计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(11):148−156 [doi: 10.15888/j.cnki.csa.008801]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

结合向前状态预测和隐空间约束的强化学习表示算法①

项　宇[1], 秦　进[1], 袁琳琳[2]

[1](贵州大学 计算机科学与技术学院, 贵阳 550025)
[2](贵州开放大学 信息工程学院, 贵阳 550023)
通信作者: 秦　进, E-mail: jqin1@gzu.edu.cn

摘　要: 虽然深度强化学习能够解决很多复杂的控制问题, 但是需要付出的代价是必须和环境进行大量的交互, 这是深度强化学习所面临的一大挑战. 造成这一问题的原因之一是仅依靠值函数损失难以让智能体从高维的复杂输入中提取有效特征. 导致智能体对所处状态理解不足, 从而不能正确给状态分配价值. 因此, 为了让智能体认识所处环境, 提高强化学习样本效率, 本文提出一种结合向前状态预测与隐空间约束的表示学习方法 (regularized predictive representation learning, RPRL). 帮助智能体从高维视觉输入中学习并提取状态特征, 以此来提高强化学习样本效率. 该方法用前向的状态转移损失作为辅助损失, 使智能体学习到的特征包含环境转移的相关动态信息. 同时在向前预测的基础上添加正则化项对隐空间的状态表示进行约束, 进一步帮助智能体学习到高维度输入的平滑、规则表示. 该方法在 DeepMind Control (DMControl) 环境中与其他的基于模型的方法以及加入了表示学习的无模型方法进行比较, 都获得了更好的性能.

关键词: 强化学习; 表示方法; 状态转移; 隐空间约束; 连续控制; 高维度输入

引用格式: 项宇,秦进,袁琳琳.结合向前状态预测和隐空间约束的强化学习表示算法.计算机系统应用,2022,31(11):148–156. http://www.c-s-a.org.cn/1003-3254/8801.html

Reinforcement Learning Representation Algorithm Combining Forward State Prediction and Latent Space Regularization

XIANG Yu[1], QIN Jin[1], YUAN Lin-Lin[2]

[1](College of Computer Science & Technology, Guizhou University, Guiyang 550025, China)
[2](College of Information Engineering, Guizhou Open University, Guiyang 550023, China)

Abstract: Although deep reinforcement learning can solve many complex control problems, it needs to pay the cost of a large number of interactions with the environment, which is a major challenge for deep reinforcement learning. One of the reasons for this problem is that it is difficult for an agent to extract effective features from a high-dimensional complex input only by relying on the loss of value function. As a result, the agent has an insufficient understanding of the state and cannot correctly assign value to the state. Therefore, this study proposes a regularized predictive representation learning (RPRL) method combining forward state prediction and latent space constraint to make agents know the environment and improve the sample efficiency of reinforcement learning. The method helps agents to learn and extract state features from high-dimensional visual observations to improve the sample efficiency of reinforcement learning. The forward state transfer loss is used as the auxiliary loss so that the features learned by agents contain dynamic information related to environmental transition. At the same time, the state representation of latent space is regularized on the basis of forward prediction, which further helps the agent to learn the smooth and regular representation of the high-dimensional input. In

① 基金项目: 国家自然科学基金 (61562009); 贵州省科学技术基金 (黔科合基础 [2020]1Y275); 贵州省科技计划 (黔科合基础 [2019]1130 号)
收稿时间: 2022-02-12; 修改时间: 2022-03-23, 2022-04-12; 采用时间: 2022-04-13; csa 在线出版时间: 2022-07-15

DeepMind Control (DMControl) environment, the proposed method achieves better performance than other model-based methods and model-free methods with representation learning.

Key words: reinforcement learning; representation method; state transition; latent space constraint; continuous control; high dimensional input

深度强化学习的成功来自于两方面: 一是深度神经网络强大的表示能力, 使得智能体可以从图像, 声音等高维输入中学习状态特征[1]; 二是强化学习连续决策能力, 给不同的状态分配不同的价值[2]. 两种能力的结合使得深度强化学习的智能体能够像人类一样通过视觉, 声音信号进行判断、决策. 深度强化学习在一些方面取得了成功包括如 StarCraft[3] 和 DoTA2[4] 这样的策略类游戏, 通过观察仪表板操控模拟汽车[5], 操控机器人抓取物品[6] 等. 深度学习让强化学习能够处理高维度的状态输入, 给强化学习的发展带来了质的飞跃. 但是也带来了新的问题和挑战. 高维的状态如图像, 声音等, 其中包含了大量的噪声, 使得智能体难以从中提取有用的信息用于决策.

研究发现, 以低维度形式表示的状态的样本效率要明显高于高维度的表示形式[7]. 而在强化学习的应用场景中, 智能体遇到的大多数环境的状态的存在形式都是充满噪声的高维度的. 因此, 如果有一种表示学习方法能让智能体从高维状态中准确提取低维的状态表示, 就能有效提高强化学习的样本效率. 现有算法难以从高维状态中提取准确特征的原因如下: 一是深度强化学习算法只用一个值函数损失作为损失函数, 来同时训练两个不同功能的网络, 即特征提取网络和值函数网络; 二是强化学习的训练过程是一个奖励稀疏的过程, 导致对网络的更新大多数情况下是无效的[8]. 这两点叠加导致智能体难于从高维度输入中提取特征, 从而影响到智能体的样本效率.

针对这个问题, 本文提出了一种以环境模型损失作为辅助损失和对隐空间表示进行正则化约束的强化学习表示方法 (regularized predictive representation learning, RPRL). 该方法通过在状态表示的隐空间中引入向前状态转移损失, 使得隐空间中学习到的特征能够预测未来状态. 通过增加一个状态转移损失作为辅助损失的方法, 建立前后状态之间的马尔可夫特性. 相对于之前的只依靠值函数损失学习到的特征, RPRL 学习到的特征直观上更有说服力. 同时本文在特征学习过程中适当地加入正则化项, 对于在隐空间中学习到紧凑、平滑的表示是有利的[9]. 有鉴于此, 本文的主要工作如下: (1) 添加预测下一时刻状态的损失函数作为辅助损失, 使智能体学习到的表示能认识到环境状态的转移, 进而加深对环境的理解. (2) 以隐空间中前后状态之间的距离作为正则项约束状态表示, 使得学习到的表示更为紧凑. (3) 本文提出的表示学习方法稳定的应用于 SAC 上, 并在 DMControl 的连续控制任务上极大地提高了 SAC 的得分, 超过了近年来一些优秀的基于模型算法和无模型的方法.

1 相关工作

如何提高样本效率一直是强化学习的一个研究热点和难点. 在连续控制任务和离散控制任务中, 都存在这样的问题. 强化学习无论是值函数还是策略函数都是以状态作为自变量, 而智能体难以从高维输入中提取到相应的状态特征. 多数强化学习环境是稀疏的奖励环境, 缺少更新信号, 导致策略的迭代缓慢. 这两点共同导致了强化学习的样本效率低.

为了解决这个问题, 一个常用的方法是在原始的无模型强化学习基础上添加一个辅助损失作为表示学习损失, 帮助智能体快速学习到状态特征. SAC+AE[10] 通过让 SAC[11] 和自编码器[12] 结合的方式, 使得学习到的状态表示能够重构当前时刻的状态. 这样的辅助损失提高了 SAC 的表示能力, 并实现了性能的提升. CURL[13] 对智能体观察到的原始状态进行数据增强, 然后用对比学习提取原始观察值和数据增强结果之间的共同信息. 用对比学习获得的特征作为状态特征进行强化学习. 从本质上, 这两种方法都是以当前时刻状态作为自监督来学习特征的方法. 自监督学习能够学习到高维状态的丰富表示, 并且在很多领域得到了广泛的应用和取得显著的成绩[14,15]. 在强化学习领域也帮助智能体从高维复杂数据中学习有意义的内在表示. 但是这种自监督的方法是在状态自身以及状态的数据增强形式之间构建损失, 表示学习过程不涉及强化学

习的状态转移和奖励, 所以学习到的特征也有一定的局限性. 它忽略了强化学习状态的马尔可夫属性, 当前时刻状态决定下一时刻状态. 因此强化学习智能体需要的状态表示不光要求能重构当前时刻的状态, 更重要的是预测下一时刻状态. 而自监督的表示学习方法没有表现出这一重要属性.

基于模型的强化学习方法被证明能够提高样本效率. 学习一个接近完美的真实世界动态模型, 然后用模型生成虚拟的数据帮助智能体进行策略规划. Dyna[16] 是早期的基于模型的强化学习的框架, 在智能体采集样本提升策略的同时训练一个环境模型, 然后环境模型生成虚拟样本和智能体采集的真实样本都被用于改进策略. Dreamer[17], PlaNet[18] 在隐空间中进行动态转移, 环境模型损失由隐空间的动态损失, 奖励损失, 重构损失组成. SimPLe[19] 在 Atari 游戏中只通过环境模型生成的经验实现策略的有效提升. 这种基于模型的强化学习方法提升了样本效率, 减少了样本的使用, 但是却在别的地方花费了更多的资源, 比如模型构建, 样本生成, 策略规划.

本文方法有效地把环境模型和无模型的离轨策略算法 SAC 相结合. 不使用环境模型生成虚拟数据来直接改进策略, 而是利用环境模型作为表示学习方法, 帮助智能体提取具有预测能力的内在表示. 同时使用正则化方法对隐空间中学习到的表示进行限制, 从而得到合理的表示, 帮助提高强化学习算法的样本效率.

2 向前预测和隐空间约束的表示学习

强化学习智能体在马尔可夫决策过程 (MDP) 框架下和环境交互, 以寻求累积奖励最大化[20]. MDP 可以用一个元组 (S, A, P, r, γ) 定义, S 表示状态空间, A 表示智能体可以选择的动作集, P 表示环境的状态转移概率, r 是奖励函数, γ 是一个取值在 0 到 1 之间的折扣系数. 策略 π 表示智能体在当前状态执行的动作的概率分布. 从状态 s 开始执行策略 π 获得的期望累积奖励 $v_\pi(s)=\mathbb{E}_\pi(\sum_{t=0}^{+\infty}\gamma^t r_{t+1}|s_0=s)$, 表示状态 s 在策略 π 下的值函数. 策略的优化过程是一个对环境不断探索, 并使值函数最大化的过程. 而值函数是基于状态的函数, 因此状态的准确表示至关重要.

为了从高维输入中提取准确的状态表示, 本文提出的解决方法是给深度强化学习算法添加一个辅助损失, 用于表示学习, 帮助智能体增强对环境状态的理解. 本文使用的辅助损失是环境模型的状态转移损失, 该损失作用于特征提取模块, 促使获得的特征包含有效的状态信息. 所谓有效信息不仅要使后续任务变得便利, 还要具备一定的泛化能力, 不局限于特定的任务.

当我们用神经网络把高维输入信号映射到低维的隐空间时, 不能保证隐空间中的表示是规则的. 因此可以加入正则化项作为约束条件, 惩罚隐空间中相邻状态之间距离过大的表示, 使状态表示在隐空间中的排列更加规则.

本文提出的表示学习方法能改进智能体从高维输入中获得有效信息的能力. 原则上该方法可以和任何的无模型强化学习算法相结合, 达到提高无模型强化学习样本效率的目的. 本文在 SAC 算法上进行改进, 以验证我们的方法. 新的结合了表示学习的算法我们称为 SAC+RPRL. 图 1 和图 2 表明, 提升了表示学习能力的 SAC+RPRL 相较于 SAC, 极大地改进了智能体的样本效率. 在 6 个实验环境中平均性能提高 3 倍以上, 性能中位数提高了 4 倍. 更详细的实验数据见表 1. 这也说明智能体在复杂环境中需要做出决策时, 不止依靠合适的算法, 智能体对环境的认知能力也很重要. 而传统的基于奖励信号为指示的智能体对环境的认知能力显然是不足的.

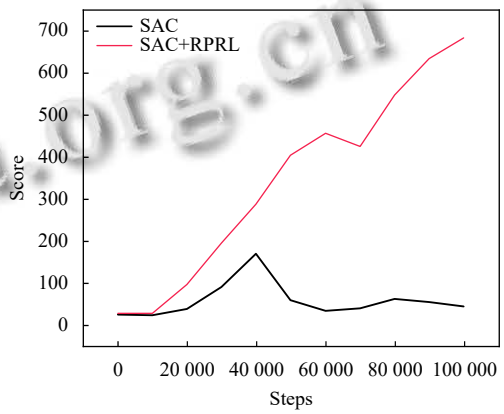


图 1 在 walker-walk 任务验证 RPRL 对 SAC 的性能提升

2.1 SAC 软策略演员评论家

SAC 是一个基于最大熵框架的离轨策略的演员-评论家算法, 演员执行动作使得累积奖励最大化的同时还要使得熵最大化, 也就是说在完成任务的同时尽可能地使动作随机. 演员学习一个参数化的策略 π_φ, 评论家学习参数化的动作值函数 Q_θ. 在每次策略迭代过程中, 智能体从经验回放池 D 中采样一个批次的 $(s_t,$

a_t, r_t, s_{t+1}) 来使得贝尔曼误差最小化:

$$J_Q(\theta)=\mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim D}[(Q_\theta(s_t,a_t)-r_t-\gamma V(s_{t+1}))^2] \quad (1)$$

其中, 目标值函数:

$$V(s_t)=(Q_{\bar{\theta}}(s_t,a_t)-\alpha\log\pi_\varphi(a_t|s_t)) \quad (2)$$

其中, 系数 α 是一个正值, 决定熵的最大化更新和值函数更新之间的优先级. 而$Q_{\bar{\theta}}(s_t,a_t)$的参数$\bar{\theta}$由动作值函数 $Q_\theta(s_t, a_t)$ 的参数 θ 通过指数移动平均方式更新得到. 评论家通过参数化的方式学习到值函数之后, 演员从策略 π_φ 执行动作来最小化以下损失:

$$J_\pi(\varphi)=\mathbb{E}_{a_t\sim\pi}[\alpha\log\pi_\varphi(a_t|s_t)-Q_\theta(s_t,a_t)] \quad (3)$$

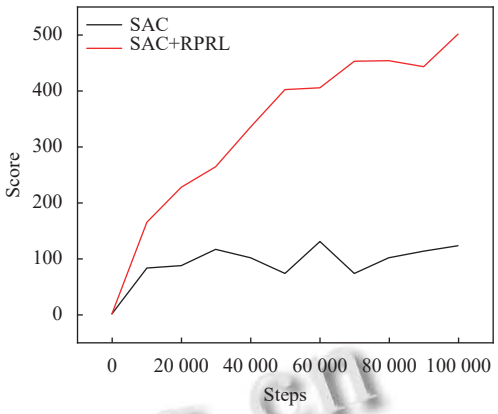


图 2 在 cheetach-run 任务验证 RPRL 对 SAC 的性能提升

表 1 SAC+RPRL 在 5 个随机种子下运行获得的得分的均值和标准差与其他算法对比

100k steps score	SAC+RPRL	CURL	SAC	SAC+AE	DREAMER	PLANET
Finger-spin	**838.26±122.51**	767±56	179±66	740±64	341±70	136±216
Cartpole-swingup	**839.38±23.78**	582±146	419±40	311±11	326±27	297±39
Reacher-easy	**585.32±46.69**	538±233	145±30	274±14	314±155	20±50
Cheetah-run	**525.16±43.89**	299±48	197±15	267±24	235±137	138±88
Walker-walk	**673.43±74.36**	403±24	42±12	394±22	277±12	224±48
Ball in cup-catch	**860.20±67.99**	769±43	312±63	391±82	246±174	0±0

2.2 向前预测的表示学习

如果智能体从高维状态提取到的特征不光能估计强化学习的值函数, 还能预测下一时刻的状态, 那么这样的特征所蕴含的信息一定是丰富的, 足以反映当前状态的内在含义. 这种蕴含丰富信息的特征相较于之前只单独依靠值函数损失学习到的简单的特征, 能有效提高智能体的样本效率. 为了获得规则的, 具有预测能力的表示, 本文提出的模型结构主要由 3 部分组成. 其中向前的状态转移模型让表示向量获得预测能力, 双编码器和正则化项让隐空间中的表示更规则、紧凑.

2.2.1 向前的状态转移模型

在强化学习任务中时序信息是非常重要的, 状态转移过程贯穿整个智能体学习过程. 以往的自监督表示学习方法完全不能够体现强化学习的马尔科夫特性, 让智能体学习到的表示具有预测下一刻状态的能力. 考虑到这一点, 本文通过状态转移模型, 建立前后状态之间的联系. 具体的做法是, 在隐空间创建向前的状态转移模型 $f_m:(h_t,a_t)\to\hat{s}_{t+1}$, 以当前时刻的状态表示 h_t 和动作 a_t 作为输入, 预测下一时刻的状态$\hat{s}_{t+1}$, 模型参数为 θ_m. 预测的状态应尽可能地与真实状态一致, 用它和真实的状态的均方误差作为预测损失:

$$J_{\text{model}}(\theta_e,\theta_m)=MSE(\hat{s}_{t+1},s_{t+1}) \quad (4)$$

其中, θ_e 为在线编码器参数, 在第 2.2.2 节介绍.

2.2.2 双编码器

表示学习的主要工作通过构造损失对隐空间中的特征向量施加影响. 而隐空间中的特征向量是由编码器映射得到, 因此最终特征向量的表示能力由编码器的映射能力作为支撑. 我们构造一个在线编码器 f_e 实现从高维的状态到低维特征的映射$h_t=f_e(s_t)$, h_t 表示隐空间中的特征向量. 使用目标编码器提高算法的稳定性, 创建一个目标编码器$f_t:s_{t+1}\to h_{t+1}$, f_t 的参数 θ_t 由 f_e 的参数 θ_e 通过指数移动平均的方式更新:

$$\theta_t=\beta\theta_t+(1-\beta)\theta_e \quad (5)$$

其中, $\beta\in[0, 1]$ 为滑动系数. 目标编码器不通过梯度来更新.

2.2.3 隐空间正则化项

深度学习会把原始输入降维到较低维度的空间中, 以实现降低噪声, 提取任务特征的目的. 一个紧凑的隐空间有利于后续任务的收敛. 通过对隐空间中特征的正则化可以实现编码器的权重衰减. 强化学习由于具有马尔科夫特性, 前后相邻两个状态之间高度相关, 处于前位的状态有概率转移到后位的状态. 这也意味着在较小的时间步内, 状态变化的差异不大. 所以在隐空间中前后两个状态的表示也应该是邻近的. 基于这种状态特征, 本文用隐空间中相邻两个状态之间的欧几里得距离 $dist(h_t, h_{t+1})$ 作为正则项, 迫使相邻两个状态

的表示在隐空间中的位置更加紧密. 用在线编码器提取 t 时刻的状态特征$h_t = f_e(s_t)$, 目标编码器提取 t+1 时刻的状态特征$h_{t+1} = f_t(s_{t+1})$.

$$J_{\text{reg}}(\theta_e) = dist(h_t, h_{t+1}) = \|h_t - h_{t+1}\|_2 \tag{6}$$

表示学习部分总的损失为:

$$J_{\text{repres}} = J_{\text{model}} + \lambda J_{\text{reg}} \tag{7}$$

2.3 前向预测的表示与强化学习联合学习

本文的方法把环境模型向前预测的表示学习和无模型的演员-评论家算法结合在一起. 演员网络、评论家网络和向前模型共用一个在线编码器 f_e. 本文的方法和经典的基于模型的强化学习不同点是: 1) 环境模型生成的数据不参与策略规划. 环境模型的作用是提升编码器的表示能力. 2) 环境模型和强化学习联合训练. 经典的基于模型的强化学习方法中环境模型的训练过程和智能体训练过程是分开的. 该算法通过共用一个在线编码器的方式, 把环境模型和智能体策略学习合并在一起. 离轨策略的强化学习算法存在不稳定因素, 多个损失同时作用于在线编码器会导致崩溃. 类似 SAC+AE 方法, 本文的方法在模型和智能体合并训练的时候, 阻止演员网络的梯度传到编码器, 有利于提高稳定性.

本文方法和 CURL、SAC+AE 都是表示学习和无模型强化学习联合学习的方法, 最大的不同是本文方法利用了状态的向前转移特点, 而后两种算法都是基于当前状态的自监督表示学习, 向前预测的表示学习更适合强化学习任务.

本文方法的框架结构如图 3 所示, 主要包括双编码器网络和用于预测下一个状态的环境模型. 其中在线编码器和目标编码器结构一致, 由 4 层卷积层组成, 每一层的卷积核大小为 3×3, 通道数为 32, 除了第一层的卷积核步长为 2, 其余层的卷积核步长均为 1. 每一层卷积之后使用 ReLU 作为激活函数. 在最后一层卷积之后加入一个层归一化[21] 的全连接层, 把卷积层的输出维度控制在 50, 然后用 tanh 函数激活. 环境模型由一层全连接层和 4 层逆卷积层构成. 逆卷积层的卷积核大小为 3×3, 除了最后一层的步长为 2, 其余层的步长均为 1. 除了最后一层逆卷积层输出结果, 其余各层之后都用 ReLU 作为激活函数.

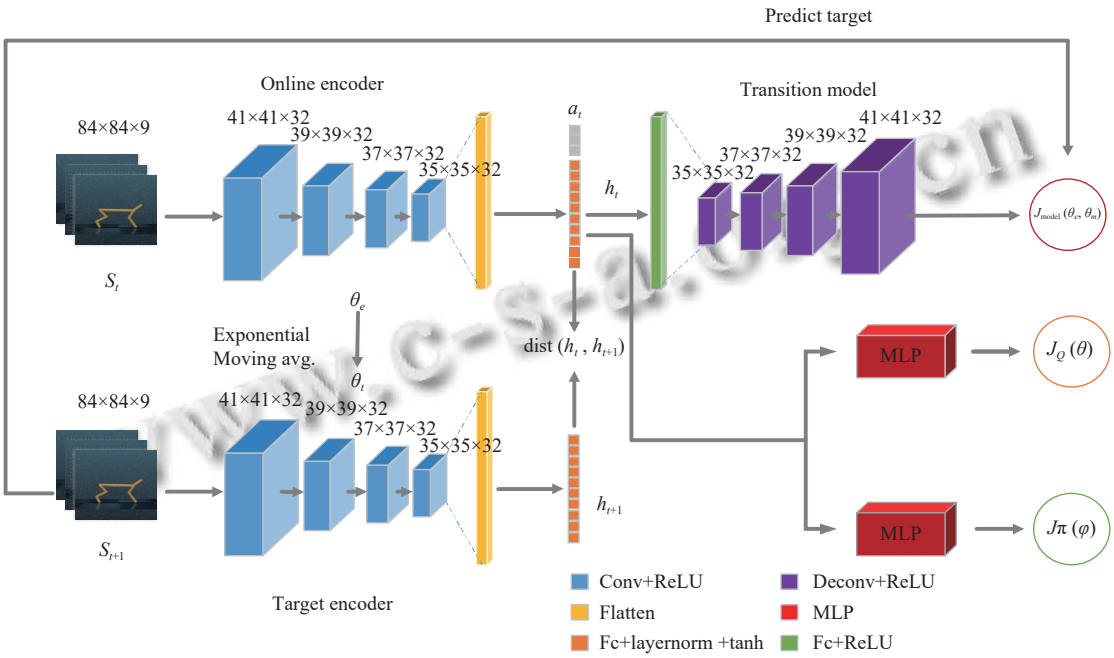


图 3 本文方法的框架结构

智能体和环境交互观察到 3 个连续的 RGB 图像并堆叠在一起作为输入, 表示状态的空间信息和时序信息. 在状态输入经过编码器提取到状态特征之后和当前时刻的动作进行拼接, 作为环境模型的输入, 以此预测下一时刻的状态.

本文提出的表示学习算法 SAC+RPRL 总结为算法 1.

算法 1. 结合向前状态预测和正则化表示学习 (SAC+RPRL) 的伪代码

初始化参数 $\theta, \varphi, \theta_e, \theta_t, \theta_m$
初始化经验回放池 D
for each iteration **do**
  for each environment step **do**
    $a_t \sim \pi_\varphi(a_t|s_t)$ //根据当前策略执行动作
    $s_{t+1}, r_t \sim p(s_{t+1}, r_t|s_t, a_t)$ //与环境交互获得下一时刻状态和奖励
    $D \leftarrow D \cup (s_t, a_t, r_t, s_{t+1})$ / /经验样本存入经验回放池
  end for
  for each gradient step **do**
    $h_t \leftarrow f_e(s_t)$ //在线编码器提取当前状态特征
    $h_{t+1} \leftarrow f_t(s_{t+1})$ //目标编码器提取下一时刻状态特征
    $(\theta_e, \theta_m) \leftarrow (\theta_e, \theta_m) - \eta_m \nabla_{(\theta_e, \theta_m)} J_{\text{model}}(\theta_e, \theta_m)$ //更新模型和在线编码器
    $\theta_e \leftarrow \theta_e - \lambda \eta_e \nabla_{\theta e} J_{\text{reg}}(\theta_e)$ // 约束在线编码器网络
    $\theta \leftarrow \theta - \eta_Q \nabla_\theta J_Q(\theta)$ // 更新 Q 函数
    $\varphi \leftarrow \varphi - \eta_\pi \nabla \varphi J_\pi(\varphi)$ // 更新策略
    $\theta_t \leftarrow \beta \theta_t - (1-\beta)\, \theta_e$ // 更新目标编码器网络
  end for
end for

3 实验结果与分析

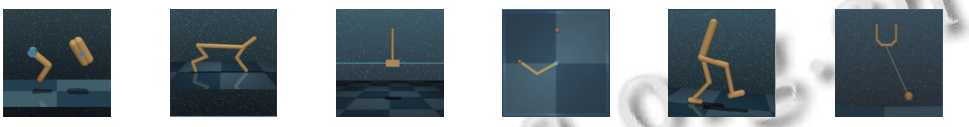
3.1 算法评价

本文在 DMControl100K 基准中评价算法的性能和样本效率. 通过与环境交互 100k 步, 比较该算法与基线算法的最终得分, 来评价算法的性能; 通过测量其他基线算法获得该算法在 100k 步时得分, 所需要消耗的时间步来评价该算法的样本效率.

3.2 实验环境和对比的基线算法

DeepMind Control 由一系列复杂的视觉连续控制任务基准环境组成. 根据奖励函数的不同, 在同一个环境中智能体可以完成不同的任务. 环境名由两部分表示, 如 walker-walk, 其中 walker 表示智能体所处环境, walk 表示智能体在这个环境中需要学习完成的任务. 本文采用如图 4 所示的 6 个连续控制任务作为实验环境. 智能体和环境交互获得的反馈是 3×84×84 的 RGB 图像, 类似于人类在自然环境中通过视觉信号做决策. 把连续的 3 幅图像堆叠起来, 给智能体提供时间上的变化信息. 这个基准环境的许多任务和现实中机器人控制的任务相关. 很多以视觉信号为输入的无模型的算法在这个环境中的样本效率很低, 因此近期有很多研究在这个环境中展开. 在这个环境中也提供了很多优秀的算法可以作为基线进行对比.

文本提出的 SAC+RPRL 方法和近期在 DMControl 环境中取得较好性能的一些算法进行性能和样本效率的对比. 它们是 SAC: 以图像作为输入的演员-评论家算法; SAC+AE: 使用自动编码器和 SAC 相结合来提高原有算法的性能; PlaNet 和 Dreamer: 是基于模型的强化学习算法, 通过学习一个世界模型, 然后用模型生成的经验进行策略规划; CURL: 用对比学习的方法, 在原始数据和增强数据之间进行对比学习, 从而提取共同的相似的特征用于无模型的强化学习算法 SAC.



(a) Finger-spin (b) Cheetah-run (c) Cartpole-swingup (d) Reacher-easy (e) Walker-walk (f) Ball in cup-catch

图 4 本文的实验中将会用到的 6 个 DeepMind Control 环境

3.3 实验结果

3.3.1 性能对比

SAC+RPRL 每个环境运行采用不同的随机种子运行 5 次, 获得的结果计算均值和方差. 表 1 展示了 SAC+RPRL 和基线算法在 100k 步交互次数下, 在 6 个不同的连续控制任务中的性能对比. SAC+RPRL 都取得了最高的得分. SAC+RPRL 获得的分数高于用对比学习提取特征的 CURL 和用自动编码器提取特征的 SAC+AE, 也说明基于向前状态转移的特征提取方式相较于自监督的提取方式, 更适合强化学习任务.

图 5 展示了不同算法在表 1 展示的 6 个环境中的得分中位数, SAC+RPRL 是对比表示学习算法 CURL 的 1.35 倍, 基于模型算法 Dreamer 的 2.55 倍.

3.3.2 样本效率对比

图 6 中横坐标为时间步数, 展示的内容是为了获得 SAC+RPRL100K 步时的得分, CURL 需要消耗的时间步数. 平均计算下来, SAC+RPRL 的样本效率是 CURL 的 2.7 倍.

3.3.3 表示方法的泛化能力验证

本文提出的表示学习方法基于环境的转移过程, 主要涉及的是状态的变化, 不涉及奖励函数. 正因为没有预测环境的奖励, 使得这样的表示更加灵活. 学习到的表示适用于同一环境的不同任务. 为了验证这一点, 把 walker-walk 任务中学习到的表示, 迁移到 walker-

run, walker-stand 任务中, 这 3 个任务是相同的环境, 但是有着不同的奖励函数.

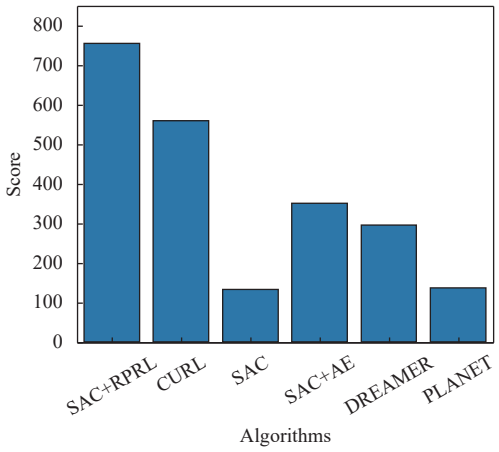


图 5 不同算法在 6 个连续控制任务的中位数得分对比

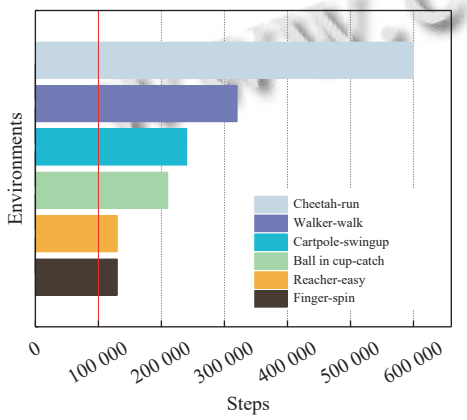


图 6 SAC+RPRL 和 CURL 的样本效率对比

图 7 和图 8 展示的是分别在 walker-stand 和 walker-run 任务中训练一个 SAC 智能体作为比较, SAC+PRETRAIN 智能体的编码器使用的参数是 SAC+RPRL 在 walker-walk 任务中学习到的编码器参数. 从实验结果可以看出 SAC+RPRL 学习到的表示有一定的泛化能力, 能够适应同一环境的不同任务. 也说明 SAC+RPRL 的表示方法学习到了环境的动态特征, 而这种动态特征是环境固有的, 不受限制于任务的奖励. 因此 SAC+RPRL 不仅能够提高强化学习的样本效率, 同时能在同一环境的不同任务之间进行迁移.

3.3.4 解耦表示学习和奖励信号

为了解正则化的向前预测表示部分在 SAC+RPRL 中贡献的多少. 本文阻断来自演员和评论家的梯度信号到达编码器. 这样编码器学习的环境特征完全由转移模型损失和正则项控制, 而与奖励函数无关. 图 9 对比了 SAC+RPRL 和 CURL 分别阻断奖励信号之后的变化. 可以看出 SAC+RPRL 基本不依赖于奖励函数, 有的任务甚至还有提升, 可能是由于切断奖励信号之后, 提高了编码器在更新过程中的稳定性. 由此看出, SAC+RPRL 的表示学习部分基本掌握了环境的特征, 使得智能体对所处环境有充分认识, 才能做出正确的决策. CURL 的表示学习部分在离开了奖励信号之后, 每个任务的性能都受到了不同程度的负面影响. 其中 ball in cup-catch 和 reacher-easy 受到的影响最大. 说明 CURL 这种基于自监督方法学习到的特征, 相对于状态预测的 SAC+RPRL, 不能更好地描述智能体所处的环境.

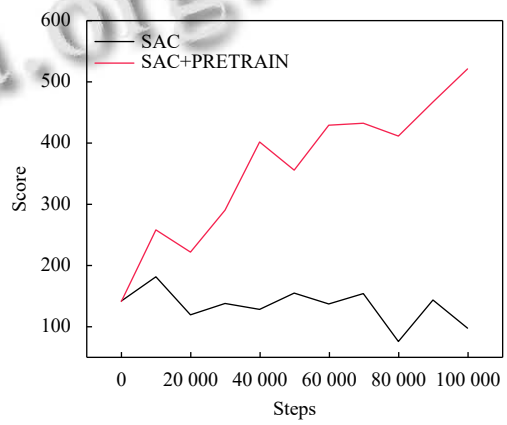


图 7 在 walker-stand 任务验证 SAC+RPRL 的表示泛化能力

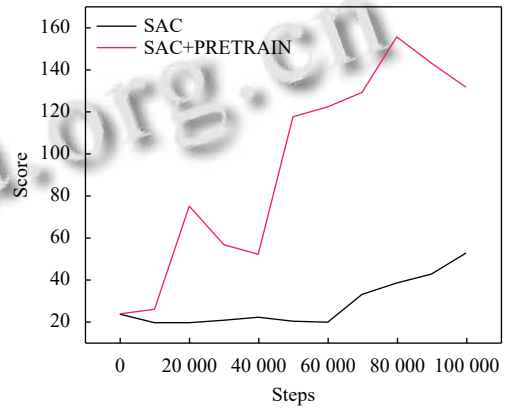


图 8 在 walker-run 任务验证 SAC+RPRL 的表示泛化能力

3.3.5 消融实验

为了分析和了解本文方法的不同部分对于算法性能的影响, 以及测试本文方法的表示能力, 文本设计了以下的消融实验.

(1) 正则化消融实验

从生成模型的角度来看, 正则化的目的是学习一个平滑的隐空间, 在这个空间中, 类似的数据点 s 被映

射到类似的隐空间表示 h, 而 h 的微小变化导致解码器的重构结果发生轻微的变化. 本文对比了 3 种不同的对于隐空间表示的处理方式, 分别是不对隐空间表示使用正则化、使用$L_2 = \|h\|_2^2$作为正则项以及采用前后相邻状态之间的欧几里得距离$dist(h_t, h_{t+1}) = \|h_t - h_{t+1}\|_2$作为正则项.

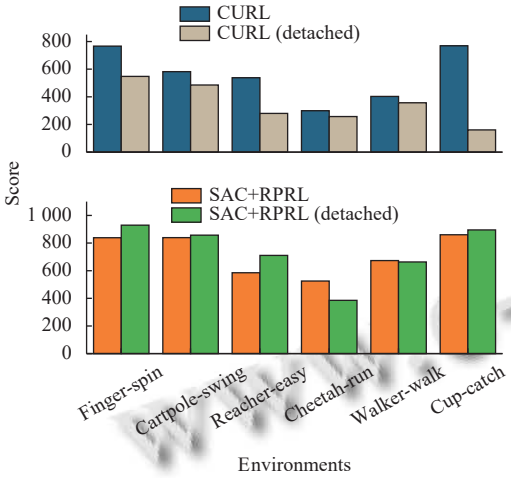


图 9 解耦表示学习和奖励信号实验对比

图 10 和图 11 分别在 finger-spin 和 ball in cup-catch 两个任务场景中展示了不同的正则化方法对于智能体性能的影响, 可以看出用前后相邻状态之间的欧几里得距离作为正则化项更适合.

(2) 目标编码器消融实验

以往的表示学习算法和强化学习的算法中都会用到目标编码器, 它对于学习到稳定的表示特征是至关重要的, 在本文中使用指数滑动平均来更新目标编码器, 滑动系数 β=0.01. 图 12 和图 13 分别在 finger-spin 和 ball in cup-catch 两个任务场景中对比了使用目标编码器和没有使用目标编码器两种模型结构, 表明使用了目标编码器能够有效提高算法的稳定性.

4 结论

想让强化学习智能体拥有人类一样的决策能力, 那么从真实世界学习的能力是必不可少. 真实世界提供的信息比如视觉信息, 声音信息都是高维的复杂信息, 其中能够影响决策的信息含量少之又少. 因此数据高效的强化学习智能体必须要有从真实世界提取有用信息的能力. 本文用辅助损失的方法, 通过向前的状态转移加强智能体对于环境的理解和认识. 用正则化约束学习到平滑、规则的状态表示. 实验证明好的表示方法能极大地提高算法的性能. 泛化实验证明了学习到的特征可以适应当前环境的不同任务.

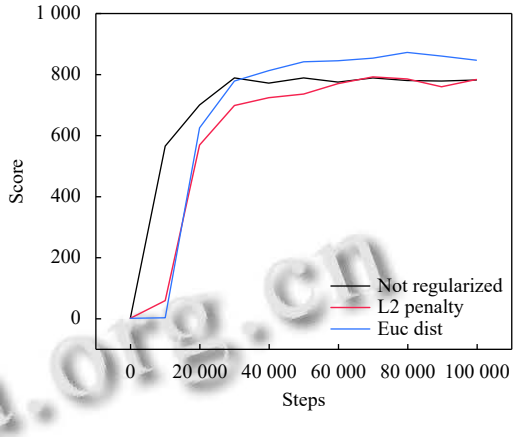


图 10 在 finger-spin 任务验证 SAC+RPRL 的正则化方法

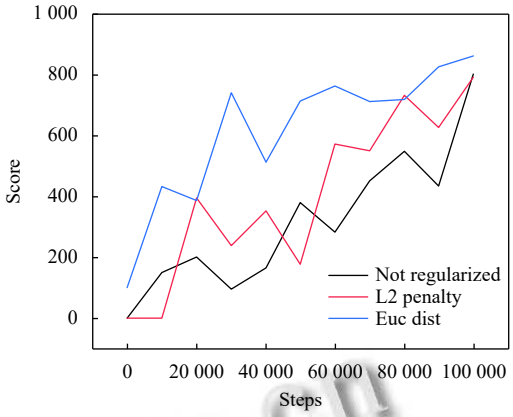


图 11 在 ball in cup-catch 任务验证 SAC+RPRL 的正则化方法

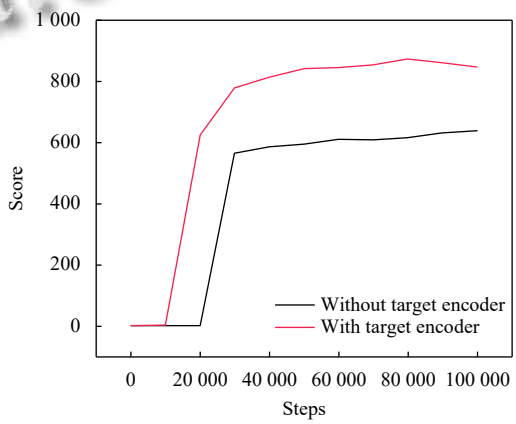


图 12 在 finger-spin 任务测试 target encoder 对算法稳定性的影响

为了提高强化学习的样本效率, 使得强化学习能大范围应用到真实场景中, 不光要改进算法本身, 同时

好的表示方法也能帮助智能体认识周围环境, 从而提升样本效率. 本文提出了一种以模型转移损失作为辅助损失结合隐空间约束的表示学习方法, 改进智能体对高维复杂输入的表征能力, 提高了强化学习样本效率. 虽然只在视觉这一高维输入情况中进行验证, 但它的应用不应局限于此, 在包括声音、文本等高维环境的强化学习问题中它的方法也值得借鉴. 后续工作将针对如何提高强化学习样本效率问题, 从表示和算法角度展开更全面, 深入的研究.

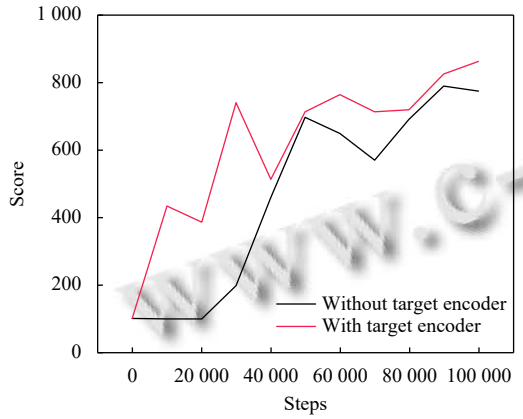


图 13 在 ball in cup-catch 任务测试 target encoder 对算法稳定性的影响

参考文献

1 Goodfellow I, Bengio Y, Courville A. 深度学习. 赵申剑, 译. 北京: 人民邮电出版社, 2017.

2 Sutton RS, Barto AG. 强化学习. 俞凯, 译. 第 2 版, 北京: 电子工业出版社, 2018.

3 Vinyals O, Babuschkin I, Czarnecki WM, *et al*. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 2019, 575(7782): 350–354. [doi: 10.1038/s41586-019-1724-z]

4 Berner C, Brockman G, Chan B, *et al*. Dota 2 with large scale deep reinforcement learning. arXiv: 1912.06680, 2019.

5 Lillicrap TP, Hunt JJ, Pritzel A, *et al*. Continuous control with deep reinforcement learning. Proceedings of the 4th International Conference on Learning Representations. San Juan, 2016.

6 Kalashnikov D, Irpan A, Pastor P, *et al*. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. arXiv: 1806.10293, 2018.

7 Tassa Y, Doron Y, Muldal A, *et al*. DeepMind Control suite. arXiv: 1801.00690, 2018.

8 杨瑞, 严江鹏, 李秀. 强化学习稀疏奖励算法研究——理论与实验. 智能系统学报, 2020, 15(5): 888–899.

9 Ghosh P, Sajjadi MSM, Vergari A, *et al*. From variational to deterministic autoencoders. arXiv: 1903.12436, 2020.

10 Yarats D, Zhang A, Kostrikov I, *et al*. Improving sample efficiency in model-free reinforcement learning from images. Proceedings of the 35th AAAI Conference on Artificial Intelligence. Online: AAAI Press, 2021. 10674–10681.

11 Haarnoja T, Zhou A, Abbeel P, *et al*. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 1861–1870.

12 Hinton GE, Zemel RS. Autoencoders, minimum description length and Helmholtz free energy. Proceedings of the 6th International Conference on Neural Information Processing Systems. Denver: Morgan Kaufmann, 1993. 3–10.

13 Laskin M, Srinivas A, Abbeel P. CURL: Contrastive unsupervised representations for reinforcement learning. Proceedings of the 37th International Conference on Machine Learning. Online: PMLR, 2020. 5639–5650.

14 Hénaff OJ, Srinivas A, De Fauw J, *et al*. Data-efficient image recognition with contrastive predictive coding. Proceedings of the 37th International Conference on Machine Learning. Online: PMLR, 2020. 391.

15 He KM, Fan HQ, Wu YX, *et al*. Momentum contrast for unsupervised visual representation learning. Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 9726–9735.

16 Sutton RS. Dyna, an integrated architecture for learning, planning, and reacting. ACM SIGART Bulletin, 1991, 2(4): 160–163. [doi: 10.1145/122344.122377]

17 Hafner D, Lillicrap TP, Ba J, *et al*. Dream to control: Learning behaviors by latent imagination. Proceedings of the 8th International Conference on Learning Representations. Addis Ababa: OpenReview.net, 2020.

18 Hafner D, Lillicrap T, Fischer I, *et al*. Learning latent dynamics for planning from pixels. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 2555–2565.

19 Kaiser Ł, Babaeizadeh M, Miłos P, *et al*. Model based reinforcement learning for Atari. Proceedings of the 8th International Conference on Learning Representations. Addis Ababa: OpenReview.net, 2020.

20 高阳, 陈世福, 陆鑫. 强化学习研究综述. 自动化学报, 2004, 30(1): 86–100.

21 Ba JL, Kiros JR, Hinton GE. Layer normalization. arXiv: 1607.06450, 2016.

(校对责编: 牛欣悦)