计算机科学
COMPUTER SCIENCE

基于相似度约束的双策略蒸馏深度强化学习方法

徐平安, 刘全

引用本文

徐平安, 刘全. 基于相似度约束的双策略蒸馏深度强化学习方法[J]. 计算机科学, 2023, 50(1): 253-261.

XU Ping'an, LIU Quan. Deep Reinforcement Learning Based on Similarity Constrained Dual Policy Distillation [J]. Computer Science, 2023, 50(1): 253-261.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.211100167

基于相似度约束的双策略蒸馏深度强化学习方法

徐平安[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 软件新技术与产业化协同创新中心 南京 210000
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 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
(paxu@stu.suda.edu.cn)

摘 要 策略蒸馏是一种将知识从一个策略转移到另一个策略的方法,在具有挑战性的强化学习任务中获得了巨大的成功。典型的策略蒸馏方法采用的是师生策略模型,即知识从拥有优秀经验数据的教师策略迁移到学生策略。获得一个教师策略需要耗费大量的计算资源,因此双策略蒸馏框架(Dual Policy Distillation,DPD)被提出,其不再依赖于教师策略,而是维护两个学生策略互相进行知识迁移。然而,若其中一个学生策略无法通过自我学习超越另一个学生策略,或者两个学生策略在蒸馏后趋于一致,则结合DPD的深度强化学习算法会退化为单一策略的梯度优化方法。针对上述问题,给出了学生策略之间相似度的概念,并提出了基于相似度约束的双策略蒸馏框架(Similarity Constrained Dual Policy Distillation,SCDPD)。该框架在知识迁移的过程中,动态地调整两个学生策略间的相似度,从理论上证明了其能够有效提升学生策略的探索性以及算法的稳定性。实验结果表明,将SCDPD与经典的异策略和同策略深度强化学习算法结合的SCDPD-SAC算法和SCDPD-PPO算法,在多个连续控制任务上,相比经典算法具有更好的性能表现。

关键词:深度强化学习;策略蒸馏;相似度约束;知识迁移;连续控制任务
中图法分类号 TP181

Deep Reinforcement Learning Based on Similarity Constrained Dual Policy Distillation

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

Abstract Policy distillation,a method of transferring knowledge from one policy to another,has achieved great success in challenging reinforcement learning tasks. The typical policy distillation approach uses a teacher-student policy model,where knowledge is transferred from the teacher policy,which has excellent empirical data,to the student policy. Obtaining a teacher policy is computationally intensive,so dual policy distillation(DPD) framework is proposed,which maintains two student policies to transfer knowledge to each other and no longer depends on the teacher policy. However,if one of the student policies cannot surpass the other through self-learning,or if the two student policies converge after distillation,the deep reinforcement learning algorithm combined with DPD degenerates into a single policy gradient optimization approach. To address the problems mentioned above,the concept of similarity between student policies is given,and the similarity constrained dual policy distillation(SCDPD) framework is proposed. The framework dynamically adjusts the similarity between two students' policies in the process of knowledge transfer,and has been theoretically shown to be effective in enhancing the exploration of students' policies as well as the stability

到稿日期:2021-11-16 返修日期:2022-03-19
基金项目:国家自然科学基金(61772355,61702055);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),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)

of algorithms. Experimental results show that the SCDPD-SAC algorithm and SCDPD-PPO algorithm, which combine SCDPD with classical off-policy and on-policy deep reinforcement learning algorithms, have better performance compared with classical algorithms on multiple continuous control tasks.

Keywords Deepre inforcement learning, Policy distillation, Similarity constraint, Knowledge transfer, Continuous control tasks

1 引言

强化学习[1](Reinforcement Learning, RL)是解决序贯决策问题的主流方法,在众多领域中取得了令人瞩目的成果,例如围棋程序 AlphaGo 战胜了人类顶尖选手[2]、控制机器人处理复杂任务[3],以及在自动驾驶领域[4]的广泛应用等。然而,传统的强化学习对高维状态动作空间的感知力不足,难以被广泛应用。随着深度学习(Deep Learning, DL)的蓬勃发展,基于深度神经网络的函数逼近器因其能够有效地提取高维特征信息,而被运用到强化学习中,将深度学习与强化学习融合在一起的方法被称为深度强化学习[5](Deep Reinforcement Learning, DRL)。在高维状态和动作空间的任务中,经过 DRL 算法训练的深度神经网络,可以准确地拟合值函数以及搜索到优秀的策略。

DRL 是人工智能和机器学习领域的一个热点,国内外学者做出了众多杰出的贡献。Minh 等[6]将深度学习和 Q 值学习算法相结合,以多层深度神经网络拟合状态-动作值函数,并引入经验回放机制,提出了深度 Q 网络方法(Deep Q-Network, DQN),该方法采用卷积神经网络提取游戏视频画面的特征信息作为输入状态,经其训练后的智能体达到了超越人类玩家的水平。为了提高训练智能体的效率,Minh 等[7]采用多核 CPU 替代 GPU 运算的方案,以异步并行的方式执行多个智能体与环境进行交互,从而降低训练过程中对硬件的要求,并且提高了训练的稳定性。在高维连续状态和动作空间中,Lillicrap 等[8]将确定策略梯度方法(Deterministic Policy Gradient, DPG)与 DQN 融合,提出了深度确定策略梯度方法(Deep Deterministic Policy Gradient, DDPG),稳定地解决了多个物理模拟的连续控制任务。Scott 等[9]在 DDPG 的基础上加入了双 Q 值学习和延迟策略更新的技术,提出了双延迟深度确定策略梯度方法(Twin Delayed Deep Deterministic Policy Gradient, TD3),以限制值函数被过高估计。Schulman 等[10]提出了置信域优化方法(Trust Region Policy Optimization, TRPO),通过理论证明了选择合适的更新步长可以保证策略被单调优化;同时引入广义优势估计[11](Generalized Advantage Estimation, GAE),有效地降低了方差。基于 TRPO 的基本思想,Schulman 等[12]提出了近端策略优化方法(Proximal Policy Optimization, PPO),通过剪枝操作对新旧策略的概率比值进行限制,从而避免出现更新步长过大的问题。Tuomas 等[13]提出了软行动者-评论家算法(Soft Actor-Critic, SAC),将最大熵引入行动者-评论家框架,使得在策略提升的过程中,其分布的随机性更大,从而增强探索性。

深度强化学习方法通常需要智能体与环境从头开始进行交互,大量的计算资源和时间被消耗后,才能获得较高水平的性能[14]。学习的计算成本取决于任务的复杂性,任务越复杂,计算成本越高,当扩展到现实任务中,往往会因为任务的高复杂性和不可控等因素,而无法取得理想的效果。迁移学习[15]在计算机视觉和自然语言处理等领域取得了显著的成功,因此部分学者提出将知识迁移运用到强化学习中[16]。知识迁移的大致思想是知识通过一个或多个预先训练的教师策略转移到学生策略,教师策略通常具有专家级别的经验。最简单有效的一种知识迁移方法是策略蒸馏[17],即使用监督学习的方法训练学生策略,以达到与教师策略相同输出分布的目标。然而,受限于昂贵的计算资源和实际问题的复杂性,获得具有专家经验的教师策略是困难的,当教师策略非最优时,学生策略如果完全克隆教师策略的知识,那么教师策略中微小的错误也会给学生策略造成数据分布误差,而且在训练的过程中会受到教师策略的制约。

受心理学中协作学习的启发,Lai 等[18]提出了 DPD。该框架使用两个学生策略进行相互知识迁移,其优势是不再需要专门获取教师策略,减小了因训练教师策略所需要的软硬件资源开销。不同于传统的师生策略模型中知识单向的迁移过程,因为参与协作学习的学生策略具有不同的视角和不一致的学习进程,所以知识可在学生策略之间相互迁移。然而,考虑如下两种情况:较差的学生策略无法从自我学习的过程中转变为较优的学生策略,较优的学生策略仅能通过自我学习的方式获取知识;在策略蒸馏的过程中,两个学生策略趋于一致,彼此间难以区分优劣。当上述两种情况发生时,基于 DPD 的方法便会退化为单一策略的强化学习过程,且由于需要维持两个学生策略,反而增大了计算资源的消耗。

针对上述问题,本文在 DPD 的基础上,给出了学生策略之间相似度的概念,并且提出了 SCDPD。该框架将学生策略之间的相似度控制在可信的范围内,与经典的深度强化学习算法结合后能够有效地提升算法性能。SCDPD 如图 1 所示,策略 1 和策略 2 与环境分别进行交互,在强化学习算法的指导下不断提升性能,与此同时,框架中的两个策略在相似度被约束的条件下,相互之间不断地进行知识蒸馏,防止学生策略出现高度一致性的情况,从而达到自我学习和知识迁移的平衡。

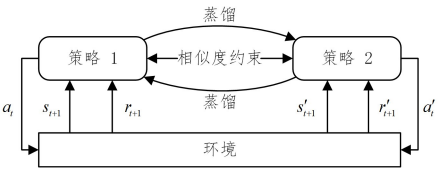


图 1 基于相似度约束的双策略蒸馏框架

Fig. 1 Framework of similarity constrained dual policy distillation

本文工作的主要贡献可以总结为以下 3 点:

(1)给出了学生策略之间相似度的概念,并且将约束相似度的方法与 DPD 结合,提出了 SCDPD。

(2)为 SCDPD 提供了理论依据,证明本文方法在稳步提升策略的同时,不仅加强了策略的探索性,而且还提升了算法的稳定性。

(3)在理论基础上,将 SCDPD 扩展到经典的深度强化

学习算法中，在 4 个连续控制任务中进行对比实验，实验结果验证了本文方法的优越性。

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使用马尔可夫决策过程(Markov Decision Process，MDP)进行建模，该模型由五元组 $M=(S,A,p,r,\gamma)$ 组成，其中 S 为状态空间，状态 $s_t\in S$ 表示智能体在 t 时刻的状态；A 为动作空间，动作 $a_t\in A$ 表示智能体在 t 时刻采取的动作；状态迁移函数 $p(s_{t+1}|s_t,a_t)$ 描述了状态动作对映射到下一个状态的概率分布；奖赏函数 r_{t+1} 表示在状态 s_t 下采取动作 a_t 获得的奖励；$\gamma\in(0,1]$ 表示折扣因子。

智能体与环境交互如图 2 所示，在状态 s_t 下，智能体采取动作 a_t，得到环境反馈的奖励 r_{t+1} 和下一个状态 s_{t+1}。智能体重复上述操作至终止状态会产生一条轨迹 $\tau=\{(s_t,a_t,r_{t+1})\}_{t=1,\cdots,T-1}$，其中 T 是终止状态。强化学习的目标是学习到最优策略 π^*，使得智能体采取服从策略 π^* 分布的动作 a 后能够获得最大的累积奖赏 $\sum_{t\geqslant 0}^{T-1}\gamma^t r(s_t,a_t,s_{t+1})$，即训练策略的最终目标是在接收到输入的状态后，能够输出最优的动作 a^*。

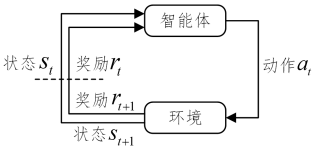


图 2 智能体与环境交互过程

Fig. 2 Interaction between agent and environment

在强化学习中，将与环境交互产生采样数据的策略称为行为策略，与之相对应的是目标策略，目标策略是强化学习任务中需要评估和改进的策略。根据行为策略和目标策略是否一致，强化学习方法可以分为同策略和异策略强化学习方法。在同策略强化学习方法中，行为策略和目标策略相同，因其实现相对简单，在实际应用中通常会优先考虑使用同策略方法，但由于行为策略和目标策略相同，探索性受到影响，效果差于异策略方法。反之，行为策略与目标策略不一致的强化学习方法是异策略强化学习方法，异策略方法可以根据示例样本或者其他智能体给出的经验数据进行学习，相比同策略方法，具有更好的效果和适用性，但存在方差大和收敛速度慢等问题。

2.2 策略蒸馏

策略蒸馏在 Rusu 等[19]的工作中首次被提出，其创新性地将知识迁移技术[20]融入到强化学习的策略训练过程中，吸引了国内外众多学者参与研究。师生策略模型的知识迁移是最典型的策略蒸馏方式，已经在许多领域取得了成功。

在多智能体强化学习领域中，Wadhwania 等[21]提出了一种新的多智能体行动者-评论家算法，通过策略蒸馏的方式结合同质的智能体知识，使得在离散和连续的动作空间中能够进行更深层的学习。为了加强多智能体之间的协作，Chen[22]提出了利用全局智能体蒸馏派生新智能体的框架，促进智能体间有效地协作学习。在元学习方法中，利用基于师生策略模型的策略蒸馏方法可以提高样本效率[23-24]。在实际应用中，Fang 等[25]提出了一种新的量化交易框架，该框架采用策略蒸馏的方法，将实际交易的教师策略的知识经验迁移给普通学生策略。此外，策略蒸馏技术还被应用于多分类问题[26]、规划控制问题[27]和模拟汽车驾驶[28]等领域。

目前，基于策略蒸馏方法的工作大部分采用师生策略模型，采用多学生策略模型的策略蒸馏方法是新的研究热点。Lai 等[18]首次提出了双学生策略模型，使用两个学生策略互相进行知识迁移，摆脱了对教师策略的依赖。Zhao 等[29]将多个学生策略分配到多个不同的环境中，通过 KL 散度相互正则化的方式迁移多个学生策略的知识，从而获得优秀的泛化性能。Cha 等[30]提出了联合强化蒸馏方法，通过提取每个学生策略的经验数据进行知识迁移。Sun 等[31]提出了随机进化策略蒸馏方法，该方法遵循进化策略，选取最优的经验数据，通过策略蒸馏的方式对目标策略进行更新。

3 SCDPD

本节详细介绍了 SCDPD 的工作原理，并且对其进行了理论分析。DPD 是一种能够在同一任务环境下，控制两个学生策略相互之间进行知识迁移的框架。在双策略蒸馏的基础上，对策略之间的知识迁移加以限制，保证其相似度在可信的阈值范围内，使得策略在提升过程中获得更多的环境反馈信息，从而在训练进程中获得更多提升。

在给定的任务中，考虑具有相同网络结构的两个策略 π 和 π'，首先从理论上证明，混合上述两个策略后，从对等的策略中提取知识并相互迁移可以使得策略性能稳步提升。然后证明了在策略蒸馏的过程中，限制两个策略在策略蒸馏过程中的相似度，可以保证两个策略具有更强的探索性以及更好的算法稳定性。最后，基于上述理论，提出了与 DRL 算法相结合的策略知识来提取目标函数，并给出了相应的算法描述。

3.1 策略混合

策略混合指将同网络结构的两个策略以特定的方式混合在一起，目的是使得在混合的两个策略之间进行知识迁移时，两个策略均会得到稳步提升。定义一个混合策略 $\tilde{\pi}(\cdot|s)$ 如下：

$$\tilde{\pi}(\cdot|s)=\begin{cases}\pi(\cdot|s), & \xi^{\pi}(s)\geqslant 0\\ \pi'(\cdot|s), & \xi^{\pi}(s)<0\end{cases}\tag{1}$$

其中，$\xi^{\pi}(s)$ 作为策略混合条件，定义了在状态 s 处，策略 π 相对于策略 π' 针对给定任务的优势，其表达式为 $\xi^{\pi}(s)=V^{\pi}(s)-V^{\pi'}(s)$。混合策略的选取标准是在状态 s 下，选择状态值较大的策略。定理 1 证明了在理想情况下，混合策略可以稳步提升。

定理 1 对于任意的状态 $s\in S$，式(1)定义的混合策略 $\tilde{\pi}$ 能够稳步提升，即 $n\geqslant 0$ 时，满足条件 $V_{n+1}(s)\geqslant V_n(s)$。

证明：根据式(1)可知，对于任意的状态 $s\in S$，混合策略 $\tilde{\pi}$ 始终满足如下条件：

$$V^{\tilde{\pi}}(s)\geqslant V^{\pi}(s),V^{\tilde{\pi}}(s)\geqslant V^{\pi'}(s)\tag{2}$$

混合策略是基于环境状态 s 定义的，由式(2)可知，在状态 s 处，混合策略 $\tilde{\pi}$ 即为当前最优策略。对于其他状态 s'，存在：

$$V^{\tilde{\pi}_s}(s)\geqslant V^{\tilde{\pi}_{s'}}(s)\tag{3}$$

其中，$\tilde{\pi}_s$ 表示在状态 s 处的混合策略，并且满足条件 $s'\neq s$。

考虑在特定任务中，定义一个任意的状态为 $s_t\in S$，则下一个状态可表示为 s_{t+1}。状态 s_t 的状态值函数的定义如下：

$$V_n(s_t)=\begin{cases}V^{\tilde{\pi}_{s_t}}(s_t), & n=0\\ E_{\tilde{\pi}_{s_t}}[r_{t+1}+\gamma V_{n-1}(s_{t+1})], & n\geqslant 1\end{cases} \tag{4}$$

其中，n 是一个正整数，表示状态值函数当前迭代的次数。当 $n=0$ 时，$V_0(s_t)=V^{\tilde{\pi}_{s_t}}(s_t)$，首先证明，对于任意的状态 $s_t\in S$，$V_1(s_t)\geqslant V_0(s_t)$。

$$\begin{aligned}V_1(s_t)&=E_{\tilde{\pi}_{s_t}}[r_{t+1}+\gamma V_0(s_{t+1})]\\&=\sum p^{\tilde{\pi}}(s_{t+1},r_{t+1}|s_t)[r_{t+1}+\gamma V_0(s_{t+1})]\\&=\sum p^{\tilde{\pi}}(s_{t+1},r_{t+1}|s_t)[r_{t+1}+\gamma V^{\tilde{\pi}_{s_{t+1}}}(s_{t+1})]\\&\geqslant\sum p^{\tilde{\pi}}(s_{t+1},r_{t+1}|s_t)[r_{t+1}+\gamma V^{\tilde{\pi}_{s_t}}(s_{t+1})]\\&=\sum p^{\tilde{\pi}_{s_t}}(s_{t+1},r_{t+1}|s_t)[r_{t+1}+\gamma V^{\tilde{\pi}_{s_t}}(s_{t+1})]\\&=V^{\tilde{\pi}_{s_t}}(s_t)\\&=V_0(s_t)\end{aligned} \tag{5}$$

基于上述的证明结果，当 $n\geqslant 1$ 时，对于任意的状态 $s_t\in S$，可以证明 $V_n(s_t)\geqslant V_{n-1}(s_t)$。

$$\begin{aligned}V_{n+1}(s_t)&=E_{\tilde{\pi}_{s_t}}[r_{t+1}+\gamma V_n(s_{t+1})]\\&=\sum p^{\tilde{\pi}}(s_{t+1},r_{t+1}|s_t)[r_{t+1}+\gamma V_n(s_{t+1})]\\&\geqslant\sum p^{\tilde{\pi}}(s_{t+1},r_{t+1}|s_t)[r_{t+1}+\gamma V_{n-1}(s_{t+1})]\\&=E_{\tilde{\pi}_{s_t}}[r_{t+1}+\gamma V_{n-1}(s_{t+1})]\\&=V_n(s_t)\end{aligned} \tag{6}$$

通过归纳法，得出对于任意 $n\geqslant 0$ 和 $s\in S$，$V_{n+1}(s)\geqslant V_n(s)$，在复杂的连续环境任务中，通过积分的形式亦可得到当前结果，证毕。

混合策略的优势在于，在策略迭代的过程中，不再以更新单一策略的方式完成迭代过程，取而代之的是从两个同网络结构的策略中选取较为优秀的策略作为当前训练阶段的最终策略。从理论上证明了更新后的策略优于先前策略，即满足策略可以被不断提升的特征。

3.2 优势策略蒸馏

混合策略是在两个策略中遴选而来的，其拥有当前最好的经验知识，将混合策略 $\tilde{\pi}$ 的优秀经验知识迁移到策略 π 和策略 π'，显然能够加速所有策略的提升过程。

在深度学习的训练过程中，通常是将批量的经验数据输入给神经网络，计算输出的损失函数，在反向传播后，进行神经网络的梯度更新。为了实现更好的拟合效果，取出的数据量必定不会太小，如果每一次都先判断出假设的混合策略后，再进行策略蒸馏会增加无用的计算量。最理想的情况是，可以通过经验数据直接进行蒸馏，减少判断出混合策略所带来的资源开销。定理 2 证明了混合策略 $\tilde{\pi}$ 向策略 π 和策略 π' 的知识迁移，等价于策略 π 和策略 π' 之间的优势蒸馏。

定理 2 从混合策略 $\tilde{\pi}$ 到策略 π 的策略蒸馏等价于最小化如下目标函数：

$$J_\pi=E_{s\sim\pi'}[D(\pi(\cdot|s),\pi'(\cdot|s))\varepsilon(\xi^{\pi'}>0)] \tag{7}$$

其中，$D(\pi(\cdot|s),\pi'(\cdot|s))$ 代表策略 π 和策略 π' 的距离度量值，$\varepsilon(\cdot)$ 表示使能函数，当函数内的判断内容成立时，$\varepsilon(\cdot)=1$；当判断内容不成立时，$\varepsilon(\cdot)=0$。

证明：由于策略 π 和策略 π' 具有相似的访问频率，因此从混合策略 $\tilde{\pi}$ 开始的策略蒸馏过程可以写成如下形式：

$$\begin{aligned}&E_{s\sim\tilde{\pi}}[D(\pi(\cdot|s),\tilde{\pi}(\cdot|s))]\\&=\sum_{s\sim\tilde{\pi}}D(\pi(\cdot|s),\tilde{\pi}(\cdot|s))\\&=\sum_{s\sim\tilde{\pi},\xi^{\pi'}>0}D(\pi(\cdot|s),\pi'(\cdot|s))+\sum_{s\sim\tilde{\pi},\xi^{\pi'}\leqslant 0}D(\pi(\cdot|s),\pi(\cdot|s))\\&=\sum_{s\sim\pi',\xi^{\pi'}>0}D(\pi(\cdot|s),\pi'(\cdot|s))+\sum_{s\sim\pi,\xi^{\pi'}\leqslant 0}D(\pi(\cdot|s),\pi(\cdot|s))\\&=E_{s\sim\pi'}[D(\pi(\cdot|s),\pi'(\cdot|s))\varepsilon(\xi^{\pi'}>0)]\end{aligned} \tag{8}$$

证毕。同理，对于策略 π'，从混合策略 $\tilde{\pi}$ 开始的策略蒸馏过程可写成如下形式：

$$J_{\pi'}=E_{s\sim\pi}[D(\pi'(\cdot|s),\pi(\cdot|s))\varepsilon(\xi^{\pi}>0)] \tag{9}$$

根据上述证明过程不难发现，当一个策略相对于另一个策略具有优势时，知识很容易被蒸馏而转移过去；然而，在不具备优势时，使能函数将会屏蔽策略蒸馏的过程。

3.3 相似度约束

DPD 的核心思想是将优秀策略的知识通过知识迁移的方式，传递给另一个策略，使得该策略即使不与环境交互进行试错学习，也能够获得性能的提升。DPD 的优点是不仅能从与环境交互的经验中学习知识，还能够从同结构的其他策略中获得知识。

然而，策略过度地进行知识迁移会使得在 DPD 中，两个策略趋于一致，对于相同的环境状态 s，输出十分相似的动作，导致策略搜索到更好动作的能力减弱，即探索性降低，从而制约了 DPD 的优势。当极端情况出现时，即经过知识迁移后，两个策略完全一致，或者每一次迭代后，较差的策略都无法进化成较优的策略，基于 DPD 的方法将会退化为单一策略的梯度优化过程。

为了解决该问题，在双策略蒸馏的框架中引入相似度约束方法，定义同网络结构的两个策略 π 和 π' 之间的相似度为 $\eta(\pi(\cdot|s),\pi'(\cdot|s))$，其是相似度的广义表达，对其最直观的理解是策略 π 和 π' 在状态 s 下动作分布的分散程度。当扩展到具体任务时，相似度可视任务的具体情况而定，通常使用 KL 散度形式化地描述两个策略间的相似性。定义策略 π 和策略 π' 的动作搜索空间为 $\rho(\pi)$ 和 $\rho(\pi')$，则混合策略的动作搜索空间为：

$$\rho(\tilde{\pi})=\rho(\pi)\bigcup\rho(\pi') \tag{10}$$

定理 3 当 $\eta(\pi(\cdot|s),\pi'(\cdot|s))$ 趋于 0 时，混合策略 $\tilde{\pi}$ 的动作搜索空间越小；当 $\eta(\pi(\cdot|s),\pi'(\cdot|s))$ 越大时，混合策略 $\tilde{\pi}$ 的动作搜索空间越大。

证明：根据式(10)，可写出混合策略动作搜索空间的极限表达式：

$$\begin{aligned}&\lim_{\eta(\pi(\cdot|s),\pi'(\cdot|s))\to 0}\rho(\tilde{\pi}(\cdot|s))\\&=\lim_{\eta(\pi(\cdot|s),\pi'(\cdot|s))\to 0}\rho(\pi(\cdot|s))\bigcup\rho(\pi'(\cdot|s))\\&=\begin{cases}\rho(\pi(\cdot|s)), & \xi^{\pi}(s)\geqslant 0\\ \rho(\pi'(\cdot|s)), & \xi^{\pi}(s)<0\end{cases}\end{aligned} \tag{11}$$

由式(11)可知，当 $\eta(\pi(\cdot|s),\pi'(\cdot|s))$ 趋于 0 时，表示策略 π 和策略 π' 越相似，在极限值处时，混合策略的动作搜索空间退化为在状态 s 处的单一策略的动作搜索空间；反之，混合策略的动作搜索空间随相似度 $\eta(\pi(\cdot|s),\pi'(\cdot|s))$ 的增大而增大，即策略 π 和策略 π' 趋于分散，其动作搜索空间差异越大，探索性就越强，证毕。

优秀动作经验是提升策略的关键因素，强有力的探索能够提高获得优秀动作经验的概率，但是大量仿真实验表明，过高的探索性往往会导致无用的动作经验被收集，对策略的

学习没有实质帮助。因此,在双策略蒸馏的框架中,不仅要限制策略 π 和策略 π' 的相似度 $\eta(\pi(\cdot|s),\pi'(\cdot|s))$ 不能过小,还要约束其不能过大。在策略的更新过程中,引入如下目标函数,约束策略 π 和策略 π' 的相似度:

$$L_\pi^\eta = E_{s\sim\pi}[\| \eta(\pi(\cdot|s),\pi'(\cdot|s)) - \varphi_1 \|_2 + \| \eta(\pi(\cdot|s),\pi'(\cdot|s)) - \varphi_2 \|_2] \quad (12)$$

$$L_{\pi'}^\eta = E_{s\sim\pi'}[\| \eta(\pi(\cdot|s),\pi'(\cdot|s)) - \varphi_1 \|_2 + \| \eta(\pi(\cdot|s),\pi'(\cdot|s)) - \varphi_2 \|_2] \quad (13)$$

其中,φ_1 和 φ_2 为相似度常数,且满足条件 $\varphi_1 \leqslant \varphi_2$,$\|\cdot\|_2$ 表示相似度与常数之间的欧氏距离,式(12)、式(13)的目的是通过梯度优化的方式,使得策略 π 和策略 π' 的相似度 $\eta(\pi(\cdot|s),\pi'(\cdot|s))$ 被约束在区间$[\varphi_1,\varphi_2]$内。

策略在提升的过程中会逐渐逼近最优策略,因此一个好的训练算法在训练开始阶段应满足高探索性的特点,当策略相当接近最优策略时,应逐渐降低探索性,保证策略能够缓慢且稳定地更新至最优策略。而当 φ_1 和 φ_2 固定时,会使得策略 π 和 π' 在训练的后期,依旧保持较高的分散程度,不利于算法达到收敛状态,在训练过程中动态地调整 φ_1 和 φ_2 的值趋于 0,有助于算法的稳定性。φ_1 和 φ_2 的更新公式如下:

$$\varphi_1 = \hat{\varphi}_1(1-n/M), \varphi_2 = \hat{\varphi}_2(1-n/M) \quad (14)$$

其中,n 表示当前的迭代次数,M 表示最大迭代次数,$\hat{\varphi}_1$ 和 $\hat{\varphi}_2$ 为固定值,在训练开始前进行初始化。

SCDPD 的描述如算法 1 所示。

算法 1 SCDPD

输入:相同网络结构的策略 π 和 π',最大迭代次数 M,相似度常数 $\hat{\varphi}_1$ 和 $\hat{\varphi}_2$

输出:训练好的策略 π 和 π'

1. 随机初始化策略 π 和 π'
2. for k← 1 to M do
3. 策略 π 与环境交互生成轨迹信息
4. 将策略 π 的经验存入经验缓冲池 B_π
5. 根据深度强化学习策略目标函数更新 π
6. 根据式(7)、式(12)和构建的目标函数更新 π:
   $L_\pi = J_\pi + L_\pi^\eta$
7. 策略 π' 与环境交互生成轨迹信息
8. 将策略 π' 的经验存入经验缓冲池 $B_{\pi'}$
9. 根据深度强化学习策略目标函数更新 π'
10. 根据式(9)、式(13)和构建的目标函数更新 π':
    $L_{\pi'} = J_{\pi'} + L_{\pi'}^\eta$
11. end for

第 1 行将相同网络结构的两个策略进行随机初始化,保证两个策略在首次与环境交互后,可以获得不同的经验数据,从而区分出策略的优劣。第 3－6 行和第 7－10 行分别表示策略 π 和策略 π' 的更新过程,两个策略的更新过程具有高度一致性,均是先将策略与环境进行交互,然后把收集到的经验分别存入相应的经验缓冲池中,并且依据对应的强化学习算法中的策略优化目标函数对策略进行梯度优化,最后采用基于策略蒸馏方法的目标函数和相似度约束条件的目标函数对策略进行梯度优化,使得策略接受知识迁移的同时受到相似度的约束,从而获得更强的探索性和更好的稳定性。

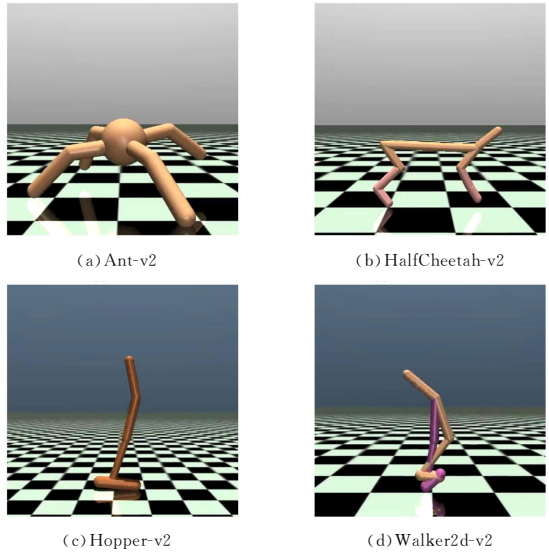
4 实验及分析

将本文提出的 SCDPD 分别应用到异策略和同策略方法中,并且在 4 个连续的控制任务中评估算法的性能。

4.1 实验环境

OpenAI Gym[32] 是面向强化学习开发和算法对比的开源工具包,其中包含众多的基准测试和接口,如 Atali 2600 游戏、Mujoco 物理引擎和模拟机械臂等,为人工智能开发者和研究者提供了可移植的环境以及具有挑战性的任务。

如图 3 所示,本文采用 OpenAI Gym 中的 Mujoco 物理引擎作为实验环境。为了有效评估算法的性能,选取 Mujoco 物理引擎模拟的连续控制任务 Ant-v2,HalfCheetah-v2,Hopper-v2 和 Walker2d-v2 进行实验,其简要介绍如表 1 所列。



(a) Ant-v2 (b) HalfCheetah-v2 (c) Hopper-v2 (d) Walker2d-v2

图 3 Mujoco 物理引擎环境

Fig. 3 Mujoco physics engine environment

表 1 实验任务简要介绍

Table 1 Introduction of tasks

任务	状态维度	动作维度	任务目标
Ant-v2	111	8	训练一个四足的智能体尽可能快地行走
HalfCheetah-v2	17	6	训练一个两足的智能体尽可能快地奔跑
Hopper-v2	11	3	训练一个单腿的智能体尽可能快地向前跳
Walker2d-v2	17	6	训练使二维双足智能体尽可能快地向前走

如图 4 所示,以任务 HalfCheetah-v2 为例,维度为 17 的状态信息包含了智能体的当前速度和各关节角度等信息。动作$[a_1,a_2,a_3,a_4,a_5,a_6]$是维度为 6 的向量,依次对应图 4 中标记出的关节处所需要的角度值。任务的目标是训练出一个策略,能够根据当前环境给出的 17 维状态信息,输出一个 6 维的动作,智能体在执行该动作后,保持较高速度的奔跑状态。本文所有实验均采用配置为 Intel Xeon E5-2680 v4 CPU、2 块 NVIDIA Tesla P40 GPU 和 128GB 内存的服务器作为实验平台。

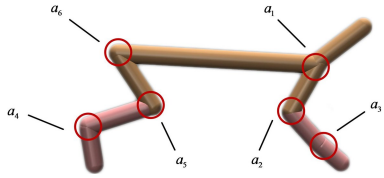


图 4 HalfCheetah-v2 任务示意图

Fig. 4 Illustration of HalfCheetah-v2

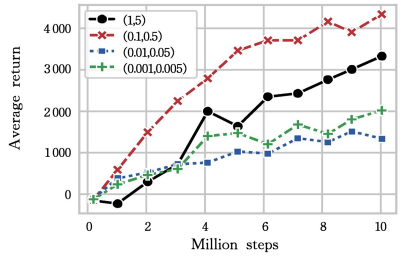
4.2 实验设置

为了验证本文方法与异策略和同策略深度强化学习方法结合后能否取得较好的效果，选取经典的异策略算法 SAC 和同策略算法 PPO 作为基础算法。将 SCDPD 与上述两种算法组合，构成 SCDPD-SAC 算法和 SCDPD-PPO 算法。同时，使用 KL 散度值描述策略 π 和策略 π' 的相似度，相似度约束即简化成控制策略 π 和策略 π' 的 KL 散度保持在指定区间内。

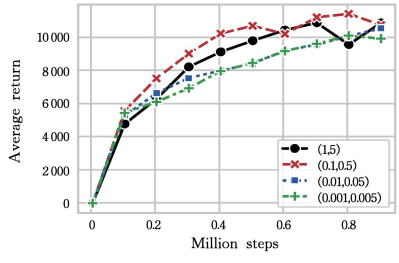
Lai 等[18]结合 DPD，扩展 DDPG 算法为 DPD-DDPG 算法，以验证与异策略强化学习结合的双策略蒸馏方法具有的优越性。本文不选择 DDPG 算法进行扩展的原因是 KL 散度是衡量随机变量的指标，而 DDPG 算法采用的是确定策略，无法使用 KL 散度描述两个确定策略的相似度，因此选取随机策略的 SAC 算法作为异策略算法，与基于相似度约束的双策略蒸馏方法结合。

衡量本文提出的框架结合异策略强化学习方法的性能时，将 SCDPD-SAC 算法与 SAC 算法分别在 4 个连续任务中与环境交互 100 万时间步；比较本文提出的框架结合同策略强化学习方法的性能时，将 SCDPD-PPO 算法、DPD-PPO 算法和 PPO 算法分别在 4 个连续任务中与环境交互 1 000 万时间步。为了保证对比实验的公平性，基于 SCDPD 或 DPD 的 SCDPD-SAC 算法、SCDPD-PPO 算法和 DPD-PPO 算法与其基础算法的所有超参数保持一致，且在交互至相同的时间步时，SCDPD-SAC 算法、SCDPD-PPO 算法和 DPD-PPO 算法中的任意一个策略网络与环境交互的次数仅为对应的基础算法的一半，从而保证每种算法与环境交互的总次数一致。在以 PPO 算法为基础的所有算法中，行动者网络和评论家网络的学习率均为10^{-4}，折扣因子设为 0.99，广义优势估计器的参数 λ 设为 0.95，截断函数的参数 ε 设为 0.2。在以 SAC 算法为基础的所有算法中，行动者网络和评论家网络的学习率均为10^{-3}，折扣因子设为 0.99，经验缓冲池容量设为 1 000 000；温度参数 α 设为 0.2，目标网络平滑参数 τ 设为 0.005。

本文方法需要设定相似度常数$(\hat{\varphi}_1, \hat{\varphi}_2)$的初始值，以达到控制两个学生策略相似性的目的。较大的相似度常数会导致策略进行过多的无用探索，较小的相似度常数使得策略趋于一致。为了得到适当的相似度常数，如图 5 所示，在 HalfCheetah-v2 任务中，设置相似度常数$(\hat{\varphi}_1, \hat{\varphi}_2)$的值分别为(1,5)，(0.1,0.5)，(0.01,0.05)，(0.001,0.005)进行对比实验，通过比较平均累积回报的方式来评价不同参数的性能。由图 5 可知，当相似度常数$(\hat{\varphi}_1, \hat{\varphi}_2)$的值取(0.1,0.5)时，SCDPD-PPO 算法和 SCDPD-SAC 算法的性能最优。因此，设定相似度常数$(\hat{\varphi}_1, \hat{\varphi}_2)$的初始值为(0.1,0.5)。



(a) SCDPD-PPO



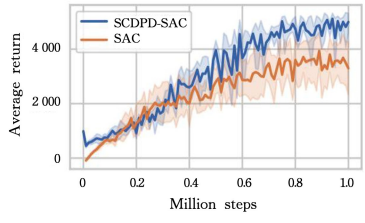
(b) SCDPD-SAC

图 5 不同相似度常数的实验结果

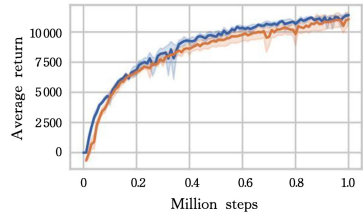
Fig. 5 Experimental results of different similarity constants

4.3 实验结果分析

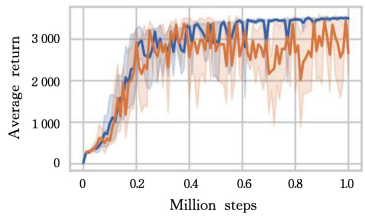
SAC 算法和 SCDPD-SAC 算法以 5 个不同的随机种子，相互独立地与环境交互 100 万步的学习曲线如图 6 所示，其中实线表示相应的算法以 5 次独立运行所训练的策略在当前环境步数评测后得到的性能均值，阴影部分为对应的波动，且阴影部分越大，表示训练该策略的算法的稳定性越差。



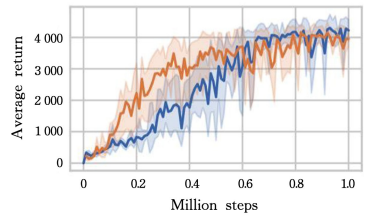
(a) Ant-v2



(b) HalfCheetah-v2



(c) Hopper-v2



(d) Walker2d-v2

图 6 SCDPD-SAC 算法和 SAC 算法的学习曲线

Fig. 6 Learning curves of SCDPD-SAC and SAC

在所提供的 4 个连续控制任务环境中，SCDPD-SAC 算法均取得了比 SAC 算法更好的性能。由于不同任务的 MDP 模型存在差异，SCDPD-SAC 算法在不同任务上相对于 SAC 算法的提升幅度也不同。增强探索性有助于获得较优的经验数据，优秀的经验数据对策略有正向提升的作用。当策略运行在一个比较容易获得优秀经验数据的 MDP 模型上时，增强探索性难以大幅度提升性能。反之，当一个任务的 MDP 模型使策略不易通过正常的交互过程得到优秀的经验数据，增强探索性便有益于策略性能大幅提升。

在 Ant-v2 任务中，SAC 算法的学习曲线在 30 万时间步时，存在略高于 SCDPD-SAC 算法的情况。而在 Walker2d-v2 任务中，SAC 算法的学习曲线在训练前期高于 SCDPD-SAC 算法，其原因在于 SCDPD-SAC 算法保持着两个策略同时进行学习，当时间步相同时，DPD 和 SCDPD 中的任一策略仅与环境交互了一半的时间步，虽然优秀经验会通过策略蒸馏的方式进行知识迁移，但依然无法弥补与环境交互不足所带来的缺陷。随着与环境交互的步数增多，SCDPD-SAC 算法凭借相似度约束和优势策略蒸馏所积累的优势，使得策略在训练过程中获得更优秀的探索性和蒸馏而来的优秀策略经验。最终在任务 Ant-v2 和 Walker2d-v2 上，分别获得了 36.5%和 5.6%的性能提升。

在 Hopper-v2 环境上，根据学习曲线可以得出，相比 SAC 算法，SCDPD-SAC 算法在训练后期更加稳定。随着与环境交互步数的增加，SCDPD-SAC 算法控制策略 π 和策略 π' 的相似度值逐渐变小，即两个策略会在训练的后期，在最优值附近相互制约，降低无效或劣势的经验数据对策略的影响。

表 2 列出了 PPO 算法、DPD-PPO 算法与 SCDPD-PPO 算法经过 1000 万时间步训练后，在 4 个连续环境上的最终性能值。基于同策略的算法由于生成样本的策略与更新改进的策略一致，因此探索性受到限制。从表 2 的数据可知，在 DPD-PPO 算法基础上提高探索性后的 SCDPD-PPO 算法比 DPD-PPO 算法拥有更好的性能，相比异策略的 SCDPD-SAC 算法，SCDPD-PPO 算法在基础算法上的性能提升幅度更大。SCDPD-PPO 算法不仅在最终的性能结果上有优势，而且相比 PPO 算法和 DPD-PPO 算法，SCDPD-PPO 的标准差最小。标准差越小，代表策略在训练过程中受到劣势经验数据的影响越小，算法的稳定性越好。

表 2 PPO，DPD-PPO 和 SCDPD-PPO 算法的最终性能

Table 2 Performance of PPO，DPD-PPO and SCDPD-PPO

Tasks	PPO	DPD-PPO	SCDPD-PPO
Ant-v2	2963.42± 352.13	3695.29± 306.21	**4109.73± 284.57**
HalfCheetah-v2	3041.58± 203.44	3067.56± 192.67	**3965.81± 153.29**
Hopper-v2	2437.52± 125.63	2567.28± 111.48	**2872.16± 92.81**
Walker2d-v2	3677.62± 219.35	3794.32± 147.12	**3954.17± 133.16**

本文提出的 SCDPD-PPO 算法整体性能大幅提升的原因是，在该算法内部建立了两个相同网络结构的策略，通过约束相似度在可信的范围内，保证了策略具有比原始算法更高的探索性外，还能够通过相互学习提升策略的性能。

使用 SCDPD-PPO 算法训练时，内部的两个策略在 HalfCheetah-v2 任务上的学习曲线如图 7 所示。在 400 万时间步前，SCDPD-PPO 算法两个策略保持着较为分散的状态，即两个策略均拥有强探索性，此时策略与环境交互获得的经验数据有积极的示教作用，因此学习曲线在这一阶段急剧上升。强探索性的作用不仅表现在训练前期的快速优化上，而且相比表 2 中的数据，在 400 万时间步结束后，SCDPD-PPO 算法中的两个策略性能值已经超过 PPO 算法和 DPD-PPO 算法经过 1000 万时间步所训练的策略的性能。在 500 万时间步之后，两个策略的学习曲线呈现震荡式的缓慢上升状态。出现这种情况的原因是，SCDPD-PPO 算法控制两个策略的相似度值缓慢降低，使得两个策略的分布逐渐趋于一致，虽然牺牲了一部分探索性，但是能够保证任意的一个策略在受到劣势经验数据的影响时，可以被相似度目标函数所约束，降低其造成的影响。

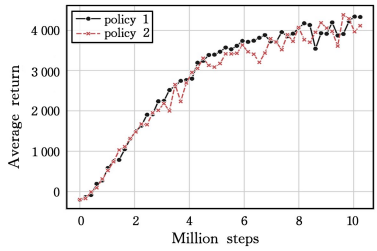


图 7 SCDPD-PPO 算法中双策略学习曲线

Fig. 7 Learning curves of dual policies in SCDPD-PPO

随机策略是采用高斯分布的形式实现的，其输入是环境中的当前状态，输出是以多维高斯分布所描述的动作。以任务 HalfCheetah-v2 为例，固定一个环境状态 s，并将其作为 SCDPD-PPO 算法中两个策略的输入，在输出的动作中，选取同一维度的动作值信息，并绘制其高斯分布。

图 8(a)－图 8(d)分别为策略与环境交互 250 万、500 万、750 万和 1000 万时间步时，两个策略输出的相应维度的动作值高斯分布。图中的横坐标表示动作值，纵坐标表示相应动作被策略选取的概率密度。

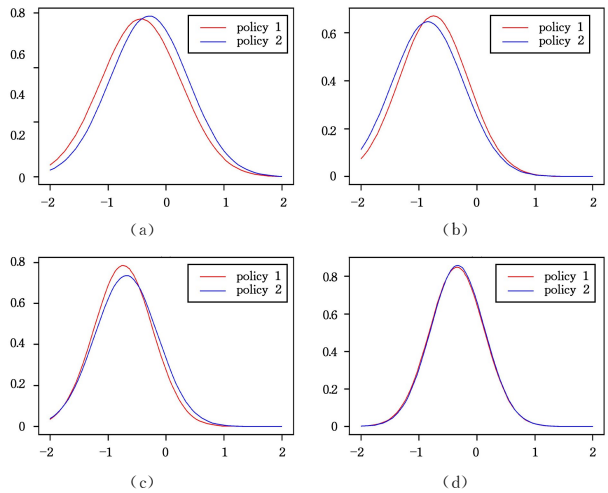


图 8 SCDPD-PPO 算法中双策略输出相应维度的动作值分布

Fig. 8 Corresponding dimension action distribution of dual policies in SCDPD-PPO

从图中可以直观地看出，随着训练进程的增加，对于固定的环境状态 s，高斯分布均值处的动作的概率密度不断提高，即策略能够根据状态输出相对确定的动作，间接地证明了随着对环境认知的深入，策略在被不断地提升。当处于 250 万时间步时，曲线相对分散，策略 1 和策略 2 差异较大，即每个策略都能以相对独立的方式搜索动作，从而增大了收集到优势动作数据的可能性。随着与环境交互的时间步增加，曲线渐渐靠近，策略 1 和策略 2 之间的差异逐渐减小，算法将重心从探索性向稳定性偏移，当到达1 000 万步时，策略 1 和策略 2 输出的动作在相应维度的高斯分布曲线接近重合，两个策略无限趋于一致。

将 SCDPD 分别与异策略和同策略深度强化学习算法结合的 SCDPD-SAC 算法和 SCDPD-PPO 算法，在 4 个连续控制任务中进行对比，验证了本文提出的框架既适用于异策略深度强化学习，也适用于同策略深度强化学习，能够有效地提升双策略框架中策略的探索性能，且加强了算法在训练后期的稳定性。

结束语 传统的策略蒸馏方法采用的是师生策略模型，而获得教师策略是相对昂贵的，基于多学生策略模型的策略蒸馏方法是最优的替代方案，学生策略以不同的视角独立地进行探索，以知识迁移的方式相互协作，从而优化策略提升的效果。当某一个学生策略始终是最优的，或者所有学生策略趋于一致的情况发生时，基于多学生策略框架的方法则会退化为单一策略的深度强化学习算法。因此，本文给出了学生策略间相似度的概念，并且提出了 SCDPD。该框架在两个相同网络结构的学生策略更新的过程中约束学生策略的相似度，避免其他学生策略趋于最优的学生策略，从而提升学生策略的探索性以及训练过程的稳定性。多个连续控制任务实验结果表明，结合本文提出的框架和经典的深度强化学习算法的 SCDPD-SAC 算法和 SCDPD-PPO 算法，具有更优秀的性能表现。

本文采用 KL 散度描述学生策略间的相似度，然而 KL 散度仅能够有效衡量随机策略，对于确定性策略，无法客观描述其相似度，下一步工作重点是将 SCDPD 融入到深度确定策略梯度方法中。

参 考 文 献

[1] SUTTON R S, BARTO A G. Reinforcement learning: An introduction [M]. Massachusetts: MIT press, 2018.

[2] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of go without human knowledge [J]. Nature, 2017, 550(7676): 354-359.

[3] KOBER J, BAGNELL J A, PETERS J. Reinforcement learning in robotics: A survey [J]. The International Journal of Robotics Research, 2013, 32(11): 1238-1274.

[4] SALLAB A E, ABDOU M, PEROT E, et al. Deep reinforcement learning framework for autonomous driving [J]. Electronic Imaging, 2017, 2017(19): 70-76.

[5] LIU Q, ZHAI J W, ZHANG Z Z, et al. A survey on deep reinforcement learning [J]. Chinese Journal of Computers, 2018, 41(1): 1-27.

[6] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deepreinforcement learning [J]. Nature, 2015, 518(7540): 529-533.

[7] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]// International Conference on Machine Learning. 2016: 1928-1937.

[8] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[C]// ICLR. 2016.

[9] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]// International Conference on Machine Learning. 2018: 1587-1596.

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

[11] SCHULMAN J, MORITZ P, LEVINE S, et al. High-dimensional continuous control using generalized advantage estimation [J]. arXiv: 1506.02438, 2015.

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

[13] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor[C]// International Conference on Machine Learning. 2018: 1861-1870.

[14] SALIMANS T, HO J, CHEN X, et al. Evolution strategies as a scalable alternative to reinforcement learning[J]. arXiv: 1703.03864, 2017.

[15] TAO Y, GENC S, CHUNG J, et al. REPAINT: Knowledge Transfer in Deep Reinforcement Learning[C]// International Conference on Machine Learning. 2021: 10141-10152.

[16] BARRETO A, BORSA D, QUAN J, et al. Transfer in deep reinforcement learning using successor features and generalised policy improvement [C]// International Conference on Machine Learning. 2018: 501-510.

[17] CZARNECKI W M, PASCANU R, OSINDERO S, et al. Distilling policy distillation[C]// International Conference on Artificial Intelligence and Statistics. 2019: 1331-1340.

[18] LAI KH, ZHA D, LI Y, et al. Dual Policy Distillation[C]// International Joint Conference on Artificial Intelligence. 2020: 3146-3152.

[19] RUSU A A, COLMENAREJO S G, GULCEHRE C, et al. Policy distillation[J]. arXiv: 1151.06295, 2015.

[20] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. arXiv: 1503.02531, 2015.

[21] WADHWANIA S, KIM DK, OMIDSHAFIEI S, et al. Policy distillation and value matching in multiagent reinforcement learning [C]// International Conference on Intelligent Robots and Systems. 2019: 8193-8200.

[22] CHEN G. A New Framework for Multi-Agent Reinforcement

Learning-Centralized Training and Exploration with Decentralized Execution via Policy Distillation[C]// International Conference on Autonomous Agents and MultiAgent Systems. 2020: 1801-1803.

[23] ZHA D,LAI K H,ZHOU K,et al. Experience replay optimization[C]// International Joint Conference on Artificial Intelligence. 2019:4243-4249.

[24] XU T,LIU Q,ZHAO L,et al. Learning to explore via meta-policy gradient[C]// International Conference on Machine Learning. 2018:5463-5472.

[25] FANG Y,REN K,LIU W,et al. Universal Trading for Order Execution with Oracle Policy Distillation[J]. arXiv: 2103.10860,2021.

[26] FAN S,ZHANG X,SONG Z. Reinforced knowledge distillation: Multi-class imbalanced classifier based on policy gradient reinforcement learning [J]. Neurocomputing,2021,463:422-436.

[27] HA J S,PARK Y J,CHAE H J,et al. Distilling a hierarchical policy for planning and control via representation and reinforcement learning[C]// IEEE International Conference on Robotics and Automation. 2021:4459-4466.

[28] LI Z H,YU Y,CHEN Y,et al. Neural-to-Tree Policy Distillation with Policy Improvement Criterion[J]. arXiv: 2108.06898,2021.

[29] ZHAO C,HOSPEDALES T. Robust domain randomised reinforcement learning through peer-to-peer distillation[C]// Asian Conference on Machine Learning. 2021:1237-1252.

[30] CHA H,PARK J,KIM H,et al. Proxy experience replay:Federated distillation for distributed reinforcement learning [J]. IEEE Intelligent Systems,2020,35(4):94-101.

[31] SUN H,PAN X,DAI B,et al. Evolutionary Stochastic Policy Distillation[J]. arXiv:2004.12909,2020.

[32] BROCKMAN G,CHEUNG V,PETTERSSON L,et al. Openai gym[J]. arXiv:1606.01540,2016.

XU Ping'an, born in 1997, postgraduate. His main research interests include reinforcement learning and deep reinforcement learning.

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

(责任编辑:喻藜)