

基于自适应调节策略熵的元强化学习算法



陆嘉猷¹ 凌兴宏^{1,2} 刘全¹ 朱斐¹

1 苏州大学计算机科学与技术学院 江苏 苏州 215006

2 苏州大学文正学院 江苏 苏州 215104

(15261868763@163.com)

摘要 传统的深度强化学习方法依赖大量的经验样本并且难以适应新任务。元强化学习通过从以往的训练任务中提取先验知识,为智能体快速适应新任务提供了一种有效的方法。基于最大熵强化学习框架的元深度强化学习通过最大化期望奖励和最大化策略熵来优化策略。然而,目前以最大熵强化学习框架为基础的元强化学习算法普遍采用固定的温度参数,这在面对元强化学习的多任务场景时是不合理的。针对这一问题,提出了自适应调节策略熵(Automating Policy Entropy, APE)算法。该算法首先通过限制策略的熵,将原本的目标函数优化问题转换为受限优化问题,然后将受限优化问题中的对偶变量作为温度参数,通过拉格朗日对偶法求解得到其更新公式。根据得到的更新公式,温度参数将在每一轮元训练结束之后进行自适应调节。实验数据表明,所提算法在 Ant-Fwd-Back 和 Walker-2D 上的平均得分提高了 200,元训练效率提升了 82%;在 Humanoid-Direct-2D 上的策略收敛所需的训练步数为 23 万,收敛速度提升了 127%。实验结果表明,所提算法具有更高的元训练效率和更好的稳定性。

关键词: 元学习;强化学习;最大熵

中图分类号 TP181

Meta-reinforcement Learning Algorithm Based on Automating Policy Entropy

LU Jia-you¹, LING Xing-hong^{1,2}, LIU Quan¹ and ZHU Fei¹

1 School of Computer Science & Technology, Soochow University, Suzhou, Jiangsu 215006, China

2 Wenzheng College of Soochow University, Suzhou, Jiangsu 215104, China

Abstract Traditional deep reinforcement learning methods rely on a large number of samples and are difficult to adapt to new tasks. By extracting prior knowledge from previous training tasks, meta reinforcement learning provides a fast and effective method for agents to adapt to new tasks. Meta deep reinforcement learning based on maximum entropy reinforcement learning framework optimizes strategies by maximizing expected reward and strategy entropy. However, the current meta reinforcement learning algorithms based on the maximum entropy reinforcement learning framework generally adopt fixed temperature parameters, which is unreasonable in the multi-task scenario of meta reinforcement learning. To solve this problem, an adaptive adjustment strategy entropy algorithm is proposed. Firstly, by limiting the entropy of the strategy, the original objective function optimization problem is transformed into a constrained optimization problem. Then, the dual variable in the constrained optimization problem is taken as the temperature parameters, and the updated formula is obtained by solving the dual variable by Lagrange dual method. According to the updated formula, the temperature parameters will be adjusted adaptively after each round of meta training. Experimental data show that the average score of the proposed algorithm on Ant-Fwd-back and Walker-2D increases by 200, the meta training efficiency improves by 82%, the strategy convergence on Human-Direct-2D requires 230 000 training steps, and the convergence speed increases by 127%. Experimental results show that the proposed algorithm has higher meta training efficiency and better stability.

Keywords Meta learning, Reinforcement learning, Maximum entropy

到稿日期:2020-06-22 返修日期:2020-07-29 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:基于云计算的苏州智能公交系统数据挖掘及应用研究(N311800117);江苏高校优势学科建设工程资助项目

This work was supported by the Research on Data Mining and Application of Suzhou Intelligent Public Transportation System Based on Cloud Computing(N311800117) and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

通信作者:凌兴宏(lingxinghong@suda.edu.cn)

1 引言

随着深度强化学习研究的不断发展,深度强化学习^[1]在人工智能领域已经取得了一些重大进展,如“AlphaGo”在围棋“人机大战”中成功击败人类顶尖围棋高手^[2]，“AlphaStar”与人类职业选手在经典即时战略电脑游戏《星际争霸2》的对战中取得压倒性胜利^[3]等。然而,基于传统深度强化学习的智能体无法充分利用在以往任务中学到的先验知识,面临新任务时必须重新训练,且训练过程依赖大量经验样本,导致训练过程十分缓慢。而人类在面临新任务时能够充分利用已有的先验知识,从而快速适应新任务。如何借鉴人类的学习方法,从多个不同的训练任务中提取先验知识,使智能体在面临未知的新任务时减少所需的经验样本,从而快速适应新任务,是目前深度强化学习的前沿挑战之一。

元学习^[4-5],又称作学会学习,旨在利用在以往任务中学习到的先验知识来指导新任务的学习,使智能体具有学会学习的能力,从而达到快速学习的目的。2016年,Wang等^[6]首次将元学习的思想与现有的深度强化学习方法相结合,提出了深度元强化学习,简称元强化学习。元强化学习是一类利用在以往任务中学习到的经验来加快新任务学习的方法,使得智能体面临相似的新任务时只需采集少量的经验样本就能够快速完成训练过程。早期的元强化学习方法主要借鉴监督学习,利用循环神经网络记忆在以往任务中获取的先验知识。Wang等将智能体建模为长短期记忆(Long Short-Term Memory,LSTM)网络。智能体将当前的观察信息、当前的奖赏以及上一个时间步的动作并行输送到传统的深度强化学习算法中,并在训练过程中调整循环神经网络的参数,使循环神经网络能够独立地实现其深度强化学习过程。同年,Duan等^[7]利用门控循环单元(Gated Recurrent Unit,GRU)网络构造智能体,并将智能体学习到的先验知识编码在网络的隐藏状态中。面临新任务时,网络的隐藏状态能够为智能体提供先验知识,辅助智能体快速适应新任务。2018年,Mishra等^[8]在传统递归神经网络的基础上结合了时序卷积与因果注意力机制,分别用于从以往的经验中聚合上下文信息和精确定位上下文中的特定信息,突破了元强化学习智能体引用过去经验能力不足的瓶颈。以上基于循环神经网络的方法重在历史信息进行建模处理,具有操作简单、扩展性强等优点,但模型具有局限性,无法扩展到其他问题中。2017年,Finn等^[9]提出了与模型无关的元学习方法(Model-Agnostic Meta-Learning,MAML),该方法适用于任何通过梯度下降进行训练的模型。其核心思想是训练模型的初始化参数,当面临新任务时,该模型只需要几步的梯度更新就可以快速收敛。在此基础上,许多学者提出了改进方法:Gupta等^[10]利用先验知识学习一个潜在的探索空间,该空间可以将结构化的随机性注入到策略中,产生基于先验知识的探索策略,从而显著提升了智能体在新任务中的探索效率;Rothfuss等^[11]通过控制元策略搜索过程中预适应策略和适应策略之间的统计距离,解决了以往方法信用分配差和估计元策略梯度的问题;Rajeswaran等^[12]提出了一种新的损失函数和相应的计算元策略梯度的方法,减少了算法在计算梯度时所需的运算资源。

然而,基于梯度的元强化学习方法在计算梯度时依赖大量的同策略样本,导致元训练时的样本利用率普遍较低;此外,基于此类方法的智能体在面临稀疏奖赏的新任务时缺乏对任务不确定性的有效推理机制,导致表现欠佳。针对上述问题,Rakelly等^[13]提出了基于任务推断的元强化学习算法PEARL。该算法构建在最大熵强化学习算法SAC^[14]之上,通过引入一个隐变量来表示任务,将元强化学习问题分解为在线的任务推断和任务条件下的策略学习。

目前,以最大熵强化学习框架^[14-15]为基础的元强化学习算法^[13,16]普遍采用了固定的温度参数,忽略了温度参数与训练任务之间的对应关系,这在元强化学习的多任务场景下是不合理的。传统强化学习^[17]的优化目标是最大化累计期望回报,在此基础上,最大熵强化学习框架还需要额外最大化策略的熵,两者的相对比重由温度参数控制。智能体将以最大化策略熵的方式学习到随机性更强的策略,从而获得更强的探索能力与泛化能力。然而,温度参数作为模型的超参数,通常是人为预设的,其选取往往取决于任务本身。当任务奖赏空间稀疏时,应调高温度参数以促进智能体探索;当任务奖赏空间密集时,应调低温度参数以促进智能体利用。在传统强化学习问题场景下,智能体面临的任务是固定不变的,因此温度参数可以在整个训练过程中保持恒定不变。而在元强化学习问题场景下,智能体将面临大量不同的新任务,每个任务的奖赏空间分布各不相同,使得所需要的探索力度也各不相同。因此,为所有任务设置相同的温度参数是不合理的,而现有的基于最大熵强化学习的元强化学习算法普遍忽略了这一问题。针对这一问题,本文提出了自适应调整策略熵算法。该算法首先通过对策略的熵进行限制,将原本的目标函数优化问题转化为受限优化问题,然后将受限优化问题中的对偶变量作为温度参数,通过拉格朗日对偶法求解得到其更新公式。根据得到的更新公式,温度参数将在每一轮元训练结束之后进行自适应调节,使得最大熵强化学习框架与元强化学习的结合更加合理。

2 相关工作

2.1 元强化学习

元强化学习解决的是新任务的快速学习问题,其目的是使智能体学会利用从以往任务中学到的各种经验,从而在面临新任务时能够快速适应。在元强化学习的问题场景下, $p(\tau)$ 是任务 $\{\tau_i\}_{i=1,2,3,\dots}$ 的概率分布,每个任务 τ_i 都可以被建模为马尔可夫决策过程(Markov Decision Process,MDP) $M_i=(S,A,p_i,r_i)$,其中 S 为状态空间, A 为动作空间, p_i 为状态转移概率函数, r_i 为奖赏函数。与传统的机器学习训练方式类似,模型首先需要在元训练任务集 $D_{\text{meta-train}}=\{M_1,M_2,\dots\}$ 上进行训练,然后根据模型在元测试任务集 $D_{\text{meta-test}}=\{M_1',M_2',\dots\}$ 中的表现来评估模型性能。在所有任务中,状态空间与动作空间是共享的,而任务之间的奖赏函数与状态转移概率函数是不同的。因此任务可以由奖赏函数 $r(s_t,a_t)$ 和状态转移概率函数 $p(s_{t+1}|s_t,a_t)$ 定义: $\tau=\{p(s_{t+1}|s_t,a_t),r(s_t,a_t)\}$ 。需要注意的是,每个任务所对应的状态转移概率函数与奖赏函数对于智能体而言是未知的,但是智能体可以通过

与环境的交互来对状态转移概率函数与奖赏函数进行采样。

在元训练期间,智能体利用在元训练集任务中收集到的经验来学习一个能够快速适应新任务的策略。令 $c_n^{\tau} = (s_n, a_n, r_n, s_n')$ 表示任务 τ 中的一次状态转移过程,那么 $c_{1:N}^{\tau}$ 包含了至今为止在任务 τ 中所收集到的经验。为了简便起见,下文中将 $c_{1:N}^{\tau}$ 写作 c 。在元测试时,面对新任务 $\tau \sim p(\tau)$,智能体将利用元训练得到的策略尽可能快速地适应新任务 τ 。

2.2 概率上下文变量与推断网络

在元强化学习问题设定下,所有任务中的动作空间与状态空间是相同的,而任务目标是唯一未知的部分。Rakelly 等^[13]提出,可以将元强化学习问题转换为概率推断问题:当面临一个新的未知任务时,智能体需要利用收集到的经验来推测当前的任务目标,从而利用该任务的先验知识来快速适应新的任务。

由于任务先验知识的表达形式未知,Rakelly 等^[13]定义了概率上下文隐变量 z ,其包含了解决任务所需要的所有信息。在此基础上,策略 $\pi(a|s, z)$ 可以利用其来快速适应任务。通过这种定义,任务目标推断问题便转化为概率上下文隐变量的推断问题。为了确保能够快速适应任务,概率上下文隐变量 z 必须包含解决任务所需要的最充分信息。由于 c 中包含了目前所收集到的经验,可以利用 c 来求出概率上下文隐变量 z 的后验概率 $p(z|c)$ 。然而, $p(z|c)$ 的分布形式是未知的。为了便于计算,采用变分推断^[18-19]的方法将其变分近似为高斯分布,并训练一个推断网络 $q_{\phi}(z|c)$ 来近似概率上下文隐变量 z 的真实后验概率分布 $p(z|c)$,最终得到的变分下界^[18]为:

$$E_{z \sim q_{\phi}(z|c)} [Q(s, a, z) + \beta D_{KL}(q_{\phi}(z|c) \parallel p(z))] \quad (1)$$

其中, $Q(s, a, z)$ 是重构目标; $p(z)$ 是隐变量 z 的单位高斯先验; KL 散度项为信息瓶颈^[19]的变分近似结果,它能够限制任务隐变量 z 与经验 c 之间的互信息,使得隐变量 z 中只包含最必要的任务信息,从而在训练过程中避免了因信息冗余而产生的过拟合;推断网络 $q_{\phi}(z|c)$ 的网络参数 ϕ 仅在元训练过程中进行优化。在元测试时,可根据收集到的经验直接推断出概率上下文隐变量: $z \sim q_{\phi}(z|c)$ 。

2.3 最大熵强化学习

传统强化学习通常用马尔可夫决策过程进行建模。在形式上,传统强化学习的优化目标是找到能够最大化累积期望回报的策略:

$$\pi^* = \arg \max_{\pi} \sum_t E_{(s_t, a_t) \sim p_{\pi}} [r(s_t, a_t)] \quad (2)$$

而最大熵强化学习框架^[14,16]在传统强化学习优化目标的基础上添加了一个额外的熵项,使得最优策略在最大化累积期望回报的同时能够最大化访问过的状态的熵:

$$\pi^* = \arg \max_{\pi} \sum_t E_{(s_t, a_t) \sim p_{\pi}} [r(s_t, a_t) + \alpha H(\pi(\cdot | s_t))] \quad (3)$$

其中, α 是温度参数,它控制着奖赏与熵的相对比重,从而影响最优策略的随机性。当 $\alpha \rightarrow 0$ 时,该目标将回归到传统强化学习的优化目标。

在最大熵强化学习框架下,经过训练的策略可以最大程度地在期望回报和熵之间进行权衡。这与探索和利用的权衡有着密切的联系:增加熵会促进更多的探索,从而加快以后的

学习速度,还可以防止策略过早收敛到不良的局部最优。此外,在最近的研究中,Eysenbach 等^[20]指出,最大熵强化学习对奖赏函数具有鲁棒性,因此非常适合奖赏函数不断变化的元强化学习问题的设定。

2018年,Haarnoja 等^[14]将最大熵强化学习框架与行动者评论家算法^[21]相结合,提出了软行动者评论家算法(Soft Actor-Critic, SAC)。相比之前提出的软 Q 学习算法(Soft Q-Learning, SQL)^[22],该算法取得了更高的样本利用率以及稳定性。为了解决大规模的高维连续控制问题,SAC 分别用网络 $Q_{\theta}(s, a)$ 和 $\pi_{\phi}(a|s)$ 来拟合软 Q 值函数和策略。软 Q 值函数网络参数 θ 与策略网络参数 ϕ 都通过随机梯度下降法进行更新。在值函数更新方面,软 Q 值函数通过贝尔曼方程进行计算:

$$Q(s_t, a_t) = r(s_t, a_t) + E[Q(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1} | s_{t+1})] \quad (4)$$

在网络参数更新方面,软 Q 值函数网络的参数通过最小化贝尔曼残差进行更新,损失函数 $J_Q(\theta)$ 如下,其中 $\bar{\theta}$ 代表目标网络。

$$E \left[\frac{1}{2} (Q_{\theta}(s_t, a_t) - (r(s_t, a_t) + (Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) - \alpha \log \pi_{\phi}(a_{t+1} | s_{t+1}))))^2 \right] \quad (5)$$

策略网络的参数通过最小化策略与软 Q 值函数指数的 KL 散度进行更新,损失函数 $J_{\pi}(\phi)$ 如下,其中 $\log Z(s_t)$ 是用于标准化分布的对数配分函数。

$$\begin{aligned} D_{KL}(\pi_{\phi}(a_t | s_t) \parallel \exp \left(\frac{1}{\alpha} Q_{\theta}(s_t, a_t) - \log Z(s_t) \right)) \\ = E \left[-\log \left(\frac{\pi_{\phi}(a_t | s_t)}{\exp \left(\frac{1}{\alpha} Q_{\theta}(s_t, a_t) - \log Z(s_t) \right)} \right) \right] \\ = E \left[\log \pi_{\phi}(a_t | s_t) - \frac{1}{\alpha} Q_{\theta}(s_t, a_t) + \log Z(s_t) \right] \end{aligned} \quad (6)$$

3 自适应调节策略熵的元强化学习

3.1 自适应调节策略熵

考虑有限时间步无折扣损失的情况,找到一个在最大化期望回报的同时能够满足一定限制条件的策略。

$$\max_{\pi_0, \dots, \pi_T} E \left[\sum_{t=0}^T r(s_t, a_t) \right] \quad \text{s. t.} \quad \forall t, H(\pi_T) \geq H_0 \quad (7)$$

其中, H_0 是预设的熵阈值。累积期望回报 $E \left[\sum_{t=0}^T r(s_t, a_t) \right]$ 可以分解为在所有时间步上的奖赏之和。由于 MDP 满足马尔可夫性质,在 t 时刻的策略 π_t 不会对上一时刻的策略 π_{t-1} 产生影响。因此可以从最后一个时间步向前推移,最大化每一个时间步上的回报。

$$\max_{\pi_0} (E[r(s_0, a_0)] + \max_{\pi_1} (E[\dots] + \max_{\pi_T} E[r(s_T, a_T)])) \quad (8)$$

首先从最后一个时间步 T 开始优化:

$$\begin{aligned} \text{maximize } E_{(s_T, a_T) \sim p_{\pi}} [r(s_T, a_T)] \\ \text{s. t. } H(\pi_T) - H(\pi_0) \geq 0 \end{aligned} \quad (9)$$

为了便于描述限制条件与优化目标,定义如下公式:

$$\begin{aligned} h(\pi_T) &= H(\pi_T) - H_0 \\ &= E_{(s_T, a_T) \sim p_{\pi}} [\log \pi_T(a_T | s_T)] - H_0 \end{aligned} \quad (10)$$

$$f(\pi_T) = \begin{cases} E_{(s_T, a_T) \sim \rho_\pi} [r(s_T, a_T)], & h(\pi_T) \geq 0 \\ -\infty, & h(\pi_T) < 0 \end{cases} \quad (11)$$

优化目标转化为如下形式:

$$\text{maximize } f(\pi_T) \quad \text{s.t. } h(\pi_T) \geq 0 \quad (12)$$

为了求解不等式限制下的最大化优化问题,构建了一个带有拉格朗日乘子(对偶变量) α_T 的拉格朗日表达式,其中拉格朗日乘子 $\alpha_T \geq 0$ 。

$$L(\pi_T, \alpha_T) = f(\pi_T) + \alpha_T h(\pi_T) \quad (13)$$

考虑如下情况,给定策略 π_T 的值,求解使 $L(\pi_T, \alpha_T)$ 取得最小值时的 α_T 值:

$$\min_{\alpha_T \geq 0} L(\pi_T, \alpha_T) = \min_{\alpha_T \geq 0} f(\pi_T) + \alpha_T h(\pi_T) \quad (14)$$

(1)如果满足约束条件,即 $h(\pi_T) \geq 0$,而 π_T 是固定的,则无法通过改变 α_T 的值来控制 $f(\pi_T)$ 的大小。因此,当 $\alpha_T = 0$ 时, $L(\pi_T, \alpha_T)$ 取得最小值,此时 $L(\pi_T, \alpha_T) = L(\pi_T, 0) = f(\pi_T)$ 。

(2)如果不满足约束条件,即 $h(\pi_T) < 0$,可以令 $\alpha_T \rightarrow +\infty$ 来最小化 $L(\pi_T, \alpha_T)$,使得 $L(\pi_T, \alpha_T) \rightarrow -\infty$,此时 $L(\pi_T, \alpha_T) = L(\pi_T, +\infty) = -\infty = f(\pi_T)$ 。

无论哪种情况,都可以得到如下等式:

$$f(\pi_T) = \min_{\alpha_T \geq 0} L(\pi_T, \alpha_T) \quad (15)$$

与此同时,优化目标是最大化 $f(\pi_T)$:

$$\max_{\pi_T} f(\pi_T) = \min_{\alpha_T \geq 0} \max_{\pi_T} L(\pi_T, \alpha_T) \quad (16)$$

为了最大化 $f(\pi_T)$,给出其如下对偶问题。需要注意的是,为了确保 $\max_{\pi_T} f(\pi_T)$ 被合理最大化(即 $f(\pi_T)$ 不会趋向 $-\infty$),必须满足限制条件 $h(\pi_T) \geq 0$ 。

$$\begin{aligned} \min_{\alpha_T \geq 0} \max_{\pi_T} L(\pi_T, \alpha_T) &= \min_{\alpha_T \geq 0} \max_{\pi_T} f(\pi_T) + \alpha_T h(\pi_T) \\ &= \min_{\alpha_T \geq 0} \max_{\pi_T} E_{(s_T, a_T) \sim \rho_\pi} [r(s_T, a_T)] + \alpha_T (E_{(s_T, a_T) \sim \rho_\pi} [-\log \pi_T(a_T | s_T)] - H_0) \\ &= \min_{\alpha_T \geq 0} \max_{\pi_T} E_{(s_T, a_T) \sim \rho_\pi} [r(s_T, a_T) - \alpha_T \log \pi_T(a_T | s_T)] - \alpha_T H_0 \\ &= \min_{\alpha_T \geq 0} \max_{\pi_T} E_{(s_T, a_T) \sim \rho_\pi} [r(s_T, a_T) + \alpha_T H(\pi_T) - \alpha_T H_0] \end{aligned} \quad (17)$$

首先,固定 α_T ,通过最大化 $L(\pi_T, \alpha_T)$ 来获得最优策略 π_T^* 。然后,固定 $\pi_T = \pi_T^*$,通过最小化 $L(\pi_T, \alpha_T)$ 来获得最优对偶变量 α_T^* 。如此往复,可以迭代地计算 π_T 与 α_T 。

$$\pi_T^* = \arg \max_{\pi_T} E_{(s_T, a_T) \sim \rho_\pi} [r(s_T, a_T) + \alpha_T H(\pi_T) - \alpha_T H_0] \quad (18)$$

$$\alpha_T^* = \arg \min_{\alpha_T \geq 0} E_{(s_T, a_T) \sim \rho_{\pi_T^*}} [\alpha_T H(\pi_T^*) - \alpha_T H_0] \quad (19)$$

优化目标(式9)最终转化为:

$$\max_{\pi_T} E[r(s_T, a_T)] = E_{(s_T, a_T) \sim \rho_{\pi_T^*}} [r(s_T, a_T) + \alpha_T^* H(\pi_T^*) - \alpha_T^* H_0] \quad (20)$$

现在回到软Q值函数(式4):

$$\begin{aligned} Q_{T-1}(s_{T-1}, a_{T-1}) &= r(s_{T-1}, a_{T-1}) + E[Q(s_T, a_T) - \alpha_T \log \pi(a_T | s_T)] \\ &= r(s_{T-1}, a_{T-1}) + E[r(s_T, a_T)] + \alpha_T H(\pi_T) \end{aligned} \quad (21)$$

代入最优策略 π_T^* :

$$Q_{T-1}^*(s_{T-1}, a_{T-1}) = r(s_{T-1}, a_{T-1}) + \max_{\pi_T} E[r(s_T, a_T)] + \alpha_T H(\pi_T^*) \quad (22)$$

当退回到时间步 $T-1$ 时,优化目标如下,其中 π_{T-1} 满足 $H(\pi_{T-1}) - H_0 \geq 0$ 。

$$\begin{aligned} \max_{\pi_{T-1}} (E[r(s_{T-1}, a_{T-1})] + \max_{\pi_T} E[r(s_T, a_T)]) \\ = \max_{\pi_{T-1}} (Q_{T-1}^*(s_{T-1}, a_{T-1}) - \alpha_T^* H(\pi_T^*)) \end{aligned} \quad (23)$$

其对偶问题为:

$$\begin{aligned} \min_{\alpha_{T-1} \geq 0} \max_{\pi_{T-1}} (Q_{T-1}^*(s_{T-1}, a_{T-1}) - \alpha_{T-1} H(\pi_{T-1}) - \alpha_{T-1} H_0) \\ = \min_{\alpha_{T-1} \geq 0} \max_{\pi_{T-1}} (Q_{T-1}^*(s_{T-1}, a_{T-1}) + \alpha_{T-1} H(\pi_{T-1}) - \alpha_{T-1} H_0) - \alpha_{T-1} H_0 \end{aligned} \quad (24)$$

与式(18)和式(19)同理,有:

$$\pi_{T-1}^* = \arg \max_{\pi_{T-1}} E_{(s_{T-1}, a_{T-1}) \sim \rho_\pi} [Q_{T-1}^*(s_{T-1}, a_{T-1}) + \alpha_{T-1} H(\pi_{T-1}) - \alpha_{T-1} H_0] \quad (25)$$

$$\alpha_{T-1}^* = \arg \min_{\alpha_{T-1} \geq 0} E_{(s_{T-1}, a_{T-1}) \sim \rho_{\pi_{T-1}^*}} [\alpha_{T-1} H(\pi_{T-1}^*) - \alpha_{T-1} H_0] \quad (26)$$

可以注意到,更新 α_{T-1} 的式(26)与更新 α_T 的式(19)具有相同的形式。不断重复上述过程,最终可以通过最小化相同的目标函数来求解每个时间步的最优对偶变量。对偶变量在实际应用中充当温度参数的角色,其损失函数 $J(\alpha)$ 表示如下:

$$J(\alpha) = E_{a_i \sim \pi_i} [-\alpha \log \pi_i(a_i | \pi_i) - \alpha H_0] \quad (27)$$

3.2 算法设计

利用APE方法对PEARL算法进行优化,最终得到的元训练与元测试算法如算法1和算法2所示。

算法1 元训练

输入:训练任务批次 $\{\tau_i\}_{i=1 \dots T} \sim p(\tau)$;学习率 $\lambda_\phi, \lambda_\psi, \lambda_Q, \lambda_\alpha$

1. 初始化温度参数 α ,推断网络参数 ϕ ,行动者网络参数 ψ ,两个评论家网络参数 θ_1 和 θ_2 ,以及对应的目标值网络参数 $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$
2. 为每个任务 τ_i 初始化经验池 B^i
3. While not done do
4. for τ in $\{\tau_i\}_{i=1 \dots T}$ do
5. 初始化上下文 $c^i = \{\}$
6. while not done do
7. 采样概率上下文变量 $z \sim q_\phi(z | c^i)$
8. 利用策略 $\pi_\psi(a | s, z)$ 收集数据并添加到经验池 B^i
9. 更新 $c^i = \{(s_j, a_j, s_j', r_j)\}_{j=1 \dots N} \sim B^i$
10. end while
11. end for
12. for each training step do
13. for τ in $\{\tau_i\}_{i=1 \dots T}$ do
14. 采集上下文 $c^i \sim B^i$ 以及RL训练数据批次 $b^i \sim B^i$
15. 采样概率上下文变量 $z \sim q_\phi(z | c^i)$
16. $L_{\text{actor}}^i = L_{\text{actor}}(b^i, z)$
17. $L_{\text{critic}}^i = L_{\text{critic}}(b^i, z)$
18. $L_{\text{KL}}^i = \beta \text{DKL}(q(z | c^i) \| r(z))$
19. $L_\alpha^i = -\log_\pi(a | s, z) - \alpha H$
20. end for
21. $\phi \leftarrow \phi - \lambda_\phi \nabla_\phi \sum_i (L_{\text{critic}}^i + L_{\text{KL}}^i)$

22. $\psi \leftarrow \psi - \lambda_\psi \nabla_\psi \sum_i L_{\text{actor}}^i$
23. $\theta_i \leftarrow \theta_i - \lambda_\theta \nabla_{\theta_i} \sum_i L_{\text{critic}}^i$ for $i \in \{1, 2\}$
24. $\alpha \leftarrow \alpha - \lambda_\alpha \nabla_\alpha \sum_i L_\alpha^i$
25. $\bar{\theta}_i \leftarrow \mu \theta_i + (1 - \mu) \bar{\theta}_i$ for $i \in \{1, 2\}$
26. end for
27. end while

算法2 元测试

输入: 新的任务 $\tau \sim p(\tau)$

1. 初始化上下文 $c^r = \{\}$
2. While not done do
3. 采样概率上下文变量 $z \sim q_\phi(z|c^r)$
4. 利用策略 $\pi_\psi(a|s, z)$ 收集数据
 $D = \{(s_j, a_j, s_j', r_j)\}_{j=1 \dots N}$
5. 更新上下文 $c^r = c^r \cup D$
6. end while

元强化学习 Agent 主要由以下 3 种类型的网络构成: 一个推断网络 $q_\phi(z|c)$, 两个评论家网络 $Q_{\theta_i}(s, a, z)$ ($i \in \{1, 2\}$) 和一个行动者网络 $\pi_\psi(a|s, z)$, 其中 $\phi, \theta_i, i \in \{1, 2\}, \psi$ 分别是它们的网络参数。推断网络 $q_\phi(z|c)$ 能够根据行动者网络 $\pi_\psi(a|s, z)$ 收集到的经验 c , 推断出任务信息 z 的近似后验分布 $p(z|c)$; 评论家网络 $Q_{\theta_i}(s, a, z)$ 能够利用从中采样的任务信息 z , 对在当前状态 s 下采取的动作 a 能够获得的期望回报进行估计; 行动者网络 $\pi_\psi(a|s, z)$ 则以类似评论家的方式, 利用任务信息 z 实现任务控制。在更新网络参数之前, Agent 将从任务信息分布 $p(z|c^i)$ 中采样得到当前任务的推测信息 z , 接着, 行动者网络 $\pi_\psi(a|s, z)$ 利用推测信息 z 在每一个训练任务 τ_i 中采样一些数据 c^i , 并将这些数据保存在该任务所对应的经验池 B^i 中。

在元训练过程中, 首先需要为参数更新准备训练数据批次。需要注意的是, 训练推断网络 $q_\phi(z|c)$ 所用的数据批次 $c^i \sim B_i$ 与训练行动者网络 $\pi_\psi(a|s, z)$ 以及评论家网络 $Q_{\theta_i}(a|s, z)$ 所用的数据批次 $b^i \sim B^i$ 不同。具体地, c^i 是从最新采集的数据 B_{new}^i 中采样获得的, 而 b^i 是从整个经验池 B^i 中随机采样获得的。其次, 分别为行动者 π_ψ 与评论家 Q_{θ_i} 计算梯度, 其梯度分别为:

$$L_{\text{actor}} = E_{s \sim B, a \sim \pi_\psi, z \sim q_\phi(z|c)} [\alpha \log(\pi_\psi(a|s, \bar{z})) - \min_{i=1,2} Q_{\theta_i}(s, a, \bar{z})] \quad (28)$$

$$L_{\text{critic}} = E_{(s, a, r, s') \sim B, z \sim q_\phi(z|c)} \left[\frac{1}{2} (\min_{i=1,2} Q_{\theta_i}(s, a, z) - (r + \min_{i=1,2} Q_{\theta_i}(s', a', \bar{z})))^2 \right] \quad (29)$$

其中, \bar{z} 代表梯度没有通过任务信息 z 进行传播, Q_{θ_i} 代表评论家 Q_{θ_i} 对应的目标值网络。在计算 L_{actor} 和 L_{critic} 时, 该算法采用了 TD3 算法^[23]中的截断双 Q 学习的技巧, 使用 $Q_{\theta_i}(s, a, z)$ ($i \in \{1, 2\}$) 中较小的 Q 值作为目标 Q 值, 接着利用变分信息瓶颈^[19]计算推断网络 $q_\phi(z|c)$ 的梯度:

$$L_{\text{KL}} = D_{\text{KL}}(q(z|c^i) \| r(z)) \quad (30)$$

然后, 利用 3.1 节中得出的式(27)计算温度参数 α 的梯度 L_α 。最后, 该算法借鉴 DDPG 算法^[24]中的目标值网络更新技巧, 对目标值网络 $Q_{\theta_i}(s, a, z)$ 的参数 θ_i 进行软更新。

在元测试时, Agent 将面临一个未知的新任务 $\tau \sim p_\tau$ 。Agent 将执行如下步骤: 1) 利用推断网络 $q_\phi(z|c)$ 获取任务的先验分布 $p(z)$, 并从中采样得到任务信息 z ; 2) 行动者网络 $\pi_\psi(a|s, z)$ 试着利用任务信息 z 来完成当前任务 τ , 并收集当前任务适应过程中采集到的数据 c ; 3) 推断网络 $q_\phi(z|c)$ 利用新采集的数据 c 更新任务的后验分布 $p(z|c)$, 并从中采样获取新的任务信息 z 。不断重复上述步骤, Agent 就能够快速适应新任务。

4 实验及比较

本节将系统地分析本文提出的方法在元强化学习连续控制 baseline 上的效果。

4.1 实验设置

本文在 Mujoco^[25] 实验平台上进行实验, 主要工作是实现智能体对新任务的快速连续控制。Mujoco 是一个物理引擎, 旨在促进机器人学、生物力学、图形和动画以及其他需要快速准确模拟的领域的研究和开发。本文在模拟机器人运动的高维连续控制环境下进行实验, 实验环境示意图如图 1 所示, 其所对应的观察空间和动作空间维度信息如表 1 所列。其中, 观察空间为机器人的关节角度与速度, 动作空间为机器人的关节力矩。在每一个环境中, 智能体都将面临多个不同的任务目标, 例如: 1) 到达不同的目标地点 (Ant-Goal-2D); 2) 往不同的方向行走 (Half-Cheetah-Fwd-Back, Ant-Fwd-Back, Humanoid-Direc-2D); 3) 以指定的速度前进 (Half-Cheetah-Vel)。其中, 目标地点、行走方向以及目标速度是随机选取的, 并且对于智能体而言是未知的。在达到目标地点的实验中, 奖赏取决于机器人到达目标地点的速度; 在往不同方向行走的实验中, 奖赏取决于机器人在前进或者后退方向上的行走速度; 在以指定的速度前进的实验中, 奖赏等于机器人当前的前进速度与目标速度的负绝对值。除了不同的任务目标之外, 本文还测试了智能体在面临不同环境动态时 (Walker-2D-Params) 的适应能力。在这类任务中, 智能体将面临随机的环境动态, 智能体获得的奖赏取决于其前进的速度。

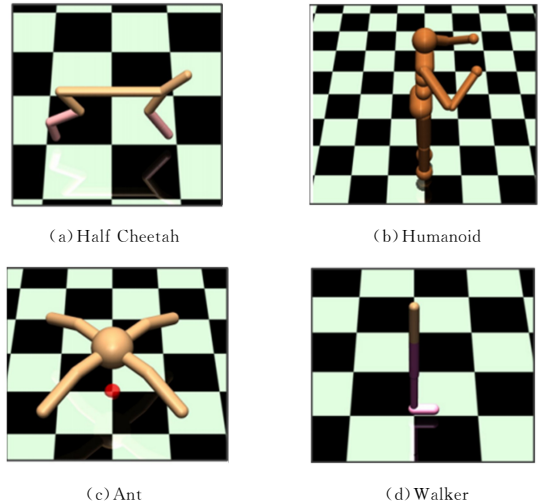


图1 连续控制环境

Fig. 1 Continuous control environments

表1 连续控制环境所对应的观察空间和动作空间维数

Table 1 Dimensions of observation space and action space dimensions corresponding to continuous control environments

Environment	Obs Dim	Action Dim
Half-Cheetah-Fwd-Back	20	6
Half-Cheetah-Vel	20	6
Humanoid-Direc-2D	376	17
Ant-Fwd-Back	27	8
Ant-Goal-2D	113	8
Walker-2D-Params	17	6

对于实验结果的评估,我们将训练得到的模型在元测试任务集上进行测试,并根据收集到的轨迹来评估模型适应新任务的能力。具体地,模型将在测试任务上被评估4次,每一次评估过程中,模型与环境进行600次交互并收集轨迹。我

们对4次评估所收集到的轨迹奖赏之和取平均值,并将其作为最终的评价指标。

实验的所有代码都采用Pytorch深度学习架构编写,运行环境为Ubuntu 16.04操作系统,并使用Tesla P40 GPU加速。

4.2 对比实验

本文选择基于梯度的MAML^[9]、ProMP^[11]、基于循环神经网络的RL²^[7]以及基于概率上下文变量的PEARL^[13]算法作为对比算法,对比实验结果如图2所示。图中,横坐标为训练步数(单位为100万),纵坐标为模型在元测试任务上获得的平均奖赏,水平虚线代表对应算法在最终收敛(经过 1×10^8 步训练)时所能获得的平均奖赏。

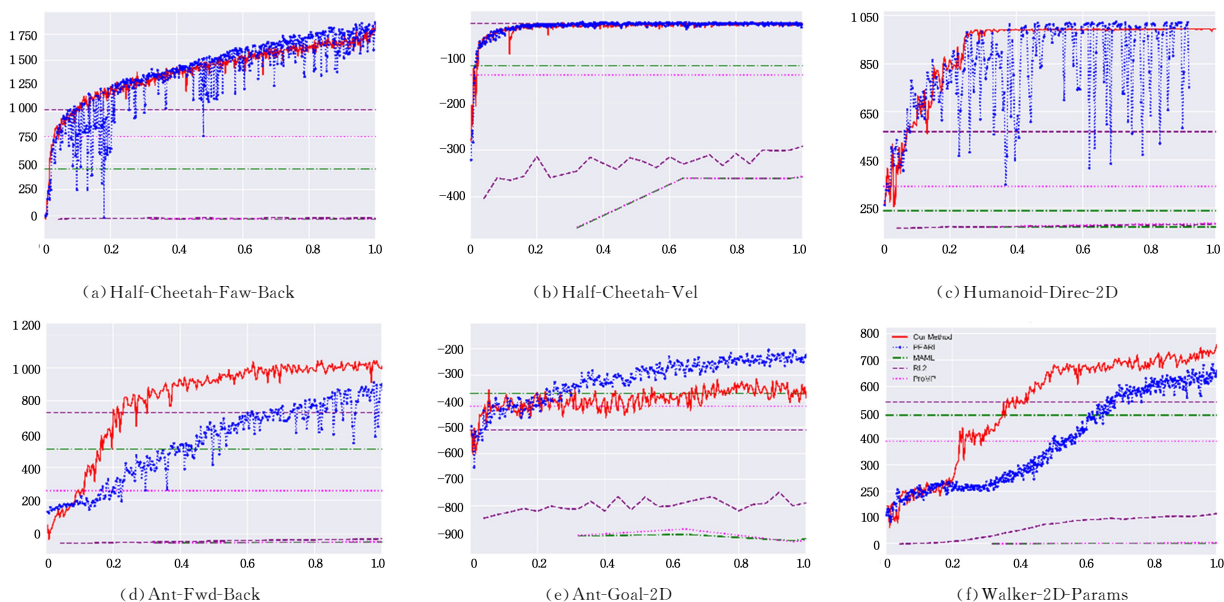


图2 元学习连续控制

Fig. 2 Meta-learning continuous control

对比图2实验结果可以看出,本文提出的方法在样本利用率以及模型性能方面都取得了最好的结果。相比同策略方法(RL²,MAML,ProMP),所提方法仅需百万数量级的训练步数就能够收敛,而同策略方法通常需要亿数量级的训练步数才能收敛;而且,所提方法在模型性能上也普遍优于同策略方法。在Half-Cheetah-Fwd-Back和Half-Cheetah-Vel环境中,所提方法获得了与PEARL相似的训练效果,在稳定性方面略有提升,这可能是因为环境本身相对简单(动作状态空间维度相对较低),易于实现连续控制。在Ant-Fwd-Back以及Walker-2D-Params环境中,所提方法的训练效果超越了PEARL,模型虽然在训练过程的初期起步较低,但是在中后期拥有比PEARL更快的学习速度和更好的稳定性。在最复杂(动作状态空间维度最高)的Humanoid-Direc-2D环境中,所提方法能够非常快速且稳定地适应,仅需不到30万的训练步数就能够收敛;而PEARL一直无法收敛,方差较大,训练曲线上大幅震荡。由此可见,自适应调节策略熵对复杂任务的控制是有效的。通过每轮元训练时对温度参数进行调节,智能体在训练中前期的学习速度显著提升,在训练中后期保持相对稳定。在Ant-Goal-2D环境中,智能体的任务是快

速达到指定的目标地点。到达目标地点的方式不受限制,因此智能体理论上具有无数种近似最优的策略。在该情况下,所提方法可能在一定程度上限制了策略的随机程度,从而导致实验效果不如PEARL。

通过对比实验可以看出,本文提出的方法不仅训练速度更快,而且训练效果也优于其他方法,其性能较为优越。

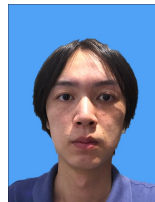
结束语 现有的基于最大熵强化学习框架的元强化学习算法普遍忽略了温度参数与训练任务之间的对应关系,在这些方法中,温度参数始终保持不变,导致智能体在任务之间的探索与利用不合理。本文提出了一种基于自适应调节策略熵的元强化学习算法,该算法首先通过对策略的熵进行限制,将原本的目标函数优化问题转化为受限优化问题;然后将受限优化问题中的对偶变量作为温度参数,通过拉格朗日对偶法求解得到其更新公式;根据得到的更新公式,温度参数将在每一轮元训练结束之后进行自适应调节,使得最大熵强化学习框架与元强化学习的结合更加合理。实验结果表明,所提方法提升了训练的速度以及模型的稳定性,因此适用于更复杂的任务。

在未来的工作中,我们将探究温度参数控制的理论解释,

并对任务推断机制和推断网络结构进行进一步的研究与改进,从而将这类方法扩展到实际应用场景中。

参 考 文 献

- [1] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. *Nature*, 2015, 518(7540):529-533.
- [2] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of go with deep neural networks and tree search[J]. *Nature*, 2016, 529(7587):484-489.
- [3] VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. *Nature*, 2019, 575(7782):350-354.
- [4] SCHMIDHUBER J. Evolutionary principles in self-referential learning[D]. Munich: Univ. Munich, 1987.
- [5] BENGIO Y, BENGIO S, CLOUTIER J. Learning a synaptic learning rule[C]// IJCNN-91-Seattle International Joint Conference on Neural Networks. IEEE, 2002.
- [6] WANG J X, KURTHNELSON Z, TIRUMALA D, et al. Learning to reinforcement learn[C]// CogSci. 2016.
- [7] DUAN Y, SCHULMAN J, CHEN X, et al. RL2: Fast Reinforcement Learning via Slow Reinforcement Learning[C]// International Conference on Learning Representations. 2017.
- [8] MISHRA N, ROHANINEJAD M, CHEN X, et al. A Simple Neural Attentive Meta-Learner[C]// International Conference on Learning Representations. 2018.
- [9] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[C]// Proceedings of the 34th International Conference on Machine Learning. 2017:1126-1135.
- [10] GUPTA A, MENDONCA R, LIU Y, et al. Meta-reinforcement learning of structured exploration strategies[C]// Advances in Neural Information Processing Systems. 2018:5302-5311.
- [11] ROTHFUSS J, LEE D, CLAVERA I, et al. ProMP: Proximal Meta-Policy Search[C]// International Conference on Learning Representations. 2019.
- [12] RAJESWARAN A, FINN C, KAKADE S M, et al. Meta-learning with implicit gradients[C]// Advances in Neural Information Processing Systems. 2019:113-124.
- [13] RAKELLY K, ZHOU A, FINN C, et al. Efficient off-policy meta-reinforcement learning via probabilistic context variables [C]// International Conference on Machine Learning. 2019: 5331-5340.
- [14] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor[C]// International Conference on Machine Learning. 2018:1856-1865.
- [15] ZIEBART B D, MAAS A L, BAGNELL J A, et al. Maximum entropy inverse reinforcement learning[C]// Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. Chicago, Illinois, USA, 2008:13-17.
- [16] WANG H, ZHOU J, HE X. Learning Context-aware Task Reasoning for Efficient Meta-reinforcement Learning [J]. arXiv: 2003. 01373, 2020.
- [17] MONTAGUE P R. Reinforcement learning: an introduction, by Sutton, RS and Barto, AG [J]. *Trends in Cognitive Sciences*, 1999, 3(9):360.
- [18] KINGMA D P, WELING M. Auto-Encoding Variational Bayes [C]// International Conference on Learning Representations. 2014.
- [19] ALEMI A A, FISCHER I, DILLON J V, et al. Deep Variational Information Bottleneck[C]// International Conference on Learning Representations. 2017.
- [20] EYSENBACH B, LEVINE S. If MaxEnt RL is the Answer, What is the Question? [J]. arXiv:1910. 01913, 2019.
- [21] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]// International Conference on Machine Learning. 2016:1928-1937.
- [22] HAARNOJA T, TANG H, ABBEEL P, et al. Reinforcement learning with deep energy-based policies[C]// Proceedings of the 34th International Conference on Machine Learning. 2017: 1352-1361.
- [23] FUJIMOTO S, VAN HOOF H, MEGER D. Addressing Function Approximation Error in Actor-Critic Methods[C]// International Conference on Machine Learning. 2018:1582-1591.
- [24] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning [C]// International Conference on Learning Representations. 2016.
- [25] TODOROV E, EREZ T, TASSA Y. Mujoco: A physics engine for model-based control [C]// 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012: 5026-5033.



LU Jia-you, born in 1996, postgraduate. His main research interests include imitation learning and meta-reinforcement learning.



LING Xing-hong, born in 1968, Ph. D., associate professor. His main research interests include machine learning, artificial intelligence technology and information processing.