软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(6):2765−2775 [doi: 10.13328/j.cnki.jos.006489]

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

基于模型的强化学习中可学习的样本加权机制*

黄文振[1,2], 尹奇跃[1,2], 张俊格[1,2], 黄凯奇[1,2,3]

[1](中国科学院大学 人工智能学院, 北京 100049)

[2](中国科学院 自动化研究所 智能系统与工程研究中心, 北京 100190)

[3](中国科学院 脑科学与智能技术卓越创新中心, 上海 200031)

通信作者: 张俊格, E-mail: jgzhang@nlpr.ia.ac.cn

摘　要: 基于模型的强化学习方法利用已收集的样本对环境进行建模并使用构建的环境模型生成虚拟样本以辅助训练, 因而有望提高样本效率. 但由于训练样本不足等问题, 构建的环境模型往往是不精确的, 其生成的样本也会因携带的预测误差而对训练过程产生干扰. 针对这一问题, 提出了一种可学习的样本加权机制, 通过对生成样本重加权以减少它们对训练过程的负面影响. 该影响的量化方法为, 先使用待评估样本更新价值和策略网络, 再在真实样本上计算更新前后的损失值, 使用损失值的变化量来衡量待评估样本对训练过程的影响. 实验结果表明, 按照该加权机制设计的强化学习算法在多个任务上均优于现有的基于模型和无模型的算法.

关键词: 基于模型的强化学习; 模型误差; 元学习; 强化学习; 深度学习

中图法分类号: TP181

中文引用格式: 黄文振, 尹奇跃, 张俊格, 黄凯奇. 基于模型的强化学习中可学习的样本加权机制. 软件学报, 2023, 34(6): 2765–2775. http://www.jos.org.cn/1000-9825/6489.htm

英文引用格式: Huang WZ, Yin QY, Zhang JG, Huang KQ. Learnable Weighting Mechanism in Model-based Reinforcement Learning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(6): 2765–2775 (in Chinese). http://www.jos.org.cn/1000-9825/6489.htm

Learnable Weighting Mechanism in Model-based Reinforcement Learning

HUANG Wen-Zhen[1,2], YIN Qi-Yue[1,2], ZHANG Jun-Ge[1,2], HUANG Kai-Qi[1,2,3]

[1](School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China)

[2](Center for Research on Intelligent System and Engineering (CRISE), Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

[3](Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China)

Abstract: Model-based reinforcement learning methods train a model to simulate the environment by using the collected samples and utilize the imaginary samples generated by the model to optimize the policy, thus they have potential to improve sample efficiency. Nevertheless, due to the shortage of training samples, the environment model is often inaccurate, and the imaginary samples generated by it would be deleterious for the training process. For this reason, a learnable weighting mechanism is proposed which can reduce the negative effect on the training process by weighting the generated samples. The effect of the imaginary samples on the training process is quantified through calculating the difference between the losses on the real samples before and after updating value and policy networks by the imaginary samples. The experimental results show that the reinforcement learning algorithm using the weighting mechanism is superior to existing model-based and model-free algorithms in multiple tasks.

Key words: model-based reinforcement learning; model-bias; meta-learning; reinforcement learning; deep learning

强化学习在许多领域取得了令人瞩目的成果, 如在雅达利游戏[1], 围棋[2]等领域达到甚至超过人类专家水平, 也被广泛应用于现实生活中, 如: 交通控制[3−5], 金融交易[6,7]等, 还被视为人机对抗、通用人工智能等技术的重要

* 基金项目: 国家自然科学基金 (61876181, 61673375); 北京市科技创新计划 (Z191100000119043); 中国科学院青年创新促进会项目; 中国科学院项目 (QYZDB-SSW-JSC006)
收稿时间: 2021-04-14; 修改时间: 2021-06-07; 采用时间: 2021-08-12; jos 在线出版时间: 2022-10-14
CNKI 网络首发时间: 2022-11-16

基石[8,9]. 这类在与环境交互的过程中直接学习策略的方法, 被称为无模型强化学习方法, 它们的样本效率较低, 通常需要大量的交互数据才能训练出良好的策略, 这一问题导致它们的应用范围大多局限于一些能够低成本大量并行的环境中. 与之相对的, 基于模型的强化学习方法在训练过程中利用已收集样本对环境进行建模, 并使用建立的动力学模型 (dynamics model) 来生成虚拟样本用于策略训练或实时规划等, 因而有望显著地提高样本效率, 使得在使用较少的交互数据的情形下训练出良好的策略.

早期的研究工作表明, 在一些简单的低维输入的控制任务上, 利用线性或贝叶斯模型对环境进行建模的强化学习方法表现出了优异的性能[10−12]. 但由于模型较为简单, 这些方法难以应用到更为复杂的高维输入的控制任务中. 基于神经网络的模型可以拟合更复杂的状态转换函数, 因而被用于对更为复杂的环境进行建模, 并辅助解决复杂的控制任务[13−16]. 然而, 无论哪种模型, 由样本缺失等因素导致的模型误差 (model bias) 问题均无可避免, 这一问题会导致训练过程容易陷入局部最优, 甚至可能导致算法灾难性的崩溃[10].

针对上述模型误差问题, 有许多不同的解决思路, 例如: 通过集成多个概率模型来模拟预测状态的后验分布, 再使用这些模型来规划, 得到该后验分布下的期望奖励最大的决策[17]; 或是, 利用元学习方法来学习一个策略, 使得该策略能够良好地适应集成模型中的所有模型, 从而得到一个对模型误差具有鲁棒性的策略[18].

另有一些工作考虑调整动力学模型的使用方式 (model usage) 以减少模型误差对策略学习的不利影响, 也取得了一定的效果, 例如: 先估计动作价值网络预测的不确定度, 仅在不确定度高于某一阈值的情况下, 才使用动力学模型生成相关数据进行训练[19]; 生成数据只用于策略梯度的计算而不被应用于价值网络的训练过程[20]; 将利用动力学模型从起始状态生成完整的轨迹, 转变为从经验池中随机状态开始生成较短的轨迹[21].

以上调整方案较为简单且大多数方案在整个训练过程中保持不变, 这导致部分生成数据即使是完全准确的, 也可能会在训练流程中的某些阶段始终被忽略. 本文考虑自适应地过滤掉具有较大预测偏差的生成样本, 来减小这些样本对价值和策略网络训练的负面影响, 进而减轻由模型偏差引起的策略性能下降. 但样本的实际预测偏差无法直接获得, 即使利用不确定度来评估潜在预测偏差的大小, 也会存在阈值难以设定的问题, 例如: 当价值网络由于欠拟合导致值估计存在较大偏差时, 即使是具有较大预测误差的样本也可用于优化该网络.

针对上述问题, 本文尝试量化生成样本对训练过程的影响, 并基于此来对它们进行重加权, 从而自适应地调整动力学模型的使用方式. 整体思路类似于交叉验证, 先使用生成样本来更新价值和策略网络, 再将更新前后的网络分别作用在真实样本上, 对比优化目标 (例如时序差分的平方) 的数值变化, 以此来衡量生成样本对训练过程的影响, 并根据影响是有利还是有害来决定生成样本的权重. 为了方便地获取新生成样本的权重, 本文考虑训练一个权重预测网络来为每个生成样本提供合适的权重, 该网络根据输入样本的特征 (如: 样本中状态和奖励预测的不确定度), 输出一个介于 0 到 1 之间的权重.

上述量化标准可以直接用于权重预测网络的优化: 给定任意生成样本, 使用权重预测网络为其预测权重, 然后使用加权后的优化目标更新价值网络和策略网络的参数. 将更新前后的参数和真实样本分别带入到优化目标中, 更新前后优化目标的差异即反映了加权样本对训练过程的影响. 由于更新后的优化目标对更新后的参数可导, 而参数更新的步长是通过权重预测网络的输出来参数化的, 并且更新前的优化目标与权重网络的输出无关, 因此可以使用链式法则通过对优化目标的前后差异求权重预测网络参数的导数来优化权重预测网络. 考虑到更新价值和策略网络时, 学习率对参数更新过程的影响, 本文将真实样本加入上述过程来调节该学习率. 考虑到同一个样本对价值网络和策略网络的作用通常并不相同, 所以本文使用两个权重预测网络分别预测样本在价值函数和策略函数训练过程中的适宜权重. 以上优化方法可以视为是元学习方法[22,23]的一种特殊形式——元梯度方法[24−26], 这类元学习方法中元学习者根据元参数对基于梯度的优化过程的影响, 来对元参数计算梯度并进行更新, 从而达到加速优化过程的目的.

实验结果表明, 在多个控制任务上, 本文提出的方法优于当前最优的基于模型和无模型的强化学习方法. 使用加权样本更新的参数所对应的价值预测损失明显小于未加权方法所对应的损失, 这一现象意味着, 对训练过程具有不利影响的样本确实地被权重预测网络以减少权重的方式过滤掉了. 对比文献 [27] 中的方法, 本文所提的方法能够提供更合理的权重, 在减小生成样本对训练的负面影响的同时, 更充分地利用生成样本.

本文在先前工作[27]的基础上进行了完善, 本文提供的额外贡献为: (1) 在使用元学习方法对权重网络进行优化的过程中, 考虑对内层优化 (单次随机梯度下降) 中的学习率进行自适应地调整, 避免因损失值过小或学习率过大而导致的样本权重低估问题; (2) 分离动作价值函数和策略函数优化过程中所使用的权重, 即针对两个优化过程分别训练权重网络, 避免单个样本在两个优化过程中的影响不一导致学习到的权重仅为折中结果.

1 环境建模方法

1.1 符号说明

标准强化学习设定下, 环境可由以下元素来定义: 状态空间 $\mathcal{S}$, 动作空间 $\mathcal{A}$, 奖励函数 $r:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$, 状态转移概率 $p:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,\infty)$, 以及折扣因子 $\gamma\in(0,1]$, 其中状态转移概率 $p(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_{t+1})$ 表示给定当前状态 $\mathbf{s}_t\in\mathcal{S}$ 和动作 $\mathbf{a}_t\in\mathcal{A}$ 时, 下一时刻状态 $\mathbf{s}_{t+1}\in\mathcal{S}$ 的概率密度, 奖励函数 $r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_{t+1})$ 表示该状态转移过程对应的奖励. 强化学习的主要目标为学习一个策略使得环境反馈的累积奖励最大化.

1.2 整体框架

基于模型的强化学习方法通常会利用已收集到的轨迹来学习一个动力学模型, 以模拟真实环境的状态转移过程, 并基于该模型更快地学习到好的决策. 但在环境较为复杂或收集到真实样本较少的情况下, 学习到的动力学模型通常是不完美的, 这样, 由它生成的样本也将带有预测误差, 而这些带有预测误差的样本往往会对价值和策略函数的训练过程产生负面影响. 因此, 本文尝试构造一种可学习的重加权机制来最小化生成样本的负面影响.

为了高效地获取新生成样本所对应的适宜权重, 即能够最小化加权后样本对训练过程不利影响的权重, 本文构建一个权重预测网络来预测输入样本所对应的权重. 针对不利影响这一抽象的概念, 本文通过类似于交叉验证的方法来进行度量: 先使用重加权的样本来更新价值和策略网络, 再将更新前后的网络参数作用于真实样本上计算损失函数数值的变化, 该变化即反映了重加权样本对训练过程的影响. 由于更新前的损失值与预测的权重无关, 因此最小化更新后的损失值即可最小化重加权样本的负面影响. 该损失对更新后的价值和策略网络的参数可导, 更新后的参数可通过更新梯度对预测权重进行求导, 而预测权重由权重预测网络的参数参数化, 因此可以利用链式法则求取更新后的损失对权重预测网络参数的梯度, 并以此来优化权重预测网络.

值得注意的是, 权重的大小与更新价值和策略网络时的学习率密切相关, 当学习率过大时, 即使是真实样本在以上评估过程中, 也会获得一个较小的权重, 因此需要对更新过程的学习率进行自适应的调整. 一个合适的学习率应该保证使用真实样本更新价值和策略网络后, 损失值减小或保持不变. 所以, 将该更新过程中的学习率视为一个可学习的参数, 按照上文优化权重网络的方法使用真实样本对其进行优化. 为了更高效地训练, 将真实样本视为权重为 1 的生成样本与普通的生成样本一起加入权重网络和学习率优化过程. 此外, 考虑到同一个样本对价值网络和策略网络的作用通常并不相同, 所以使用两个权重预测网络分别预测样本在价值网络和策略网络训练过程中的适宜权重. 权重预测网络的训练过程如图 1 所示.

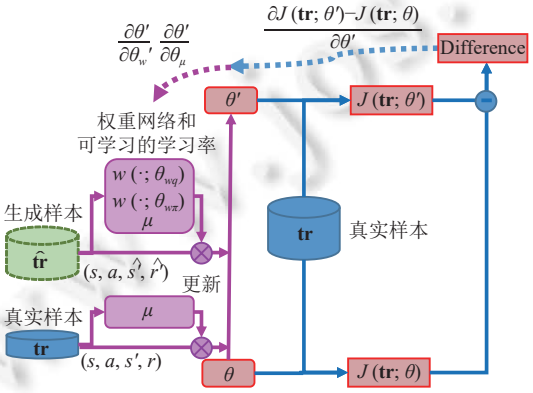

图 1 训练框架

为保证权重预测网络有足够的信息来为每个输入样本提供合适的权重, 本文使用集成的自举概率神经网络 (bootstrapped probabilistic neural network) 为动力学模型, 该网络结构可以估计生成样本中状态和奖励预测结果的不确定度, 该不确定度可以提供生成样本所带有的潜在预测误差大小的信息. 加权后的样本将通过无模型的强化学习算法对价值和策略网络进行更新, 本文选用 SAC (soft actor-critic) 算法[28]. SAC 是一种离策略强化学习算法, 因此经验池里的真实样本可以直接用于计算权重网络训练过程中更新后的损失值. 本文将此实现称为 LR-MPO (learnable reweighted model-based policy optimization).

接下来, 将详细介绍以下 3 个部分: 动力学模型的训练方法, 权重预测网络的网络结构, 以及权重预测网络的训练方法.

1.3 动力学模型 (dynamics model) 的训练方法

在上文描述的算法框架中, 动力学模型的功能有两个: (1) 根据输入状态和动作对下一时刻的状态进行预测, 从而获取到生成样本; (2) 提供一些信息, 例如不确定度, 用以辅助对这些样本的权重预测.

为了保证动力学模型的功能齐全, 本文仿照 PETS[17]中的方法训练了一个由 B 个自举概率模型组成的模型集成. 每个概率模型都是一个神经网络, 它根据输入状态 $\mathbf{s}$ 和动作 $\mathbf{a}$ 预测下一个状态 $\mathbf{s}'$ 的概率分布, 该概率分布由高斯分布 $\mathcal{N}(\mu_{\theta_b}(\mathbf{s},\mathbf{a}),\Sigma_{\theta_b}(\mathbf{s},\mathbf{a}))$ 来近似. 这 B 个模型拥有相同的网络结构, 但它们的初始参数 θ_{b0} 和训练数据集 $\mathcal{D}_b$ 却各不相同. 训练数据集 $\mathcal{D}_b$ 是通过对经验池 (replay buffer) $\mathcal{R}$ 进行 N 次有放回的随机采样产生的, 其中 N 等于经验池 $\mathcal{R}$ 的大小. 下一时刻状态的预测可以通过对该高斯分布进行采样得到. 与其他基于模型的强化学习的工作[17,18]相同, 本文也假定奖励函数 $r(s,a,s'):\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$ 预先给定.

给定状态 $\mathbf{s}_t$ 和动作序列 $\mathbf{a}_{t:t+H-1}=(\mathbf{a}_t,\dots,\mathbf{a}_{t+H-1})$, 动力学模型通过以下方式来对后续轨迹 $\mathbf{s}_{t+1:t+H}$ 进行递归地预测. 针对输入状态 $\mathbf{s}_t$ 和动作 $\mathbf{a}_t$, 概率模型的集成会对下一个时刻状态 $\mathbf{s}_{t+1}$ 的分布产生 B 个不同的预测, 从每个预测的高斯分布 $\mathcal{N}(\mu_{\theta_b}(\mathbf{s}_t,\mathbf{a}_t),\Sigma_{\theta_b}(\mathbf{s}_t,\mathbf{a}_t))$ 中采样出 M 个状态预测 $\{\hat{\mathbf{s}}_{t+1}^{mb}\}_{m=1}^{M}$. 将预先给定的奖励函数作用于采样出的下一个状态上以评估它们对应的奖励, $\hat{r}_{t+1}^{mb}=r(\mathbf{s}_t,\mathbf{a}_t,\hat{\mathbf{s}}_{t+1}^{mb})$. 从 $M\times B$ 个预测状态 $\{\hat{\mathbf{s}}_{t+1}^{mb}\}_{m=1,b=1}^{M,B}$ 中随机选择一个作为下一个输入 $\hat{\mathbf{s}}_{t+1}$. 然后, 使用所选状态 $\hat{\mathbf{s}}_{t+1}$ 和动作 $\mathbf{a}_{t+1}$ 来生成新的 $M\times B$ 个状态预测. 这样, 在每个时间步 $t+k,k=0,\dots,H-1$ 上都将获得一个样本集 $\hat{\mathbf{tr}}_k=\{(\hat{\mathbf{s}}_{t+k},\mathbf{a}_{t+k},\hat{r}_{t+k}^{b,m},\hat{\mathbf{s}}_{t+k+1}^{b,m})\}_{m=1,b=1}^{M,B}$.

1.4 权重预测网络的网络结构

由于单个样本 $(\mathbf{s},\mathbf{a},\hat{r},\hat{\mathbf{s}}')$ 无法提供任何与下一状态 $\hat{\mathbf{s}}'$ 和奖励 $\hat{r}$ 的预测精度相关的信息, 所以权重预测网络难以做到为单个样本生成预测权重. 集成概率模型在每一步的预测结果——样本集 $\hat{\mathbf{tr}}_k=\{(\hat{\mathbf{s}},\mathbf{a},\hat{r}^{b,m},\hat{\mathbf{s}}'^{b,m})\}_{m=1,b=1}^{M,B}$, 则可以提供关于下一状态 $\hat{\mathbf{s}}'$ 和奖励 $\hat{r}$ 的预测结果的不确定度, 该不确定度一定程度上能够反映这批样本的预测精度, 为权重网络的预测提供更多的信息. 因此, 本文选择样本集作为权重预测网络的输入.

为了更好地预测样本的权重, 本文为每个样本集 $\hat{\mathbf{tr}}$ 构造了一个简单的特征向量 $\mathbf{x}_{\mathbf{tr}}$. 该特征向量包含预测奖励 $\hat{r}$ 的不确定度、预测的下一状态 $\hat{\mathbf{s}}'$ 的不确定度以及预测的下一状态的价值 $V(\hat{\mathbf{s}}')$ 的不确定度, 这些不确定度可以通过对 $\{(\hat{r}^{b,m},\hat{\mathbf{s}}'^{b,m},V(\hat{\mathbf{s}}'^{b,m}))\}_{m=1,b=1}^{M,B}$ 计算标准差来近似. 其中 $V(\mathbf{s})=\mathbb{E}_{a\sim\pi(s;\theta_\pi)}[Q(\mathbf{s},\mathbf{a};\theta_q)]$, 该期望通过蒙特卡罗采样的方式来近似计算.

特征向量中的预测奖励、预测状态和预测状态价值的不确定度, 直接反映了生成样本在这 3 个层面的置信程度, 或者说是生成样本与其对应的真实样本在这 3 个层面上的潜在偏差程度, 所以权重预测网络需要这些信息来调整权重. 为了避免不同特征的数值尺度间存在巨大差异, 本文对特征向量的每一维分别维护一个移动平均值和一个移动方差值, 在将特征向量送入权重预测网络之前, 会使用移动平均值和移动方差值对每一维进行归一化.

针对动作价值网络和策略网络的训练, 本文使用两个结构相同但参数不同的权重预测网络分别为样本提供权重, 它们的参数分别记为 θ_{wq} 和 $\theta_{w\pi}$. 权重网络根据输入样本集的特征向量 $\mathbf{x}_{\mathbf{tr}}$ 预测其对应的权重, $w(\mathbf{x}_{\mathbf{tr}};\theta_{w\cdot}):\mathbb{R}^D\to(0,1)$. 因为这些样本集 $\mathbf{tr}_t$ 是使用动力学模型迭代生成的, 其中的状态 $\mathbf{s}_t$ 大多是模型的预测结果而非真实的状态, 所以这些生成数据的置信程度往往与它们前驱 $\mathbf{tr}_{t-1}$ 的置信程度相关. 因此, 本文选择 GRU (gated recurrent units)[29]来整合前驱样本集的特征. 单个权重预测网络的整体结构如图 2 所示.

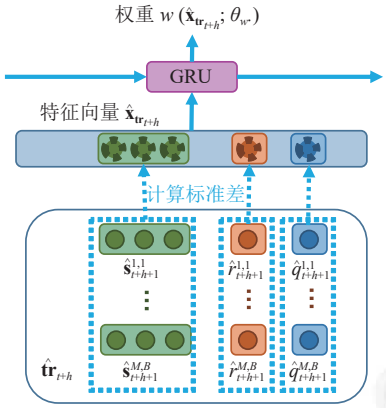

图 2 神经网络框架

1.5 权重预测网络的训练方法

本节将说明如何训练权重预测网络, 使其能够为生成样本提供适当的权重, 从而最大程度地减少这些样本对训练过程的不利影响.

权重预测网络的训练过程和其他元学习方法一样可以分为内层优化和外层优化两个步骤: 内层优化——使用重加权后的样本来更新动作价值网络和策略网络, 外层优化——使用真实样本计算更新后参数的损失, 并利用链式法则通过最小化更新后的损失来优化权重网络. 为了更高效且更稳定地训练, 每次会随机生成一批生成样本并随机选择一批真实样本进行以上训练过程.

● 内层优化. 从经验池中随机选择 N_e 个真实状态 $\{\mathbf{s}_{l_i}^i\}_{i=1}^{N_e}$ 以及对应的长度为 H 的真实动作序列 $\{\mathbf{a}_{l_i:l_i+H-1}^i\}_{i=1}^{N_e}$, 并使用动力学模型生成对应的预测序列 $\{\hat{\mathbf{tr}}_h^i\}_{i=1,h=1}^{N_e,H}$. 然后构建它们的特征 $\mathbf{x}_{\hat{\mathbf{tr}}_h^i}$, 并使用权重预测网络估计这些样本的权重. 然后, 使用这些权重对生成样本的损失函数进行加权, 并更新动作价值网络和策略网络的参数 θ_q 和 θ_π.

$$\theta_q' = \theta_q - \mu\frac{\partial\sum_{i,h} w(\mathbf{x}_{\hat{\mathbf{tr}}_h^i};\theta_{wq})J_Q(\hat{\mathbf{tr}}_h^i;\theta_q)}{\partial\theta_q},$$

$$\theta_\pi' = \theta_\pi - \mu\frac{\partial\sum_{i,h} w(\mathbf{x}_{\hat{\mathbf{tr}}_h^i};\theta_{w\pi})J_\pi(\hat{\mathbf{tr}};\theta_\pi)}{\partial\theta_\pi},$$

其中, μ 表示内层优化过程中可优化的学习率, J_Q 表示软化的贝尔曼残差 (soft Bellman residual), J_π 表示策略网络输出与软化后动作价值预测的指数间的 KL-散度 (KL-divergence). 对于任意样本集 $\hat{\mathbf{tr}} = \{(\hat{\mathbf{s}}, \mathbf{a}, \hat{r}^{b,m}, \hat{\mathbf{s}}'^{b,m})\}_{m=1,b=1}^{M,B}$, J_Q 和 J_π 的计算公式为:

$$J_Q(\mathbf{tr};\theta_q) = \sum_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\in\mathbf{tr}}\frac{1}{2}\{Q(\mathbf{s},\mathbf{a};\theta_q) - [r+\gamma(Q(\mathbf{s}',\mathbf{a}';\bar{\theta}_q) - \alpha\log\pi(\mathbf{a}'|\mathbf{s}'))]\}^2,$$

$$J_\pi(\mathbf{tr};\theta_\pi) = \sum_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\in\mathbf{tr}}\{\alpha\log(\pi(\hat{\mathbf{a}}|\mathbf{s};\theta_\pi)) - Q(\mathbf{s},\hat{\mathbf{a}};\theta_q)\},$$

其中, $\bar{\theta}_q$ 表示目标动作价值网络的参数, α 表示温度参数, 它们的更新方式与原始的 SAC 算法[28]一致.

● 外层优化. 从经验池里再随机选出 N_v 个真实样本, 并将它们的组合记为 $\mathbf{tr}$. 使用更新后的动作价值网络和策略网络的参数 θ_q' 和 θ_π', 来计算相应的损失:

$$J_Q(\mathbf{tr};\theta_q') + J_\pi(\mathbf{tr};\theta_\pi'),$$

该损失对参数 θ_{wq}、$\theta_{w\pi}$ 和 μ 的梯度可以通过链式法则来计算:

$$\frac{\partial J_Q(\mathbf{tr};\theta_q') + J_\pi(\mathbf{tr};\theta_\pi')}{\partial\theta_{wq}} = -\mu\sum_{h,i}\left[\left(\frac{\partial J_Q(\hat{\mathbf{tr}}_h^i;\theta_q)}{\partial\theta_q}\right)^{\mathrm{T}}\frac{\partial J_Q(\mathbf{tr};\theta_q')}{\partial\theta_q'}\right]\frac{\partial w(\mathbf{x}_{\hat{\mathbf{tr}}_h^i};\theta_{wq})}{\partial\theta_{wq}},$$

$$\frac{\partial J_Q(\mathbf{tr};\theta'_q)+J_\pi(\mathbf{tr};\theta'_\pi)}{\partial\theta_{w\pi}}=-\mu\sum_{h,i}\left[\left(\frac{\partial J_\pi(\hat{\mathbf{tr}}_h^i;\theta_\pi)}{\partial\theta_\pi}\right)^{\mathrm{T}}\frac{\partial J_\pi(\mathbf{tr};\theta'_\pi)}{\partial\theta'_\pi}\right]\frac{\partial w(x_{\hat{\mathbf{tr}}_h^i};\theta_{w\pi})}{\partial\theta_{w\pi}},$$

$$\frac{\partial J_Q(\mathbf{tr};\theta'_q)+J_\pi(\mathbf{tr};\theta'_\pi)}{\partial\mu}=-\sum_{h,i}\left[\left(\frac{\partial J_Q(\hat{\mathbf{tr}}_h^i;\theta_q)}{\partial\theta_q}\right)^{\mathrm{T}}\frac{\partial J_Q(\mathbf{tr};\theta'_q)}{\partial\theta'_q}+\left(\frac{\partial J_\pi(\hat{\mathbf{tr}}_h^i;\theta_\pi)}{\partial\theta_\pi}\right)^{\mathrm{T}}\frac{\partial J_\pi(\mathbf{tr};\theta'_\pi)}{\partial\theta'_\pi}\right].$$

求取出梯度后, 可以通过任何优化算法来更新参数 θ_{wq}、$\theta_{w\pi}$ 和 μ.

● 交替优化权重预测网络与动作价值网络和策略网络. 随着动作价值网络和策略网络的更新, 生成样本的适宜权重也会随之改变, 所以需要交替地优化权重预测网络与动作价值网络和策略网络, 以保证前者可以随后者精度的变化而自适应地进行调整. 动作价值网络和策略网络的更新方式为, 从经验池中选择 N_t 个真实状态, 使用探索策略 π_e 在学习到的动力学模型上进行采样, 生成长度为 H 的虚拟轨迹 $\{\hat{\mathbf{tr}}_h^i\}_{i=1,h=1}^{N_t,H}$. 这里的探索策略 π_e 是通过将当前的温度参数 α 变为 $\lambda_e\alpha$ 得到的 (本文中的 λ_e 设为 10), 较大的温度参数会加大采样出的动作的浮动范围, 从而增加生成样本的多样性. 使用权重预测网络为这些生成样本提供合适的权重, 并使用对应的加权损失为优化目标来计算动作价值网络和策略网络的参数的梯度, 然后使用 Adam 优化算法[30]来更新参数 θ_q 和 θ_π. 而温度参数 α 直接使用未进行重加权的优化目标进行优化. 为了更有效的训练, 真实样本也会加入训练过程, 但不需要为它们额外计算权重.

在该算法中, 真实样本不仅被应用于训练动力学模型和优化权重预测网络, 而且还被用于训练动作价值网络和策略网络. 真实样本可以一定程度抑制因预测偏差较大的样本导致的动作价值网络预测误差过大问题, 而且在生成样本的预测权重均较低时, 能够避免算法陷入停滞.

2 实验结果与可视化分析

本节将使用基于模型的强化学习基准测试集[31]中的 6 个复杂的连续控制任务对本文提出的方法进行评估. 这 6 个任务分别是 Ant、HalfCheetah、Hopper、SlimHumanoid、Swimmer-v0 和 Walker2D, 每个任务的总时间步数均固定为 1 000. 本节结构如下: 首先, 介绍本文中各模块的具体网络结构以及训练过程的各种超参数; 然后, 将评估 LR-MPO 的性能并对其各种特性进行分析, 包括: 在上述 6 个任务上将 LR-MPO 与当前最优的无模型和基于模型的强化学习方法进行比较, 评估是否加权对动作价值预测损失的影响, 以及分析可优化的学习率对权重网络输出的影响.

2.1 网络结构

● 动力学模型. 动力学模型由 5 个全连接神经网络 (fully connected neural networks) 组成, 每个网络均包含 4 层宽度为 200 的隐藏层, 网络中的非线性激活函数为: $f(x)=x\times Sigmoid(x)$. 该模型使用 Adam 算法进行优化, 算法的参数为: 每批样本数量 (batchsize) 等于 64, 学习率等于 1E–3, *betas*=(0.9, 0.999) .

● 动作价值网络和策略网络. 动作价值网络和策略网络是两个单独的全连接神经网络, 每个网络均包含 2 层宽度为 256 的隐藏层, 网络中的非线性激活函数为 ReLU. 该模型同样使用 Adam 算法进行优化, 算法的参数为: 学习率等于 3E–4, *betas*=(0.99, 0.9999). 生成样本的重加权可能导致梯度的尺度会发生巨大浮动, 因此将 *betas* 设置为较大的值可以提高训练的稳定性.

● 权重预测网络. 权重预测网络从输入到输出依次由一个宽度为 64 的全连接层、一个隐藏单元为 16 的 GRU 模块和一个宽度为 1 的全连接层组成, 最后一层的输出会通过一个 Sigmoid 激活函数转化 0 到 1 之间. 该模型同样使用 Adam 算法进行优化, 算法的参数为: 学习率等于 1E–4, *betas*=(0.9, 0.999).

2.2 超参数设置

在最初的 10 000 个时间步 (智能体与环境交互一次为一步) 中, 智能体的行动是通过对所有可行的动作进行均匀随机采样来决定的. 从第 3 000 个时间步, 开始训练权重预测网络以及动作价值网络和策略网络. 在每一个时间步, 从经验池中随机采样出 $N_e=128$ 个真实状态, 并根据相应的长度为 $H=5$ 的真实动作序列来生成虚拟样本, 使用这些样本与随机选出的 $N_m=64$ 个真实样本来进行内层优化, 再随机采样出 $N_v=2560$ 个真实样本进行外层优

化. 之后, 从经验池中随机采样出 2 560 个真实状态, 并使用上文提到的探索策略来产生对应的长度为 $H=5$ 的动作序列, 以生成虚拟样本. 这些生成样本被随机分为 10 份, 并用于更新动作价值网络和策略网络 10 次. 再随机采样出 256 个真实样本, 更新动作价值网络和策略网络的参数 1 次. 每次任务结束 (智能体与环境交互 1 000 次) 时, 动力学模型使用所有收集到的样本进行 5 轮训练, 每轮训练均会遍历全部样本.

2.3 与当前最优方法的性能对比

本节将对 LR-MPO 与当前最优的无模型和基于模型的强化学习方法进行了对比, 其中, 无模型的方法包括: SAC[28]和 TD3[32]. 基于模型的方法包括: ME-TRPO[33]、MB-MPO[18]、PETS[17]、MBPO[21]、POPLIN[34]和 Rew-PE-SAC[27]. LR-MPO 算法在每个任务上运行 200 000 个时间步, 并使用不同的随机种子重复执行了 8 次. 为了更好地评估本文所提出的可学习的加权机制, 在 6 个任务上运行了 PE-SAC 算法[27], 该算法不学习权重网络, 直接使用生成样本进行动作价值和策略网络的训练, 其他设置与 Rew-PE-SAC 完全相同. 此外, 为了衡量 LR-MPO 的样本效率, 额外地运行了 SAC 算法 1 000 000 个时间步长. 实验结果记录在表 1 中, PE-SAC、Rew-PE-SAC 和 LR-MPO 在整个训练过程中的收益曲线绘制于图 3 中.

表 1 算法的最终性能

算法	Ant	HalfCheetah	Hopper	SlimHumanoid	Swimmer-v0	Walker2D
ME-TRPO	282.2±18.0	2 283.7±900.4	1 272.5±500.9	−154.9±534.3	30.1±9.7	−1 609.3±657.5
MB-MPO	705.8±147.2	3 639.0±1 185.8	333.2±1 189.7	674.4±982.2	85.0±98.9	−1 545.9±216.5
PETS	1 165.5±226.9	2 795.3±879.9	1 125.0±679.6	1 472.4±738.3	22.1±25.2	260.2±536.9
POPLIN	2 330.1±320.9	4 235.0±1 133.0	2 055.2±613.8	−245.7±141.9	37.1±4.6	597.0±478.8
MBPO	4 332.5±1 277.6	10 758.9±1 413.7	**3 279.8±455.0**	2 950.4±819.1	26.3±13.3	4 154.7±846.1
TD3	956.1±66.9	3 614.3±82.1	2 245.3±232.4	1 319.1±1 246.1	40.4±8.3	−73.8±769.0
SAC-200k	922.0±283.0	6 129.3±775.7	2 365.1±193.4	1 891.6±379.2	49.7±5.8	1 642.7±606.9
PE-SAC	4 033.5±1 480.5	**11 854.3±102.8**	2 202.6±363.5	1 436.8±490.8	26.6±25.4	2 673.8±2 264.8
Rew-PE-SAC	**4 614.4±931.1**	9 779.8±546.6	2 824.0±159.9	11 755.9±11 152.2	82.2±33.4	**4 961.9±457.8**
LR-MPO	**4 544.1±466.1**	**11 825.7±484.8**	**3 395.6±127.6**	**24 965.8±9 171.1**	**110.4±15.9**	3 857.8±2 514.5
SAC-1000K	4 994.9±719.5	10 283.8±648.4	2 990.3±214.3	29 122.5±11 129.0	86.8±6.4	5 094.0±1 371.3

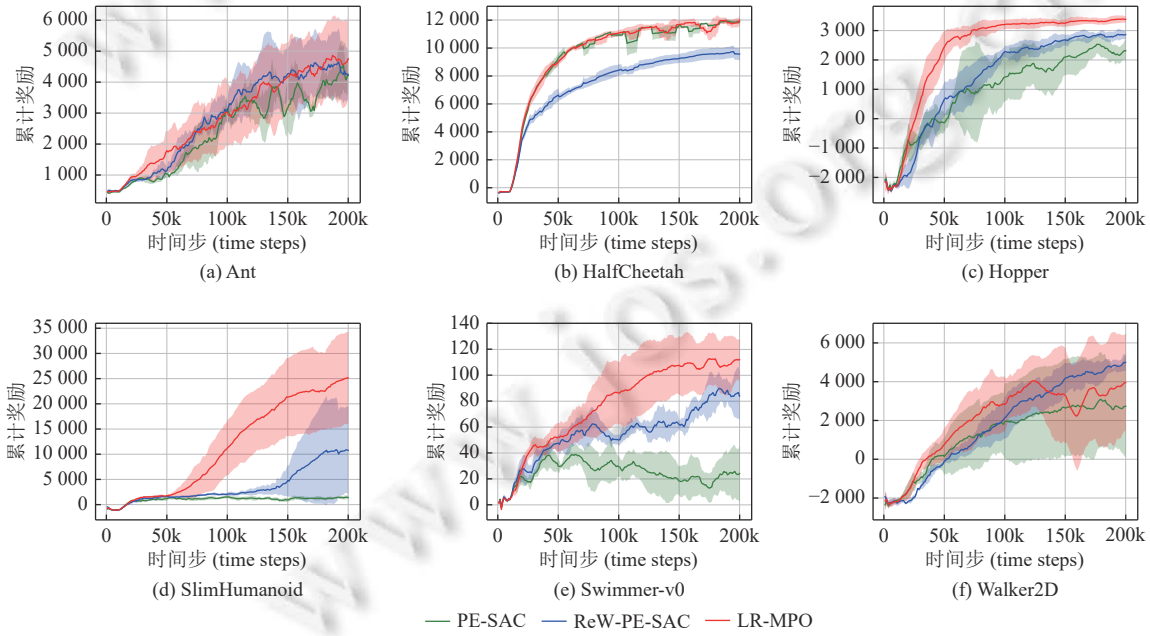

图 3 训练过程中收益的可视化曲线

如表 1 所示, LR-MPO 在 Walker2D 以外的所有环境中都具有最好的性能. 尤其是在环境 Ant、HalfCheetah、Hopper 和 Swimmer-v0 中, 运行 200 000 步的 LR-MPO 性能可以与运行步数为 1 000 000 的 SAC 相媲美, 这表明 LR-MPO 具有良好的采样效率.

将加权与不加权的方法——算法 PE-SAC 与算法 Rew-PE-SAC 和 LR-MPO 进行对比, 可以看出在大多数环境中, 加权的方法都有着更好的性能. 这表明学习到的权重预测网络为生成样本提供了适当的权重, 有效地帮助算法训练出更好的策略. 在环境 HalfCheetah 上 Rew-PE-SAC 性能次于 PE-SAC, 可能的原因是样本权重被低估所导致的.

通过观察图 3 中 Rew-PE-SAC 和 LR-MPO 的学习曲线可以发现, LR-MPO 在环境 HalfCheetah、 Hopper、SlimHumanoid 和 Swimmer-v0 中算法性能的提升速度明显快于 Rew-PE-SAC, 这进一步说明 LR-MPO 能够避免样本权重被低估, 从而避免生成样本对训练产生负面影响的同时, 保证它们被最大化地利用.

2.4 可学习的加权机制对动作价值预测损失的影响

本节将比较加权和不加权的方法下动作价值预测损失的不同. 在环境 Ant、HalfCheetah、SlimHumanoid 和 Swimmer 中运行 PE-SAC、ReW-PE-SAC 和 LR-MPO, 并记录每一轮 (1 000 个时间步) 真实样本上的平均动作价值预测损失. 多次重复实验, 记录相同时间步下损失值的最小值、最大值和平均值并绘制于图 4 中, X 轴对应时间步, Y 轴对应每 1 000 个时间步内的平均损失.

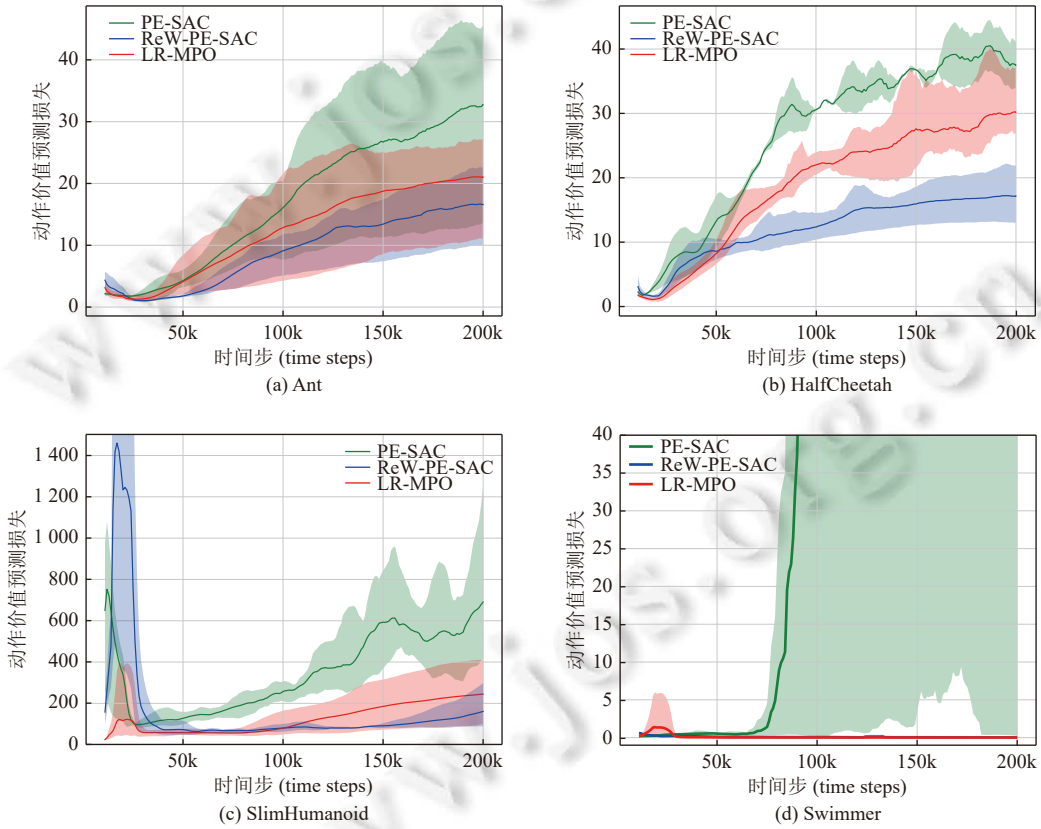

图 4 训练过程中动作价值预测损失的可视化曲线

如图 4 所示, 加权的方法在大多数情况下都维持着比不加权方法更低的损失值, 并能够有效避免出现过大的损失值. 结合环境 Swimmer 中的损失值曲线 (图 4(d)) 和学习曲线 (图 3(e)), 可以发现 PE-SAC 的性能在经历大约 70 000 个时间步长后开始下降, 动作价值预测损失也大约在此时出现明显增加. 由此可以推断, 在大多数情况下,

维持较低的损失值能够保证学习到的动作价值网络有较高的预测精度, 进而有助于学习到一个更好的策略, 提高算法的性能.

值得注意的是环境 HalfCheetah 上的结果, ReW-PE-SAC 和 LR-MPO 在该环境中都有着较低的预测损失, 但 LR-MPO 有着更好的性能. 这说明 LR-MPO 能够避免对部分样本权重的低估, 从而更充分地利用生成样本进行训练.

2.5 预测权重的变化趋势

本节将分析预测权重的总体变化趋势. 在环境 HalfCheetah 中运行 ReW-PE-SAC 和 LR-MPO 算法, 在每一轮 (1 000 时间步) 的开始时记录生成样本的权重. 预测权重是随着动作价值和策略网络的训练而不断变化, 由于随机性的存在, 每次实验的训练过程往往是不一样的, 强行将不同训练情况下的权重进行整合, 反而难以看出权重的变化趋势, 因此这里只进行一次实验. 内部优化过程中的学习率与样本权重的 25% 分位数, 中位数和 75% 分位数被记录并绘制于图 5 中.

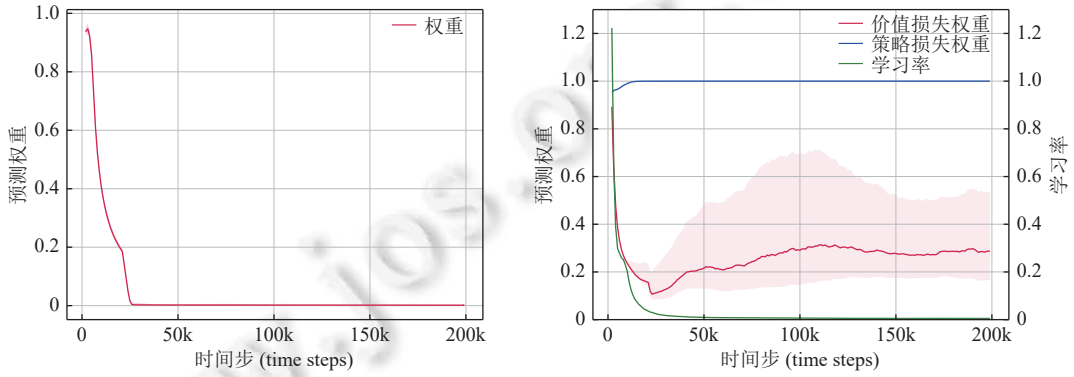

图 5 预测权重变化趋势的可视化

对比 ReW-PE-SAC 和 LR-MPO 的权重的变化趋势, 可以发现: 前者的权重在训练的中后期快速下跌到接近零, 生成样本基本无法加入训练过程, 但这并不是一个合理的加权方案, 由图 3(b) 可知, 即使全部生成样本都加入训练, 策略依然能得到改进; 而后者的权重 (包括价值函数和策略函数对应的两个预测权重) 在训练中后期依然能保持在一个相对较高的水平, 这得益于内部优化过程中的学习率从原先的权重中被解耦合出来.

3 结 语

本文提出了一种有效的基于模型的强化学习方法 LR-MPO, 该方法通过训练一个权重预测网络来自适应地调整生成样本的权重, 以减少它们对训练过程的负面影响. 重加权后的生成样本对训练过程的影响通过以下流程来量化: 使用它们对动作价值网络和策略网络进行更新, 在真实样本上计算更新前后优化目标的变化. 量化后的负面影响被用于改进权重预测网络, 即通过链式法则对更新后的优化目标求权重网络参数的导数, 然后根据梯度更新网络参数. 考虑到更新动作价值网络和策略网络时, 学习率对参数变化的影响, LR-MPO 将真实样本加入上述过程来调节该学习率. 同时, 考虑到同一个样本对价值网络和策略网络的作用通常并不相同, 所以使用两个权重预测网络分别预测样本在动作价值网络和策略网络训练过程中的适宜权重.

实验结果表明, LR-MPO 在多个复杂的连续控制任务上均获得了当前最优的性能. 学习到的权重预测网络可以在训练过程的不同阶段为不同的生成样本提供合理的权重, 将动作价值的预测损失保持在较低的水平的同时, 充分地利用生成样本进行训练.

References:

[1] Mnih V, Kavukcuoglu K, Silver D, *et al*. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529–533.

[doi: 10.1038/nature14236]

[2] Silver D, Huang A, Maddison CJ, *et al*. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529(7587): 484–489. [doi: 10.1038/nature16961]

[3] Mousavi SS, Schukat M, Howley E. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. IET Intelligent Transport Systems, 2017, 11(7): 417–423. [doi: 10.1049/iet-its.2017.0153]

[4] Shao ML, Cao E, Hu M, Zhang Y, Chen WJ, Chen MS. Traffic light optimization control method for priority vehicle awareness. Ruan Jian Xue Bao/Journal of Software, 2021, 32(8): 2425−2438. (in Chinese with English abstract) http://www.jos.org.cn/1000-9825/6191.htm [doi: 10.13328/j.cnki.jos.006191]

[5] Liang XY, Du XS, Wang GL, Han Z. A deep reinforcement learning network for traffic light cycle control. IEEE Trans. on Vehicular Technology, 2019, 68(2): 1243–1253. [doi: 10.1109/TVT.2018.2890726]

[6] Deng Y, Bao F, Kong YY, Ren ZQ, Dai QH. Deep direct reinforcement learning for financial signal representation and trading. IEEE Trans. on Neural Networks and Learning Systems, 2017, 28(3): 653–664. [doi: 10.1109/TNNLS.2016.2522401]

[7] Liang TX, Yang XP, Wang L, Han ZY. Review on financial trading system based on reinforcement learning. Ruan Jian Xue Bao/Journal of Software, 2019, 30(3): 845−864 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5689.htm [doi: 10.13328/j.cnki.jos.005689]

[8] Huang KQ, Xing JL, Zhang JG, Ni WC, Xu B. Intelligent technologies of human-computer gaming. SCIENTIA SINICA Informationis, 2020, 50(4): 540–550 (in Chinese with English abstract). [doi: 10.1360/N112019-00048]

[9] Arel I. Deep reinforcement learning as foundation for artificial general intelligence. In: Wang P, Goertzel B, eds. Theoretical Foundations of Artificial General Intelligence. Paris: Atlantis Press, 2012. 89–102.

[10] Deisenroth MP, Rasmussen CE. PILCO: A model-based and data-efficient approach to policy search. In: Proc. of the 28th Int'l Conf. on Machine Learning. Bellevue: Omnipress, 2011. 465–472.

[11] Levine S, Koltun V. Guided policy search. In: Proc. of the 30th Int'l Conf. on Machine Learning. Atlanta: JMLR.org, 2013. 1–9.

[12] Levine S, Abbeel P. Learning neural network policies with guided policy search under unknown dynamics. In: Proc. of the 27th Int'l Conf. on Neural Information Processing Systems. Montreal: MIT Press, 2014. 1071–1079.

[13] Gal Y, McAllister R, Rasmussen CE. Improving PILCO with Bayesian neural network dynamics models. In: Proc. of the 33rd Int'l Conf. on Machine Learning. JMLR.org, 2016. 25–31.

[14] Depeweg S, Hernández-Lobato JM, Doshi-Velez F, Udluft S. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: ICLR, 2017.

[15] Nagabandi A, Kahn G, Fearing RS, Levine S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In: Proc. of the 2018 IEEE Int'l Conf. on Robotics and Automation. Brisbane: IEEE, 2018. 7559–7566. [doi: 10.1109/icra.2018.8463189]

[16] Liang XX, Feng YH, Huang JC, Wang Q, Ma Y, Liu Z. Novel deep reinforcement learning algorithm based on attention-based value function and autoregressive environment model. Ruan Jian Xue Bao/Journal of Software, 2020, 31(4): 948–966 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5930.htm [doi: 10.13328/j.cnki.jos.005930]

[17] Chua K, Calandra R, McAllister R, Levine S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 4759–4770.

[18] Clavera I, Rothfuss J, Schulman J, Fujita Y, Asfour T, Abbeel P. Model-based reinforcement learning via meta-policy optimization. In: Proc. of the 2nd Annual Conf. on Robot Learning. Zürich: PMLR, 2018. 617–629.

[19] Kalweit G, Boedecker J. Uncertainty-driven imagination for continuous deep reinforcement learning. In: Proc. of the 1st Annual Conf. on Robot Learning. Mountain View: PMLR, 2017. 195–206.

[20] Heess N, Wayne G, Silver D, Lillicrap TP, Erez T, Tassa Y. Learning continuous control policies by stochastic value gradients. In: Proc. of the 28th Int'l Conf. on Neural Information Processing Systems. Montreal: MIT Press, 2015. 2944–2952.

[21] Janner M, Fu J, Zhang M, Levine S. When to trust your model: Model-based policy optimization. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2019. 12498–12509.

[22] Thrun S, Pratt L. Learning to Learn. Boston: Springer, 1998. 3–17. [doi: 10.1007/978-1-4615-5529-2_1]

[23] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: JMLR.org, 2017. 1126–1135.

[24] Xu ZW, Van Hasselt H, Silver D. Meta-gradient reinforcement learning. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 2402–2413.

[25] Zheng ZY, Oh J, Singh S. On learning intrinsic rewards for policy gradient methods. In: Proc. of the 32nd Int'l Conf. on Neural

Information Processing Systems. Montréal: Curran Associates Inc., 2018. 4649–4659.
[26] Veeriah V, Hessel M, Xu ZW, Rajendran J, Lewis RL, Oh J, Van Hasselt H, Silver D, Singh S. Discovery of useful questions as auxiliary Tasks. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2019. 9306–9317.
[27] Huang WZ, Yin QY, Zhang JG, Huang KQ. Learning to reweight imaginary transitions for model-based reinforcement learning. In: Proc. of 35th AAAI Conf. on Artificial Intelligence. AAAI, 2021. 7848–7856.
[28] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: JMLR.org, 2018. 1856–1865.
[29] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing. Doha: Association for Computational Linguistics, 2014. 1724–1734. [doi: 10.3115/v1/D14-1179]
[30] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proc. of 3rd Int'l Conf. on Learning Representations. San Diego: ICLR, 2015.
[31] Wang TW, Bao XC, Clavera I, Hoang J, Wen YM, Langlois E, Zhang SS, Zhang GD, Abbeel P, Ba J. Benchmarking model-based reinforcement learning. arXiv:1907.02057, 2019.
[32] Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: JMLR.org, 2018. 1582–1591.
[33] Kurutach T, Clavera I, Duan Y, Tamar A, Abbeel P. Model-ensemble trust-region policy optimization. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: ICLR, 2018.
[34] Wang TW, Ba J. Exploring model-based planning with policy networks. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: ICLR, 2020.

附中文参考文献:

[4] 邵明莉, 曹鹗, 胡铭, 章玥, 陈闻杰, 陈铭松. 面向优先车辆感知的交通灯优化控制方法. 软件学报, 2021, 32(8): 2425–2438. http://www.jos.org.cn/1000-9825/6191.htm [doi: 10.13328/j.cnki.jos.006191]
[7] 梁天新, 杨小平, 王良, 韩镇远. 基于强化学习的金融交易系统研究与发展. 软件学报, 2019, 30(3): 845−864. http://www.jos.org.cn/1000-9825/5689.htm [doi: 10.13328/j.cnki.jos.005689]
[8] 黄凯奇, 兴军亮, 张俊格, 倪晚成, 徐博. 人机对抗智能技术. 中国科学: 信息科学, 2020, 50(4): 540–550. [doi: 10.1360/N112019-00048]
[16] 梁星星, 冯旸赫, 黄金才, 王琦, 马扬, 刘忠. 基于自回归预测模型的深度注意力强化学习方法. 软件学报, 2020, 31(4): 948–966. http://www.jos.org.cn/1000-9825/5930.htm [doi: 10.13328/j.cnki.jos.005930]

黄文振(1992－), 男, 博士, 主要研究领域为强化学习.

张俊格(1986－), 男, 博士, 研究员, 主要研究领域为博弈决策, 强化学习, 模式识别, 人工智能.

尹奇跃(1990－), 男, 博士, 副研究员, CCF 专业会员, 主要研究领域为机器学习, 数据挖掘, 人工智能与游戏.

黄凯奇(1977－), 男, 博士, 研究员, 博士生导师, CCF 杰出会员, 主要研究领域为计算机视觉, 模式识别, 人机对抗, 视觉监控应用.