计算机科学
COMPUTER SCIENCE

深度强化学习中的知识迁移方法研究综述

张启阳, 陈希亮, 曹雷, 赖俊, 盛蕾

引用本文

张启阳, 陈希亮, 曹雷, 赖俊, 盛蕾.深度强化学习中的知识迁移方法研究综述[J]. 计算机科学, 2023, 50(5): 201-216.

ZHANG Qiyang, CHEN Xiliang, CAO Lei, LAI Jun, SHENG Lei. Survey on Knowledge Transfer Method in Deep Reinforcement Learning [J]. Computer Science, 2023, 50(5): 201-216.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220400235

深度强化学习中的知识迁移方法研究综述

张启阳　陈希亮　曹　雷　赖　俊　盛　蕾
陆军工程大学指挥控制工程学院　南京 210007
(qiyangz@foxmail.com)

摘　要　深度强化学习是人工智能研究中的热点问题,随着研究的深入,其中的短板也逐渐暴露出来,如数据利用率低、泛化能力弱、探索困难、缺乏推理和表征能力等,这些问题极大地制约着深度强化学习方法在现实问题中的应用。知识迁移是解决此问题的非常有效的方法,文中从深度强化学习的视角探讨了如何使用知识迁移加速智能体训练和跨领域迁移过程,对深度强化学习中知识的存在形式及作用方式进行了分析,并按照强化学习的基本构成要素对深度强化学习中的知识迁移方法进行了分类总结,最后总结了目前深度强化学习中的知识迁移在算法、理论和应用方面存在的问题和发展方向。

关键词:人工智能;知识迁移;强化学习;深度强化学习;迁移学习
中图法分类号　TP181

Survey on Knowledge Transfer Method in Deep Reinforcement Learning

ZHANG Qiyang,CHEN Xiliang,CAO Lei,LAI Jun and SHENG Lei
College of Command and Control Engineering,Army Engineering University of PLA,Nanjing 210007,China

Abstract　Deep reinforcement learning is a hot issue in artificial intelligence research. With the deepening of research,some shortcomings are gradually exposed,such as low data utilization,weak generalization ability,difficult exploration,lack of reasoning and representation ability,etc. These problems greatly restrict the application of deep reinforcement learning method in practical problems. Knowledge transfer is a very effective method to solve this problem. This study discusses how to use knowledge transfer to accelerate the process of agent training and cross domain transfer from the perspective of deep reinforcement learning,analyzes the existing forms and action modes of knowledge in deep reinforcement learning,and classifies and summarizes the knowledge transfer methods in deep reinforcement learning according to the basic elements of reinforcement learning. Finally,the existing problems and cutting-edge development direction of knowledge transfer in deep reinforcement learning in algorithm,theory and application are reported.

Keywords　Artificial intelligence,Knowledge transfer,Reinforcement learning,Deep reinforcement learning,Transfer learning

1　引言

传统强化学习方法通过构建一个交互式的学习框架,为求解序贯决策问题提供了解决方案。然而,受限于表格值、组合人工特征表征价值函数和策略函数等方法对环境的表征能力,强化学习方法难以向较为复杂的问题扩展。深度学习能够从高维数据中进行自动特征提取,具有较强的感知能力。因此,自 Deepmind 在以雅达利[1]为代表的视频游戏中利用深度神经网络进行图像特征提取与强化学习算法结合取得巨大成功以来,深度强化学习在视频游戏、棋类博弈、机器人控制、自然语言处理、即时战略游戏、医学决策支持等众多领域中不断攻城掠地,取得了诸多突破性进展[2-7]。深度强化学习在不断深入应用的过程中也面临诸多挑战,当前以无模型深度强化学习为代表的算法通常需要与环境的巨量交互,以改善模型的性能。在雅达利游戏中,智能体需要通过 1800 万帧的数据交互才能训练出一个可用的模型[8]。其原因在于交互环境的部分可观察性、反馈的稀疏性、延迟性和欺骗性等特点,导致深度强化学习算法在训练过程中存在数据利用率低、泛化能力弱、探索困难、缺乏推理和表征能力不足等问题,这些问题极大地制约着深度强化学习方法在实际复杂问题中的应用。

强化学习的模型是受到人类学习过程的启发而提出的,但事实上,人类却不需要如此规模庞大的数据交互来学习新任务。在雅达利游戏中,人类玩家只需要数次交互就能获取任务经验,在任务中表现良好,这主要得益于人类可以重用之前学习到的知识,将学习过的知识迁移到新的学习任务中,可以极大地提升学习效率。

事实上,智能体之所以难以在巨量策略空间中收敛,是

到稿日期:2022-04-24　返修日期:2022-07-12
基金项目:国家自然科学基金(61806221)
This work was supported by the National Natural Science Foundation of China(61806221).
通信作者:陈希亮(383618393@qq.com)

因为其没有利用任何先验知识。知识迁移最早是用来研究人类心理活动的一种方法。心理学家认为,人类学习认知事物并将其举一反三的过程就是一种知识迁移行为。受到人类学习模式的启发,知识迁移被引入机器学习领域中,引起了学术界的广泛讨论。结合知识迁移的方法进行深度强化学习,可以使智能体的学习过程更加靠近人类思维,利用已掌握的知识帮助新任务的学习,提升智能体的学习效率。因此,作为一种利用领域知识提升深度强化学习算法效率的技术,迁移学习方法成为了深度强化学习走向实际应用的一个非常重要的研究方向。

本文第 2 节中介绍了强化学习和知识迁移的相关基础概念。第 3 节阐述了强化学习中知识迁移的基本模式。与监督学习相比,深度强化学习在训练过程中涉及的要素更多,知识迁移所涉及的因素与监督学习不同。此外,知识的存在形式、迁移方式都与监督学习中的知识迁移有较大差异。因此,本文在阐述强化学习及知识迁移的相关基础概念的基础上,在第 4 节中从深度强化学习的视角提出了一些关于强化学习中知识的存在形式和进行知识迁移的一些想法,对深度强化学习中的知识迁移最前沿研究进展进行了综述,并对不同方法的特点、应用场景等进行了分析和总结。第 5 节介绍了目前该领域的研究进展及实际应用。最后总结全文并对未来的研究方向进行了展望。

2 相关概念

2.1 强化学习

强化学习(Reinforcement Learning,RL)是机器学习中最活跃的领域之一,它是一种试错性质的学习,智能体需要在和环境的不断交互中找到一个最优的策略,同时在智能体试错的过程中要尽可能地利用已知的回报。因此智能体在决策时需要平衡探索和利用两个方面。探索指智能体在某个状态下去尝试选择一个新的动作,发掘环境中的更多信息;而利用是智能体根据已知知识选取最优行为来获取最大回报。

强化学习的基本设定如图 1 所示。智能体根据外界环境来获取信息,环境给智能体提供初始状态,智能体决策出一个动作投入到环境中,环境给智能体反馈该动作的奖励和下一个状态,如此循环往复,智能体参考不同状态下采取不同动作所获得的奖励值,就能和人类一样学习出一个最优策略,来保证自身的奖励最大化[9]。

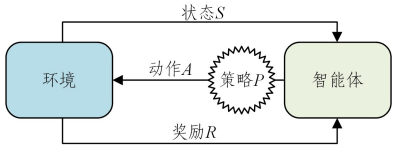


图 1 强化学习基本设定

Fig. 1 Basic setting of reinforcement learning

2.2 深度强化学习

在强化学习中的智能体与环境交互的过程中,有时需要从图像或文本中获取环境信息,环境中的视觉语音等高维输入对于强化学习中的智能体来说是一个巨大的挑战。另外,在智能体学习的过程中会处理大量的数据。例如在 Q 学习中,智能体需要储存大量的“状态-动作”和对应的 Q 值,在查找过程中需要耗费的大量的时间和空间。深度学习使用神经网络来反映输入值和输出值之间的映射关系,是一种通用的函数逼近方法。深度强化学习结合了深度学习中感知能力的特点和强化学习中决策能力的优势,可以有效地解决高维输入的决策问题。在深度强化学习中,使用函数逼近器来近似拟合表示函数值,替代了强化学习中的线性函数、决策树、瓦片编码(Tile Coding)等传统表示方法,极大地提高了学习效率。近年来,研究者们提出了很多深度强化学习算法,其中 DQN,DDPG,A3C,TRPO,PPO 是最具代表性的几种单智能体深度强化学习算法,Qmix 和 MADDPG 是效果较好、使用较广泛的多智能体算法。

深度 Q 网络(Deep Q Network,DQN)是首个将深度学习与强化学习相结合的成熟算法,其利用了深度卷积神经网络逼近值函数,还在学习过程中加入了经验回放以保证训练的收敛和稳定,在其 2015 年的改进版本中又独立地设置了目标网络来单独处理时间差分算法中的 TD 偏差[10]。DQN 中采用 Q 网络来近似表示值函数解决了连续的状态空间问题,但却无法解决连续的动作空间问题。利用 DQN 扩展 Q 学习的思路,Lillicrap 等对确定性策略梯度(Deterministic Policy Gradient,DPG)算法[11]进行扩展,提出了深度确定性策略梯度(Deep Deterministic Policy Gradient,DDPG)的算法[12]。DDPG 的学习过程如图 2 所示,这是一种基于演员-评论家(Actor-Critic,AC)框架的算法,可以用于解决连续动作空间的深度强化学习问题。

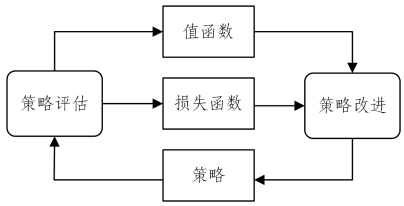


图 2 DDPG 方法的学习过程

Fig. 2 Learning process of DDPG

很多策略梯度方法都采用了经验回放机制来消除训练数据之间的相关性,经验回放机制需要智能体与环境实时交互,对计算能力要求较高。异步优势演员-评论家(Asynchronous Advantage Actor-Critic,A3C)是一种基于异步思想的轻量级 AC 框架[13],其在学习过程中使用 n 步回报来同时更新策略函数和价值函数,在效果、时间和资源消耗上都优于传统方法。

普通的策略梯度方法存在两个问题,一是采样效率较低,与环境交互的次数过多,浪费资源;二是步长很难设置适当,步长过小会导致更新速度慢,效率低下;步长过大会导致算法不稳定,需要大量的时间进行参数调整。信任域策略优化(Trust Region Policy Optimization,TRPO)通过设置置信域来解决步长设计的问题[14],使得每步都能够找到合适的步长,在策略更新后保证回报函数单调递增。近端策略优化(Proximal Policy Optimization,PPO)算法也是一种解决该问题的策略梯度算法,其将 KL 散度作为惩罚项[15],比 TRPO 更容易求解,是目前非常流行的单智能体强化学习算法。

Qmix 是多智能体强化学习中效果较好的算法[16],其在

值分解网络（Value Decomposition Network，VDN）的基础上做了一些改进。VDN 对每个智能体的值函数进行线性相加得到联合动作值函数[17]，Qmix 在 VDN 的基础上增加了全局的状态信息，在多个智能体之间的差异较大的环境中效果更好。

Lowe 等将 DDPG 扩展到多智能体环境，提出了多智能体深度确定性策略梯度（Multi-agent Deep Deterministic Policy Gradient，MADDPG）算法[18]。其最核心的部分是每个智能体的 Critic 部分都能够获取其余所有智能体的动作信息，进行集中式的训练和分布式的执行。MADDPG 还调整了经验回放缓存的数据结构，使其能够用于动态环境。在智能体的训练过程中，其对每个智能体学习多个策略，利用所有策略的整体效果进行优化，以提高算法的稳定性和鲁棒性。

2.3 深度强化学习中的知识

在深度强化学习中，基于值函数的方法利用神经网络拟合状态价值函数，基于策略的方法利用神经网络拟合策略函数，这些方法的本质都是存储通过与环境交互获取的知识来进行决策。在学习知识的过程中，智能体通过和环境的交互，经过“行动-评价”的模式得到奖励回报，这个回报一般和该动作是否有助于智能体得到最优策略正相关。智能体通过这种不断试错的方式来学习，在每个“行动-评价”周期中不断获取知识，改进行动方案以适应环境。目前，强化学习的发展方向之一是通用人工智能，即通过端到端强化学习，获取解决问题的策略[19]；另一个方向是把强化学习作为一种解决问题的框架，通过领域历史数据和经验知识的利用，在尽可能不约束强化学习算法能力上限的情况下提升强化学习的效率。因此，如何选择好知识介入强化学习的环节是第二个方向重点研究的内容。

深度强化学习中的知识即为智能体的内部信息，知识可以是状态、策略、奖励值，还可以包含于用于选择动作、预测累积奖励或预测未来观测特征的函数参数中。在这些知识中，有些是先验知识，例如对已有专家知识的总结利用和人工对智能体的设计；有些知识是通过学习获得的，需要在智能体和环境的不断交互中获得。随着环境的不断丰富，对于知识需求的平衡将越来越倾斜于学习知识。在强化学习过程中知识的作用体现在多个方面，它既可以直接用于策略模型的生成，也可以在强化学习智能体的进化过程中辅助智能体进行学习。

深度强化学习正处在蓬勃发展的阶段，每年都有更为先进的算法被提出并应用于更广泛的领域。深度强化学习的出现使得强化学习的技术从理论真正走向实用，真正解决现实环境中的复杂问题。

3 深度强化学习中的知识迁移模式

深度强化学习中，将源任务中学习到的各类知识应用于新的学习任务或学习阶段中，可以促进新的学习任务更快地学习或提升学习效果，这一过程被称为知识迁移。在大量的实验中发现，大多强化学习任务都是相对独立的，在一个任务中训练出的模型和经验等知识在新的任务中无法被较好地利用，造成大量训练资源的浪费，每次都需从零开始学习。在相同环境的不同任务中，或者同一个学习任务的不同阶段都可以利用知识迁移的思想进行学习，对已有知识最大化利用可以提高强化学习的效率。如 2.3 节所述，深度强化学习中的知识迁移的实现过程涉及价值网络、策略网络、初始状态、探索与利用、经验缓存回放、回报函数等多个方面。需要注意的是，在实际运用中，这些知识迁移模式往往不是独立存在的而是相互结合以达成知识迁移的目的，本节将结合强化学习中知识存在的方式，对强化学习中的知识利用的基本模式进行分析。

3.1 神经网络初始化

强化学习算法可以分为基于价值（Value-based）和基于策略（Policy-based）两类。在强化学习的经典算法 Q 学习（Q-learning）中[20]，通过储存一张 Q 表来记录每个“状态-动作”值与未来预估奖励的映射关系，但是 Q 表只能解决状态空间和动作空间较少的问题。受限于表格值方式的表征能力，强化学习在解决规模较大或复杂实际问题中进展缓慢。随着深度学习领域的发展，直接从原始感知数据中自适应地提取高层特征成为可能，出现了将二者结合的技术，即深度强化学习。深度强化学习的方法包括：使用深度神经网络拟合强化学习中最优动作估值函数[10]；使用限制玻尔兹曼机（Restricted Boltzmann Machine，RBM）模型[21]估计价值函数或策略函数等；使用自编码机模型来提取特征，然后使用这些特征在神经拟合 Q 值迭代（Neural Fitted Q Iteration，NFQ）[22]算法中学习非线性策略，并在多个基准数据集中获得比传统算法更好的效果。

深度强化学习的突破性进展体现在 Google Deepmind 的 Mnih 等提出的神经网络和 Q-learning 相结合的 DQN 算法[1]，其中神经网络用于代替 Q 表，以获得状态和动作对应的 Q 值。DQN 在训练时采用了 Q-learning 的思路，用神经网络拟合 Q-learning 中的估值误差项。

$$L(\theta)=E((R+\gamma\max_{a}(s,a,\theta)-Q(s,a,\theta))^2)$$

通过梯度下降使 $L(\theta)$ 最小化，其中 θ 为神经网络的参数。除此之外，在深度强化学习中类似的神经网络还被用于策略参数化（策略网络）、设计奖励函数等。事实上，深度神经网络和表格值的作用相同，都是拟合状态-动作之间的映射关系。因此，具备表征能力的模型，如深度森林[23]、线性模型等都可以作为表征模型使用。

在训练这些以神经网络为代表的表征模型的过程中，往往要耗费大量的时间成本来使其收敛。然而，当任务稍有变化时，原始训练的神经网络需要从头开始训练，将不再适用于新任务。基于知识迁移的想法，出现了一个很自然的问题，深度强化学习中的神经网络是否可以利用另一个训练好的相似神经网络的参数进行初始化？Yosinski 等[24]以 2012 年 ImageNet 大赛的冠军 Alexnet 模型[25]为实验对象进行了一系列关于神经网络各个层级上的可迁移性研究，发现模型的前三层基本都是一般特征，比较适合进行迁移；在神经网络的迁移过程中进行微调（Fine-tune），效果会得到明显提升，甚至超过原网络，微调操作还可以消除迁移过程中数据的差异性，迁移神经网络的效果优于随机初始化网络权重。

在现有的深度强化学习研究中，网络初始化大都采用随机初始化网络权重的方式，其原因主要是深度强化学习算法

的泛化性能很差，通常针对一个任务训练好一个模型，这个模型只对当前任务有效，不适用于其他任务。当任务之间有较强的相关性或网络结构比较适合进行迁移时，可以使用训练好的神经网络参数初始化另一个新任务的模式来实现知识迁移。

3.2 状态初始化

深度强化学习中学习效果不理想通常是由探索困难造成的，其原因主要在于环境中给出的奖励很稀疏，智能体往往在连续做了若干动作之后也得不到奖励回报，一个正向的奖励需要智能体做出一连串的特定动作才能获得[26]。例如在雅达利中的经典游戏蒙特祖玛的复仇中，游戏中的人物需要完美避开骷髅、螃蟹、深坑等静态的阻碍，还要穿过电墙、蛇等动态的障碍物才有机会拿到一枚金币，而要进入新的空间还需要获取到足够的钥匙。这就造成智能体在学习过程中稍有不慎就无法获取得奖励，大多数的学习过程都发生在游戏的起始阶段，而缺少对远端环境的学习。另一个重要原因是，在一些任务中，奖励具有误导性，比如在雅达利游戏集中的另一款叫做陷阱的游戏中，人物初始有 2000 点金币，碰到锯齿等障碍物或者掉落深坑时会损失几十到几百不等的金币，但是跨过若干障碍后钱袋可以立即获得 2000 金币。智能体在学习过程中遇到大的正奖励之前会遇到一些较小的负奖励，这会使得智能体停止探索，而错过了后面更大的奖励[27]。

为了在这种稀疏奖励的环境下更加有效地探索，研究者们通常使用内在动机(Intrinsic Motivation，IM)[28]的方法，即智能体在尽量获取更大的环境奖励的基础上，也使用内部奖励尽可能激励智能体去探索没有遇到过的状态空间。但这类方法也存在一些问题，即内在奖励虽然鼓励智能体去探索更大的未知状态空间，但是当前状态到已探索过的状态空间边界之间隔着很多状态，IM 方法并不能激励智能体越过这些内在奖励很小的状态走到边界上进行探索。此外，噪音和人工加入的随机扰动也限制了智能体完美地复刻先前的轨迹直接走到探索边界，通常会发生路径偏移。这种情况下让智能体自己到达探索的边界状态是很困难的，我们不妨考虑在训练过程中直接进行状态初始化，即将智能体探索到较好的状态轨迹时进行存档，再利用存档的边界状态将环境初始化，让智能体能够进一步探索到更大的状态空间，这种迁移模式可以有效地帮助智能体扩展探索边界。

3.3 探索与利用

在深度强化学习训练过程中，智能体的核心工作之一就是平衡探索与利用之间的关系，这也是强化学习领域的一个重点难题。对于智能体来说，其面临着一个艰难的抉择，即到底是利用已经训练好的模型选择动作，还是探索未知空间，以增大获取更大回报的可能性。

在平衡开发与探索二者之间，ε-greedy 是随机探索的一种常用策略[29]，它以 ε 的概率在所有可能的动作中随机选择，$1-\varepsilon$ 的概率按照策略 π 选择价值最高的动作，则选择动作的概率为：

$$P(a|s)=\begin{cases}\dfrac{1}{k}*\varepsilon+(1-\varepsilon), & a=\underset{a\in A}{\operatorname{argmax}}\,Q(s,a)\\ \dfrac{1}{k}*\varepsilon, & a\neq\underset{a\in A}{\operatorname{argmax}}\,Q(s,a)\end{cases}$$

其中，k 为所有可能的动作数。ε-greedy 相较于改进前的方法，在增加随机探索概率的同时又可以保证智能体大体上还是在最优策略上前进。除了 ε-greedy 以外，这种加入随机因素的思想还被用在玻尔兹曼探索策略[30]和高斯探索策略中。这些策略的探索方法都是在原始策略上加上一个随机无向的噪声，虽然所有的动作都因此有了被选择的概率，但是这种选择太过随机，即便有达到全局最优的可能，也需要大量的探索来实现，数据利用效率极低。

置信区间上界(Upper Confidence Bound，UCB)算法则弥补了这一不足。UCB 是一种对不确定性大的动作进行试探性探索的方法，UCB 值主要包括两项：$Q_t(a)+U_t(a)$。其中 $U_t(a)$ 代表探索，是对动作 a 不确定性的一种度量值；$Q_t(a)$ 代表利用，表示当前的动作奖励分布，也就是 Q 函数。UCB 的目标是最大化动作的置信度，也就是置信区间，表示为：

$$a_t=\arg\max_{a\in A}(Q_t(a)+U_t(a))$$

其中，$U_t(a)=c\sqrt{\ln N_t/N(a)}$，$N(a)$ 表示动作 a 被选择的次数，$\ln N_t$ 表示选择动作总次数 N_t 的对数，c 为权值。UCB 虽然效果比 ε-greedy 略好，但也存在收敛速度慢、依赖静态分布的问题，且对复杂动作的问题解决效果较为有限。文献[31]提出了一种通用规则表示方式，尝试将定性规则知识引入到强化学习中，通过云推理模型对定性规则进行表示，将其作为探索策略引导智能体的动作选择，以减少智能体在状态-动作空间探索的盲目性。Uber 公司提出了 Go-Explore 的方法[32]用于改善探索与利用的平衡问题，其采用状态初始化的方法将以往状态存档并重新加载在环境中开始探索；此外，其精巧地设计了一个“存档库”和加载机制来平衡探索与利用的抉择问题，该方法可以极大地提高探索的效率。但它是基于一个可重置的确定性环境而提出的，对于不可重置状态的不确定环境，可以考虑通过奖励塑形的方法构造内在奖励来实现平衡探索与利用的效果。

3.4 经验缓存回放

受到生物学的启发，深度强化学习中学习过程会产生大量的交互数据，这些数据只学习一次就会被丢弃，造成了大量的数据浪费。如果我们能像大脑中的海马体一样对这些知识加以利用，可能会有意想不到的效果。在强化学习中，我们把这些交互数据称为经验，在训练时把经验存入经验缓存池，在需要使用这些经验并进行学习时再进行回放，这个过程被称作经验缓存回放。在 Deepmind 于 2015 年提出的 NIPS DQN[1]中，首次将经验回放技术与深度强化学习相结合，使深度强化学习的研究取得了突破性的进展。

基于常规经验回放的 DQN 在抽取经验池中的以往经验样本时，采用的是随机抽取方法，忽略了经验样本的重要程度。在人类的学习过程中，大脑会区分更为重要的经验和次重要的经验并进行针对性的学习，2016 年 Schaul 等提出了一项改进，即在抽取经验池中的过往经验样本时采取按优先级抽取的方法，也被称为优先经验回放(Prioritized Experience Replay，PER)[33]。PER 以 49 种游戏中 41 场胜利的成绩超过了常规经验回放的 DQN 算法，达到了 SOTA 效果。这项改进使用时间差分偏差(TD-error)方法来权衡经验的重要

程度，TD-error 方法计算真实奖励 $R_{t+1}+\gamma V(S_{t+1})$ 和预估奖励 $V(S_t)$ 的差值 $\delta_t=R_{t+1}+\gamma V(S_{t+1})-V(S_t)$，$\delta_t$ 越大，优先级越高，在学习时采样的权重越大。TD-error 容易受到噪声的影响，使预估奖励产生偏差，这样就可能存在需要学习的经验片段却产生较小 TD-error 的假象，优先级很小，甚至不再进行学习。为了使具有较低 TD-error 的经验片段也有抽取的可能，可以使用两种方式进行改进，一是使用比例优先级的方法加入一个较小的 ε，使得当偏差 $|\delta_i|$ 为 0 时也能有机会被回放到，通过 $p_i=|\delta_i|+\varepsilon$ 计算优先级；二是采用基于等级的优先级方法，将排序时的序列的等级记作 $rank(i)$，通过 $p_i=1/rank(i)$ 计算优先级。为了弥补优先回放机制引入的偏置(Bias)，采用重要性采样的方法减少偏置，重要性采样权重为 $w_i=(1/N\cdot P(i))^{\beta}$，可以使用此权重乘以 δ_i，然后加入到网络更新中确保更新的无偏性，使结果收敛更稳定。

在自然语言处理领域，Narasimhan 等[34]将 DQN 与长短期记忆人工神经网络[35]相结合来解决文本游戏。该模型将上下文作为状态集合，单词作为动作进行强化学习。基于优先经验回放的原理，该方法创建了经验缓存区，并采用了优先采样的方法，根据经验能获得的正奖励大小和对最优 Q 值的学习速度，设置两个经验池用于储存较高优先级和较低优先级的样本，从高的经验池中采样比例为 ρ，从另一个较低的经验池中的采样比例则为 $1-\rho$。

Hou 等[36]将强化学习中基于优先级的经验回放机制(Prioritized Experience Replay，PER)从离散控制领域拓展到了连续控制领域，并指出该做法可以显著减少网络的训练时间，提高训练过程的稳定性及模型的鲁棒性。在传统的连续控制领域，让智能体直接通过像素级的视觉输入来完成复杂的动作是一项十分困难的任务[37]。DDPG[12]在很多连续控制任务中都取得了很好的效果，它使用一个经验回放池(Replay Buffer)来消除输入经验间存在的强相关性。这里的经验指一个四元组 (s_t,a_t,r_t,s_{t+1})，DDPG 使用目标网络法来稳定训练过程。作为 DDPG 算法的一个基本组成部分，经验回放极大地影响了网络的训练速度和最终效果。因此该方法的思路是用优先级经验回放替代 DDPG 中原有的传统经验回放，并使用重要性采样权重来修正偏差[38]，该方法在倒立摆任务等连续控制领域取得了良好的效果。

为了使深度强化学习的网络训练更有效率，在 PER 的基础上，Horgan 等[39]提出了一种分布式获取回放缓存数据并进行优先经验回放的方法，将其称为 Ape-X。其核心思想是设计多个 Actor，其中每个 Actor 采用不同 ε 的随机探索策略，在运行时，多个 Actor 每隔一段时间同时与环境交互，从 Critic 网络中获取经验并存入经验池中，然后按照优先级经验回放的 AC 算法进行学习。图 3 给出了 Ape-X 的架构。简单来说就是：有多个 Actor，每个 Actor 都有自己的环境实例，它们各自生成经验，并将其添加到共享的经验回放内存中，然后计算数据的初始优先级。单个学习者从这个经验回放内存中取样，更新记忆中的网络和经验的优先级。Actor 的网络会根据学习者提供的最新网络参数进行定期更新。

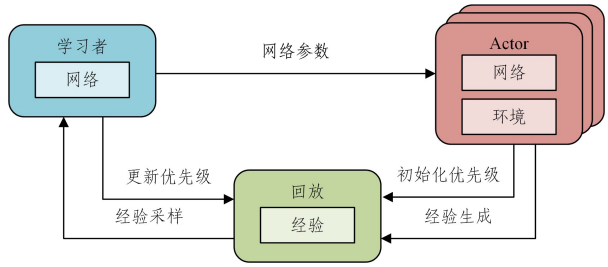


图 3 Ape-X 结构示意图

Fig. 3 Diagram of Ape-X structure

Fedus 等[40]基于使用经验回放的 DQN 算法讨论了经验缓存回放的基本原理，定义了经验回放的 3 个特征：回放容量(Replay Capacity)、经验回放率(Replay Ratio)和转换寿命(Age of a Transition)。回放容量指经验存放池的大小，即经验池中存放的转换数量；经验回放率被定义为每个转换进行梯度更新的次数，即每次训练从经验池中采样的比例；经验回放中的转换寿命指每个转换产生之后进行梯度更新的总次数。通过实验得出结论：转换寿命越长学习效率越低，增大回放容量可以改善学习效果，尤其在使用 Rainbow 组件改进的 DQN 中效果提升明显。经验缓存回放是一种使用较广泛的知识迁移模式，其通过反复利用前阶段的经验参与到后续的训练过程中，在很多强化学习方法中均有应用。如何设计回放机制是决定最后学习效果的关键。

3.5 回报函数设计

回报函数一般指在智能体探索的过程中获取的奖励，很多探索困难的问题，多是由外部的稀疏奖励造成的，因此研究者们引入了内部奖励的概念，通过智能体探索时的自身属性来获得奖励。内在奖励的设计模式主要有基于动态模型预测误差的方法和基于各种信息增益的方法。

奖励塑形(Reward Shaping，RS)是一种人工设置奖励函数的方法，其使用先验知识重新构造马尔可夫决策过程(Markov Decision Process，MDP)的奖励分配，使智能体的动作选择出现偏移。除了学习环境中的反馈，RS 还额外学习一个奖励塑形函数 $F:S\times A\times S\mapsto\mathbb{R}$ 作为附加奖励。在新的 MDP 中，对于相同的动作，在原有奖励的基础上还附加一个包含先验知识的新奖励，即 $R(s,a,s')+F(s,a,s')$，奖励塑形方法可以人为地改变智能体的探索方向。好奇心模型[41]、赋能(Empowerment)[42]、变分信息最大化探索(Variational Information Maximizing Exploration，VIME)[43]等强化学习方法等都采用了不同的方式来构造额外的奖励函数参与到学习过程中。

对于内在奖励来说，比较好的方法是做非片段式的探索，即一个回合结束之后，相应的回报还继续累计而不截断。否则，一旦智能体做出某个动作导致回合结束，那么其得到的回报为零，这会使得智能体过度地规避危险而不去积极探索。但是对于外部奖励来说，比较好的方法是做片段式的探索，原因在于，如果把回报做累计计算，智能体可能会不断地在游戏开始的附近产生一些微小奖励，然后迅速自杀，再继续回到这个地方继续刷分进行累加。对于内在奖励和外部奖励这两种不同的探索模式，应该同时训练两个奖励函数进行求和得到

总的奖励函数 $V=V_E+V_I$，这样做的好处在于两个奖励函数能用不同的折扣系数来训练。内在奖励的规模在不同的环境和时间点会有很大的差异，选择合适的超参数也会变得困难。使用内在奖励标准化的方法可以解决这一难题，增大外部奖励的折扣系数可以明显提升训练效果，而内部奖励的折扣系数恰恰相反。

自然界中智慧生物学习的过程对通过强化学习达到通用人工智能有很大的借鉴意义，Silver 等[44]认为奖励最大化足以驱动自然和人工智能领域所研究的智能行为，包括知识和学习、感知能力、社交智商、语言、概括、模仿、通用智能等。他们首先提出了一个假说：智力及其相关能力可以被理解为对智能体在其环境中行动的奖励达到最大的效果。基于以上假设，一个优秀的奖励最大化智能体，在实现其目标的过程中，可能隐式地产生与智力相关的能力。对这些能力很难做形式化的理解，而奖励最大化可以为理解这些能力提供基础。最后提出了一种猜想：在实践中，智能可以从足够强大的强化学习智能体中产生，这些智能体学习如何使未来的回报最大化。因此，奖励是强化学习的直接驱动力，回报函数的设计直接决定了最后学习的效果好坏，利用源任务或训练过程中产生的知识构造回报函数可以针对性地解决训练过程中出现的具体问题。

4 深度强化学习中的知识迁移方法

深度强化学习是一种从与环境的交互反馈中进行学习的方法。强化学习智能体在建立的 MDP 模型的基础上，通过感知自身状态，依据与环境的反馈，按照某种策略生成动作并交付环境执行的过程。因此，深度强化学习中的知识迁移方法主要围绕这个过程展开。本节将详细介绍深度强化学习中与知识迁移有关的方法，按照 MDP 知识迁移、状态知识迁移、奖励知识迁移、策略知识迁移和特征知识迁移 5 类方法进行介绍，并对典型方法进行对比分析。

4.1 MDP 知识迁移

MDP 是解决大部分强化学习问题的基本框架。一个典型的强化学习问题可以被理解为是训练一个智能体与满足 MDP 标准的环境进行交互。MDP 由一个智能体和一个环境 E、一组可能的状态 S、一组的动作 A 和奖励函数 r 构成，表示为：$S\times A\rightarrow r$。专家的演示轨迹是一种可以以 MDP 的形式迁移的知识，迁移专家经验进行学习是智能体学习复杂行为策略的实用框架，这种学习过程被称为从演示中学习（Learn from Demonstrations，LfD）[45]，也叫做模仿学习（Imitation Learning）[46]或学徒学习（Apprenticeship Learning）[47]。在大多数传统的 LfD 方法中，智能体通过观察专家轨迹中状态与动作之间的映射，使用监督学习来估计一个可以近似再现这种映射的策略函数。但传统的 LfD 方法局限性较大，需要专家给出最优且足够的行为轨迹，并且专家始终与智能体在一起进行训练。结合深度强化学习可以消除这些局限。

4.1.1 使用演示的近似策略迭代

策略搜索于 1960 年首次被提出并应用于 MDP 中，而后，Munos 又提出了近似策略迭代的概念并给出了严格的理论证明[48]。针对不同的应用场景，研究人员又对其进行了各种改进，其中最小二乘策略迭代（Least-Squares Policy Iteration，LSPI）[49]最有代表性，它学习状态-动作值函数，该函数允许在没有模型的情况下进行动作选择，以及在策略迭代框架内增加策略改进。近似策略迭代[50]是一种使用动态规划的强化学习算法，Kim 等[51]在 LSPI 算法的基础上结合强化学习进行了改进，提出了使用演示的近似策略迭代算法（Approximate Policy Iteration with Demonstration，APID），它通过在策略评估步骤中添加线性的约束来修改正则化的 LSPI，使用专家演示 $D_E=\{(S_i,A_i)\}_{i=1}^{m}$ 来学习价值函数，使用自己派生的演示 D_π 来逼近新的 Q 函数。对于 D_E 中的任意状态 s_i 该价值函数都会给出学习到的专家动作 $\pi_E(s_i)$，额外引入松弛变量 ξ_i，使得该专家动作有着更高的 Q 值分类边际。

$$Q_{\text{margin}}=Q(s_i,\pi_E(s_i))-\max_{a\in A\backslash\pi_E(s_i)}Q(s_i,a)$$

其中，$Q_{\text{margin}}\geqslant 1-\xi_i$。$\xi_i$ 可以用来解决不完美的专家演示，这种方法在策略评估的过程中被实例化为最小化铰链损失函数（Hinge-loss）。

$$Q\leftarrow\arg\min_{Q}f(Q),$$

$$\text{where } f(Q)=\left\{\mathscr{L}^{\pi}(Q)+\frac{\alpha}{N_E}[1-Q_{\text{margin}}]_+\right\}$$

$\mathscr{L}^{\pi}(Q)$是最优贝尔曼残差的经验范数导致的 Q 函数的损失函数：

$$\mathscr{L}^{\pi}=\mathbb{E}_{(s,a)\sim D_\pi}\mathscr{T}^{\pi}Q(s,a)-Q(s,a)$$

其中，$\mathscr{T}^{\pi}Q(s,a)$为贝尔曼收缩算子，其值为 $R(s,a)+\gamma\mathbb{E}_{s'\sim p(\cdot\mid s,a)}[Q(s',\pi(s'))]$。实验证明，APID 可以处理理轨迹不均匀的次优专家演示。

Piot 等[52]在此基础上对 APID 进行了改进，将损失函数改为：

$$\mathscr{L}^{\pi}=\mathbb{E}_{(s,a)\sim D_\pi}\mathscr{T}^{*}Q(s,a)-Q(s,a)$$

其中，$\mathscr{T}^{*}Q(s,a)$的值为：

$$R(s,a)+\gamma\mathbb{E}_{s'\sim p(\cdot\mid s,a)}[Q(s',a')]$$

与 APID 相比，该方法在理论上可以收敛到最优的 Q 函数，原因在于 $\mathscr{L}^{\pi}$是最小化的最优贝尔曼残差（Optimal Bellman Residual）而不是经验范数。

4.1.2 使用演示的直接策略迭代

直接策略迭代是一种基于分类的策略迭代，其中策略改进被分类问题取代，因为模仿专家策略本质上是一个分类问题。Chemali 等[53]将其与演示相结合提出了使用演示的直接策略迭代算法（Direct Policy Iteration with Demonstrations，DPID），该方法使用了专家策略 π_E 中的完全演示集 D_E 和在环境中自生成的演示集 D_π。DPID 利用 $D_E\cup D_\pi$ 生成一个 Q 值的蒙特卡罗估计 $\hat{Q}$，从中可以使用贪婪策略得出学习策略 $\pi(s)=\arg\max_{a\in A}\hat{Q}(s,a)$，策略 π 通过一个损失函数 $\mathscr{L}(s,\pi_E)$来最小化专家策略的偏差。

$$\mathscr{L}(\pi,\pi_E)=\frac{1}{N_E}\sum_{i=1}^{N_E}I\{\pi_E(s_i)\neq\pi(s_i)\}$$

其中，N_E 为专家演示的样本数量，$I\{\pi_E(s_i)\neq\pi(s_i)\}$是指示函数。

4.1.3 使用演示的深度 Q 学习

除了策略迭代的方法，还可以将演示数据集成到时间差分学习中，Hester 等[54]将时间差分更新和演示动作的大边际（Margin）分类相结合提出了演示深度 Q 学习（Deep Q-lear-

ning from Demonstrations，DQfD)。DQfD 首先在演示数据上单独进行预训练，该过程的目标是学习使用一个价值函数来模拟专家演示，该价值函数满足贝尔曼方程，因此可以通过时间差分的方式进行更新。在预训练阶段，智能体从演示中小批量采样并通过 n 步双重 Q 学习损失、有监督大边际分类损失和 L2 正则化损失来更新网络。其中有监督大边际分类损失被用于对演示的分类，是至关重要的，因为专家演示不会覆盖所有可能的情形，所以很多(s,a)不会包含在内。如果只使用 Q 学习来更新训练该网络，那么该网络会将这些没覆盖到的情况向着最高值更新。加入一个大边际分类损失 $J_E(Q)$可以将这些值回归到合理范围。

$$J_E(Q)=\max_{a\in A}[Q(s,a)+l(s,a_E,a)]-Q(s,a_E)$$

其中，a_E 是专家演示在状态 s 所采取的动作，$l(s,a_E,a)$是边际函数，当 $a=a_E$ 时其值为 0，否则为正。该损失会保证其他动作的值不超过专家演示的边际。在此基础上，加入 L2 正则化损失以防止过拟合，最终用于更新该网络的整体损失函数是这 3 种损失的结合。

$$J(Q)=J_{DQ}(Q)+\lambda_1 J_E(Q)+\lambda_2 J_{L2}(Q)$$

其中，λ 用于控制权重比例。使用通过上述预训练得到的策略在环境中运行得到自生成的演示数据 D^{replay}，将其存入智能体回放缓存中并不断更新，之前的专家演示数据也被存放在演示存放缓存 D^{demo} 中一直保持不变。在小批量采样的过程中，演示数据和自生成的数据通过一个确定的比例进行控制。对于自生成的演示数据，仅会应用双重 Q 学习损失；而对于演示数据，会应用监督和双重 Q 学习两种损失。

4.1.4 使用演示的深度确定性策略梯度

Vecerik 等[55]修改了 DDPG 算法提出了从演示中学习的深度确定性策略梯度算法(Deep Deterministic Policy Gradient from Demonations，DDPGfD)。DDPGfD 与 DQfD 有着相似的想法，它们都维护两个独立的回放缓存区。为了解决奖励稀疏问题，DDPGfD 使用优先经验回放来实现奖励信息的有效传播，其被用于演示数据与自生成数据之间对样本进行优先级排序，以控制两者之间的比例。在更新 critic 函数时，DDPGfD 混合采用单步返回和 n 步返回，这有助于在稀疏奖励环境中沿轨迹传播 Q 值。

此外，DDPGfD 在每个环境步骤中进行多次学习更新，达到了数据效率和稳定性之间的平衡，并且对 actor 和 critic 网络参数进行 L2 正则化以稳定学习性能，最终的损失函数为：

$$\begin{aligned}L_{\text{Critic}}(\theta^Q)&=L_1(\theta^Q)+\lambda_1 L_n(\theta^Q)+\lambda_2 L_{\text{reg}}^{C}(\theta^Q)\nabla_{\theta^\pi}L_{\text{Actor}}(\theta^\pi)\\&=-\nabla_{\theta^\pi}J(\theta^\pi)+\lambda_2\ \nabla_{\theta^\pi}L_{\text{reg}}^{A}(\theta^\pi)\end{aligned}$$

Nair 等[56]也使用演示与 DDPG 算法相结合的方式来解决更困难的探索问题，并成功地学习到了执行长视距多步机器人任务的连续控制。其通过将事后经验回放构建的演示回放缓存区、行为克隆损失、Q 值筛选和重置演示状态 4 种方法将 DDPG 和演示结合起来，最大限度地利用演示来提高学习效率。

4.1.5 使用演示的策略优化

从策略优化的角度来看，Kang 等[57]在生成对抗模仿学习[58]的基础上扩展提出了使用演示的策略优化算法(Policy Optimization from Demonstrations，POfD)，它通过匹配当前演示和已学习的策略的使用率，诱导形成隐性的动态奖励。POfD 结合策略梯度的方法在演示数据很少且不是最优演示的情况下也可以发挥出较好的效果，算法伪代码如算法 1 所示。

算法 1 使用演示的策略优化

输入：专家演示 $D_E=\{d_1^E,\cdots,d_N^E\}$，策略和判别器参数 θ_0 和 ω_0，权重 λ_1,λ_2 和最大迭代次数 I

输出：更新后的策略 ω

1. 初始化策略和判别器参数 θ_0 和 ω_0，权重 λ_1,λ_2 和最大迭代次数 I
2. for i=1…I do
3. 采样轨迹 $D_i=\{\tau\}$，$\tau\sim\pi_{\theta_i}$
4. 采样专家轨迹 $D_i^E\subset D^E$
5. 使用梯度$\hat{\mathbb{E}}_{D_i}[\nabla_\omega\log(D_w(s,a))]+\hat{\mathbb{E}}_{D_i^E}[\nabla_\omega\log(1-D_w(s,a))]$更新 ω_i 到 ω_{i+1}
6. $\forall(s,a,r)\in D_i$ 使用 $r'(s,a)=r(a,b)-\lambda_1\log(D_{\omega_i}(s,a))$更新 D_i 中的奖励值
7. 使用策略梯度法更新策略，梯度为$\hat{\mathbb{E}}_{D_i}[\nabla_\theta\log\pi_\theta(a|s)Q'(s,a)]-\lambda_2\ \nabla_\theta H(\pi_{\theta_i})$
8. end for

4.1.6 算法分析与总结

对以上 MDP 知识迁移的相关工作进行总结，表 1 分别列出了各方法的关键思想、使用的技术、解决的问题和存在的局限。从表 1 可看出，APID 和 DPID 解决了从不完美和不完全的 MDP 演示中学习的问题。DQfD 通过从演示中学习缓解了训练冷启动的问题，但它并不适用于连续的动作空间。DDPGfD 和 POfD 给强化学习中的策略梯度算法与 MDP 知识迁移相结合提供了一种思路。

表 1 MDP 知识迁移方法

Table 1 MDP knowledge transfer methods

算法	关键思想	技术手段	解决问题	局限性
APID[50]	在策略评估步骤中加入线性约束，使用专家建议来定义这些线性约束，这些约束被用来指导近似策略迭代的优化	API	可以从不完美的 MDP 中学习最优策略	对演示数据的利用率较低
DPID[53]	利用专家演示和自生成的演示集共同进行策略评估，使用贪婪策略进行学习	DPI	可以从不完全的 MDP 中学习最优策略	专家演示和交互样本较少时效果较差
DQfD[54]	结合优先经验回放的机制对演示数据进行重要性评估	PER，DQN	预训练初始策略，减少早期探索时间，避免冷启动	对连续动作不适用
DDPGfD[55]	混合采用单步返回和 n 步返回，这有助于在稀疏奖励环境中沿轨迹传播 Q 值	PER，DDPG，AC	可解决高维连续控制问题	复杂环境下调整参数繁琐，鲁棒性较差
POfD[57]	通过匹配当前演示和已学习的策略的使用率，诱导形成隐性的动态奖励	PPO，TRPO	提高演示数据利用率	对演示数据的质量要求较高

4.2 状态知识迁移

状态是强化学习模型中最重要的数据之一，每次探索过程都是从一个状态开始到另一个状态结束。将状态作为知识迁移到新的学习任务或学习过程中可以加速智能体的探索效率，我们将这类方法叫做状态知识迁移。状态知识迁移适合解决扩展探索边界的问题，尤其是在在稀疏奖励的环境中解决探索困难的问题，状态知识迁移的方法效果显著。

4.2.1 内在好奇心模块

Pathak 等[41]针对稀疏奖励环境中的探索困难问题提出了内在好奇心模块(Intrinsic Curiosity Module，ICM)的解决方案，ICM 是一种基于奖励偏差的探索方法，它将预测奖励与实际奖励偏差看做智能体对环境的了解程度，偏差越大智能体的好奇心就越强。在 s_t 状态的智能体执行通过其当前策略 π 中抽取的动作 a_t 与环境进行交互，到达下一状态 s_{t+1}，训练策略 π 来优化通过环境得到的外部奖励 r_t^e 和由 ICM 产生的基于好奇心内在奖励 r_t^i 的总和。ICM 的技术流程如图 4 所示，首先将状态 s_t 和 s_{t+1} 编码到特征 $\varphi(s_t)$ 和 $\varphi(s_{t+1})$，使用逆向动力学模型利用这些特征预测 a_t。正向的模型以 $\varphi(s_t)$ 和 a_t 作为输入，预测 s_{t+1} 的特征表示为 $\hat{\varphi}(s_{t+1})$，利用特征空间中的预测误差作为基于好奇心的内在奖励信号。ICM 可以鼓励智能体去探索新奇的状态，但这种机制也会导致智能体在随机加噪环境中被局部的状态熵不断吸引，从而干扰对下一状态的预测。

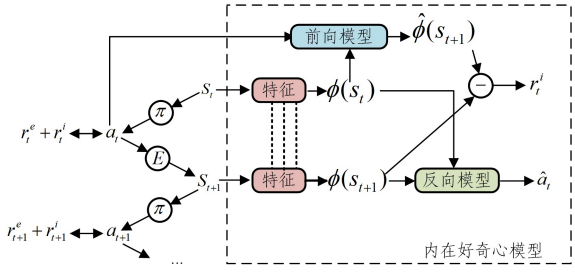


图 4 内在好奇心模型示意图

Fig. 4 Schematic diagram of ICM

4.2.2 随机网络蒸馏

受到状态访问计数的探索方法[28]的启发，Burda 等提出了随机网络蒸馏(Random Network Distillation，RND)[59]的方法来设计内在奖励，由于是高维连续空间，因此这个计数更多地可以被看成是一种密度估计。如果类似的状态之前访问的次数较少，那么说明这个状态比较新奇，就给予比较高的内在奖励。这种方法需要构建两个神经网络，其中目标网络 $f(x)$ 是一个确定性的、随机初始化的网络，可以通过它设置预测问题对插入的 x 值进行观测，预测网络 $\hat{f}(x;\theta)$ 根据智能体收集的数据通过梯度下降进行训练，目标是得到使期望均方误差 $\| \hat{f}(x;\theta) - f(x) \|^2$ 最小的参数。使用 MINST 数据集对该方法进行了验证，结果表明，出现次数越少的数字状态均方误差越大，可探索的空间也越大，证明了该方法可以用来检测状态是否新颖。在 RND 中，只要样本充分，预测误差总会是一个很小的值，可以克服随机加噪环境的困扰。

4.2.3 Novelty-Pursuit

文献[60]提出了一种名为 Novelty-Pursuit 的高效探索方法。Novelty-Pursuit 以内在动机目标探索过程为框架，基于最大状态熵的探索方法，采用随机网络蒸馏用一个到达状态频率的近似器来确定所有已探索过状态的到达频率，尽可能多地采样未知的状态空间，扩展已知的状态空间中的状态。已探索到的状态中到达频率较低的状态集合被称为探索边界(Exploration Boundary)，将探索边界进行存储，使用目标导向策略引导智能体快速到达探索边界后使用随机策略进行探索。在 Go-Explore 中需要将环境初始化为存档的中间状态进行探索[32]，而 Novelty-Pursuit 方法不受此限制，可以适用于更广泛的实验环境。

4.2.4 算法分析与总结

表 2 列出了上述状态知识迁移方法。ICM 是基于状态预测偏差的一种探索方法，可以较好地解决稀疏奖励环境中的探索问题，但是会受到局部噪声的干扰。RND 采用另一种思路来利用状态知识，其根据状态出现的次数来定义状态的重要性，在实践中采用预测状态和实际状态的均方误差来代替状态计数，效果较好。Novelty-Pursuit 在 RND 的基础上提出了一种目标导向的探索方法，其利用状态熵的最大化来确定探索的目标条件。

4.3 奖励知识迁移

在强化学习的过程中，奖励是驱动智能体进行探索的关键信息。奖励知识的利用方式主要有两种：首先是奖励塑形；另外，对于多智能体强化学习任务，我们可以将单个智能体学习到的奖励知识迁移到总体的价值函数中，以提升价值函数的学习效率。奖励知识迁移可用于稳健地驱动智能体的探索和将单智能体的价值函数扩展到多智能体环境。

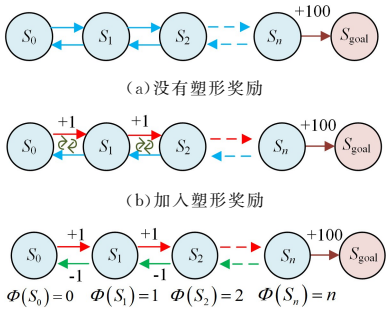
表 2 状态知识迁移方法

Table 2 State knowledge transfer methods

算法名字	关键思想	使用技术	解决问题	局限性
ICM[41]	利用预测状态和实际状态的偏差来定义状态的新奇性，驱动智能体的探索	A3C	在外部奖励稀疏的环境中探索	局部噪声环境会影响预测效果
RND[59]	根据均方误差估计状态出现的次数，均方误差越大可探索的空间越大	MSE，PPO	对于高维连续空间，用密度估计代替状态计数，特征化表示状态	不能做到全局规划
Novelty-Pursuit[60]	基于最大熵探索的思想，构建状态频率近似器辅助探索	PPO	提出一种目标导向的探索方法，可以在非确定性环境中探索	目标条件的表示需要人工干预

4.3.1 基于势能的奖励塑形

奖励塑形通过人为设计附加奖励来指导智能体训练，但是在一些问题中，人为设计的奖励函数常常会导致智能体投机取巧而学习不到最优的策略。为了解决这个问题，Ng等[61]提出了基于势能的奖励塑形（Potential Based Reward Shaping，PBRS）。PBRS将两个势函数的差值作为奖励塑形函数 $F(s,a,s')=\gamma\Phi(s')-\Phi(s)$，相当于给每个状态定义一个势能，从低势能的状态到高势能的状态给予一个正奖励，反之则给予一个负奖励，这样就解决了原始的奖励塑形方法中智能体在原地循环累加奖励的问题。如图5所示，图5(a)表示在没有塑形奖励的环境中，在状态之间转移智能体得不到任何奖励而难以到达最终的目标状态；图5(b)表示加入设定的微小奖励值后，智能体会在原地打转来重复获取微小奖励；图5(c)表示给每个状态赋予势能，可以避免智能体在原地打转的问题。可以看出，利用这种奖励塑形模式，PBRS可以显著减少学习最优策略所需要的训练时间，并且在多智能体系统中可以提升最终联合策略的性能。该算法不改变单个智能体学习的最优策略，也不改变多个智能体共同学习的纳什均衡，保证了策略的一致性。



(a)没有塑形奖励

(b)加入塑形奖励

(c)加入基于势能的塑形奖励

图5 智能体在环境中获取奖励的示意图

Fig. 5 Schematic diagram of agent obtaining reward in environment

4.3.2 动态势能奖励塑形

PBRS的局限性在于假设状态的势能在学习过程中不会发生动态变化，而这种假设时常不成立，特别是当奖励是自动生成时。Devlin等[62]对PBRS进行了进一步扩展，提出了一种基于动态势能的奖励塑形方法（Dynamic Potential Based Reward Shaping，DPBRS），该方法使势能同时成为状态和时间的函数以允许动态改变，并在单智能体和多智能体系统中保持策略的一致性。DPBRS将时间参数引入到塑形函数中。

$$F(s,t,s',t')=\gamma\Phi(s',t')-\Phi(s,t)$$

其中，t 为智能体到达上一状态 s 的时间，t' 为智能体到达当前状态 s' 的时间。

4.3.3 基于 N 步返回的价值函数迁移

在稀疏交互的多智能体系统中利用单智能体知识可以大大加快多智能体的学习过程。Liu等基于一个新的MDP相似概念提出了扩展性更高的基于 N 步返回的价值函数迁移[63]。该方法首先根据MDP的 N 步返回（N-Step Return，NSR）值定义MDP的相似性，然后提出了两种基于深度神经网络的知识传递方法，即直接价值函数迁移和基于NSR的值函数迁移。通过NSR可以测量单智能体和多智能体环境之间的MDP相似度并识别出交互区域，以此来确定单智能体策略是否适合当前的联合状态。该方法可以实现选择性迁移，更好地避免负迁移，减少学习的时间，同时具有较好的渐近性能。

4.3.4 算法分析与总结

对以上3种方法进行了对比分析，结果如表3所列。PBRS是一个基于奖励塑形提出的较为成熟的方法，可以在保持策略不变的条件下利用势能函数激励智能体进行探索。DPBRS在PBRS的基础上改进了势能的塑造奖励过程，缩小了模型的规模，并且可以将单智能体策略扩展到多智能体领域。除了采用奖励塑形的方法，还有直接迁移价值函数的代表NSR-VFT，其通过匹配MDP的相似度来进行迁移，实现了从单智能体环境到多智能体的选择性迁移。此外，文献[64-65]也为价值函数的迁移提供了另外的思路。

表3 奖励知识迁移方法

Table 3 Reward knowledge transfer methods

算法	关键思想	技术手段	解决问题	局限性
PBRS[61]	将两个势函数的差值作为奖励塑形函数	RS	在激励智能体探索的同时保持策略不变性	模型规模较大
DPBRS[62]	使势能同时成为状态和时间的函数以允许动态改变	RS，MARL	缩小模型的规模，将单智能体策略扩展到多智能体	需要学习额外的值函数
NSR-VFT[63]	通过NSR测量单智能体和多智能体环境之间的MDP相似度并识别出交互区域，以此来确定单智能体策略是否适合当前的联合状态	NSR，DQN	实现选择性迁移，减少学习的时间，提高了性能	计算量较大，内存需求较大

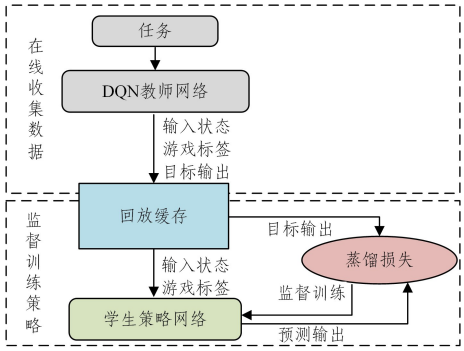
4.4 策略知识迁移

强化学习的目标是学习到一个奖励最大化的策略，在模型相似的任务之间可以直接把策略知识迁移到新的任务中使用，这个过程叫做策略复用。但在一些任务之间使用直接迁移策略的效果较差，这时我们可以建立一个通用的策略，在此基础上再根据任务的特点进行进一步的训练。当有多个已经训练好的模型时，可以用策略知识迁移的方法将既有策略提炼用于新的相似任务。

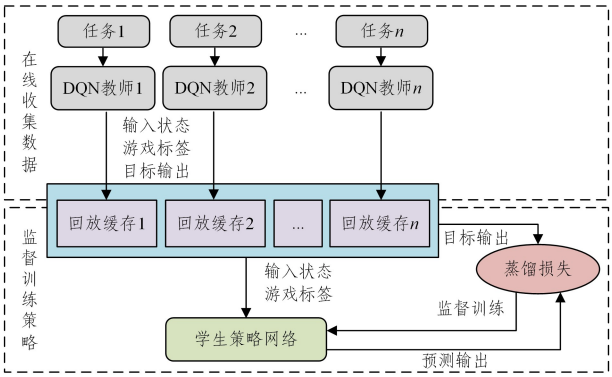
4.4.1 策略蒸馏

在使用DQN学习复杂视觉任务的策略时，要获得良好的性能，需要相对较大的网络和多次的训练。Rusu等[66]提出了一种名为策略蒸馏（Policy Distillation）的新方法，如图6所示，DQN智能体定期地将游戏数据添加到回放缓存中。策略蒸馏可以用来提取深度强化学习中智能体的策略，并训练一个新的更小、更高效的能达到专家水平的网络。学生策略通过最小化教师策略 π_T 和学生策略 π_S 之间的动作分布差异

进行学习，对于 N 个任务，使用 N 个教师策略进行学习，每个教师策略产生一个数据集 $D^T=\{s_i,q_i\}_{i=0}^N$，由状态和相应的 q 值组成，使用负对数似然损失、均方误差和 KL 散度 3 种方法进一步提炼教师策略，得到一个学生智能体。



(a)单任务数据收集和策略蒸馏



(b)多任务数据收集和策略蒸馏

图 6 单任务策略蒸馏和多任务策略蒸馏

Fig. 6 Single task strategy distillation and multi task strategy distillation

此外，还可以使用相同的方法将多个特定任务的策略合并为单个策略。该方法的可行性在雅达利游戏中得到了证实，表明多任务蒸馏的强化学习智能体优于单任务以及联合训练的 DQN 智能体。即使不使用迭代方法，不允许学生网络控制它训练的数据分布，蒸馏也可以应用于强化学习。

4.4.2 Distral

另一种策略蒸馏的思路是最大化教师策略访问学生网络生成轨迹的概率，Teh 等提出了一种鲁棒的多任务联合训练方法[67]，称为 Distral，如图 7 所示，其思想是通过多任务学习让智能体学习到一些共有的策略知识，这样就能够在一个新环境下通过少量样本学习到新的策略。其在多个任务各自策略的基础上建立一个中间策略 π_0，各个不同的策略综合起来蒸馏提炼出中间策略 π_0，各个任务中的策略也在 π_0 的指导下进行学习。对于多个任务及对应的策略 $\pi_i=(S,A,p_i(s'|s,a),\gamma,R_i(a,s))$，和由这些策略蒸馏出的中间策略 π_0，要最大化的最终目标是：

$$J(\pi_0,\{\pi_i\}_{i=1}^n)=\sum_i \mathbb{E}_{\pi_i}\left[ac_{\mathrm{KL}}\gamma^t\log\frac{\pi_i(a_t|s_t)}{\pi_0(a_t|s_t)}-c_{\mathrm{Ent}}b\right]$$

$$=\sum_i \mathbb{E}_{\pi_i}\left[a+\frac{\gamma^t\alpha}{\beta}\log\pi_0(a_t|s_t)-\frac{1}{\beta}b\right]$$

该目标主要约束了各个任务的策略 π_i 不偏离 π_0，其中 $a=\sum_{t\geqslant 0}\gamma^t R_i(a_t,s_t)$，$b=\gamma^t\log\pi_i(a_t|s_t)$，$c_{\mathrm{KL}}\geqslant 0$ 是决定 KL 散度，$c_{\mathrm{Ent}}\geqslant 0$ 是熵，$\alpha=c_{\mathrm{KL}}/c_{\mathrm{KL}}+c_{\mathrm{Ent}}$，$\beta=1/c_{\mathrm{KL}}+c_{\mathrm{Ent}}$。

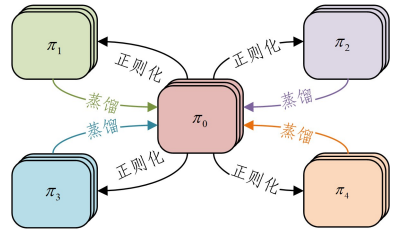


图 7 Distral 框架示意图

Fig. 7 Schematic diagram of Distral

Distral 提出了另一种更好的方法来表示策略，即各个特定策略 π_i 表示为与 π_0 共有的部分与其自己特有部分的和，这样做的目的是使各特定策略在学习的过程中可以集中精力学习自己特有的部分。中间策略 π_0 使用一个神经网络 $h_{\theta_0}(a|s)$ 来表示。

$$\hat{\pi}_0(a_t|s_t)=\frac{\exp(h_{\theta_0}(a_t|s_t))}{\sum_{a'}\exp(h_{\theta_0}(a'|s_t))}$$

对于特定的策略 π_i，使用中间策略 π_0 的神经网络 $h_{\theta_0}(a|s)$ 和各自的神经网络 $f_{\theta_i}(a|s)$ 来表示：

$$\hat{\pi}_i(a_t|s_t)=\hat{\pi}_0^{\alpha}(a_t|s_t)\exp(\beta\hat{A}_i(a_t|s_t))$$

$$=\frac{\exp(\alpha h_{\theta_0}(a_t|s_t)+\beta f_{\theta_i}(a_t|s_t))}{\sum_{a'}\exp((\alpha h_{\theta_0}(a'|s_t)+\beta f_{\theta_i}(a'|s_t))}$$

这是一个以 π_0^{α} 为先验的玻尔兹曼策略，其中 $\hat{A}_i(a_t|s_t)$ 为优势函数。

4.4.3 Actor-mimic

对于从多个任务的策略中提取通用策略的问题，Parisotto 等[68]也提出了一种方法演示者-模仿者(Actor-mimic)，其将多个任务中的 DQN 网络转化为一个玻尔兹曼策略。

$$\pi_{E_i}(a|s)=\frac{e^{\tau^{-1}Q_{E_i}(s,a)}}{\sum_{a'\in A_{E_i}}e^{\tau^{-1}Q_{E_i}(s,a')}}$$

然后最小化这些策略和目标策略网络之间的差距。Actor-mimic 的目标是从多任务各自的策略中产生一个名为 AMN(Actor-Mimic Network)的通用专家策略，训练的目标是最小化上述策略和 AMN 策略之间的交叉熵损失函数。

$$\mathcal{L}^i_{\mathrm{policy}}(\theta)=\sum_{a\in A_{E_i}}\pi_{E_i}(a|s)\log\pi_{\mathrm{AMN}}(a|s;\theta)$$

此外，Actor-mimic 还定义了一个特征回归网络(Feature Regression Network) $h_{\mathrm{AMN}}(s;\theta)$ 及其对应的特征回归损失。

$$\mathcal{L}_{\mathrm{FR}}(\theta,\theta_{f_i})=f_i(h_{\mathrm{AMN}}(s;\theta);\theta_{f_i})-h_{E_i}(s)_2^2$$

该回归损失与交叉熵损失函数一起指导训练。

4.4.4 算法分析与总结

表 4 列出了上述策略知识迁移的相关工作，策略蒸馏的方法突破性地实现了将多个任务的特定策略合并为单个策略。Distral 结合多任务学习迁移策略知识，使得在新环境中只需要少量的样本就可以学习到新的策略，但是它对任务的相关性要求较高。Actor-mimic 方法可以把多个环境中的策略迁移到新环境中，但其存在一定的负迁移。除此之外，在文献[69-71]中也提出了多种迁移策略知识的方法。

表 4 策略知识迁移方法

Table 4 Policy knowledge transfer methods

算法	关键思想	技术手段	解决问题	局限性
策略蒸馏[66]	通过最小化教师策略和学生策略之间的动作分布差异进行学习	DQN、MSE、KL 散度	将多个任务的特定策略合并为单个策略	收敛性能有时不理想
Distral[67]	最大化教师策略访问学生网络生成轨迹的概率	Soft Q Learning、KL 散度	通过多任务学习学习到共有策略知识，在新环境下可以通过少量样本学习新策略	对任务的相关性要求较高
Actor-mimic[68]	最小化玻尔兹曼策略和 AMN 策略的交叉熵损失函数，引入特征回归损失	DQN	可以把多个环境中的行动策略迁移到新的环境中	存在负迁移的现象

4.5 特征知识迁移

特征是深度强化学习中的一种隐性知识表示，这类知识虽然不像状态、奖励等强化学习的基本要素一样直接参与到学习过程中，但也是学习过程中不可或缺的一部分，如神经网络中的特征参数。这种知识迁移方法适合在多任务之间进行迁移，在之前的神经网络参数的基础上进行微调或利用之前的参数知识在新的任务中进行训练，可以有效地利用先前任务的训练成果。

价值函数也可以被分解为特征表示，受到 Dayan 的后继代表（Successor Representation，SR）[72] 的启发，后继特征（Successor Features，SFs）[73] 将 SR 从离散空间扩展到连续空间以便使用函数逼近，SFs 将价值函数解耦表示为环境的动态和奖励 $r(s,a,s')=\varphi(s,a,s')^{\mathrm{T}}\boldsymbol{w}$，其中 $\varphi(s,a,s')$ 是 (s,a,s') 的后继特征，它和环境的动态有关，因此只要环境本身不变，这个特征就是固定的，$\boldsymbol{w}\in\mathbb{R}^d$ 为权重，它决定了不同任务的奖励函数。

当任务和任务之间从表面上看关联性较小时，我们可以考虑使用特征知识迁移的方法，提取任务中抽象的特征属性并应用于新任务中。

4.5.1 渐进式神经网络

通过微调进行神经网络的迁移不适合跨多个任务的迁移，Rusu 等[74] 提出了渐进式神经网络结构（Progressive Neural Networks，PNN）以渐进的方式实现跨多任务的知识迁移，它保留了神经网络中的特征知识，使用横向连接将之前任务的特征传递给新任务的神经网络进行训练。PNN 由若干列神经网络组成，每列都是训练一个特定任务的神经网络。首先构造一个多层神经网络，从一个单独的列（Column）开始训练第一个任务，该神经网络共有 L 个隐藏层 $h_i^{(1)}\in\mathbb{R}^{n_i}$，其中 n_i 是第 i 层的单元个数，使用参数 $\theta^{(1)}$ 训练到收敛；然后在构建第二个多层神经网络时，固定第一列的神经网络参数 $\theta^{(1)}$，将上一列神经网络的每一层都通过适配器层 a 处理连接到第二列神经网络的每一层作为额外输入，$h_i^{(2)}$ 通过横向连接接收来自 $h_{i-1}^{(2)}$ 和 $h_{i-1}^{(1)}$ 的输入。构建第三个神经网络时，将前两列的神经网络参数固定，然后用同样的方法将前两个网络中的每一层连接到第三个神经网络中。用这种连接方式扩展到 k 个任务时，可以记为：

$$h_i^{(k)}=f(\boldsymbol{W}_i^{(k)}h_{i-1}^{(k)}+\sum_{j<k}\boldsymbol{U}_i^{(k:j)}h_{i-1}^{(j)})$$

图 8 清晰地描述了这一过程，上面的两行（虚线箭头）分别按照任务 1 和任务 2 进行训练。标记为 a 的箭头表示经过适配器层处理过。第三行是为最后一个任务添加的，它可以访问所有之前学习过的特性。

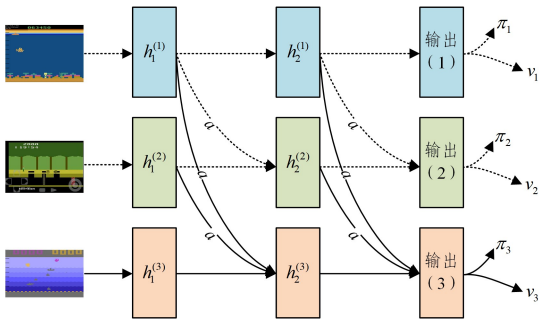


图 8 渐进神经网络结构

Fig. 8 Progressive neural network structure

4.5.2 动态模型奖励解耦

当前的强化学习方法可以成功地学习单个任务，但加入一些适度的扰动后，泛化效果较差。Zhang 等[75] 受到后继特征的启发提出了一种解耦（Decoupling）的学习策略来解决此问题，并称之为动态模型奖励解耦（Decoupling Dynamics and Reward，DDR）。该方法创建了一个共享的表征空间 z，其中的知识可以被鲁棒地迁移。这是一种基于模型的方法，它学习动态模型，将观察到的状态 s 映射到潜在表征 z，智能体使用 AC 算法从潜在表征 z 中学习价值函数和策略。这种潜在表征方法使得知识可以在具有不同奖励但是动态转换相同的任务之间进行迁移，动态模型可以直接用于目标任务，算法伪代码如算法 2 所示。

算法 2 动态模型奖励解耦

输入：$\theta_{\mathrm{dynamics}}$，$\mathrm{h_d}$，$\lambda_{\mathrm{inv}}$，$\lambda_{\mathrm{dec}}$，$\lambda_{\mathrm{for}}$

输出：$\theta'_{\mathrm{dynamics}}$

1. 初始化模型参数 $\theta_{\mathrm{dynamics}}$，动态 LSTM 的隐藏状态 $\mathrm{h_d}$
2. 设置动态的超参数 λ_{inv}，λ_{dec}，λ_{for}
3. for e=1，…，E do
4. for($\mathrm{s_i}$，$\mathrm{a_i}$，$\mathrm{s'_i}$)，i=0，1，…，N do
5. 编码 $\mathrm{z_i}=\mathrm{f_{enc}}(\mathrm{s_i};\theta_{\mathrm{enc}})$，$\mathrm{z'_i}=\mathrm{f_{enc}}(\mathrm{s'_i};\theta_{\mathrm{enc}})$
6. $\hat{\mathrm{z}}\leftarrow\mathrm{f_{for}}(\mathrm{z_i},\mathrm{a})$
7. $\hat{\mathrm{a}}\leftarrow\mathrm{f_{inv}}(\mathrm{z_i},\mathrm{z_i'})$
8. 解码 $\hat{\mathrm{s}}_\mathrm{i}=\mathrm{f_{dec}}(\mathrm{z_i};\theta_{\mathrm{dec}})$；$\hat{\mathrm{s}}_\mathrm{i}'=\mathrm{f_{dec}}(\mathrm{z_i'};\theta_{\mathrm{dec}})$
9. $\mathrm{L_{dynamics}}(\theta_{\mathrm{dynamics}})=\sum_{\mathrm{t}=0}^{\mathrm{T}}(\lambda_{\mathrm{dec}}\mathrm{L_{t,dec}}+\lambda_{\mathrm{for}}\mathrm{L_{t,for}}+\lambda_{\mathrm{inv}}\mathrm{L_{t,inv}})$
10. 更新 $\theta_{\mathrm{dynamics}}$ 为 $\theta'_{\mathrm{dynamics}}$

11. end for
12. end for

4.5.3 广义策略更新

强化学习中的迁移指发生在任务内部也发生在任务之间的泛化过程。Barreto 等[76]提出了一个迁移框架，此场景是奖励功能在任务之间改变，但环境保持不变。该框架主要基于两个关键思想：后继特征(Successor Features，SFs)和广义策略改进(Generalized Policy Improvement，GPI)，广义策略改进是动态规划的策略改进操作的泛化，它考虑的是一组策略而不是单个策略。这两种想法结合在一起形成了一种方法，可以无缝地集成到强化学习框架中，并允许跨任务自由交换信息。该方法还能为迁移策略提供性能保证，甚至在任何学习发生之前。通过专门提供的基本组件，可以在该方法的基础上构建能够跨各种任务出色执行的智能体，从而扩展 GPI 可以成功处理的环境范围。

Barreto 等[77]在上述想法的基础上加入广义策略评估，扩展为广义策略更新(Generalized Policy Update，GPU)。广义策略评估(Generalized Policy Evaluation，GPE)是将强化学习中的另一种基本操作策略评估从单任务中的操作推广到任务集合中的操作，通过后继特征使用不同的偏好集合评估一条路线(策略)，与 GPI 结合可以加快解决强化学习问题。

图 9 给出了广义策略更新生成策略的过程。

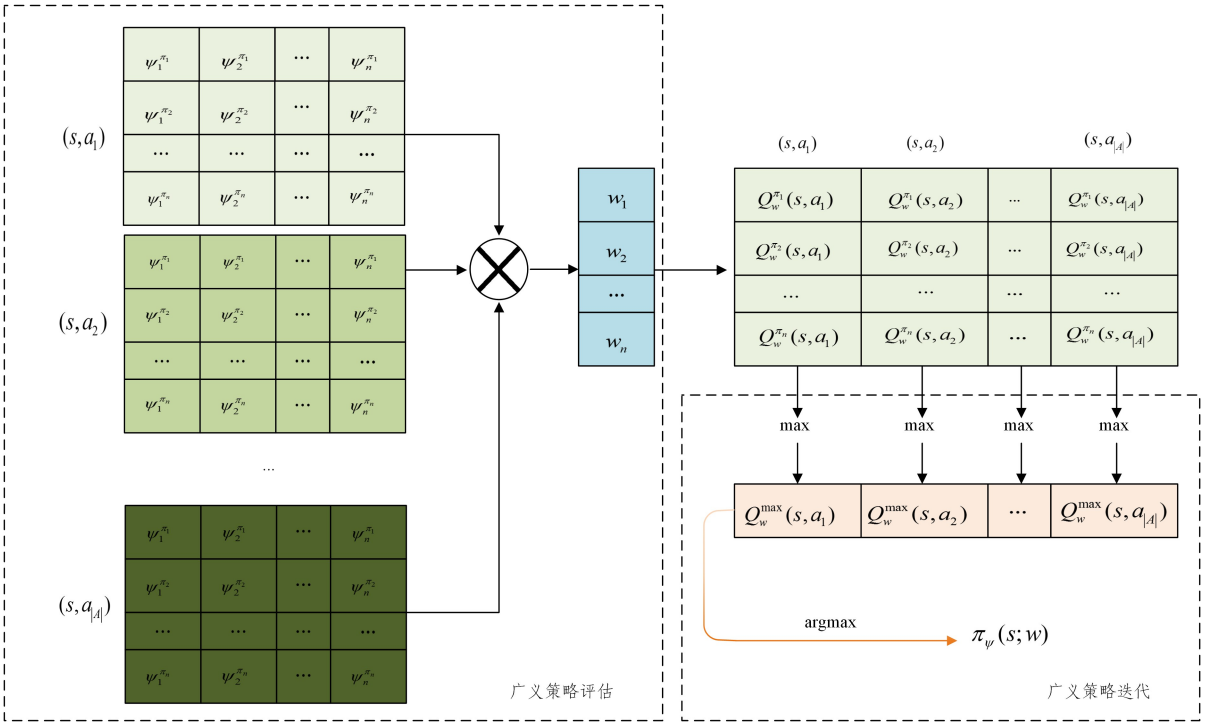


图 9 通过广义策略更新实现策略 π_ψ 的过程

Fig. 9 Process of implementing policy π_ψ through generalized policy updates

4.5.4 通用后继特征近似器

Schaul 等[78]在 2015 年的一篇论文中提出了通用价值函数近似器(Universal Value Function Approximators，UVFA)，以指导智能体完成多种目标的任务。UVFA 先在一个任务空间上对若干个不同任务进行训练，如果这些任务和新任务在任务空间中服从同一概率分布，则训练出来的模型就可以很好地迁移到新任务中。Borsa 等[79]在 UVFA 的基础上结合 SFs 的即时推断性和 GPI 的强通用性，扩展提出了通用后继特征近似的方法(Universal Successor Features Approximators，USFA)，其可以学习到更一般的值函数。USFA 将每个任务和对应的策略进行参数化，使得算法对参数化空间中的任何任务和策略都能够很快得到对应的值函数，进而提升策略，其流程如图 10 所示。因此，该算法只需要在一定的任务分布和策略分布下学习到更一般的值函数表征，就能够得到新任务下比较好的策略，实现任务迁移。

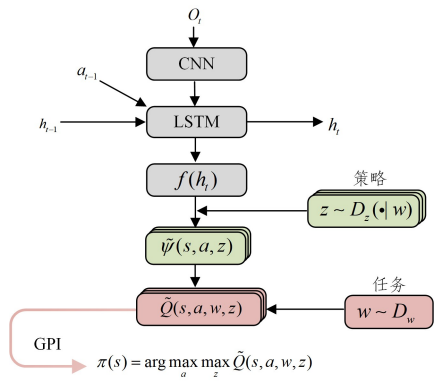


图 10 USFA 流程图

Fig. 10 USFA flow chart

4.5.5 算法分析与总结

表 5 列出了以上几种典型的特征知识迁移方法，其中 PNN 提出了一种迁移神经网络特征参数的重要手段，可以跨多个任务而不是仅在两个任务之间进行迁移，但是随着任务

的增加，其参数也大量增加。DMRD 通过创建共享的表征空间，在动态转换相同的任务之间进行迁移，对任务的相关性要求较高。GPU 结合了 GPI 和 GPE 的方法，可以在相同环境中提炼任务的特征维度，通过抽象的特征表示任务，大大减少学习新任务所需的样本数。USFA 将任务和策略都进行参数化表达或将其映射到线性空间中作为值函数的输入，可以得到通用的值函数表达，但由于其对每个任务和策略都单独进行操作，计算代价较大。

表 5 特征知识迁移方法

Table 5 Feature knowledge transfer methods

算法	关键思想	使用技术	解决问题	局限性
PNN[74]	保留了神经网络中的特征知识，使用横向连接将之前任务的特征传递给新任务的神经网络训练	A3C	渐进迁移神经网络的特征参数，实现跨多个任务的知识迁移	参数会随着任务的增加而大量增加
DDR[75]	创建共享的表征空间，学习动态模型，将观察到的状态映射到潜在表征空间，智能体从潜在表征中学习	AC，LSTM	可以在具有不同奖励但是动态转换相同的任务之间进行迁移	可移植性较差，对任务的相关性要求较高
GPU[76]	将策略评估和策略迭代推广到任务集中	GPI，GPE	在样本数很少的情况下，进行同一环境中不同任务的迁移	对环境的要求较高，任务需要有限个特征维度
USFA[79]	将每个任务和对应的策略进行参数化，使得算法对参数化空间中的任何任务和策略都能够很快得到对应的值函数，进而提升策略	GPI	可以学习到更一般的值函数表征	数据量和计算量较大

5 研究进展及应用

在第 4 节中，我们按照迁移的知识类型介绍了若干有代表性的深度强化学习方法，每种方法都有各自的优势，对比情况如表 6 所列。

表 6 知识迁移算法总结

Table 6 Summary of knowledge transfer algorithms

算法类型	算法代表	优势
MDP 知识迁移	[51-58]	解决实时交互不足时利用既有数据学习的问题
状态知识迁移	[28,41,59-60]	解决在稀疏奖励的环境中探索困难的问题
奖励知识迁移	[61-65]	可将单智能体的价值函数迁移到多智能体
策略知识迁移	[66-71]	从多任务中提取通用策略用于新任务的训练
特征知识迁移	[74-79]	寻找任务之间抽象的隐性关联进行迁移

MDP 知识迁移相对比较成熟，对于常用的深度强化学习算法均有成型的解决方案，适用于给定演示数据的情况，为离线强化学习的研究提供了解决途径。状态知识迁移主要是对智能体探索过程中的状态空间加以利用，它引入除了环境奖励因素之外的其他类型奖励，非常适合在稀疏奖励环境中探索时使用。对于如何利用状态知识来加速训练，本文在 4.2 节中介绍了几种方法，但还有较大的研究空间。奖励知识迁移利用奖励塑形和迁移价值函数的方法对强化学习的奖励系统进行改造，对于前者的研究方向为缩减模型的规模，其通常和状态知识迁移结合起来使用以加速智能体的学习，后者可以实现将单智能体中的价值函数选择性地迁移到多智能体环境中。策略知识迁移从多个任务的特定策略中提取策略知识，通过提取已有的策略知识来加速新任务中的策略学习，这些源任务可以是同一环境中的不同任务，也可以是不同环境中的特定任务，使用该方法可以大幅减少新任务的训练时间。需要注意的是，根据迁移条件的不同，源任务的选择需要满足一定的要求。此外，在 4.5 节中还介绍了几种将强化学习中的特征参数作为知识进行迁移的方法，包括网络特征参数、共享表征空间、将任务和策略等使用若干维度特征进行表示，这类方法提取了任务之间的抽象联系，可以用于通用强化学习的研究。

在通用视频游戏 AI（General Video Game AI，GVG-AI）中，游戏的样本效率一直是强化学习中的一大难题，Braylan 等[80]提出了一种迁移学习方法来解决在有限数据下训练视频游戏 AI 的问题。该方法将游戏分解为对象，学习对象模型并将模型从已知的游戏迁移到不熟悉的游戏，以指导学习。该方法提高了样本效率和预测精度，使探索更有效。通过从以前学习的游戏中迁移对象模型，可以加速游戏智能体的训练。

为了高效地完成任务，机器人经常要作为一个团队一起工作。然而，鉴于制造这些机器人的公司和研究实验室越来越多，有时难以预先协调相关团队，这就需要机器人自己学习团队协作的策略。以往对机器人团队协作的研究主要集中在相对简单的领域，对于复杂环境的团队协作，Barrett 等[81]提出了一种新的算法——快速适应队友提高合作策略的规划与学习（Planning and Learning to Adapt quickly to Teams to Improve Cooperation-Policy，PLASTIC-Policy）。PLASTIC-Policy建立在现有的临时团队协作方法[82]之上，它使用基于策略的无模型方法，并且能独立于学习到的策略的状态空间大小。具体来说，PLASTIC-Policy 学习与过去的队友合作的策略，并重用这些策略以快速适应新的队友。

谷歌健康的 Roy 等[83]提出了基于顺序子网络路由（Sequential Subnetwork Routing，SeqSNR）架构的多任务迁移学习模型，该模型的任务是预测患者进入 ICU 后 24～48 h 内每小时发生的不良事件。选择的预测因素包括急性肾损伤、连续性肾脏替代治疗透析、血管加压药和正性肌力药的给药、机械通气、死亡率和剩余住院时间。多任务迁移学习提供了一种有效的方法来捕捉器官系统之间的相互依赖性并平衡竞争风险。在实践中，联合训练的任务通常会因为负迁移而相互损害，使用 SeqSNR 可以减轻这种影响。SeqSNR 基于 Johnson 等[84]的工作使用编码器和 RNN 堆栈的分层模块化，自动优化了信息在多个任务之间的共享方式，它在标签效率方面优于一般的多任务和单任务学习。

6 结论

以深度神经网络为代表的人工智能系统通常被认为是不透明的、不可解释的，但 Google Brain 的研究表明[85]，基于深度强化学习方法训练出的神经网络模型在概念探测、行为变化等方面学习到的国际象棋知识和内部学习机制与人类对国际象棋的概念密切相关。然而，一个纯粹的数据驱动机器学习模型可能无法满足自然法则或任务约束，尽管这些法则和约束对于人工智能系统的模型至关重要[86]。因此，通过在学习的过程中利用知识迁移方法添加先验知识，一方面能够提高以深度强化学习为代表的机器学习方法的学习效率，提升其泛化性能；另一方面也能够更好地实现数据与知识混合驱动在智能生成中更好地发挥作用。在知识迁移的方法上，除了将人类的经验知识和历史数据引入深度强化学习的训练过程中，将强化学习这种交互式学习模式过程中的隐形知识从源任务向目标任务迁移也是未来深度强化学习能力提升的关键。

结束语 本文介绍了深度强化学习中的知识存在形式，针对深度强化学习中的知识迁移方法研究现状进行了归纳、总结，并对深度强化学习中的知识迁移方法进行了展望。然而，虽然深度强化学习中的知识迁移方法在理论与应用层面有了一定的研究，但仍然还有很多亟待解决的问题，如知识图谱迁移[87]、博弈强化学习中的平衡迁移[88]等。在若干知识迁移方法中仍然存在鲁棒性差、适用条件苛刻、适用模型简单、在复杂模型中扩展性弱等问题。因此，将更为通用的知识迁移方法与框架引入深度强化学习中，提升深度强化学习解决问题的能力是深度强化学习发展的趋势。

参考文献

[1] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with Deep Reinforcement Learning[J]. arXiv: 1312.5602, 2013.

[2] TORRADO R R, BONTRAGER P, TOGELIUS J, et al. Deep Reinforcement Learning for General Video Game AI[C]// 14th IEEE Conference on Computational Intelligence and Games, CIG 2018. IEEE Computer Society, 2018: 14-17.

[3] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of go without human knowledge[J]. Nature, 2017, 550(7676): 354-359.

[4] GU S, HOLLY E, LILLICRAP T, et al. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates[C]// 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 3389-3396.

[5] LI J, MONROE W, RITTER A, et al. Deep Reinforcement Learning for Dialogue Generation[C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016: 1192-1202.

[6] ANDERSEN P A, GOODWIN M, GRANMO O C. Deep RTS: a game environment for deep reinforcement learning in real-time strategy games[C]// 2018 IEEE Conference on Computational Intelligence and Games (CIG). IEEE, 2018: 1-8.

[7] LING Y, HASAN S A, DATLA V, et al. Learning to diagnose: assimilating clinical narratives using deep reinforcement learning [C]// Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2017: 895-905.

[8] HESSEL M, MODAYIL J, VAN HASSELT H, et al. Rainbow: Combining improvements in deep reinforcement learning[C]// Thirty-second AAAI Conference on Artificial Intelligence. 2018: 156-167.

[9] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. Massachusetts: MIT press, 2018.

[10] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[11] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms[C]// International Conference on Machine Learning. PMLR, 2014: 387-395.

[12] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[C]// ICLR (Poster). 2016.

[13] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]// International Conference on Machine Learning. PMLR, 2016: 1928-1937.

[14] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization[C]// International Conference on Machine Learning. PMLR, 2015: 1889-1897.

[15] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv: 1707.06347, 2017.

[16] RASHID T, SAMVELYAN M, SCHROEDER C, et al. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning[C]// International Conference on Machine Learning. PMLR, 2018: 4295-4304.

[17] SUNEHAG P, LEVER G, GRUSLYS A, et al. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based on Team Reward[C]// Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. 2018: 2085-2087.

[18] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6382-6393.

[19] ASLANIDES J, LEIKE J, HUTTER M. Universal reinforcement learning algorithms: survey and experiments[C]// Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017: 1403-1410.

[20] WATKINS C J C H, DAYAN P. Q-learning[J]. Machine Learning, 1992, 8(3/4): 279-292.

[21] FISCHER A, IGEL C. An introduction to restricted Boltzmann machines[C]// Iberoamerican Congress on Pattern Recognition. Berlin: Springer, 2012: 14-36.

[22] RIEDMILLER M. Neural fitted Q iteration-first experiences with a data efficient neural reinforcement learning method [C]// European Conference on Machine Learning. Berlin: Springer, 2005: 317-328.

[23] ZHOU Z H, FENG J. Deep forest: towards an alternative to deep neural networks[C]// Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017: 3553-

3559.

[24] YOSINSKI J，CLUNE J，BENGIO Y，et al. How transferable are features in deep neural networks? [C]// Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2. 2014：3320-3328.

[25] KRIZHEVSKY A，SUTSKEVER I，HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks[C]// NIPS. 2012：654-669.

[26] AYTAR Y，PFAFF T，BUDDEN D，et al. Playing hard exploration games by watching YouTube[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018：2935-2945.

[27] HENDERSON P，ISLAM R，BACHMAN P，et al. Deep reinforcement learning that matters[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2018，32(1)：245-256.

[28] BELLEMARE M G，SRINIVASAN S，OSTROVSKI G，et al. Unifying count-based exploration and intrinsic motivation[C]// Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016：1479-1487.

[29] WATKINS C J C H. Learning from delayed rewards[D]. Cambridge：University of Cambridge，1989.

[30] KAELBLING L P，LITTMAN M L，MOORE A W. Reinforcement learning：A survey[J]. Journal of Artificial Intelligence Research(S1076-9757)，1996，4：237-285.

[31] LI C X，CAO L，CHEN X L，et al. Cloud Reasoning Model-based Exploration for Deep Reinforcement Learning[J]. Journal of Electronics & Information Technology，2018，40(1)：244-248.

[32] ECOFFET A，HUIZINGA J，LEHMAN J，et al. First return，then explore[J]. Nature，2021，590(7847)：580-586.

[33] SCHAUL T，QUAN J，ANTONOGLOU I，et al. Prioritized Experience Replay[C]// ICLR(Poster). 2016：1312-1320.

[34] NARASIMHAN K，KULKARNI T，BARZILAY R. Language Understanding for Text-based Games using Deep Reinforcement Learning[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015：1-11.

[35] HOCHREITER S，SCHMIDHUBER J. Long short-term memory[J]. Neural Computation，1997，9(8)：1735-1780.

[36] HOU Y，LIU L，WEI Q，et al. A novel DDPG method with prioritized experience replay[C]// 2017 IEEE International Conference on Systems，Man，nd Cybernetics(SMC). IEEE，2017：316-321.

[37] HEESS N，WAYNE G，SILVER D，et al. Learning continuous control policies by stochastic value gradients[J]. Advances in Neural Information Processing Systems，2015，28：1056-1068.

[38] MAHMOOD A R，VAN HASSELT H，SUTTON R S. Weighted importance sampling for off-policy learning with linear function approximation[C]// NIPS. 2014：3014-3022.

[39] HORGAN D，QUAN J，BUDDEN D，et al. Distributed Prioritized Experience Replay [C] // International Conference on Learning Representations. 2018.

[40] FEDUS W，RAMACHANDRAN P，AGARWAL R，et al. Revisiting fundamentals of experience replay[C] // International Conference on Machine Learning. PMLR，2020：3061-3071.

[41] PATHAK D，AGRAWAL P，EFROS A A，et al. Curiosity-driven exploration by self-supervised prediction[C]// International Conference on Machine Learning. PMLR，2017：2778-2787.

[42] MOHAMED S，REZENDE D J. Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning [C]// NIPS. 2015：456-468.

[43] HOUTHOOFT R，CHEN X，DUAN Y，et al. VIME：variational information maximizing exploration[C] // Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016：1117-1125.

[44] SILVER D，SINGH S，PRECUP D，et al. Reward is enough[J]. Artificial Intelligence，2021：1035-1046.

[45] ARGALL B D，CHERNOVA S，VELOSO M，et al. A survey of robot learning from demonstration[J]. Robotics and Autonomous Systems，2009，57(5)：469-483.

[46] SCHAAL S. Is imitation learning the route to humanoid robots? [J]. Trends in Cognitive Sciences，1999，3(6)：233-242.

[47] ABBEEL P，NG A Y. Exploration and apprenticeship learning in reinforcement learning[C] // Proceedings of the 22nd International Conference on Machine Learning. 2005：1-8.

[48] MUNOS R. Error bounds for approximate policy iteration[C]// ICML. 2003：560-567.

[49] THIERY C，SCHERRER B. Least-squares λ policy iteration：Bias-variance trade-off in control problems[C] // International Conference on Machine Learning. 2010：2058-2072.

[50] BERTSEKAS D P. Approximate policy iteration：A survey and some new methods[J]. Journal of Control Theory and Applications，2011，9(3)：310-335.

[51] KIM B，FARAHMAND A，PINEAU J，et al. Learning from Limited Demonstrations[C]// NIPS. 2013：2859-2867.

[52] PIOT B，GEIST M，PIETQUIN O. Boosted bellman residual minimization handling expert demonstrations[C]// Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin：Springer，2014：549-564.

[53] CHEMALI J，LAZARIC A. Direct policy iteration with demonstrations[C]// Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015：1045-1065.

[54] HESTER T，VECERIK M，PIETQUIN O，et al. Deep Q-learning from demonstrations[C]// Thirty-second AAAI Conference on Artificial Intelligence. 2018：746-752.

[55] VECERIK M，HESTER T，SCHOLZ J，et al. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards[J]. arXiv：1707. 08817，2017.

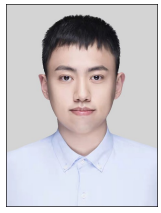
[56] NAIR A，MCGREW B，ANDRYCHOWICZ M，et al. Overcoming exploration in reinforcement learning with demonstrations [C]// 2018 IEEE International Conference on Robotics and Automation(ICRA). IEEE，2018：6292-6299.

[57] KANG B，JIE Z，FENG J. Policy optimization with demonstrations[C]// International Conference on Machine Learning. PMLR，2018：2469-2478.

[58] HO J，ERMON S. Generative Adversarial Imitation Learning [C]// NIPS. 2016：198-211.

[59] BURDA Y，EDWARDS H，STORKEY A，et al. Exploration by random network distillation[C]// Seventh International Conference on Learning Representations. 2019：1-17.

[60] LI Z, CHEN X H. Efficient Exploration by Novelty-Pursuit [C]// International Confe-rence on Distributed Artificial Intelligence. Cham: Springer, 2020: 85-102.
[61] NG A Y, HARADA D, RUSSELL S J. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping[C]// Proceedings of the Sixteenth International Conference on Machine Learning. 1999: 278-287.
[62] DEVLIN S M, KUDENKO D. Dynamic potential-based reward shaping[C]// Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. IFAAMAS, 2012: 433-440.
[63] LIU Y, HU Y, GAO Y, et al. Value Function Transfer for Deep Multi-Agent Reinforcement Learning Based on N-Step Returns [C]// IJCAI. 2019: 457-463.
[64] TIRINZONI A, RODRÍGUEZ-SÁNCHEZ R, RESTELLI M. Transfer of Value Functions via Variational Methods [C]// NeurIPS. 2018: 6182-6192.
[65] GE H, SONG Y, WU C, et al. Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control[J]. IEEE Access, 2019, 7: 40797-40809.
[66] RUSU A A, COLMENAREJO S G, GÜLÇEHRE Ç, et al. Policy Distillation[C]// ICLR(Poster). 2016.
[67] TEH Y W, BAPST V, CZARNECKI W M, et al. Distral: Robust multitask reinforcement learning[C]// NIPS. 2017.
[68] PARISOTTO E, BA L J, SALAKHUTDINOV R. Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning[C]// International Conference on Learning Representations. 2016: 23-28.
[69] YIN H, PAN S J. Knowledge transfer for deep reinforcement learning with hierarchical experience replay[C]// Thirty-First AAAI Conference on Artificial Intelligence. 2017: 68-82.
[70] ARNEKVIST I, KRAGIC D, STORK J A. Vpe: Variational policy embedding for transfer reinforcement learning[C]// 2019 International Conference on Robotics and Automation(ICRA). IEEE, 2019: 36-42.
[71] YANG J, PETERSEN B, ZHA H, et al. Single Episode Policy Transfer in Reinforcement Learning[C]// International Conference on Learning Representations. 2019: 1256-1268.
[72] DAYAN P. Improving generalization for temporal difference learning: The successor representation[J]. Neural Computation, 1993, 5(4): 613-624.
[73] BARRETO A, DABNEY W, MUNOS R, et al. Successor features for transfer in reinforcement learning[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 4058-4068.
[74] RUSU A A, RABINOWITZ N C, DESJARDINS G, et al. Progressive neural networks[J]. arXiv: 1606. 04671, 2016.
[75] ZHANG A, SATIJA H, PINEAU J. Decoupling dynamics and reward for transfer learning[J]. arXiv: 1804. 10689, 2018.
[76] BARRETO A, BORSA D, QUAN J, et al. Transfer in deep reinforcement learning using successor features and generalized policy improvement [C]// International Conference on Machine Learning. PMLR, 2018: 501-510.
[77] BARRETO A, HOU S, BORSA D, et al. Fast reinforcement learning with generalized policy updates[J]. Proceedings of the National Academy of Sciences, 2020, 117(48): 30079-30087.
[78] SCHAUL T, HORGAN D, GREGOR K, et al. Universal Value Function Approximators[C]// International Conference on Machine Learning. PMLR, 2015: 1312-1320.
[79] BORSA D, BARRETO A, QUAN J, et al. Universal Successor Features Approximators [C] // International Conference on Learning Representations. 2018: 359-369.
[80] BRAYLAN A E, MIIKKULAINEN R. Object-model transfer in the general video game domain[C]// Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference. 2016: 142-168.
[81] BARRETT S, STONE P. Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork[C]// Twenty-ninth AAAI Conference on Artificial Intelligence. 2015: 178-190.
[82] BARRETT S, STONE P, KRAUS S, et al. Teamwork with limited knowledge of teammates[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2013: 6984-6992.
[83] ROY S, MINCU D, LOREAUX E, et al. Multitask prediction of organ dysfunction in the intensive care unit using sequential subnetwork routing[J]. Journal of the American Medical Informatics Association, 2021, 28(9): 986-997.
[84] JOHNSON A E W, POLLARD T J, SHEN L, et al. MIMIC-III, a freely accessible critical care database [J]. Scientific Data, 2016, 3(1): 1-9.
[85] MCGRATH T, KAPISHNIKOV A, TOMAŠEV N, et al. Acquisition of Chess Knowledge in AlphaZero[J]. arXiv: 2111. 09259, 2021.
[86] VON RUEDEN L, MAYER S, BECKH K, et al. Informed Machine Learning-A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(1): 12-25.
[87] AMMANABROLU P, RIEDL M. Transfer in Deep Reinforcement Learning Using Knowledge Graphs[C]// Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing(TextGraphs-13). 2019: 1-10.
[88] HU Y, GAO Y, AN B. Accelerating multiagent reinforcement learning by equilibrium transfer[J]. IEEE Transactions on Cybernetics, 2014, 45(7): 1289-1302.



ZHANG Qiyang, born in 1998, postgraduate. His main research interests include deep reinforcement learning and knowledge transfer.

CHEN Xiliang, born in 1985, Ph. D, associate professor. His main research interests include command information system engineering and deep reinforcement learning.

(责任编辑:何杨)