CN 43-1258/TP
ISSN 1007-130X

计算机工程与科学
Computer Engineering & Science

第 46 卷第 11 期 2024 年 11 月
Vol. 46, No. 11, Nov. 2024

文章编号:1007-130X(2024)11-2053-10

基于 D2GA 的逆强化学习算法*

段成龙,袁　杰,常乾坤,张宁宁
(新疆大学电气工程学院,新疆 乌鲁木齐 830017)

摘　要:针对传统生成对抗逆强化学习存在的专家样本获取困难以及生成样本利用率低的问题,提出一种基于事后经验回放策略 HER 的双鉴别器生成对抗 D2GA 逆强化学习算法。在该算法中,HER 自动合成类专家的正样本,通过 D2GA 与强化学习方法柔性动作-评价 SAC 生成的负样本进行对抗性训练,基于所求解的最优奖励函数,利用 SAC 求解最优策略。将所提出的 D2GA 算法与经典的逆强化学习算法在 Fetch 机械臂环境中的 4 种任务进行了比较实验。结果表明:在没有可用演示数据的情况下,D2GA 在相对少的回合数内完成任务的成功率可以达到理想性能,优于当前流行的逆强化学习算法。

关键词:深度强化学习;事后经验回放;逆强化学习;生成对抗网络
中图分类号:TP183　　**文献标志码**:A
doi:10.3969/j.issn.1007-130X.2024.11.018

Inverse reinforcement learning algorithm based on D2GA

DUAN Cheng-long, YUAN Jie, CHANG Qian-kun, ZHANG Ning-ning
(School of Electrical Engineering, Xinjiang University, Urumqi 830017, China)

Abstract: Aiming at the difficulty in obtaining expert demonstrations and the low utilization rate of generated samples in the traditional generative adversarial reinforcement learning, a double discriminator generative adversarial (D2GA) inverse reinforcement learning algorithm based on hindsight experience replay (HER) is proposed. In this algorithm, HER automatically synthesizes positive expert-like samples, and conducts adversarial training with negative samples generated by D2GA and reinforcement learning algorithm soft actor-critic (SAC). Based on the solved optimal reward function, SAC is used to solve the optimal strategy. The proposed D2GA algorithm is compared with the classical inverse reinforcement algorithm on four tasks in the Fetch environment. The results show that the success rate of D2GA in completing the task in relatively few rounds can reach ideal performance without available demonstration data, which is better than the current popular inverse reinforcement learning algorithm.

Key words: deep reinforcement learning; hindsight experience replay; inverse reinforcement learning; generative adversarial network

1　引言

强化学习 RL(Reinforcement Learning)[1] 是机器学习中的一个强大范例,它使智能体能够做出一系列决策以实现特定目标。强化学习从行为心理学中汲取灵感,专注于通过与环境的互动来学习最佳行动。在强化学习中,智能体采取行动并通过接收奖励或惩罚的形式来与环境进行交互。智能体的目标是学习一种策略,该策略可以选择每个状态下的行动,这种行动可以随着时间的推移最大化地累积奖励,但是,该学习过程涉及在探索和利用

* **收稿日期**:2023-09-12;**修回日期**:2024-02-20
基金项目:国家自然科学基金(62263031);新疆维吾尔自治区自然科学基金(2022D01C53)
通信地址:830017 新疆乌鲁木齐市新疆大学电气工程学院
Address: School of Electrical Engineering, Xinjiang University, Urumqi 830017, Xinjiang, P. R. China

之间找到平衡。在常规强化学习算法中,即时奖励需要人为或环境给定,然而在面临复杂任务时,奖励函数的构建较困难:一方面,在面对诸如机械臂抓取任务的环境时,在完成最终抓取前,获得的回报信息较少;另一方面,人为对奖励函数进行准确设计较困难,并且易受到主观判断与个人经验的制约。更重要的是,由于不同奖励函数对应的最优策略不同,如果即时奖励设定不合适,将导致强化学习算法难以收敛[2]。以安全高效驾车为例,一个新手司机凭借自身的经验和常识难以直接完成这个任务,但是通过学习其他经验丰富的司机则可以完成这个任务。也就是说,可以通过恢复"专家"对应的奖励函数,然后利用它来生成令人满意的行为,这种方法就是逆强化学习 IRL(Inverse Reinforcement Learning)[3]。

目前,IRL 的应用场景涵盖了多个领域,其中包括自动驾驶、机器人导航、机器人作业、金融交易、能源管理、智能交通系统等。本文主要涉及机器人领域。在机器人领域,IRL 通过观察人类专家的行为来学习各种任务,例如机器人导航、操作或协作。通过模仿人类专家的决策,机器人可以更自然地执行任务,适应不同的环境和情境。

与专注于如何在环境中采取行动以最大限度地累积预先指定的奖励的强化学习相反,逆强化学习专注于从观察到的行为中学习马尔可夫决策过程 MDP(Markov Decision Process)中主体的奖励函数[4]。Abbeel 等[5]考虑在马尔可夫决策过程中,通过观察专家演示想要学习的任务,在不断迭代优化的内部循环中求解 MDP,该迭代可以找到一个性能无限接近专家策略的策略,以此来恢复专家的奖励函数。在深度学习快速发展的推动下,Wulfmeier 等[6]利用深度神经网络来近似奖励函数,进一步提高了 IRL 的灵活性,并消除了手动设计奖励函数的需要。通常情况下,采用逆强化学习方法的训练策略需要大量或高质量的专家演示数据。在某些训练场景中,获得专家演示并非易事[7]。为了实现模仿学习算法在较少的演示数据下良好工作的目标,研究人员已经做了大量的工作,如元学习框架[8]、神经任务编程[9]等。在最近的工作中,Liu 等[10]提出了自模仿学习方法来训练策略,以在没有外部演示的情况下再现智能体过去的良好体验。但是,这些方法只适用于低维的环境,具有一定的局限性。

事后经验回放策略 HER(Hindsight Experience Replay)[11]被提出用于处理强化学习中的稀疏奖励。HER 的关键机制是,即使在没有获得有价值奖励的失败回合中,智能体也可以通过假设在回合中看到的状态是实际目标来将其转化为成功的回合。由此得到启发,利用 HER 的这一机制,专家数据可以从失败的样本中自我合成,这样便可节省收集专家数据的人工成本和时间成本。

Fu 等[12]和 Ho 等[13]将 RL、IRL 与生成对抗性网络 GAN(Generative Adversarial Network)联系起来,在图像生成、视频预测和机器翻译领域取得了显著成功。在这 2 个方法中,IRL 被解释为 GAN 鉴别器,其目标为确定经验的来源是提取自专家样本还是生成自 RL 步骤。GAN 生成器通过 RL 步骤实现,并生成无法通过 IRL 区分的负样本。生成对抗性模仿学习 GAIL(Generative Adversarial Imitation Learning)[13]表明,RL 和 IRL 的迭代过程产生的策略优于行为克隆 BC(Behavior Clone)。然而,由于一般 RL 算法简单地根据由 IRL 步骤计算的奖励来训练策略,故 GAIL 被认为是样本低效的。

为了提高 RL 步骤的采样效率,Jena 等[14]在 GAIL 生成器的损耗中增加了 BC 损耗。Kinose 等[15]将 GAIL 鉴别器与强化学习相结合,使用鉴别器计算的原始奖励和额外奖励来训练策略。然而,由于鉴别器的结构是独立于生成器设计的,很难将 IRL 步骤的结果应用于 RL 步骤,反之亦然,故该方法的样本效率仍然较低。

针对样本低效问题,蔡钺等[16]将近端策略优化算法和生成对抗网络相结合,以提高样本的效率。Zhang 等[17]提出了双鉴别器生成对抗网络 D2GA(Double Discriminator Generative Adversarial)模型,解决了常规生成对抗网络难以训练、训练不稳定的问题。陆彦辉等[18]构建了一种多鉴别器生成对抗网络模型,利用小规模数据集合成得到与真实数据分布相似的时间序列数据集。这些改进都在提升生成样本效率方面做出了突出的贡献。

为了进一步提高样本生成效率,本文将逆强化学习与双鉴别器生成对抗网络相结合,提出了一种无模型的逆强化学习算法,称为 D2GA 逆强化学习算法。该算法通过训练 2 个二进制鉴别器来计算专家样本分布和非专家样本分布的对数比:第 1 个鉴别器是状态鉴别器,它将非专家产生的状态与专家的状态区分开来;第 2 个鉴别器是与当前状态、当前动作和下一时间步状态相关的函数。双鉴别器生成对抗网络不仅可以有效地利用生成的样本,而且克服了单鉴别器训练不稳定的问题,提高

了算法收敛速度和生成的策略质量。

2 相关理论

2.1 熵正则化马尔可夫决策过程

在经典 MDP 中,智能体通过采取行动,在状态之间转换来与环境交互,并根据其行动获得奖励。MDP 的目的是学习一种能随着时间的推移最大化预期累积奖励的策略。

将 S 和 A 分别设为连续的状态空间和离散的动作空间。在第 t 个时间步,智能体观察环境当前状态 $\boldsymbol{s}\in S$,并执行根据随机策略 $\pi(\boldsymbol{a}\mid\boldsymbol{s})$ 采样的动作。随后,环境给出了即时奖励 $r(\boldsymbol{s},\boldsymbol{a})$,并且通过执行 $\boldsymbol{a}$ 的动作,基于从 $\boldsymbol{s}$ 到下一时间步的状态 $\boldsymbol{s}'$ 的状态转换概率 $p_{\mathrm{T}}(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})$ 进行状态转换。强化学习的目标是构造一个最优策略 $\pi^{*}(\boldsymbol{a}\mid\boldsymbol{s})$,使给定的目标函数最大化。目前最广泛使用的目标函数如式(1)所示:

$$V(\boldsymbol{s})=E\left[\sum_{t=0}^{\infty}\gamma r(\boldsymbol{s},\boldsymbol{a})\right] \tag{1}$$

其中,$\gamma\in[0,1)$ 为折扣因子,$E(\cdot)$ 表示数学期望。强化学习通过获得最佳状态价值函数来选取最优策略 $\pi^{*}(\boldsymbol{a}\mid\boldsymbol{s})$。MDP 的最优状态价值函数 $V_{\mathrm{old}}^{*}(\boldsymbol{s})$ 如式(2)所示:

$$V_{\mathrm{old}}^{*}(\boldsymbol{s})=\max_{\boldsymbol{a}}\left[r^{*}(\boldsymbol{s},\boldsymbol{a})+\gamma E_{\boldsymbol{s}'\sim p_{\mathrm{T}}(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})}\left[V(\boldsymbol{s}')\right]\right] \tag{2}$$

其中,$r^{*}(\boldsymbol{s},\boldsymbol{a})$ 表示最优奖励函数,$E_{\boldsymbol{s}'\sim p_{\mathrm{T}}(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})}[V(\boldsymbol{s}')]$ 表示 $\boldsymbol{s}'$ 根据状态转换概率 $p_{\mathrm{T}}(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})$ 进行状态转换以获得累计状态价值的数学期望。式(2)也被称为 Bellman 最优方程[19]。

在许多现实场景中,探索(尝试新的行动来收集有关环境的信息)对于做出明智的决策至关重要。熵正则化 MDP 通过引入正则化熵项来解决经典 MDP 中面临的探索和利用困境[20],该项鼓励智能体探索新的行动和状态,同时仍以最大化奖励为目标。正则化熵项惩罚智能体采取不确定或不太熟悉的行动,在获取新知识和利用已知的奖励行动之间取得平衡。Kozuno 等[21]对熵正则化 MDP 进行扩展,将熵项引入奖励函数中,其正则化形式如式(3)所示:

$$\tilde{r}(\boldsymbol{s},\boldsymbol{a})=r(\boldsymbol{s},\boldsymbol{a})+\frac{1}{k}H(\pi(\boldsymbol{a}\mid\boldsymbol{s}))-\frac{1}{\eta}KL(\pi(\boldsymbol{a}\mid\boldsymbol{s})\,||\,b(\boldsymbol{a}\mid\boldsymbol{s})) \tag{3}$$

其中,$r(\boldsymbol{s},\boldsymbol{a})$ 是在 IRL 设置中未知的标准奖励函数。k 和 η 是正超参数,$H(\pi(\boldsymbol{a}\mid\boldsymbol{s}))$ 是策略 $\pi(\boldsymbol{a}\mid\boldsymbol{s})$ 的(微分)熵,$KL(\pi(\boldsymbol{a}\mid\boldsymbol{s})\,||\,b(\boldsymbol{a}\mid\boldsymbol{s}))$ 是相对熵,也称为 $\pi(\boldsymbol{a}\mid\boldsymbol{s})$ 和基线策略 $b(\boldsymbol{a}\mid\boldsymbol{s})$ 之间的 KL(Kullback-Leibler)散度。当奖励函数由熵函数正则化时,可以通过使用拉格朗日乘子的方法,最大化式(1)的等号右边。因此,熵正则化 MDP 的最佳状态价值函数 $V_{\mathrm{new}}^{*}(\boldsymbol{s})$ 可以表示为式(4)所示:

$$V_{\mathrm{new}}^{*}(\boldsymbol{s})=\frac{1}{\beta}\ln\int\exp(\beta Q(\boldsymbol{s},\boldsymbol{a}))\mathrm{d}\boldsymbol{a} \tag{4}$$

$$\beta\triangleq\frac{k\eta}{k+\eta} \tag{5}$$

$$Q(\boldsymbol{s},\boldsymbol{a})=r(\boldsymbol{s},\boldsymbol{a})+\frac{1}{\eta}\ln b(\boldsymbol{a}\mid\boldsymbol{s})+\gamma E_{\boldsymbol{s}'\sim p_{\mathrm{T}}(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})}\left[V(\boldsymbol{s}')\right] \tag{6}$$

其中,β 由正超参数定义,$Q(\boldsymbol{s},\boldsymbol{a})$ 是最优状态-动作值函数。

相应的最优策略如式(7)所示:

$$\pi^{*}(\boldsymbol{a}\mid\boldsymbol{s})=\frac{\exp(\beta Q(\boldsymbol{s},\boldsymbol{a}))}{\exp(\beta V_{\mathrm{new}}^{*}(\boldsymbol{s}))} \tag{7}$$

其中,$\exp(\beta V_{\mathrm{new}}^{*}(s))$ 表示 $\pi(\boldsymbol{a}\mid\boldsymbol{s})$ 的归一化常数。

2.2 生成对抗网络

标准 GAN 由一个生成器和一个鉴别器组成。假设 $p^{\mathrm{E}}(\boldsymbol{z})$ 和 $p^{\mathrm{L}}(\boldsymbol{z})$ 分别表示专家和生成器在数据 $\boldsymbol{z}$ 上的概率分布。

鉴别器是区分非专家样本和专家样本的函数,用 $D(\boldsymbol{z})$ 表示,并使如式(8)所示的负对数似然 NLL(Negative Log Likelihood)最小化:

$$J_{\mathrm{GAN}}^{(D)}=-E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}^{\mathrm{L}}}\left[\ln D_{\mathrm{GAN}}(\boldsymbol{z})\right]-E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}^{\mathrm{E}}}\left[1-\ln D_{\mathrm{GAN}}(\boldsymbol{z})\right] \tag{8}$$

最优鉴别器 $D_{\mathrm{GAN}}^{*}(\boldsymbol{z})$ 如式(9)所示[22]:

$$D_{\mathrm{GAN}}^{*}(\boldsymbol{z})=\frac{p^{\mathrm{L}}(\boldsymbol{z})}{p^{\mathrm{L}}(\boldsymbol{z})+p^{\mathrm{E}}(\boldsymbol{z})} \tag{9}$$

GAIL 是用于模仿学习的 GAN 的扩展,其目标函数 $V_{\mathrm{GAIL}}(\boldsymbol{\omega}_{\mathrm{G}},\boldsymbol{\omega}_{\mathrm{D}})$ 如式(10)所示:

$$V_{\mathrm{GAIL}}(\boldsymbol{\omega}_{\mathrm{G}},\boldsymbol{\omega}_{\mathrm{D}})=-E_{(\boldsymbol{s},\boldsymbol{a})\sim\pi^{\mathrm{L}}}\left[\ln D_{\mathrm{GAIL}}(\boldsymbol{s},\boldsymbol{a})\right]-E_{(\boldsymbol{s},\boldsymbol{a})\sim\pi^{\mathrm{E}}}\left[\ln(1-D_{\mathrm{GAIL}}(\boldsymbol{s},\boldsymbol{a}))\right]+\lambda_{\mathrm{GAIL}}H(\pi^{\mathrm{L}}) \tag{10}$$

其中,λ_{GAIL} 是正超参数。添加熵项是与熵正则化 MDP 关联的关键。GAIL 鉴别器的目标函数基本上与 GAN 鉴别器的相同。

AIRL(Adversarial Inverse Reinforcement

Learning)[12]采用了一种特殊的鉴别器结构,如式(11)所示:

$$D(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')=\frac{\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})}{\exp(f(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}'))+\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})} \tag{11}$$

其中,$\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})$表示非专家策略,$f(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$由2个状态相关函数$q(\boldsymbol{s})$和$o(\boldsymbol{s})$定义,其表达式如式(12)所示:

$$f(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')\triangleq q(\boldsymbol{s})+\gamma o(\boldsymbol{s}')-o(\boldsymbol{s}) \tag{12}$$

2.3 逆强化学习

在很多实际任务中,专家完成任务的序列拥有相对较高的累积奖励。当专家在完成复杂任务时,可能未考虑奖励函数,但这并不意味着专家在完成任务时就没有奖励函数。相反,专家在完成实际任务时具有潜在的奖励函数。专家在完成某项具体任务时,其决策往往是最优或接近最优的。可以假设,当所有的策略产生的累积奖励期望均没有专家策略产生的累积奖励期望大时,所对应的奖励函数就是根据示例学到的奖励函数。虽然IRL以MDP为基础,但奖励函数却是未知的,因此IRL可以定义为从专家演示中学习奖励函数的过程。给定一组由专家演示轨迹组成的集合,每一条专家演示轨迹均包括了一个状态动作对的集合,而IRL的目标就是从专家演示中发现其背后奖励函数的架构。

Ziebart等[20]认为,IRL可以扩展到具有大特征空间的模型,这些模型能够有效地表示复杂的、非线性的奖励结构。在这种情况下,深度架构是一种自然的选择。

3 双鉴别器生成对抗逆强化学习

逆强化学习的目的是利用专家样本来获取奖励函数,在此过程中涉及到复杂的迭代计算。但是在实际训练时,专家样本的缺乏以及生成样本利用率的低下会导致学习速率缓慢且学到的策略质量差。因此,本文针对以上问题提出一种基于HER策略的D2GA逆强化学习算法。

3.1 类专家样本生成

类专家样本的生成过程包含2个步骤:最优奖励函数与相应策略的更新、类专家轨迹的合成。

本文算法的具体框架如图1所示。首先,本文算法利用HER生成类专家样本,并将其作为逆强化学习的专家样本进行训练;其次,随机初始化策略π_0^{L}生成非专家样本,利用专家样本与非专家样本找出当前条件下的最优奖励函数,并将其与非专家样本一起用于策略更新;再次,用新策略π_{h+1}生成新的非专家样本与专家样本再次进行奖励函数的更新;最后,通过奖励函数更新与策略优化的交替,找到最优的奖励函数与相应策略。

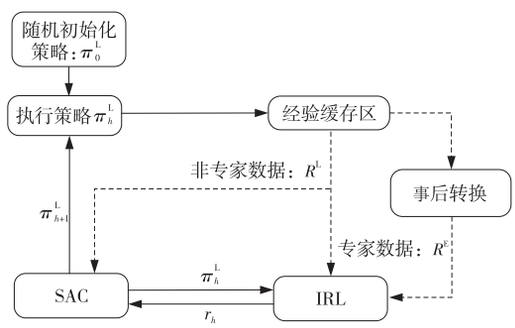


Figure 1 Overall structure of hindsight inverse reinforcement learning

图1 事后逆强化学习总体结构

使用具有随机权重的策略生成轨迹τ_0。根据3.4节算法1将τ_0通过HER转化为对应的类专家轨迹τ_{hero}。τ_0为负样本,τ_{hero}为正样本。在每轮最后都生成一段轨迹序列,如当第h轮结束时,生成轨迹序列$\tau^h\leftarrow\langle\boldsymbol{s}_0\|G,\boldsymbol{a}_0,\boldsymbol{s}_1\|G,\boldsymbol{a}_1,\cdots,\boldsymbol{s}_T\|G\rangle$,其中$T$是轨迹的长度,其中$G$表示当前智能体选择的目标位置,$\|$为维度联合符号。重复上述过程$N$轮,获得一系列轨迹$\tau\leftarrow\langle\boldsymbol{\tau}^0,\boldsymbol{\tau}^1,\cdots,\boldsymbol{\tau}^N\rangle$。

为了在没有专家演示的情况下训练鉴别器,本文算法利用事前变换技术(如算法1所示)将展开的轨迹τ转换为类专家轨迹τ_{her}。

更具体地说,从展开的轨迹τ中自合成类专家轨迹τ_{her}的详细步骤可以描述如下:首先,对于τ中的第h条轨迹$\boldsymbol{\tau}^h$,选择概率为p_{ht}的每个时间步t进行变换,其中$t\in[0,T]$。τ^h中用于事后变换的所有选择的时间步被添加到集合B_I。其次,对于B_I中的每个时间步j,随机设置状态$\boldsymbol{s}_j$的新目标,在状态$\boldsymbol{s}_l$处实现位置$\boldsymbol{p}_l$,其中l是从$\boldsymbol{\tau}^h$中的时间步$j\sim T$中随机选择的。换句话说,随机设置状态s_j的新目标,观察状态$\boldsymbol{s}_j$之后达到的位置,然后,成功地将轨迹$\boldsymbol{\tau}^h$转换为类专家轨迹$\boldsymbol{\tau}_{\mathrm{her}}^h$。重复上述过程,直到所有轨迹都已变换。

3.2 鉴别器模型构建

本文将专家策略表示为$\pi^{\mathrm{E}}(\boldsymbol{a}\mid\boldsymbol{s})$、非专家策略表示为$\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})$、由过渡元组构成的专家数据集表示为$R^{\mathrm{E}}=\{(\boldsymbol{s}_h,\boldsymbol{a}_h,\boldsymbol{s}'_h)\}_{h=1}^{N^{\mathrm{E}}}$。其中,$\boldsymbol{a}_h\sim\pi^{\mathrm{E}}(\cdot\mid\boldsymbol{s}_h)$,$N^{\mathrm{E}}$表示数据集中过渡元组的数量。

考虑 2 个联合密度函数 $\pi^{\mathrm{E}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 和 $\pi^{\mathrm{L}}(\boldsymbol{s},\boldsymbol{a}_h,\boldsymbol{s}')$,在马尔可夫假设下,$\pi^{\mathrm{E}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 被分解为 $\pi^{\mathrm{E}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')=p_{\mathrm{T}}(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})\pi^{\mathrm{E}}(\boldsymbol{a}\mid\boldsymbol{s})\pi^{\mathrm{E}}(\boldsymbol{s})$,$\pi^{\mathrm{L}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 也以同样的方式被分解。

逆强化学习的目标函数是最小化策略 $\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})$ 和 $\pi^{\mathrm{E}}(\boldsymbol{a}\mid\boldsymbol{s})$ 的反向 KL 散度,如式(13)所示:

$$V_{\mathrm{D2GA}}(\boldsymbol{\omega}_\pi)=KL(\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})\parallel\pi^{\mathrm{E}}(\boldsymbol{a}\mid\boldsymbol{s})) \tag{13}$$

其中,反向 KL 散度的表达式如式(14)所示:

$$KL(\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})\parallel\pi^{\mathrm{E}}(\boldsymbol{a}\mid\boldsymbol{s}))=E_{\boldsymbol{\pi}^{\mathrm{L}}}\left[\ln\frac{\pi^{\mathrm{L}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')}{\pi^{\mathrm{E}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')}\right] \tag{14}$$

由于 $\pi^{\mathrm{E}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 是未知的,所以对数比的求解很困难。解决该问题的基本思想是采用密度比技巧[23],这可以通过解决二进制分类任务来有效地实现。

由正则马尔可夫可知,正则化(最优)奖励函数的表达式如式(15)所示:

$$\tilde{r}^{*}_{\mathrm{D2GA}}(\boldsymbol{s},\boldsymbol{a})=r(\boldsymbol{s})+\frac{1}{k}H[\pi^{\mathrm{E}}(\boldsymbol{a}\mid\boldsymbol{s})]-\frac{1}{\eta}KL[\pi^{\mathrm{E}}(\boldsymbol{a}\mid\boldsymbol{s})\,||\,\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})] \tag{15}$$

其中,$r(\boldsymbol{s})$ 是由 $\boldsymbol{\omega}_{\mathrm{r}}$ 参数化的奖励函数。

状态-动作价值函数 $Q_h(\boldsymbol{s},\boldsymbol{a})$ 满足贝尔曼最优方程,如式(16)所示:

$$Q_h(\boldsymbol{s},\boldsymbol{a})=r_h(\boldsymbol{s},\boldsymbol{a})+\frac{1}{\eta}\ln\pi^{\mathrm{L}}_h(\boldsymbol{a}\mid\boldsymbol{s})+\gamma E_{\boldsymbol{s}'\sim p_{\mathrm{T}}(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})}[V^{*}_h(\boldsymbol{s}')] \tag{16}$$

其中,h 为迭代次数,$V^{*}_h(\cdot)$ 表示第 h 轮的最优状态价值函数。

根据式(6)和式(15)推导得出的逆强化学习的贝尔曼最优方程如式(17)所示:

$$\frac{1}{\beta}\ln\frac{\pi^{\mathrm{E}}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')}{\pi^{\mathrm{L}}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')}=r_h(\boldsymbol{s})-\frac{1}{k}\ln\pi^{\mathrm{L}}_h(\boldsymbol{a}\mid\boldsymbol{s})+\gamma E_{\boldsymbol{s}'\sim p_{\mathrm{T}}(\cdot\mid\boldsymbol{s},\boldsymbol{a})}[V^{*}_h(\boldsymbol{s}')]-V^{*}_h(\boldsymbol{s}) \tag{17}$$

根据密度比技巧,将式(16)重写为式(18)所示:

$$\frac{1}{\beta}\ln\frac{D^{(2)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')}{1-D^{(2)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')}=\frac{1}{\beta}\ln\frac{D^{(1)}_h(\boldsymbol{s})}{1-D^{(1)}_h(\boldsymbol{s})}-r_h(\boldsymbol{s})+\frac{1}{k}\ln\pi^{\mathrm{L}}_h(\boldsymbol{a}\mid\boldsymbol{s})-\gamma V^{*}_h(\boldsymbol{s}')+V^{*}_h(\boldsymbol{s}) \tag{18}$$

其中,$D^{(1)}_h(\boldsymbol{s})$ 和 $D^{(2)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 为鉴别器,它们可以将专家数据从生成器中分类出来。

添加一个参数为 $\boldsymbol{\omega}_{\mathrm{g}}$ 的神经网络 $g_h(\boldsymbol{s})$,令 $\ln[D^{(1)}_h(\boldsymbol{s})/(1-D^{(1)}_h(\boldsymbol{s}))]$ 近似为 $g_h(\boldsymbol{s})$,则第 1 个鉴别器如式(19)所示:

$$D^{(1)}_h(\boldsymbol{s})=\frac{1}{1+\exp(-g_h(\boldsymbol{s}))} \tag{19}$$

根据式(18)和式(19)推出第 2 个鉴别器如式(20)所示:

$$D^{(2)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')=\frac{\exp(\beta k^{-1}\ln\pi^{\mathrm{L}}_h(\boldsymbol{a}\mid\boldsymbol{s}))}{\exp(\beta f_h(\boldsymbol{s},\boldsymbol{s}'))+\exp(\beta k^{-1}\ln\pi^{\mathrm{L}}_h(\boldsymbol{a}\mid\boldsymbol{s}))} \tag{20}$$

其中,

$$f_h(\boldsymbol{s},\boldsymbol{s}')\triangleq r_h(\boldsymbol{s})-\beta^{-1}g_h(\boldsymbol{s})+\gamma V_h(\boldsymbol{s}')-V_h(\boldsymbol{s})$$

本文所提的双鉴别器生成对抗网络框架如图 2 所示,包含 2 个部分。

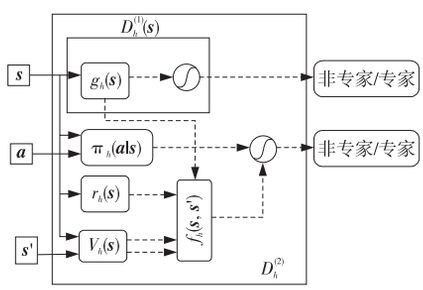


Figure 2 Structure of double discriminator

图 2 双鉴别器结构

首先,第 1 个鉴别器 $D^{(1)}_h(\boldsymbol{s})$ 通过求解式(21)的最大似然进行优化:

$$J^{(1)}_{\mathrm{D}}(\boldsymbol{\omega}_{\mathrm{g}})=E_{\boldsymbol{s}\sim R^{\mathrm{L}}_h}[\ln D^{(1)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')]+E_{\boldsymbol{s}\sim R^{\mathrm{E}}_h}[\ln(1-D^{(1)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}'))] \tag{21}$$

其中,$\boldsymbol{s}\sim R^{\mathrm{E}}_h$ 表示数据来自于 R,R^{L}_h 是由策略 $\pi^{\mathrm{L}}_h(\boldsymbol{a}\mid\boldsymbol{s})$ 在第 h 次迭代生成的过渡数据集。

第 2 个鉴别器 $D^{(2)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 以同样的方式进行训练,训练时 $\pi^{\mathrm{L}}_h(\boldsymbol{a}\mid\boldsymbol{s})$ 和 $g_h(\boldsymbol{s})$ 固定。第 2 个鉴别器通过最小化交叉熵进行优化,如式(22)所示:

$$J^{(2)}_{\mathrm{D}}(\boldsymbol{\omega}_{\mathrm{r}})=E_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')\sim R^{\mathrm{L}}_h}[\ln D^{(2)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')]+E_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')\sim R^{\mathrm{E}}_h}[\ln(1-D^{(2)}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}'))] \tag{22}$$

3.3 策略优化

策略(生成器)$\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})$ 采用策略梯度法进行优化,目标函数的表达式如式(23)所示:

$$V_{\mathrm{D2GA}}(\boldsymbol{\omega}_\pi)=E_{\pi^{\mathrm{L}}}[\ln\pi^{\mathrm{L}}_h(\boldsymbol{s}\mid\boldsymbol{a})-\beta(Q_h(\boldsymbol{s},\boldsymbol{a})-V_h(\boldsymbol{s}))+g_h(\boldsymbol{s})] \tag{23}$$

目标函数的梯度如式(24)所示:

$$\nabla L_{\mathrm{D2GA}}(\boldsymbol{\omega}_\pi)=E_{\pi^{\mathrm{L}}}[\nabla_{\boldsymbol{\omega}_\pi}\ln\pi^{\mathrm{L}}_h(\boldsymbol{s}\mid\boldsymbol{a})[\ln\pi^{\mathrm{L}}_h(\boldsymbol{s}\mid\boldsymbol{a})-\beta(Q_h(\boldsymbol{s},\boldsymbol{a})-V_h(\boldsymbol{s}))+g_h(\boldsymbol{s})]] \tag{24}$$

采用 SAC 算法更新状态价值函数和状态-动

作价值函数。状态价值函数的损失函数如式(25)所示：

$$L_{\mathrm{D2GA}}(\boldsymbol{\omega}_{\mathrm{V}})=\frac{1}{2}E_{\boldsymbol{s}\sim D}\left[\left(V(\boldsymbol{s})-E_{\boldsymbol{a}\sim\pi_h^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})}\left[Q_h(\boldsymbol{s},\boldsymbol{a})-\frac{1}{\beta}\ln\pi_h^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})\right]\right)^2\right] \tag{25}$$

状态-动作价值函数的损失函数如式(26)所示：

$$L_{\mathrm{D2GA}}(\boldsymbol{\omega}_{\mathrm{Q}})=\frac{1}{2}E_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')}\left[(Q(\boldsymbol{s},\boldsymbol{a})-\overline{Q}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')^2\right] \tag{26}$$

其中，

$$\overline{Q}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')=r(\boldsymbol{s})+\frac{1}{\eta}\ln\pi_h^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})+\gamma\overline{V}_h(\boldsymbol{s}') \tag{27}$$

其中，$\boldsymbol{\omega}_{\mathrm{Q}}$ 为状态-动作价值函数的网络参数；$\overline{V}_h(\boldsymbol{s}')$ 是由 $\overline{\boldsymbol{\omega}}_{\mathrm{V}}$ 参数化的目标状态价值函数，因为是无模型算法，所以无法计算式(4)，于是用 $\overline{Q}_h(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 近似 $Q(\boldsymbol{s},\boldsymbol{a})$。$\overline{\boldsymbol{\omega}}_{\mathrm{V}}$ 使用 Polyak 平均更新，如式(28)所示：

$$\overline{\boldsymbol{\omega}}_{\mathrm{V}}\leftarrow\tau\boldsymbol{\omega}_{\mathrm{V}}+(1-\varphi)\overline{\boldsymbol{\omega}}_{\mathrm{V}},\varphi\leqslant 1 \tag{28}$$

3.4 D2GA 算法流程

本文提出一种基于双鉴别器生成对抗网络的熵正则化逆强化学习算法。算法通过随机梯度方法对 IRL 目标进行更新，在每次迭代时对生成的专家样本数据回放缓冲区 R^{E} 和非专家样本数据回放缓冲区 R^{L} 进行部分采样(当批次中的样本数量很少时，需要将专家样本添加到非专家样本集中生成一个新的混合集)，通过双鉴别器网络和 SAC 重复利用经验池积累的样本来估计关于状态的策略梯度期望值，交替执行 IRL 奖励函数更新与策略优化过程。因此，D2GA 能够提高样本的利用效率，最终快速地找到最优奖励函数参数 $\boldsymbol{\omega}_{\mathrm{r}}$ 和相应策略 π。具体算法如算法 1 所示。

算法 1 基于双鉴别器生成对抗网络的熵正则化逆强化学习算法

输入：迭代次数 N，每个 Episode 的最大步数 T，熵正则化超参数 k 和 η。

输出：奖励函数网络参数 $\boldsymbol{\omega}_{\mathrm{r}}$、策略 $\boldsymbol{\omega}_{\pi}$。

```
1. 随机初始化 SAC,D(1),D(2)的各个网络参数,初始化 2 个空的经验回放缓冲区 R^L 和 R^E;
2. for episodes do
3.   for h = 0,1,2,… do
4.     以初始策略 π_h 和奖励函数 r_h 执行 SAC 算法,得到轨迹的集合 τ 和新策略 π_{h+1};
5.     利用式(23)更新 ω_π ;
6.     利用式(25)更新 ω_V ;
7.     利用式(26)更新 ω_Q ;
8.     将 τ 存储到 R^L 中;
9.     使用 HER 将 τ 合成类专家轨迹集合 τ_her ;
10.    将 τ_her 存储到 R^L 中;
11.    利用式(21)更新 ω_g ;
12.    利用式(22)更新 ω_r ;
13.  end for
14. end for
```

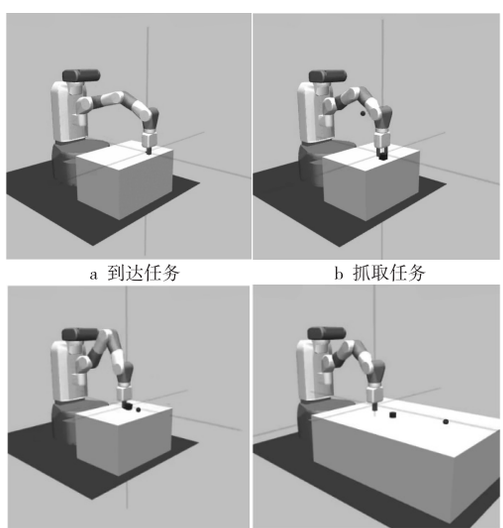
4 实验

本节的目标是测试通过本文算法学习到的策略能否在没有外部专家演示数据的情况下运行良好。此外，通过转换策略与转换概率对比实验，探讨不同的转换方式和事后转换率对策略学习的影响。

4.1 仿真实验

本节仿真实验所用的电脑显卡为NVIDIA® GeForce® RTX™ 2080Ti，系统为 Ubuntu 18.04，机器人模拟物理引擎采用 mujoco210，采用 OpenAI 发布的 Fetch[24]模拟机器人实验环境。

为了测试本文算法的可行性，在 Fetch 环境中对几种常见的机械臂任务：到达、推动、滑行和抓取进行了实验，如图 3 所示。本文将 D2GA 与 AIRL、GAIL 进行了比较。近似函数的网络架构均采用全连接网络，其中隐藏层有 3 层，每层 96 个节点，隐藏层中使用了整流线性单元 ReLU 激活函数。



a 到达任务 b 抓取任务 c 推动任务 d 滑动任务

Figure 3 Environment simulation of Fetch

图 3 Fetch 环境仿真

输出节点使用线性激活函数,$\mu(\boldsymbol{s})$ 和 $\boldsymbol{\sigma}(\boldsymbol{s})$ 除外。函数 $\mu(\boldsymbol{s})$ 和 $\boldsymbol{\sigma}(\boldsymbol{s})$ 通过高斯分布表示非专家策略,如式(29)所示:

$$\pi^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})=N(\boldsymbol{a}\mid\mu(\boldsymbol{s}),\boldsymbol{\sigma}(\boldsymbol{s})) \tag{29}$$

其中,$\mu(\boldsymbol{s})$ 和 $\boldsymbol{\sigma}(\boldsymbol{s})$ 分别表示均值和对角协方差矩阵。$\mu(\boldsymbol{s})$ 和 $\boldsymbol{\sigma}(\boldsymbol{s})$ 的输出节点使用 tanh 和 sigmoid 函数。

将 $\pi_h^{\mathrm{L}}(\boldsymbol{a}\mid\boldsymbol{s})$ 生成的轨迹数设置为 100。在所有的实验中,将正则化奖励的超参数统一设置为 $k=1$ 和 $\eta=10$,折扣因子 $\lambda=0.99$。本文使用 Adam 优化器和衰减学习率对所有网络进行训练。

将本文提出的 D2GA 算法与以下算法进行了比较:(1)GAIL[13] 与可用的演示,用 GAIL-demo 表示;(2)AIRL[12] 与可用的演示,用 AIRL-demo 表示。学习策略的性能由成功率指标进行衡量。

成功率定义为在允许误差 δ 内成功到达目标位置的次数与所消耗的所有时间步的比率,其表达式如式(30)所示:

$$s_{\mathrm{rate}}=\frac{\sum_{i}^{N}1(d_i\leqslant\delta)}{N} \tag{30}$$

其中,$1(\cdot)$ 是以 true 为输入,给出 1 为输出,以 false 为输入,输出 0 的指标函数,d_i 表示第 i 次迭代实验结果与终点的误差距离。

各种算法学习策略的性能学习曲线如图 4 所示,表 1 总结了最终学习策略的性能。

Table 1 Performance of policies trained with three different algorithms

表 1 3 种不同算法训练策略的性能表现

算法	到达任务成功率	抓取任务成功率
GAIL-demo	0.95±0.03	0.98±0.02
AIRL-demo	0.96±0.02	0.97±0.03
D2GA(Ours)	0.99±0.01	0.99±0.01
算法	**推动任务成功率**	**滑动任务成功率**
GAIL-demo	0.95±0.03	0.78±0.02
AIRL-demo	0.96±0.02	0.80±0.02
D2GA(Ours)	0.98±0.02	0.96±0.04

图 4 采用 5 种不同的随机种子对每种算法进行训练,折线表示成功率的中位数,阴影部分表示分位数,以此来显示算法在不同随机数种子下的有效性与稳定性。

观察图 4 可以发现,与已有专家演示数据的 GAIL、AIRL 相比,无专家演示数据的 D2GA 算法展现出了更快的收敛速度。这表明在没有演示数据的情况下,HER 是本文提出的 D2GA 算法取得效率优势的关键因素。从表 1 中可以看到,D2GA 算法在到达任务和抓取任务中学到的最终策略表现略微优于 GAIL 和 AIRL 的;而在较为复杂的推动和滑动任务中,D2GA 的最终策略表示则远远超过了 GAIL 与 AIRL 的。这是由于本文算法本质上被 HER 赋予了课程学习机制,在策略训练开始阶段,D2GA 算法相比 GAIL 算法的优化速度更

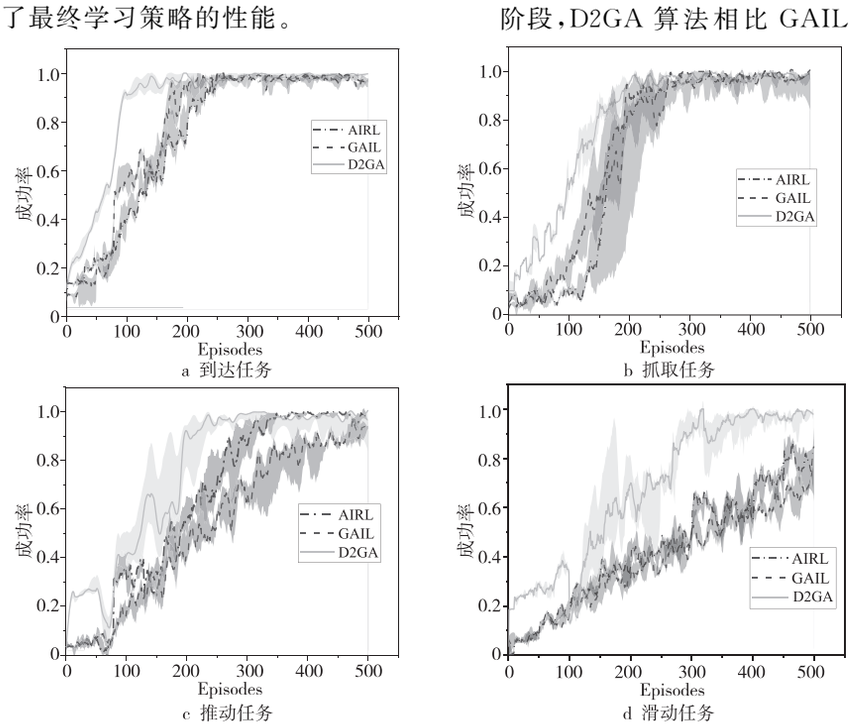


Figure 4 Comparison of the performance of three algorithms for training robotic arms on different tasks

图 4 3 种算法训练机械臂不同任务的性能对比

快。实验结果表明,在无专家演示数据的情况下,D2GA 可以成功地学习策略。

4.2 转换策略和转换概率对比实验

在实验中,对 4 种任务的消融实验得出了相同的结论。因此,为了使本文的内容更加简洁紧凑,本节中主要展示关于到达任务的实验结果。

4.2.1 转换策略对比实验

HER[11]提出了 2 种不同的事后转换策略,分别称为 Future 转换和 Final 转换。Future 转换将每个状态的目标替换为最终达到的状态在其自身情节中的位置。而 Final 转换是将每个状态的目标位置随机地改变为它之后观察到的状态位置。

2 种不同的事后转换的学习曲线如图 5 所示。从图 5 中可以看出,Final 后见之明转化效果不佳,在训练过程中所学到的策略相较 Future 转换逐渐变差。

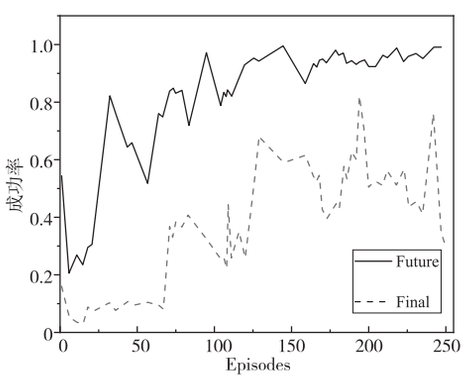


Figure 5 Learning curves of policy performance under different transition modes

图 5 不同转换方式下的策略性能学习曲线

4.2.2 转换概率对比实验

在 4.1 中,所有学习策略的性能都是用事后转换概率 $p_{ht}=1$ 进行训练的。本节对 p_{ht} 值对最终学习策略性能的影响进行了研究。实验设置 p_{ht} 值分别为 0.25,0.5,0.75 和 1,学习曲线如图 6 所示。

图 6 结果表明,将每个状态转换为概率为 1 的事后变换表现最好,这与 HER 不同。后见之明转换概率的值越大,最终学习的策略表现得越好。

4.3 实机部署验证

为了验证本文算法训练的策略在实际物理系统中部署的可行性和适应性,本节在真实的 Niryo-Ned 机械臂上进行策略部署。图 7 为对 3 个不同位置的立方体目标执行抓取与推动任务的实际效果图像。Ned 机械臂采用学习策略抓取目标物体和推动目标物体的帧分别如图 7 所示。

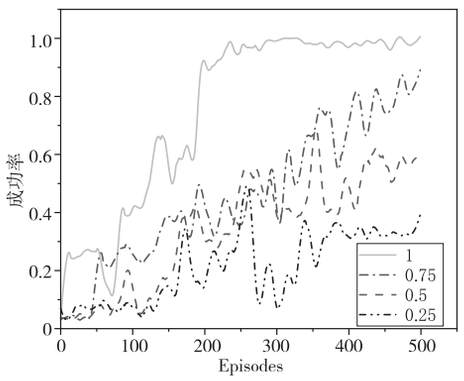
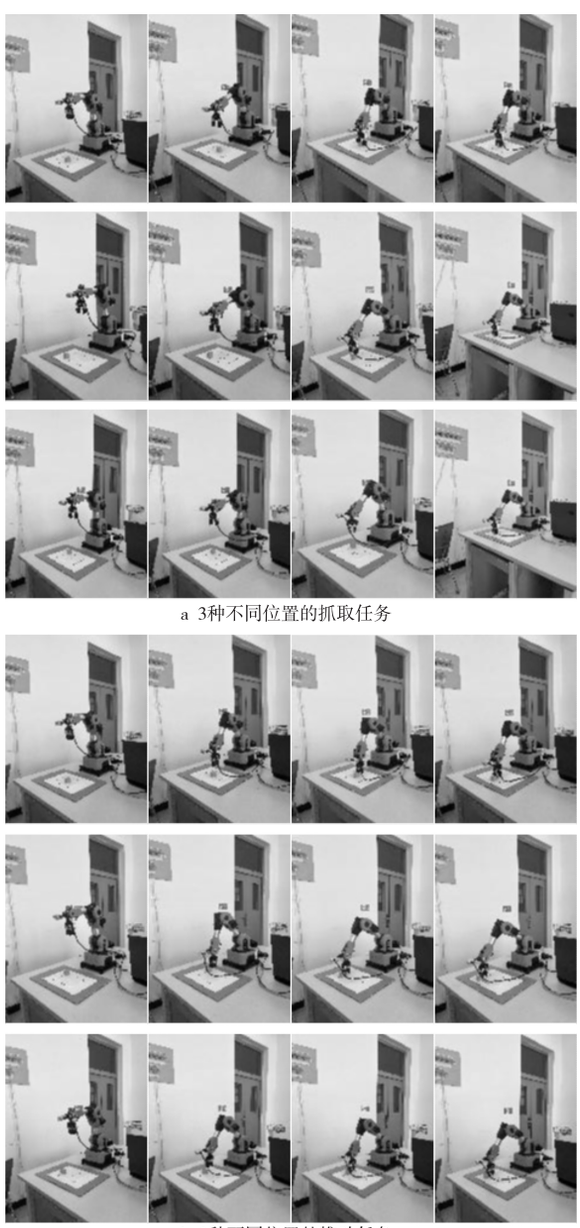


Figure 6 Learning curves of policy performance under different transition probabilities

图 6 不同转换概率下的策略性能学习曲线



a 3种不同位置的抓取任务

b 3种不同位置的推动任务

Figure 7 Deploy policies in realistic environments

图 7 真实场景策略部署

每种任务在不同位置各运行 500 次,抓取和推动任务的成功率分别为 0.99 和 0.98,接近仿真环境的理想性能。结果表明,使用 D2GA 学习的策略可以成功地从仿真环境迁移到现实场景,并且在不进行额外训练的情况下,可以保证真实场景中的性能与仿真环境一致,验证了本文算法的可行性和鲁棒性。

5 结束语

本文提出的 D2GA 算法解决了单鉴别器存在的样本获取困难以及生成样本利用率低的问题。该算法采用 HER 合成类似专家的演示数据,极大地压缩了因数据采集产生的人工成本与时间成本。除此之外,通过本文设计的双鉴别器网络模型,还可以充分利用生成样本的信息,加快学习速率。实验结果表明,本文算法在复杂信息维度空间环境中 4 种任务下的成功率均可以快速地达到理想性能。本文还通过真实场景中的策略布置实验,验证了该算法训练的可行性,得到了与仿真环境类似的优异性能表现。鉴于该算法训练策略的有效性和在复杂维度空间中的鲁棒性,该算法的提出对真实环境中的应用存在一定价值。

参考文献:

[1] Morales E F, Murrieta-Cid R, Becerra I, et al. A survey on deep learning and deep reinforcement learning in robotics with a tutorial on deep reinforcement learning[J]. Intelligent Service Robotics, 2021, 14: 773-805.

[2] 陈佳盼,郑敏华. 基于深度强化学习的机器人操作行为研究综述[J]. 机器人, 2022, 44(2): 236-256.
Chen Jia-pan, Zheng Min-hua. A survey of robot manipulation behavior research based on deep reinforcement learning[J]. Robot, 2022, 44(2): 236-256.

[3] Adam S, Cody T, Beling P A, et al. A survey of inverse reinforcement learning[J]. Artificial Intelligence Review, 2022, 55(6): 4307-4346.

[4] 陈希亮,曹雷,何明,等. 深度逆向强化学习研究综述[J]. 计算机工程与应用, 2018, 54(5): 24-35.
Chen Xi-liang, Cao Lei, He Ming, et al. Overview of deep inverse reinforcement learning[J]. Computer Engineering and Applications, 2018, 54(5): 24-35.

[5] Abbeel P, Ng A Y. Apprenticeship learning via inverse reinforcement learning[C]//Proc of the 21st International Conference on Machine Learning, 2004: 1-8.

[6] Wulfmeier M, Ondruska P, Posner I, et al. Maximum entropy deep inverse reinforcement learning[J]. arXiv: 1507.04888, 2015.

[7] Zhang T H, McCarthy Z, Jow O, et al. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation[C]//Proc of 2018 IEEE International Conference on Robotics and Automation, 2017: 5628-5635.

[8] Ajay A, Gupta A, Ghosh D, et al. Distributionally adaptive meta reinforcement learning[C]//Proc of the 36th International Conference on Neural Information Processing Systems, 2022: 25856-25869.

[9] Xu D F, Nair S, Zhu Y K, et al. Neural task programming: Learning to generalize across hierarchical tasks[C]//Proc of 2018 IEEE International Conference on Robotics and Automation, 2018: 3795-3802.

[10] Liu F C, Liu H, Grover A, et al. Masked autoencoding for scalable and generalizable decision making[C]//Proc of the 36th International Conference on Neural Information Processing Systems, 2022: 12608-12618.

[11] Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay[C]//Proc of the 31st International Conference on Neural Information Processing System, 2017: 5055-5065.

[12] Fu J, Luo K, Levine S. Learning robust rewards with adversarial inverse reinforcement learning[J]. arXiv: 1710.11248, 2017.

[13] Ho J, Ermon S. Generative adversarial imitation learning[C]//Proc of the 30th International Conference on Neural Information Processing Systems, 2016: 4572-4580.

[14] Jena R, Liu C L, Sycara K. Augmenting GAIL with BC for sample efficient imitation learning[C]//Proc of 2020 3rd Conference on Robot Learning, 2021: 80-90.

[15] Kinose A, Taniguchi T. Integration of imitation learning using GAIL and reinforcement learning using task-achievement rewards via probabilistic graphical model[J]. Advanced Robotics, 2020, 34(16): 1055-1067.

[16] 蔡钺,游进国,丁家满. 基于近端策略优化与对抗学习的对话生成[J]. 计算机工程与科学, 2020, 42(9): 1680-1689.
Cai Yue, You Jin-guo, Ding Jia-man. Proximal policy optimization and adversarial learning based dialog generation[J]. Computer Engineering & Science, 2020, 42(9): 1680-1689.

[17] Zhang Z Y, Li M Y, Xie H N, et al. TWGAN: Twin discriminator generative adversarial networks[J]. IEEE Transactions on Multimedia, 2022, 24: 677-688.

[18] 陆彦辉,柳寒,李航,等. 基于多鉴别器生成对抗网络的时间序列生成模型[J]. 通信学报, 2022, 43(10): 167-176.
Lu Yan-hui, Liu Han, Li Hang, et al. Time series generation model based on multi-discriminator generative adversarial network[J]. Journal on Communications, 2022, 43(10): 167-176.

[19] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT Press, 1998.

[20] Ziebart B D, Maas A, Bagnell J A, et al. Maximum entropy inverse reinforcement learning[C]//Proc of the 23rd National Conference on Artificial Intelligence, 2008: 1433-1438.

[21] Kozuno T, Uchibe E, Doya K. Theoretical analysis of efficiency and robustness of softmax and gap-increasing operators in reinforcement learning[C]//Proc of the 22nd Inter-

national Conference on Artificial Intelligence and Statistics, 2019:2995-3003.

[22] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]// Proc of the 27th International Conference on Neural Information Processing Systems, 2014:2672-2680.

[23] Sugiyama M, Suzuki T, Kanamori T. Density ratio estimation in machine learning[M]. Cambridge: Cambridge University Press, 2012.

[24] Plappert M, Andrychowicz M, Ray A, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research[J]. arXiv:1802.09464, 2018.

作者简介:

段成龙(1996-),男,山西晋城人,硕士生,研究方向为深度强化学习和逆强化学习。**E-mail**:1921440194@qq.com

DUAN Cheng-long, born in 1996, MS candidate, his research interests include deep reinforcement learning and inverse reinforcement learning.

袁杰(1975-),男,重庆人,博士,教授,研究方向为计算机应用。**E-mail**:yuanjie222@126.com

YUAN Jie, born in 1975, PhD, professor, his research interest includes computer application.

常乾坤(1996-),男,新疆库车人,硕士生,研究方向为视觉机械臂抓取。**E-mail**:1069855042@qq.com

CHANG Qian-kun, born in 1996, MS candidate, his research interest includes visual robotic arm grasping.

张宁宁(1982-),女,山东威海人,博士生,研究方向为机器人和智能控制。**E-mail**:157267828@qq.com

ZHANG Ning-ning, born in 1982, PhD candidate, her research interests include robot and intelligent control.