基于自适应噪声的最大熵进化强化学习方法

王君逸 [1] 王 志 [1] 李华雄 [1] 陈春林 [1]

摘 要 近年来, 进化策略由于其无梯度优化和高并行化效率等优点, 在深度强化学习领域得到了广泛的应用. 然而, 传统基于进化策略的深度强化学习方法存在着学习速度慢、容易收敛到局部最优和鲁棒性较弱等问题. 为此, 提出了一种基于自适应噪声的最大熵进化强化学习方法. 首先, 引入了一种进化策略的改进办法, 在 "优胜"的基础上加强了 "劣汰", 从而提高进化强化学习的收敛速度; 其次, 在目标函数中引入了策略最大熵正则项, 来保证策略的随机性进而鼓励智能体对新策略的探索; 最后, 提出了自适应噪声控制的方式, 根据当前进化情形智能化调整进化策略的搜索范围, 进而减少对先验知识的依赖并提升算法的鲁棒性. 实验结果表明, 该方法较之传统方法在学习速度、最优性收敛和鲁棒性上有比较明显的提升.

关键词 深度强化学习, 进化策略, 进化强化学习, 最大熵, 自适应噪声

引用格式 王君逸, 王志, 李华雄, 陈春林. 基于自适应噪声的最大熵进化强化学习方法. 自动化学报, 2023, **49**(1): 54−66

DOI 10.16383/j.aas.c220103

Adaptive Noise-based Evolutionary Reinforcement Learning With Maximum Entropy

WANG Jun-Yi[1] WANG Zhi[1] LI Hua-Xiong[1] CHEN Chun-Lin[1]

Abstract Recently, evolution strategies have been widely investigated in the field of deep reinforcement learning due to their promising properties of derivative-free optimization and high parallelization efficiency. However, traditional evolutionary reinforcement learning methods suffer from several problems, including the slow learning speed, the tendency toward local optima, and the poor robustness. A systematic method is proposed, named adaptive noise-based evolutionary reinforcement learning with maximum entropy, to tackle these problems. First, the canonical evolution strategies is introduced to enhance the influence of well-behaved individuals and weaken the impact of those with bad performance, thus improving the learning speed of evolutionary reinforcement learning. Second, a regularization term of maximizing the policy entropy is incorporated into the objective function, which ensures moderate stochastically of actions and encourages the exploration to new promising solutions. Third, the exploration noise is proposed to automatically adapt according to the current evolutionary situation, which reduces the dependence on prior knowledge and promotes the robustness of evolution. Experimental results show that this method achieves faster learning speed, better convergence to global optima, and improved robustness, compared to traditional approaches.

Key words Deep reinforcement learning, evolution strategies, evolutionary reinforcement learning, maximum entropy, adaptive noise

Citation Wang Jun-Yi, Wang Zhi, Li Hua-Xiong, Chen Chun-Lin. Adaptive noise-based evolutionary reinforcement learning with maximum entropy. *Acta Automatica Sinica*, 2023, **49**(1): 54−66

近年来, 深度强化学习作为一种能够有效解决机器学习问题的方法[1], 在自动驾驶[2−3], 轨迹追踪[4−5]与目标定位[6] 问题, 复杂多对象任务[7], 组合优化问题[8], 自动控制[9−10] 等领域都得到了广泛应用. 而作为这一方法的核心, 深度神经网络的更新方法一直以来都备受学界的关注[11−12]. 其中, 许多研究中提出的方法在更新网络参数时都采用了基于梯度的策略迭代方法[13], 这些研究结果表明基于梯度的策略迭代方法能够有效解决非线性系统[14], 多智能体强化学习[15] 等问题. 基于梯度的策略迭代方法即通过计算目标函数的梯度对深度神经网络进行反向传播, 例如蒙特卡洛策略梯度[1], 近端策略优化 (Proximal policy optimization, PPO)[16] 以及它们的改进方法[17−18]. 其他一些相关研究成果则大多着眼于将 PPO 的思想应用于诸如参数分布优化的更多领域[19]. 但这些方法都仍然是以基于梯度的策略迭代方法为核心的. 然而, 随着机器学习问题日趋复杂, 深度强化学习所需要处理的目标函数的维度也随之不断增

收稿日期 2022-02-18 录用日期 2022-06-16
Manuscript received February 18, 2022; accepted June 16, 2022
国家自然科学基金 (62006111, 62073160, 62176116), 江苏省自然科学基金 (BK20200330) 资助
Supported by National Natural Science Foundation of China (62006111, 62073160, 62176116), Natural Science Foundation of Jiangsu Province (BK20200330)
本文责任编委 王雪松
Recommended by Associate Editor WANG Xue-Song
1. 南京大学控制科学与智能工程系 南京 210008
1. Department of Control Science and Intelligence Engineering, Nanjing University, Nanjing 210008

大, 导致基于梯度的策略迭代方法的训练速度和收敛性受到较大影响[20].

为了找到一种更容易探索到全局最优、提高收敛速度的方法, 一些研究者开始尝试使用无梯度优化方法. 2017 年, OpenAI 发表的论文[20] 中采用进化计算中的进化策略 (Evolution strategies, ES) 对深度神经网络参数进行更新, 并借助多线程并行运算来提高效率, 为强化学习问题提供了一种新型的解决方案—进化强化学习方法. 不同于传统强化学习通过对动作进行扰动来优化参数, 进化方法则是直接对网络参数进行扰动, 在每次循环中根据计算得到的适应性函数的大小对参数添加噪声, 因此进化方法不需要通过梯度下降的反向传播. 此外, 进化策略由于高度的并行性并且只需要在不同进程中传递随机种子, 相对于策略梯度而言训练速度较快. 但相较于传统深度强化学习方法而言具有比较大的随机性, 因此被称为 "黑盒优化"[20]. 在 2018 年 Uber 发表的论文[21] 中, 研究者提出了利用进化方法中的另外一个方法—遗传算法来实现对网络参数的优化. 后续的研究中一般更多采用相对更为容易实现的 ES 来作为新算法的框架基础和用于对比的基准算法.

研究者们通过理论论证分析了在深度强化学习中使用进化策略的若干优势. Lehman 等[22] 解释了为何进化策略能够探索到与梯度优化的搜索范围不同的区域, 从而可以使进化强化学习求解到与梯度优化方法不同的最优点. Zhang 等[23] 通过一系列基于混合国家标准和技术研究所 (Mixed National Institute of Standards and Technology, MNIST) 数据库的实验来揭示进化策略和随机梯度下降之间的联系与区别, 并证明了进化强化学习的高准确度. 在理论工作完成之后, 众多学者们不断尝试将进化策略应用于多种强化学习前沿问题. Khadka等[24] 提出了用于改进深度确定性策略梯度[1] 的进化强化学习方法. Shi 等[25] 将进化策略应用于最大熵强化学习, 提出了最大熵策略搜索方法. Song 等[26] 用进化策略改进了元学习方法, 提出了基于进化策略的模型无关的元学习. Majumdar 等[27] 提出的多智能体进化强化学习和 Long 等[28] 提出的进化种群学习模式都是利用进化策略对多智能体强化学习方法的改进. Wang 等[29] 提出的实例加权增量进化策略是一种将进化策略应用于动态空间中的学习问题的方法.

然而, 大多数上面所提及的方法把重点放在将自然进化策略 (Natural ES, NES)[30] 最基本的形式用在不同的强化学习算法上, 只有较少的研究涉及对进化强化学习本身的方法改进上[31−32]. 尽管使用 NES 能够使方法的结构相对得到简化, 但是同时也将进化策略本身具有的一些问题带到了深度强化学习方法中. 目前进化策略的问题主要包括:

1) 进化策略的梯度下降方向与最优下降方向存在偏差, 导致了进化强化学习收敛速度较慢[33−34];

2) 进化策略的探索性较弱, 容易陷入局部最优[25, 33];

3) 进化策略对超参数变化比较敏感, 方法需要依赖超参数设计上的先验知识, 鲁棒性需要提升[32−33].

为了提高进化强化学习算法在这 3 个方面的性能, 本文提出了一种基于自适应噪声的最大熵进化强化学习方法 (Adaptive noise-based evolutionary reinforcement learning with maximum entropy, AERL-ME), 这一方法的贡献如下:

1) 通过改进进化策略的更新公式, 在传统算法只注重 "优胜"的基础上增加了自然选择中的 "劣汰", 削弱了由于随机探索带来的下降方向偏移问题的影响, 提高了方法的收敛速度;

2) 通过在目标函数中引入策略最大熵来鼓励智能体对于新策略的探索, 使智能体在探索新策略和利用已知策略的达到平衡, 提升了方法跳出局部最优的能力;

3) 提出了一种基于 Kullback-Leibler (KL) 散度的自适应噪声控制机制, 这一机制能够根据当前进化的情形智能地调节算法的一个关键参数—噪声标准差的值来调整进化策略的探索范围, 改善了传统进化策略方法由于依赖超参数设计上的先验知识而导致的鲁棒性较弱的缺陷.

本文的组织结构如下: 第 1 节介绍进化强化学习相关背景知识; 第 2 节介绍本文提出的改进方法 AERL-ME; 第 3 节介绍实验中用到的模拟环境和实验设计思路, 给出了对比实验, 消融实验和灵敏度分析实验的结果和分析; 第 4 节总结全文, 说明本文方法仍存在的不足和可以改进的方向.

1 背景知识—进化强化学习

强化学习是基于马尔科夫决策过程的一种应用于解决机器学习问题的方法[1]. 一个马尔科夫决策过程由一个五元组 $< S, A, T, R, \gamma >$ 组成, 其中 S 代表状态集合, A 代表动作集合, $T : S \times A \times S \to [0, 1]$ 代表状态转移概率矩阵, $R : S \times A \to R$ 代表奖励函数, $\gamma \in (0, 1]$ 代表折扣因子, 策略 π 被定义为一种将状态映射到动作集上的概率分布函数: $S \times A \to [0, 1]$, 并且 $\sum_{a \in A} \pi(a|s) = 1, \forall s \in S$. 强化学习的目标在于找到能够使长期收益 $J(\pi)$ 最大的最优策略 π^*:

$$J(\pi) = \mathrm{E}_{\tau \sim \pi(\tau)}[r(\tau)] = \mathrm{E}_{\tau \sim \pi(\tau)} \left(\sum_{i=0}^{\infty} \gamma^i r_i \right) \quad (1)$$

式中, $\tau=(s_0, a_0, s_1, \cdots)$被称为一个学习片段, $\pi(\tau)=p(s_0)\Pi_{i=0}^{\infty}\pi(a_i|s_i)p(s_{i+1}|a_i, s_i)$, r 指的是在状态s_i下执行动作 a_i 之后得到的即时奖励, $r=R(s_i, a_i)$, γ 为衰减因子.

受自然界生物进化的启发, 进化强化学习被提出以解决具有高维深度强化学习的问题. 不同的进化强化学习方法在种群表示形式和更新方式上各有差异, 本文所使用的进化强化学习方法基于的是自然进化策略 NES, 其核心思想是利用期望适应度的抽样梯度迭代更新搜索分布的参数. 定义搜索分布为基于参数 θ 的密度函数$p(z|\theta)$. 用 $f(z)$ 表示在实例 z 下计算得到的适应性函数值 (一般取为蒙特卡洛估计下的目标函数值). 在每一次迭代中, 通过参数化的搜索分布抽样生成新的种群, 并对种群中的每个实例评估适应性函数值. 由此, 在特定的搜索分布上适应性函数值的期望可以写作:

$$J(\theta)=\mathrm{E}_{\theta}[f(z)]=\int f(z)p(z|\theta)\mathrm{d}z \tag{2}$$

与策略梯度方法的更新公式类似, 自然进化策略根据下式对 θ 进行梯度更新:

$$\nabla_{\theta}J(\theta)=\nabla_{\theta}\int f(z)p(z|\theta)\mathrm{d}z=\mathrm{E}_{\theta}[f(z)\nabla_{\theta}\ln p(z|\theta)] \tag{3}$$

显然, 通过遍历所有可能的实例来求期望的准确值是不现实的. 一般常用的方法是通过随机抽样之后对期望值进行估计. 对于一个由 m 个实例组成的种群 $(z_1, z_2, \cdots, z_m)$, 式 (3) 中的搜索梯度可被估计为:

$$\nabla_{\theta}J(\theta)=\mathrm{E}_{\theta}[f(z)\nabla_{\theta}\ln p(z|\theta)]\approx\frac{1}{m}\sum_{i=1}^{m}f(z_i)\nabla_{\theta}\ln p(z_i|\theta) \tag{4}$$

在每一次迭代中, 自然进化策略向着能够获得更大适应性函数的方向估计参数的搜索梯度. 而在强化学习领域, 自然进化策略将直接在神经网络的参数空间中搜索有效的策略, 种群 $(z_1, z_2, \cdots, z_m)$ 通常被实例化为一个具有以 θ 为中心的对角协方差矩阵特征的多元高斯分布, 即 $z_i=\theta+\epsilon_i$, $\epsilon_i\sim\mathrm{N}(0, \sigma^2)$. 根据多元高斯分布的密度函数, 式 (3) 中的搜索梯度公式在多元高斯分布下就可表示为:

$$\nabla_{\theta}J(\theta)=\nabla_{\theta}\mathrm{E}_{\epsilon\sim\mathrm{N}(0, \sigma^2 I)}[f(\theta+\epsilon)]=\frac{1}{\sigma^2}\mathrm{E}_{\epsilon\sim\mathrm{N}(0, \sigma^2 I)}[f(\theta+\epsilon)\epsilon] \tag{5}$$

沿用式 (4) 的思路, 通过随机抽样之后就可以根据抽样样本将式 (5) 中的搜索梯度估计为:

$$\nabla_{\theta}J(\theta)\approx\frac{1}{m\sigma^2}\sum_{i=1}^{m}f(\theta+\epsilon_i)\epsilon_i \tag{6}$$

根据式 (6), 在每次迭代中, 网络参数的更新公式就可以表示为:

$$\theta_{t+1}=\theta_t+\frac{lr}{m\sigma^2}\sum_{i=1}^{m}f(\theta+\epsilon_i)\epsilon_i \tag{7}$$

式中, lr 代表学习率.

通过应用自然进化策略, 传统强化学习梯度优化的过程就简化成了通过高斯扰动向量$\epsilon\sim\mathrm{N}(0, \sigma^2 I)$进行采样, 评估被扰动之后的各个搜索实例的表现 (以适应性函数为评价标准), 汇总一系列搜索实例的结果并迭代更新几个步骤.

在应用进化强化学习方法时, 一般会将所有实例以随机种子的方式一一分配给子线程去处理, 如果主线程预先记录了分配给各个子线程的随机种子, 那么主线程就能知道所有子线程生成的扰动. 这样, 子线程就只需要向主线程回传该实例的适应性函数值, 这大大提高了通信效率, 从而实现高度的并行化.

2 本文的方法

2.1 本文方法总体框架

尽管进化强化学习算能够高效的利用并行化处理来进行无梯度优化, 但这一方法仍存在若干不足之处. 为了进一步增强进化强化学习方法的学习性能. 本文根据一系列改进思路, 提出了 AERL-ME 方法, 该方法通过 3 个贡献点分别对传统进化强化学习的 3 个不足之处做出了改进. AERL-ME 方法的整体结构如图 1 所示.

1) 传统进化强化学习的收敛速度相对较慢, 这主要是因为方法在更新时采用了所有实例的信息, 这导致了方法的下降方向会受到一些进化方向错误的实例的影响而产生偏移. 而这样的偏移会减慢算法的收敛速度. 为此, 本文引入了一种进化策略的改进方法, 该方法除了沿用进化策略的更新方法, 即适应性越好的样本占有更大权重之外, 还进一步采取了淘汰种群中排名靠后的实例的方法, 使这些实例不会在更新时对下降方向产生干扰, 从而提高了方法的收敛速度.

2) 进化强化学习在中后期容易陷入局部最优, 这是因为传统进化强化学习的目标中仅包含了适应性函数. 这导致了智能体更倾向于直接利用已经学习到的策略而忽视了对新策略的探索. 为了鼓励智能体在学习过程中对新策略进行探索, 本文在方法

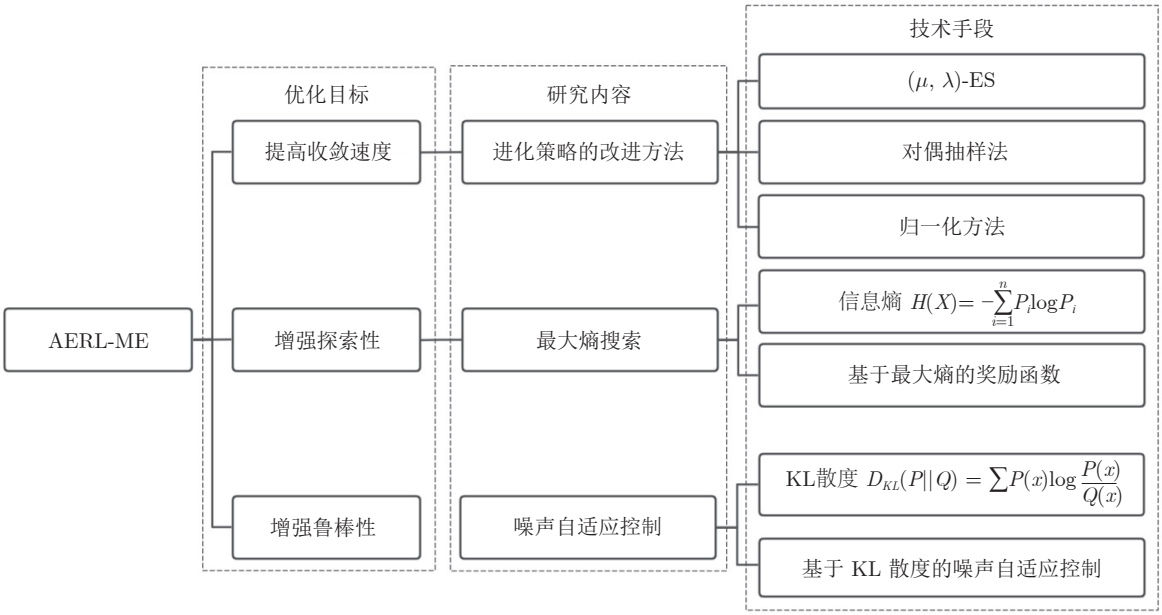


图 1　基于自适应噪声的最大熵进化强化学习方法的结构

Fig. 1　The structure of AERL-ME

的目标函数中增加了最大熵正则化项, 平衡了智能体的探索能力和利用能力, 使方法更容易跳出局部最优, 获得更好的学习结果.

3) 进化强化学习对于参数噪声标准差 σ 的变化比较敏感. 随着初始的 σ 选取的不同, 方法的学习性能会有相当大的偏差, 甚至导致无法收敛. 进化强化学习方法的鲁棒性需要提升. 在实际应用中进化强化学习方法相当依赖参数设置的先验知识, 为了得到更好的结果需要实现花费许多时间在调参上. 本文仿照自适应学习率的形式, 提出了一种基于 KL 散度的自适应噪声控制机制, 该机制能够在学习过程中根据当前情形智能地调整噪声标准差 σ 的值, 从而改变进化策略的探索范围. 从结果来看, 能够使噪声标准差在学习过程中慢慢调整到使性能更好的范围之内. 这样即使是在初始参数设置不合理的情况下也能获得更好的学习性能, 提升了方法的鲁棒性.

2.2　进化策略的改进方法

为了提升方法的收敛速度, 本文引入了一种进化策略的改进方法 (Canonical ES, CES)[34], 这一方法主要是在子代实例的生成和迭代更新上采取了一些改进措施.

1) 在传统进化强化学习方法中, 生成的子代种群 $(z_1, z_2, \cdots, z_m)$ 中的 m 个实例都将根据式 (7) 参与到网络参数的更新中. 但实际上, 在这些实例中有一部分的性能并没有比扰动前的性能更好. 用生物学的观点来看, 这些实例实际上是朝着错误的方向进行了进化. 但方法中这些进化方向错误的子代实例仍然会根据式 (6) 影响方法的梯度下降方向. 这样的结果会导致方法的梯度下降方向向着错误的方向产生了偏移. 尽管这样的偏移可以通过权重的调节来削弱, 但仍然无法彻底无视. 这一问题就是导致进化强化学习前期收敛速度较慢的原因.

CES 采用了进化策略的一种改进形式 (μ, λ)-ES 来改进传统方法, 在每一次迭代中生成 λ 个子代实例, 但仅选择其中适应性函数值前 μ 个子代实例参与更新. 这样的做法在传统方法 “优胜” (即适应性函数值越高的实例在更新时拥有更大的权重) 的基础上补上了 “劣汰” (即种群排名较后的实例将直接被淘汰, 不参与网络更新). 从而一定程度上提高了进化强化学习的收敛速度.

2) 如果仔细考虑子代种群的生成过程时, 会注意到进化强化学习的噪声扰动机制有可能导致有偏的搜索. 比如当大部分子代实例的搜索方向都是与正确的下降方向相反的, 那么方法的更新就会收到较大的影响. 尽管根据大数定理, 当子代实例足够多时, 能够保证搜索是无偏的. 但是在实际应用中, 进化强化学习用到的线程数毕竟是有限的, 仍然不能忽视有偏搜索可能带来的影响.

为了防止有偏搜索, CES 运用了对偶抽样法, 根据这一方法, 在主线程每一次抽样完成后, 将同时生成正负两个噪声, 即 $z_{i_+} = \theta + \epsilon_i$, $z_{i_-} = \theta - \epsilon_i$. 对偶抽样法有 2 个优势: a) 根据对偶抽样法生成的

子代种群中的实例始终是两两对称的, 这就避免了可能存在的有偏搜索; b) 通过运用对偶抽样法, 对于方法本身来说也只需要进行原先一半次数的抽样, 提高了运行和存储效率.

3) 式 (7) 参数更新时 ϵ_i 的权重 $f(\theta+\epsilon_i)$ 的取值一般根据的是蒙特卡洛估计下的目标函数值 $J(\pi_{\theta+\epsilon_i})$ (具体形式见第 3 节), 但目标函数在不同问题下的取值范围会有比较大的区别, 如果直接将其直接作为权重可能会使得方法更新的结果会受到问题本身的影响. 这可能会影响进化强化学习对于不同问题的学习能力. CES 运用了一种更为通用的归一化方法来避免这一问题, 归一化后第 i 个实例的权重 $f(\theta+\epsilon_i)$ 被表示为:

$$f(\theta+\epsilon_i)=\begin{cases}\dfrac{\log(\mu+0.5)-\log j}{\sum\limits_{k=1}^{\mu}\log(\mu+0.5)-\log k}, & 0<j\leq\mu\\ 0, & j>\mu\end{cases}\tag{8}$$

式中, j 表示 $J(\pi_{\theta+\epsilon_i})$ 在所有 $J(\pi_{\theta+\epsilon})$ 中的排名. 通过这样的归一化方法, 更新时的权重就只和该实例的排名有关, 而与它在具体问题下的目标函数取值无关, 从而增强了方法对于不同问题的学习能力.

2.3 最大熵搜索

在第 2 节中已经给出了归一化后的权重 $f(\theta+\epsilon_i)$ 的形式, 而它是根据 $J(\pi_{\theta+\epsilon_i})$ 的排名得到的, 在本节中将介绍 $J(\pi_{\theta+\epsilon_i})$ 的改进方法.

与一般的强化学习问题一样, 传统的进化强化学习的目标仍然是找到长期期望收益最大的策略 π^*, 因此类比式 (1), 进化强化学习的目标函数 $J(\pi_{\theta+\epsilon_i})$ 可以表示为:

$$J(\pi_{\theta+\epsilon_i})=\mathrm{E}_{\tau\sim\pi_{\theta+\epsilon_i}(\tau)}\left(\sum_{j=0}^{\infty}\gamma^j r_j\right)\tag{9}$$

但实际应用中是不可能通过遍历的方式求得这个期望的精确值, 所以一般通过蒙特卡洛估计法, 通过贪心策略获取一个学习片段 $\tau=(s_0, a_0, s_1, \cdots)$, 并根据这个片段计算得到的回报作为 $J(\pi_{\theta+\epsilon_i})$ 的估计:

$$J(\pi_{\theta+\epsilon_i})\approx\sum_{j=0}^{\infty}\gamma^j r_j\tag{10}$$

然而, 因为这一估计方法依据的的是贪心策略, 因此在任一状态智能体只会选择当前已知期望回报最大的动作, 而不会去探索其他未知的策略, 这样仅仅追求利用的方法会导致方法学习到一个局部最优的策略后就不再对新的策略进行学习. 因此传统的进化强化学习的探索性不足, 容易陷入局部最优.

为了尽可能在学习过程中平衡探索与利用, 本文参考了柔性行为—评判模型[35] 和最大熵策略搜索[25] 的工作, 在目标函数中添加了熵正则化项, 以此增强方法对于未知策略的探索性.

根据香农的的信息论, 对于一个有着 n 个可能结果的随机事件 X, 它的信息熵 $H(X)$ 可以表示为:

$$H(X)=-\sum_{i=1}^{n}p_i\log p_i\tag{11}$$

式中, p_i $(i=1, 2, \cdots, n)$ 为第 i 个结果发生的概率. 根据式 (11), 信息熵 $H(X)$ 的值越大, 事件 X 的结果越不确定. 而在深度强化学习领域, 一个离散随机性策略 π 可以表示为一个在不同状态下的随机事件. 当智能体处于某一状态 s_i 时, 它根据策略 π 选择执行动作 a_j 的概率可以表示为 $\pi(a_j|s_i)$. 由式 (11) 就可以给出该策略 π 在状态 s_i 时信息熵 $H(\pi(\cdot|s_i))$:

$$H(\pi(\cdot|s_i))=-\sum_{j=1}^{m}\pi(a_j|s_i)\log\pi(a_j|s_i)\tag{12}$$

式中, m 为可选动作的数量.

由于离散随机性策略同样是一个随机事件, 它也遵循上文提到的信息熵的性质. 也就是说, 如果一个策略在某一状态的信息熵越大, 当智能体到达这一状态时, 它所选择的动作就越无法确定, 这在一定程度上就增加了方法的探索性, 这就是引入最大熵模型的作用. 但如果仅仅考虑将信息熵最大化, 获得的将是一个一个完全随机的策略, 这也没有任何的利用价值. 综上所述, 本文方法所使用的目标函数将传统的奖励函数与信息熵函数进行了结合, 在蒙特卡洛估计的情况下可表示为:

$$J_e(\pi_{\theta+\epsilon_i})\approx\sum_{j=0}^{\infty}\gamma^j(r_j+\alpha H(\pi_{\theta+\epsilon_i}(\cdot|s_j)))\tag{13}$$

式中, α 表示温度因子, 它衡量了目标函数中奖励函数与信息熵函数的相对重要性. 在这一目标函数中, 奖励函数衡量的是智能体在整个过程中收获的奖励, 它更加鼓励智能体去利用已经学习到的策略, 更倾向于学到一个确定性的策略; 而信息熵函数则相反, 更加鼓励智能体去探索新的策略. 而温度因子 α 就是为了平衡这两者. 在本文中, 为了更方便实现, 将温度因子设置一个固定的超参数.

2.4 噪声自适应控制

在进化强化学习方法中, 噪声标准差 σ 是一个对方法性能相当重要超参数. 根据前面的进化强化

学习方法可以看出, 这一超参数在整个方法中发挥了 2 个作用: 1) 在生成噪声扰动的时候, 扰动量 ϵ 是由高斯分布 $\mathrm{N}(0, \sigma^2)$ 产生的, 因此 σ 的大小决定了噪声生成的范围, 反映到方法中就相当于每次迭代中生成新一批网络的搜索范围, 所以本文一般习惯称 σ 为噪声标准差; 2) 在网络参数的更新公式中, σ 还起到了修正学习率 lr 的作用:

$$lr' = \frac{lr}{\sigma} \tag{14}$$

这样做的目的相当于是为方法的稳定性提供了一定的反馈调节, 当方法的 σ 过大时, 每次迭代的探索范围就会比较大, 那这时就需要一个较小的学习率; 反之当方法的 σ 过小时, 那么学习率就相应取一个较大的值.

但是, 实验表明进化强化学习对于 σ 的变化仍然比较敏感, 随着初始的 σ 的改变, 方法的学习性能会有比较明显的差异. 而在实际应用中, 就必须通过反复实验不断调参来找到在当前环境下使进化强化学习性能最优的 σ, 这样的做法大大增加了进化强化学习准备工作的工作量. 本文希望找到一种方法能降低进化强化学习对于初始 σ 设置的敏感性, 减小方法对于超参数相关的先验知识的依赖性.

许多研究者在机器学习问题中引入了自适应控制方法[36−37], 这些方法包括参数空间搜索[38], 启发式函数[39], 状态估计反馈[40] 等, 根据这些方法提出了一种基于 KL 散度的对 σ 进行自适应控制的机制, 这一机制在主线程完成一次迭代更新之后, 能够根据当前的结果来智能地修正 σ 的值, 通过计算迭代前后两个网络的 KL 散度来决定是增大还是减小 σ, 最终将噪声调整到能够使方法学习性能更好的范围之内, 同时也为下一次实验提供了可行的调参方向. 具体来说 σ 的更新公式如下:

$$\sigma_{k+1} = \begin{cases} K\sigma_k, & D_{KL}(\pi_k||\pi_{k+1}) \geq \delta \\ \dfrac{1}{K}\sigma_k, & D_{KL}(\pi_k||\pi_{k+1}) < \delta \end{cases} \tag{15}$$

式中, $D_{KL}(\pi_k||\pi_{k+1})$ 表示迭代前后两个网络 π_k 和 π_{k+1} 的 KL 散度. K 为噪声自适应控制的系数, 本文 K 取值为 1.01.

噪声自适应控制机制能够发挥作用的关键在于找到一种合适的方法来估计更新前后两个策略网络的距离. 在这一机制中选择 KL 散度作为距离度量, 一方面是因为 KL 散度是所有概率分布距离度量方法中计算量和复杂度较小的方法, 不会对方法整体性能造成过大的影响; 另一方面也是参考了深度强化学习领域一些前沿方法的经验. KL 散度又被称为相对熵, 它可以将两个策略之间的差异转换为计算这两个策略生成的概率分布具有的信息熵的差异, 对于两个离散概率分布 P 和 Q, 它们之间的 KL 散度被定义为:

$$D_{KL}(P||Q) = \sum P(x)\log\frac{P(x)}{Q(x)} \tag{16}$$

这一方法大大降低了距离度量计算的复杂度, 并且实现起来相对容易, 因此 KL 散度被广泛应用于机器学习领域中, 很多机器学习任务的损失函数都选择使用 KL 散度来估计预测分布与真实分布的差异. 而一些前沿的深度强化学习方法, 比如信赖域策略优化[41] 和 PPO[16] 都运用了 KL 散度来将网络的更新限制在一定范围之内. 这些研究成果表明, 通过 KL 散度来估计更新前后网络的距离从而实现对 σ 进行控制是可行的.

本文依据式 (16) 给出 $D_{KL}(\pi_k||\pi_{k+1})$ 的具体计算式, 当智能体处于某一状态 s_i 时, 它根据策略 π 选择执行动作 a_j 的概率可以表示为 $\pi(a_j|s_i)$, 则 $D_{KL}(\pi_k||\pi_{k+1})$ 在状态 s_i 的公式为:

$$D_{KL}(\pi_k||\pi_{k+1})|_{s=s_i} = \sum_{j=1}^{m}\pi_k(a_j|s_i)\log\frac{\pi_{k+1}(a_j|s_i)}{\pi_k(a_j|s_i)} \tag{17}$$

为了尽可能的准确的计算出两个网络在整个环境下的 KL 散度, 本文仿照第 3 节中最大熵搜索的方法, 利用一个完整的片段进行估计. 在实际操作中, 方法在旧网络 π_k 上进行一次蒙特卡洛抽样, 记录该片段经过的所有状态 $s_0, s_1, \cdots, s_n$, 结合式 (17) 就可以估计 KL 散度为:

$$D_{KL}(\pi_k||\pi_{k+1}) = \frac{1}{n}\sum_{i=0}^{n}D_{KL}(\pi_k||\pi_{k+1})|_{s=s_i} = \frac{1}{n}\sum_{i=0}^{n}\sum_{j=1}^{m}\pi_k(a_j|s_i)\log\frac{\pi_{k+1}(a_j|s_i)}{\pi_k(a_j|s_i)} \tag{18}$$

式 (15) 本身是一个 “正反馈”, 当 KL 散度大于阈值 δ 时方法会增大 σ. 但根据前文所述, σ 会起到控制探索范围和调整学习率两个作用, 所以它对 KL 散度的影响同样是这两个方面协同作用的结果. 通过实验发现, 从整个方法来看, 噪声标准差与 KL 散度是 “负相关”的关系, 即 σ 越大, KL 散度反而越小. 根据这个结论, 本文提出的噪声自适应控制机制实际上仍然是一种 “负反馈”. 这一机制的作用与自适应学习率是相似的, 但是自适应学习率只能对 lr 进行修改, 而通过改变 σ 还能够在调整学习率的同时改变探索范围, 从而能够进一步提升方法的鲁棒性.

在进一步的实验中也发现由于阈值 δ 的选取不

当可能导致方法会过度地调整 σ. 为了尽可能避免这一情况的发生, 为噪声的调整范围设定了上界和下界, 分别为初始噪声标准差 σ_0 的 2 倍和 1/2:

$$\sigma_{k+1}=\begin{cases}K\sigma_k, & D_{KL}\geq\delta \ \text{或} \ \sigma_k<0.5\sigma_0\\ \dfrac{1}{K}\sigma_k, & D_{KL}<\delta \ \text{或} \ \sigma_k\geq 2\sigma_0\end{cases} \tag{19}$$

另外本文还为噪声自适应控制机制增加了一个终止条件—未扰动网络的目标函数 $J_e(\pi_\theta)$ 在种群中排名第 1. 当运行结果满足这一条件时, 基于 KL 散度的噪声自适应控制机制就被终止并且方法保持在当前噪声下运行直到重新不满足该条件, 这样做是为了防止方法早探索到最优解之后由于继续调整噪声标准差可能导致的结果不稳定. 通过添加上述两个终止条件, 减弱了由于阈值 δ 的不同取值对实验结果的影响, 因此在实验中没有必要刻意追求最为合理的 δ 取值, 本文在后续实验中均选取 $\delta=1$.

根据上述对传统进化强化学习 3 个方面的改进与设计, 本节给出 AERL-ME 的整体方法伪代码, 见算法 1.

算法 1. AERL-ME 方法的伪代码

输入. 初始噪声标准差 σ_0, 种群数量 n (必须是偶数), 学习率 lr, 衰减因子 γ, 温度系数 α, 阈值 δ 以及其他一些必要的超参数.

初始化. n 个 CPU 子线程.

1) for 主线程中第 k 轮迭代, $k=1, 2, \cdots$.

2) 生成 $n/2$ 个随机种子.

3) 噪声采样 $\epsilon_{2i-1}\sim \mathrm{N}(0, \sigma_{k-1}^2)$, $i=1, 2, \cdots, n/2$.

4) 生成负向噪声 $\epsilon_{2i}=-\epsilon_{2i-1}$, $i=1, 2, \cdots, n/2$.

5) 向子线程传递噪声和对应的随机种子.

6) for 第 j 个子线程, $j=1, 2, \cdots, n$.

7) 根据接收到的噪声生成新的实例.

8) 进行一轮蒙特卡洛抽样.

9) 按式 (13) 计算 $J_e(\pi_{\theta+\epsilon_j})$.

10) 向主线程回传本实例的 $J_e(\pi_{\theta+\epsilon_j})$.

11) end for.

12) 接收回传的所有 $J_e(\pi_{\theta+\epsilon_j})$ 并排序.

13) 根据式 (8) 计算每个子代实例的权重 $f(\theta+\epsilon)$.

14) 进行一轮蒙特卡洛抽样, 记录经过的状态.

15) 按式 (13) 计算未扰动网络的目标函数 $J_e(\pi_\theta)$.

16) 比较 $J_e(\pi_\theta)$ 与 $\max\left[J_e(\pi_{\theta+\epsilon})\right]$.

17) 根据式 (7) 更新 θ.

18) 根据式 (18) 计算 $D_{KL}(\pi_k||\pi_{k+1})$.

19) 根据式 (19) 和终止条件计算 σ_k.

20) end for.

在一轮迭代开始时, 主线程随机生成 $n/2$ 个随机种子 (步骤 2), 根据当前的噪声标准差通过对偶抽样法得到 n 个两两相反的噪声 (步骤 3 ~ 4); 接着将噪声及其对应的随机种子一一发送给各个子线程, 这样做就确保了主线程能够获知子线程的扰动信息 (步骤 5).

每一个子线程在接收到主线程发来的信息后生成新的子代实例, 然后在该实例上通过蒙特卡洛抽样形成一个学习片段 $\tau=(s_0, a_0, s_1, \cdots)$ (步骤 7 ~ 8), 之后根据第 3 节最大熵搜索中式 (13) 计算该实例目标函数的估计值并回传给主线程 (步骤 9 ~ 10). 这一部分通过实现并行化处理, 所有子线程能够同时进行采样和计算, 从而大大提高了运算效率.

当主线程收到所有子线程的目标函数值之后, 首先需要对这些值进行排序 (步骤 12), 再根据第 2 节中的 (μ, λ)-ES, 归一化算法和更新公式对主线程的网络参数进行更新 (步骤 13, 17). 而在网络更新前, 为了自适应控制机制中相关条件的判断, 还需要在旧网络上提前生成一个学习片段 (步骤 14). 在网络更新后再根据这个片段估计新旧两个网络的 KL 散度和确认终止条件 (步骤 15 ~ 16, 18), 并根据结果计算新的噪声标准差 (步骤 19).

3 实验

为了测试 AERL-ME 的方法性能并证明其相对于其基准方法的性能优势, 本文利用 Python 的 Gym 包中提供的多种模拟环境进行了实验. 在第 3.1 节简单介绍本次实验所用到的所有模拟环境的信息. 本文设计了 3 个实验来解答对 AERL-ME 方法的性能所提出的 3 个疑问:

问题 1. AERL-ME 较之当前强化学习领域前沿的方法, 尤其是其基准方法进化强化学习是否有性能上的提升?

问题 2. AERL-ME 的 3 个贡献点是否都能够从不同角度提升整体方法的性能, 它们各自对方法性能起到了何种作用?

问题 3. AERL-ME 对于主要超参数变化的敏感程度如何?

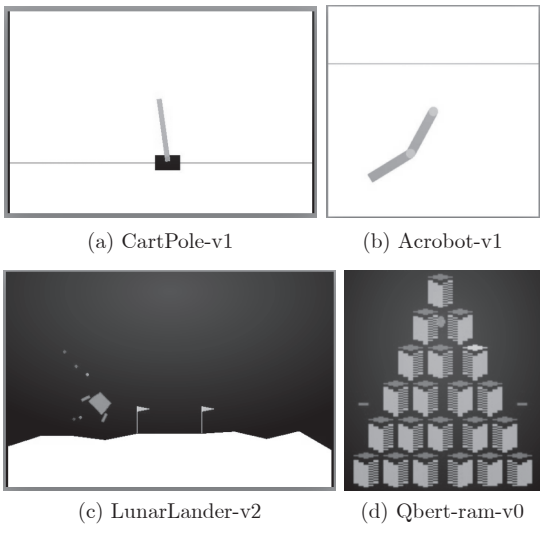
在实验的初始设置上, 由于不同方法的结构存在差异, 除了作为基准的进化强化学习方法之外, 对于其它方法只能尽可能使得实验在完全相同的条件下进行. 比如在无梯度优化方法和策略梯度方法中, 都能应用多线程来加速学习性能, 所以在这些方法下都使用了相同的线程数, 并且迭代相同的次数. 而对于无法应用多线程的方法, 则只能在输出数据上做一些整理, 确保在同一评估点上, 各个方法经历了相同的训练次数, 以此确保实验结果的相

对公平. 而作为对比实验的关键, AERL-ME 与其基准方法使用了完全一致的公共超参数. 此外, 为了防止实验的偶然性, 对每种方法在每个环境下以不同的随机种子为初始条件进行了 10 次完全随机的实验, 在后续数据处理中通过取平均绘制学习曲线. 而实验的数值结果也以表格的形式给出附在学习曲线之后, 以此更为清晰地展现不同方法学习结果的差异.

3.1　实验环境简介

Gym 是由 OpenAI 开发的基于 Python 语言的开源代码库. 被广泛应用于训练和比较深度强化学习方法的性能. 在 Gym 包中内置了众多模拟环境, 包括经典的控制模型 (如倒立摆、爬山小车), Atrai 2 600 像素游戏, Mujoco 仿真环境和一些简单的机械臂模型. 实验采用了 Gym 包中的 CartPole-v1、Acrobot-v1、LunarLander-v2 和 Qbert-ram-v0 共 4 个环境.

1) CartPole-v1: CartPole-v1 (CP) 是一个经典的倒立摆模型, 如图 2(a) 所示, 一根杆连接在小车上, 而小车在光滑的水平面上. 系统通过对小车施加正向或负向的力来进行控制, 杆每保持一个时间单位的直立就获得一分, 而当杆偏离竖直的角度或是小车距起始点的距离超过了一定范围, 当次实验就结束. 最大的时间单位为 500, 即该模型的最大得分也为 500.



(a) CartPole-v1　(b) Acrobot-v1

(c) LunarLander-v2　(d) Qbert-ram-v0

图 2　实验环境

Fig. 2　Experimental environments

2) Acrobot-v1: Acrobot-v1 (AB) 如图 2(b) 所示, 该系统包括两个关节和两个连杆, 其中两个连杆之间的关节被驱动. 最初, 连杆是向下悬挂的, 目标是将较低连杆的末端向上摆动到给定的高度.

3) LunarLander-v2: LunarLander-v2 (LL) 是一个登月模型, 如图 2(c) 所示, 登月器从屏幕顶部开始移动, 最终以零速度到着陆台的奖励大约是在 100 到 140 分之间, 坠毁或成功着陆都将停止学习并分别获得额外的 −100 或 +100 分, 每条腿着陆 +10 分, 每次发动引擎 −0.3 分, 一般将实验成功的标准定为 200 分.

4) Qbert-ram-v0: Qbert-ram-v0 (Qbert) 是一个雅达利 2 600 像素游戏, 如图 2(d) 所示, 玩家控制主角在一个由正方体构成的三角立面上来回跳跃, 每一次地面接触都会改变方块表层的颜色, 只要将所有色块踩成规定的颜色即告胜利. 状态观测选用雅达利的随机存储器 (Random access memory, RAM) 状态.

3.2　对比实验结果

本文将 AERL-ME 与以下方法进行了对比:

1) ES[20]: 进化强化学习的基准方法, 通过随机生成子代进行网络更新.

2) 深度 Q 学习 (Deep Q-learning, DQN)[42]: 基于值函数的强化学习经典方法, 通过训练深度神经网络来拟合状态动作价值函数, 并根据该网络依贪心策略选取最优动作.

3) PPO[16]: 策略梯度方法中的前沿方法之一, 能够通过重要性采样利用离线数据进行学习. 并通过计算行为策略与当前策略的 KL 散度防止迭代前后两个策略差距较大.

上面这 3 种方法分别代表了无梯度优化方法和基于价值的强化学习方法和策略梯度强化学习方法. 通过这些对比, 可以更为全面地展示 AERL-ME 与当前深度强化学习主流方法之间的性能优劣. 实验结果如图 3 所示.

通过实验数据整理的部分数值结果表 1 所示. 在此表中, 给出了 4 种方法在终止评估点上 10 次实验结果的平均回报, 而置信区间是由相应的标准误差给出. 最佳性能 (以均值作为判别依据) 以粗体的形式给出.

根据图 3 和表 1 的对比实验结果, 可以得到以下结论:

1) 对 AERL-ME 与其基准方法 ES 之间的性能差异进行比较. 总结来看, AERL-ME 的性能优势在简单环境和复杂环境下的表现并不一样的. 在一些简单环境下, 环境基本没有太多局部最优点, 此时大部分强化学习方法在一定时间之后都能学习到最优的策略. 而 AERL-ME 的优势在于它相对于

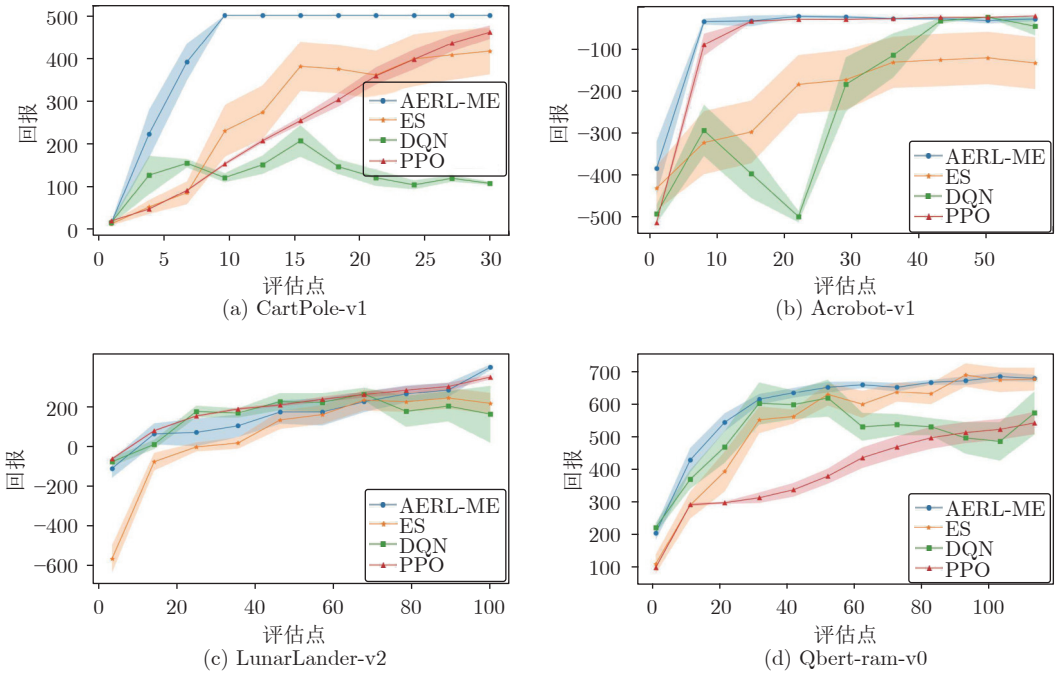


图 3 对比实验结果

Fig. 3 Comparative experimental results

表 1 以平均回报表示的数值结果

Table 1 The numerical results in terms of average received returns

环境	AERL-ME	ES	DQN	PPO
CP	**500.0 ± 0.0**	416.1 ± 54.0	108.3 ± 3.6	460.4 ± 15.8
AB	−83.9 ± 4.5	−173.7 ± 52.9	−98.6 ± 18.5	**−77.8 ± 0.5**
LL	**245.0 ± 11.8**	82.8 ± 47.8	35.2 ± 126.9	201.2 ± 10.4
Qbert	**677.5 ± 10.3**	675.0 ± 34.6	571.0 ± 65.9	540.4 ± 34.0

基准方法有着更快的收敛速度. 比如在 CartPole-v1 环境中, AERL-ME 用了不到 10 次迭代就找到了最优的策略, 而 ES 在 30 次迭代之后仍然没有收敛到最优策略. 而在复杂环境下, 包括传统 ES 在内的一些基于策略的方法容易陷入局部最优, 而 AERL-ME 能够在这一情况下更为有效地搜索到更好的策略, 方法规避局部最优的能力得到了一定的加强.

2) 将 AERL-ME 与策略梯度的代表方法 PPO 进行比较. 从网络输出来看, 进化强化学习方法与策略梯度是一致的, 均是根据当前状态输出一个动作的概率分布列. PPO 方法的最大优势就在它的学习曲线相对来说较为稳定, 在各个环境的实验全过程中都没有出现明显的振荡. 实验结果表明 AERL-ME 较之 PPO 有两个优势: a) 在大部分环境下, AERL-ME 的的学习性能都优于 PPO, 并且 AERL-ME 的前期收敛速度要比 PPO 快, 尤其是在一些较为简单的环境下表现得更为明显. b) 即使 PPO 在多轮迭代后也能获得一个不错的学习结果, 但 PPO 这样的梯度优化方法为了学习到这样的结果较之 AERL-ME 需要更多的算力要求和运算时间. 为了展现 AERL-ME 在运算时间上的优势, 本文在 Qbert 的实验过程中记录了 4 种方法的运算时间并根据实验结果绘制了图 4. 其中, 进化策略类算法 AERL-ME 以及其基准算法都使用了并行化处理, 在每一次迭代过程中使用了 40 个子线程进行并行运算. 在同样的实验条件下, AERL-ME 的时间花费不到 PPO 的 1/3, 更是远低于 DQN 的时间花费. 尽管需要进行额外的蒙特卡洛抽样以进行噪声的自适应控制, AERL-ME 的学习时间会略高于基准算法, 但进化强化学习方法的高并行化效率仍然使其能够在相当短的时间内完成策略的学习, 这

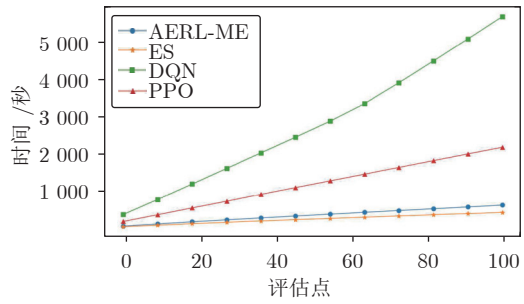


图 4 运算时间对比 ($n = 40$)

Fig. 4 Comparison of operation time ($n = 40$)

是梯度优化算法难以做到的.

3) 将 AERL-ME 与基于值函数方法的代表 DQN 进行比较, 以价值为基础一类的方法尽管在复杂环境下能够更好的规避局部最优, 但它的最大缺点就是收敛性较弱而且学习结果不稳定, 但它前期的收敛速度远远慢于其他三种方法. 进化强化学习本身的短板就是探索性, 然而通过引入最大熵模型, AERL-ME 一定程度上弥补了原有方法在规避局部最优上的缺陷, 从结果来看在复杂环境下反而能比 DQN 学到更好的策略.

3.3　消融实验结果

为了进一步研究方法中各个部分对方法性能的贡献, 本文还设计了消融实验, 实验中每次使用减少一个贡献点的方法与完整方法在相同环境和初始设置下进行对比实验, 实验环境为 CartPole-v1, 实验结果如图 5 所示. 为了凸显各个方法之间上升趋势的差异, 主要选取了方法从开始学习到收敛到一个相对稳定的策略这一过程内的数据作图. 对于不同的初始条件, 这一过程的长度是不一致的, 因此图 5 中子图的横坐标范围各不相同.

通过消融实验结果可以看出 3 个贡献各自对原有方法性能起到的优化作用:

1) 进化策略的改进方法 (Canonical evolution strategies, CES), 它主要是加快了方法在运算开始时的收敛速度, 实验结果表明这一优化方法必须要在初始参数设置 (尤其是 σ_0) 较为合适的情况下才能发挥出更大的作用, 否则反而可能因为搜索方向的不准确影响方法的性能.

2) 最大熵搜索 (Maximum entropy, ME), 它的作用是提升方法对新策略的探索能力, 避免陷入局部最优, 所以在一些复杂环境下或是当初始设置的 σ_0 偏小导致进化策略方法本身的探索性能不足时, ME 能够一定程度上弥补这一缺陷.

3) 噪声自适应控制 (Adaptive noise, AN), 在正常情况下 AN 不会发挥明显的作用, 但它的价值在于当初始 σ_0 的设置较为不合理尤其是偏大时, 能够帮助方法缓慢地将这个超参数调整到一个更为合理的值, 从而提高学习性能, 这一点从图 5(a)、(d) 两个相对极端的 σ_0 的学习曲线能够看出. 总之, AERL-ME 的三个贡献点对于传统方法各有针对性的优化, 三者是相辅相成, 缺一不可的.

3.4　超参数灵敏度分析

在最后一部分, 本文对 AERL-ME 中的最关键的两个超参数初始噪声标准差 σ_0 和温度因子 α 进行了灵敏度分析实验, 实验环境为 CartPole-v1, 实验结果如图 6 所示.

尽管自适应机制一定程度上使得 AERL-ME 较之其基准方法对于超参数的敏感性已经有所下降, 但是图 6(a) 的学习曲线表明, 初始噪声标准差 σ_0 的设置仍然对方法的性能有着较大的影响.

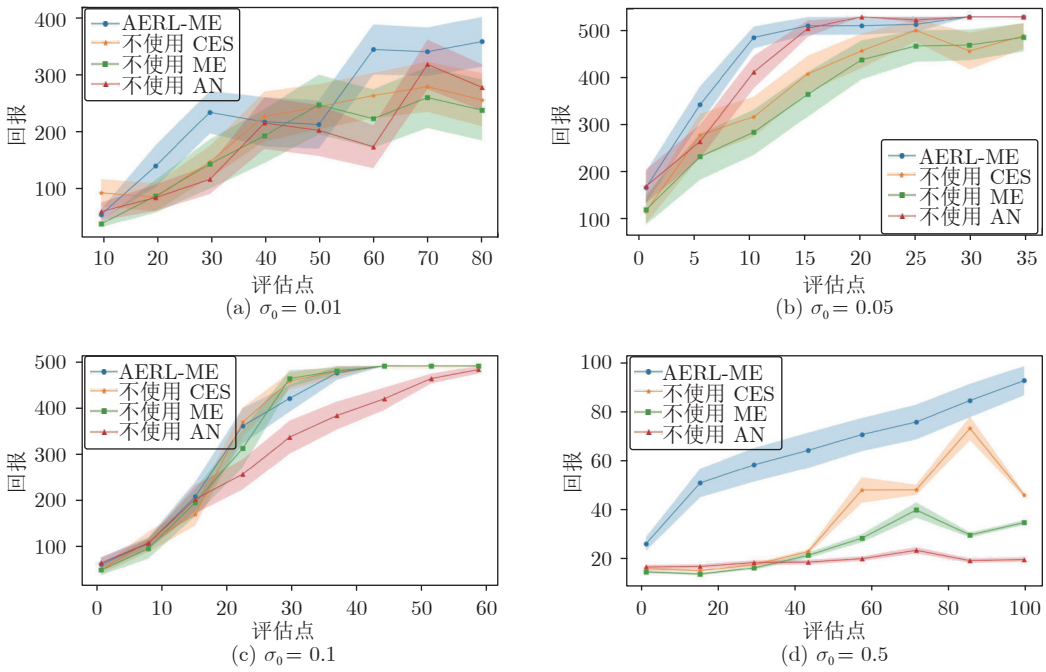
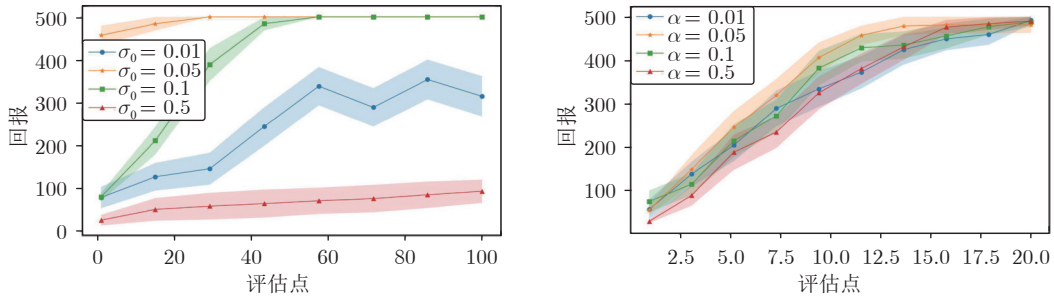


图 5　消融实验结果

Fig. 5　Ablation experimental results



(a) 初始噪声标准差 σ_0
(a) Initial noise standard deviation σ_0

(b) 温度因子 α
(b) Temperature factor α

图 6 初始噪声标准差 σ_0 和温度因子 α 的灵敏度分析

Fig. 6 Sensitivity analysis of initial noise standard deviation σ_0 and temperature factor α

初始噪声标准差 σ_0 的变化主要影响了方法的收敛性和收敛速度, 当 σ_0 取值在较为合适的范围之内时, 方法能够实现快速稳定的收敛; 当 σ_0 的取值与最优区间有了一定偏差之后, 方法的收敛速度有一个相当明显的下降, 并且在前期出现了一些微小的振荡. 而当 σ_0 的取值与最优区间的偏差过大时, 方法将不收敛. 如果 σ_0 取值过大, 根据式 (12), 方法的学习率过小, 网络基本没有明显的更新; 而如果 σ_0 取值过小, 则学习率反而过大但探索范围小, 方法有可能选择错误的下降方向, 导致学习曲线出现振荡.

在实际应用中, 如果事先不确定初始噪声标准差 σ_0 的合理区间, 可以先随机取一个值进行测试, 由于自适应噪声机制, 方法在运行过程中会逐渐将噪声调整到更接近合理区间的位置. 在测试结束后就可以通过实验结果进一步调整 σ_0, 这一定程度上降低了调参的难度与工作量.

与初始噪声标准差 σ_0 相比, 方法对于温度因子 α 的变化不敏感. 当温度因子 α 取值过大或过小时, 主要是轻微影响了方法的前期收敛速度, 但对整体的收敛趋势没有太明显的影响. 所以在实际应用中, 没有必要刻意追求最为合理的 α 取值.

4 结束语

本文提出了一种对基于自适应噪声的最大熵进化强化学习方法 AERL-ME. 首先, 该方法引入了一种进化策略的改进方法 CES 削弱了进化方向错误的子代实例对网络更新造成的影响; 其次, 通过在方法的目标函数中添加熵正则化项, 有效地平衡了方法的探索与利用的能力, 使方法能够跳出局部最优; 最后, 该方法提出了一种对噪声标准差的自适应控制机制, 能够在网络更新后根据 KL 散度智能地调整方法的探索范围. 实验结果表明, 这一方法在继承了进化强化学习的高效率, 低运算成本和高并行化等优点的同时, 又一定程度上解决了传统方法收敛速度慢, 探索性不足和依赖先验知识而导致的鲁棒性较弱的缺点.

未来可以改进的方向是: 1) 本文方法只能在离散状态—动作空间的模型下进行学习. 如果需要将之拓展连续空间, 就必须要修改方法中的一些公式, 比如信息熵就需要改成积分的形式. 2) 本文方法在面对具有稀疏奖励模型时, 比如在 Gym 中的爬山小车环境中, 会由于在初始网络周围搜索不到更优的策略而导致在多步迭代之后仍然无法实现网络的更新. 3) 可以探索能够进一步降低方法对于超参数初始设置敏感性的自适应控制机制.

References

1 Sutton R S, Barto A G. *Reinforcement Learning: An Introduction (2nd edition)*. Cambridge: MIT Press, 2018.

2 Li H, Zhang Q, Zhao D. Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE Transactions on Neural Networks and Learning Systems*, 2019, **31**(6): 2064–2076

3 Li D, Zhao D, Zhang Q, Chen Y. Reinforcement learning and deep learning based lateral control for autonomous driving. *IEEE Computational Intelligence Magazine*, 2019, **14**(2): 83–98

4 Yang W, Shi Y, Gao Y, Yang M. Online multi-view subspace learning via group structure analysis for visual object tracking. *Distributed and Parallel Databases*, 2018, **36**(3): 485–509

5 Luo B, Liu D, Huang T, Wang D. Model-free optimal tracking control via critic-only Q-learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, **27**(10): 2134–2144

6 Yao Hong-Ge, Zhang Wei, Yang Hao-Qi, Yu Jun. Joint regression object localization based on deep reinforcement learning. *Acta Automatica Sinica*, 2020, **41**: 1–10
(姚红革, 张玮, 杨浩琪, 喻钧. 深度强化学习联合回归目标定位. 自动化学报, 2020, **41**: 1–10)

7 Zhang Z, Zhao D, Gao J, Wang D. FMRQ: A multiagent reinforcement learning algorithm for fully cooperative tasks. *IEEE Transactions on Cybernetics*, 2016, **47**(6): 1367–1379

8 Li Kai-Wen, Zhang Tao, Wang Rui, Qin Wei-Jian, He Hui-Hui, Huang Hong. Research reviews of combinatorial optimization methods based on deep reinforcement learning. *Acta Automatica Sinica*, 2021, **47**(11): 2521–2537
(李凯文, 张涛, 王锐, 覃伟健, 贺惠晖, 黄鸿. 基于深度强化学习的组合优化研究进展. 自动化学报, 2021, **47**(11): 2521–2537)

9 Wang Yun-Peng, Guo Ge. Signal priority control for trams using deep reinforcement learning. *Acta Automatica Sinica*, 2019, **45**(12): 2366−2377
(王云鹏, 郭戈. 基于深度强化学习的有轨电车信号优先控制. 自动化学报, 2019, **45**(12): 2366−2377)

10 Wu Xiao-Guang, Liu Shao-Wei, Yang Lei, Deng Wen-Qiang, Jia Zhe-Heng. A gait control method for biped robot on slope based on deep reinforcement learning. *Acta Automatica Sinica*, 2021, **47**(8): 1976−1987
(吴晓光, 刘绍维, 杨磊, 邓文强, 贾哲恒. 基于深度强化学习的双足机器人斜坡步态控制方法. 自动化学报, 2021, **47**(8): 1976−1987)

11 Zhao Dong-Bin, Shao Kun, Zhu Yuan-Heng, Li Dong, Chen Ya-Ran, Wang Hai-Tao, et al. Review of deep reinforcement learning and discussions on the development of computer Go. *Control Theory and Applications*, 2016, **33**(6): 701−717
(赵冬斌, 邵坤, 朱圆恒, 李栋, 陈亚冉, 王海涛, 等. 深度强化学习综述: 兼论计算机围棋的发展. 控制理论与应用, 2016, **33**(6): 701−717)

12 Liu Jian-Wei, Gao Feng, Luo Xiong-Lin. Survey of Deep Reinforcement Learning Based on Value Function and Policy Gradient. *Chinese Journal of Computers*, 2019, **42**(6): 1406−1438
(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)

13 Xu X, Hu D, Lu X. Kernel-based least squares policy iteration for reinforcement learning. *IEEE Transactions on Neural Networks*, 2007, **18**(4): 973−992

14 Zhu Y, Zhao D, Yang X, Zhang Q. Policy iteration for H_∞ optimal control of polynomial nonlinear systems via sum of squares programming. *IEEE Transactions on Cybernetics*, 2017, **48**(2): 500−509

15 Liu D, Wu J, Xu X. Multi-agent reinforcement learning using ordinal action selection and approximate policy iteration. *International Journal of Wavelets, Multiresolution and Information Processing*, 2016, **14**(6): 1650053

16 Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms [Online], available: https://arxiv.org/abs/1707.06347v2, August 28, 2017

17 Gu Y, Cheng Y, Chen C L P, Wang X. Proximal Policy Optimization With Policy Feedback. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, **52**(7): 4600−4610

18 Cheng Y, Huang L, Wang X. Authentic boundary proximal policy optimization. *IEEE Transactions on Cybernetics*, 2022, **52**(9): 9428−9438

19 Wang X, Li T, Cheng Y. Proximal parameter distribution optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, **51**(6): 3771−3780

20 Salimans T, Ho J, Chen X, Sidor S, Sutskever I. Evolution strategies as a scalable alternative to reinforcement learning [Online], available: https://arxiv.org/abs/1703.03864v2, September 7, 2017

21 Such F P, Madhavan V, Conti E, Lehman J, Stanley K O, Clune J. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning [Online], available: https://arxiv.org/abs/1712.06567v3, April 20, 2018

22 Lehman J, Chen J, Clune J, Stanley K O. Es is more than just a traditional finite difference approximator. In: Proceedings of the Genetic and Evolutionary Computation Conference. Kyoto, Japan, 2018. 450−457

23 Zhang X, Clune J, Stanley K O. On the relationship between the openai evolution strategy and stochastic gradient descent [Online], available: https://arxiv.org/abs/1712.06564, December 18, 2017

24 Khadka S, Tumer K. Evolution-guided policy gradient in reinforcement learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, Canada, 2018. 1196−1208

25 Shi L, Li S, Zheng Q, Cao L, Yang L, Pan G. Maximum entropy reinforcement learning with evolution strategies. In: Proceedings of the International Joint Conference on Neural Networks. Glasgow, United Kingdom: IEEE, 2020. 1−8

26 Song X, Gao W, Yang Y, Choromanski K, Pacchiano A, Tang Y. Es-maml: Simple hessian-free meta learning. In: Proceedings of the 8th International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020

27 Majumdar S, Khadka S, Miret S, Mcaleer S, Tumer K. Evolutionary reinforcement learning for sample-efficient multiagent coordination. In: Proceedings of the 37th International Conference on Machine Learning. New York, USA: 2020. 6651−6660

28 Long Q, Zhou Z, Gupta A, Fang F, Wu Y, Wang X. Evolutionary population curriculum for scaling multi-agent reinforcement learning. In: Proceedings of the 8th International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020

29 Wang Z, Chen C, Dong D. Instance weighted incremental evolution strategies for reinforcement learning in dynamic environments [Online], available: https://arxiv.org/abs/2010.04605v2, March 31, 2022

30 Wierstra D, Schaul T, Glasmachers T, Sun Y, Peters J, Schmidhuber J. Natural evolution strategies. *The Journal of Machine Learning Research*, 2014, **15**(1): 949−980

31 Conti E, Madhavan V, Such F P, Lehman J, Stanley K, Clune J. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, Canada, 2018. 5032−5043

32 Lehman J, Chen J, Clune J, Stanley K O. Safe mutations for deep and recurrent neural networks through output gradients. In: Proceedings of the Genetic and Evolutionary Computation Conference. Kyoto, Japan, 2018. 117−124

33 Choromanski K, Pacchiano A, Parker-Holder J, Tang Y, Jain D Yang Y, et al. Provably robust blackbox optimization for reinforcement learning. In: Proceedings of the Conference on Robot Learning. New York, USA: 2020. 683−696

34 Chrabaszcz P, Loshchilov I, Hutter F. Back to basics: Benchmarking canonical evolution strategies for playing Atari. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm, Sweden, 2018. 1419−1426

35 Haarnoja T, Zhou A, Hartikainen K, Tucker g, Ha S, Tan J, et al. Soft actor-critic algorithms and applications [Online], available: https://arxiv.org/abs/1812.05905v2, January 29, 2019

36 Zhang Hua-Xiang, Lu Jing. An adaptive evolutionary programming algorithm based on Q learning. *Acta Automatica Sinica*, 2008, **34**(7): 819−822
(张化祥, 陆晶. 基于Q学习的适应性进化规划算法. 自动化学报, 2008, **34**(7): 819−822)

37 Zhu Y, Zhao D, Li X. Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, **28**(3): 714−725

38 Plappert M, Houthooft R, Dhariwal P, Sidor S, Chen R Y, Chen X, et al. Parameter space noise for exploration [Online], available: https://arxiv.org/abs/1706.01905v2, January 31, 2018

39 Naumov M, Blagov A. Development of the heuristic method of evolutionary strategies for reinforcement learning problems solving. In: Proceedings of the 6th International Conference on Information Technology and Nanotechnology. Samara, Russia, 2020. 19−22

40 Wang Liu-Jing, Zhang Gui-Jun, Zhou Xiao-Gen. Strategy self-adaptive differential evolution algorithm based on state estimation feedback. *Acta Automatica Sinica*, 2020, **46**(4): 752−766
(王柳静, 张贵军, 周晓根. 基于状态估计反馈的策略自适应差分进化算法. 自动化学报, 2020, **46**(4): 752−766)

41 Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust re-

gion policy optimization. In: Proceedings of the 32th International Conference on Machine Learning. Lille, France: 2015. 1889−1897

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

王君逸 南京大学控制科学与智能工程系硕士研究生. 2021 年获南京大学学士学位. 主要研究方向为强化学习, 机器学习与人工智能.
E-mail: mf21150062@smail.nju.edu.cn
(**WANG Jun-Yi** Master student in the Department of Control Science and Intelligence Engineering, Nanjing University. He received his bachelor degree from Nanjing University in 2021. His research interest covers reinforcement learning, machine learning, and artificial intelligence.)

王 志 南京大学控制科学与智能工程系讲师. 2015 年获南京大学学士学位. 2019 年获中国香港城市大学博士学位. 主要研究方向为强化学习, 机器学习与人工智能. 本文通信作者.
E-mail: zhiwang@nju.edu.cn
(**WANG Zhi** Lecturer in the Department of Control Science and Intelligence Engineering, Nanjing University. He received his bachelor degree from Nanjing University in 2015. He received his Ph.D. degree from City University of Hong Kong, China, in 2019. His research interest covers reinforcement learning, machine learning, and artificial intelligence. Corresponding author of this paper.)

李华雄 南京大学控制科学与智能工程系副教授. 2009 年获南京大学博士学位. 主要研究方向为机器学习与数据挖掘, 模式识别与智能系统.
E-mail: huaxiongli@nju.edu.cn
(**LI Hua-Xiong** Associate professor in the Department of Control Science and Intelligence Engineering, Nanjing University. He received his Ph.D. degree from Nanjing University in 2009. His research interest covers machine learning, and data mining, pattern recognition, and intelligent systems.)

陈春林 南京大学控制科学与智能工程系教授. 分别于 2001 年和 2006 年获中国科学技术大学学士学位和博士学位. 主要研究方向为强化学习及智能无人系统.
E-mail: clchen@nju.edu.cn
(**CHEN Chun-Lin** Professor in the Department of Control Science and Intelligence Engineering, Nanjing University. He received his bachelor and the Ph.D. degrees from University of Science and Technology of China in 2001 and 2006, respectively. His research interest covers reinforcement learning and intelligent unmanned systems.)