

基于信念投影与最大熵的强化学习算法

杜志斌, 于学冰

(华南师范大学人工智能学院, 广东 佛山 528225)

摘要: 在传统强化学习环境中, 下一个状态的信息和当前状态的奖励只依赖于当前状态信息和智能体的动作, 但在许多环境中存在下一个状态的信息不仅依赖于当前状态和动作, 还依赖于一段时间步前的状态和动作, 这种延迟反馈的特点使传统的强化学习算法难以训练。鉴于此, 提出一种结合信念投影与最大熵的强化学习算法, 信念投影模块用于处理算法在有延迟反馈环境中所面临的状态信息和奖励问题, 使智能体可以在有延迟反馈特性的环境中得以有效训练。运用最大熵原理使智能体进行更广泛的搜索, 获得探索能力以保证其在复杂环境中搜索更多最优策略以避免陷入局部最优。在OpenAI Gym中有延迟反馈特性的环境中开展实验测试, 结果表明, 相比传统的强化学习算法, 该算法在延迟反馈环境中的探索能力和性能均有显著提升。

关键词: 深度强化学习; 马尔可夫决策过程; 延迟反馈; 双延迟深度确定策略梯度算法; 最大熵强化学习; 信念投影

DOI: 10.11907/rjdk.251068

扫描二维码阅读全文:



中图分类号: TP181

文献标识码: A

文章编号: 1672-7800(2026)005-0043-07

Reinforcement Learning Algorithm Based on Belief Projection and Maximum Entropy

DU Zhibin, YU Xuebing

(School of Artificial Intelligence, South China Normal University, Foshan 528225, China)

Abstract: In traditional reinforcement learning environments, the information of the next state and the reward of the current state depend only on the previous state and the actions of the agent. However, in many environments, the information of the next state depends not only on the current state and actions, but also on the state and actions of a period of time ago. This delayed feedback characteristic makes it difficult to train traditional reinforcement learning algorithms. This article proposes a reinforcement learning algorithm that combines belief projection and maximum entropy. The belief projection module is used to handle the state information and reward problems faced by the algorithm in environments with delayed feedback, enabling agents to effectively train in environments with delayed feedback characteristics. By applying the principle of maximum entropy, intelligent agents are able to conduct more extensive searches and acquire exploration capabilities to ensure that they can search for more optimal strategies in complex environments and avoid falling into local optima. The experimental tests were conducted in several environments with delayed feedback in OpenAI Gym, and the results showed that compared to traditional reinforcement learning algorithms, algorithm has significantly improved exploration ability and performance in delayed feedback environments.

Key Words: deep reinforcement learning; Markov decision process; delayed feedback; twin delayed deep deterministic policy gradient algorithm; maximum entropy reinforcement learning; belief projection

0 引言

强化学习(Reinforcement Learning, RL)是解决顺序决策问题的一种有效方法, 智能体通过与环境的交互来学习策略^[1]。与其他方法不同, 智能体无需依赖对环境动态的

预先了解即可解决问题。成功的强化学习系统可以利用试错获得的经验最大化外部训练信号, 即奖励信号, 以训练出正确的策略。

随着深度学习的发展, 强化学习与深度学习的结合形成了深度强化学习(Deep Reinforcement Learning, DRL)^[2-3]。深度强化学习可以从原始数据中提取出抽象

收稿日期: 2025-03-25

基金项目: 国家自然科学基金项目(11701505); 广东省自然科学基金面上项目(2023A1515011472)

作者简介: 杜志斌(1985-), 男, 博士, 华南师范大学人工智能学院教授、硕士生导师, 研究方向为人工智能; 于学冰(1998-), 男, 华南师范大学人工智能学院硕士研究生, 研究方向为深度强化学习。本文通讯作者: 杜志斌。

特征,以深度Q网络(Deep Q-network, DQN)和Rainbow为例,算法直接从高维图像中提取关键信息,在Atari 2600等经典游戏中展现出了超越人类的表现^[4]。近年来,深度强化学习应用于许多方面,它不仅在围棋这一复杂的智力游戏中击败了世界顶级选手,还在DOTA 2^[5]和星际争霸II^[6]等协作型3D多人电子游戏中展现了与人类相似的实力。

深度强化学习的应用在现实世界中也越来越广泛,比如自动驾驶系统^[7]、机器人^[8-10]和对话系统^[11]等。但是,搭建现实世界的强化学习应用程序具有挑战性,主要原因是强化学习算法在从有限的环境交互中高效学习并生成有效策略方面存在显著瓶颈。此外,真实世界的强化学习系统必须具有严格限制,以避免损坏设备并能够支持安全的探索活动。这些系统还必须处理实时需求,实现状态和动作的同步变动,这与典型深度强化学习的训练方式不同。

在强化学习领域,延迟反馈环境是指智能体在采取动作之后,需要经过若干延迟,才能获得奖励和状态的反馈^[12]。这是因为在实际生活中受限于物理条件,诸多方面都存在延迟反馈现象。具体而言,智能体在执行某一动作后,并不能立即得到反馈,而是需要经过一段时间才能获得反馈信息。在延迟反馈环境下,智能体每次执行动作后,其对于后续结果的影响是进行决策时一个重要考量因素。智能体执行每一个动作,其效果都可能需要经过长时间的延迟才有可能体现出来^[13-15]。传统强化学习算法依赖于即时反馈信息,以帮助智能体从反馈结果中调整策略,一旦存在延迟反馈现象,智能体的探索能力和性能会出现较差表现。

为了解决传统的深度强化学习算法在延迟反馈环境中训练智能体时面临的挑战和智能体训练时探索能力不足的情况,本文从基于信念投影的强化学习和最大熵强化学习框架中获得了启发。本文利用带有临时缓冲区的信念投影,结合最大熵强化学习框架,使新算法能够有效地处理延迟反馈环境,利用最大熵强化学习原理以增强智能体的探索能力。本文主要贡献如下:①研究了深度强化学习算法在探索能力和处理延迟反馈任务环境时的的问题;②提出了一种结合信念投影和最大熵的强化学习算法,解决了深度强化学习算法探索能力不足和在延迟反馈环境中无法有效训练的问题;③在OpenAI的Gym任务上,将本文算法与多种基线算法进行对比,实验表明,在延迟反馈环境中,本文算法的性能和探索能力优于传统的强化学习算法。

1 相关工作

1.1 马尔可夫决策过程

马尔可夫决策过程(Markov Decision Process, MDP)是一个描述智能体与其环境之间相互作用的数学模型,由五

元组 (S, A, P, R, γ) 组成^[16]。其中, S 表示状态空间,包含环境的所有可能状态; A 是动作空间,表示智能体在每个状态下可以采取的动作; $P(s'|s, a)$ 是转移概率,描述在状态 s 中采取动作 a 后转移到状态 s' 的概率; $R(s, a)$ 是奖励函数,给出了智能体在状态 s 中采取行动 a 时获得的奖励; γ 是折扣因子,用于表示未来奖励的重要性,通常取值在 $[0, 1]$ 区间。

在马尔可夫决策过程中,目标是找到一个策略 π ,这是每个状态下动作的概率分布,使智能体最大化其长期回报。为了实现这一点,使用状态值函数 $V^\pi(s)$ 和动作值函数 $Q^\pi(s, a)$ 评估预期回报。前者代表从状态 s 开始并遵循策略 π 的预期回报,而后者代表在状态 s 中采取行动 a 的预期回报。

贝尔曼方程描述了当前状态与未来状态之间的联系,它由即时奖励和未来奖励的折扣总和构成。贝尔曼方程为计算这些值函数提供了递归关系。状态价值函数 $V^\pi(s)$ 使用递归公式计算,如式(1)所示。

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} [R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')] \quad (1)$$

动作价值函数 $Q^\pi(s, a)$ 使用类似的递归关系计算,如式(2)所示。

$$Q^\pi(s, a) = \mathbb{E}_{s' \sim p(\cdot|s, a)} [R(s, a) + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s', a')] \quad (2)$$

最后,智能体连续优化这些值函数以找到最优策略 π^* ,并最大化长期反馈。

固定延迟马尔可夫决策过程(Constant-delay MDP, CDMDP)被定义为一个六元组 (S, A, P, R, γ, k) ^[17]。其中, k 是一个非负整数,表示智能体进入状态和接收环境反馈之间的时间步数,环境反馈包括状态信息和奖励。当 $k = 0$ 时,CDMDP变为常规MDP;其余情况下,智能体仅在采取前 k 个动作后才观察其初始状态。

1.2 双延迟深度确定策略梯度算法

在基于值的强化学习中,函数近似误差会导致值估计过高,从而使智能体生成次优策略。鉴于DDPG算法对超参数和各种调整的敏感性,该问题将会影响行动者—批评者(Actor-Critic)框架。Fujimoto等^[18]提出双延迟深度确定策略梯度算法(Twin Delayed Deep Deterministic policy gradient algorithm, TD3),该算法引入了减少这些误差对行动者—评论家影响的机制。TD3使用基于双Q学习(Double Q-learning)的方法,通过取两个评论家值中的最小值以限制高估。TD3还研究了目标网络和高估偏差之间的联系,使用延迟策略更新以减少每次更新的错误并提升智能体的整体性能。

截断双Q学习是指TD3算法学习两个Q函数。TD3通过最小化均方误差,同时训练两个Q函数(Q_{ϕ_1}, Q_{ϕ_2}),并共享目标。两个Q函数中值较小的一个用作Q目标,如式(3)所示。

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\phi_i, \text{target}}(s', a(s')) \quad (3)$$

TD3 采取延迟策略更新,即训练行动者和评论家网络时,在没有目标网络的情况下,可能引发训练不稳定性。然而,仅当行动者的网络是固定时,评论家网络才收敛于正确的值。TD3 算法以低频率更新行动者网络,以高频率更新评论家网络。TD3 引入了平滑机制,通过给目标动作增加噪声,平滑 Q 值对动作的函数,让策略更难以利用 Q 函数误差,这就是目标策略平滑。目标策略平滑作用原理如式(4)所示。

$$a(s') = \text{clip}(\mu_{\theta, \text{target}}(s') + \text{clip}(\epsilon, -c, c), a_{\text{low}} + a_{\text{high}}) \quad (4)$$

ϵ 本质上是从正态分布中采样的噪声,目标策略平滑是一种正则化方法。

1.3 最大熵强化学习

在最大熵强化学习框架中,行动者的目标是同时最大化预期回报和熵,这意味着它的目标是在尽可能随机的情况下完成任务。通过引入最大熵,在最大化期望奖励的同时,智能体的探索能力得到显著增强,其策略在面对环境扰动时也表现出更高的鲁棒性^[19]。基于最大熵强化学习框架,已经出现了几种算法,如 Soft Q-Learning 和 SAC (Soft Actor Critic)^[20]。熵是随机变量随机性的度量,具体而言,如果 X 是一个具有概率密度函数 p 的随机变量,其熵定义为如式(5)所示。

$$H(X) = \mathbb{E}_{x \sim p}[-\log p(x)] \quad (5)$$

在强化学习中,可以使用熵表示给定状态 s 下策略 π 的随机性,表示为 $H(\pi(\cdot|s))$ 。最大熵强化学习的思想不仅是最大化累积奖励,也是鼓励智能体在策略选取的过程中增大随机性。最大熵强化学习在强化学习目标中添加熵的正则化项,定义为如式(6)所示。

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_t r(s_t, a_t) + \alpha H(\pi(\cdot|s_t)) \right] \quad (6)$$

其中, α 是控制熵重要性的正则化系数。熵正则化提高了强化学习算法的探索水平,正则化系数 α 越大,探索性越强,这有助于加速后续的策略学习,并降低策略陷入局部最优的可能性。

1.4 基于信念投影的强化学习

为了解决传统强化学习算法在进行训练时遇到的状态空间爆炸问题, Kim 等^[21]提出基于信念投影的 Q 学习 (Belief Projection Based Q-learning, BPQL)。BPQL 通过评估输入状态大小等于原始状态空间而不是增广状态空间大小的临界值以缓解状态空间爆炸问题。其中,增广状态 $\bar{s}_t$ 是在具有 d 个延迟反馈时间步的环境中,通过连接最后一个观测到的状态和之前的 d 个动作进行构建,如式(7)所示。

$$\bar{s}_t = (s_{t-d}, a_{t-d}, a_{t-d+1}, \dots, a_{t-1}) \quad (7)$$

在 BPQL 算法中,具有增广状态的扩展转换元组存储在经验回放池 $\mathcal{D}$ 中。为了构建增广状态, BPQL 使用临时缓冲区 $\mathcal{B}$, 该缓冲区存储原始状态和动作的历史。智能体使用从临时缓冲区获得的增广状态作为策略的输入以选取动作。

2 本文算法

2.1 问题分析

传统强化学习算法的策略探索性有限。在训练的前期阶段,智能体采取的策略较为保守,导致其很容易过早收敛,尤其是在状态空间庞大或者非常复杂的问题中,很难完全探索状态空间,即使采用 epsilon 贪婪或加入噪声加以探索,也极易陷入局部最优。

延迟反馈是强化学习中较为普遍的问题。许多机器人控制或博弈等应用中智能体动作与环境反馈之间具有一定的时延,这使得智能体难以基于实时的反馈对策略进行及时调优,对学习过程不利。传统强化学习算法是基于实时反馈对策略进行更新,在延迟反馈环境下算法很难准确度量延迟的影响,从而在稳定性、收敛速度等方面存在诸多问题。在复杂的延迟反馈环境中,由于反馈信息不完备,智能体很容易选择效率较低或次优的动作,最终无法收敛于恰当的策略。

为了解决传统强化学习算法在策略探索和处理延迟反馈任务方面的不足,本文提出了基于信念投影和最大熵的强化学习算法 (Reinforcement Learning Algorithm Based on Belief Projection and Maximum Entropy, BPME)。将改进的 TD3 算法与最大熵原理相结合,并引入基于信念投影的方法,在网络结构、策略更新方式、智能体动作评价体系和网络损失函数方面加以创新改进,增强了算法的探索能力,并使其在具有延迟反馈的环境中可进行有效学习。

2.2 算法设计

传统强化学习算法缺乏足够的探索,本文采用最大熵强化学习框架以增强强化学习算法探索能力。相比传统强化学习,它在目标函数中增加了一个熵项,即增加该项后,激励智能体除最大化累积奖励外,还要保持其采取的策略具有更高的随机性与多样化的优点。加入熵项后的策略具有更强的探索性,有助于避免其陷入局部最优,使得智能体在具有较大复杂度和动态更新的环境中更加灵活,同时对不断探索出更优策略非常有效。

由于智能体的最优策略在不同状态下可能大相径庭,故有必要动态调整熵系数以满足不同策略的搜索需求。当智能体对于某些状态下的最优动作不确定性较高时,熵系数更大,有利于更多探索,避免局限于目前的策略空间内;反之,当智能体确定一些状态下的最优动作时,熵系数应更小,将更多精力投入到当前最优动作上。因此,本文采用了一种动态熵调节功能,可根据环境和策略不确定性情况自动调整熵系数,在探索和利用上取得平衡。引入熵项后,策略的损失函数可以表示为如式(8)所示。

$$L_{\pi}(\phi) = \mathbb{E}_{(s, \bar{s}) \sim \mathcal{D}, a \sim \pi_{\phi}} [\alpha \log(\pi_{\phi}(a|\bar{s})) - \min_{i=1,2} Q_{\theta_i}(s, a)] \quad (8)$$

其中, α 表示熵系数,控制策略熵对损失的影响。 $\bar{s}$ 表示增广状态,通过将最近观察到的状态与之前的动作连接

起来而构建。在基于增广状态计算值后,在策略改进阶段更新策略,以增加所有增广状态的值。通过优化这一目标,智能体可以在策略更新中不断增强其探索能力,自动熵调整确保了不同任务和状态之间的最佳探索利用平衡,从而提高了学习效率和整体性能。这种动态熵调整机制不仅提高了探索效率,还提高了智能体在不确定环境中的适应性。

为了使算法在具有延迟反馈的环境中加以有效训练,本文对行动者网络结构进行了多项调整。在TD3算法的行动者网络中,只输入当前状态信息,无法捕捉延迟反馈环境中的有效特征信息。而在BPME算法中,将输入维度进行扩展,引入历史动作序列信息,使得行动者网络能够捕捉延迟反馈环境中的特征信息,从而更好地推断延迟状态下的动作关联性。并且,在BPME的行动者网络中,输出层包含了均值和标准差的信息,使其支持随机策略的生成,并结合最大熵原理增强探索能力,引入自动熵以平衡探索和利用。

在评论家网络的训练过程中,本文采用了目标网络的方法对函数近似误差加以控制,使训练过程更加稳定。目标网络采用延迟更新目标值网络以减少策略更新造成的高估,增强了算法面对复杂环境的鲁棒性。具体而言,在策略网络选取动作时,采用目标策略网络得到当前状态的下一个动作,使动作选择过程不受当前策略网络更新影响,保证了估计的稳定性。

在重放经验缓冲区中采样增广下一个状态并反馈至目标策略网络中,得到目标动作。将目标动作和增广状态信息反馈至两个目标评论家网络中计算Q值。该过程中,使用双Q学习方法,从两个目标批评网络中选择Q值较小值,以减少高估带来的偏差;使用熵项对最小Q值进行权重加权,进一步权衡探索利用。

将当前动作的即时奖励加上获得的最小Q值以计算目标Q值。以目标Q值更新评论家网络,同时计算评论家网络的损失函数,计算公式如式(9)所示。

$$L_{critic}(\theta) = \mathbb{E}_{(s, s', s'', a, r) \sim \mathcal{D}} \left[\left(r + \gamma \left[\min_{i=1,2} Q_{\theta_i}(s', a') - \alpha \log \pi_{\phi'}(a' | s') \right] - Q_{\theta_i}(s, a) \right)^2 \right] \quad (9)$$

其中, γ 是折扣因子, a' 由 $\pi_{\phi'}(s')$ 得出。通过该损失函数,评论家网络可以有效地更新其参数,使Q值估计更加准确,减少了估计误差引起的训练不稳定性。该机制确保智能体可以在具有延迟反馈的环境中有效学习,加速向最优策略收敛。

BPME算法架构如图1所示。在训练过程中,智能体收到来自临时经验回放池给出的增广状态 $\bar{s}_t$ 后,便根据自身的策略选择动作 a_t 输入到具有延迟反馈特性的环境中。有别于常规环境反馈的信息,此时具有延迟反馈特性的环境会反馈出 d 个时间步之前的状态信息和奖励信号,当时间步 t 达到设定的步数时,便会从临时经验回放池取出一

个元组给经验回放池,再将序列信息反馈给行动者和评论家网络,通过各自的策略优化后,再使用软更新的方式更新目标网络的结构。

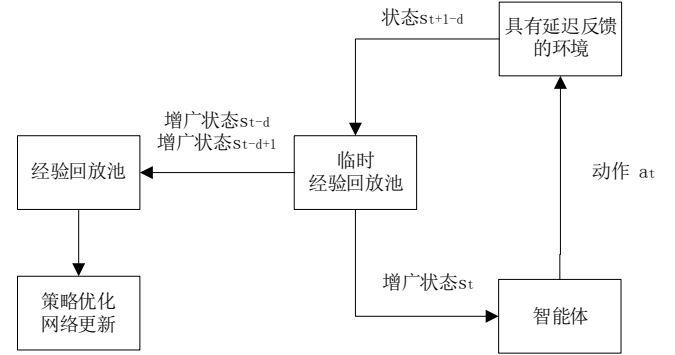


Fig. 1 BPME architecture

图1 BPME架构

BPME算法的伪代码如算法1所示。

算法1 BPME算法

输入:环境当前状态

输出:智能体根据策略所选取的动作

1. 初始化行动者网络 π_ϕ ,评论家网络 $Q_{\theta_1}, Q_{\theta_2}$,以及相应的随机参数 ϕ, θ_1, θ_2
2. 初始化目标网络 $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2, \phi' \leftarrow \phi$
3. 初始化经验回放池 $\mathcal{D}$,临时经验回放池 $\mathcal{B}$
4. 初始化延迟时间步 d ,软更新频率 τ ,以及延迟策略更新值 f
5. FOR 每个时间步 t DO
6. IF $t < d$ THEN
7. 随机选取动作 a_t 并执行
8. 将 a_t 存入 $\mathcal{B}$
9. ELSE IF $t == d$ THEN
10. 随机选取动作 a_d 并执行,得到观测状态 s_1
11. 将 s_1 和 a_d 存入 $\mathcal{B}$
12. ELSE
13. 从 $\mathcal{B}$ 中采样得到增广状态 $\bar{s}_t$
14. 通过策略 π_ϕ 选取动作 a_t 并执行,得到观测状态 s_{t+1-d}
15. 将 s_{t+1-d} 和 a_t 存入 $\mathcal{B}$
16. IF $t > 2d$ THEN
17. 从 $\mathcal{B}$ 中采样得到增广状态 $\bar{s}_{t-d}, \bar{s}_{t-d+1}$
18. 将 $\bar{s}_{t-d}, \bar{s}_{t-d+1}$ 存入 $\mathcal{D}$
19. END
20. END
21. 从 $\mathcal{D}$ 中采样得到 $\bar{s}, \bar{s}', s, s', a, r$
22. 用式(8)更新评论家网络
23. IF $t \bmod f$ THEN
24. 用式(9)更新行动者网络
25. 更新目标网络:
26. $\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i$
27. $\phi' \leftarrow \tau \phi + (1 - \tau) \phi'$
28. END
29. END

3 实验分析

3.1 实验环境

MuJoCo 是 OpenAI 提供的一个高性能物理引擎,已普遍应用于强化学习环境。它能够模拟多关节和复杂接触的物理动力学系统,是模拟机器人操控、机械臂、生物力学系统研究的较佳选择。MuJoCo 为复杂场景的实时模拟提供了高效计算、精确建模,主要包含刚体动力学、碰撞检测、模拟摩擦等多个功能模块,并且支持复杂的物理系统模型。在强化学习领域,OpenAI Gym 的强化学习环境通

常会结合使用 MuJoCo 以训练算法或者验证算法,MuJoCo 环境也有内置一些标准任务,例如 HalfCheetah、Humanoid 等。这些任务可以模拟多种难度的运动控制、运动平衡和运动追踪等,很好地搭建了测试算法平台。目前,MuJoCo 被作为强化学习的重要工具之一,可实现非常成熟的环境交互。本文在 OpenAI Gym 的 6 个 MuJoCo 任务环境中对该算法进行了测试,详细任务环境如图 2 所示。

6 个实验环境的状态空间维度和动作空间维度具体参数如表 1 所示。其中,Humanoid 环境最为复杂,其状态空间和动作空间的维度远多于其他实验环境。

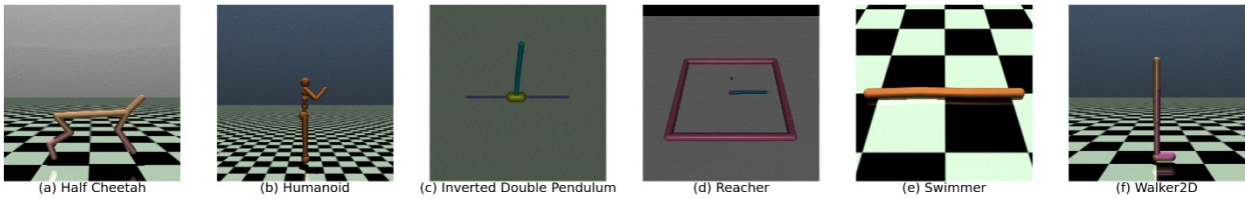


Fig. 2 MuJoCo environment

图 2 MuJoCo 环境

Table 1 Specific parameters of the experimental environment

表 1 实验环境具体参数

环境	状态空间	动作空间
HalfCheetah-v3	17	6
Humanoid-v2	376	17
InvertedDoublePendulum-v2	11	1
Reacher-v2	11	2
Swimmer-v3	8	2
Walker2d-v3	17	6

本文算法网络都是在 PyTorch 框架下构建,并在具有 RTX 3060 Ti 的服务器上完成实验。实验中所用的具体硬件环境如表 2 所示,软件环境如表 3 所示。

Table 2 The hardware environment of the experiment

表 2 实验硬件环境

硬件	配置
CPU	12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz
显卡	NVIDIA GeForce RTX 3060 Ti
内存	32GB

Table 3 The software environment of the experiment

表 3 实验软件环境

软件	配置
操作系统	Ubuntu 18.04.6 LTS
框架	PyTorch 1.12.1

3.2 性能对比

为了验证 BPME 算法在延迟反馈环境中的有效性,在 OpenAI Gym 的 6 个具有延迟反馈的 MuJoCo 任务环境中开展实验,并将 BPME 与两种基线算法——TD3 和 SD3(Softmax Deep Double Deterministic Policy Gradients)算法进行了性能对比^[22]。实验结果表明,BPME 在所有任务中均表现出显著的性能提升。

图 3 展示了 BPME、TD3 和 SD3 在 6 个具有延迟反馈的

MuJoCo 任务环境中的训练曲线,其中延迟反馈时间步长 $d = 6$ 。可以看出,在训练过程中,BPME 在大多数任务中的表现均优于 TD3 和 SD3,尤其是在训练初期,BPME 能够更快地捕捉任务中的关键策略,避免了陷入局部最优,验证了 BPME 可以有效地处理具有延迟反馈的任务。例如,在 HalfCheetah-v3 任务中,BPME 在训练初期即表现出明显的性能优势,最终得分显著高于 TD3 和 SD3。类似地,在 Swimmer-v3 任务中,BPME 的得分曲线始终位于 TD3 和 SD3 之上,表明其在任务中的探索能力和策略优化能力更强。

表 4 显示了当环境中的延迟反馈时间步长 $d = 6$ 时,BPME、TD3 和 SD3 3 种算法在 6 个任务中获得的最高分数及其标准差。本文进一步比较了每种算法在不同任务中的最终性能。可以看出,在所有任务中,BPME 的最高得分均高于 TD3 和 SD3,特别是在 HalfCheetah-v3 和 Humanoid-v2 任务中,BPME 的得分值分别比 TD3 高出 3 870.7 和 219.2,比 SD3 高出 3 870.6 和 232.7。这一结果表明,BPME 在处理延迟反馈任务时具有更强的探索能力和性能。

3.3 消融实验

在延迟反馈环境中,BPME 算法相比 TD3 最明显的特点就是引入了基于信念投影的强化学习思想,使之适合在延迟反馈环境下使用。在不同的延迟反馈时间步长(d)下,增广状态、临时缓冲区和回放经验池会有所不同。本文在 HalfChetah-v3 环境中进行了具有不同延迟反馈时间步长的实验,重点分析了不同延迟反馈时间步长对算法性能的影响。主要方法是:固定 BPME 算法中的最大熵模块,仅调整信念投影模块的延迟参数,通过训练过程中的平均得分及其标准差进行评估。通过图 4 可以看出,BPME 的性能随着不同的延迟反馈时间步长而变化。延迟

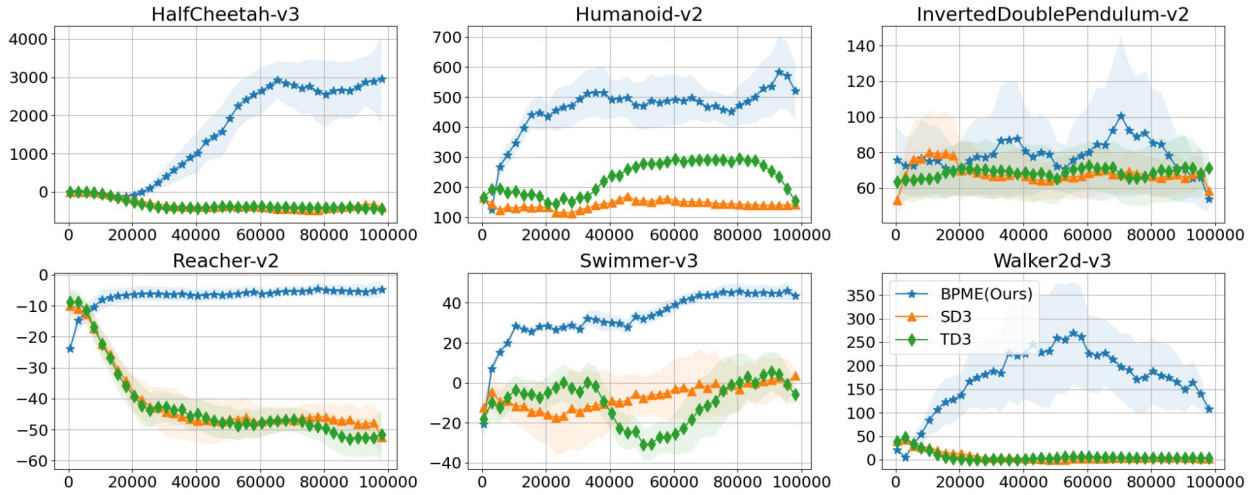


Fig. 3 Performance comparison of BPME, TD3, and SD3

图3 BPME、TD3和SD3性能对比

Table 4 Score data for BPME, TD3, and SD3 in 6 delayed feedback environment tasks

表4 6个延迟反馈环境任务中BPME、TD3和SD3的得分数据

环境 ($d=6$)	TD3	SD3	BPME
HalfCheetah-v3	-0.025 ± 12.2	0.11 ± 32.5	3870.7 ± 172.7
Humanoid-v2	449.6 ± 43.1	436.1 ± 29.2	668.8 ± 311.3
InvertedDoublePendulum-v2	89.9 ± 33.2	104.7 ± 46.9	115.7 ± 84.1
Reacher-v2	-5.26 ± 1.87	-9.31 ± 3.18	-3.98 ± 1.72
Swimmer-v3	34.8 ± 9.4	24.5 ± 16.6	55.6 ± 2.9
Walker2d-v3	282.9 ± 111.3	265.8 ± 59.8	383.9 ± 197.2

反馈时间步长越小,性能越好,当 $d=2$ 和 $d=3$ 时,算法在训练初期便快速获取奖励,最终获得较高得分;随着延迟反馈时间步长的增加,最终得分依次递减,表明延迟反馈步长更大的环境会导致算法性能降低。

为了探讨最大熵模块对于BPME算法的影响,在HalfCheetah-v3环境中进行了两个不同延迟反馈时间步($d=14, d=18$)的实验,在这两个实验中,去除了BPME算法中的最大熵模块。从图5可以看出,在两个不同延迟反馈时间步的环境中,具有最大熵模块的BPME算法比没有最大熵模块的BPME算法在任务环境探索和算法整体性能上具有显著优势。

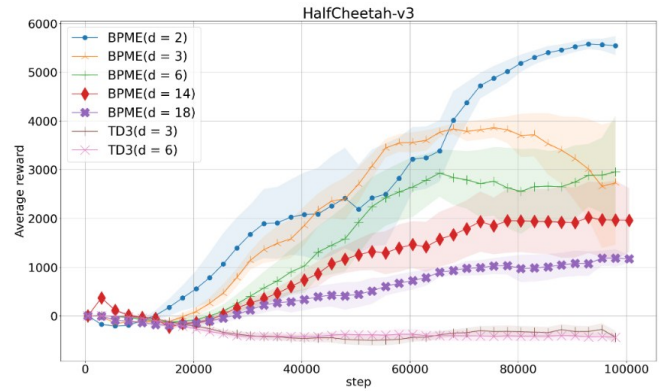


Fig. 4 Performance of BPME in environments with different delayed feedback time steps

图4 BPME在不同延迟反馈时间步长环境中的性能

4 结语

为了解决传统强化学习算法在探索能力不足和在延迟反馈环境中性能不佳的问题,本文基于改进的TD3,引入了最大熵原理以增强智能体的探索能力,并采用基于信念投影的强化学习方法,通过增广状态解决延迟反馈环境

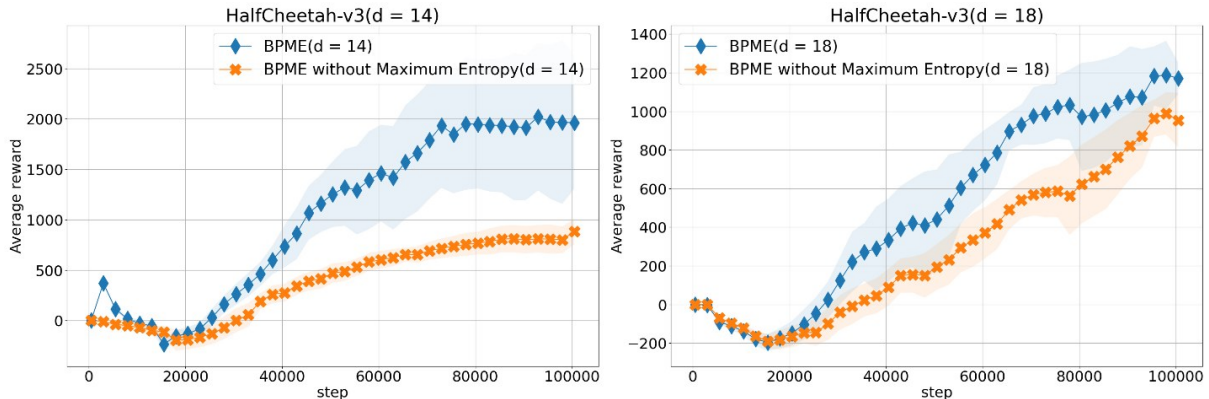


Fig. 5 Ablation experiment of BPME algorithm on maximum entropy module

图5 BPME算法关于最大熵模块的消融实验

中的任务。在OpenAI Gym的MuJoCo任务环境下的测试表明,本文算法在处理具有延迟反馈环境任务时可以进行有效训练,性能相较于基线算法具有明显提升。虽然BPME算法可以在具有延迟反馈的环境中有效地进行训练,但当面对反馈时间步长延迟较大的环境时,其性能会下降。如何在信息延迟反馈较大的环境中有效地训练并提高性能是未来工作中亟待解决的问题。

参考文献:

- [1] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. Cambridge: MIT press, 2018.
- [2] VOLODYMYR M, KAVUKCUOGLU K, SILVER D, et al. Playing ATARI with deep reinforcement learning[DB/OL]. <https://arxiv.org/abs/1312.5602>, 2013.
- [3] VOLODYMYR M, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518: 529–533.
- [4] HESSEL M, MODAYIL J, VAN HASSELT H, et al. Rainbow: combining improvements in deep reinforcement learning[C]//Vancouver: The AAAI Conference on Artificial Intelligence, 2022.
- [5] BERNER C, BROCKMAN G, CHAN B, et al. Dota 2 with large scale deep reinforcement learning[DB/OL]. <https://arxiv.org/abs/1912.06680>, 2019.
- [6] VINIYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 2019, 575: 350–354.
- [7] JIN Y L, FAN B R, GAO Y, et al. Autonomous driving algorithm based on meta-learning and reinforcement learning[J]. Journal of Applied Sciences, 2024, 42(5): 795–809.
金彦亮, 范宝荣, 高源, 等. 基于元学习和强化学习的自动驾驶算法[J]. 应用科学学报, 2024, 42(5): 795–809.
- [8] KURNIAWATI H. Partially observable Markov decision processes and robotics[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2022, 5(1): 253–277.
- [9] LAURI M, HSU D, PAJARINEN J. Partially observable Markov decision processes in robotics: a survey[J]. IEEE Transactions on Robotics, 2022, 39(1): 21–40.
- [10] GAO Y N, WANG A C, ZHAO H K, et al. A review on visual navigation methods based on deep reinforcement learning[J]. Computer Engineering and Applications, 2025, 61(10): 66–78.
高宇宁, 王安成, 赵华凯, 等. 基于深度强化学习的视觉导航方法综述[J]. 计算机工程与应用, 2025, 61(10): 66–78.
- [11] XU K, WANG Z Y, WANG X, et al. A survey of task-oriented dialogue policies based on reinforcement learning[J]. Chinese Journal of Computers, 2024, 47(6): 1201–1231.
徐恺, 王振宇, 王旭, 等. 基于强化学习的任务型对话策略研究综述[J]. 计算机学报, 2024, 47(6): 1201–1231.
- [12] DERMAN E, DALAL G, MANNOR S. Acting in delayed environments with non-stationary Markov policies [DB/OL]. <https://arxiv.org/abs/2101.11992>, 2023.
- [13] KARAMZADE A, KIM K, KALSI M, et al. Reinforcement learning from delayed observations via world models [DB/OL]. <https://arxiv.org/abs/2403.12309>, 2024.
- [14] BOUTEILLER Y, RAMSTEDT S, BELTRAME G, et al. Reinforcement learning with random delays[C]//Addis Ababa: The Eighth International Conference on Learning Representations, 2020.
- [15] CHEN B, XU M, LI L, et al. Delay-aware model-based reinforcement learning for continuous control [J]. Neurocomputing, 2021, 450: 119–128.
- [16] OTTERLO M V, WIERING M. Reinforcement learning and Markov decision processes[M]. Heidelberg: Springer, 2012.
- [17] WALSH T J, NOURI A, LI L, et al. Learning and planning in environments with delayed feedback[J]. Autonomous Agents and Multi-Agent Systems, 2009, 18: 83–105.
- [18] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]//Proceedings of International Conference on Machine Learning, 2018: 1587–1596.
- [19] WU L, LIU Q, HUANG Z G, et al. Maximum entropy hierarchical reinforcement learning with advantage-weighted mutual information maximization[J]. Chinese Journal of Computers, 2023, 46(10): 2066–2083.
乌兰, 刘全, 黄志刚, 等. 优势加权互信息最大化的最大熵分层强化学习[J]. 计算机学报, 2023, 46(10): 2066–2083.
- [20] HAARNOJA T, ZHOU A, HARTIKAINEN K, et al. Soft actor-critic algorithms and applications [DB/OL]. <https://arxiv.org/abs/1812.05905>, 2018.
- [21] KIM J, KIM H, KANG J, et al. Belief projection-based reinforcement learning for environments with delayed feedback[J]. Advances in Neural Information Processing Systems, 2023, 36: 678–696.
- [22] PAN L, CAI Q, HUANG L. Softmax deep double deterministic policy gradients[J]. Advances in Neural Information Processing Systems, 2020, 33: 11767–11777.

(责任编辑: 孙 娟)