计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200900208

基于情节经验回放的深度确定性策略梯度方法

张建行[1] 刘 全[1,2,3,4]
1 苏州大学计算机科学与技术学院 江苏 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 江苏 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
(20185227051@stu.suda.edu.cn)

摘 要 强化学习中的连续控制问题一直是近年来的研究热点。深度确定性策略梯度(Deep Deterministic Policy Gradients,DDPG)算法在连续控制任务中表现优异。DDPG算法利用经验回放机制训练网络模型,为了进一步提高经验回放机制在DDPG算法中的效率,将情节累积回报作为样本分类依据,提出一种基于情节经验回放的深度确定性策略梯度(Deep Deterministic Policy Gradient with Episode Experience Replay,EER-DDPG)方法。首先,将经验样本以情节为单位进行存储,根据情节累积回报大小使用两个经验缓冲池分类存储。然后,在网络模型训练阶段着重对累积回报较大的样本进行采样,以提升训练质量。在连续控制任务中对该方法进行实验验证,并与采取随机采样的DDPG方法、置信区域策略优化(Trust Region Policy Optimization,TRPO)方法以及近端策略优化(Proximal Policy Optimization,PPO)方法进行比较。实验结果表明,EER-DDPG方法有更好的性能表现。

关键词:深度确定性策略梯度;连续控制任务;经验回放;累积回报;分类经验回放
中图法分类号 TP181

Deep Deterministic Policy Gradient with Episode Experience Replay

ZHANG Jian-hang[1] and LIU Quan[1,2,3,4]
1 School of Computer and Technology,Soochow University,Suzhou,Jiangsu 215006,China
2 Provincial Key Laboratory for Computer Information Processing Technology,Soochow University,Suzhou,Jiangsu 215006,China
3 Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education,Jilin University,Changchun 130012,China
4 Collaborative Innovation Center of Novel Software Technology and Industrialization,Nanjing 210000,China

Abstract The research on continuous control in reinforcement learning has been a hot topic in recent years. The deep deterministic policy gradient (DDPG) algorithm performs well in continuous control tasks. DDPG algorithm uses experience replay mechanism to train the network model,and in order to further improve the efficiency of experience replay mechanism in the DDPG algorithm,the cumulative reward is used as the transition classification basis,a deep deterministic policy gradient with episodic experience replay (EER-DDPG) algorithm is proposed. First of all,the transitions are stored in the unit of episode,and two replay buffers are introduced respectively to classify the transitions according to the cumulative reward. Then,the quality of policy can be improved in network model training period by random sampling of the episodes with large cumulative reward. In the continuous control tasks,this algorithm is verified by experiments,and compared with DDPG algorithm,trust region policy optimization (TRPO) algorithm and proximal policy optimization (PPO) algorithm. The experimental results show that EER-DDPG algorithm has better performance.

到稿日期:2020-09-30 返修日期:2020-12-30 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61772355,61702055,61502323,61502329);江苏省高等学校自然科学研究重大项目(18KJA520011,17KJA520004);吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172014K04,93K172017K18);苏州市应用基础研究计划工业部分(SYG201422);江苏省高校优势学科建设工程资助项目
This work was supported by the National Natural Science Foundation of China(61772355,61702055,61502323,61502329),Jiangsu Province Natural Science Research University Major Projects(18KJA520011,17KJA520004),Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education,Jilin University(93K172014K04,93K172017K18),Suzhou Industrial Application of Basic Research Program Part(SYG201422) and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.
通信作者:刘全(quanliu@suda.edu.cn)

Keywords Deep deterministic policy gradient, Continuous control tasks, Experience replay, Cumulative reward, Classifying experience replay

1 引言

强化学习(reinforcement learning)是目前解决序贯决策问题的一种主流途径[1],智能体(agent)通过执行某种策略与环境进行交互,在每一时间步中获得奖赏,其目标是以最大化累积回报的方式来寻求最优策略[2]。强化学习主要分为两类算法,基于值函数计算的方法和基于策略梯度的方法。

基于值函数计算的方法通过不断迭代策略评估与策略改进过程获得最优策略。这类方法适用于状态动作空间离散的强化学习任务,其优点是训练稳定,算法波动性较小,但对于连续状态动作空间的任务来说并不适用,因为此时状态动作空间为连续值,所以计算所有状态动作对的值函数变得不可实现。对比基于值函数计算的方法,基于策略梯度的方法直接对策略进行参数化表示,并与累积回报联系起来,通过最大化累积回报期望得到最优策略[3]。此时策略梯度为一个概率密度函数,实际的输出是一个实值,大大降低了计算复杂度。策略的对数梯度被称为评价函数,在策略梯度方法的基础上通过高斯策略来计算评价函数,可以完成基于连续动作空间的任务。

传统强化学习受限于对高维状态动作空间感知力的不足,难以得到广泛应用。随着现代机器设备计算能力的不断提高,深度学习(deep learning)通过构造神经网络结构,可以提取高维特征[4]。深度强化学习(deep reinforcement learning)方法将深度学习的高维特征提取能力和强化学习的序贯决策能力相结合,是目前解决高维特征输入求解最优策略任务的主要方法[5],已在机器人控制[6]、游戏博弈[7]、推荐系统[8]等领域得到广泛应用。

深度强化学习方法可以分为基于值函数的深度强化学习方法和基于策略梯度的深度强化学习方法两大类。典型的算法分别为深度Q网络(Deep Q-Network, DQN)[9]方法和深度确定性策略梯度(Deep Deterministic Policy Gradients, DDPG)[10]方法。DQN通过将神经网络与Q-Learning算法相结合,利用卷积神经网络表征游戏画面数据作为强化学习中状态的输入。该方法在视觉感知任务中取得了重大突破,在部分游戏中超越了人类最佳水平。与前者不同,DDPG算法在行动者-评论家框架的基础上使用确定性策略,利用深度神经网络参数化策略,在连续动作空间任务中表现优异。

在线深度强化学习算法采用增量式方法训练神经网络参数,每一时间步,将智能体与环境交互产生的样本数据直接用于网络训练,使用过后立即丢弃该样本。这会导致以下两个问题:1)深度神经网络的训练数据要求满足独立同分布的性质[11],而强化学习中相邻时间步产生的经验样本之间存在时序相关性,增量式训练方法会导致网络训练波动较大;2)神经网络训练需要大量样本数据,每次经验样本使用过后直接丢弃会造成样本不足,样本利用率低,需要智能体与环境进行更多的交互,降低了算法训练的效率[12]。针对以上问题,Mnih等提出了经验回放机制,使用经验缓冲池来存放智能体与环境交互产生的样本,随后在经验缓冲池中随机选取批量样本训练神经网络。由于该方法随机选取经验样本,没有考虑不同经验样本对算法收敛速度和性能的影响,因此无法高效利用对网络更新有更大作用的样本。Schaul等[13]提出了一种优先级经验回放方法,对经验缓冲池中每一个样本赋予不同的优先级,该方法将经验样本中的时序差分误差(Temporal Difference error, TD-error)作为优先级赋予的标准,即误差绝对值越大表明该样本的重要性越高,对网络训练起到的促进作用就越大。优先级经验回放方法不仅需要对经验样本进行优先级的赋予与更新操作,而且还需要频繁地扫描经验缓冲池以获得优先级高的经验样本,因此该算法的时间复杂度较高,训练较慢。

在强化学习中,智能体通过与环境不断交互直至终止状态,来获得情节累积回报。情节累积回报是对智能体一系列交互行为的整体反馈。本文提出的基于情节分类的经验回放方法,将情节累积回报作为经验样本的分类标准,并应用到DDPG算法中。首先,经验样本以情节为单位进行存储,并计算每个情节的累积回报大小,若该值大于累积回报平均值则将该情节样本存储到经验池1中,否则将其存储到经验池2中。在网络模型训练阶段通过着重选取经验池1中的样本来提升网络训练质量。基于情节分类的经验回放方法具有与普通经验回放方法相同级别的时间复杂度,且未增加空间复杂度。在连续动作空间任务中进行实验,结果表明,与采用随机采样的DDPG算法相比,本文提出的EER-DDPG算法有更好的性能表现,并且与基于策略单调提升的置信域策略优化(Trust Region Policy Optimization, TRPO)[14]算法以及近端策略优化(Proximal Policy Optimization, PPO)[15]算法进行实验比较,进一步验证了该算法的有效性。

2 相关工作

2.1 强化学习

强化学习中智能体与环境的交互可以用马尔可夫决策过程(Markov Decision Process, MDP)[16]进行建模。MDP依赖于马尔可夫性,即在序贯决策问题中,在给定当前状态及过去所有状态的情况下,其未来状态的条件概率分布仅依赖于当前状态,与之前的状态无关。一般把强化学习问题定义为一个四元组(S,A,R,P)。其中S为状态集合,$s_t \in S$表示智能体在t时刻的状态;A为动作集合,$a_t \in A$表示智能体在t时刻所采取的动作;R为奖赏函数,$r_{(s_t,a_t)} \in R$表示智能体在状态s_t下采取动作a_t所获得的奖赏;P为状态迁移函数,是智能体在t时刻位于状态s_t下执行动作a_t后迁移到下一状态s_{t+1}的概率。智能体执行的动作由策略π选择,策略π表示在状态s_t下采取动作a_t的映射。如果策略$\pi(s)$的期望回报值大于其他策略的期望回报值,那么策略π就被称为最优策略,用π^*表示。

智能体与环境的交互过程如图1所示,在状态s_t下智能体采取动作a_t,通过环境反馈得到奖赏$r_{(s_t,a_t)}$以及下一个状态

s_{t+1}，智能体根据新的状态重复上述行为直至终止状态，整个过程称为一个情节。

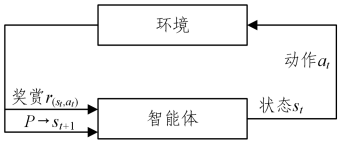


图 1 智能体与环境交互过程

Fig. 1 Interaction process between agent and environment

在每一个情节中累积回报表示为：

$$G_t=\sum_{t'=t}^{T}\gamma^{t'-t}r(s_{t'},a_{t'}) \tag{1}$$

其中，γ 为折扣因子，T 为该情节终止时间步。

状态动作值函数 $Q^\pi(s,a)$ 表示智能体在状态 s 下采取动作 a，并一直遵循策略 π 所获得的期望回报：

$$Q^\pi(s,a)=E_\pi[G_t\mid s_t=s,a_t=a] \tag{2}$$

在强化学习中，状态动作值函数 $Q^\pi(s,a)$ 满足贝尔曼方程形式：

$$Q^\pi(s,a)=E_{r_t,s_{t+1}\sim E}[r(s_t,a_t)+\gamma E_{a_{t+1}\sim\pi}[Q^\pi(s_{t+1},a_{t+1})] \tag{3}$$

通过求解式(3)的状态动作值函数可以解决基于离散动作空间任务的强化学习问题，而针对基于连续动作空间任务的强化学习问题，策略梯度方法提供了解决方式。

在策略梯度方法中，目标函数定义为关于策略的期望回报，即：

$$J(\theta)=E_\pi[\sum_{t=0}^{\infty}\gamma^t r_t] \tag{4}$$

目标函数 $J(\theta)$ 采用策略梯度法求解梯度，进而学习出策略参数 θ。为了使式(4)中的累积回报的期望最大化，使用梯度上升法更新目标函数的策略参数 θ。

$$\theta_{t+1}=\theta_t+\alpha\frac{\partial J}{\partial\theta} \tag{5}$$

2.2 行动者-评论家方法

行动者-评论家(Actor-Critic，AC)[17]方法通过将基于值函数的方法与基于策略梯度的方法进行结合，一方面缓解了策略梯度方法训练方差较大的问题，另一方面解决了值函数方法应用面狭窄的问题。AC 方法由两部分组成，进行动作选择的行动者(actor)和评估策略优劣的评论家(critic)。评论家需要对行动者给出的行为进行反馈，以时间差分误差的形式返回。时间差分误差是指状态 s_t 下的 1 步或者 n 步回报与状态 s_t 下执行动作 a_t 的动作值函数 $Q^\pi(s_t,a_t)$ 的差值：

$$\delta_t=\sum_{i=0}^{n-1}\gamma^i r_{t+i}+\gamma^n Q^\pi(s_{t+1},a_{t+1})-Q^\pi(s_t,a_t) \tag{6}$$

行动者也需要不断根据评论家给予的反馈更新行动。两者相互影响，相互借鉴。AC 算法的模型如图 2 所示。

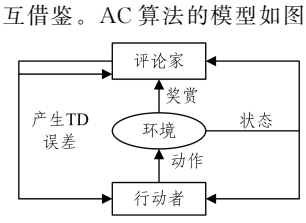


图 2 行动者-评论家算法结构图

Fig. 2 Diagram of actor-critic framework

2.3 深度确定性策略梯度方法

DDPG 方法将深度神经网络与确定性策略梯度(Deterministic Policy Gradients，DPG)[18]算法进行融合，并使用 AC 框架作为该算法的基本架构。DDPG 方法分别使用参数 θ^μ 与 θ^Q 来表示行动者网络 $\mu(s\mid\theta^\mu)$ 和评论家网络 $Q(s,a\mid\theta^Q)$。其中，行动者网络用于更新策略，评论家网络用来逼近状态动作值函数。将非线性神经网络作为近似器时，会使得值函数或者策略出现无法收敛的情况。受 DQN 算法的启发，我们通过设立目标行动者网络 $\mu'(s\mid\theta^{\mu'})$ 和目标评论家网络 $Q'(s,a\mid\theta^{Q'})$ 以及经验回放机制来解决该问题。不同于 DQN 直接将当前网络复制到目标网络，DDPG 通过“soft”方式来进行目标网络更新，保证每次的参数更新幅度较小，从而达到稳定训练的效果。

$$\begin{aligned}\theta^{Q'}&\leftarrow\tau\theta^Q+(1-\tau)\theta^{Q'}\\ \theta^{\mu'}&\leftarrow\tau\theta^\mu+(1-\tau)\theta^{\mu'}\end{aligned} \tag{7}$$

在 DDPG 中，目标函数被定义为带折扣奖赏的和：

$$J(\theta^\mu)=E_{\theta^\mu}[r_1+\gamma r_2+\gamma r_3+\cdots] \tag{8}$$

网络模型训练时，评论家网络通过最小化损失函数 $L(\theta^Q)$ 来更新网络参数：

$$L(\theta^Q)=\frac{1}{N}\sum_i(y_i-Q(s_i,a_i\mid\theta^Q))^2 \tag{9}$$

其中，

$$y_i=r_i+\gamma Q'(s_{i+1},\mu'(s_{i+1}\mid\theta^{\mu'})\mid\theta^{Q'}) \tag{10}$$

行动者网络参数更新通过链式求导法来对目标函数进行求导：

$$\nabla_{\theta^\mu}J(\theta^\mu)\approx\frac{1}{N}\sum_i\nabla_a Q(s,a\mid\theta^Q)\mid_{s=s_i,a=\mu(s_i)}\nabla_{\theta^\mu}\mu(s\mid\theta^\mu)\mid_{s_i} \tag{11}$$

为解决 DPG 方法中因行动者将状态映射到确定动作上造成的探索不足问题，DDPG 算法通过奥恩斯坦·乌伦贝克(Ornstein-Uhlenbeck，OU)[19]过程产生时序相关的噪声，以提高算法在确定性策略下的探索能力。

DDPG 采用基于随机采样的经验回放机制，未考虑到重要性不同的经验样本对智能体学习的不同影响，因此无法高效利用重要性程度高的样本促进网络训练。如何高效划分及采样重要性程度高的经验样本进行网络训练，是经验回放机制面临的主要挑战[20]。

3 基于情节经验回放的 DDPG 算法

3.1 情节分类经验回放

传统经验回放方法通过设立一个经验缓冲池来存储所有经验样本，随后在网络模型训练阶段随机选取批量经验样本进行训练。经验回放机制在消除训练样本之间的时序相关性的同时提高了经验样本的使用效率，提升了算法性能。在进行网络训练的过程中，不同的经验样本对于网络训练的作用不同，等概率选取方法无法高效利用那些重要性程度高的样本。因此，本文从情节累积回报的角度出发，考虑情节累积回报对智能体学习的影响，并采用分类经验回放思想，将不同重要性程度的经验样本分类存储，在网络模型训练时通过选取较多的重要性程度高的经验样本来提升网络训练的质量。

强化学习中智能体与环境交互的方式与人类和环境交互

的方式类似。智能体通过与环境交互产生一系列动作是为了完成某一任务的完整尝试，而累积回报是对这次完整尝试的评价[21]。智能体再次面对相同任务时会利用之前较为成功的经历来学习，这表明智能体能够在成功的经历中学习到更多有效动作。强化学习通过最大化累积回报的方式来寻求最优策略，从累积回报大的情节中选择经验样本进行网络训练能更好地帮助智能体完成任务。

在 EER-DDPG 方法中，经验样本以情节为单位进行存储，根据情节样本重要性程度的不同（累积回报值大小）使用两个经验缓冲池分类存储。初始化网络模型时，设置情节累积回报的平均值为 0，每当一个情节结束，首先更新情节累计回报的平均值，再将该情节的累积回报与之相比，将高于平均值的情节样本存入经验池 1 中，否则将其存入经验池 2 中。在进行网络模型训练时，通过选取较多的经验池 1 中的经验样本来提升网络训练的质量。EER-DDPG 算法的结构如图 3 所示。

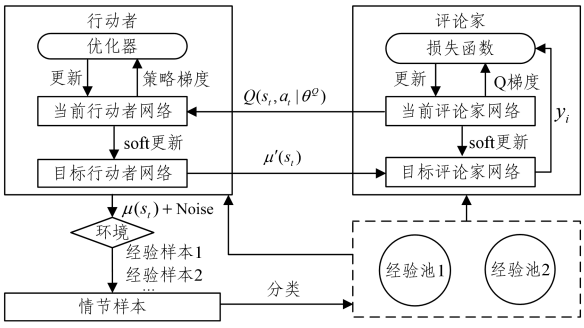


图 3 EER-DDPG 算法的结构示意图

Fig. 3 Structure diagram of EER-DDPG algorithm

在传统优先级经验回放方法中，经验样本使用一个经验缓冲池存储，算法需要根据样本的不同重要性程度赋予相对应的优先级。网络训练时需要频繁扫描经验缓冲池以获得重要性程度高的经验样本，并更新对应优先级。EER-DDPG 方法在存储经验样本前，按其重要性程度不同进行分类存储，减少了赋予以及更新优先级的操作，并且选取批量经验样本进行网络训练时，只需从不同经验缓冲池中随机选取，这一做法大大降低了算法的训练成本。

3.2 算法总结

算法 1 基于情节经验回放的深度确定性策略梯度算法

输入：最大时间步 M，开始训练时间步 S，批次采样数量 N_1 和 N_2，折扣因子 γ，目标网络更新参数 τ

输出：策略网络参数 θ^μ

1. 初始化行动者网络 $\mu(s|\theta^\mu)$ 和评论家网络 $Q(s,a|\theta^Q)$、目标网络参数 $\theta^{\mu'} \leftarrow \theta^\mu$ 和 $\theta^{Q'} \leftarrow \theta^Q$、累积回报平均值、临时经验池 D 以及经验池 D_1 和 D_2
2. 获取初始状态 s_t
3. for t=1 to M do:
4. 选择动作 $a_t = \mu(s_t|\theta^\mu) + \text{Noise}$
5. 执行动作 a_t，获得立即奖赏 r 和下一状态 s_{t+1}
6. 存储经验样本 $e_i = (s_t, a_t, r_t, s_{t+1})$ 到临时经验池 D 中
7. if done:
8. 将临时经验池 D 中经验样本根据累积回报大小分类存储到经验池 D_1 或 D_2 中
9. 清空临时经验池 D
10. if t>s:
11. 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
12. 计算 $y_i = r_i + \gamma k_1$，其中，$k_1 = Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
13. 通过最小化损失函数 $L(\theta^Q)$ 更新评论家网络参数 θ^Q：

$$L(\theta^Q) = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

14. 通过策略梯度方法更新行动者网络参数 θ^μ：

$$\nabla_{\theta^\mu} J(\theta^\mu) = \frac{1}{N}\sum_i k_2 k_3$$

其中，$k_2 = \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)}$，$k_3 = \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$

15. 更新目标网络：

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

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

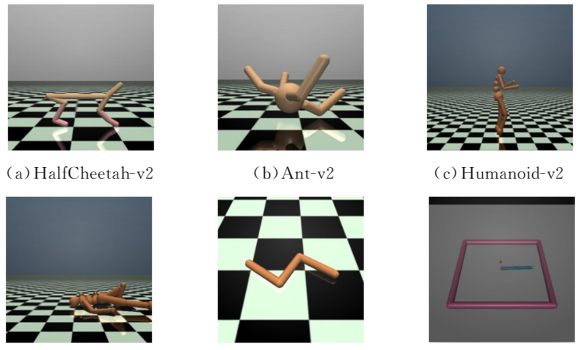
16. end for

算法 1 中，第 4，5 步为经验样本的产生过程，第 6－9 步为经验样本的分类过程，第 11－15 步为网络模型的训练过程。为了方便算法的实现，算法在第 6－9 步中将经验样本先存入临时经验池 D 中，形成以情节为单位的经验样本，在计算相应的累积回报值后分类存储到经验池 D_1 或 D_2 中，最后再清空临时经验池 D，方便下一个情节样本的存储。在第 11－14 步中，算法通过在重要性程度不同的经验池中随机采样不同比例的经验样本来提升性能。

4 实验结果与分析

4.1 实验平台描述

本算法的实验环境采用 OpenAI GYM 平台，在基于 MuJoCo 模拟器的 6 个连续控制任务中进行实验。图 4 为 MuJoCo 任务截图，其中，HalfCheetah-v2 任务使二维猎豹机器人快速奔跑；Ant-v2 任务使三维四足机器人快速行走；Humanoid-v2 任务使三维两足机器人快速行走；Humanoid-Standup-v2 任务使三维两足机器人快速站立；Swimmer-v2 任务使三连杆机器人通过制动两个关节快速向前移动；Reacher-v2 任务使二维机器人到达随机定位的位置。



(a) HalfCheetah-v2 (b) Ant-v2 (c) Humanoid-v2

(d) HumanoidStandup-v2 (e) Swimmer-v2 (f) Reacher-v2

图 4 MuJoCo 任务截图

Fig. 4 Screenshot of MuJoCo tasks

实验硬件平台采用 Inter i7-9700K CPU，NVIDIA RTX2080 GPU，平台内存为 16 GB，操作系统为 Ubuntu18.04。

4.2 实验参数设置

本文提出的方法需要设置两个经验缓冲池的比例参数，若高累积回报的经验缓冲池采样比例 P 过大会导致经验样本的多样性缺失，采样比例 P 过小会导致算法性能提升不明显。为得到适当的比例参数，在 HalfCheetah 任务中设置 P 为0.7，0.8，0.9，1 这 4 组比例参数进行实验，通过对比每组

实验的累积回报值大小来比较算法的性能。从图 5 可知，采样比例 P 取 0.8 时算法性能最优，采样比例 P 取 1 时，算法训练后期会由于样本多样性缺失使得算法性能不佳。因此，设置高累积回报的经验缓冲池采样比例 P 为 0.8。

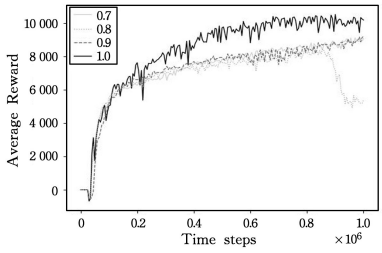


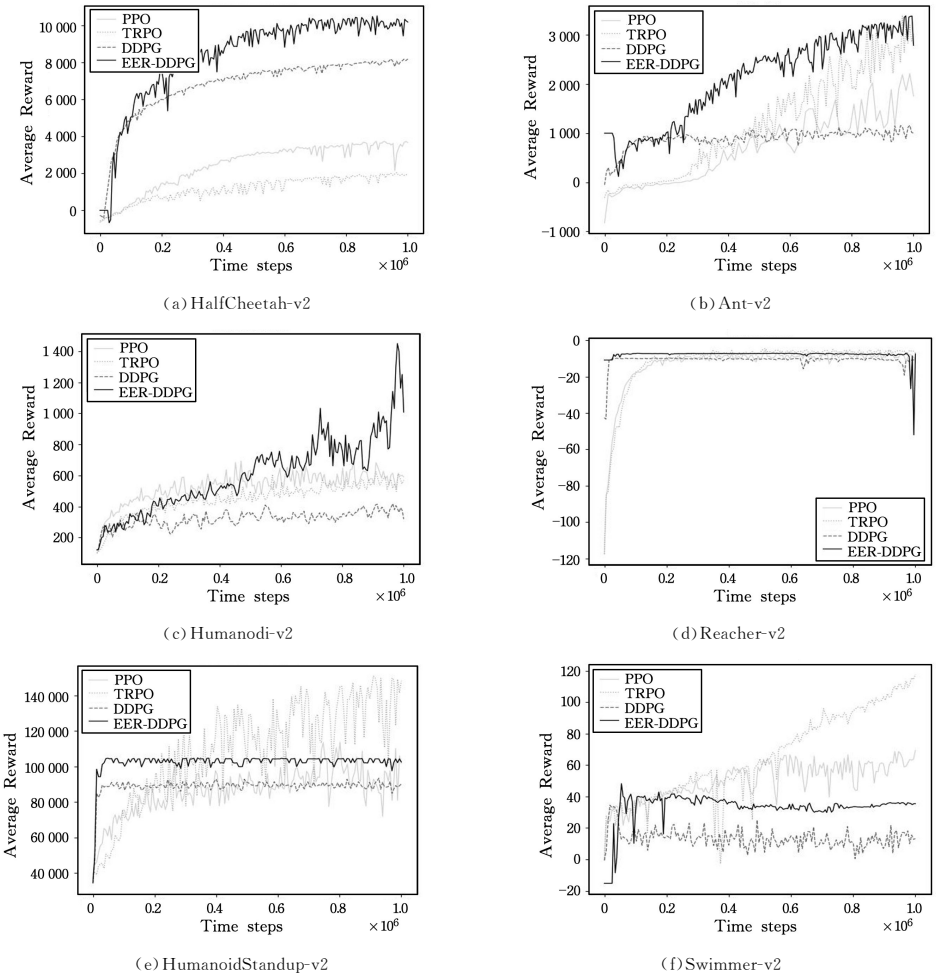
图 5 不同比例参数的实验结果

Fig. 5 Experimental results of different scale parameters

为了保证实验对比的公平性，本实验默认参数设置均与基于随机采样的 DDPG 算法保持一致。TRPO 与 PPO 算法的实验参数均保持与 OpenAI Baseline 官方算法集参数一致。基于随机采样的 DDPG 中，经验缓冲池大小设置为 1 000 000，批次选取经验样本数量为 256。EER-DDPG 中设置经验缓冲池 D_1 和 D_2 均为 500 000，经验缓冲池 D_1 和 D_2 的采样比例分别为 0.8 和 0.2，情节最大时间步数为 1 000，开始训练时间步设置为 25 000，折扣因子 $\gamma=0.99$，目标网络更新参数 $\tau=0.005$。

4.3 实验结果及分析

图 6 给出了在 6 种不同任务中各个算法的性能表现，每个任务训练一百万个时间步，每 5 000 步计算平均累积回报值，通过对比每个任务的累积回报值来衡量算法的优劣。



(a) HalfCheetah-v2 (b) Ant-v2

(c) Humanodi-v2 (d) Reacher-v2

(e) HumanoidStandup-v2 (f) Swimmer-v2

图 6 实验效果对比图

Fig. 6 Comparison of experimental results

如图 6 所示，在所有实验任务中 EER-DDPG 算法的性能表现都优于基于随机采样的 DDPG 算法，在部分任务中甚至超越了基于策略单调提升的 TRPO 和 PPO 算法。实验结果说明，基于情节分类的经验回放方法能够对智能体的学习起到促进作用，在相同的时间步内智能体能够学习到累积回报更高的策略。

在 HalfCheetah，Ant 以及 Humanoid 任务中，EER-DDPG 算法和 DDPG 算法在网络训练前期均能以较快的速度进行学习，网络模型处于平稳上升阶段。但在网络训练中期，DDPG 算法逐渐趋于收敛，EER-DDPG 算法由于是在高累积回报经验池中进行采样，学习到了前期的一些优秀经验，因此在网络训练中后期利用重要性程度高的样本进行训练时仍能够保持较快的速度进行学习，最终表现出明显的性能优势。基于策略单调提升的 TRPO 和 PPO 方法在整个训练周期中都处于稳步上升的状态，但由于这两种方法都采用了在线学习的方式，在训练过程中无法高效利用重要性程度高的

样本提升网络训练质量，因此在网络训练中后期 EER-DDPG 算法逐渐超越了这两种算法。

在 Reacher，HumanoidStandup 以及 Swimmer 任务中，DDPG 算法在网络训练初期已趋于收敛，导致 EER-DDPG 算法在进行情节经验分类时只能学习到训练初期的一些优秀经验，在算法初期有一定性能提升并逐步趋于平稳。算法中后期两个经验缓冲池中的经验样本重要性程度基本相同，EER-DDPG 算法性能提升不明显，最终未能超越 TRPO 和 PPO 这两种基于策略单调提升的在线学习算法。

表 1 列出了 4 种算法在不同任务中的平均奖赏、最高奖赏以及标准差。从表 1 中可以看出，EER-DDPG 算法在大多数任务中均能够取得较高的平均奖赏和最高奖赏，说明本文算法对比于其他 3 种算法有着不错的性能提升。部分任务中由于不同训练阶段累积回报值差异偏大，最终造成 EER-DDPG 算法的标准差略大于其他 3 种算法。

表 1 MuJocC 任务中的实验结果

Table 1 Experimental results in MuJoCo tasks

任务名称	算法	平均奖赏	最高奖赏	标准差
HalfCheetah-v2	DDPG	6742.92	8191.08	1681.88
	PPO	2421.20	3732.71	1234.31
	TRPO	1172.29	2036.27	663.05
	EER-DDPG	8324.02	10517.74	2492.69
Ant-v2	DDPG	889.40	1158.64	192.50
	PPO	706.39	2214.79	697.31
	TRPO	1194.61	3384.38	1035.12
	EER-DDPG	2096.58	3387.82	882.50
Humanoid-v2	DDPG	321.47	418.60	48.23
	PPO	528.88	693.76	97.82
	TRPO	449.79	623.65	99.66
	EER-DDPG	601.26	1450.66	238.55
Reacher-v2	DDPG	−10.85	−9.73	3.48
	PPO	−13.19	−5.35	14.01
	TRPO	−11.78	−4.45	15.47
	EER-DDPG	−7.91	−7.05	3.46
Humanoid-Standup-v2	DDPG	88627.77	93378.53	5616.40
	PPO	86546.59	113756.64	12669.91
	TRPO	108895.97	150891.69	28454.02
	EER-DDPG	102329.45	104816.12	6926.52
Swimmer-v2	DDPG	13.40	34.53	5.80
	PPO	50.20	69.40	12.03
	TRPO	65.43	117.83	27.99
	EER-DDPG	33.18	48.25	10.11

结束语 传统经验回放方法未考虑重要性不同的经验样本对于网络训练的不同作用，采用赋予经验样本优先级的方法增加了算法的时间复杂度与训练成本。强化学习通过最大化累积回报的方式来寻求最优策略，因此本文在探究情节累积回报对网络训练有促进作用的基础上，提出基于情节分类的经验回放方法，并将此方法应用到 DDPG 算法中。本文在连续动作空间任务中进行实验，结果表明，EER-DDPG 算法具有更好的性能表现。本文提出的算法相比同类算法增加了网络训练的波动，训练过程中造成这种网络波动的原因有待进一步探究。如何在提高本文算法性能的同时提高算法的稳定性是下一步工作的研究重点。

参 考 文 献

[1] DORPINGHAUS M, ROLDAN E, NERI I, et al. An information theoretic analysis of sequential decision-making[C]// International Symposium on Information Theory (ISIT). IEEE, 2017: 3050-3054.

[2] QIN Z H, LI N, LIU X T, et al. Overview of Research on Model-free Reinforcement Learning[J]. Computer Science, 2021, 48(3): 180-187.

[3] SUTTON R S, MCALLESTER D A, SINGH S P, et al. Policy gradient methods for reinforcement learning with function approximation[C]// Advances in Neural Information Processing Systems. 2000: 1057-1063.

[4] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2019, 521(7553): 436-444.

[5] TORRADO R R, BONTRAGER P, TOGEL-IUS J, et al. Deep reinforcement learning for general video game[C]// Conference on Computational Intelligence and Games (CLG). IEEE, 2018: 1-8.

[6] KRETZSHMAR H, SPIES M, SPRUNK C, et al. Socially compliant mobile robot navigation via inverse reinforcement learning [J]. The International Journal of Robotics Research, 2016, 35(11): 1289-1307.

[7] LAMPLE G, CHAPLOT D S. Playing FPS games with deep reinforcement learning[C]// AAAI Conference on Artificial Intelligence. 2017: 2140-2146.

[8] ZHAO X, ZHANG L, DING Z, et al. Recommendations with negative feedback via pairwise deep reinforcement learning [C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 1040-1048.

[9] MMIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[10] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. Computer Science, 2015, 8(6): A187.

[11] SCHMIDHUBER J. Deep learning in neural networks: An overview[J]. Neural Networks, 2015, 61: 85-117.

[12] BAI C J, LIU P, ZHAO W, et al. Active Sampling for Deep Q-Learning Based on TD-error Adaptive Correction[J]. Journal of Computer Science & Information Systems, 2019, 56(2): 262-280.

[13] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[J]. arXiv: 1511.05952, 2015.

[14] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization[C] // International Conference on Machine Learning. 2015: 1889-1897.

[15] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv: 1707.06347, 2017.

[16] LEVIN E, PIERACCINI R, ECKERT W. Using Markov decision process for learning dialogue strategies[C]// Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing. 1998: 201-204.

[17] GRONDMAN I, BUSONIU L, LOPES G A D, et al. A survey of actor-critic reinforcement learning: standard and natural policy gradients[J]. IEEE Transactions on Systems, Man, and Cyber-

netics,Part C (Applications and Reviews),2012,42(6):1291-1307.

[18] SILVER D,LEVER G,HEESS N,et al. Deterministic policy gradient algorithms[C]// Proceedings of the International Conference on Machine Learning. 2014:387-395.

[19] UHLENBECK G E,ORNSTEIN L S. On the theory of the Brownian motion[J]. Physical Review,1930,36(5):823.

[20] NOVATI G,KOUMOUTSAKOS P. Remember and forget for experience replay[C]// International Conference on Machine Learning. 2019:4851-4860.

[21] ZHAO Y N,LIU P,ZHAO W,et al. Twice Sampling Method in Deep Q-Network[J]. Acta Automatic Sinica, 2019,45(10):1870-1882.

ZHANG Jian-hang, born in 1995, postgraduate. His main research interests include deep reinforcement learning and so on.

LIU Quan, born in 1969, Ph.D, professor, is a member of China Computer Federation. His main research interests include deep reinforcement learning and automated reasoning.