采用分类经验回放的深度确定性策略梯度方法

时圣苗 [1] 刘 全 [1, 2, 3, 4]

摘 要 深度确定性策略梯度 (Deep deterministic policy gradient, DDPG) 方法在连续控制任务中取得了良好的性能表现. 为进一步提高深度确定性策略梯度方法中经验回放机制的效率, 提出分类经验回放方法, 并采用两种方式对经验样本分类: 基于时序差分误差样本分类的深度确定性策略梯度方法 (DDPG with temporal difference-error classification, TDC-DDPG) 和基于立即奖赏样本分类的深度确定性策略梯度方法 (DDPG with reward classification, RC-DDPG).在 TDC-DDPG 和 RC-DDPG 方法中, 分别使用两个经验缓冲池, 对产生的经验样本按照重要性程度分类存储, 网络模型训练时通过选取较多重要性程度高的样本加快模型学习. 在连续控制任务中对分类经验回放方法进行测试, 实验结果表明, 与随机选取经验样本的深度确定性策略梯度方法相比, TDC-DDPG 和 RC-DDPG 方法具有更好的性能.

关键词 连续控制任务, 深度确定性策略梯度, 经验回放, 分类经验回放

引用格式 时圣苗, 刘全. 采用分类经验回放的深度确定性策略梯度方法. 自动化学报, 2022, **48**(7): 1816−1823

DOI 10.16383/j.aas.c190406

Deep Deterministic Policy Gradient With Classified Experience Replay

SHI Sheng-Miao[1] LIU Quan[1, 2, 3, 4]

Abstract The deep deterministic policy gradient (DDPG) algorithm achieves good performance in continuous control tasks. In order to further improve the efficiency of the experience replay mechanism in the DDPG algorithm, a method of classifying the experience replay is proposed, where transitions are classified in two branches: deep deterministic policy gradient with temporal difference-error classification (TDC-DDPG) and deep deterministic policy gradient with reward classification (RC-DDPG). In both methods, two replay buffers are introduced respectively to classify the transitions according to the degree of importance. Learning can be speeded up in network model training period by selecting a greater number of transitions with higher importance. The classification experience replay method has been tested in a series of continuous control tasks and experimental results show that the TDC-DDPG and RC-DDPG methods have better performance than the DDPG method with random selection of transitions.

Key words Continuous control task, deep deterministic policy gradient (DDPG), experience replay, classifying experience replay

Citation Shi Sheng-Miao, Liu Quan. Deep deterministic policy gradient with classified experience replay. *Acta Automatica Sinica*, 2022, **48**(7): 1816−1823

收稿日期 2019-05-24 录用日期 2019-09-24
Manuscript received May 24, 2019; accepted September 24, 2019
国家自然科学基金 (61772355, 61702055, 61876217, 62176175), 江苏高校优势学科建设工程项目资助
Supported by National Natural Science Foundation of China (61772355, 61702055, 61876217, 62176175) and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD)
本文责任编委 魏庆来
Recommended by Associate Editor WEI Qing-Lai

1. 苏州大学计算机科学与技术学院 苏州 215006 2. 苏州大学江苏省计算机信息处理技术重点实验室 苏州 215006 3. 吉林大学符号计算与知识工程教育部重点实验室 长春 130012 4. 软件新技术与产业化协同创新中心 南京 210000
1. School of Computer Science and Technology, Soochow University, Suzhou 215006 2. Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006 3. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012 4. Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210000

强化学习 (Reinforcement learning, RL) 中, Agent 采用 “试错”的方式与环境进行交互, 通过从环境中获得最大化累积奖赏寻求最优策略[1]. RL 算法根据 Agent 当前所处状态求解可执行动作, 因此 RL 适用于序贯决策问题的求解[2]. 利用具有感知能力的深度学习作为 RL 状态特征提取的工具, 二者结合形成的深度强化学习 (Deep reinforcement learning, DRL) 算法是目前人工智能领域研究的热点之一[3−4].

在线 DRL 算法采用增量式方法更新网络参数, 通过 Agent 与环境交互产生经验样本 $e = (s_t, a_t, r_t, s_{t+1})$, 直接将此样本用于训练网络参数, 在一次训练后立即丢弃传入的数据[5]. 然而此方法存在两个问题: 1) 训练神经网络的数据要求满足独立同分

布, 而强化学习中产生的数据样本之间具有时序相关性. 2) 数据样本使用后立即丢弃, 使得数据无法重复利用. 针对以上问题, Mnih 等[6] 采用经验回放的方法, 使用经验缓冲池存储经验样本, 通过随机选取经验样本进行神经网络训练. 然而经验回放方法中未考虑不同经验样本具有不同的重要性, 随机选取无法充分利用对网络参数更新作用更大的经验样本. Schaul 等[7] 根据经验样本的重要性程度赋予每个样本不同的优先级, 通过频繁选取优先级高的经验样本提高神经网络训练速度. 优先级经验回放一方面增加了对经验样本赋予和更改优先级的操作, 另一方面需要扫描经验缓冲池以获取优先级高的经验样本, 因此增加了算法的时间复杂度. 与优先级经验回放不同, 本文提出的分类经验回放方法对不同重要性程度的经验样本分类存储. 将此方法应用于深度确定性策略梯度 (Deep deterministic policy gradient, DDPG) 算法中, 提出了采用分类经验回放的深度确定性策略梯度 (Deep deterministic policy gradient with classified experience replay, CER-DDPG) 方法. CER-DDPG 采用两种分类方式: 1) 根据经验样本中的时序差分误差 (Temporal difference-error, TD-error); 2) 基于立即奖赏值进行分类. 其中, TD-error 代表 Agent 从当前状态能够获得的学习进度, RL 经典算法 Sarsa、Q-leaning 均采用一步自举的方式计算 TD-error 实现算法收敛. CER-DDPG 中, 将大于 TD-error 平均值或立即奖赏平均值的经验样本存入经验缓冲池 1 中, 其余存入经验缓冲池 2 中. 网络训练时每批次从经验缓冲池 1 中选取更多数量的样本, 同时为保证样本的多样性, 从经验缓冲池 2 中选取少量的经验样本, 以此替代优先级经验回放中频繁选取优先级高的经验样本. 分类经验回放方法具有和普通经验回放方法相同的时间复杂度, 且未增加空间复杂度.

本文主要贡献如下:

1) 采用双经验缓冲池存储经验样本, 并根据经验样本中的 TD-error 和立即奖赏值完成对样本的分类;

2) 从每个经验缓冲池中选取不同数量的经验样本进行网络参数更新;

3) 在具有连续动作空间的 RL 任务中进行实验, 结果表明, 相比随机采样的 DDPG 算法, 本文提出的基于时序差分误差样本分类的深度确定性策略梯度方法 (DDPG with temporal difference-error classification, TDC-DDPG) 和基于立即奖赏样本分类的深度确定性策略梯度方法 (DDPG with reward classification, RC-DDPG) 能够取得更好的实验效果. 并与置信区域策略优化 (Trust region policy optimization, TRPO) 算法以及近端策略优化 (Proximal policy optization, PPO) 算法进行比较, 进一步证明了本文所提出算法的有效性.

1　背景

1.1　强化学习

马尔科夫决策过程 (Markov decision process, MDP) 是序贯决策的经典形式, 其中动作不仅影响到立即奖赏, 同样影响后续的状态或动作, 以及采取后续动作所获得的未来奖赏. 因此, 通常使用 MDP 对 RL 问题进行建模, 将 RL 问题定义为一个五元组 (S, A, P, R, γ). S 表示状态空间, A 表示动作空间, $P: S\times A\times S\rightarrow [0, 1]$ 表示状态迁移概率, $R: S\times A\rightarrow \mathbf{R}$ 为奖赏函数, γ 为折扣因子[8]. 通过 MDP 可以构建 Agent 与环境的交互过程, 每一离散时间步 t, Agent 接收到来自环境的状态表示 s_t, 在此基础上执行动作 a_t. 该时间步之后, Agent 收到来自环境反馈的标量化奖赏 r_t 并处于下一状态 s_{t+1}.

Agent 执行的动作由策略 π 定义, 策略 π 为状态映射到每个动作的概率: $S\rightarrow P(A)$. RL 的目标为求解最优策略 π^*, 在遵循策略 π^* 的情况下能够获得最大的累积奖赏 $G_t=\sum_{t'=t}^{T}\gamma^{t'-t}r(s_{t'}, a_{t'})$, 其中, T 表示该情节终止时间步.

状态动作值函数 $Q^{\pi}(s, a)$ 表示 Agent 在当前状态 s_t 下执行动作 a_t, 遵循策略 π 所获得的期望累积奖赏

$$Q^{\pi}(s, a)=\mathrm{E}_{\pi}[G_t|s_t=s, a_t=a] \tag{1}$$

$Q^{\pi}(s, a)$ 满足具有递归属性的贝尔曼方程

$$Q^{\pi}(s, a)=\mathrm{E}_{\pi}[r_{t+1}+\gamma Q^{\pi}(s_{t+1}, a_{t+1})|s_t=s, a_t=a] \tag{2}$$

迭代计算贝尔曼方程可实现值函数的收敛. 当前时刻状态动作估计值函数与更好地估计 $r_{t+1}+\gamma Q^{\pi}(s_{t+1}, a_{t+1})$ 之间的误差称为 TD-error

$$\delta_t=r_{t+1}+\gamma Q^{\pi}(s_{t+1}, a_{t+1})-Q^{\pi}(s_t, a_t) \tag{3}$$

通过求解状态动作值函数仅局限于解决具有离散动作空间的 RL 问题, 面对具有连续动作空间的 RL 问题, 策略梯度方法提供了解决问题的方式[9].

1.2　深度确定性策略梯度方法

RL 算法分为基于值函数和基于策略两种方法. 基于策略的方法可以解决大状态动作空间或连续动作空间 RL 问题[10]. 确定性策略梯度 (Deterministic policy gradient, DPG) 方法以行动者−评论家

(Actor-critic, AC) 算法为基础, 通过行动者将状态映射到特定动作, 评论家利用贝尔曼方程实现值函数的收敛[11−12].

DDPG 中, 使用深度神经网络作为非线性函数逼近器构造行动者 $\mu(s|\boldsymbol{\theta}^{\mu})$ 和评论家 $Q(s,\, a|\boldsymbol{\theta}^{Q})$ 的网络模型. 受到深度 Q 网络 (Deep Q-network, DQN) 的启发, 设置行动者目标网络 $\mu'(s|\boldsymbol{\theta}^{\mu'})$ 和评论家目标网络 $Q'(s,\, a|\boldsymbol{\theta}^{Q'})$. 由于 DPG 中行动者将状态映射到确定动作, 因此解决连续动作空间 RL 任务存在缺乏探索性问题[13]. DDPG 算法通过添加独立于行动者网络的探索噪声 $Noise$ 构造具有探索性的行动者网络 μ' [14]

$$\mu'(s_t) = \mu(s_t|\boldsymbol{\theta}^{\mu}) + Noise \tag{4}$$

网络模型学习时, 评论家网络通过最小化损失函数 $L(\boldsymbol{\theta}^{Q})$ 更新网络参数

$$L(\boldsymbol{\theta}^{Q}) = \mathrm{E}_{s_t,\, a_t,\, r_t,\, s_{t+1}\sim D}[(y_t - Q(s_t,\, a_t|\boldsymbol{\theta}^{Q}))^2] \tag{5}$$

其中,

$$y_t = r(s_t,\, a_t) + \gamma Q'(s_{t+1},\, \mu'(s_{t+1}|\boldsymbol{\theta}^{\mu'})|\boldsymbol{\theta}^{Q'}) \tag{6}$$

行动者网络采用蒙特卡罗方法进行采样以逼近期望值, 可通过链式法则近似更新行动者网络参数, 如式 (7) 所示

$$\nabla_{\boldsymbol{\theta}^{\mu}} J(\boldsymbol{\theta}^{\mu}) = \frac{1}{N}\sum_{i} \nabla_a Q(s,\, a|\boldsymbol{\theta}^{Q})|_{s=s_i,\, a=\mu(s_i)} \nabla_{\boldsymbol{\theta}^{\mu}} \mu(s|\boldsymbol{\theta}^{\mu})|_{s_i} \tag{7}$$

目标网络采用 “soft”更新方式, 通过缓慢跟踪学习的网络更新参数

$$\boldsymbol{\theta}' \leftarrow \tau\boldsymbol{\theta} + (1-\tau)\boldsymbol{\theta}' \tag{8}$$

“soft”更新方式使得不稳定问题更接近于监督学习, 虽减慢了目标网络参数更新速度, 但在学习过程中能够获得更好的稳定性.

DDPG 同样用到了经验回放机制, 将行动者网络与环境交互产生的经验样本 $e = (s_t,\, a_t,\, r_t,\, s_{t+1})$ 存入经验缓冲池中, 网络训练时通过从经验缓冲池中随机选取每批次经验样本用于网络参数的更新. 随机选取方式未考虑不同经验样本的重要性, 如何更有效利 用缓冲池中的样本数据成为经验回放机制面临的主要挑战.

2 采用分类经验回放的 DDPG 算法

本节将介绍 CER-DDPG 算法的思想和结构, 对采用的分类方法分析说明, 最后描述算法流程并分析.

2.1 分类经验回放

经验回放机制在消除数据样本之间关联性的同时能够提高样本利用率. 在 Agent 与环境交互产生的经验样本中, 不同经验样本对网络训练所起作用不同, 某些经验样本比其他经验样本更能有效地促进网络模型学习. 等概率选取每一个经验样本会在简单样本上花费较多的时间, 增加了算法训练时间步数. 因此, 本文所提出的分类经验回放方法最主要的一点是对不同重要性程度经验样本分类存放, 在网络模型学习时分别从不同类别经验样本中选取每批次样本数据. 对于重要性程度高的经验样本, 每批次以较多数量选取, 同时为保证样本数据多样性, 每批次选取少量重要性程度低的经验样本.

TDC-DDPG 中, 使用两个经验缓冲池存放经验样本. 初始化网络模型时, 将两个经验缓冲池中所有样本 TD-error 的平均值设置为 0. 每产生一条新的经验样本, 首先更新所有经验样本 TD-error 的平均值, 再将该条样本数据的 TD-error 与平均值进行比较, 若该经验样本中的 TD-error 大于所有样本 TD-error 的平均值, 则将该样本存入经验缓冲池 1 中, 否则存入经验缓冲池 2 中.

RC-DDPG 方法根据经验样本中的立即奖赏值进行分类, 具体分类方法与 TDC-DDPG 方法相同. CER-DDPG 算法结构如图 1 所示.

图 1 中, 在每一时间步 t, 行动者网络执行动作 a_t, 产生经验样本 $e = (s_t,\, a_t,\, r_t,\, s_{t+1})$ 后, 首先对该样本数据进行分类, 然后再进行存储操作. 优先级经验回放中使用一个经验缓冲池存储所有经验样本, 根据样本不同重要性程度赋予每个样本不同优先级, 网络训练时扫描经验缓冲池获取经验样本, 通过更频繁地选取优先级高的样本加快网络模型训练速度. CER-DDPG 方法在经验样本存储前, 将其按照重要性程度分类, 减少了赋予以及更改优先级的操作, 并且在选取每批次数据样本时从不同经验缓冲池中随机选取, 不需要扫描经验缓冲池, 能够获取高重要性程度经验样本的同时减少了算法时间复杂度.

分类经验回放中最关键的是经验样本分类的衡量标准. 本文提出的 CER-DDPG 方法分别采用经验样本中的 TD-error 和立即奖赏值对样本进行分类.

1) TD-error 经验样本分类. DDPG 算法中, 评论家采用时序差分误差的形式对行动者网络做出的动作选择进行评价, 网络参数更新时使用一步自举的方式计算 TD-error, TD-error 反映了 Agent 从当 前状态经验样本中的学习进度, 利用 TD-error

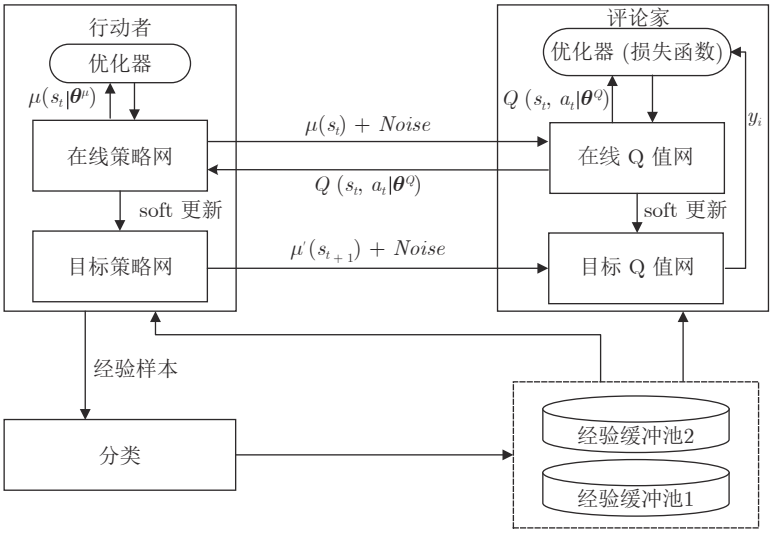


图 1　CER-DDPG 算法结构示意图

Fig. 1　CER-DDPG algorithm structure diagram

尤其适用于增量式 DRL 算法参数的更新. 因此, TDC-DDPG 中根据经验样本 TD-error 进行分类, 认为 TD-error 大的经验样本对神经网络参数更新幅度更大, 重要性程度更高, 并将 TD-error 值大于平均值的经验样本存入经验缓冲池 1 中.

2) 立即奖赏经验样本分类. 神经科学研究表明啮齿动物在清醒或睡眠期间海马体中会重播先前经历的序列, 与奖赏相关的序列会更频繁地被重播[15−16]. 受到该观点启发, RC-DDPG 方法中根据经验样本中的立即奖赏值对样本进行分类, 认为立即奖赏值大的经验样本重要性程度更高, 将立即奖赏值大于平均值的经验样本存入经验缓冲池 1 中, 其余存入经验缓冲池 2 中.

2.2　算法

为更有效地利用经验样本以及提高经验回放机制的效率, 将对经验样本的分类方法应用到 DDPG 算法中, 提出的 CER-DDPG 算法描述如算法 1 所示:

算法 1. 采用分类经验回放的深度确定性策略梯度方法

1) 初始化行动者网络 $\mu(s|\boldsymbol{\theta}^{\mu})$ 和评论家网络 $Q(s, a|\boldsymbol{\theta}^{Q})$, 目标网络参数 $\boldsymbol{\theta}^{\mu'} \leftarrow \boldsymbol{\theta}^{\mu}$ 和 $\boldsymbol{\theta}^{Q'} \leftarrow \boldsymbol{\theta}^{Q}$, 经验缓冲池 D_1, D_2, 批次抽样数量 N_1, N_2, 折扣因子 γ, 最大情节数 E, 每情节最大时间步 $T_{\max}$.

2) **for** $episode = 1$, E **do**

3) 初始化探索噪声 $Noise$

4) 获取初始状态 s_t

5) **for** $t = 1$, $T_{\max}$ **do**

6) 选择动作 $a_t = \mu(s_t|\boldsymbol{\theta}^{\mu}) + Noise$

7) 执行动作 a_t, 获得立即奖赏 r_t 和下一状态 s_{t+1}

8) 根据经验样本 $e_i = (s_t, a_t, r_t, s_{t+1})$ 的 TD-error 或 r_t 分类并存入经验缓冲池 D_1 或 D_2 中

9) 从 D_1 中选取 N_1 个经验样本, D_2 中选取 N_2 个经验样本

10) 计算 $y_i = r_i + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\boldsymbol{\theta}^{\mu'})|\boldsymbol{\theta}^{Q'})$

11) 通过最小化损失函数 $L(\boldsymbol{\theta}^{Q})$ 更新评论家网络参数:

$$L(\boldsymbol{\theta}^{Q}) = \mathrm{E}_{s_t, a_t, r_t, s_{t+1}}[(y_i - Q(s_i, a_i|\boldsymbol{\theta}^{Q}))^2]$$

12) 通过策略梯度方法更新行动者网络:

$$\nabla_{\boldsymbol{\theta}^{\mu}} J(\boldsymbol{\theta}^{\mu}) \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\boldsymbol{\theta}^{Q})|_{s=s_i, a=\mu(s_i)} \nabla_{\boldsymbol{\theta}^{\mu}} \mu(s|\boldsymbol{\theta}^{\mu})|_{s_i}$$

13) 更新目标网络:

$$\boldsymbol{\theta}^{Q'} \leftarrow \tau\boldsymbol{\theta}^{Q} + (1-\tau)\boldsymbol{\theta}^{Q'}$$
$$\boldsymbol{\theta}^{\mu'} \leftarrow \tau\boldsymbol{\theta}^{\mu} + (1-\tau)\boldsymbol{\theta}^{\mu'}$$

14) **end for**

15) **end for**

算法 1 中, 第 3 步为对行动者网络添加探索噪声过程, 第 5 ∼ 7 步为产生经验样本的过程, 第 8 ∼ 9 步为经验样本的分类和获取过程, 第 10 ∼ 13 步为网络模型学习过程.

由于不同任务中 Agent 每一时刻获得的立即奖赏值不同, 因此产生的经验样本 TD-error 和立即奖赏值存在差异, 难以采用固定数值作为分类的衡量标准. CER-DDPG 方法中, 使用 TD-error 和立即奖赏平均值进行样本分类, 并且在产生经验样本过程中不断更新 TD-error 和立即奖赏平均值, 能够动态性地将不同经验样本准确分类. 分类经验

回放方法相比普通经验回放方法仅增加了 O(1) 的时间复杂度, 可忽略不计. 优先级经验回放中根据优先级大小频繁选取优先级高的经验样本, CER-DDPG 方法通过每批次从经验缓冲池 1 中选取较多样本数量同样能够选取到重要性程度高的样本, 与优先级经验回放相比, CER-DDPG 方法效率更高.

3 实验

为验证 CER-DDPG 方法的有效性, 在 OpenAI Gym 工具包中 MuJoCo 环境下进行实验测试. MuJoCo 环境包含了一系列具有连续动作空间的 RL 任务, 本文分别在 HalfCheetah、Ant、Humanoid、Walker、Hopper 和 Swimmer 任务中进行测试. 实验以深度确定性策略梯度 (DDPG) 算法作为 baseline, 分别以 TD-error 分类的深度确定性策略梯度 (TDC-DDPG) 方法和立即奖赏分类的深度确定性策略梯度 (RC-DDPG) 方法进行对比实验.

3.1 实验参数设置

为保证实验对比公平性, 本文实验参数设置与参考文献中 DDPG 算法一致, TRPO 与 PPO 算法来自 OpenAI baselines 算法集. 对行动者网络添加的噪声均使用参数相同的 Ornstein-Uhlenbeck 噪声分布, 每批次样本数量均相等. DDPG 中, 经验缓冲池大小设置为 1 000 000, 批次选取样本数量取 $N = 64$. TDC-DDPG 和 RC-DDPG 中, D_1 和 D_2 均为 500 000, 批次样本数量取 $N_1 = 48$, $N_2 = 16$. 每情节最大时间步数设置为 $T_{\max} = 1\,000$, 时间步数超过 1 000 时情节重新开始. 行动者网络学习率 $\alpha^{\mu} = 1 \times 10^{-4}$, 评论家网络学习率 $\alpha^{Q} = 1 \times 10^{-3}$. 折扣因子 $\gamma = 0.99$, 目标网络更新时 $\tau = 0.001$.

3.2 实验结果及分析

图 2 展示了在不同任务中 3 种算法的实验效果, 每个任务训练 500 个阶段, 每阶段包含 2 000 个时间步, 通过对比每个训练阶段获得的平均累积奖赏衡量算法优劣.

如图 2 所示, 在大多数任务中 TDC-DDPG 和 RC-DDPG 算法性能优于随机选取经验样本的 DDPG 算法, 说明采用分类经验回放的方法能够选取到对网络模型学习更有效的经验样本, 在相同训练阶段内学习到累积奖赏更高的策略.

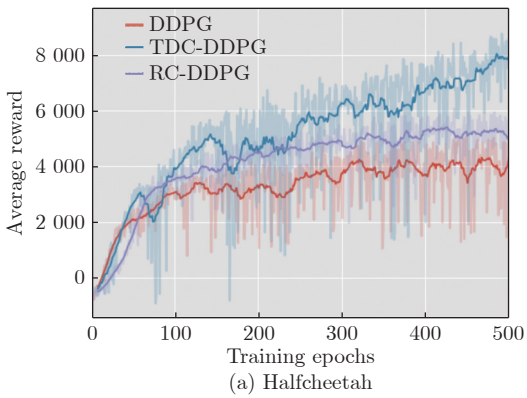
在 HalfCheetah 任务中, 通过控制双足猎豹 Agent 快速奔跑获取奖赏. 在网络模型训练的初始阶段中 3 种算法均能够取得较快学习速度. 而第 20 个训练阶段后, DDPG 算法表现趋于平稳, TDC-DDPG 和 RC-DDPG 算法仍然能够以较快的学习速度提升算法性能, 最终训练阶段具有明显优势.

在 Humanoid 和 Swimmer 任务中, 训练初始阶段 TDC-DDPG 和 RC-DDPG 算法优势并不显著, 随着训练时间步的增加, 在训练阶段后期算法优势逐渐明显. 因为在这两个任务中, 每一时间步 Agent 获得的立即奖赏值在很小的范围内波动, 导致 RC-DDPG 方法中两个经验缓冲池中样本类型很相近, TDC-DDPG 方法根据经验样本 TD-error 分类, 立即奖赏值同样会影响到 TD-error 的大小, 因此初始训练阶段算法性能优势表现不明显. 然而在 Walker 任务中, 每一时间步获得的立即奖赏值大小不均导致 3 种算法训练得到的实验结果波动性均较大, 但本文提出方法实验效果更优.

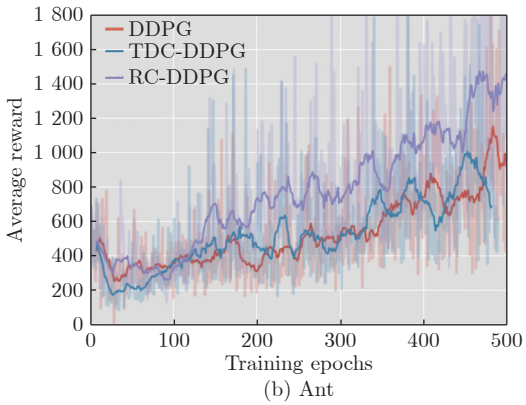
Hopper 任务通过控制双足机器人 Agent 向前跳跃获取奖赏. 由于状态动作空间维度低, Agent 会执行一些相似动作导致经验样本相似, 因此分类经验回放方法性能提升不明显.

表 1 展示了 500 个训练阶段内 3 种算法在不同任务中所取得的平均奖赏值、最高奖赏值以及标准差.

从表 1 可以看出, 与 DDPG 方法相比, TDC-



(a) Halfcheetah



(b) Ant

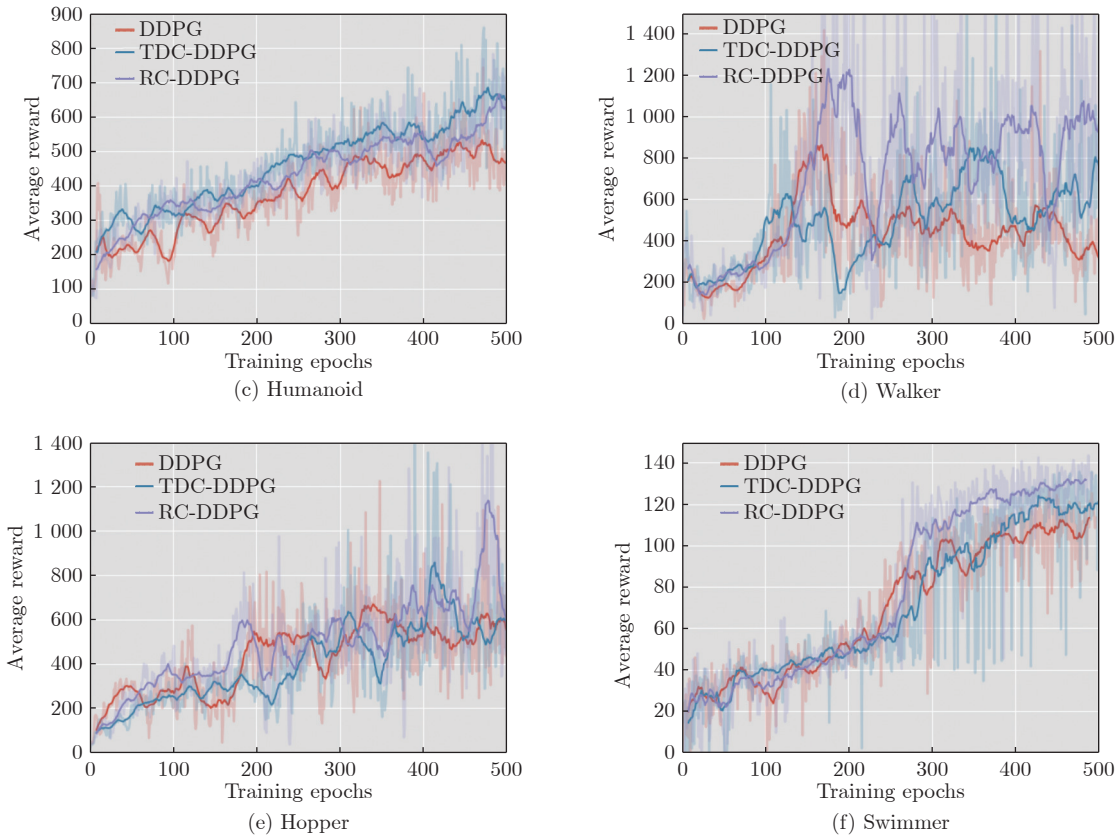


图 2　实验效果对比图

Fig. 2　Comparison of experimental results

表 1　连续动作任务中实验数据

Table 1　Experimental data in continuous action tasks

任务名称	算法	平均奖赏	最高奖赏	标准差
HalfCheetah	DDPG	3 360.32	5 335.23	1 246.40
	TDC-DDPG	5 349.64	9 220.27	2 368.13
	RC-DDPG	3 979.64	6 553.49	1 580.21
Ant	DDPG	551.87	1 908.30	307.86
	TDC-DDPG	521.42	1 863.99	296.91
	RC-DDPG	772.37	2 971.63	460.05
Humanoid	DDPG	404.36	822.11	114.38
	TDC-DDPG	462.65	858.34	108.20
	RC-DDPG	440.30	835.75	100.31
Walker	DDPG	506.10	1 416.00	243.02
	TDC-DDPG	521.58	1 919.15	252.95
	RC-DDPG	700.57	3 292.62	484.65
Hopper	DDPG	422.10	1 224.68	180.04
	TDC-DDPG	432.64	1 689.48	223.61
	RC-DDPG	513.45	2 050.72	257.82
Swimmer	DDPG	34.06	63.16	16.74
	TDC-DDPG	44.18	69.40	19.77
	RC-DDPG	38.44	71.70	21.59

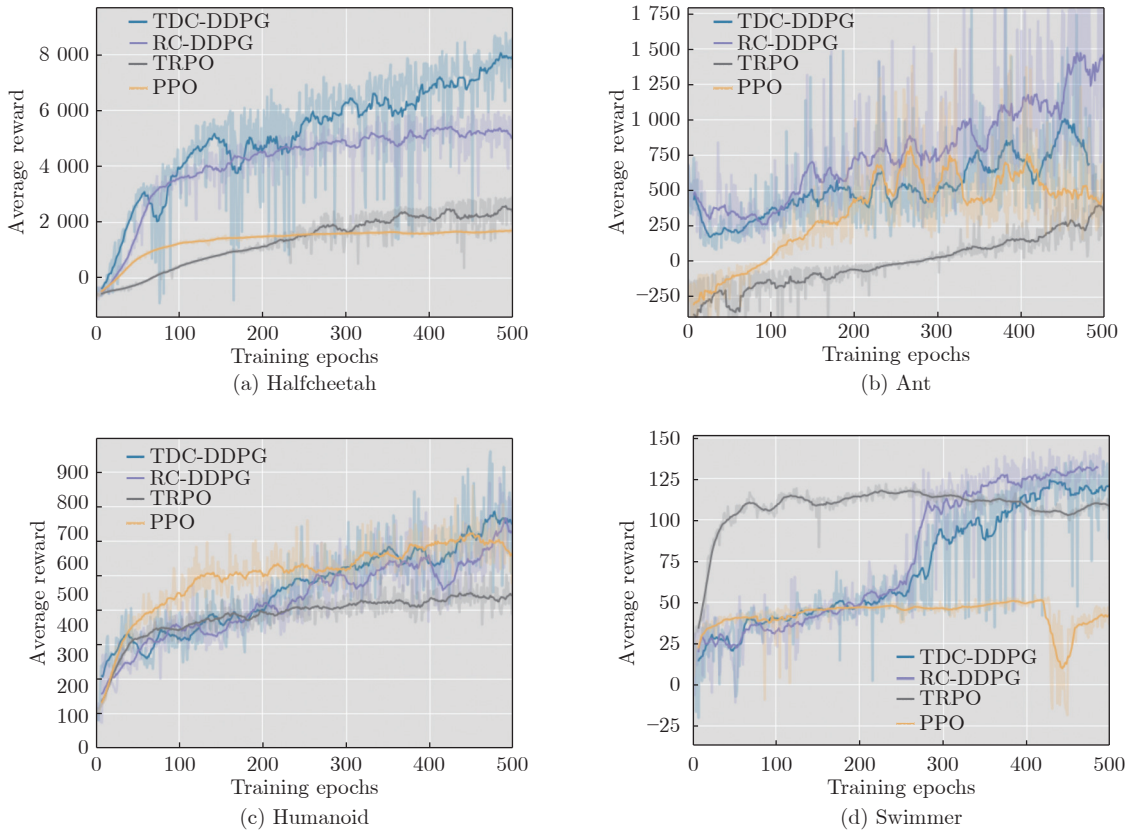


图 3 CER-DDPG 与最新策略梯度算法的实验对比

Fig. 3 Experimental comparison of CER-DDPG with the latest policy gradient algorithm

DDPG 和 RC-DDPG 方法取得的平均奖赏和最高奖赏值更高, 不同训练阶段累积奖赏值差异更大, 导致标准差更大.

为进一步证明算法的有效性, 在 HalfCheetah、Ant、Humanoid 和 Swimmer 任务中增加了与 TRPO 算法以及 PPO 算法的对比实验. 从图 3 可看出, TDC-DDPG 和 RC-DDPG 方法在与最新策略梯度算法比较中同样取得了更好的实验效果.

4 结束语

DDPG 算法在解决连续动作空间 RL 问题上取得了巨大成功. 网络模型学习过程中, 使用经验回放机制打破了经验样本之间存在的时序相关性. 然而经验回放未考虑不同经验样本的重要性, 不能有效利用样本数据, 对样本设置优先级又增加了算法时间复杂度. 因此, 本文提出分类经验回放方法并利用经验样本的 TD-error 和立即奖赏值进行分类存储用于解决经验回放中存在的问题. 在具有连续状态动作空间 RL 任务中的实验结果表明, 本文提出的 TDC-DDPG 和 RC-DDPG 方法在连续控制任务中表现更优.

References

1 Zhang Yao-Zhong, Hu Xiao-Fang, Zhou Yue, Duan Shu-Kai. A novel reinforcement learning algorithm based on multilayer memristive spiking neural network with applications. *Acta Automatic Sinica*, 2019, **45**(8): 1536−1547
(张耀中, 胡小方, 周跃, 段书凯. 基于多层忆阻脉冲神经网络的强化学习及应用. 自动化学报, 2019, **45**(8): 1536−1547)

2 Dorpinghaus M, Roldan E, Neri I, Meyr H, Julicher F. An information theoretic analysis of sequential decision-making. *Mathematics*, 2017, **39**(6): 429−437

3 Li Y X. Deep reinforcement learning: An overview. *Machine Learning*, 2017, **12**(2): 231−316

4 Qin Rui, Zeng Shuai, Li Juan-Juan, Yuan Yong. Parallel enterprises resource planning based on deep reinforcement learning. *Acta Automatic Sinica*, 2017, **43**(9): 1588−1596
(秦蕊, 曾帅, 李娟娟, 袁勇. 基于深度强化学习的平行企业资源计划. 自动化学报, 2017, **43**(9): 1588−1596)

5 Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou A, Wierstra D, et al. Playing atari with deep reinforcement learning. In: Proceedings of the Workshops at the 26th Neural Information Processing Systems 2013. Lake Tahoe, USA: MIT Press, 2013. 201−220

6 Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, et al. Human-level control through deep reinforcement learning. *Nature*, 2015, **518**(7540): 529−533

7 Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. In: Proceedings of the 4th International Conference

on Learning Representations. San Juan, PuertoRico, USA: ICLR, 2016. 322−355

8 Gao Yang, Chen Shi-Fu, Lu Xin. Research on reinforcement learning: A review. *Acta Automatic Sinica*, 2004, **30**(1): 86−100 (高阳, 陈世福, 陆鑫. 强化学习研究综述. 自动化学报, 2004, **30**(1): 86−100)

9 Ertel W. *Reinforcement Learning*. London: Springer-Verlag, 2017. 12−16

10 Peters J, Bagnell J A, Sammut C. Policy gradient methods. *Encyclopedia of Machine Learning*, 2010, **5**(11): 774−776

11 Sutton R S, Barto A G. Reinforcement learning: An introduction. Cambridge, USA: MIT Press, 2018.

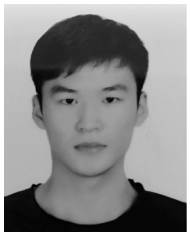
12 Thomas P S, Brunskill E. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines. *Artificial Intelligence*, 2018, **16**(4): 23−25

13 Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmillerm M. Deterministic policy gradient algorithms. In: Proceedings of the 31st International Conference on Machine Learning. New York, USA: ACM, 2014. 387−395

14 Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. *Computer Science*, 2015, **8**(6): A187

15 Atherton L A, Dupret D, Mellor J R. Memory trace replay: The shaping of memory consolidation by neuromodulation. *Trends in Neurosciences*, 2015, **38**(9): 560−570

16 Olafsdottir H, Barry C, Saleem A B, Hassabis D, Spiers H J. Hippocampal place cells construct reward related sequences through unexplored space. *Elife*, 2015, **4**: e06063



时圣苗 苏州大学计算机科学与技术学院硕士研究生. 主要研究方向为深度强化学习.
E-mail: 20175227045@stu.suda.edu.cn
(**SHI Sheng-Miao** Master student at the School of Computer Science and Technology, Soochow University. His main research interest is deep reinforcement learning.)

刘 全 苏州大学教授. 主要研究方向为深度强化学习, 自动推理. 本文通信作者.
E-mail: quanliu@suda.edu.cn
(**LIU Quan** Professor at Soochow University. His research interest covers deep reinforcement learning and automated reasoning. Corresponding author of this paper.)