarrow\eta\phi_j+(1-\eta)\bar\phi_j$$

　结束循环

结束循环

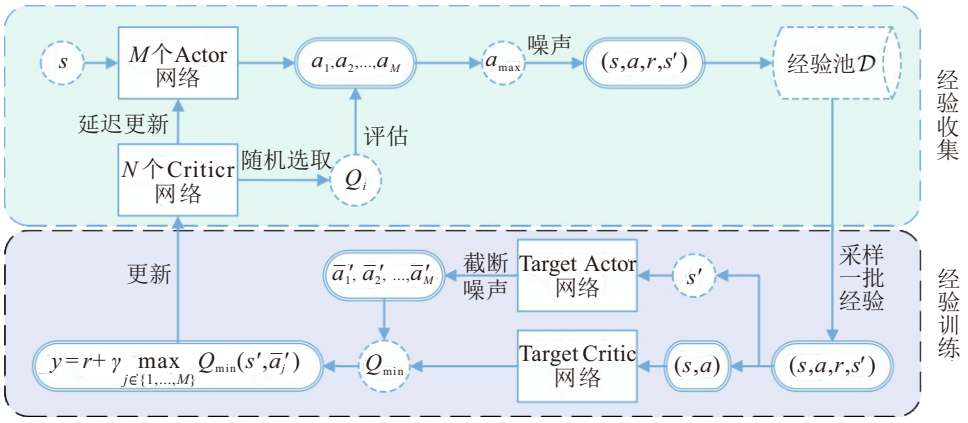


图1 D3PG-CNO 流程

从图 1 可以看出, D3PG-CNO 在 Actor-Critic 框架中灵活设置多个 Actor 和 Critic 网络来缓解值函数低估偏差. D3PG-CNO 与 maxmin Q-learning 的区别在于: 在动作选取方面, maxmin Q-learning 通过遍历 Q 表进行动作选取, 不适用于连续动作空间任务. 而 D3PG-CNO 通过策略网络进行动作选取, 使其不仅能处理离散动作空间任务, 还可以解决连续动作空间任务. 在值函数更新方面, maxmin Q-learning 通过贝尔曼方程递归来更新 Q 值, 这会导致智能体训练过程中的不稳定性, 而 D3PG-CNO 通过采用目标网络和延迟策略更新的方式缓解了这一问题. 此外, maxmin Q-learning 依赖于传统经验回放机制, 而 D3PG-CNO 采用 Actor-Critic 框架, 在经验收集阶段采用一种多 Actor-单 Critic 网络的评估机制, 用一个 Critic 网络对 M 个 Actor 网络的输出动作进行评估, 选择最优的动作存入经验池, 提高了经验池中数据的质量. 同时, D3PG-CNO 在经验训练阶段先从 N 个 Critic 网络中选出在当前状态-动作对 (s,a) 下值函数最小的 Critic 网络, 再用该网络对 M 个 Actor 网络的输出动作进行评估, 选择最大值进行目标值计算, 进一步确保了目标值的准确性.

经验收集阶段的择优存储经验操作使经验池中采样出的数据更优, 有助于后续值函数估计器的选择. 同时, 为了提高智能体探索性, 增设服从高斯分布的随机噪声. 该阶段的动作选取公式为

$$a=\underset{j\in\{1,\dots,M\}}{\arg\max}\left[Q_{\theta_1}(s,\pi_{\phi_j}(s))\right]+\varepsilon, \tag{9}$$

其中 $\varepsilon\sim\mathcal{N}(0,\sigma)$ 为动作的噪声. 因为不同估计器在评估动作时只用于选取最优动作, 而不将该状态-动作对的估计值用于目标网络, 因此不会影响估计器的更新. 随着训练步数的增加, 不同估计器都是用同一目标网络进行更新, 因此不同估计器之间的差异将减小, 不同估计器进行动作选取对最终算法性能影响甚微. 综上, 这里直接选择 Q_{θ_1} 对 M 个 Actor 网络的输出动作进行评估.

目标值的计算公式为

$$y\leftarrow r+\gamma\max_{\tilde{a}}Q_{\min}(s',\tilde{a}). \tag{10}$$

其中: $Q_{\min}$ 和 $\tilde{a}$ 分别为

$$Q_{\min}(s,a)=\min_{i\in\{1,\dots,N\}}Q_{\bar{\theta}_i}(s,a)=Q^{\pi}(s,a)+\min_{i\in\{1,\dots,N\}}e^i_{s,a}; \tag{11}$$

$$\tilde{a}\sim\pi_{\bar{\phi}_j}(s)+\varepsilon,\ j\in\{1,\dots,M\},\quad \varepsilon\sim\mathrm{clip}(N(0,\tilde{\sigma}),-c,c). \tag{12}$$

$Q_{\bar{\theta}_i}$ 为 Critic 目标网络, $\pi_{\bar{\phi}_j}$ 为 Actor 目标网络, $\tilde{\sigma}$ 为噪声的方差, c 为裁剪系数.

将式 (7) 中的 $\min\limits_{i=1,2}Q_{\theta_i}(s',\pi_{\tilde{\phi}}(s'))$ 替换为式 (10) 中的 $Q_{\min}(s',\tilde{a})$, 可得 D3PG-CNO 的估计偏差 Z_N 为

$$Z_N=\gamma\min_{i\in\{1,\dots,N\}}e^i_{s'a'}. \tag{13}$$

这里将估计偏差 $e^i_{s'a'}$ 近似为服从高斯分布 $\mathcal{N}(0,\sigma^2)$ 的无偏估计, 即 $e^i_{s'a'}$ 的概率密度函数 $f(x)$ 和累计分布函数 $F(x)$ 分别为

$$f(x)=\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{x^2}{2\sigma^2}}, \tag{14}$$

$$F(x)=\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{t^2}{2\sigma^2}}\mathrm{d}t=\frac{\mathrm{erf}\left(\dfrac{x}{\sqrt{2\sigma^2}}\right)+1}{2}. \tag{15}$$

由于估计偏差之间是独立同分布的, 因此 $\min\limits_{i\in\{1,\dots,N\}}e^i_{s'a'}$ 的概率密度函数为

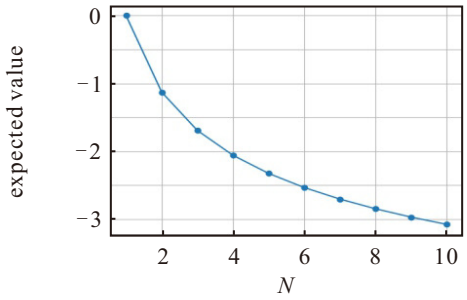
$$f_N(x)=Nf(x)[1-F(x)]^{N-1},$$

可得 $f_N(x)$ 和 Z_N 的期望

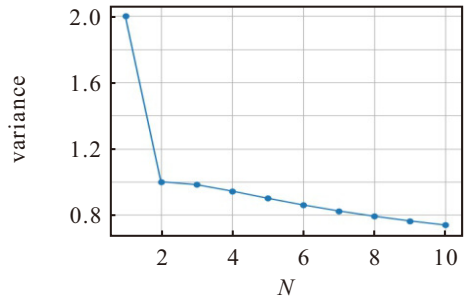
$$f_N(x)=N\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{x^2}{2\sigma^2}}\left[1-\frac{\mathrm{erf}\left(\dfrac{x}{\sqrt{2\sigma^2}}\right)+1}{2}\right]^{N-1}, \tag{16}$$

$$\mathbb{E}[Z_N]=\gamma\int_{-\infty}^{+\infty}xf_N(x)\mathrm{d}x=\frac{\gamma N}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{+\infty}x\mathrm{e}^{-\frac{x^2}{2\sigma^2}}\left[1-\frac{\mathrm{erf}\left(\frac{x}{\sqrt{2\sigma^2}}\right)+1}{2}\right]^{N-1}\mathrm{d}x. \tag{17}$$

当$\gamma=1$, $\sigma=2$时, 该期望随N变化的图像如图 2(a) 所示. 可得, $\mathbb{E}[Z_N]$随着N的增大而减小.



(a) 估计偏差的期望随N的变化



(b) 估计偏差方差随N的变化

图2　Critic 网络数量对估计偏差的影响

在实际应用中, 对于单个估计器而言, 其估计偏差往往大于零. 因此真实的曲线应该整体向上平移一段距离. 这表明, 随着 Critic 网络数量的增加, 对所有值函数估计器取最小的操作会使估计偏差由高估偏差转为低估偏差. $Q_{\min}$的方差看作与$\min\limits_{i\in\{1,\dots,N\}}e^i_{s'a'}$的方差一致, 即

$$\mathrm{Var}[Q_{\min}]=\mathbb{E}[Q_{\min}^2]-\mathbb{E}[Q_{\min}]^2. \tag{18}$$

当$\sigma=2$时, 该方差随N变化的图像如图 2(b) 所示. 可以看出, $\mathrm{Var}[Q^{\min}]$随着N的增大而减小, 并趋于 0. 这表明, 可以通过控制N的数量来控制值估计的方差, 进一步缓解低估偏差.

接下来, 通过定理 1 证明 D3PG-CNO 的收敛性.

引理 1 (Jensen 不等式)　对于任意凸函数$f(x)$, 都有函数值的期望大于等于期望的函数值: $\mathbb{E}[f(x)]\geqslant f(\mathbb{E}[x])$.

定理 1　$|\mathcal{A}|<\infty$, $|\mathcal{R}|<\infty$, 从任意初始Q_0开始, 按照$Q_{l+1}=\tilde{\mathcal{B}}^{\pi}_{\text{D3PG-CNO}}Q_l$重复迭代. 下面将贝尔曼最优算子简写为$\tilde{\mathcal{B}}^{\pi}_{\text{CNO}}$, 其定义为

$$\tilde{\mathcal{B}}^{\pi}_{\text{CNO}}Q=r+\gamma\max_{a'\sim\pi_j(\cdot|s')}\mathbb{E}_{s'\sim p(s'|s,a)}\left[Q^{\pi}_{\min}(s',a')\right]. \tag{19}$$

$\lim\limits_{l\to\infty}Q_l=Q^*$, 其中$l$为迭代步数.

证明 设$Q_1(s,a)$和$Q_2(s,a)$为两个任意的Q函数, 结合引理 1 可以推出

$$\begin{aligned}&|\mathcal{B}^{\pi}_{\text{CNO}}Q_1(s,a)-\mathcal{B}^{\pi}_{\text{CNO}}Q_2(s,a)|=\\&|r+\gamma\max_{a'\sim\pi_j(\cdot|s')}\mathbb{E}_{s'\sim p(s'|s,a)}[Q_1(s',a')]-\\&(r+\gamma\max_{a'\sim\pi_j(\cdot|s')}\mathbb{E}_{s'\sim p(s'|s,a)}[Q_2(s',a')])|=\\&\gamma\max_{a'\sim\pi_j(\cdot|s')}|\mathbb{E}_{s'\sim p(s'|s,a)}[Q_1(s',a')]-\\&\mathbb{E}_{s'\sim p(s'|s,a)}[Q_2(s',a')]|\leqslant\\&\gamma\max_{a'\sim\pi_j(\cdot|s')}\mathbb{E}_{s'\sim p(s'|s,a)}\left[|Q_1(s',a')-Q_2(s',a')|\right]=\\&\gamma\|Q_1(s',a')-Q_2(s',a')\|_{\infty}.\end{aligned} \tag{20}$$

由此可得, $\mathcal{B}^{\pi}_{\text{CNO}}Q$为$\gamma$压缩映射, 存在唯一不动点$Q^*$. 因此, D3PG-CNO 收敛. □

4　实　验

为评估 D3PG-CNO 的性能及其对低估偏差的校正效果, 本文在 MuJoCo 平台下的多个连续控制任务中开展实验测试, 并与 TD3 和 WD3 进行对比.

图 3 展示了 D3PG-CNO 和 TD3 在 8 种测试任务上的估计值和真实值曲线. TD3 的参数设置参照文献 [18], WD3 的参数设置参照文献 [20]; D3PG-CNO 中网络数量取值为$M=2$, $N=2$, 参数设置如表 1 所示, 真实值为当前策略的累计奖励值. 表 2 展示了测试算法在不同任务上最后 10 % 时间步的平均性能, 表 3 展示了不同算法估计值与真实值之间的偏差.

表1　D3PG-CNO 参数设置

设置	参数名称	值
网络结构	Critic 网络隐藏层数量	2
	Critic 网络隐藏层神经元数量	256
	Critic 网络激活函数	ReLU
	Actor 网络隐藏层数量	2
	Actor 网络隐藏层神经元数量	256
	Actor 网络激活函数	ReLU
超参数	Critic 网络学习率	3×10^{-4}
	Actor 网络学习率	3×10^{-4}
	最大时间步长	1×10^{6}
	经验池大小	1×10^{5}
	训练时的批次大小	256
	策略网络探索噪声	0.2
	噪声裁剪系数	0.5
	折扣因子 γ	0.99
	目标网络更新率	0.005

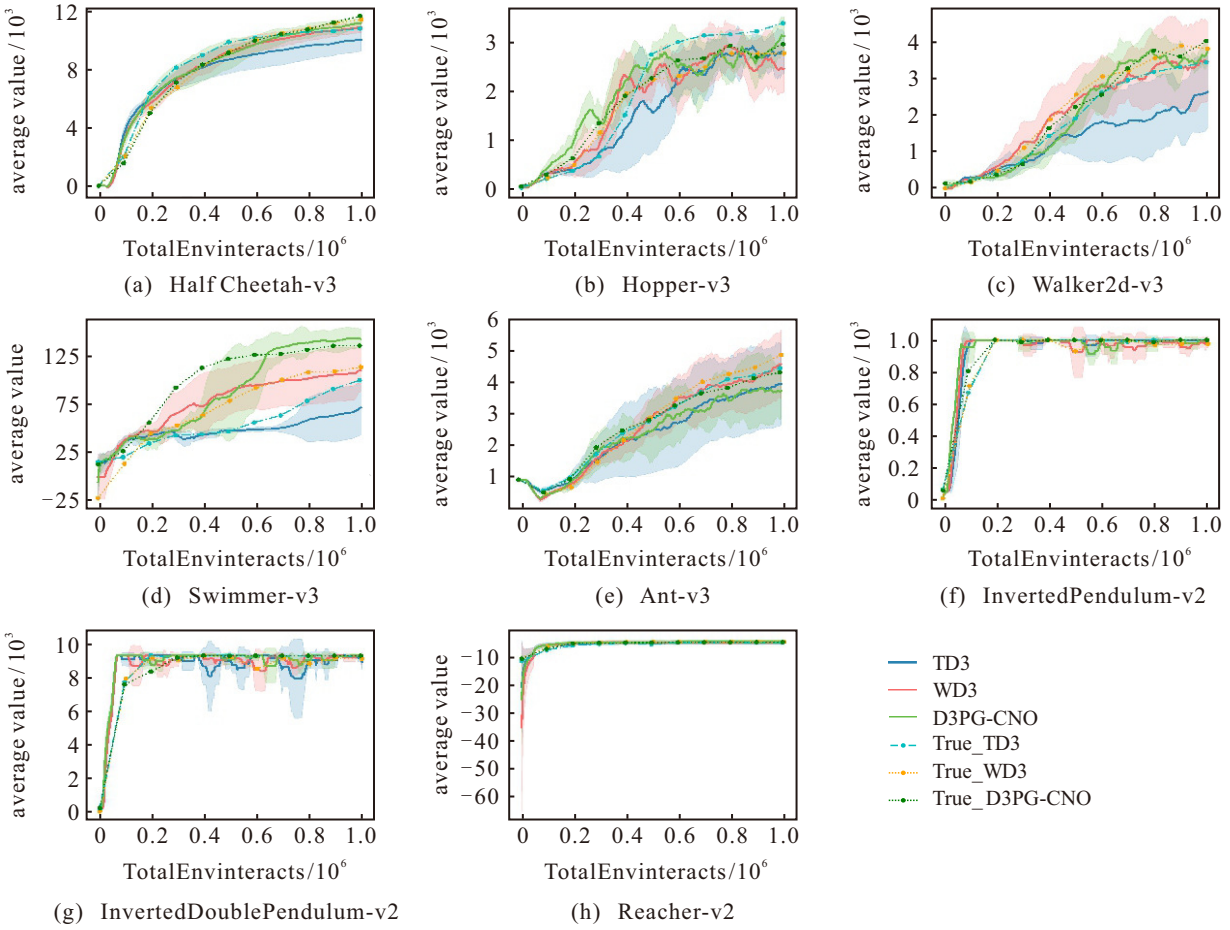


图3 不同算法的回报曲线及真实值

表2 MuJoCo 上最后 10 % 时间步的平均回报

任务名称	TD3	WD3	D3PG-CNO
HalfCheetah-v3	9 972.17	10 807.10	**11 059.59**
Hopper-v3	2 763.36	2 559.94	**2 968.69**
Walker2d-v3	2 574.61	3 364.57	**3 552.42**
Swimmer-v3	68.03	107.76	**142.27**
Ant-v3	3 943.89	**4 493.27**	3 802.99
IPendulum-v2	1 000.00	979.89	**1 000.00**
IDPendulum-v2	9 111.87	9 162.81	**9 295.67**
Reacher-v2	−4.11	−4.12	**−4.04**

表3 算法估计值与真实值之间的偏差

任务名称	TD3 偏差	WD3偏差	D3PG-CNO偏差
HalfCheetah-v3	799.42	576.60	**570.04**
Hopper-v3	618.53	204.79	**−17.4**
Walker2d-v3	853.98	**432.68**	451.34
Swimmer-v3	31.17	**5.11**	−7.29
Ant-v3	570.85	**433.86**	578.23
IPendulum-v2	0.00	−5.37	**0**
IDPendulum-v2	151.39	−47.61	**11.85**
Reacher-v2	−0.22	0.19	**−0.04**

从表 2 和表 3 的数据可以得出, 相较于 TD3 和 WD3, D3PG-CNO 在大多数任务中都有更好的性能和较低的估计偏差. 在 Walker2d-v3 和 Swimmer-v3 任务中, D3PG-CNO 虽然偏差略微逊色于 WD3, 但仍较 TD3 有显著提升, 且 D3PG-CNO 的最终平均回报优于 WD3.

从图 3 的估计值和真实值曲线可以看出:

1) 在图 3(a) ～ 图 3(d) 中, TD3 的估计值均明显低于其真实值, 说明 TD3 中存在较为严重的低估现象. 此外, TD3 的偏差均大于 WD3 和 D3PG-CNO 的偏差, 说明 WD3 和 D3PG-CNO 都有效减小了 TD3 中的估计偏差. 同时, D3PG-CNO 的估计值明显高于 WD3 的估计值, 说明相较于 WD3, D3PG-CNO 对算法性能的提升更加显著.

2) 在图 3(a)、图 3(c)、图 3(d) 中, D3PG-CNO 的平均回报值曲线明显高于 TD3 和 WD3 的平均回报值曲线, 说明 D3PG-CNO 不仅缓解了 TD3 中的低估偏差, 还表明缓解低估偏差会使算法表现出更好的性能.

3) 在图 3(a)、图 3(c)、图 3(e)、图 3(g)、图 3(h) 中, D3PG-CNO 的阴影区域比 TD3 和 WD3 的阴影区域相对较小, 说明 D3PG-CNO 有更小的方差, 也表明通过减小方差和低估偏差, 能使算法有更好的稳定性.

4) 在图 3(b)、图 3(d)、图 3(f) 中, D3PG-CNO 曲线的收敛明显比 TD3 和 WD3 更快, 说明 D3PG-CNO

学习到好策略的用时更短, 进一步表明对低估偏差的缓解能有效提升算法的收敛速度.

图 4 展示了在各任务中, 不同数量的 Actor 网络对应的 D3PG-CNO 估计值的变化, 各分图图例同图 4(a). 表 4 为不同数量的 Actor 网络对应的 D3PG-CNO 在不同任务上最后 10 % 时间步的平均性能.

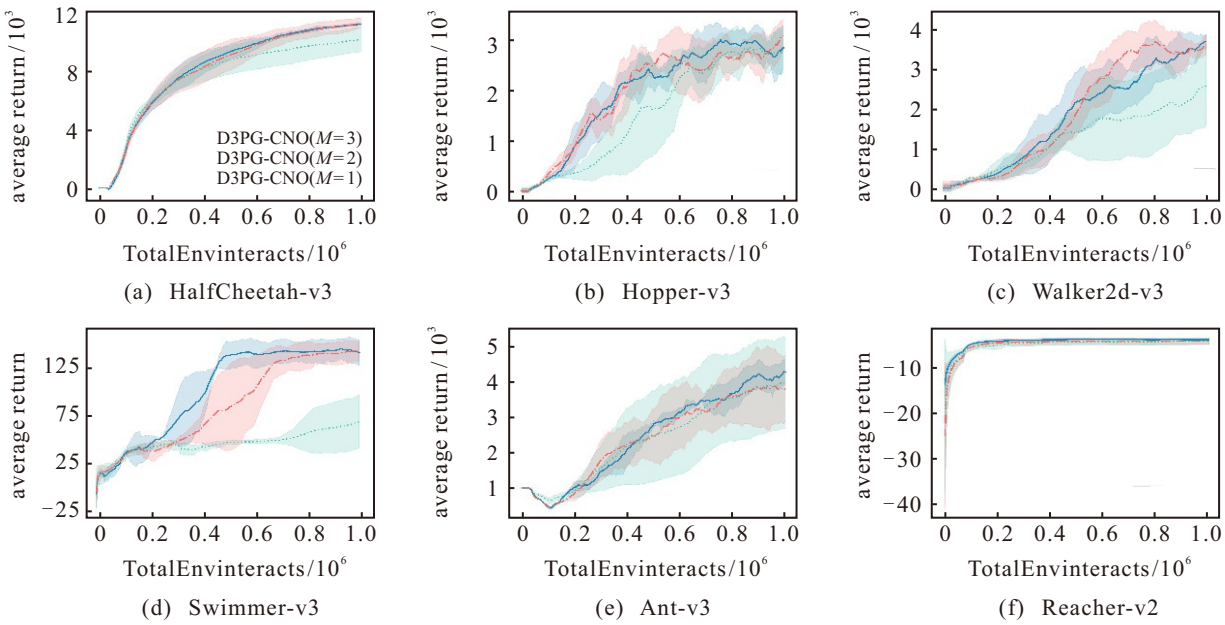


图4　Actor 网络数量对 D3PG-CNO 性能的影响

表4　不同数量 Actor 网络在 MuJoCo 上最后 10 % 时间步的平均回报

Actor 数量	HalfCheetah	Hopper	Walker2d	Swimmer	Ant	Reacher
$M=1$	9972.17	2763.36	2574.61	68.03	3943.89	−4.11
$M=2$	**11059.59**	**2968.69**	3552.42	142.27	3802.99	−4.04
$M=3$	11006.52	2806.69	**3594.72**	**142.44**	**4150.65**	**−3.72**

从图 4 可以看出:

1) 在图 4(a)、图 4(b)、图 4(e)、图 4(f) 中, $M=1$时算法的阴影区域明显大于$M=2$和$M=3$时的阴影区域, 且该阴影区域面积随着M的增大逐渐减小. 说明 D3PG-CNO 的方差与M成反比, 随着M的增大, 算法有更好的稳定性.

2) 在图 4(a) ～ 图 4(d) 中, $M=2$和$M=3$时 D3PG-CNO 的估计值明显高于$M=1$时, 且大多任务中$M=3$的估计值略高于$M=2$时. 说明随着M的增大, 算法表现出了更好的性能.

图 4 和表 4 的数据显示, 在 Actor 网络数量为 2 和 3 时, 算法的性能变化较小. 图 5 和图 6 分别展示了不同 Actor 网络数量和不同 Critic 网络数量的算法运行时间对比. 从图 5 和图 6 可以看出, 随着 Actor 和 Critic 网络数量的增加, 算法的运行时间迅速增长. 这是因为采用了更多的网络数量, 增加了算法的计算成本. 因此, 为了综合考虑算法的时间复杂度, 对比不同 Critic 网络数量对算法性能影响的实验将在$M=2$下进行.

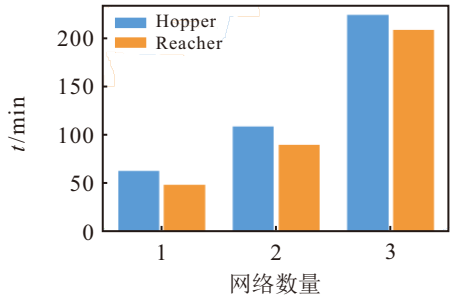


图5　不同 Actor 网络数量运行用时对比

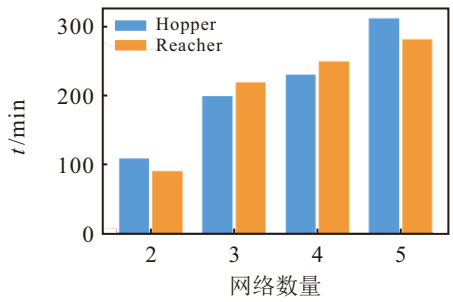


图6　不同 Critic 网络数量运行用时对比

图 7 展示了在不同测试任务中, 不同数量的 Critic 网络对应的 D3PG-CNO 估计值的变化, 各分图图例同图 7(a). 表 5 为不同数量的 Critic 网络对应的 D3PG-CNO 在不同任务上最后 10 % 时间步的平

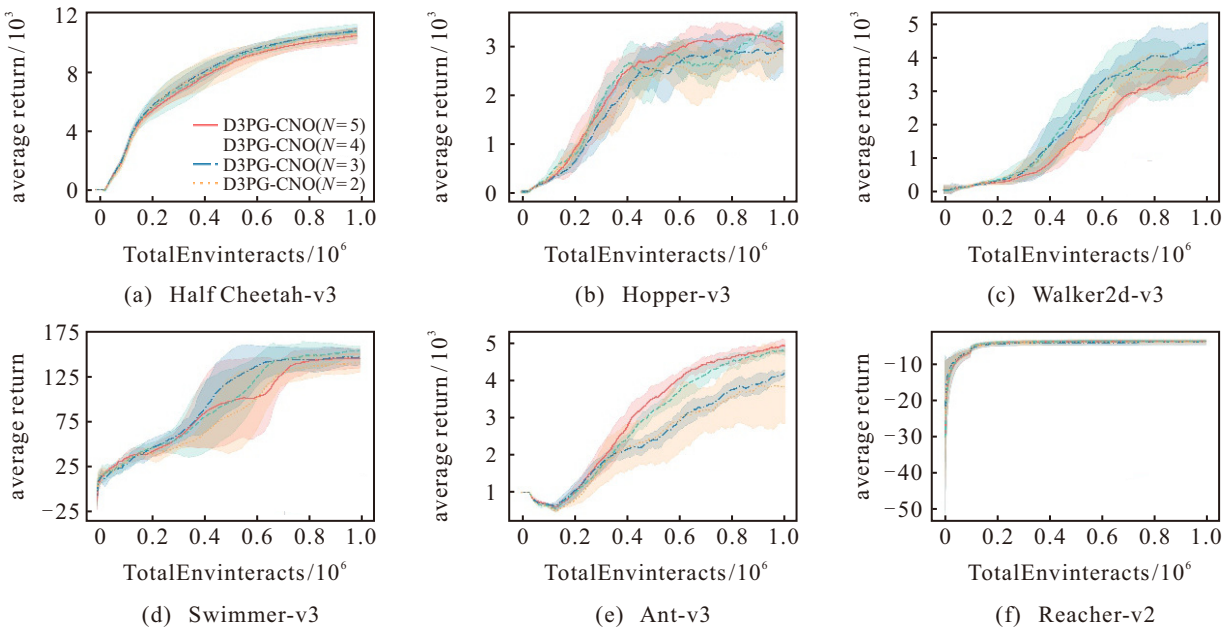


图7 Critic 网络数量对 D3PG-CNO 性能的影响

表5 不同数量 Critic 网络在 MuJoCo 上最后 10 % 时间步的平均回报

Actor 数量	HalfCheetah	Hopper	Walker2d	Swimmer	Ant	Reacher
$N=2$	11059.59	2968.69	3552.42	142.27	3802.99	−4.04
$N=3$	**11264.01**	2969.62	**4424.79**	148.43	4159.97	−4.20
$N=4$	11213.52	**3316.64**	4072.57	**156.25**	4788.22	**−3.88**
$N=5$	10910.33	3079.79	3864.07	147.12	**4884.20**	−3.93

均性能.

从图 7 和表 5 可以看出, 在大多数任务中, D3PG-CNO 的 Critic 网络数量取 3 或 4 时表现最优. 同时, 随着网络数量的继续增加, 算法性能会有所下降. 其中, 在 Ant-v3 任务中算法的表现不同于其他任务. 在该任务上, 随着 Critic 网络数量从 2 到 5 的变化, 算法也表现出了越来越好的性能. 另外, Ant-v3 任务的状态维度和动作维度远高于其他任务. 因此, 此处认为是 Ant-v3 任务的维度过高, 导致了 D3PG-CNO 在该任务下的不同表现.

为了证明经验收集阶段直接选择值函数估计器 Q_{θ_1} 进行动作择优的合理性, 本文针对选择不同函数估计器和随机选择函数估计器进行了对比实验, 如图 8 所示. 从图 8 可以看出, 经验收集阶段选择不同函数估计器训练的最终结果趋于一致, 因此直接选择 Q_{θ_1} 进行动作择优是可行的.

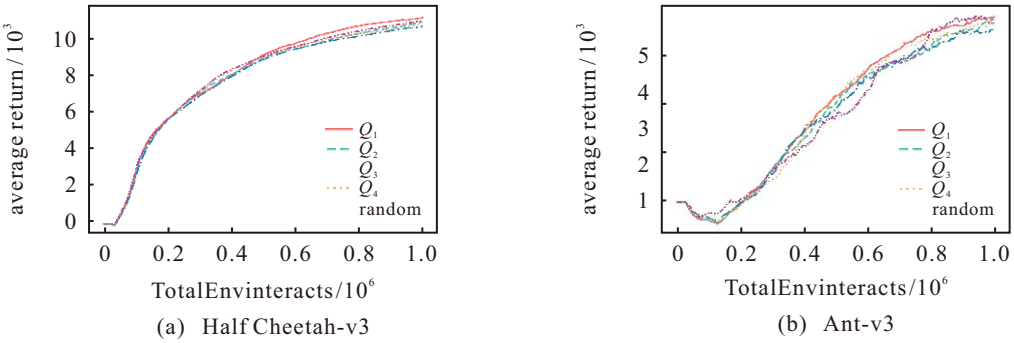


图8 选择不同函数估计器的对比实验

5 结 论

在深度强化学习算法中, 由累计近似误差引起的高估偏差已经被 TD3 算法解决. 但是在确定性策略梯度的 Actor-Critic 框架中, 克服累计近似误差的 TD3 算法导致了低估偏差. 低估偏差也会影响强化学习算法的性能, 是一个亟待解决的问题. 为此, 本文提出 D3PG-CNO 算法来解决深度强化学习领域的低估问题. 此算法在 Actor-Critic 框架中灵活设置多个 Actor 和 Critic 网络来缓解值函数低估偏差. D3PG-CNO 在经验收集阶段用一个 Critic 网络对 M 个 Actor 网络的输出动作进行评估, 选择最优的动作存入经验池. 同时, 在经验训练阶段, 先从 N 个 Critic 网络中选出在当前状态-动作对 (s,a) 下最小的 Critic 网络, 再用该网络对 M 个 Actor 网络的输出

动作进行评估, 选择最大值进行目标值的计算. 实验表明, D3PG-CNO 可以显著地降低低估误差, 提高算法稳定性和收敛性. D3PG-CNO 相比于现有的深度确定性策略梯度算法能获得更高的回报, 且此算法可以容易地适用于各种确定性策略梯度的 Actor-Critic 框架, 从而为解决低估偏差问题提供一种通用且有效的解决方案.

参考文献 (References)

[1] Dong S, Wang P, Abbas K. A survey on deep learning and its applications[J]. Computer Science Review, 2021, 40: 100379.

[2] Sutton R S, Barto A G. Reinforcement learning: An introduction[J]. IEEE Transactions on Neural Networks, 1998, 9(5): 1054.

[3] Thrun S, Schwartz A. Issues in using function approximation for reinforcement learning[C]. Proceedings of the 1993 Connectionist Models Summer School. Psychology Press, 2014: 255-263.

[4] Wang H N, Liu N, Zhang Y Y, et al. Deep reinforcement learning: A survey[J]. Frontiers of Information Technology & Electronic Engineering, 2020, 21(12): 1726-1744.

[5] 刘云飞, 张俊然. 深度神经网络学习率策略研究进展[J]. 控制与决策, 2023, 38(9): 2444-2460.
(Liu Y F, Zhang J R. Research advances in deep neural networks learning rate strategies[J]. Control and Decision, 2023, 38(9): 2444-2460.)

[6] 代学武, 吴越, 石琦, 等. 基于优先经验回放可迁移深度强化学习的高铁调度[J]. 控制与决策, 2023, 38(8): 2375-2388.
(Dai X W, Wu Y, Shi Q, et al. A transferable deep reinforcement learning high-speed railway rescheduling method based on prioritized experience replay[J]. Control and Decision, 2023, 38(8): 2375-2388.)

[7] 闫超, 相晓嘉, 徐昕, 等. 多智能体深度强化学习及其可扩展性与可迁移性研究综述[J]. 控制与决策, 2022, 37(12): 3083-3102.
(Yan C, Xiang X J, Xu X, et al. A survey on scalability and transferability of multi-agent deep reinforcement learning[J]. Control and Decision, 2022, 37(12): 3083-3102.)

[8] 刘磊, 杨晔, 刘赛, 等. 基于生存理论训练机器学习的智能驾驶路径生成方法[J]. 控制与决策, 2020, 35(10): 2433-2441.
(Liu L, Yang Y, Liu S, et al. Path generation method for intelligent driving based on machine learning trained by viability theory[J]. Control and Decision, 2020, 35(10): 2433-2441.)

[9] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J]. Nature, 2016, 529: 484-489.

[10] Liu R R, Nageotte F, Zanne P, et al. Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review[J]. Robotics, 2021, 10(1): 22.

[11] Yang X T, Ji Z, Wu J, et al. Hierarchical reinforcement learning with universal policies for multistep robotic manipulation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(9): 4727-4741.

[12] Pentaliotis A. Investigating overestimation bias in reinforcement learning[D]. Groningen: University of Groningen, 2020.

[13] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[14] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J/OL]. 2015, arXiv: 1509.02971.

[15] van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double *Q*-learning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2016, 30(1): 2094-2100.

[16] Hasselt H V. Double *Q*-learning[J]. Proceedings of Advances in Neural Information Processing Systems, 2010, 23: 2613-2621.

[17] Duan J L, Guan Y, Li S E, et al. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(11): 6584-6598.

[18] Fujimoto S, van Hoof H, Meger D. Addressing function approximation error in actor-critic methods[J/OL]. 2018, arXiv: 1802.09477.

[19] Lan Q F, Pan Y C, Fyshe A, et al. Maxmin *Q*-learning: Controlling the estimation bias of *Q*-learning[J/OL]. 2020, arXiv: 2002.06487.

[20] He Q, Hou X W. WD3: Taming the estimation bias in deep reinforcement learning[C]. 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence. Baltimore, 2020: 391-398.

[21] Wu D M, Dong X P, Shen J B, et al. Reducing estimation bias via triplet-average deep deterministic policy gradient[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(11): 4933-4945.

[22] Wei W, Zhang Y J, Liang J Y, et al. Controlling underestimation bias in reinforcement learning via quasi-median operation[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(8): 8621-8628.

作者简介

程玉虎 (1973−), 男, 教授, 博士生导师, 主要研究方向为机器学习、强化学习, E-mail: chengyuhu@163.com;

安冰清 (1998−), 女, 硕士生, 主要研究方向为强化学习, E-mail: 1399396213@qq.com;

孔毅 (1991−), 男, 副教授, 博士, 主要研究方向为机器学习, E-mail: kongyicumt@163.com.