华 中 科 技 大 学

研究生课程《先进制造技术》

学生姓名 郝海强

学生学号 D202180296

专业班级 机械工程2102

导 师 朱海平

日 期 2022.02.21

深度强化学习综述

摘要:深度强化学习(DRL)已成为人工智能研究的热点之一。它已被广泛应用于端到端控制、机器人控制、推荐系统、自然语言对话系统等各个领域。本综述系统地对深度RL算法及其应用进行了分类，并将现有的DRL算法分为基于模型的方法、无模型的方法和先进RL方法进行了详细的综述。深入分析了探索、逆RL和转移RL的研究进展。最后，概述了目前具有代表性的应用，并分析了四个有待进一步研究的问题。

关键词:强化学习;深度强化学习;强化学习应用

1引言

随着深度学习和大数据的结合，人工智能研究取得了革命性进展。人们对后深度学习领域的新技术探索越来越感兴趣。在传统的RL算法中使用神经网络建模的深度强化学习(Deep reinforcement learning, DRL)尤其具有吸引力。具体来说，DRL用于解决决策优化问题，并决定在面对特定状态时采取何种行动以实现利益最大化。因此，对深度学习理论的分析和应用受到了学术界和产业界的广泛关注。

深度RL是一种结合了深度学习和强化学习的一般范式，并在下棋、投资、驾驶、动作模仿等多种场景中取得了成功，被认为是最接近通用人工智能(artificial general intelligence，AGI)的智能之一。

深度RL与传统机器学习在处理和分析方面的差异是巨大的。目前主流的机器学习范式大多是事先收集或构造数据集标签，然后基于已有的静态数据进行机器学习。相比之下，RL是闭环学习范式的典型代表，它使用动态数据和标签将反馈信号带入学习过程。在这篇综述中，本文试图提供一个最新的深度RL算法的概述。

已经有DRL发表了几篇全面的综述。例如，Mousavi等人[1]总结了一些早期的RL算法，并将其分为6个核心元素、6个重要机制和12个应用，列举了RL资源的集合。Vanschoren[2]回顾了元学习算法，并根据元数据类型对其进行了分类。其对几种流行技术的性能特征进行了实验评估和分析，包括从任务属性学习、从模型评估学习和从先前模型学习。Huisman等人[3]研究和总结了关键算法，这些方法分为度量、模型和基于优化的技术，并确定了主要的开放挑战，例如对异构基准的性能评估，以及降低元学习的计算成本。

相对于这些研究，本文系统地、全面地回顾了不同的深度RL算法，而不是专注于一个特定的分支，展示了它们之间的关系和发展趋势。此外，本文分类了先进的RL方法及其应用，并提出了解决RL挑战的有前景的方向。综上所述，本文对RL进行了广泛的研究，贡献如下:

1. 对最新的RL方法进行了详细的回顾，并将它们分为三类:无模型方法、基于模型的方法和高级方法。
2. 系统地总结了高级RL算法的发展，并将其分为探索式RL、逆式RL和转移式RL三大类。
3. 详细地分析和讨论了RL在机器人技术、自然语言处理和计算机系统等方面的应用。
4. 提出了几个有待进一步研究的问题(如数据饥饿问题和AGI问题)，以及对每个问题的分析和进一步研究的方向。

2无模型强化学习

DRL的发展仍处于起步阶段。学术研究集中在确定性和静态环境中，状态主要是离散的和充分观察。因此，大多数RL工作都是基于无模型方法的。无模型RL可以通过大量样本估计agent的状态、价值函数和奖励函数，从而优化以在s状态下做a动作获得更多奖励为目标的行为策略。由于其实现简单，开放资源丰富，无模型RL吸引了越来越多的学者进行深入的研究。

在本节中，本文简单地将无模型RL划分为两个场景:(1)基于值函数的RL;(2)基于策略梯度的RL。

2.1基于值函数的RL

DRL的发展仍处于起步阶段。学术研究集中在确定性和静态环境中，状态主要是离散的和充分观察。因此，大多数RL工作都是基于无模型方法的。无模型RL可以通过大量样本估计agent的状态、价值函数和奖励函数，从而优化以在s状态下做a动作获得更多奖励为目标的行为策略πθ(a|s)。由于其实现简单，开放资源丰富，无模型RL吸引了越来越多的学者进行深入的研究。本节将无模型RL划分为两个场景:(1)基于值函数的RL;(2)基于政策策略的RL。

2.1.1深度Q-网络

深度Q-网络 (DQN)[4]是DRL的典型代表，它使用卷积神经网络(CNN)[5]作为模型，并使用Q-learning[6]的变体进行训练。DQN使用最大Q值作为低维动作输出，解决高维状态输入(如游戏画面的原始像素)的令人困惑的表示。

此外，DQN将奖励值和误差项减小到一个有限的区间，从而减轻了非线性网络所代表的值函数的不稳定性。与Q-learning算法不同的是，DQN将学习过程与训练过程同步，主要改进如下:(1)使用经验重放缓冲区[7]来减少样本之间的关联;(2)深度神经网络——目标网络——用于行为-价值函数逼近[8]。

实验结果表明，在雅达利2600的大多数游戏中，DQN的表现超过了之前的算法[9]，与人类专业测试人员的表现相当。在解决各种类型的基于视觉感知的深度RL任务时，DQN使用了相同的一组网络模型、参数设置和训练算法，这表明该方法具有很强的适应性和通用性。

2.1.2 DQN的发展

随着DQN算法的成功，提出了大量的改进算法。在这一小节中，本文主要关注与系统的整体结构、训练样本的构建和神经网络的结构相关的代表性方法。

双深度Q-网络(Double deep Q-network, DDQN)通过自举行为的解耦选择和评估，降低了Q-learning过高估计偏差的风险[10]。因为经验转换从经验回放中统一均匀采集，DQN显然没有充分考虑每个样本的重要性。改进的经验重演机制DDQN[11]通过计算经验池中每个样本的优先级，增加有价值训练样本的概率来解决这一问题。

由于受到传统架构(如卷积网络、长时间内存(LSTMs)或自动编码器)的限制，DQN算法还存在其他一些缺点，如缺乏长期记忆能力。Swain等人[12]研究了在DQN中加入递归的效果，方法是用递归LSTM替换第一个后卷积全连接层。Shuai等人[13]基于决斗架构扩展了DQN，该架构通过分别表示状态值和动作优势，有助于在动作之间进行泛化。

由于前面提到的DQN的独立改进基于一个共享的框架，Rainbow[14]可以将它们合理地结合起来。实验结果表明，该组合在雅达利2600基准上提供了最先进的数据效率和最终性能。

根据对Rainbow的研究，确定优先级是agent性能的最重要因素。然而，使用一个actor来收集数据，效率很低。Ape-X[15]为深度RL在规模上利用分布式架构，并使用数百个参与者收集数据，并通过各种参与者获得具有不同优先级的大量重放缓冲区。实验结果表明，在较短的训练时间内，ApeX的性能几乎是Rainbow的两倍。揭示了分布式计算正在成为深度RL的重要组成部分之一。

2.2基于策略梯度的RL

强化是策略梯度(PG)算法的原型。与基于值的RL相比，基于策略的RL不仅避免了由于值函数误差引起的策略退化，而且更容易应用于连续动作空间问题。具体来说，基于值的方法，如Q-learning和SARSA，需要一步操作才能计算出最大值，这在连续空间或高维空间中是很难找到的。此外，基于值的方法可以学习隐式策略，而基于策略的RL方法可以学习随机策略。也就是说，在基于价值的方法中，通过政策改进得到的政策都是确定性的政策，会遇到一些在石头剪刀布等任务中无法解决的问题。基于策略的方法也有一些共同的缺点:(1)数据效率或样本利用率低;(2)方差较大，难以收敛。

本文从两个方面介绍了典型的改进:(1)基于演员评论(actor-critic)的改进框架;(2)基于信任域的改进方法。

2.2.1基于actor-critic的改进框架

克服基于策略的方法的常见缺点的一个经典而有效的方法是actor-critic算法[16]，它学习策略和状态值函数。状态值函数用于bootstrapping，即从随后的估计中更新状态，以减少方差和加速学习。然而，在连续和离散的动作空间中稳定地应用演员评论方法的简单而高效的设计一直是RL的一个长期障碍。通过扩展DQN和DPG，一种深度确定性策略梯度(DDPG)算法[17]可以使用基于相同超参数和网络结构的低维观测(如笛卡尔坐标和关节角)学习任务的竞争策略。此外，双延迟DDPG算法[18]被提出作为一种确定性算法，可以并行工作，比DDPG有很大的改进。

而DDPG的策略是确定性的，由于策略一般需要具有一定的随机性，因此尤其不适用于有噪声干扰的复杂环境。另一个问题是，许多常用的无模型RL算法，如信赖域策略优化(TRPO) [19]，近端策略优化(PPO) [20]，或A3C[21]，每个梯度步长都需要大量的新样本。由于这种严格的要求，学习一项有效的政策需要花费巨大的费用，而且随着任务的复杂性的增加，情况会变得更糟。软演员批评家(SAC)[22]是解决这些问题的有效途径。SAC将非策略更新与一个稳定的随机AC公式结合起来，并使用一个最大熵框架来增加标准的最大奖励RL目标。与off-policy和on-policy先验方法相比，基于最大熵的RL算法在性能和样本效率方面都有很大的改进，这是因为基于最大熵的RL算法有几个特殊的优点:

1. 更复杂的特定任务的初始化。利用最大熵学习的策略可以作为更复杂的特定任务的初始化，并学习解决任务的方法。
2. 更强的探索能力。很明显，最大熵使得在多模式奖励下更容易找到更好的模式。
3. 鲁棒性强，泛化性强。最大熵需要从不同的方式探索各种最佳可能性，使它更容易在面对干扰时进行调整。

2.2.2基于信任域的改进方法

策略梯度法由于采用神经网络作为非线性函数逼近器，在数据不稳定的情况下难以稳定更新策略。近年来，学术界在RL中引入了信任域方法，并在各种实验场景中实现了显著的性能改进。

基于保守策略的结论迭代[23], TRPO将总变异散度的最大值作为学习速率，并考虑总变异散度与Kullback-Leibler散度之间的关系，从而将混合策略扩展为一般随机策略。广义优势估计器[24]在贴现马尔可夫决策过程(MDP)中使用两个因素作为可调参数，进一步在参与者批评算法中实现偏差和方差之间的权衡。

TRPO的一个麻烦是与环境的大量交互作用，与TRPO通过约束限制政策更新不同，ACER保持一个滑动平均值，该平均值代表过去的政策，并保持政策更新偏离该平均值。类似地，actor-critic使用张量积因式分解信任域[25]利用信任域的张量积因子近似曲率优化参与者和批评者，而约束策略优化[26]使用约束MDP。

TRPO在复杂的计算或与某些架构(例如，噪声或参数共享)的兼容性上失败。PPO的实现要简单得多，因为它使用了一个带有修剪概率比的目标函数来形成基于一阶优化的悲观估计，但这可能会导致样本效率低下。分布式PPO被进一步提出，以探索丰富的环境如何有助于促进复杂行为的学习。在A3C中，数据收集和梯度计算被部署到多个分布式工人。这不仅提高了性能和可伸缩性，而且还在丰富多样的环境中支持稳定的行为。

结合策略上和策略外的优点，路径一致性学习(PCL)[27]在熵正则化条件下，基于软最大值时间值一致性与策略最优性之间的关系，增强了搜索能力。信任-PCL[28]对PCL进行了改进，在约束优化中增加了折扣相对熵信赖域。这样既保证了优化的稳定性，又充分利用了off-policy数据，提高了样本效率。

3基于模型的强化学习

事实上，知道转移动力学会使问题解决变得更容易，这种动力学被称为模型。基于模型的方法是学习转换动力学的算法，它决定了在当前状态st执行动作at下一个状态将会是st+1在当前状态，接着方法将找出如何选择动作。简而言之，这种算法学习系统动力学模型，并使用最优控制选择动作。基于模型的RL是从最优控制领域发展而来的。通常，具体问题通过模型如高斯过程和贝叶斯网络建立，然后通过机器学习方法或最优控制方法解决，如模型预测控制(MPC)，线性二次调节器(LQR)，线性二次高斯控制。

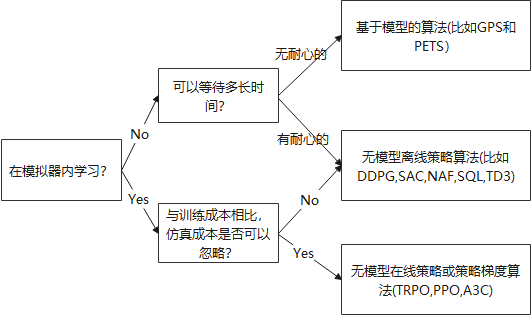


图1. 基于模型和无模型方法的不同使用场景

与无模型的RL相比，基于模型的RL以数据高效的方式学习价值函数或策略，不需要与环境进行持续的交互。但是，它可能会遇到模型识别的问题，导致对真实环境的描述不准确。在本节中，本文将基于模型的RL的上下文划分为三个场景，并系统地分析了它们的优缺点。

表1 不同强化学习算法的比较

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Method | Reference | Number of steps | Number of episodes | Time |
| 无模型 | 完全在线(如A3C) | Wang ZY等人[29] | 108 | 105 | ~15天 |
| 策略梯度(如TROPO) | Schulman等人[20] | 107 | 104 | ~1.5天 |
| 值估计(如DDPG和SAG) | Gu SX等人[30] | 106 | 103 | ~3小时 |
| 基于模型 | 如PETS和GPS | Chua等人[31] | 30000 | 30 | ~5min |

表1显示了需要不同样本数量和不同计算时间的不同算法。这是一个复杂的问题。这张表是一个粗略的指南，但本文认为它可以成为一个起点。本文根据标准基准任务“half-cheetah”列出了这些算法的数据，包括学习的步数、片段数和总时间。图1展示了基于模型和无模型方法的不同使用场景。如果本文有一个模拟器，并且模拟成本与培训成本相比可以忽略不计(在模型上采取一个模拟步骤要比一个梯度步骤便宜得多)，那么使用在线策略或策略梯度算法(比如TRPO, PPO, and A3C)是一个非常好的选择，因为这些算法更容易调整和有更少的超参数，然后更可靠地收敛。如果本文有一个模拟器，但模拟是昂贵的，那么本文可能会选择无模型的非策略算法，如DDPG、NAF和SQL。对于所有无策略模型的算法，都有许多不同的选择，这些算法可以是基于Q学习的，也可以是基于额外严格的。最后，如果本文没有模拟器，主要问题是可接受的等待时间是多长，若不是很长(不耐烦)，那么基于模型的算法，如引导策略搜索(GPS)和带轨迹采样的概率集成(PETS)是很好的选择。否则，您可以选择无模型的非策略算法，这种算法需要较少的假设，并且可能不那么特定于领域。

3.1全局模型和局部模型

对于基于模型的算法，第一个问题是，如果动力学未知，应该拟合哪一个:全局动力学模型还是局部动力学模型?在这一部分，本文介绍了相关的算法和这两种模型的比较。

尽管全局模型方法具有运行时计算成本低等优点，但在数值稳定性方面常常失效，尤其在随机域内，因为它们使用MPC迭代收集数据，并直接将它们反向传播到策略中。

为克服这一缺点已作了一些努力。Nagabandi等人[32]在特殊的深度神经网络动力学模型的帮助下初始化了一个无模型学习器，该模型可以将基于模型的方法的样本效率与无模型方法的高任务特异性性能结合起来。在MuJoCo运动任务上的结果表明，该算法具有良好的样本效率，能够在高速基准任务上加速无模型学习。

在大多数状态空间中，全局模型的规划者可能会寻找模型错误乐观的区域，因此需要一个非常精确的模型来收敛于一个好的解。此外，获得一个适当的模型往往比学习一项政策要困难得多，特别是在环境难以描述的情况下。因此，具有约束的局部模型越来越受到重视。局部模型需要弄清楚执行哪个控制器来获得正确的数据，以及如何确保整个模型不会出现严重的偏离。这里本文列出了一些典型的算法。

基于模型的迭代线性二次调节器(iLQR)[32]被扩展到GPS上，可以学习一系列具有高度一般性策略表示的动态操作行为，而无需使用已知的模型或示例演示。结合无模型RL的优点，Dyna-Q[16]是一个经典的集成架构，其中利用模型来更新q值。基于粒子的策略搜索的概率推理进一步允许几乎任意的模型和策略，通过同时匹配以前的数据高效学习算法的性能。为了借鉴无模型RL算法和基于模型的RL算法的优点，Chebotar等人[33]专注于时变线性高斯策略，并将基于模型的LQR算法与基于模型的路径积分策略改进算法相结合。

3.2 不确定性意识(Uncertainty-aware)模型

基于纯基于模型的方法和无模型的方法之间存在性能差距。与无模型方法需要10天的时间相比，基于模型的方法只需要10分钟就可以完成整个训练过程。然而，无模型方法可以获得更好的性能，最多相差三个数量级[32]。主要原因是过度拟合。不确定性感知模型是解决这一问题的有效方法。根据不确定性的表现形式，本文将不确定性分为两类:随机不确定性(固有的系统随机性)和认知不确定性(由于数据有限而产生的主观不确定性)。主要有两个方向:模式不确定性估计和利用输出熵建立不确定性感知模型。本文将在本节介绍前者，在4.1.1节介绍后者。

通过学习概率动态模型和明确地将模型不确定性纳入长期计划，学习控制的概率推理(PILCO)可以处理很少的数据，并在几次试验中促进从零开始的学习。Blundell等人[34]使用变分贝叶斯学习来估计神经网络中的不确定性(图2)。与前馈神经网络相比，该方法不仅缓解了过拟合问题，而且正确评估了训练数据中的不确定性。类似地， PETS[31]将不确定性感知的深度网络动力学模型与基于采样的不确定性传播结合在一起，最终将这些组件整合到一个深度RL框架中，从而达到无模型RL方法在基准控制任务上的渐近性能。

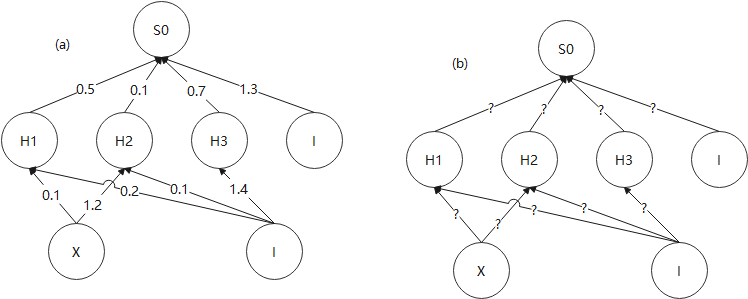


图2神经网络中估计不确定性的过程:(a)每个权值都有一个由经典反向传播提供的固定值;(b)每个权重的分配由贝叶斯方法提供

在RL中，将无模型和基于模型的方法结合起来，以实现高性能但低样本复杂度的算法，这是一个日益增长的兴趣。基于模型的价值扩展[35]允许想象固定深度来控制模型的不确定性。随机集合值展开[36]通过在不同视界长度的模型展开之间进行动态插值，确保模型只有在这样做时才会引入显著误差。

3.3复杂观测模型

基于模型的RL已被证明是一种学习控制任务的有效方法，但它难以用于具有复杂观测(如图像)的部分可观测MDPs。这是因为agent必须根据观察而不是准确的环境状态做出决定。

空间自动编码器架构提出了在潜在空间学习和自主学习图像的低维嵌入。然而，有一个不可避免的问题，即自动编码器可能无法恢复正确的表示，并且不适合基于模型的RL。为了解决这一难题，embed To control (E2C)在潜在空间中应用带有iLQR的变分自编码器，将高维非线性系统的局部最优控制问题转化为低维潜在状态空间。

将深度动作条件视频预测模型与模型预测控制相结合的方法[37]是第一个在观察空间中直接学习的机器人操作实例。它用完全未标记的数据和行动计划来训练agent，将环境中用户指定的对象移动到用户定义的位置，这两种方法都可以帮助生成新的、以前未见过的对象。视频预测模型(Ebert等人，2017)也可以通过合并时间跳跃连接来跟踪对象，并完全通过自监督机器人学习在技能范围和复杂性方面取得显著进步。因为准确的正向预测可能非常昂贵[31,32]， SOLAP[38]使用简单模型，通常是线性模型，为地方政策的改进提供梯度方向，而不是前向预测和规划。

4高级强化学习

4.1探索

在许多复杂的RL任务中，agent在与未知动态交互时面临着平衡探索和开发的挑战。随着RL的快速发展，各种有效的、可扩展的方法被提出，以克服缺乏探索的缺点。本文将介绍每一类的典型算法，并分析它们的优缺点。

4.4.1积极的探索

因为未经探索的行动可以带来更好的奖励，所以优化它们以有效地增加探索是很有价值的。因此，如何量化这些未被访问状态的状态新颖性是解决这类复杂问题的关键。

基于频率的分配奖励机制得到了广泛的应用。它们将行为(状态行为)的出现频率作为奖励:



其中N(s)代表状态s出现的频率。B(N(s))代表随着N(s)的增加而减少的奖励。积极探索使用r+(s, a)而不是r(s, a)作为任何无模型算法的奖励。不同的奖金具有相同的基本特征，即倾向于选择最优或最新的行动。除了置信度上限，AlphaGo在MCTS中作为奖金使用，还有一些其他的奖金，如基于模型的区间估计与勘探奖金(Strehl和Littman, 2008)和贝叶斯勘探奖金(Kolter和Ng, 2009)。

其中N(s)代表状态s出现的频率。B(N(s))代表随着N(s)的增加而减少的奖励。乐观探索使用r+(s, a)而不是r(s, a)作为任何无模型算法的奖励。不同的奖金具有相同的基本特征，即倾向于选择最优或最新的行动。除了置信度上限，AlphaGo在MCTS中作为奖金使用，还有一些其他的奖金，如基于模型的区间估计与勘探奖金(Strehl和Littman, 2008)和贝叶斯勘探奖金(Kolter和Ng, 2009)。

在一些复杂的高维连续环境中(如《星际争霸2》)，几乎不可能出现两次完全相同的状态。这将导致上述方法变得毫无意义，因为几乎每个状态都是1。基于密度模型，引入了伪计数的概念，以推广基于计数的非表列RL agent勘探奖金(Bellemare et al.， 2016)。从那时起，人们对伪计数算法进行了大量的改进。Ostrovski等人(2017)通过基于计数的神经密度模型探索改进了工作。基于哈希的计数(Tang等人，2017)将状态映射到哈希码，并通过哈希表计算状态出现的次数。通过范例模型探索(Fu等人，2017a)，分类器被训练来区分每个访问过的状态与所有其他状态，新状态与旧状态进行隐式比较。

4.1.2后验抽样探索

用随机行为(例如，epsilon-greedy)进行探索会遭遇来回振荡的情况。它可能不会进入一个连贯或有趣的地方，而使用随机q函数探索可能会在整个事件中使用随机但内部一致的策略。受此启发，提出了后验采样算法，以实现更有针对性的探索。

Thompson抽样方法从先验分布中对每个行动的q值进行抽样。然后，将每个行动的奖励概率视为Beta分布，为调整后验提供指导，实现更小的后悔。自展(bootstrapped)DQN[34]将深度勘探与深度神经网络相结合，克服了与非线性参数化值函数不兼容的缺点。图3显示了bootstrap DQN的体系结构。

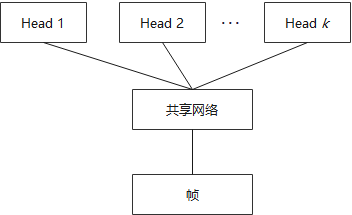


图3引导DQN的体系结构

head表示Q-function。共享网络从所有数据中学习联合特征表示。帧是一个存储样本的重放缓冲区

自展DQN在多个并联网络中训练自展数据以随机化值函数，这种训练方式有助于从噪音中学习和探索复杂的状态/动作空间之间的平衡。通过这种分布式深度探索，自展DQN能够充分保证各种策略的探索，产生多样化的样本，并且能够更好地推广到未知状态空间。

4.1.3信息获取探索

在一些复杂的环境中，对一个动作进行采样可以帮助agent评估其他动作。在此基础上，在信息获取探索中充分考虑了信息结构的优势，有助于agent在困难任务中更高效地学习。

信息导向抽样(Russo and Roy, 2014)衡量真正的最优行为与下一个观察之间的互信息，然后选择一个行为来量化学习的数量。对每个选择的行为进行采样，使期望单周期后悔与信息增益之间的比率最小化，这也在一定程度上平衡了勘探和开发。

然而，本文不可能构建一个成形的奖励函数，因为在现实世界的许多场景中，环境中的奖励极其稀少或完全缺失。与上述探索方法相比，利用贝叶斯神经网络中的变分推理，变分信息最大化探索(VIME)[39]逼近了动态模型的概率，大大降低了对奖励的依赖。内部探索机制(其中奖励函数被信息增益修正) 激励agent去探索未知领域并允许与其他方法(如TRPO)集成。

与VIME类似，好奇驱动学习(curiosity-driven )[40,41]根据agent预测其自身行动的结果的难度(即，根据当前状态和执行的行动，预测下一个状态)产生内在奖励信号。正向动力学模型的预测误差被用来作为激励个体好奇心的内在奖励。实验表明，即使在奖励很少的环境中，好奇心驱动学习也能在各种连续控制任务中取得显著的表现。

4.2逆RL

通常本文学习过渡模型时，假设奖励模型已经为人所知。然而，一旦奖励功能设计不当，就会对绩效产生巨大影响。引入逆RL (IRL)[42]，从观察到的专家例子中学习适当的奖励函数。然而，IRL存在一些挑战:(1)问题定义不足，缺乏先验知识;(2)对学习奖励进行评估是困难的;(3)演示不一定是最优的。在下文中，本文讨论了基于最大边际和最大熵的解。

4.2.1基于最大margin的IRL

学徒学习[43]使用最大边际法从一个专家实例中找到当前的奖励函数。在奖励函数下得到的最优策略保证在专家实例策略附近。

此外，神经逆RL (NIRL)[44]主要关注具有大规模高维状态空间的IRL。在神经网络的帮助下，NIRL不仅可以将专家的行为推广到状态空间的未访问区域，而且可以很容易地表达出明确的策略表示，即使是对于随机的专家策略。

总之，基于最大边际的方法往往是模糊的。例如，许多不同的奖励功能会导致相同的专家政策。在这种情况下，学习到的奖励函数通常具有随机偏好。主要挑战如下:(1)边际最大化具有一定的随意性;(2)没有明确的专家次优度模型;(3)深度规划不适用于复杂约束优化问题。

4.2.2基于最大熵的IRL

最大熵方法由于其概率分布没有对除约束外的任何位置信息的分布作任何假设，因此可以避免歧义问题。当在与所示行为的奖励值相匹配的约束下选择一个分配而不是决策时，它可能会引起歧义。最大熵IRL (Max Ent IRL)[6]是利用最大熵原理提出的解决问题的方法。

MaxEnt IRL虽然解决了歧义问题，但不能适应大而连续的状态和动作空间，也不能满足未知动态下有效学习的要求。基于最大熵的方法在实际应用中也很难应用，因为:(1)奖励函数的学习需要人工选择特征(对于许多实际问题来说，特征的选择是非常困难的);(2) IRL的许多子循环都包含正向RL，这是一个比较困难的问题。为了解决前一个挑战，Ganesh等人[6]利用神经网络的表征能力来近似复杂的非线性奖励函数。为了解决后一个挑战，引导成本学习[37]进一步制定了一种基于样本的有效近似，而不是前向RL。

生成式对抗模仿学习(GAIL)[45]映射了生成对抗网络(GAN)情境下IRL中的奖励函数目标。

GAIL的策略模型就像GAN中的生产模型，以状态为输入生成动作。GAIL的奖励函数模型可以作为一个判别模型来判别行为在多大程度上近似专家行为。实验结果表明，GAIL能够在模拟大型、高维环境中的复杂行为方面取得显著的效果。对抗式反强化学习(AIRL)[46]提供了价值函数和奖励函数的同时学习。与GAIL相比，AIRL充分利用了高效的对抗性配方，也恢复了可推广的、可携带的奖励功能。

4.3转移RL

人工智能的一个基本问题是，它不能像人类那样高效地学习。许多RL算法展示了超人的性能，但需要数百万个训练样本。许多迁移RL算法都是基于这样一种直觉提出的，即有用的知识可能会从之前的任务中获得，以解决新的任务。这种算法的目标是在新环境中使用少量的可用数据快速学习最优策略。

根据源域的选择，将迁移RL分为三种类型:(1)前向迁移，即在一个任务上进行训练，然后迁移到新的任务上;(2)多任务转移，即多任务训练，转移到新任务上;(3)学习从多个任务中学习的元RL。

4.3.1正向转移

前向转移最简单的方法就是抱着最大的希望去尝试[24]。针对某一种情况训练的策略可能会奏效，如果运气好的话，可以成功地处理新任务，因为有时在训练中有足够的可变性来概括。

微调优仍然是神经网络迁移学习最流行的选择之一。首先，在数据通常丰富的源域上对模型进行预训练。其次，模型的输出层适应于目标域。最后，通过反向传播对网络进行微调。例如，在多样性前培训的方法中[22]，它学习给定任务的所有可能的方法，以实现特定的策略。这个策略可以作为一个很好的初始化，用于微调到更具体的行为(即，首先学习机器人可以向前移动的所有方式，然后使用这个初始化来学习单独的奔跑和跳跃技能)。

不幸的是，当在大型深度网络中使用少量经验进行微调时，微调方法往往会导致过度拟合。渐进式神经网络[47]通过构建来缓解过拟合并解决表达性问题。这个框架在整个训练过程中保留了一个预训练模型库，并从这些模型中学习横向连接，从而为新任务提取有用的特征。图4显示了渐进式神经网络的体系结构。标记为“a”的灰框表示适配器，它使前排的隐藏层激活值与原始输入尺寸保持一致。

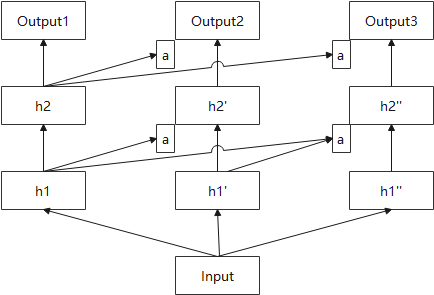


图4一个简单的渐进式神经网络

事实上，如果特工们在训练中看到更多的多样性，他们就会更好地调动。为了达到更好的性能，这些方法通过修改源域和使用适量的目标域数据进行传输，使网络无法区分源域和目标域的观测值。集成策略优化算法[48]利用来自目标域的数据和随机化模拟源域的物理参数，可以适应集成中源域的概率。该方法可以帮助学习更健壮的策略，并将其推广到更广泛的可能目标域。对于基于感知的任务，使用循环神经网络(RNN)在真实环境中输出未知参数(如dictionary/mass)来预先计算机器人在现实世界中行动时可能遇到的许多情况，减少与现实世界的交互。

4.3.2多任务转移

RL的典型应用更注重掌握而不是一次性学习，需要大量的训练集。多任务转移提供了一种解决这些挑战的方法，并且更接近于人们所做的——建立一生的经验。

最简单的解决方案之一是学习一个可以同时执行许多任务的模型。在线动态适应[46]将先前任务的先验知识与动态模型的在线适应相结合。这种基于模型的方法不仅减少了使用模型对训练数据的需求，而且还利用机器人在其他任务上的经验一次性学习新的任务，而不需要设计师提供明确的领域知识或演示。

然而，有时候学习一个模型是非常困难的。在对每个MDP分别进行培训后，提出了无模型方法来组合这些策略[49,50]。这种算法利用DQN和模型压缩技术来训练一个单一的政策网络，并在多位专家教师的指导下学习如何在一组不同的任务中行动。

如果具有可重用组件的体系结构具有共享部分和不同部分，则可以将其设计为充分利用任务的特征。由于上述各种方法为所有任务构建了一个单一的网络，因此模块化策略(Devin et al.， 2017)可以将神经网络策略分解为任务特定模块和机器人特定模块，从而获得更好的性能。该算法对所有机器人训练相同的特定于任务的组件，对所有任务训练相同的特定于机器人的组件。然后，机器人和特定任务的模块将被混合和匹配，以执行新的任务。这种特殊的架构允许agent在机器人之间共享任务信息(如感知)，并在任务之间共享机器人信息(如动力学和运动学)。

在用task1和task2训练robot1和用task1训练robot2之后，robot2可以在混合使用机器人和任务特定组件进行测试时执行新的task2

4.3.3 元-RL

Meta-RL提供了一个可行的框架，以解决在非常复杂的环境中需要战略和战术思考的少镜头学习。设Mi = (S, A, Ti, Ri)表示具有状态空间S、行为空间A、转移概率分布Ti和奖励函数Ri的MDP。在元-RL中，对于一个族的MDPs其包括一个任务分布，从而每个MDP都具有统计规律性。Meta-RL旨在通过学习策略πθ、模型Tθ和奖励函数等MDPs簇中的元项来提高后续任务的学习效率。本文介绍了两种最具代表性的元RL算法，并列举了一些主要的改进。

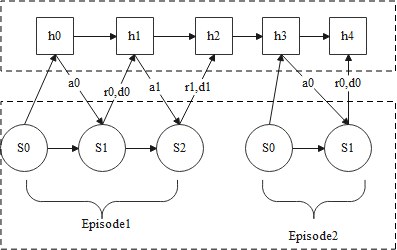


图5媒介与环境之间相互作用的过程

对于每个MDP中的每个事件，从特定于相应MDP的初始状态分布中提取一个新的S0。探员是按照历史轨迹训练的。在一集的结尾，政策的隐藏状态会保留到下一集，但不会在审判之间保留

Duan等人[51]定义了一个新的概念，深度元-RL，并提出了称为RL2的递归模型，该模型被编码在RNN的权值中并通过通用RL算法慢慢学习。RL2的主要思想是用所有历史轨迹(状态、行动和奖励)训练神经网络，并自动确定任务级别(元级别)的信息。这种设计可以显著加快新任务的训练过程，被认为是RL向元RL最基本的改变之一。图5展示了agent与环境的交互过程。进一步，提出了一个简单的神经专注学习者(SNAIL)[52]来解决上述原始模型中的信息丢失问题。为了从经验中收集信息，并确定信息中的具体细节，SNAIL利用LSTM调整RNN的结构，并采用交叉一维卷积和软注意的新组合。E-RL2 (Stadie等人，2018)为贴现收益的标准总和增加了一个指标，如果事件有助于解释[53]这一初始抽样分布的影响，则返回1，否则返回0。实验表明，这鼓励RNN考虑更广泛的抽样分布对最终元奖励的影响。ReBAL[54]使用循环模型来学习过渡模型，而不是行动策略，以在模拟的连续控制任务上表现良好。

循环模型在架构中有多种设计选择，通常是通用的和具有表现力的。然而，它们在处理复杂的任务时往往失败，需要复杂的模型和不切实际的数据。另一种典型的元学习方法是模型不可知论元学习(mml)[55]。为了克服循环模型的缺点，MAML使用了一个固定的优化器，并学习了一组基本参数，这些参数可以调整为通过几个梯度下降步骤来最小化任何任务损失。

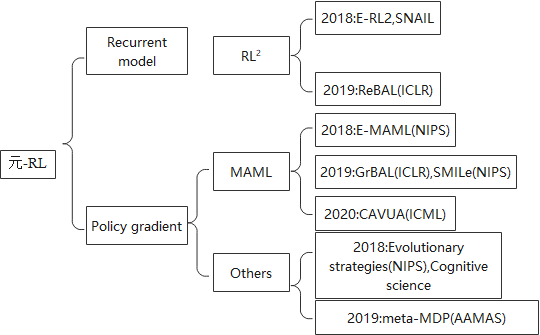


图6深度元RL算法之间的关系

由于有固定的优化器，MAML可以直接应用于任何用梯度下降程序训练的学习问题和模型。由于MAML简单而强大，它引起了研究者们的极大兴趣。改进探索是一个有吸引力的前景。除了基于MAML框架的改进之外，还有其他一些新的方法。元MDP探索方法[56]将寻找最佳探索政策的问题作为一个MDP模型，并将政策分为探索政策和开发政策，以实现终身学习。

综上所述，深度元RL是一个很有前途的研究方向，目前正利用多种元角度和多种探索方法来实现更好的元训练和适应效率。图6给出了改进的深度元RL算法之间的关系表示。

5应用

在RL开发的早期阶段，DQN主要用于各种二维视频游戏，如雅达利游戏[14,15]。随着AlphaGo利用深度神经网络和MCTS技术成功击败国际象棋世界冠军，近年来人们对RL在不同领域提高处理复杂任务的性能越来越感兴趣。

在游戏领域，《星际争霸2》[57]提供了一个具有挑战性的多agent环境，多个玩家通过交互来探索RL算法和架构。深度元RL也可以在游戏中用于进行少量的学习，例如通过从单个记录的动作序列[58]或在线视频[59]模仿人类Atari游戏玩法。

Deep RL在各个领域都取得了显著的成功。下面本文将概述当前具有代表性的RL应用，包括机器人技术、自然语言处理(NLP)和计算机系统。

5.1机器人

机器人技术是RL的一个经典领域。RL可以实现复杂机器人在仿真环境中的行为控制，从而实现对扰动和环境变化的真实响应。除了雅达利和Mujoco中的简单agent(如半猎豹、蚂蚁和蜘蛛)，DeepMimic进一步开发具有挑战性的多技能agent，包括多个角色(如人类、阿特拉斯机器人、双足恐龙和龙)和多种技能(如移动、杂技和武术)。

此外，RL在实际情况下的机器人控制任务方面也取得了大量的研究成果。现实世界中的一系列任务都是接触丰富的，需要视觉和控制之间的密切协调，比如堆叠紧密的乐高积木和把瓶盖拧到瓶子上。然而，在真实场景下，真实机器人的训练数据非常缺乏。该方法结合了以前任务的知识和动态模型的在线自适应[46]，有助于一次性解决各种复杂的机器人操作任务。多机器人[30,60]协作学习，以并行采样和训练。

基于迁移学习[61]，机器人可以跨机器人共享与任务相关的模块，也可以跨所有任务共享与机器人相关的模块。meta-RL的改进方法[62]使agent能够从新环境中的少量数据中快速学习。

5.2自然语言处理

RL方法在自然语言处理领域具有广阔的应用前景，已成功地应用于神经机器翻译(NMT)、对话系统和语音生成等领域。

NMT系统通常依赖于对齐的并行训练语料库，但这种并行数据在实践中收集的成本很高。这一矛盾可能会造成规模上的严重限制，制约相关研究和应用。机器翻译的双学习通过RL(如PG)从无标记数据中训练翻译模型，解决了NMT中的数据匮乏问题。利用该机制，单语数据可以发挥与并行双语数据相似的作用，显著降低了训练过程中对并行双语数据的要求。

5.3计算机系统

计算机系统为RL提出了许多具有挑战性的问题，包括时变的状态或动作空间(例如，在计算机集群中动态变化的任务和机器数量)，结构化的数据源(例如，表示任务数据流的图形或网络拓扑)，以及高度随机的环境(例如，随机时变负载)。在这里，本文总结了计算机系统中使用的一些典型的RL方法，并表明RL在这个领域可以提供显著的现实世界的好处。

6挑战与未来

RL是最接近AGI的智能之一。但问题仍然很多，其中很多是根本性的困难:

6.1挑战

6.1.1. 低效的样本

如表1所示，为了使模型达到一定的水平，需要大量的训练样本。大多数人在几分钟内就能上手的雅达利游戏需要花费大量时间。Rainbow DQN需要很长时间来训练模型，需要大约83小时的游戏体验才能超过人类的水平。基于模型的RL计算成本高，并且有自己的规划谬误:学习一个好的策略通常需要更多的样本，而实际的样本效率往往远低于预期的结果。

6.1.2. 苛刻的奖励功能

RL假设存在一个奖励函数，这个函数必须被精确地定义，以确保agent在任何时候都能做正确的事情。然而，奖励功能的设计通常是困难的:(1)需要一个合适的事先;(2)对所有对象状态都有完善的认识;对这个问题有一个好的定义是必要的。此外，有时在奖励功能上的过多努力可能会引入新的偏见。即使给了很好的奖励，也很难摆脱局部最优。前面提到的几种直观的探索方法(第4.1节)在一定程度上缓解了这个问题。然而，据本文所知，没有一种方法能够在所有环境中一致地工作。此外，IRL和模仿学习不需要奖励函数，但它们的表现通常是有限的和不令人满意的。

6.1.3.过度拟合和不稳定

RL的agent很少能够适应多种环境。即使是深度的元RL，也不能保证agent在新任务上表现良好。RL是不稳定的，对训练过程的初始化和动态非常敏感。在这方面，每一类算法都有自己的挑战。例如，具有深度网络函数估计的拟合Q(或值)方法通常是不收缩的，因此不能保证收敛。此外，还有许多稳定性参数，如目标网络延迟、重放缓冲区大小和剪辑。政策梯度方法有一个非常高的方差梯度估计。虽然有很多方法试图减少方差，但大多数也引入了超参数，因此他们需要很多样本和复杂的基线。基于模型的RL算法必须选择其模型类和拟合方法。优化策略是不平凡的，因为反向传播通过时间。相比之下，监督学习中不同的超参数会在训练中出现或多或少的变化。在RL中，运气不好可能意味着模型的曲线在很长一段时间内不会改变，或者RL方法根本不起作用。更糟糕的是，即使所有的超参数和随机种子都是已知的，只要实现稍有不同，性能就会非常不同。

6.2未来方向

本文相信RL有很大的前景，并列出了一些未来可能的方向:

1. 需要具有良好改进和收敛性的算法。尽管TRPO在假设下提供了一种有保证的改进，但它在现实世界中可能并不适用。虽然自适应调整参数(Gu SX et al.， 2017b)需要大量的样本来提供统计保证，但它可能是一个很好的起点。
2. 人为地添加一些监督信号。在稀疏奖励的情况下，本文可以引入内在奖励或添加一些辅助任务来增加探索能力。
3. IRL可以自动学习奖励函数，而模仿学习对奖励函数的要求不高。此外，对于多样化行为的无监督或弱监督学习会避免像奖励功能那样的严格监督。这两个方向可以弥补RL的不足。
4. 从多任务学习中归纳。深度元RL越来越被认为是实现AGI的最有可能的方法之一。近年来的研究表明，可以通过提高meta-RL的探索能力来提高agent的性能。meta-RL与IRL的结合也是一个很有吸引力的方向。

7总结

在过去的几年中，深度RL在处理复杂问题方面变得越来越强大和重要。在本综述中，本文对深度RL算法进行了全面的综述。首先，本文引入了无模型和基于模型的深度RL算法来确定深度RL的特征。在此基础上，总结了近年来的研究进展，并将其分为三大类:探索方法、逆学习方法和迁移学习方法。本文对每个类别进行了详细的回顾。在应用方面，本文讨论了深度RL在机器人、自然语言处理和计算机系统中的应用。最后，本文提出了一些有待解决的问题，并指出了未来的研究方向。

参考文献：

[1] Mousavi S S, Schukat M, Howley E. Deep Reinforcement Learning: An Overview[C].SAI Annual Conference on Areas of Intelligent Systems and Artificial Intelligence and their Applications to the Real World (IntelliSys),2018: 426-440.

[2] Joaquin, Vanschoren.Meta-learning: a survey[J].Computer Science,2018.

[3] Huisman M, Van Rijn J N, Plaat A.A survey of deep meta-learning[J].Artificial Intelligence Review,2021, 54 (6): 4483-4541.

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

[5] Zhong J Y, Xiong X F.An Orderly EV Charging Scheduling Method Based on Deep Learning in Cloud-Edge Collaborative Environment[J].Advances in Civil Engineering,2021, 2021: 12.

[6] Ganesh A H, Xu B.A review of reinforcement learning based energy management systems for electrified powertrains: Progress, challenge, and potential solution[J].Renewable & Sustainable Energy Reviews,2022, 154: 22.

[7] Shi J L, Zhao L, Wang X W, et al.A Novel Deep Q-Learning-Based Air-Assisted Vehicular Caching Scheme for Safe Autonomous Driving[J].Ieee Transactions on Intelligent Transportation Systems,2021, 22 (7): 4348-4358.

[8] Meng F Q, Tian K S, Wu C F.Deep Reinforcement Learning-Based Radar Network Target Assignment[J].Ieee Sensors Journal,2021, 21 (14): 16315-16327.

[9] Bellemare M G, Naddaf Y, Veness J, et al.The Arcade Learning Environment: An Evaluation Platform for General Agents[J].Journal of Artificial Intelligence Research,2013, 47: 253-279.

[10] Van Hasselt H, Guez A, Silver D, et al. Deep Reinforcement Learning with Double Q-Learning[C].30th Association-for-the-Advancement-of-Artificial-Intelligence (AAAI) Conference on Artificial Intelligence,2016: 2094-2100.

[11] Liu X M, Zhu T Q, Jiang C Q, et al.Prioritized Experience Replay based on Multi-armed Bandit[J].Expert Systems with Applications,2022, 189.

[12] Swain P, Kamalia U, Bhandarkar R, et al. CoDRL: Intelligent Packet Routing in SDN Using Convolutional Deep Reinforcement Learning[C].13th IEEE International Conference on Advanced Networks and Telecommunication Systems (IEEE ANTS),2019.

[13] Shuai H, Li F X, Pulgar-Painemal H, et al.Branching Dueling Q-Network-Based Online Scheduling of a Microgrid With Distributed Energy Storage Systems[J].Ieee Transactions on Smart Grid,2021, 12 (6): 5479-5482.

[14] Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining Improvements in Deep Reinforcement Learning[C].32nd AAAI Conference on Artificial Intelligence / 30th Innovative Applications of Artificial Intelligence Conference / 8th AAAI Symposium on Educational Advances in Artificial Intelligence,2018: 3215-3222.

[15] Horgan D, Quan J, Budden D, et al.Distributed prioritized experience replay[J].arXiv preprint arXiv:1803.00933,2018.

[16] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press,2018.

[17] Lilicrap T, Hunt J, Pritzel A, et al.Continuous control with deep reinforcement learning[J].arXiv,2016.

[18] Fujimoto S, Van Hoof H, Meger D. Addressing Function Approximation Error in Actor-Critic Methods[C].35th International Conference on Machine Learning (ICML),2018.

[19] Schulman J, Levine S, Moritz P, et al. Trust Region Policy Optimization[C].32nd International Conference on Machine Learning,2015: 1889-1897.

[20] Schulman J, Wolski F, Dhariwal P, et al.Proximal policy optimization algorithms[J].arXiv preprint arXiv:1707.06347,2017.

[21] Pan Y, Wang W Y, Li Y B, et al.Research on Cooperation Between Wind Farm and Electric Vehicle Aggregator Based on A3C Algorithm[J].Ieee Access,2021, 9: 55155-55164.

[22] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C].International conference on machine learning,2018: 1861-1870.

[23] Kakade S, Langford J. Approximately optimal approximate reinforcement learning[C].In Proc. 19th International Conference on Machine Learning,2002.

[24] Schulman J, Moritz P, Levine S, et al.High-dimensional continuous control using generalized advantage estimation[J].arXiv preprint arXiv:1506.02438,2016.

[25] Wu Y, Mansimov E, Grosse R B, et al.Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation[J].Advances in neural information processing systems,2017, 30.

[26] Achiam J, Held D, Tamar A, et al. Constrained policy optimization[C].International conference on machine learning,2017: 22-31.

[27] Nachum O, Norouzi M, Xu K, et al.Bridging the gap between value and policy based reinforcement learning[J].Advances in neural information processing systems,2017, 30.

[28] Nachum O, Norouzi M, Xu K, et al.Trust-pcl: An off-policy trust region method for continuous control[J].arXiv preprint arXiv:1707.01891,2017.

[29] Wang Z, Bapst V, Heess N, et al.Sample efficient actor-critic with experience replay[J].arXiv preprint arXiv:1611.01224,2016.

[30] Gu S, Lillicrap T, Sutskever I, et al. Continuous deep q-learning with model-based acceleration[C].International conference on machine learning,2017: 2829-2838.

[31] Chua K, Calandra R, Mcallister R, et al.Deep reinforcement learning in a handful of trials using probabilistic dynamics models[J].Advances in neural information processing systems,2018, 31.

[32] Nagabandi A, Kahn G, Fearing R S, et al. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning[C].2018 IEEE International Conference on Robotics and Automation (ICRA),2018: 7559-7566.

[33] Chebotar Y, Hausman K, Zhang M, et al. Combining model-based and model-free updates for trajectory-centric reinforcement learning[C].International conference on machine learning,2017: 703-711.

[34] Osband I, Blundell C, Pritzel A, et al.Deep exploration via bootstrapped DQN[J].Advances in neural information processing systems,2016, 29.

[35] Ko H, Pack S, Leung V C M.Mobility-Aware Vehicle-to-Grid Control Algorithm in Microgrids[J].Ieee Transactions on Intelligent Transportation Systems,2018, 19 (7): 2165-2174.

[36] Yang L L, Wei J T, Ma Z, et al.The Fabrication of Micro/Nano Structures by Laser Machining[J].Nanomaterials,2019, 9 (12): 69.

[37] Finn C, Levine S. Deep visual foresight for planning robot motion[C].2017 IEEE International Conference on Robotics and Automation (ICRA),2017: 2786-2793.

[38] Zhang J J, Zhang C, Chien W C.Overview of Deep Reinforcement Learning Improvements and Applications[J].Journal of Internet Technology,2021, 22 (2): 239-255.

[39] Tang H, Houthooft R, Foote D, et al.# exploration: A study of count-based exploration for deep reinforcement learning[J].Advances in neural information processing systems,2017, 30.

[40] Pathak D, Agrawal P, Efros A A, et al. Curiosity-driven exploration by self-supervised prediction[C].International conference on machine learning,2017: 2778-2787.

[41] Burda Y, Edwards H, Pathak D, et al.Large-scale study of curiosity-driven learning[J].arXiv preprint arXiv:1808.04355,2018.

[42] Ng A Y, Russell S J. Algorithms for inverse reinforcement learning[C].Icml,2000: 2.

[43] Abbeel P, Ng A Y. Apprenticeship learning via inverse reinforcement learning[C].Proceedings of the twenty-first international conference on Machine learning,2004: 1.

[44] Xia C, El Kamel A.Neural inverse reinforcement learning in autonomous navigation[J].Robotics and Autonomous Systems,2016, 84: 1-14.

[45] Ho J, Ermon S.Generative adversarial imitation learning[J].Advances in neural information processing systems,2016, 29.

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

[47] Rusu A A, Rabinowitz N C, Desjardins G, et al.Progressive neural networks[J].arXiv preprint arXiv:1606.04671,2016.

[48] Rajeswaran A, Ghotra S, Ravindran B, et al.Epopt: Learning robust neural network policies using model ensembles[J].arXiv preprint arXiv:1610.01283,2016.

[49] Parisotto E, Ba J L, Salakhutdinov R.Actor-mimic: Deep multitask and transfer reinforcement learning[J].arXiv preprint arXiv:1511.06342,2015.

[50] Rusu A A, Colmenarejo S G, Gulcehre C, et al.Policy distillation[J].arXiv preprint arXiv:1511.06295,2015.

[51] Duan Y, Schulman J, Chen X, et al.Rl $^ 2$: Fast reinforcement learning via slow reinforcement learning[J].arXiv preprint arXiv:1611.02779,2016.

[52] Mishra N, Srivastava S, Sharan S N, et al. RADDPG: Resource Allocation in Cognitive Radio with Deep Reinforcement Learning[C].International Conference on COMmunication Systems and NETworkS (COMSNETS),2021: 589-595.

[53] Stadie B C, Yang G, Houthooft R, et al.Some considerations on learning to explore via meta-reinforcement learning[J].arXiv preprint arXiv:1803.01118,2018.

[54] Nagabandi A, Clavera I, Liu S, et al.Learning to adapt in dynamic, real-world environments through meta-reinforcement learning[J].arXiv preprint arXiv:1803.11347,2018.

[55] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks[C].International conference on machine learning,2017: 1126-1135.

[56] Garcia F, Thomas P S.A meta-MDP approach to exploration for lifelong reinforcement learning[J].Advances in Neural Information Processing Systems,2019, 32.

[57] Vinyals O, Babuschkin I, Czarnecki W M, et al.Grandmaster level in StarCraft II using multi-agent reinforcement learning[J].Nature,2019, 575 (7782): 350-354.

[58] Pohlen T, Piot B, Hester T, et al.Observe and look further: Achieving consistent performance on atari[J].arXiv preprint arXiv:1805.11593,2018.

[59] Aytar Y, Pfaff T, Budden D, et al.Playing hard exploration games by watching youtube[J].Advances in neural information processing systems,2018, 31.

[60] Yahya A, Li A, Kalakrishnan M, et al. Collective robot reinforcement learning with distributed asynchronous guided policy search[C].2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),2017: 79-86.

[61] Devin C, Gupta A, Darrell T, et al. Learning modular neural network policies for multi-task and multi-robot transfer[C].2017 IEEE international conference on robotics and automation (ICRA),2017: 2169-2176.

[62] Finn C, Yu T, Zhang T, et al. One-shot visual imitation learning via meta-learning[C].Conference on robot learning,2017: 357-368.