软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(11):4217–4238 [doi: 10.13328/j.cnki.jos.006391]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于样本效率优化的深度强化学习方法综述*

张峻伟[1,2], 吕 帅[1,2], 张正昊[1,2], 于佳玉[1,3], 龚晓宇[1,2]

[1](符号计算与知识工程教育部重点实验室(吉林大学), 吉林 长春 130012)

[2](吉林大学 计算机科学与技术学院, 吉林 长春 130012)

[3](吉林大学 软件学院, 吉林 长春 130012)

通信作者: 吕帅, E-mail: lus@jlu.edu.cn

摘 要: 深度强化学习将深度学习的表示能力和强化学习的决策能力结合, 因在复杂控制任务中效果显著而掀起研究热潮. 以是否用 Bellman 方程为基准, 将无模型深度强化学习方法分为 *Q* 值函数方法和策略梯度方法, 并从模型构建方式、优化历程和方法评估等方面对两类方法分别进行了介绍. 针对深度强化学习方法中样本效率低的问题进行讨论, 根据两类方法的模型特性, 说明了 *Q* 值函数方法过高估计问题和策略梯度方法采样无偏性约束分别是两类方法样本效率受限的主要原因. 从增强探索效率和提高样本利用率两个角度, 根据近年来的研究热点和趋势归纳出各类可行的优化方法, 分析相关方法的优势和仍存在的问题, 并对比其适用范围和优化效果. 最后提出增强样本效率优化方法的通用性、探究两类方法间优化机制的迁移和提高理论完备性作为未来的研究方向.

关键词: 深度强化学习; *Q* 值函数方法; 策略梯度方法; 样本效率; 探索与利用

中图法分类号: TP18

中文引用格式: 张峻伟, 吕帅, 张正昊, 于佳玉, 龚晓宇. 基于样本效率优化的深度强化学习方法综述. 软件学报, 2022, 33(11): 4217–4238. http://www.jos.org.cn/1000-9825/6391.htm

英文引用格式: Zhang JW, Lü S, Zhang ZH, Yu JY, Gong XY. Survey on Deep Reinforcement Learning Methods Based on Sample Efficiency Optimization. Ruan Jian Xue Bao/Journal of Software, 2022, 33(11): 4217–4238 (in Chinese). http://www.jos.org.cn/1000-9825/6391.htm

Survey on Deep Reinforcement Learning Methods Based on Sample Efficiency Optimization

ZHANG Jun-Wei[1,2], LÜ Shuai[1,2], ZHANG Zheng-Hao[1,2], YU Jia-Yu[1,3], GONG Xiao-Yu[1,2]

[1](Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun 130012, China)

[2](College of Computer Science and Technology, Jilin University, Changchun 130012, China)

[3](College of Software, Jilin University, Changchun 130012, China)

Abstract: Deep reinforcement learning combines the representation ability of deep learning with the decision-making ability of reinforcement learning, which has induced great research interest due to its remarkable effect in complex control tasks. This study classifies the model-free deep reinforcement learning methods into *Q*-value function methods and policy gradient methods by considering whether the Bellman equation is used, and introduces the two kinds of methods from the aspects of model structure, optimization process, and evaluation, respectively. Toward the low sample efficiency problem in deep reinforcement learning, this study illustrates that the over-estimation problem in *Q*-value function methods and the unbiased sampling constraint in policy gradient methods are the main factors that affect the sample efficiency according to model structure. Then, from the perspectives of enhancing the exploration efficiency and improving the sample exploitation rate, this study summarizes various feasible optimization methods according to the recent research hotspots and trends, analyzes advantages together with existing problems of related methods, and compares them according to the scope of application and optimization effect. Finally, this study proposes to enhance the generality of optimization methods, explore migration

* 基金项目: 国家重点研发计划(2017YFB1003103); 国家自然科学基金(61300049); 吉林省自然科学基金(20180101053JC)
收稿时间: 2020-11-11; 修改时间: 2020-12-19, 2021-01-18; 采用时间: 2021-06-02; jos 在线出版时间: 2021-08-03

of optimization mechanisms between the two kinds of methods, and improve theoretical completeness as future research directions.

Key words: deep reinforcement learning; *Q*-value function method; policy gradient method; sample efficiency; exploration and exploitation

强化学习(reinforcement learning, RL)与监督学习和无监督学习并称为机器学习的 3 类方法, 主要目的是收集和学习通过控制智能体与环境交互得到的观测数据, 并求得给定决策优化任务的最优策略.

20 世纪 60 年代, 强化学习已用于最优控制领域, 主要应用于状态与动作空间较小的任务中, 如井字棋游戏、赌博机问题、倒立摆问题等[1]. 起初, 强化学习算法用表格来记录模型信息, 并可以在理论上证明求解最优化问题的可行性. 但强化学习问题的求解难度会因为环境复杂度的增加而显著增大, 受限于时间与空间复杂度而难以完成复杂的训练任务[2].

随着 21 世纪人工智能的兴起, 深度学习(deep learning, DL)以巧妙的神经网络构建方式, 展现出强大的函数表示能力, 为强化学习的算法研究与实际应用带来了重大突破. 深度强化学习(deep reinforcement learning, DRL)通过使用深度神经网络作为函数逼近器, 将深度神经网络中高效的表示与感知能力和强化学习中的决策能力相结合, 使智能体可以在与复杂环境的交互过程中不断优化策略[3]. 自此, 深度强化学习迅速成为人工智能领域的焦点.

近年来, 深度强化学习技术已经在决策控制、自动驾驶、自然语言处理、计算机视觉、推荐与检索系统、金融等诸多领域都发挥了不可替代的作用. 其中包括但不限于: AlphaGo 击败世界围棋冠军李世乭和柯洁[4], 深度强化学习在街机游戏 Atari 2600 上获得了远超人类专家的得分表现[5], OpenAI 和腾讯公司分别在 MOBA 游戏中完成智能体训练并战胜人类顶尖职业选手[6], 强化学习框架已能高效地完成城市自动驾驶任务[7], 利用强化学习在文本分类中优化表示结构[8], 在目标导向对话中利用强化学习完成主题分割和标记[9], 利用强化学习选择图像修复的工具链[10], 通过强化学习生成视频摘要[11], 利用强化学习挖掘不同搜索结果的内在联系[12], 在金融交易中利用强化学习预测各类资产未来的发展趋势[13]. DeepMind 公司中负责 AlphaGo 项目的研究员 David Silver 声称: “GAI=RL+DL”, 认为结合了深度学习表示能力与强化学习推理能力的深度强化学习将会是人工智能的终极答案.

在深度强化学习展现其巨大应用前景的同时, 人们发现, 现阶段的成果重心主要集中在游戏和模拟任务中. 要想把深度强化学习的成果从虚拟的游戏或仿真的环境转换到现实应用方向, 往往还要面临任务难度更高、回报变化更加复杂、对智能体鲁棒性要求更高以及采样成本消耗过大等多方面的考验. 对此, 也衍生出对于强化学习中稀疏奖励问题[14,15]、环境奖励噪声问题[16,17]、现实环境中摩擦力与重力变化问题[18,19]以及增强智能体训练效率问题[20,21]等多方面的研究, 以至于目前深度强化学习研究方向繁多, 由于侧重点的不同也发展出更加多样的算法.

根据强化学习模型的构建方式不同, 可以将现有的无模型深度强化学习算法分为两大类: *Q* 值函数方法和策略梯度方法. 值函数方法希望通过 Bellman 方程优化的方式, 在全局范围中寻找贪婪策略下最优累计回报的函数值, 虽然部分状态或动作在最优策略下不会被访问到, 但也要计算对应累计回报值[3,22,23]. 而策略梯度方法将直接以当前的动作策略为优化目标, 试图直接计算当前策略下期望回报的梯度值, 进而将当前策略不断朝着回报增加的方向进行优化[24−26]. 所以, 两类方法虽然适用的领域一致, 但在模型架构、实现方式、对于不同任务的效果、动作策略以及算法的优化方法等方面上都有显著的差别.

本文从现有深度强化学习算法广泛研究的增强探索效率、经验高效利用以及减少采样数量等方向进行归纳, 并统一为强化学习算法的样本效率问题. 以如何使用更少的环境交互采样次数获得更高的回合回报为目标, 将现有优化归纳为提高环境的采样效率和提高已有样本的利用率两个角度, 结合 *Q* 值函数方法和策略梯度方法, 分别进行算法与改进策略的分析和比较, 并根据具有不同特点的主流实验环境说明各类改进算法的优劣, 最后给出结论并对未来研究方向进行展望.

1 强化学习问题模型

强化学习作为实现人工智能的重要技术之一, 以最优化策略为目标,需要控制智能体与环境交互, 并通过不断试错的方式对当前策略进行优化. 由于最优化问题有着广泛的适用性和应用范围, 所以现实中大部分任务都能构建为强化学习问题模型.

1.1 强化学习系统

在使用强化学习解决决策优化类问题时, 需要将任务系统划分为环境与智能体两部分, 并定义二者间的数据交互, 完成强化学习系统框架的构建, 以便应用强化学习算法. 强化学习系统框架通常如图 1 所示.

图 1 强化学习系统框架

定义 1(强化学习系统). 强化学习系统将决策优化问题划分为环境和智能体两个部分, 要求智能体可以观测当前环境状态, 根据环境状态给出所要执行的动作, 并可以接收环境的回报值和观测环境变化情况.

在强化学习系统中, 智能体包含可以观察环境状态的感知器、可以影响环境变化的动作执行器和可以进行动作选择的动作策略. 其中, 动作策略是强化学习训练的目标, 强化学习算法通过某种方式对动作策略进行建模, 针对每个感知器采集到的环境状态数据, 根据当前的动作策略模型在候选动作集中选出将执行的动作, 并利用动作执行器影响环境后, 观测到环境的回报值和下一时刻的状态数据. 所以, 环境即智能体的任务域, 与真实世界对应, 环境的状态变化情况可以被智能体观测, 任务环境在建模时还要加入回报函数来衡量目标任务的完成程度, 即环境内部拥有执行动作后的状态变化规则和回报规则.

对于强化学习任务, 智能体不会被告知应该采取哪个动作, 而需要通过执行动作后所产生的回报值来优化动作策略. 智能体必须在试错中学习, 不断尝试不同的动作以获取数据, 以达到回报值最大的目标. 所以, 试错是强化学习的重要特征之一[1]. 同时, 多数强化学习任务中的动作不仅可以影响到当前时刻的回报, 还将影响到后续某些时刻的回报, 即强化学习方法往往还需要解决延迟回报的问题[27]. 此外, 由于强化学习多用于游戏任务和物理仿真模拟任务, 所以也可以将回报称为奖励.

1.2 Markov决策过程

在强化学习中, 通常使用 Markov 决策过程(Markov decision process, MDP)来表示强化学习系统. 作为定义强化学习系统的数学模型, MDP 可以在具有 Markov 性的环境中, 模拟动作选择后的回报值反馈和状态转移过程[2]. 求解 MDP 问题即寻求回报值最大的最优动作策略.

定义 2(Markov 决策过程). 强化学习系统中的任务环境可以建模为 Markov 决策过程. Markov 决策过程由五元组$\langle S,A,P,R,\gamma\rangle$表示, 其中,

- S 为环境中所有状态的集合.
- A 为智能体可执行动作的集合.
- P 为状态转移矩阵, 其中, $P_{ss'}^{a}=P(S_{t+1}=s'\mid S_t=s,A_t=a)$.
- R 为回报函数, r_t 表示在 t 时刻采取动作 a_t 后所获得的回报.
- γ为折扣因子, 用于累计回报的计算, 其中, $\gamma\in[0,1$.

定义 3(智能体的策略表示). 在强化学习中, 智能体根据动作策略选取将要执行的动作, 并试图最优化动作策略. 动作策略即给定观测状态下的动作选择概率分布, 用$\pi(a|s)$表示, 其中, $\pi(a|s)=P(A_t=a|S_t=s)$.

强化学习以回报值最大为目标，所以在训练过程中，通常将累计回报值定义为 G_t，如公式(1)所示.

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{1}$$

其中，折扣因子γ作为平衡立即回报和延迟回报的权重参数，用以减少延迟回报的不确定性影响以及避免出现无限时长的任务中累计回报值无穷的问题. 经过在众多任务环境上实践检验可知，γ接近 1 时训练效果最好，大多设置在 0.97–0.99 之间.

1.3 深度强化学习的任务环境

在 20 世纪 60 年代，经典强化学习算法主要应用于小规模任务环境. 随着将深度神经网络的强大函数表示能力与强化学习的决策能力结合形成深度强化学习方法后，所能解决任务的复杂度也显著提高. 随着任务难度的不断增大，也衍生出多个不同种类的实验任务环境. 本文根据任务动作和奖励特性的不同，归纳出深度强化学习方向广泛研究的 3 类任务.

- 离散动作任务，即动作空间离散且可选择动作个数有限的决策任务.

在对应的 MDP 模型中，离散动作空间集合 A 表示为$\{a_i|1\leqslant i\leqslant n\}$，动作策略在每次与环境交互过程中，在动作空间中选取合适的动作 a_i. 离散动作任务主要求解完成某个任务目标的最优序贯决策问题. 目前，评估深度强化学习算法效果的主流离散动作任务集为 Atari 2600，部分任务如图 2 所示. Atari 2600 作为经典的街机游戏集合，目前拥有 60 个不同类别的离散动作游戏任务，可以从各方面评估算法的效果. 在任务中，以像素显示作为环境观测值，以摇杆与按键的多种组合操控方案作为动作值，有着动作空间较小但状态空间变化复杂的特点，需要智能体通过多次动作选择间的合理配合来获取更高的奖励值.

图 2 离散动作任务 Atari 2600 部分任务

- 连续动作任务，即动作空间的维度有限但动作值连续的决策任务.

在对应的 MDP 模型中，连续动作空间集合 A 表示为$\{(a_1,a_2,\ldots,a_n)|l_i\leqslant a_i\leqslant u_i\}$，其中，$u_i$ 和 l_i 分别表示动作 a_i 取值的上下界. 动作策略在每次与环境交互过程中给出动作向量$(a_1,a_2,\ldots,a_n)$，表示在每个动作维度上选择的动作值. 连续动作任务主要解决工程控制问题，评估深度强化学习算法效果的主流连续动作任务集为 GYM-MuJoCo，部分任务如图 3 所示. GYM-MuJoCo 即 OpenAI 公司基于物理仿真环境 MuJoCo 构建的一系列控制任务，不同任务在奖励规则、任务目标、任务类别和任务时长等方面加以区分，可以较为全面地评估算法效果. 在任务中，以感知器获取的环境物理量作为观测值，以各个控制器对关节施加力量的程度为动作值，有着动作空间大的特点，需要智能体以合理的方式表示动作，在兼顾动作精度和动作有效性的同时，给出合理的控制策略.

从动作选择的角度看，深度强化学习任务可能是离散动作、连续动作或同时具有这两种形式动作的任务. 由于不同强化学习方法在离散动作和连续动作任务上的效果差异较大，所以将两种任务分开讨论. 而从奖励值函数的角度看，无论动作是连续还是离散的，都有部分任务由于在大部分状态下无奖励值而导致训练难度极大，通常称这些任务为稀疏奖励任务. 深度强化学习方法在稀疏奖励任务中的训练效果很差，得分与随机策略几乎相当. 稀疏奖励任务往往能够反映出真实环境中回报函数难以确定的情况，且样本效率极低，具有重要的研究意义.

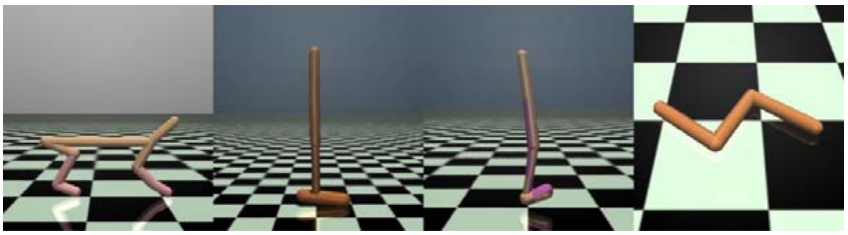


图 3 连续动作任务 GYM-MuJoCo 部分任务

- 稀疏奖励任务, 即大部分状态中的奖励值为 0, 几乎在所有状态下都不会获得环境的有效奖励反馈.

在对应的 MDP 模型中, 对于大多数状态-动作对(s,a), 奖励函数 $r(s,a)=0$. 游戏和物理仿真模拟环境由专家精心设计奖励, 但对于部分真实环境, 会因奖励设计困难而经常导致任务环境的奖励稀疏[28]. 稀疏奖励任务的动作可以是离散或连续的, 具有任务难度大、训练效率低的特点. 目前, 评估深度强化学习在稀疏奖励问题中效果的任务主要有“Super Mario Bros”、冒险类街机游戏和 3D 导航任务, 如图 4 所示. 这些任务只有到达目标点或完成任务目标后才会获得环境的奖励反馈, 需要智能体通过合理的策略, 在无环境奖励驱动的情况下进行合理交互采样, 通过具有多样性的样本数据进行有效的优化.

图 4 稀疏奖励任务

此外, 连续动作、离散动作和稀疏奖励任务环境虽然具有不同的性质, 但在合理的调整环境模型中动作划分方式和回报函数构建方式的情况下可以相互转化. 例如, 可以通过调整动作划分和回报函数, 在单个 MuJoCo 环境中同时构造出 3 类任务.

2 深度强化学习方法分类

强化学习更加接近生物学习的思维模式, 没有监督学习中繁重的标签数据需求, 比无监督学习有着更加广泛的应用范围, 同时也面临着更加复杂的模型构建与训练过程. 随着深度神经网络的应用, 为其注入了强大的函数表示能力, 改善了强化学习的训练过程, 显著提升了强化学习的训练效果.

可以解决深度强化学习任务的算法众多, 本文主要介绍无模型的深度强化学习算法, 即在进行模型训练的过程中不对环境进行模拟建模, 状态与回报数据均以直接与任务环境进行交互的方式获得.

在深度强化学习领域中, 通常将无模型的深度强化学习算法分为 Q 值函数方法和策略梯度方法, 在部分改进算法出现后, 分类标准逐渐模糊. 例如, DDPG 算法作为基于确定性动作策略并结合 Q 值函数的策略梯度算法, 被人们认为是 Q 值函数和策略梯度方法的结合[29]. 本文给出算法分类标准, 以是否使用 Bellman 方程为界限, 将算法分为 Q 值函数方法和策略梯度方法.

在本节中, 首先针对两类方法给出定义与模型构建过程, 随后阐述其在深度强化学习领域的主要发展历程与训练效果的衡量标准.

2.1 Q值函数方法

Q 值函数方法源于经典的强化学习方法, 在深度强化学习领域有着强大的问题求解能力和应用范围.

2.1.1 方法定义与模型构建

定义 4 (*Q* 值函数方法). *Q* 值函数方法以贪婪的动作选取机制, 通过 Bellman 方程的方式迭代优化, 求得 MDP 问题中最优策略下每个状态-动作对的累计回报值的函数, 每个状态下, *Q* 值最大的动作即选择的最优动作.

Q 值函数方法由动态规划思想衍生而来, 最初在 1988 年由 Watkins 等人提出[30]. *Q* 值函数即状态-动作值函数, 用 $Q^{\pi}(s_t,a_t)$表示, 可以衡量智能体以策略π选择动作时, 在当前状态 s_t 下执行动作 a_t 获得的累计回报值, 如公式(2)所示.

$$Q^{\pi}(s_t,a_t)=E_{s_{t+1}\in S,a_{t+1}\in A}(r_t+\gamma Q^{\pi}(s_{t+1},a_{t+1})) \tag{2}$$

根据 Bellman 方程优化过程, 可以由公式(3)迭代更新 Q^{π}.

$$Q^{\pi'}(s_t,a_t)=Q^{\pi}(s_t,a_t)+\alpha(r_t+\gamma\max_{a_{t+1}}Q^{\pi}(s_{t+1},a_{t+1})-Q^{\pi}(s_t,a_t)) \tag{3}$$

其中, $Q^{\pi'}$为单轮优化后的新策略π'对应的 *Q* 值函数. 随着更新过程的进行, Q^{π}将逐渐收敛于最优 *Q* 值函数 Q^*, 此过程的收敛性已经得到证明[30].

在深度强化学习中, 需要构建神经网络逼近器来表示 *Q* 值函数, 用 $Q_{\theta}^{\pi}(s,a)$ 表示, 以状态 *s* 和动作 *a* 为输入, θ为神经网络参数, 损失函数如公式(4)所示, 满足 Bellman 方程优化过程并符合神经网络可以对批量数据输入进行训练的特点, 其中 *D* 为任务环境的状态-动作域.

$$L(\theta)=E_{(s_t,a_t)\in D}(r_t+\gamma\max_{a_{t+1}}Q_{\theta}^{\pi}(s_{t+1},a_{t+1})-Q_{\theta}^{\pi}(s_t,a_t))^2 \tag{4}$$

在深度强化学习环境中, *Q* 值函数方法还需要引入经验池和目标网络. 经验池用来存储数据, 并完成训练采样时的数据复用, 可以降低从环境中采样经验间的相关性, 目标网络用 $\hat{Q}_{\theta^-}^{\pi}$ 表示, 用于计算公式(4)中 max*Q* 项, 达到延迟更新的目的, 可以有效避免优化目标随优化过程而变化, 提高训练的稳定性并缓解过高估计问题. 基于 *Q* 值函数的深度强化学习方法流程如算法 1 所示.

算法 1. 基于 *Q* 值函数的深度强化学习方法.

1: Initialize replay memory *D*, action-value function Q_{θ}, target action-value function $\hat{Q}_{\theta^-}$.

2: **For** episode=1,…,*M* **do**

3: **For** t=1,…,*T* **do**

4: Select action a_t by $\max_a Q_{\theta}(s_t)$ and some exploration strategies.

5: Execute action a_t in environment, observe reward r_t and new state s_{t+1}.

6: Store transition (s_t,a_t,r_t,s_{t+1}) in *D*.

7: Sample minibatch *B*of transitions (s_j,a_j,r_j,s_{j+1}) from *D*.

8: Perform a gradient descent step on $\frac{1}{n}\sum_{j=1}^{n}\left(r_j+\gamma\max_{a_{j+1}}\hat{Q}_{\theta^-}^{\pi}(s_{j+1},a_{j+1})-Q_{\theta}^{\pi}(s_j,a_j)\right)^2$ with respect to the network parameter θ.

9: Every *C* steps reset $\hat{Q}_{\theta^-}=Q_{\theta}$.

2.1.2 *Q* 值函数方法在深度强化学习中的发展

2015 年, Mnih 等人提出了 DQN (deep *Q*-learning)算法, 使用基于多层卷积层和全连接层建立的深度神经网络作为函数逼近器来实现基于 *Q* 值函数的深度强化学习算法, 以与人类相同的像素观察条件, 经过在 Atari 2600 的 49 个游戏中 200 M 帧的交互训练, 得到了超越人类专家得分水平的智能体[3].

在 DQN 算法提出后, 研究者在不改变原框架与基本思想的基础上进行了深入研究, 并提出了各类改进方案, 同样以 Atari 2600 为基准, 获得了更好的训练效果. Hasselt 等人分析了 DQN 算法中 *Q* 值函数逼近器的过高估计问题, 提出了 DDQN (double deep *Q*-learning)算法, 使用包含目标 *Q* 网络和当前 *Q* 网络的两个 *Q* 值函数, 有效地缓解了过高估计导致的策略向局部最优偏移的问题[22]. Schaul 等人发现: DQN 算法的经验池中, 只

有少部分的经验对于当前 Q 值函数训练有效, 并且经验的有效性会随着训练而变化. 他们提出了优先经验回放(prioritized experience replay, PER)机制, 通过时间差分误差衡量经验的重要性, 并以重要性来决定经验训练的采样优先级, 加快了训练速度并获得了更高的任务得分[31]. Wang 等人则认为: 在很多状态中, 无论采取哪种动作都不影响状态转换的结果, 计算状态价值函数要好于计算 Q 值函数. 他们提出了 Dueling 机制, 更改原有的输出层网络结构为状态价值输出和动作价值优势输出, 再将价值和优势值重新整合为 Q 值. 虽然无法用精准的数学理论说明新网络结构的作用, 但在实验中, 训练效果提升显著[32]. Fortunato 等人发现, 在动作空间中加入噪声的探索能力有限. 他们提出了 NoisyNet-DQN 算法, 在神经网络参数中加入噪声, 使用较小的额外计算成本获得了更优的训练效果[33]. Bellemare 等人指出: 环境回报值存在随机性, Q 函数只能模拟出价值分布的期望. 他们提出了 Categorical-DQN 算法, 构建出每个状态-动作对下的 Q 值分布情况, 减小了部分动作的精度损失和误差[34]. Hessel 等人提出了 Rainbow 算法, 将此前 DQN 的多种改进方案进行整合, 并应用多步学习策略, 在 Atari 2600 上得到了更高的任务得分[5].

近年来, 研究者大多以 Atari 2600 游戏上的得分情况来衡量基于 Q 值函数的深度强化学习算法的效果. 由于游戏间存在着种类和玩法上的不同, 所以在各游戏上得分的平均值也是对训练效率和泛化性的综合评定. 本文根据文献中的实验结果整理出已提到的 DQN 改进算法的效果, 见表 1.

表 1 基于 Q 值函数的深度强化算法在 Atari 2600 上的效果

Agent	Score (%)
Human[3]	100
DQN[3]	79
DDQN[22]	117
Noisy-DQN[33]	118
PER-DDQN[31]	140
Dueling-DDQN[32]	151
Categorical-DDQN[34]	178
Rainbow[5]	223

表 1 中, 得分以人类专家数据为基准(100%), 各类算法在 57 个游戏中进行 200 M 帧训练, 以最后 100 个回合内的得分均值作为对应游戏的最终得分, 并以 57 个游戏得分的中位数来评估算法的效果. 由于受到优化算法的基线影响, PER, Dueling 和 Categorical 机制基于 DDQN 实现, 与 DDQN 相比获得了更高的得分. Noisy 机制基于 DQN 实现, 与 DQN 相比有较大的提升. Rainbow 集合了 6 种改进策略, 获得了最高的得分. 由此可得: 每种改进策略均有良好的效果, 并且具有一定的可扩展性, 多种改进策略同时使用可以获得更好的效果.

DQN 算法要求输出层神经元个数需要与动作个数相等, 适用于 Atari 2600 等离散动作任务. 而对于连续动作任务, 需要将单维的连续动作分解为多个离散的控制动作进行, 同时还要受到维度灾难和控制不精准两方面问题的制约. Lillicrap 等人提出了适用于连续动作域的算法 DDPG (deep deterministic policy gradient, 深度确定性策略梯度), 引入动作选择网络, 并以最大化 Q 值为目标, 在连续动作任务中获得了良好的效果[35]. DDPG 算法虽然存在动作策略网络, 但与 Q 值函数的思想相同, 均使用贪婪的动作策略, 通过 Bellman 方程进行全局的 Q 函数值优化, 并需要构建经验池对历史样本进行存储利用, 动作策略网络的优化方向仅为使 Q 值函数最大化, 所以本文将 DDPG 算法归于 Q 值函数方法. 此外, DDPG 算法也存在着与 DQN 算法相同的 Q 函数过高估计问题, Fujimoto 等人分析过高估计问题后, 提出了 TD3 算法, 通过使用多个 Q 函数, 在引入少量偏差的情况下, 提出了有效的过高估计修正方案[23].

2.2 策略梯度方法

策略梯度方法作为一类深度强化学习方法, 使用概率分布表示动作策略, 弥补了 Q 值函数方法中策略确定性导致的不稳定问题, 因在连续动作任务上有着更好的训练效果而引起广泛关注.

2.2.1 方法定义与模型构建

定义 5(策略梯度方法). 策略梯度方法通常使用服从某种概率分布的随机策略, 通过直接计算当前随机策略在回报值上的梯度, 调整策略分布的参数, 使当前策略不断向梯度方向优化, 以求得回报值最大的最优动

作策略.

策略梯度方法由 Sutton 等人于 1999 年提出[36]. 考虑到最优的动作策略可能不是确定性策略, 而是需要在特定的概率下选择不同的行动, 策略梯度方法不再通过值函数来间接地进行动作选择, 而是直接地构建动作选择函数, 得到从观测状态到动作选择分布的直接映射.

在策略梯度方法中, π_θ表示动作策略分布函数, 其中, θ为动作策略的参数; $\tau=\{s_1,a_1,s_2,a_2,\ldots,s_T,a_T,s_{T+1}\}$表示从环境的初始状态 s_1 开始到结束状态 s_{T+1} 为止, 智能体与环境交互一个回合后得到的轨迹. 由于训练目标为回合内获得所有回报值之和最大, 所以当前策略能获得的回报值即为目标函数, 如公式(5)所示.

$$L(\theta)=E_{\tau\sim\pi_\theta}[R(\tau)]=\sum_{\tau\sim\pi_\theta}P_\theta(\tau)R(\tau) \tag{5}$$

其中, $\tau\sim\pi_\theta$表示智能体以策略π_θ与环境交互所能得到的所有轨迹, $R(\tau)$表示轨迹τ中所有回报值之和, $P_\theta(\tau)$表示当前轨迹发生的概率. 计算目标函数 $L(\theta)$的梯度值, 即得到当前策略梯度的值, 如公式(6)所示.

$$\nabla_\theta L(\theta)=E_{\tau\sim\pi_\theta}[R(\tau)\nabla_\theta\log P_\theta(\tau)] \tag{6}$$

所以, 策略梯度方法主要通过如下两步进行不断循环迭代更新.

1) 用π_θ与环境交互采样, 得到观测数据计算$\nabla_\theta L(\theta)$.

2) 用梯度值以α的学习率来更新θ, $\theta\leftarrow\theta+\alpha\nabla_\theta L(\theta)$.

由于τ需要以当前策略π_θ与环境交互进行采样, 所以在实际训练中无法精确地计算梯度值. 如何使用较少的采样次数获得低偏差、低方差的策略梯度估值, 成为决定策略梯度方法实际训练效果的重要因素.

在假设采样数量足够, 即不考虑采样消耗的情况下, 可以使用公式(7)计算策略梯度的估值.

$$\nabla_\theta L(\theta)=\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}R(\tau^n)\nabla_\theta\log P_\theta(a_t^n\mid s_t^n) \tag{7}$$

然而, 考虑到采样数量有限, 通常使用方差减小策略来降低估计方差的影响. 其中, 基于优势值的方差减小函数应用最为广泛, 如公式(8)所示.

$$\nabla_\theta L(\theta)=\frac{1}{N}\sum_{n=1}^{N}\left(\sum_{t=1}^{T}\left(\left(\sum_{t'=t}^{T}r_{t'}\right)-V_\varphi(s_t)\right)\nabla_\theta\log P_\theta(a_t^n\mid s_t^n)\right) \tag{8}$$

其中, $\left(\sum_{t'=t}^{T}r_{t'}\right)-V_\varphi(s_t)$ 通常用 $\hat{A}_t$ 来表示, 意为当前动作的优势值, 代表当前动作价值 $\sum_{t'=t}^{T}r_{t'}$ 与当前状态下平均动作价值 $V_\varphi(s_t)$之差. 在训练中, 需要构建神经网络来拟合状态价值函数 $V_\varphi(s_t)$, φ表示网络参数.

目前, 应用策略梯度方法大都以 Actor-Critic 的形式构建, 即需要构建并训练动作网络和价值网络, 如算法 2 所示. 动作策略网络$\pi_\theta(s_t)$被称为 Actor, 可以选择动作与环境交互; 状态价值网络 $V_\varphi(s_t)$被称为 Critic, 进行低方差的优势值计算, 通过 k 轮优化得出合理的策略梯度值.

算法 2. 基于策略梯度的深度强化学习方法.

1: Initialize policy function π_θ and value function V_φ.

2: **For** k=1,…,M **do**

3: Collect set of trajectories $D_k=\{\tau_i\}$ by running policy π_θ in the environment.

4: Compute cumulative reward G_t^i by r_t^i.

5: Compute advantage estimation $\hat{A}_t^i$ based on value function V_φ.

6: Perform a gradient ascent step on $P_\theta(a_t^i\mid s_t^i)\hat{A}_t^i$ with respect to the policy network parameter θ.

7: Perform a gradient descent step on $(G_t^i-V_\varphi(s_t^i))^2$ with respect to the value network parameter φ.

2.2.2 策略梯度方法在深度强化学习中的发展

在 DQN 算法引起广泛关注后, 研究者发现: 引入深度神经网络作为函数逼近器, 可以将 1999 年由 Sutton 提出的策略梯度方法应用于复杂的动作环境中, 尤其在连续动作环境中效果显著. Schulman 等人首次在策略梯度方法中加入深度神经网络, 说明了在策略梯度方法中, 函数估计方差是影响训练效果的关键因素, 并给出了基于优势值的低方差估计方式 GAE (generalized advantage estimation)[24]. 人们通常将 Schulman 提出的方

法作为深度强化学习中策略梯度方法的基准框架, 并称为 VPG (vanilla policy gradient)算法.

在 VPG 算法提出后, 研究者以 GYM-MuJoCo 环境为主, 分析训练过程中存在的问题, 在策略梯度方法领域提出了更加高效的算法. 其中, Schulman 等人发现 VPG 算法中稳定性不足, 提出了 TRPO (trust region policy optimization)算法, 通过在训练中加入约束条件, 限制每次更新的步长, 试图使每次更新都能得到更优的策略[25]. Mnih 等人同样针对稳定性问题提出了 A3C (asynchronous advantage actor-critic)算法, 通过在多个环境上异步且并行地运行多个智能体的方式, 使当前策略可以经历更多不同状态, 稳定训练过程的同时获得了更高得分[37]. 此后, Schulman 等人再次针对 TRPO 算法由于引入约束导致训练过程的复杂性, 给出了 PPO (proximal policy optimization)算法, 可以通过惩罚或裁剪的方式, 起到与约束条件同样的功效, 并使用重要性采样的方式提高梯度估值的准确性, 使算法实现更加简洁的同时, 获得更好的训练效果[26].

对于策略梯度方法, 在动作分布中进行采样完成动作选择, 而不是固定的贪婪动作, 更适合连续动作任务. 虽然在 Atari 2600 等离散任务环境也有着良好的训练效果, 但往往弱于 Q 值函数方法. 所以大多在 GYM-MuJoCo 模拟环境中选取 4–8 个任务, 利用整个训练过程中的得分变化情况来进行相关算法的比较. 其中, HalfCheetah, Hopper, Walker2d 和 Swimmer 这 4 个任务在时长、类型与难度上均有所不同, 可以在泛化性、稳定性、训练效率上对算法效果进行较为全面的衡量. 本文在 MuJoCo 上针对几种应用广泛的策略梯度算法进行了实验比较, 如图 5 所示.

图 5 中绘制了 A2C 算法(A3C 的同步版本, A3C 算法使用多 Actor 并行和异步更新的方法, 使其在复杂离散动作任务中训练的智能体达到人类水平. 但在连续动作任务中, 只使用多 Actor 并行而不用异步更新的 A2C 算法要优于 A3C 算法)、PPO 算法、TRPO 算法和 VPG 算法的训练效果. 各算法在每个任务下用不同的随机数种子运行 5 次, 图中实线表示 5 次实验的平均得分变化情况, 阴影部分表示训练得分的方差.

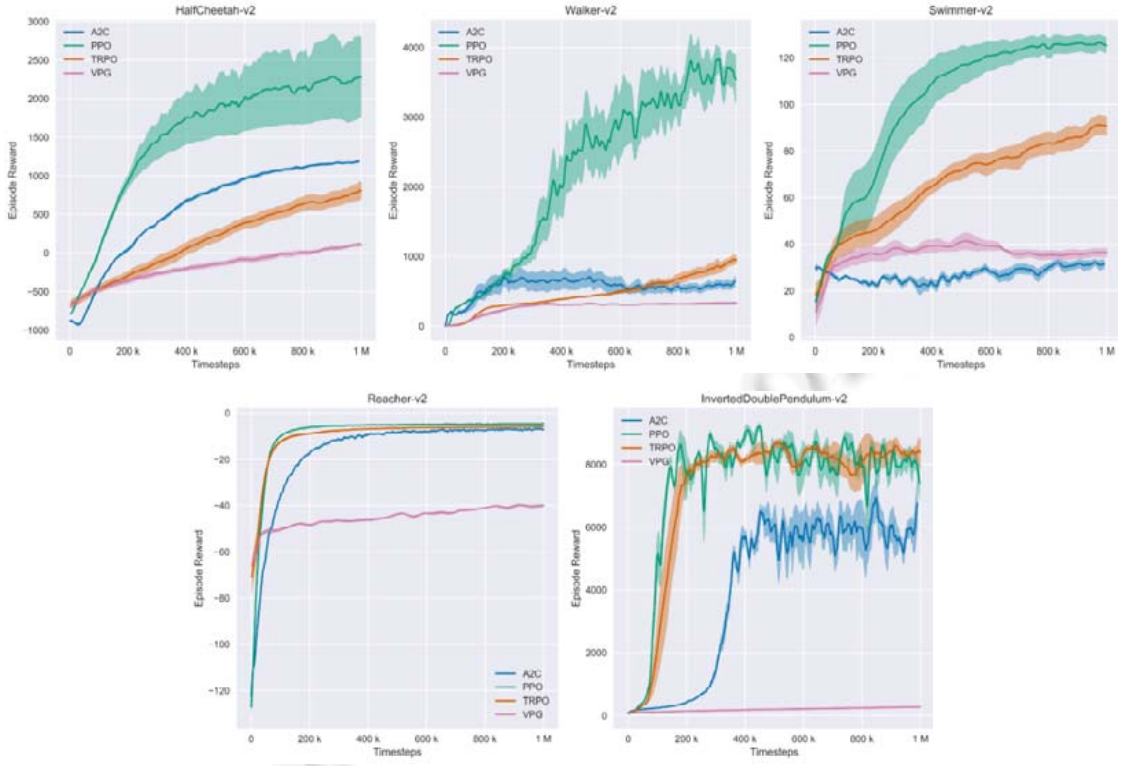


图 5 多种基于策略梯度的深度强化算法在 MuJoCo 上的效果

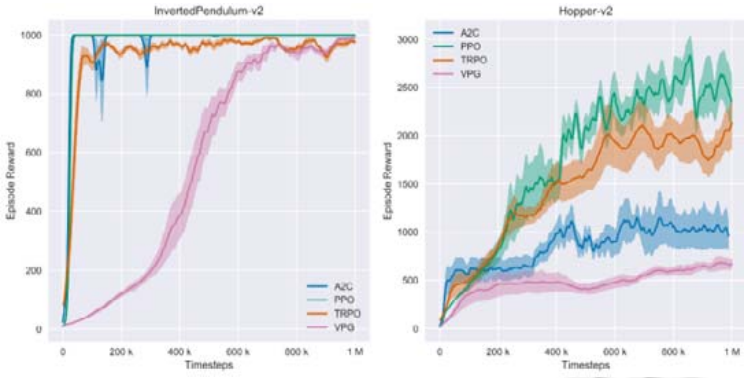


图 5 多种基于策略梯度的深度强化算法在 MuJoCo 上的效果(续)

由图中的曲线变化情况可以看出:

1) PPO 算法在 HalfCheetah, Walk2d 和 Swimmer 等环境中有着比其他算法更快的训练速度和更高的最终得分, 所以 PPO 算法在连续动作任务上有着更好的训练效率.
2) PPO 算法在所有的环境中都强于或接近于其他算法的训练速度和最终得分, 所以 PPO 算法能够适应各类连续动作任务环境, 有着较好的泛化性.
3) PPO 算法在 HalfCheetah 环境中方差最大, 说明 PPO 算法容易受到随机性的影响. 虽然拥有了更好的训练效果, 但还是存在着陷入局部最优解的问题.
4) A2C 算法和 TRPO 算法作为 VPG 算法的改进, 几乎能够在所有任务中有着比 VPG 算法更快的学习速度和更高的最终得分. 但 A2C 算法更适用于 HalfCheetah 环境, TRPO 算法则在 Walker2d 和 Hopper 任务中有着更好的效果.

此外, 由于模型、环境和参数复杂性的影响, 同种算法在固定的任务中, 使用了不同的网络结构、随机数种子、超参数设置和与算法非直接相关的优化(如奖励归一化、参数正则化、学习率衰减)都将影响最终的实验结果, 所以要将深度强化学习算法应用在现实任务中, 往往需要实际检验和细致的参数调优来选择最合适的算法[38].

3 深度强化学习方法的样本效率

深度强化学习已在街机游戏和物理模拟环境中取得显著成果, 但受到采样成本的影响, 难以向真实环境任务中迁移. 对于模拟环境, 采样次数仅与计算速度有关, 在训练过程中, 每秒可以完成几万次的环境交互过程; 而对于真实环境, 由于控制器与感知器物理情况的限制, 单位时间内能完成的采样次数急剧降低. 例如, DQN 算法在数小时内就能完成 200 M 帧的训练任务, 相当于 50 帧/秒的实际环境中 46 天的游戏时间. 同时, 在深度强化学习算法中, 训练效果受超参数影响较大. 为了获得更好的训练效果, 往往需要通过多次训练来进行超参数调优, 进一步提高了训练成本. 考虑到真实深度强化学习任务的采样成本问题, 本文给出样本效率的定义.

定义 6(样本效率). 在深度强化学习任务中, 样本效率即利用有限的交互次数, 通过合理的策略与环境交互进行采样, 并利用采样数据进行训练, 以获取策略优化的指标.

评估算法的样本效率, 既可以在使用相同交互次数的前提下衡量动作策略的训练效果, 也可以在到达相同训练效果的前提下衡量达到此效果所需的交互采样次数. 由于在强化学习算法中, 智能体以当前策略与环境交互进行采样, 并利用已得到的样本数据对策略进行优化, 探索过程与训练过程相互促进, 所以可以通过增强对环境的探索能力和提高对数据的利用能力两个方向提高样本效率.

1) 增强对环境的探索能力. 增强对环境的探索能力需要更有效地对环境中可能存在更优策略的动作

方向进行探索, 即对极少和未曾执行的动作进行选择, 或对极少和未曾访问过的状态进行访问. 探索能力更强的算法可以在与环境交互的过程中得到更有价值的样本数据, 能够有效地避免陷入局部最优值并有着更高的训练效率.

2) 提高对数据的利用能力. 提高对数据的利用能力需要在样本数据确定的情况下, 更有效地利用已有样本对动作策略进行训练, 得到更优的动作策略. 更优的动作策略也可以指导交互动作选择, 进而获得更有价值的样本数据, 加速学习进程.

本文将基于 Q 值函数和策略梯度的两类深度强化学习方法, 以增强探索能力和提高利用能力两个方向为主, 探究提高样本效率的方案.

4 通过增强探索机制提高样本效率

对于监督学习和无监督学习任务, 训练数据已知, 训练数据分布固定. 而在强化学习中, 需要智能体根据当前的动作策略网络以合适的方式选择动作, 并通过在环境中执行动作来获得采样数据. 因此在强化学习的样本数据中, 数据的分布将受到动作策略的影响.

在多数强化学习任务环境中, 可行解的空间范围极大, 难以遍历全部的状态-动作空间寻找最优解, 通常要求算法利用启发式搜索的思想, 能够基于当前的动作策略网络输出, 朝着可能存在最优解的方向进行探索, 并尽力保证探索的范围合适, 以免陷入局部最优解. 例如, 在 MuJoCo 环境的 Hopper 任务中, 智能体不仅需要保证模拟各关节的稳定性以防止由于摔倒导致的任务提前结束, 即状态空间的探索深度, 还需要保证尝试多种未知的动作以学习控制关节快速前进的能力, 即动作空间的探索广度.

4.1 经典探索策略

在强化学习任务中, 解空间内的任一可行解都可能是最优解, 所以要想保证算法在理论上能够使动作策略收敛于最优策略, 当且仅当探索策略具有能够访问到所有可行解的能力.

在 Q 值函数方法中, 通常使用ε-贪心和固定范围的高斯噪声作为探索策略进行动作探索. 其中, ε-贪心应用在离散动作任务中, 即在动作空间有限的情况下, 以ε的概率随机选择动作空间内任意动作, 并以 $1-\varepsilon$的概率, 基于贪婪的思想选取 Q 值最大的动作, DQN, DDQN 等算法均使用ε-贪心进行动作探索; 固定范围的高斯噪声通常应用在连续任务中, 即动作维度有限但动作域连续的情况下, 在动作策略网络输出动作 a 中加入均值为 0 的高斯噪声$\mathcal{N}(0,\sigma^2)$得到动作 a', 并以 a'与环境交互进行采样, 其中, σ^2 为超参数, 在 DDPG, TD3 等算法中通常固定为 0.05 或 0.1.

在策略梯度方法中, 以动作概率分布表示策略, 而非 Q 值函数方法中固定的贪婪策略, 所以动作策略中包含探索策略. 对于离散动作空间, 深度神经网络输出所有动作的选择概率, 经过 *softmax* 层处理后, 进行采样完成动作选择. 对于连续动作空间, 则同时输出动作选择的均值μ和方差σ^2, 并在高斯分布$\mathcal{N}(0,\sigma^2)$中采样完成动作选择. VPG, A3C, TRPO, PPO 等算法均以该方式进行动作选择.

ε-贪心和高斯分布均以一定概率可以选择到动作域内的任意动作. 当交互次数足够大时, 在环境的状态转移作用下, 可以访问到任务环境中所有状态-动作对, 能够保证理论上训练策略收敛于最优策略.

4.2 策略梯度方法中的探索能力增强机制

在 Atari 2600 的 60 个游戏任务中, 同一算法对于不同类型的任务训练效果差异较大. 例如, 在 Atlantis, Breakout, Video Pinball 等游戏中可以获得超越人类数十倍的得分, 但在 Montezuma's Revenge, Venture 等冒险类游戏中, 得分远低于人类水平. 所以通过观察 Atari 不同游戏的特性和分析智能体在不同游戏中的得分情况可知: 在任务环境具有密集的、引导性强的奖励时, 用深度强化学习算法可以获得显著的效果; 而在任务环境需要较长的正确动作决策序列才能获得奖励时, 却难以取得良好的训练效果. 即以随机概率分布进行探索的算法, 在密集奖励环境中训练效果显著, 但在稀疏奖励环境中难以进行有效训练[3]. 同时, 在真实世界的任务中, 可以确定的合理奖励通常是稀疏的, 人们难以获取足够多的专家经验来为任务设计合理的密集奖励. 在

奖励不合理的情况下，智能体训练得到的优化策略无法满足真实任务需求[39].

对于稀疏奖励任务，智能体需要经过一系列正确的动作决策后才能获得奖励，而在获得足够奖励前，无法进行有效的训练优化. 智能体在通过随机概率分布进行探索的同时，还需利用现有数据增强探索能力.

通常，人们将任务环境反馈的奖励值称为外在奖励，智能体以最大化外在奖励为训练目标，外在奖励也是评估动作策略优劣的标准. 对于策略梯度方法，在外在奖励的基础上增加内在奖励，是增强探索能力效果提升最显著的方案，得到用于训练的混合奖励如公式(9)所示.

$$r_t = r_t^e + r_t^i \tag{9}$$

其中，r_t^e 表示 t 时刻的外在奖励，r_t^i 表示 t 时刻的内在奖励，并将混合奖励 r_t 用于智能体的训练. 内在奖励 r_t^i 中包含着动作探索的方向信息，在环境奖励稀疏时或在训练的初期，内在奖励发挥主要作用，并引导智能体进行高效探索；在训练的后期，内在奖励趋向于 0，主要通过外在奖励寻找最优策略. 由于环境观察数据有限且要同时考虑内在奖励计算的时间和空间复杂度，所以需要构建较为简单的模型来近似计算内在奖励. 如何构建内在奖励，使其能够包含有效的探索方向信息，成为增强策略梯度方法中探索能力的研究热点. 基于内在奖励的探索能力增强模型框架如图 6 所示.

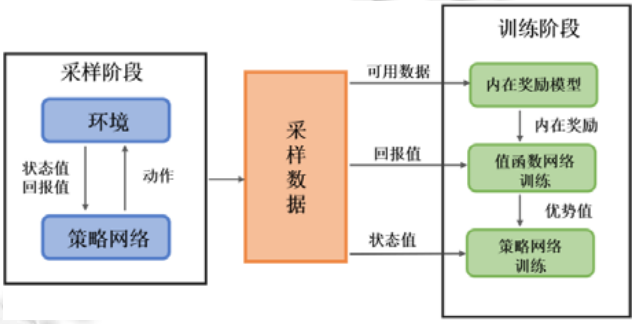


图 6　基于内在奖励的探索能力增强模型框架

目前，效果突出的内在奖励模型构建方法有信息熵最大化方法、伪计数方法和状态预测方法.

1) 信息熵最大化方法. 信息熵最大化方法基于信息增益理论，在训练中增加状态转移预测模型，将内在奖励的目标定义为最大化动作执行后状态转移预测模型可以获得的信息增益，即希望利用有限的交互次数获取更多的环境信息. 但由于整个模型的信息增益计算困难，信息熵最大化理论难以应用到算法中. Houthoof 等人提出了 VIVE (variational information maximizing exploration)算法，利用变分贝叶斯理论，对信息增益进行合理的近似估算，有效增强了智能体的探索能力[40].

2) 伪计数方法伪计数方法. 基于不确定性理论，认为不确定性越高的状态越有可能存在未知的动作奖励，并且状态的不确定性与智能体对状态的访问次数成反比. 而深度强化学习的状态空间极大，如何准确衡量某一确定状态的访问次数，是伪计数方法的关键问题. 其中，Bellemare 等人提出了基于上下文树切换(context tree switching, CTS)的伪计数方法，首次利用近似计数的思想，使用 CTS 密度模型对状态访问的次数变化情况进行模拟，将基于访问次数的奖励引入深度强化学习任务中[41]. Ostrovski 等人提出了基于像素卷积神经网络(pixel convolutional neural networks, PixelCNN)的伪计数方法，使用像素卷积神经网络进行伪计数，获得了比 CTS 密度模型更好的效果[42]. Tang 等人提出了基于自编码器的局部敏感哈希(locality-sensitive Hashing, LSH)方法，先使用自编码器进行特征降维，随后对降维后的特征进行哈希值计算，并基于每个哈希值对应的状态访问次数计算内在奖励，以更直接的计数方式达到了密度模型的效果[43].

3) 状态预测方法. 状态预测方法基于好奇心理论，认为生物个体学习技能的动机是对未知世界的好奇. 在强化学习中，建立预测智能体行为后果的模型，将好奇心定义为智能体预测自身行为后果能

力的误差, 通过对预测误差较大的状态给予更多的内在奖励, 引导探索方向. 随着预测模型的不断训练, 智能体将能均匀地向各方向探索环境. 所以, 如何利用采样数据构建并训练合适的状态预测模型, 是状态预测方法的主要研究方向. Stadie 等人提出了基于模型预测的探索奖励方法, 首次在深度强化学习中引入状态预测模型, 利用当前状态与动作来预测后续状态, 并以预测误差作为内在奖励进行训练, 在稀疏奖励任务中获得了显著的效果[44]. Pathak 等人提出了好奇心驱动探索方法, 将好奇心的取值定义为对环境变化预测的误差, 并在状态预测模型中引入自监督的逆动力学模型, 可以有效地在高维像素特征中提取与智能体动作结果相关的主要特征, 排除不影响智能体行为的部分环境因素, 不仅有着更高的训练效率, 还对不可控的环境噪声具有鲁棒性[14]. Burda 等人在好奇心驱动探索方法的基础上, 基于状态预测方法进行大量实验, 对比包括由像素、随机特征、自编码特征、逆动态特征的 4 种不同编码特征对学习效果的影响, 验证了自监督逆动力学模型的有效性[45]. 此外, 好奇心机制受到无法预测的随机状态特征的影响, 在环境中带有特殊噪声的情况下无法有效判断状态的新颖性. Burda 等人提出了基于随机网络蒸馏(random network distillation, RND)的奖励驱动探索方法, 将环境变化的不确定性分为偶然不确定性和认知不确定性, 并通过神经网络间的认知误差衡量认知不确定性, 解决了在特殊噪声情况下的失效问题, 并获得了更好的效果[15].

目前, 基于信息熵最大化的方法受限于信息熵计算精度而难以获得良好的效果, 对增强策略梯度方法探索能力的研究主要集中在伪计数方法和状态预测方法中. 与状态预测方法相比, 伪计数方法的偏差和误差更小, 不易受到环境的影响, 更适合用于奖励相对较多的任务; 而状态预测方法依赖于对环境变化情况的预测, 更适用于奖励极度稀疏和任务极度复杂的环境.

4.3 *Q*值函数方法中的探索能力增强机制

不同于策略梯度方法, *Q* 值函数方法受限于 *Q* 值函数的优化过程, 难以通过内在奖励的方式增强探索效率. 在策略梯度方法中, 只使用当前策略与环境交互得到的采样样本进行策略梯度值计算, 内在奖励在训练前期起到激励智能体探索的作用; 在训练后期, 内在奖励值趋近于 0, 几乎不会对训练过程有负面影响, 且可以保证最优策略的一致性. 但 *Q* 值函数方法需要在整个任务的状态-动作空间中计算全局 *Q* 值函数, 面临着过高估计的问题. 过高估计问题将导致智能体在训练过程中陷入局部最优值甚至使智能体策略崩溃, 所以奖励值的低偏差与低方差成为保证 *Q* 值函数方法训练效果的主要约束条件. 内在奖励在前期引入的高偏差和高方差导致无法增强 *Q* 值函数的探索能力, 还会引起训练策略的崩溃, 所以需要其他方案增强探索能力[46].

定义 7(过高估计). 在 *Q* 值函数方法的训练过程中, 受环境的不确定性或奖励噪声影响, 对应状态的确定动作的奖励为概率分布值而非固定值. 以奖励分布期望为中心, 大于奖励期望的奖励值更容易随着贪婪的 *Q* 函数优化过程传播, 小于奖励期望的奖励值则难以被学习, 导致通过深度神经网络得到的 *Q* 估计值通常大于真实值, 且无法正确估算真实的 *Q* 值函数的现象, 该现象称为过高估计.

然而可以使探索能力增强的方案并不局限于奖励空间中, 研究者已从动作空间、参数空间、网络结构等方面对探索能力进行研究, 设计了多样的探索方式. *Q* 值函数方法中探索能力增强方案的结构如图 7 所示.

- 在网络结构方向, Osband 等人提出了 Bootstrapped DQN 算法, 利用样本分布来近似总体分布的思想, 用多个随机初始化的 *Q* 网络代替原有的单个网络, 在实现中, 为保证效率只改变输出层构建不同的 *Q* 值函数, 由此得到的多个 *Q* 值形成样本集合, 可以对环境空间进行更多样化地探索, 并在采样数据中加入掩码, 使样本以一定概率用于 *Q* 值函数的训练, 保证 *Q* 值函数的多样性. 该算法摒弃了原有的无方向的ε-贪心探索策略, 通过 *Q* 值函数分布, 在保证探索范围的同时, 引导了探索的总体方向[47]. 由此, 研究者发现, 利用多个 *Q* 值输出表来表示动作价值分布会有着更好的稳定性. Bellemare 等人和 Dabney 等人进行了分布式 *Q* 值函数的尝试, 提出了更灵活的网络结构和优化方式, 分别称为 C51 算法[34]和 IQN 算法[48].
- 在参数空间方向, Fortunato 等人提出了 NoisyNet-DQN 算法, 改变原有动作空间的探索方式, 即利用噪声函数在参数空间中加入无偏噪声, 并通过随机采样的方式决定每个时刻噪声函数的参数, 有着

更强的探索能力和训练效率，并可以提高泛化能力[33]. Plappert 等人提出了基于自适应噪声比例的参数噪声扰动方法，在为参数空间加入噪声的基础上加入自适应的噪声范围，并验证此类方法在连续动作任务以及在策略梯度类方法上的效果，证明参数空间的噪声可以提升不同类别强化学习方法的训练效率[49]. Han 等人提出了基于噪声降低和在线权重调整(noise reduction and online weight adjustment)的参数扰动 DQN 算法 NROWAN-DQN，即在 NoisyNet-DQN 算法的基础上，考虑到引入网络噪声而导致不稳定的负面影响，提出在线权重调整的降噪机制，在提供探索能力的基础上使训练过程更稳定[50].

- 在动作空间方向，Hong 等人提出了动作差异驱动探索方法，希望增大当前动作选择网络与之前动作选择网络的差距，在损失函数中加入新、旧策略之间的距离衡量损失，使得在网络训练难以得到更高回报时，尽可能地朝不同方向进行探索，有效地防止动作策略陷入局部最优[51]. Ciosek 等人专注于连续动作空间任务，提出了 OAC (optimistic actor critic)算法，认为动作值在 Q 值方向上的梯度可以指示出更具价值的采样方向，为选择的动作加上梯度正方向的偏移，提升了与环境交互采样的质量[52].

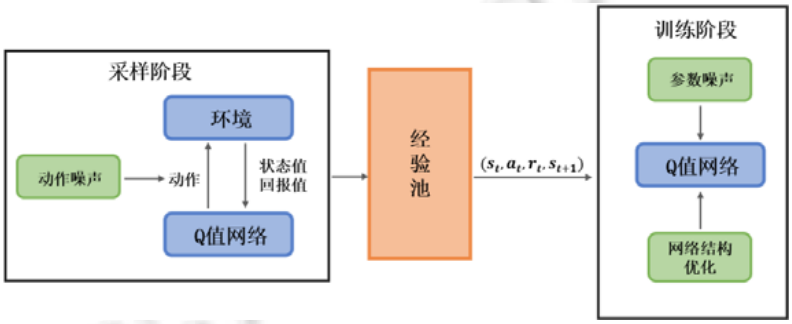


图 7 Q 值函数方法中的探索能力增强方案

目前，对于 Q 值函数方法，已有多种增强探索能力的方案，虽然都能通过增强探索能力在一定程度上提升算法的训练效率，但在对探索能力的提升有限，在复杂程度高、奖励稀疏的环境中难以获得显著成效.

4.4 多种探索增强机制的比较

在探索增强机制中，主要问题是如何能够在固定探索次数的情况下，采集到更多样性或更有价值的样本，需要探究如何评判样本的潜在价值或新颖性，并试图朝着使样本更新颖的方向进行动作探索. 在具体实现中，要求既能找到简单有效的样本新颖性衡量标准，又能在朝着样本新颖方向采样的同时不影响最大化回报的训练目标.

策略梯度方法根据当前策略与环境进行交互采样，通过估计当前策略的梯度进行优化. 所以在设计探索机制时，首要考虑与环境交互时探索机制的无偏性，即能够在不改变当前策略与策略梯度计算精度的情况下提升探索范围，且无动作偏差.

Q 值函数方法在任务的整个状态-动作空间内寻找最优 Q 值，且受环境的奖励噪声和奖励不确定性影响，经常会出现过高估计的问题. 所以在设计探索机制时，首要考虑不改变环境奖励的精度和不增加任务的不确定性，即在保证不会因引入不确定性而过高估计的情况下提升探索范围，且无奖励偏差.

对于本文所述的多种增强探索机制，在动作偏差、奖励偏差、探索能力和适用范围方向的对比情况见表 2. 其中，内在奖励只适用于策略梯度方法且有着最好的增强探索效果. 例如，可以在稀疏奖励甚至不使用外在奖励的情况下，通关复杂的冒险类游戏 Super Mario Bros 和 Montezuma's Revenge[45]，所以当涉及复杂任务的策略训练时，往往以策略梯度方法和内在奖励为基本框架进行算法设计[6]. 部分增强探索机制，例如分布式价值函数优化，由于结构限制只适用于 Q 值函数方法中. 此外，部分探索机制对各类强化学习算法均适用，例如参数空间探索[49]和动作差异驱动[51]，可以在任务的训练速度和训练得分上有所提升，但效果有限.

表 2 多种探索增强机制的对比

优化方向	策略梯度方法	Q 值函数方法	动作偏差	奖励偏差	探索能力
信息熵最大化	适用	不适用	无偏差	有偏差	中
伪计数	适用	不适用	无偏差	有偏差	高
状态预测	适用	不适用	无偏差	有偏差	高
分布式价值函数优化	不适用	适用	无偏差	无偏差	中
参数空间噪声	适用	适用	无偏差	无偏差	中
动作差异驱动	适用	适用	无偏差	无偏差	低
动作梯度驱动	不适用	适用	有偏差	无偏差	低

5 通过优化样本利用提高样本效率

20 世纪兴起的 Q 表法, 虽然受到时间和空间复杂度的限制而只能解决简单任务, 但其表格记录的动作奖励值只受其对应样本的影响, 且动作策略不会因学习率较大而崩溃, 有着较高的准确性和样本利用率.

在深度强化学习任务中, 受到采样能力的限制, 很难对状态和动作空间内所有的状态-动作对进行探索, 这就需要利用深度学习的特征提取能力进行特征空间的降维, 并利用泛化能力完成对未知状态-动作对的回报值进行估计, 使训练智能体的动作策略完成高维复杂任务成为可能.

在使用深度学习的特征提取能力和泛化能力与强化学习结合完成复杂任务的同时, 原有的强化学习模型框架也受到神经网络训练特性的影响, 具体表现在网络的不稳定性、样本分布偏差两个方面.

1) 网络的不稳定. 网络的不稳定性是指在深度神经网络中, 对单个样本进行梯度训练将导致整个神经网络的变化, 进而改变所有样本的网络输出值. 为防止因少量异常样本的训练导致的动作策略崩溃, 需要将学习率设为较小值, 以保证动作策略的稳定性.
2) 样本分布偏差. 样本分布偏差是指在深度强化学习中, 网络训练的目标函数与训练样本的分布相关, 如果进行训练的样本分布不能正确反映训练目标, 则会影响训练效果. 训练目标计算的不准确、训练过程中状态值函数或动作值函数误差的累积、采样策略与当前策略网络不一致等问题, 均有可能引起样本分布的偏差.

所以, 深度强化学习将面临更加复杂的样本利用挑战, 需要算法同时考虑不稳定性和样本分布偏差的问题. 高质量的训练样本不仅可以加快训练速度, 而且可以得到更优的动作策略. 如何在采集到的样本中更好地提取有效信息, 已成为深度强化学习中提高样本效率的重要方向.

5.1 方差减小策略

在深度强化学习模型中, 需要利用深度神经网络对状态价值函数或动作价值函数进行拟合. 由于价值函数的估值同时受到环境不确定性、策略变化、神经网络误差和采样数量限制的影响, 难以完成精准的价值估计, 而价值函数估计的精度直接影响训练效果. 偏差和方差是衡量估计值精准度的两个重要指标, 通常在深度强化学习中, 价值函数估值的无偏性很容易满足, 但在估值过程中存在较大的方差, 导致动作策略可能受到部分误差过大的估值影响而陷入局部最优甚至崩溃. 如何利用已有的采样数据获得精度更高的价值函数估值, 是能否有效地利用采样样本的决定性因素.

在策略梯度方法中, 通常通过优化策略梯度计算过程来降低梯度估计值的方差. 理论上, 如果能使用当前策略与环境进行足够多的交互采样, 就能精准地估计策略梯度的期望来进行策略优化. 但对于实际任务, 能够进行的采样次数远远小于能够准确估计期望所需的采样次数, 所以利用 Actor-Critic 框架引入状态价值函数对当前策略下状态潜在的奖励进行估计, 这种能够有效减小梯度估值的方差的框架, 使动作策略在训练过程更加稳定. GAE 方法则进一步改进策略梯度的计算方式, 利用优势值表示当前动作价值与当前状态下所有动作平均价值的差距, 并引入衰减因子将优势值序列进行拟合, 完成价值函数的重塑, 有效地提升了稳定性与训练效率[24]. 经过大量的实现验证, GAE 方法已成为策略梯度计算中降低估计方差并提高样本利用率的必备方法.

在 Q 值函数方法中, 可以通过优化 Bellman 方程计算方式来降低方差. Jiang 等人从理论上分析了 Q 值函数方法的方差, 并提出了双鲁棒估值计算方法, 同时使用动作价值网络和状态价值网络进行 Bellman 方程计算, 并在理论上证明了该方法在保证无偏性的同时有着更小的方差, 实验表明其有着更高的准确性[53]. 此外, 多步更新的方法也可以降低 Q 值函数的方差并加快训练速度, DQN 算法通过使用单步样本中的回报值与 Q 估计值来计算 Bellman 方程并完成训练更新, 偏差很小但方差很大, 如果同时使用多步动作的实际回报值之和来代替单步回报进行训练, 将在增大偏差的同时有效地降低方差, 所以在合理调整步数后, 多步更新的方法可以在偏差和方差之间权衡, 获得更好的训练效果. Munos 等人提出了 Retrace(λ)算法, 并指出, 多步学习的偏差主要来自与当前策略差异较大的样本, 并通过重要性采样的方式修正偏差, 得到了更稳定的低方差多步学习算法[21]. He 等人提出了基于最优收紧(optimality tightening, OT)的多步强化学习方法, 在多步强化学习中引入值函数的上下界约束, 在引入微小偏差的情况下, 提高优化速度和稳定性[54].

对于方差减小方向, 虽然深度强化学习模型中都存在一定的方差, 但策略梯度方法受方差影响要远大于 Q 值函数方法. 所以在策略梯度方法中, 方差的大小是决定模型稳定性和训练效率的决定性因素; 而在 Q 值函数方法中, 降低方差的优化并非必要且对训练效果的提升有限.

5.2 Q值函数方法的样本利用效率优化

Q 值函数方法需要在任务空间内寻找最优策略对应的 Q 值, 并通过贪婪的策略进行动作选择. 该类方法不要求训练样本完全由当前动作策略与环境进行交互采集, 在训练中可以用到此前所有的采集样本, 故可以归类为离策略(off-policy)优化方法. 在算法实现中, 将与环境交互得到的样本存入经验池中, 在网络优化阶段进行样本采集和训练优化. 由于经验池的容量有限和可供训练的样本量过大, 所以如何合理地进行经验存放和训练经验选取, 成为 Q 值函数方法中影响样本利用率的主要问题. 目前, 经验重要性和回合更新是提高样本利用率的主要优化方向.

对于经验重要性方向, 早在 1992 年, Lin 等人即发现数据样本的有效性有显著差异, 部分数据样本对于当前 Q 函数是没有训练价值的, 而在一定次数的训练后才具有训练价值[55]. 在深度强化学习广泛应用后, Schaul 等人更加深入地分析了采样样本的重要性, 提出了 PER 方法, 说明时间差分误差(TD-error)能够有效地衡量样本对于动作策略优化的重要程度, 以基于时间差分误差的概率权重为标准, 通过二叉堆优先队列来完成训练时对重要的经验进行选取, 在 Atari 2600 中获得了显著的效果提升[31]. Han 等人提出了定期更新的确定性策略梯度(regularly updated deterministic, RUD)算法, 通过对 DDPG 算法研究后发现, Q 值函数方法更偏重于学习旧经验而欠缺对新产生经验的学习, 所以引入定期更新经验池的机制来增大学习新经验的概率, 并降低 Q 值估计的标准差, 获得了比 DDPG 算法更好的学习效果[56]. Zha 等人提出了基于经验重放优化(experience replay optimization, ERO)的 DDPG 算法, 利用强化学习的思想引入样本选择网络, 选择对动作策略优化效率高的样本, 并通过样本选择网络间接导致的动作策略提升来反馈调节样本选择网络参数, 但受限于网络训练难度影响而效果有限[57].

回合更新的思想在 20 世纪的经典强化学习中就已出现, 至少可以追溯到 1993 年 Moore 等人提出的优先经验扫描方法, 对于强化学习采集到的回合经验中, 以回合的逆向顺序作为样本顺序来进行训练优化有着更高效的训练效果[58]. 但在深度强化学习中, 这种逆序优化虽然效率高, 但会引起 Q 值网络严重的过高估计问题. Lee 等人深入研究了回合逆序优化的思想, 提出了 EBU (episodic backward update)策略并用于 DQN 中, 在将回合逆序优化理论用于深度强化学习的同时, 使用自适应的扩散因子来抑制训练中的过高估计问题. 通过分析认为, 其在理论上具有高效训练的优势, 并在 Atari 中仅用 5%和 10%的样本就达到与 DQN 相同的平均值和中位数的性能[20]. 但 EBU 策略在抑制过高估计问题的同时也限制了逆序优化的性能, 没完全发挥逆序优化的优势. 所以, 如何更好地平衡过高估计问题和逆序优化性能, 仍是有待深入研究的方向.

目前, 对于 Q 值函数方法中样本利用效率的优化已取得了一定进展, 尤其是 PER 策略因兼容性较好已被广泛用于各类 Q 值函数方法中[5]. 但样本利用策略受限于计算性能影响, 仍未达到对经验重要性精准区分和对回合更新优势的有效利用的目标, 还具有优化提高的空间, 所以样本利用问题仍是目前研究的热点问题.

5.3 策略梯度方法的样本利用优化

策略梯度方法需要利用当前动作策略与环境进行交互, 并通过采集到的样本对动作策略的梯度值进行计算, 需要保证样本与当前动作策略的一致性, 故可以归类为在策略(on-policy)优化方法. 理论上, 用于梯度计算的样本必须由当前动作策略采集, 不能使用经验池来存储由其他策略采集到的经验, 每次的动作策略网络优化都需要重新在环境中进行采样, 会造成样本的严重浪费. 在实际任务中, 受到采样成本的限制, 该类方法的采样次数十分有限, 在每轮优化中只能进行数千次交互采样, 无法精准计算动作策略的梯度值. 为了策略优化的稳定性, 需要控制学习率而无法在当前轮次的优化中有效利用样本中的梯度信息. 所以, 如何在保证稳定性的同时应用当前样本完成更多的优化, 或从旧策略采集到的样本中获取有效信息, 成为策略梯度方法中样本利用的重要优化方向.

由于实际任务中的训练无法满足理论上样本必须由当前动作策略采集的要求, 如何应用当前样本完成更多的优化, 就成为策略梯度方法用于深度强化学习任务所要解决的首要问题. 对于 VPG 算法, 虽然使用深度神经网络完成了动作策略函数和价值函数的拟合, 但为保证学习效率, 需要在每轮训练中进行多次动作策略梯度优化, 导致训练过程不稳定, 经常因为少数几次偏差较大的优化使整个策略崩溃. 所以, TRPO 和 PPO 算法主要通过重要性采样的方式, 让动作策略梯度优化考虑到采样样本分布变化的影响, 同时约束每轮训练的策略变化程度, 有效地提高了样本利用率和稳定性, 并广泛用于策略梯度方法中.

对于旧策略采集到的样本, 即大多数策略梯度类算法抛弃的其他轮次的训练样本, 毫无疑问存在着对策略优化可以利用的有效信息, 但也混杂着大量无法利用的无效信息和噪声. 在考虑计算效率的同时, 如何对有效信息进行区分并利用到策略优化中, 成为旧样本利用的难题. Wang 等人提出了 ACER (actor-critic with experience replay)算法, 使用经验池来记录其他策略采集到的经验, 并将偏差矫正的重要性采样方法用于所有经验中, 可以有效计算不同策略间采样样本分布的变化[59]. 这种可以利用旧样本的方法, 为原有的在策略优化方法赋予了离策略优化的能力, 有着更高的样本利用率, 但其更复杂的实现过程也意味着更难与其他优化方法结合, 同时需要更多的存储空间与训练时间, 可以称为离策略优化的策略梯度方法[60]. Han 等人提出了基于维度空间重要性采样裁剪(dimension-wise importance sampling clipping, DISC)的策略梯度算法, 将 PPO 算法对重要性采样的裁剪扩展到每个维度中, 提高了对离策略样本的处理能力. 引入经验池对历史样本进行学习后, 使策略梯度方法的效果获得了显著的提升[61].

5.4 多种基于样本利用的优化方向比较

对于样本利用优化方向, 主要问题是如何能在样本数据确定的情况下, 更好地挖掘已有样本中的训练价值, 需要探究如何能够准确地对样本中的信息进行利用, 使优化的偏差和方差最小. 在具体实现中, 可以基于样本的重要性或学习价值, 选择更有学习价值且引入误差较小的样本进行训练, 或探究更稳定且快速的优化方式.

对于样本利用优化问题, 本文所提到的优化方向与算法如图 8 所示.

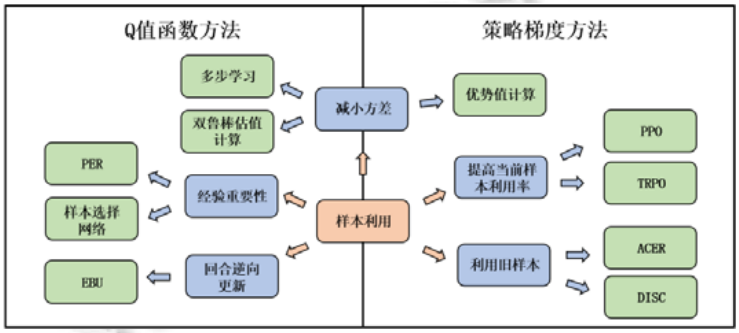


图 8 样本利用优化方向

在深度强化学习中，各类方法都需要考虑样本中高方差对训练效果的影响. 此外，因 Q 值函数方法和策略梯度方法框架的差异限制，Q 值函数方法的样本利用优化主要考虑在经验池中选择更合适的训练样本或以回合逆序来选取训练样本，策略梯度方法则着眼于如何对提高当前样本的利用率或使用其他策略采集到的样本.

对于样本利用优化方向的不同方法，在实际应用中还要考虑到与各类算法的兼容性、不同任务间的稳定性、实现复杂性与时间空间开销. 所以，目前方差减小策略和提高当前样本利用率是策略梯度方法中成熟且广泛使用的优化，经验重要性采样则是 Q 值函数方法中的首选.

6 结论与展望

深度强化学习方法结合了深度学习的强大表示能力和强化学习的决策优化能力，能够寻找序贯动作决策优化问题中的最优动作策略. 作为一种更接近人类学习思维的方法，深度强化学习有着显著的学习效果和广泛的应用前景；但同时，受到需要大量采样需求的限制，难以用于采样成本高的环境中. 所以，提高深度强化学习的样本效率成为研究热点[62].

对于提高样本效率问题，可以从“优化与环境交互过程以获得更有价值的采样样本”和“优化训练过程通过有限的样本训练出更优的动作策略”两个方面入手. 基于 Q 值函数的方法和策略梯度方法作为深度强化学习的两类方法，由于模型架构与训练目标的不同，导致在优化方向和思路中也存在着差异. 其中，

- Q 值函数的方法需要保证采样奖励的低噪声和无偏性来抑制过高估计带来的影响，可以利用所有的历史样本进行训练优化，所以目前主要通过在动作或参数等空间上增加噪声来增强探索效率，对探索能力效果提升较小；但对于样本利用优化方向，可以通过重要性采样和多步更新的机制来大幅度提升训练效率.
- 策略梯度方法需满足与环境交互的策略和当前动作策略函数相似，难以使用历史样本进行优化，但不会出现过高估计问题. 所以利用构建内在奖励的方式可以有效地提高策略的探索能力，更适合解决奖励极度稀疏的问题；但对样本利用优化方向，目前只有通过方差减小策略和修改训练目标等方式来略微提高样本效率.

本文提及的所有优化方向和相关算法的整理和对比如表 3 所示.

表 3 明确了各种优化方法的优化类别、优化方向、适用基线、相关方法和特点. 由此可知，虽然同时具有理论保证且实现简单的方差减少策略目前应用最广泛，但对于每类优化方法，均具有其独特的优势或发展潜力，在实际应用中，需要同时考虑到任务中动作与状态特点、时间与空间开销、数据采集代价、稳定性和模型复杂度等因素来选择合适的算法. 例如，目前深度强化学习算法的应用中有如下算法选择方案.

1) 在奖励极度稀疏的任务中，状态预测方法仍是提高样本效率的首选.
2) 在样本采集开销大的任务中，具有快速优化和收敛能力的回合更新方法更加适用.
3) 在计算资源充足的条件下，基于网络结构优化的 Q 值函数方法和基于旧样本利用的策略梯度方法往往能获得最佳的性能.

对于样本效率优化方向的深入研究，可以从以下 3 个角度进行.

1) 虽然已经提出了多种有效的优化策略，但受到复杂度和样本数量的影响，仍未能有效地解决问题，并且大多优化策略都在特定的任务中进行，并未应用到大部分强化学习任务中. 所以，以目前已有的研究方向为基础，还需进行深入的研究来寻找更加有效的方案，包括但不限于：在基于内在奖励设计的策略梯度探索增强方案中，大多数研究者倾向于在完成任务目标才能获得数值为 1 的奖励和完全无奖励的环境中进行实验，而忽略了内在奖励与密集的外在奖励结合时，内外奖励系数难以平衡而引发调参工作过于繁重和效果急剧下降的问题；在 Q 值函数方法中，目前回合逆向更新的方法无法解决过高估计问题，只能以牺牲优化效果的方式减少过高估计，同时也说明 DDQN 和 TD3 的多 Q 值函数优化未能完全解决过高估计问题.

2) 目前, 效果最为显著的方向集中在 Q 值函数方法的样本利用和策略梯度方法的环境探索上, 虽然 Q 值函数和策略梯度方法有所差异, 但作为解决同种问题的方法, 可以考虑进行方法思路的迁移, 将原有方法中的优化方案进行调整和改进, 用于另一方法中, 包括但不限于: Q 值函数与策略梯度方法中方差减小策略的转化; 在 Q 值函数方法中, 使用内在奖励的构建方式来提升探索效率; 在策略梯度方法中构建样本选择策略, 进行对历史采样中的有效样本进行选择.

3) 近年来, 深度强化学习的发展十分迅速, 但在已有的样本效率优化算法中, 理论保证不够完备, 主要原因是, 能够有效计算的数据较少和深度学习可解释性差. 由此, 理论上可行的方案往往难以运用到实际应用中, 信息熵最大化和优先经验回放等理论完善的方法需要近似计算而在实际中收效甚微, 依靠好奇心机制的探索模型却在实际中有极好的效果. 所以, 深入研究经典强化学习理论, 尝试将经典理论分析迁移至深度强化学习中, 或挖掘深度强化学习算法内在的理论依据, 对增强探索效率也将有重要的研究意义.

表 3 深度强化学习的样本效率优化方向

优化类别	优化方向	适用基线	相关方法	特点
增强对环境的探索能力	信息熵最大化	策略梯度	VIVE[40]	具有完善的理论和推导过程,但难以对信息增益值进行准确计算
	伪计数	策略梯度	基于 CTS 的伪计数[41]、基于 PixelCNN 的伪计数[42]、基于自编码器的 LSH[43]	理论上具有高效的探索能力,但在实际任务中依赖特征提取和编码过程的效果
	状态预测	策略梯度	基于模型的探索奖励[44]、好奇心驱动探索[14]、RND 奖励驱动探索[15]	目前具有最强的探索能力,适用于奖励极度稀疏的任务,但在简单任务上会因神经网络误差而引入偏差
	网络结构优化	Q 值函数	Bootstrapped-DQN[47]、C51[34]、IQN[48]	能够提升算法整体性能,但增加了模型的复杂度,提高结构和参数调优的难度
	参数空间优化	Q 值函数 策略梯度	NoisyNet-DQN[33]、基于自适应噪声比例的参数噪声扰动[49]、NROWAN-DQN[50]	探索机制可以不受任务环境的状态相关性与动作相关性影响,较好地避免陷入局部最优
	动作空间优化	Q 值函数 策略梯度	动作差异驱动探索[51]、OAC[52]	实现过程简单,对探索能力的提升有限,且会引入微小的偏差
提高对数据的利用能力	减小方差	Q 值函数 策略梯度	双鲁棒估值计算[53]、GAE[24]	具有良好的理论保证,应用广泛,依赖于对函数值或梯度值更精准的计算
	多步学习	Q 值函数	Retrace(λ)[21]、基于 OT 的多步强化学习[54]	显著提升了训练速度,但会因增大训练方差而影响收敛性
	经验重要性	Q 值函数	PER[31], RUD[56], ERO[57]	理论上具有很大的优化潜力,但难以准确衡量经验的重要性
	回合更新	Q 值函数	优先经验扫描法[58]、EBU[20]	具有快速优化和收敛能力,但需要合理的抑制过高估计
	利用旧样本	策略梯度	ACER[59], DISC[61]	目前效果提升最大的策略梯度方法的优化方案,但会使模型更复杂,提高结构和参数调优的难度

References:

[1] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd ed., MIT Press, 2018.

[2] Liu Q, Zhai JW, Zhang ZZ, Zhong S, Zhou Q, Zhang P, Xu J. A survey on deep reinforcement learning. Chinese Journal of Computers, 2018, 41(1): 1−27 (in Chinese with English abstract).

[3] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529−533.

[4] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y, Lillicrap T, Hui F, Sifre L, Driessche G, Graepel T, Hassabis D. Mastering the game of Go without human knowledge. Nature, 2017, 550(7676): 354−359.

[5] Hessel M, Modayil J, Hasselt H, Schaul T, Ostrovski G, Dabney W, Horgan D, Piot B, Azar M, Silver D. Rainbow: Combining improvements in deep reinforcement learning. In: Proc. of the AAAI. 2018. 3215−3222.

[6] Ye DH, Liu Z, Sun MF, Shi B, Zhao PL, Wu H, Yu HS, Yang SJ, Wu XP, Guo QW, Chen QB, Yin YT, Zhang H, Shi TF, Wang L, Fu Q, Yang W, Huang LX.. Mastering complex control in MOBA games with deep reinforcement learning. In: Proc. of the AAAI. 2020. 6672−6679.

[7] Chen JY, Yuan B, Tomizuka M. Model-free deep reinforcement learning for urban autonomous driving. In: Proc. of the ITSC. 2019. 2765−2771.

[8] Zhang TY, Huang ML, Zhang L. Learning structured representation for text classification via reinforcement learning. In: Proc. of the AAAI. 2018. 6053−6060.

[9] Takanobu R, Huang M, Zhao Z, Li F, Chen H, Zhu X, Nie L. A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning. In: Proc. of the IJCAI. 2018. 4403−4410.

[10] Yu K, Dong C, Lin L, Loy CC. Crafting a toolchain for image restoration by deep reinforcement learning. In: Proc. of the CVPR. 2018. 2443−2452.

[11] Zhou KY, Qiao Y, Xiang T. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In: Proc. of the AAAI. 2018. 7582−7589.

[12] Hu YJ, Da Q, Zeng AX, Yu Y, Xu YH. Reinforcement learning to rank in E-commerce search engine: Formalization, analysis, and application. In: Proc. of the KDD. 2018. 368−377.

[13] Zarkias KS, Passalis N, Tsantekidis A, Tefas A. Deep reinforcement learning for financial trading using price trailing. In: Proc. of the ICASSP.2019. 3067−3071.

[14] Pathak D, Agrawal P, Efros AA, Darrell T. Curiosity-driven exploration by self-supervised prediction. In: Proc. of the ICML. 2017. 2778−2787.

[15] Burda Y, Edwards H, Storkey A, Klimov O. Exploration by random network distillation. In: Proc. of the ICLR. 2019.

[16] Wang J, Liu Y, Li B. Reinforcement learning with perturbed rewards. In: Proc. of the AAAI. 2020. 6202−6209.

[17] Everitt T, Krakovna V, Orseau L, Legg S. Reinforcement learning with a corrupted reward channel. In: Proc. of the IJCAI. 2017. 4705−4713.

[18] Pinto L, Davidson J, Sukthankar R, Gupta A. Robust adversarial reinforcement learning. In: Proc. of the ICML. 2017. 2817−2826.

[19] Gu ZY, Jia ZZ, Choset H. Adversary A3C for robust reinforcement rearning. arXiv:1912.00330, 2019.

[20] Lee SY, Choi S, Chung SY. Sample-efficient deep reinforcement learning via episodic backward update. In: Proc. of the NeurIPS. 2019. 2110−2119.

[21] Munos R, Stepleton T, Harutyunyan A, Bellemare MG. Safe and efficient off-policy reinforcement learning. In: Proc. of the NIPS. 2016. 1046−1054.

[22] Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In: Proc. of the AAAI. 2016. 2094−2100.

[23] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: Proc. of the ICML. 2018. 1582−1591.

[24] Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P. High-dimensional continuous control using generalized advantage estimation. In: Proc. of the ICLR. 2016.

[25] Schulman J, Levine S, Moritz P, Jordan MI, Abbeel P. Trust region policy optimization. In: Proc. of the ICML. 2015. 1889−1897.

[26] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.

[27] Asis KD, Chan A, PitisS, Sutton RS, Graves D. Fixed-horizon temporal difference methods for stable reinforcement learning. In: Proc. of the AAAI. 2020. 3741−3748.

[28] Yang WY, Bai CJ, Cai C, Zhao YN, Liu P. Survey on sparse reward in deep reinforcement learning. Computer Science, 2020, 47(3): 182−191 (in Chinese with English abstract).

[29] Liu JW, Gao F, Luo XL. Survey of deep reinforcement learning based on value functionpolicygradient. Chinese Journal of Computers, 2019, 42(6): 1406−1438 (in Chinese with English abstract).

[30] Watkins CJCH, Dayan P. Technical note: *Q*-learning. Machine Learning, 1992, 8(3−4): 279−292.

[31] Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. In: Proc. of the ICLR. 2016.

[32] Wang ZY, Schaul T, Hessel M, Hasselt H, Lanctot M, Freitas N. Dueling network architectures for deep reinforcement learning. In: Proc. of the ICML. 2016. 1995−2003.

[33] Fortunato M, Azar MG, Piot B, Menick J, Hessel M, Osband I, Graves A, Mnih V, Munos R, Hassabis D, Pietquin O, Blundell C, Legg S. Noisy networks for exploration. In: Proc. of the ICLR. 2018.

[34] Bellemare MG, Dabney W, Munos R. A distributional perspective on reinforcement learning. In: Proc. of the ICML. 2017. 449−458.

[35] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. In: Proc. of the ICLR. 2016.

[36] Sutton RS, McAllester DA, Singh S, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: Proc. of the NIPS. 1999. 1057−1063.

[37] Mnih V, Badia AP, Mirza M, Graves A, Harley T, LillicrapTP, Silver D, Kavukcuoglu K. Asynchronous methods for deep reinforcement learning. In: Proc. of the ICML. 2016. 1928−1937.

[38] Ilyas A, Engstrom L, Santurkar S, Tsipras D, Janoos F, Rudolph L, Madry A. A closer look at deep policy gradients. In: Proc. of the ICLR. 2020.

[39] Li Y, Shao ZZ, Zhao ZD, Shi ZP, Guan Y. Design of reward function in deep reinforcement learning for trajectoryplanning. Computer Engineering and Applications, 2020, 56(2): 226−232 (in Chinese with English abstract).

[40] Houthooft R, Chen X, Duan Y, Schulman J, Turck FD, Abbeel P. VIME: Variational information maximizing exploration. In: Proc. of the NIPS. 2016. 1109−1117.

[41] Bellemare MG, Srinivasan S, Ostrovski G, Schaul T, Saxton D, Munos R. Unifying count-based exploration and intrinsic motivation. In: Proc. of the NIPS. 2016. 1471−1479.

[42] Ostrovski G, Bellemare MG, Oord A, Munos R. Count-based exploration with neural density models. In: Proc. of the ICML. 2017. 2721−2730.

[43] Tang H, Houthooft R, Foote D, Stooke A, Chen X, Duan Y, Schulman J, Turck FD, Abbeel P. #Exploration: A study of count-based exploration for deep reinforcement learning. In: Proc. of the NIPS. 2017. 2753−2762.

[44] Stadie BC, Levine S, Abbeel P. Incentivizing exploration in reinforcement learning with deep predictive models. arXiv:1507.00814, 2015.

[45] Burda Y, Edwards H, Pathak D, Storkey A, Darrell T, Efros AA. Large-scale study of curiosity-driven learning. In: Proc. of the ICLR. 2019.

[46] Yang M, Wang J. Bayesian deep reinforcement learning algorithm for solving deep exploration problems. Journal of Frontiers of Computer Science and Technology, 2020, 14(2): 307−316 (in Chinese with English abstract).

[47] Osband I, Blundell C, Pritzel A, Roy BV. Deep exploration via bootstrapped DQN. In: Proc. of the NIPS. 2016. 4026−4034.

[48] Dabney W, Ostrovski G, Silver D, Munos R. Implicit quantile networks for distributional reinforcement learning. In: Proc. of the ICML. 2018. 1104−1113.

[49] Plappert M, Houthooft R, Dhariwal P, Sidor S, Chen RY, Chen X, Asfour T, Abbeel P, Andrychowicz M. Parameter space noise for exploration. In: Proc. of the ICLR. 2018.

[50] Han S, Zhou WB, Liu J, Lü S. NROWAN-DQN: A stable noisy network with noise reduction and online weight adjustment for exploration. Arxivabs/2006.10980, 2020.

[51] Hong ZW, Shann TY, SuSY, Chang YH, Fu TJ, Lee CY. Diversity-driven exploration strategy for deep reinforcement learning. In: Proc. of the NeurIPS. 2018. 10510−10521.

[52] Ciosek K, Vuong Q, Loftin R, Hofmann K. Better exploration with optimistic actor critic. In: Proc. of the NeurIPS. 2019. 1785−1796.

[53] Jiang N, Li LH. Doubly robust off-policy value evaluation for reinforcement learning. In: Proc. of the ICML. 2016. 652−661.

[54] He FS, Liu Y, Schwing AG, Peng J. Learning to play in a day: Faster deep reinforcement learning by optimality tightening. In: Proc. of the ICLR. 2017.

[55] Lin LJ. Self-improving reactive agents based on reinforcement learning, planning and teaching. Machine Learning, 1992, 8: 293−321.

[56] Han S, Zhou WB, Lü S, Yu JY. Regularly updated deterministic policy gradient algorithm. Knowledge-based Systems, 2021, 214: Article No.106736.

[57] Zha D, Lai KH, Zhou K, Hu X. Experience replay optimization. In: Proc. of the IJCAI. 2019. 4243–4249

[58] Moore AW, Atkeson CG. Reinforcement learning with less data and less time. Machine Learning, 1993, 13: 103–130.

[59] Wang ZY , Bapst V, Heess N, Mnih V, Munos R, Kavukcuoglu K, Freitas N. Sample efficient actor-critic with experience replay. In: Proc. of the ICLR. 2017.

[60] Degris T, White M, Sutton RS. Linear off-policy actor-critic. In: Proc. of the ICML. 2012. Article No.268.

[61] Han S, Sung Y. Dimension-wise importance sampling weight clipping for sample-efficient reinforcement learning. In: Proc. of the ICML. 2019. 2586–2595.

[62] Lü S, Han S, Zhou WB, Zhang JW. Recruitment-imitation mechanism for evolutionary reinforcement learning. Information Sciences, 2021, 553: 172–188.

附中文参考文献:

[2] 刘全, 翟建伟, 章宗长, 钟珊, 周倩, 章鹏, 徐进. 深度强化学习综述. 计算机学报, 2018, 41(1): 1–27.

[28] 杨惟轶, 白辰甲, 蔡超, 赵英男, 刘鹏. 深度强化学习中稀疏奖励问题研究综述. 计算机科学, 2020, 47(3): 182–191.

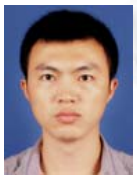
[29] 刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述. 计算机学报, 2019, 42(6): 1406–1438.

[39] 李跃, 邵振洲, 赵振东, 施智平, 关永. 面向轨迹规划的深度强化学习奖励函数设计. 计算机工程与应用, 2020, 56(2): 226–232.

[46] 杨珉, 汪洁. 解决深度探索问题的贝叶斯深度强化学习算法. 计算机科学与探索, 2020, 14(2): 307–316.

张峻伟(1998－), 男, 硕士生, 主要研究领域为人工智能, 机器学习.

于佳玉(1997－), 男, 硕士, 主要研究领域为人工智能, 机器学习.



吕帅(1981－), 男, 博士, 副教授, 博士生导师, CCF 高级会员, 主要研究领域为人工智能, 机器学习, 自动推理.

龚晓宇(1997－), 男, 硕士生, CCF 学生会员, 主要研究领域为人工智能, 机器学习.

张正昊(1996－), 男, 硕士, 主要研究领域为人工智能, 机器学习.