深度强化学习算法与应用研究现状综述

刘朝阳[1]，穆朝絮[1]，孙长银[2]
（1. 天津大学电气自动化与信息工程学院，天津 300072；2. 东南大学自动化学院，江苏 南京 210096）

摘　要：深度强化学习主要被用来处理感知−决策问题，已经成为人工智能领域重要的研究分支。概述了基于值函数和策略梯度的两类深度强化学习算法，详细阐述了深度 Q 网络、深度策略梯度及相关改进算法的原理，并综述了深度强化学习在视频游戏、导航、多智能体协作以及推荐系统等领域的应用研究进展。最后，对深度强化学习的算法和应用进行展望，针对一些未来的研究方向和研究热点给出了建议。

关键词：人工智能；深度强化学习；值函数；策略梯度；导航；协作；复杂环境；泛化性；鲁棒性
中图分类号：TP181
文献标识码：A
doi: 10.11959/j.issn.2096−6652.202034

An overview on algorithms and applications of deep reinforcement learning

LIU Zhaoyang[1], MU Chaoxu[1], SUN Changyin[2]
1. School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
2. School of Automation, Southeast University, Nanjing 210096, China

Abstract: Deep reinforcement learning (DRL) is mainly applied to solve the perception-decision problem, and has become an important research branch in the field of artificial intelligence. Two kinds of DRL algorithms based on value function and policy gradient were summarized, including deep Q network, policy gradient as well as related developed algorithms. In addition, the applications of DRL in video games, navigation, multi-agent cooperation and recommendation field were intensively reviewed. Finally, a prospect for the future research of DRL was made, and some research suggestions were given.

Key words: artificial intelligence, deep reinforcement learning, value function, policy gradient, navigation, cooperation, complex environment, generalization, robustness

1　引言

近年来，强化学习[1]方法受到了广泛的关注，其主要被用于解决序列决策问题。强化学习受到动物学习中试错法的启发，将智能体与环境交互得到的奖励值作为反馈信号对智能体进行训练。强化学习一般可以用马尔可夫决策过程（Markov decision process，MDP）表示，主要元素包含（S, A, R, T, γ），其中，S 表示所处的环境状态，A 表示智能体采取的动作，R 表示得到的奖励值，T 表示状态转移概率，γ 表示折扣因子。智能体的策略 π 表示状态空间到动作空间的一个映射。当智能体状态 $s_t \in S$ 时，根据策略 π 采取动作 $a_t \in A$，进而根据状态转移概率 T 转移到下一个状态 s_{t+1}，同时接收环境反馈的奖励值 $r_t \in R$。强化学习的目标是不断地优化智能体的策略，从而得到最大的奖励值。智能体的值函数和动作值函数分别为 $V(s_t)$ 和 $Q(s_t, a_t)$，用来评估智能体在状态 s_t 下所能得到的长期奖励的期望。智能体的最优策略可以通过优化值函数得到。

深度学习拥有强大的感知能力，在一些应用场

收稿日期：2020−11−18；**修回日期：**2020−12−03
通信作者：穆朝絮，cxmu@tju.edu.cn
基金项目：国家自然科学基金资助项目（No.61773373）
Foundation Item: The National Natural Science Foundation of China (No.61773373)

景下甚至已经超越了人类的感知水平[2]。它采用深度神经网络提取原始输入的特征，在图像识别、语音识别、机器翻译等多个领域取得了成功。深度强化学习（deep reinforcement learning，DRL）基于深度学习强大的感知能力来处理复杂的、高维的环境特征，并结合强化学习的思想与环境进行交互，完成决策过程[3-4]。2015 年 DeepMind 团队在 *Nature* 上发表了深度 Q 网络（deep Q-network，DQN）的文章[5]，认为 DRL 可以实现类人水平的控制。2017 年，DeepMind 团队根据深度学习和策略搜索的方法推出了 AlphaGo[6]，击败了围棋世界冠军李世石。此后，基于 DRL 的 AlphaGo Zero 在不需要人类经验的帮助下，经过短时间的训练就击败了 AlphaGo[7]。2018 年，OpenAI 团队基于多智能体 DRL（multi-agent DRL，MADRL）推出了 OpenAI Five，在 Dota2 游戏 5v5 模式下击败了人类玩家[8]，并且在经过一段时间的训练后，击败了 Dota2 世界冠军 OG 战队。2019 年，DeepMind 团队基于 MADRL 推出的 AlphaStar 在 StarCraftII 游戏中达到了人类大师级的水平，并且在 StarCraftII 的官方排名中超越了 99.8%的人类玩家[9]。可以看到，DRL 在封闭、静态和确定性的环境（如围棋、游戏等）下，可以达到甚至超越人类的决策水平。

DRL 算法主要分为两类：值函数算法和策略梯度算法[10-11]。值函数算法通过迭代更新值函数来间接得到智能体的策略，当值函数迭代达到最优时，智能体的最优策略通过最优值函数得到。策略梯度算法直接采用函数近似的方法建立策略网络，通过策略网络选取动作得到奖励值，并沿梯度方向对策略网络参数进行优化，得到优化的策略最大化奖励值。在算法应用的场景上，值函数算法需要对动作进行采样，因此只能处理离散动作的情况，而策略梯度算法直接利用策略网络对动作进行搜索，可以被用来处理连续动作的情况。近年来，将值函数算法和策略梯度算法结合得到的执行器–评价器（actor-critic，AC）结构也受到了广泛的关注。在 AC 结构中，执行器使用策略梯度法选取动作，通过值函数对执行器采取的动作进行评价，并且在训练时，执行器和评价器的参数交替更新。

本文详述了基于值函数和策略梯度的 DRL 算法，对深度 Q 网络、深度策略梯度及相关算法进行了梳理，并概述了近年来 DRL 在视频游戏、导航、多智能体协作以及推荐系统等领域的应用。最后，总结了 DRL 目前面临的挑战，建议在采样和探索效率、奖励值设置、泛化能力等方面开展研究，更好地提升 DRL 算法的性能和应用效果。

2　基于值函数的 DRL 算法

基于值函数的 DRL 算法采用深度神经网络对值函数或者动作值函数进行近似，通过时间差分（temporal difference，TD）学习[12]或者 Q 学习[13]的方式分别对值函数或者动作值函数进行更新。

2.1　深度 Q 网络

2015 年，DeepMind 团队提出的 DQN 在 Atari 2600 游戏中达到了人类玩家水平，DQN 采用的整个网络结构如图 1 所示。DQN 采用卷积神经网络处理原始的视频游戏图像，以端对端的方式对智能体进行训练。为了保证训练的稳定性，DQN 使用经验回放机制，将智能体与环境进行在线交互的经验数据存储到经验池中，并在训练时对经验池中的数据进行随机小批量的采样，打破数据之间的相关性；采用每隔 N 次迭代更新目标 Q 网络的方法，避免训练时目标 Q 网络随着当前 Q 网络变化导致的不稳定。DQN 算法更新流程如图 2 所示。

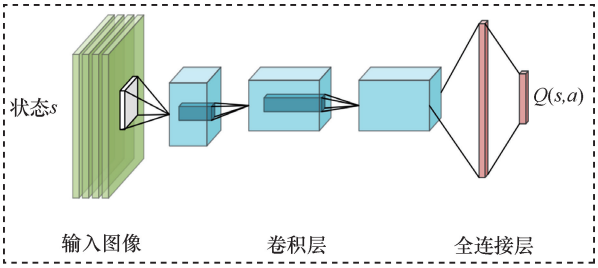


图 1　DQN 的网络结构

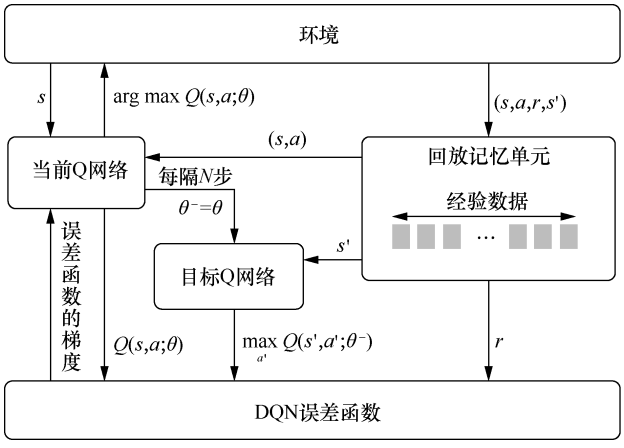


图 2　DQN 算法更新流程

当前 Q 网络用 $Q(s,a;\theta_i)$ 来表示，目标 Q 网络用 $Q(s',a';\theta_i^-)$ 来表示，其中 θ_i 和 θ_i^- 表示第 i 次迭代的网络参数。DQN 的误差函数表示为：

$$L(\theta_i)=E_{(s,a,r,s')\sim U(D)}[(r+\gamma\max_{a'}Q(s',a';\theta_i^-)-Q(s,a;\theta_i))^2] \quad (1)$$

其中，$Y_i=r+\gamma\max_{a'}Q(s',a';\theta_i^-)$ 表示目标网络，(s,a,r,s') 表示经验数据，$U(D)$ 表示经验数据的回放记忆单元。在训练过程中，DQN 从回放记忆单元中随机小批量地抽取经验数据进行训练，用误差函数对参数 θ 求偏导得到：

$$\nabla_{\theta_i}L(\theta_i)=E_{(s,a,r,s')\sim U(D)}[(Y_i-Q(s,a;\theta_i))\nabla Q(s,a;\theta_i)] \quad (2)$$

同时，神经网络的参数采用梯度下降的方式进行更新。实验表明，DQN 不仅在多种 Atari 2600 游戏中达到人类玩家的水平，还显示出很强的适应性和通用性。

2.2　深度双 Q 网络

DQN 的出现促进了 DRL 的兴起。然而，DQN 还存在一些不足之处，例如对动作值函数的过估计。DQN 的优化目标为 $Y_i=r+\gamma\max_{a'}Q(s',a';\theta_i^-)$，每一次更新时它都会对目标 Q 网络采取最大化操作，这样会导致对 Q 值的过高估计。深度双 Q 网络（double DQN，DDQN）[14]在目标 Q 函数中采用双网络结构，根据当前 Q 网络选取最优动作，并使用目标 Q 网络对选取的最优动作进行评估，两套参数将动作选择和策略评估分开，降低了过估计的风险。DDQN 的目标函数表示为：

$$Y_i=r+\gamma Q(s',\arg\max Q(s',a;\theta_i);\theta_i^-) \quad (3)$$

DDQN 采用与 DQN 相同的更新方式。实验结果表明，DDQN 能在大部分 Atari 2600 游戏上取得比 DQN 更好的表现，并且得到更加稳定的策略。

2.3　基于优先经验回放的深度 Q 网络

经验回放技术打破了数据样本之间的相关性，保证了 DQN 训练的稳定性。但 DQN 中的经验数据是均匀随机采样的，有些关键的经验数据可能无法被采样到，这降低了算法的更新效率。为此，Schaul T 等人[15]采用优先经验回放（prioritized experience replay）的方式来代替均匀随机采样，从而使策略更新速度加快。具体实施方法是：在提取经验数据时，根据 TD 误差的大小来判断优先级。TD 误差表示为 $r+\gamma\max_{a'}Q(s',a';\theta^-)-Q(s,a;\theta)$，并且 TD 误差的绝对值越大，该经验样本被选取的概率越高。同时，优先经验回放在采样过程中使用随机比例化（stochastic prioritization）和重要性采样权重（importance-sampling weight）两种技术。随机比例化使智能体以 TD 误差的大小进行概率采样，从而扩展了采样的多样性，保证了各个样本都有概率被采样到。重要性采样权重的使用放缓了参数更新的速度，保证了学习的稳定性。在 Atari 2600 游戏中，使用优先经验回放的 DQN 不仅可以提升算法收敛的速度，同时也能够取得更好的性能表现。

2.4　基于竞争架构的深度 Q 网络

基于竞争架构的深度 Q 网络（Dueling DQN）[16]从网络结构上对 DQN 进行改进。与 DQN 相同，Dueling DQN 采用卷积神经网络处理原始输入，但经过卷积神经网络处理的特征被分流到两个全连接网络中，一个是值函数网络 $V(s;\theta,\beta)$，另一个是优势函数网络 $A(s,a;\theta,\alpha)$，其中 θ 是卷积神经网络的权重，β 是值函数网络全连接层的权重参数，α 是优势函数网络全连接层的权重参数。Dueling DQN 的网络结构如图 3 所示，它的值函数网络被用来评

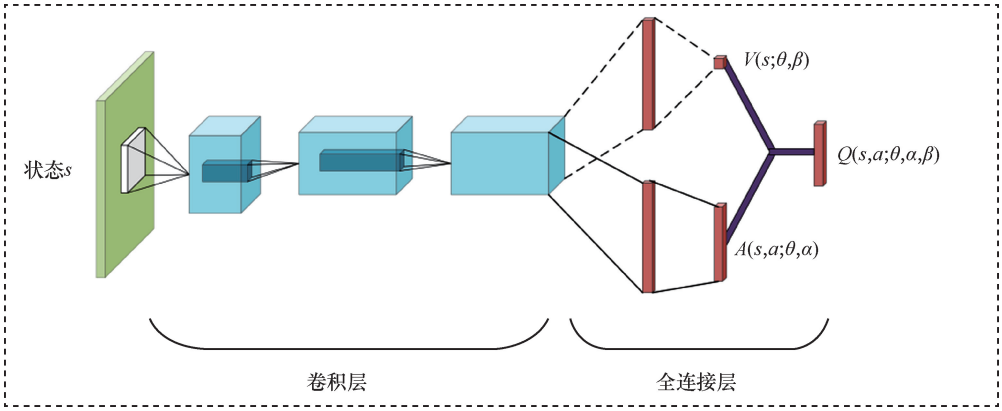


图 3　Dueling DQN 的网络结构

估当前状态的价值，优势函数网络被用来处理与当前状态有关的动作。最后，将这两个网络进行合并，得到：

$$Q(s,a;\theta,\alpha,\beta)=V(s;\theta,\beta)+A(s,a;\theta,\alpha) \tag{4}$$

在实际中，一般要将优势函数减去当前状态下所有动作优势函数的平均值，获得的动作值函数如下：

$$Q(s,a;\theta,\alpha,\beta)=V(s;\theta,\beta)+$$
$$\left(A(s,a;\theta,\alpha)-\frac{1}{|A|}\sum_{a'}A(s,a';\theta,\alpha)\right) \tag{5}$$

这样做可以保证该状态下各动作的优势函数相对排序不变，而且可以缩小 Q 值的范围，去除多余的自由度，提高算法的稳定性。实验表明，Dueling DQN 能够更准确地估计 Q 函数，并且能够在 Atari 2600 游戏中取得比 DQN 更好的表现。

2.5　分布式深度 Q 网络

在训练过程中，DQN 使用单个机器进行训练，这导致在实际中训练时间较长。为了充分利用计算资源，Nair A 等人[17]提出一种分布式架构来加快算法的训练速度。它主要包括 4 个部分：①并行的行动者，算法采用 N 个不同的行动者，每个行动者复制一份 Q 网络，并在同一个环境中执行不同的动作，从而得到不同的经验；②经验回放存储机制，它将 N 个行动者与环境交互的经验存储到经验池中；③并行的学习者，算法采用 N 个学习者使用经验池存储的经验数据来计算损失函数的梯度，并发送到参数服务器中，从而对 Q 网络的参数进行更新；④参数服务器，用来接收学习者发送的梯度，并通过梯度下降的方式对 Q 网络参数进行更新。在 49 个 Atari 2600 游戏中，有 41 个游戏的基于分布式 DQN 算法的性能超过了 DQN，并且在多种 Atari 2600 游戏中，分布式 DQN 算法的训练时间大大减少。

3　基于策略梯度的 DRL 算法

基于策略梯度的 DRL 算法主要包括策略梯度算法、AC 算法以及基于 AC 的各种改进算法，如深度确定性策略梯度（deep deterministic policy gradient，DDPG）算法、异步优势 AC（asynchronous advantage AC，A3C）算法和近端策略优化（proximal policy optimization，PPO）算法等。

3.1　策略梯度算法

策略梯度算法的策略网络用 π_θ 来表示，其中，θ 表示策略的权重参数。它的优化目标为 $\max\limits_{\theta} E[R\,|\,\pi_\theta]$，其中 $R=\sum\limits_{t=0}^{T}r_t$ 表示智能体经历一个序列得到的所有奖励值。策略梯度算法的具体更新过程如下。假设一个完整序列的状态、动作和奖励的轨迹为 $\tau=(s_0,a_0,r_1,\cdots,s_{T-1},a_{T-1},r_{T-1},s_T)$，则策略梯度可以表示为以下形式：

$$g=R\nabla_\theta\sum_{t=0}^{T-1}\ln\pi(a_t\,|\,s_t;\theta) \tag{6}$$

其中，R 控制着序列更新的方向，当 R 为正且越大时，该动作序列的轨迹出现的概率就会越高；反之，该轨迹出现的概率就会越小。$\nabla_\theta\sum\limits_{\tau=0}^{T-1}\ln\pi(a_t\,|\,s_t;\theta)$ 使策略朝着变化最快的方向进行更新。对策略参数的更新可以表示为：

$$\theta\leftarrow\theta+\alpha g \tag{7}$$

其中，α 表示学习率。经过一段时间的迭代之后，策略网络将会提高总奖励 R 较高的序列 τ 出现的概率，同时降低总奖励 R 较低的序列出现的概率。

3.2　基于执行器–评价器的深度策略梯度算法

策略梯度算法直接对智能体的策略进行优化，它需要收集一系列完整的序列数据 τ 来更新策略。在 DRL 中，对序列数据进行收集往往很困难，并且以序列的方式对策略进行更新会引入很大的方差。一种可行的方案是将传统强化学习中的 AC 结构应用到 DRL 中。AC 结构主要包括执行器和评价器两部分，其中执行器基于策略梯度算法更新动作，评价器则基于值函数法对动作进行评价。AC 结构的优点是将策略梯度中的序列更新变为单步更新，不用等序列结束后再对策略进行评估和改进，这样可以减少数据收集的难度，同时可以减小策略梯度算法的方差。

对于值函数部分，也可以用优势函数来代替。优势函数可以表示为：

$$A_t=Q(s_t,a_t)-V(s_t) \tag{8}$$

或

$$A_t=r_t+\gamma V(s_{t+1})-V(s_t) \tag{9}$$

使用优势函数代替 Q 函数，可以提高“好”动作出现的概率。使用优势函数可以进一步地减小算法的方差，基于优势函数的 AC 结构被称为优势 AC（advantage AC，A2C）算法。A2C 的基本结构如图 4 所示。

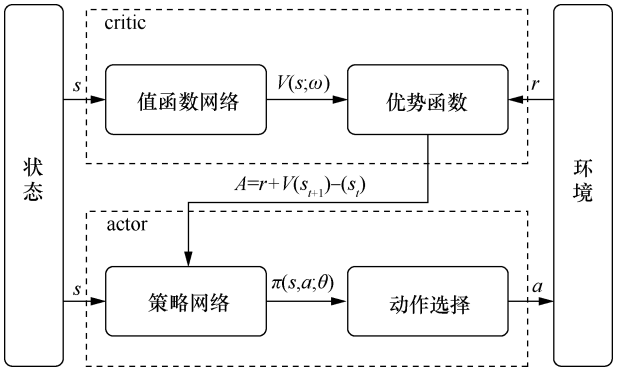


图 4　A2C 的基本结构

3.3　深度确定性策略梯度算法

策略梯度算法一般采用随机性策略进行表示，表示为 $\pi_\theta(a|s)=P[a|s;\theta]$。然而随机性策略梯度算法需要对动作进行采样，当动作空间较大时，采样的计算量也会随之增加。为此，Silver D 等人[18]提出确定性策略梯度（deterministic policy gradient，DPG）算法，采用确定性的方式对动作进行采样。确定性策略表示为 $\mu_\theta(s)=a$，即在当前状态下采取确定的动作。为进一步提升算法的通用性，Lillicrap P T 等人[19]将 DQN 和 DPG 算法进行结合，提出了 DDPG 算法。DDPG 分别将 θ^μ 和 θ^Q 作为神经网络的参数来表示确定性策略 $a=\mu(s|\theta^\mu)$ 和值函数 $Q(s,a|\theta^Q)$。其中，策略网络被用来更新策略，相当于 AC 结构中的执行器；值函数网络被用来对动作进行评价，并提供梯度信息，相当于 AC 结构中的评价器。策略网络的更新过程表示为：

$$\nabla_{\theta^\mu}\mu\approx E_{\mu'}[\nabla_a Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t)}\nabla_{\theta_\mu}\mu(s|\theta^\mu)|_{s=s_t}] \tag{10}$$

$$\theta^\mu_{t+1}=\theta^\mu_t+\alpha_\mu\nabla_{\theta^\mu}\mu \tag{11}$$

值函数网络的更新过程为：

$$\delta_t=r_t+\gamma Q'(s_{t+1},\mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})-Q(s_t,a_t|\theta^Q) \tag{12}$$

$$\theta^Q_{t+1}=\theta^Q_t+\alpha_Q\delta_t\nabla_{\theta^Q}Q(s_t,a_t|\theta^Q) \tag{13}$$

其中，α_μ 和 α_Q 表示学习率，$\theta^{\mu'}$ 和 $\theta^{Q'}$ 表示目标网络的参数，更新方法为：

$$\theta^{Q'}\leftarrow\tau\theta^Q+(1-\tau)\theta^{Q'} \tag{14}$$

$$\theta^{\mu'}\leftarrow\tau\theta^\mu+(1-\tau)\theta^{\mu'} \tag{15}$$

其中，τ 表示更新率，且值远小于 1。

同时 DDPG 算法还加入噪声来增加探索，进一步提升算法的性能。DDPG 算法在一系列连续动作空间的任务中都能表现稳定。相对于 DQN 来说，DDPG 在 Atari 2600 游戏中能够取得更高的效率，训练时间更少。

3.4　异步优势算法

A3C 基于异步强化学习（asynchronous reinforcement learning）的思想[20]，采用多线程操作，每一个线程异步执行智能体的动作。在每一时刻，各个执行器都经历不同的状态，并采取不同的动作，去除了训练过程中样本之间的相关性，因此这种异步的方式能够很好地代替经验回放，并且可以使用同策略（on-policy）的方式对参数进行更新。A3C 算法显著地降低了对硬件的要求。以往的策略梯度算法需要计算能力很强的处理器 GPU，而 A3C 算法在训练过程中只需要一个多核的 CPU。在 Atari 2600 的游戏仿真中，A3C 算法不仅大大降低了训练时间，而且平均性能也有明显提升。此外，A3C 算法可以广泛地应用于各种 2D、3D 离散和连续动作的情况，具有很强的通用性。

3.5　近端策略优化算法

对于近年来的 DRL 算法来说，DQN 由于不能处理连续动作，导致它的应用范围有限，A3C 算法面临着超参数调整以及较低的采样效率。为此，Schulman J 等人[21]提出了 PPO 算法。PPO 算法采用一种“替代”目标：

$$L(\theta)=\hat{E}[r_t(\theta)\hat{A}_t] \tag{16}$$

其中，$r_t(\theta)=\dfrac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ 表示新策略与旧策略的比率，$\hat{A}_t$ 表示对优势函数的估计。PPO 算法采用旧策略对新策略进行优化，目标是使新策略产生的动作比旧策略好，但对新策略的改进不能太大，否则会造成训练算法的不稳定。为此，PPO 算法对目标函数进行改进，新的目标函数表示为：

$$L(\theta)=\hat{E}[\min((r_t(\theta),\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon))\hat{A}_t)] \tag{17}$$

其中，$\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon)$ 表示对策略的比率 $r_t(\theta)$ 进行修剪，使其处于 $[1-\varepsilon,1+\varepsilon]$。修剪使得新策略不会发生太大的变化，从而使算法稳定。在 MuJoCo、Atari 2600 以及 3D 环境中，PPO 算法能够取得比 A2C 算法和 A3C 算法更好的表现。

3.6　基于最大熵的执行器–评价器算法

策略梯度算法面临的一个很大的挑战是收敛性问题，即需要对学习率、探索因子等超参数进行

精细的调整。为此，Haarnoja T 等人[22]提出了一种基于最大熵的 AC（soft AC，SAC）算法。SAC 算法将熵的思想加入目标函数中，通过最大化目标函数得到最优奖励，同时也最大化熵，即智能体在完成任务的同时尽可能采取随机动作。SAC 算法的目标函数的计算式为：

$$J=\sum_{t=0}^{T}E_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t)+\alpha H(\pi(\cdot|s_t))] \quad (18)$$

其中，$H(\pi(\cdot|s_t))$ 表示策略在当前状态下的熵，ρ_π 表示策略 π 下的状态−动作对的分布，α 表示熵相对于奖励值的重要性。

SAC 算法通过使熵最大化来激励智能体探索，一方面可以避免智能体收敛到次优策略，另一方面可以提升算法的鲁棒性，并且 SAC 算法能够在多种连续控制的任务中取得比 DDPG 算法和 PPO 算法更好的表现。

4　深度强化学习的几类应用

DRL 的不断发展推动了它在多个领域中的应用，其在视频游戏、导航、多智能体协作以及推荐系统等领域取得了较好的表现。表 1 是几类 DRL 的应用领域及研究意义。

4.1　DRL 在视频游戏中的应用

DRL 首先应用于视频游戏领域[23]，主要的原因是 DRL 需要大量的采样和试错训练，而游戏环境能够提供充足的样本，并且避免了试错的成本。从目前的文献来看，研究 DRL 所采用的游戏环境可以分为两类：一类用来提升算法的通用性，如 Atari 2600；另一类用来处理复杂的游戏场景，如 ViZDoom、StarCraftII 等。

图 5 展示了几个典型的 Atari 2600 游戏环境，它们是很多 DRL 算法的测试环境。Atari 2600 包含 57 款游戏，可以用来模拟真实环境中遇到的情况，且游戏环境具有多样性。DQN 较早应用于 Atari 2600 的算法[5]，它在 22 款游戏中取得了人类玩家的平均水平。之后，DQN 的各种改进版本（如 DDQN[14]、优先经验回放的 DQN[15]、Dueling DQN[16]，以及分布式 DQN[17]）分别在 Atari 2600 游戏中取得了不同程度的提升。然而，很难有一种算法能够在所有 Atari 2600 游戏中都达到人类的水平，这主要有两个原因：一个是长期的信度分配，另一个是探索困境。信度分配在 DRL 中用来解决动作−奖励值分配的问题，DRL 中存在奖励延时，因此很难将奖励值分配到具体的行动中，而且游戏运行时间越长，信度分配问题就越难解决。探索困境是指当环境状态的维数很大并且奖励值的设置比较稀疏时，智能体需要很多探索行为才能得到积极的反馈，导致算法很难收敛。为了解决这些问题，Badia A P 等人[24]提出了 Agent57 算法，并成功地在所有 Atari 2600

表 1　几类 DRL 的应用领域及研究意义

应用领域	分类方式	参考文献	研究意义
视频游戏	Atari 2600	[5,14-17,24]	将 DRL 应用在多种游戏环境中，提升 DRL 算法的通用性
	ViZDoom、StarCraftII 等	[25-34]	将 DRL 应用到复杂的游戏场景中，提升智能体的决策能力
导航	迷宫导航	[35-40]	根据应用场景设计迷宫环境，采用 DRL 处理特定的导航问题
	室内导航	[41-45]	采用 DRL 算法训练智能体在室内环境进行导航，并尝试将虚拟环境中训练好的智能体应用到现实环境中
	街景导航	[46-49]	采用 DRL 处理城市与城市之间的长距离导航，并提升 DRL 算法的泛化能力
多智能体协作	独立学习者协作	[50-59]	协作智能体在训练时使用独立 DRL 的方式，方便进行数量上的扩展
	集中式评价器协作	[60-64]	协作智能体在训练时通过集中式的评价器获取其他智能体的信息，解决环境非静态问题
	通信协作	[65-69]	利用 DRL 处理多智能体之间可以通信的情况，并采用通信促进智能体之间的协作
推荐系统	推荐算法	[70-73]	利用 DRL 进行推荐可以实时地对推荐策略进行调整，从而满足用户的动态偏好，并且推荐算法能够得到长期的回报

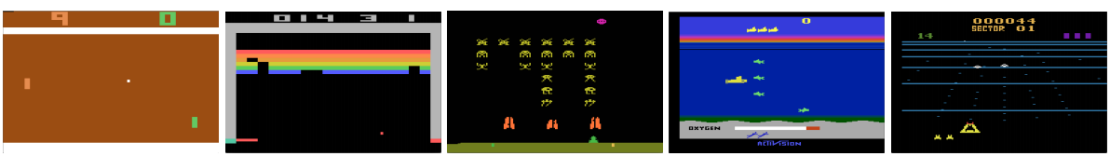


图 5　Atari 2600 典型游戏环境

游戏中超越了人类平均水平。针对长期的信度分配问题，Agent57 算法动态地对折扣因子进行调整，权衡未来奖励对当前状态−动作的重要性。同时 Agent57 算法加入内部奖励值来解决探索困境，内部奖励值的大小由未探索过状态的新奇程度来决定。Agent57 算法采用两种内部的奖励值：一种是在一个学习周期（episode）内根据新奇的状态得到的奖励值；另一种是长期的，在学习周期之间根据状态的新奇程度得到的奖励值。最后的内部奖励值由这两部分组成。通过这些方法，Agent57 算法成功地在所有 Atari 2600 游戏中超越人类玩家的平均水平，提高了 DRL 算法的通用性。

复杂游戏环境的状态和动作空间显著增大，更多地考虑 3D 环境的情况，并且涉及单智能体的多任务学习，或者多智能体之间的交互。Kempka M 等人[25]提出将 ViZDoom 作为 DRL 的测试平台。ViZDoom 采用第一视角的 3D 游戏环境，且游戏环境可以自由设计，以测试算法在不同任务上的性能。Lample G 等人[26]提出了一种改进的 DQN 算法并将其应用于 ViZDoom，该算法主要解决游戏中的两个任务，一个是在地图中导航搜寻敌人或者弹药，另一个是在发现敌人时采取射击行动，并且对每个任务分别采用一个网络，两个网络交替训练能够取得很好的效果。Dosovitskiy A 等人[27]将一种有监督学习的方法应用于 ViZDoom，使用高维感知流和低维测量流分别处理高维的原始图像以及与智能体当前状态有关的信息，并且训练好的模型可以被用于动态的指定目标。Pathak D 等人[28]引入好奇心机制来解决 ViZDoom 游戏环境中奖励的稀疏性问题，将当前状态特征和动作输入前向模型中，对下一时刻的状态特征进行预测，并将预测误差作为内部奖励，鼓励智能体探索新奇的环境状态，同时还使用逆模型来提取只与当前动作有关的环境特征。针对 ViZDoom 游戏的复杂性，Wu Y 等人[29]提出了一种主从式课程 DRL 的方法，该方法引入了主从智能体的概念，其中一个主智能体被用来处理目标任务，多个从智能体被用来处理子任务，并且主从智能体可以使用不同的动作空间，同时引入课程学习，大大提高了算法的训练速度，显著地提高了智能体在游戏中的表现性能。

Vinyals O 等人[30]提出了一种基于实时策略（real-time strategy，RTS）的 DRL 游戏环境 StarCraftII。游戏环境中存在多智能体模式，由多个智能体通过协作或者竞争来取得游戏的胜利。在游戏环境中智能体接收有限的观测范围，通过高维、多变的动作空间选取动作，并且智能体还需要处理长期的信度分配问题以及探索困境。为了方便对 DRL 算法的研究，将 StarCraftII 简化为 7 个微型游戏，每个微型游戏分别代表不同的子任务。针对这个游戏环境，Zambaldi V 等人[31]提出了一种关系型 DRL（relational DRL）算法，该算法将关系型学习引入 DRL 中，将环境编码成二维输入，其中每个区域代表一个“实体”，并采用自我注意力机制[32]来计算各个“实体”之间的关系。实验表明，提出的算法在 6 个 StarCraftII 的微型游戏中取得了较好的结果，并且在 4 个微型游戏中超越了人类大师水平的玩家。Rashid T 等人[33]采用一种 MADRL 算法来处理 StarCraftII，提出 QMIX 算法将每个智能体的 Q 函数进行单调非线性的结合，形成一个联合 Q 函数，并使用联合 Q 函数进行学习。这种集中式训练、分布式执行的方式在 StarCraftII 上有很好的表现。

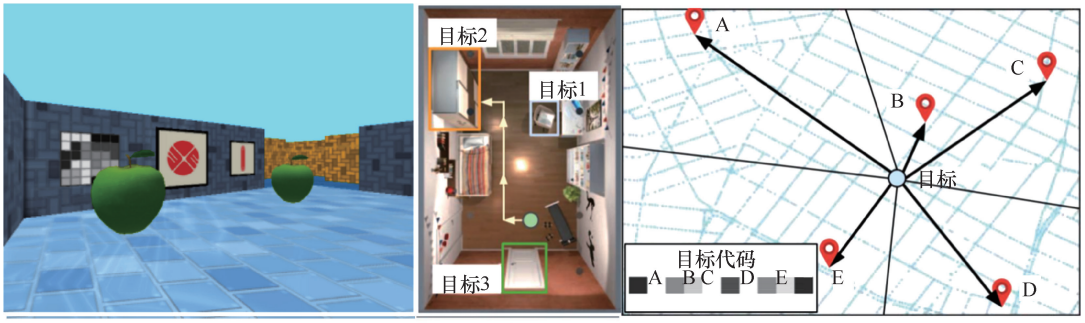
腾讯的 AI Lab 利用 DRL 研究了多人在线战术竞技（multi-player online battle arena，MOBA）游戏的 1v1 模式，该游戏具有很复杂的环境，并且需要很多的控制量。以 MOBA 中的王者荣耀（Honor of Kings）为例，它具有比围棋更大的状态和动作空间，给策略搜索带来了巨大的挑战。Ye D 等人[34]提出了一种包含人工智能服务器、调度模块、记忆池以及强化学习学习者的 DRL 架构来处理该游戏环境，其中人工智能服务器负责与环境进行交互来产生经验数据；调度模块负责将人工智能服务器产生的经验数据进行压缩、打包，送到记忆池中；记忆池用于存储数据，并支持各种长度的样本数据，以及基于生成时间的数据采样；强化学习学习者采用分布式训练，并行地从记忆池中采样数据得到梯度，在同步梯度取均值后，对策略参数进行更新，并且将更新的策略参数传到人工智能服务器中。经过训练后，提出的算法在 2 100 场 Honor of Kings 的 1v1 竞赛中的获胜率为 99.81%。

4.2 DRL 在导航中的应用

导航是 DRL 的另一个重要应用，它的目标是使智能体找到一条从起点到目标点的最优路径，同时，在导航中还需要完成各种任务，如避障、搜集物品以及导航到多个目标等。近年来，利用 DRL 在迷宫导航、室内导航、街景导航的研究取得了一系列的成果。

对迷宫导航的研究首先根据应用场景设计迷宫环境，然后根据导航环境中要解决的问题采用相应的 DRL 算法。Oh J 等人[35]针对多变的 3D 导航环境 Minecraft，提出了一种基于记忆的 DRL 架构。智能体提前对导航环境信息进行记忆，然后根据记忆的信息到达指定的目标点并获得奖励。Jaderberg M 等人[36]提出采用无监督的辅助任务来增强 A3C 算法在迷宫导航中的性能。该辅助任务一方面将输入的迷宫图像分为多个不重叠的小区域，根据区域像素的变化生成伪奖励，鼓励智能体探索未知环境；另一方面使用先前的经验数据对下一时刻状态的即时奖励进行预测，解决环境奖励的稀疏性问题。经过辅助任务的加速训练，提出的算法在收敛速度、鲁棒性、成功率上都取得了较大的提升。Mirowski P 等人[37]提出了一种基于 A3C 的 DRL 算法，使用其在复杂的迷宫环境中进行导航，迷宫环境如图 6（a）所示。除了 A3C 算法的训练任务外，该算法还加入了两个辅助任务。第一个辅助任务是重建一个低维的深度图进行避障及短期路径规划，第二个辅助任务是直接从同步定位与建模（simultaneous localization and mapping，SLAM）中调用循环闭包来防止环路的形成。经过算法的训练，智能体能够在目标不断变化的 3D 迷宫环境中达到人类水平。Wang Y 等人[38]提出了一种模块化的 DRL 算法来处理复杂的动态迷宫环境，并将导航任务分为避障模块和导航模块。避障模块采用时间和空间两个网络来处理动态障碍物信息。导航模块分为离线部分和在线部分，离线部分要进行预训练，用于快速地在无障碍的迷宫环境中找到终点，在线部分实时地在迷宫中探索、寻找路径。最后将两个模块采取的动作发送到动作调度器中选取智能体的动作。Shi H 等人[39]提出了一种端对端的 DRL 导航策略，用于迷宫导航。该导航策略采用好奇心机制解决导航环境中奖励稀疏的问题，同时将低维、稀疏的传感器用于智能体的观测输入，确保训练好的智能体可以直接应用于现实的导航环境中。Savinov N 等人[40]将已经访问过的状态放入记忆模块中，并训练一个神经网络近似器来判断当前状态跟记忆中的状态的距离，并根据这个距离来判断这个状态是否为新奇的状态。经过该算法的训练，智能体能够在多种导航任务中取得好的表现，并且能够被用在无奖励的导航任务中。

室内导航采用 DRL 算法在室内环境中找到指定的目标点，主要的研究包括多目标导航以及将在仿真环境中训练好的智能体直接用到现实环境中。Zhu Y 等人[41]提出了一种多目标室内导航的 DRL 算法，为了保证训练好的智能体能够被直接应用到新的导航目标，将导航的目标点和智能体的观测合并，作为输入，得到策略及值函数，采用 A3C 算法进行训练。同时，将算法应用于图 6（b）所示的 3D 室内导航环境中，训练后的智能体经过参数微调可以被直接应用到现实环境中。Tai L 等人[42]采用 DRL 算法训练智能体在迷宫中避开各种障碍物，将原始的 RGB 图像作为输入，使用深度学习在导航环境中进行预训练，并把预训练的权重复制到 DRL 的卷积层上，然后端对端地对智能体进行训练。此后 Tai L 等人[43]又提出了一种连续控制的 DRL 导航方法，将虚拟导航环境中训练好的智能体直接应用到现实环境中。为了减少虚拟环境和现实环境的偏差，其采用低维的激光传感器作为视觉输入，并且输出智能体的线速度和角速度，以达到连续控制的效果。Wu Y 等人[44]提出了两个设计来提升 DRL 方法室内多目标导航的表现性能：一是将逆动态模型（inverse dynamics model，InvDM）引



（a）迷宫导航　　（b）室内导航　　（c）街景导航

图 6　DRL 导航环境

入 AC 结构中，该 InvDM 被训练为一个辅助任务，用于预测给定了当前状态和最后一个状态的情况下智能体所采取的最后一个动作；二是提出了一种多目标共同学习的方法，智能体导航到一个目标的路径中可能包含其他的目标，因此这条路径可以为导航到其他目标点提供帮助。Zhang W 等人[45]提出了一种适用于不同大小智能体的 DRL 算法，先利用 SAC 算法在虚拟导航环境中进行训练，然后将训练好的智能体直接应用到现实的导航环境中，进一步采用元学习方法将现实环境中训练好的导航技巧扩展到其他不同大小的智能体上。

街景导航的研究涉及长距离导航，主要通过 DRL 算法来解决城市之间的路径规划问题。Mirowski P 等人[46]提出了一种无地图城市之间导航的 DRL 算法，主要使用谷歌街景地图的内容构建了一个覆盖全球的交互导航环境 StreetLearn，并提出了一种模块化、以目标为导向的 DRL 算法来处理这个导航场景。Li A 等人[47]提出了利用鸟瞰以及地面两种视图来完成导航任务的 DRL 方法，先使用街景环境的地面和鸟瞰视图对智能体进行训练，通过新的导航环境的鸟瞰视图对智能体的训练结果进行调整，然后将调整的结果迁移到新的地面视图街景导航环境中。Hermann K M 等人[48]在 StreetLearn 的基础上，加入了谷歌地图中的驾驶说明，用于指导从起点到终点的路径，使智能体像人类一样按照指示在城市之间导航，构建了一种新的街景导航环境 StreetNav。针对 DRL 算法在现实导航任务中采样效率不足的问题，Chancán M 等人[49]提出了一种交互式街景导航架构 CityLearn，采用视觉位置识别和深度学习模型对传感器的输入图像进行编码，将目标位置生成状态作为智能体的输入状态，再使用 PPO 算法对智能体进行训练，得到导航策略，训练好的智能体在极度变化的视觉环境中（如从白天到黑夜）也能应用。

4.3 DRL 在多智能体协作中的应用

近年来，DRL 在多智能体协作方面也得到了广泛的应用。采用 DRL 解决多智能体协作问题涉及智能体之间的交互，需要考虑状态–动作维数过大、环境非静态、部分可观测等问题[50]，因此相对于单智能体来说，对多智能体的研究更具有挑战性。多智能体协作是指多个智能体通过相互合作达到共同的目标，从而得到联合的奖励值。DRL 在多智能体协作方面的研究主要包括独立学习者协作、集中式评价器协作以及通信协作等[51]。

独立学习者是指智能体在更新自身的策略时，把其他智能体作为环境的一部分，每个智能体采用独立更新的方式，不考虑其他智能体的状态和动作。若采用这种更新方式，每个智能体在训练时比较简单，方便智能体进行数量上的扩展。但其他智能体的策略在不断更新，因此智能体所处的环境是不断变化的，这导致智能体不满足 MDP 条件，即 MADRL 面临环境非静态问题。Omidshafiei S 等人[52]提出使用滞回强化学习[53]的方法来解决环境非静态问题，通过对不同的 TD 误差采用不同大小的学习率，减弱环境变化对 Q 值的影响，并采用一种并行经验回放的方式，保证多个智能体在使用经验回放时能够得到最优的联合动作。对于多智能体来说，当智能体的策略不断变化时，经验回放技术也不再适用，这给 MADRL 带来了很大的挑战。Palmer G 等人[55]提出了宽松 DQN（lenient DQN）算法来解决环境非静态问题，以宽松条件来决定经验池中的采样数据，不满足条件的经验数据将被忽略。Jin Y 等人[57]提出了对其他智能体的动作进行估计的方法来解决环境非稳态问题，在评估 Q 函数时加入对其他智能体动作的估计，减弱环境非静态带来的影响。Liu X 等人[58]对邻近智能体的关系进行建模，提出了注意力关系型编码器来聚合任意数量邻近智能体的特征，并采用参数共享[59]的方式来减少参数的更新量，使算法可扩展到大规模智能体的训练。

Lowe R 等人[60]采用集中式训练、分布式执行的训练机制，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，结构如图 7 所示。该机制采用集中式评价器，假设智能体的评价器在训练时能够得到其他所有智能体的状态和动作信息，这样即使其他智能体的策略发生变化，环境也是稳定的。而执行器只能得到环境的局部信息来执行动作，训练结束后，算法只采用独立的执行器进行分布式执行。这种机制能够很好地解决环境非静态问题，利用了 AC 结构的优势，方便智能体的训练和执行。Foerster J 等人[61]提出了 COMA 策略梯度算法，采用集中式训练、分散式执行的机制，使得每个智能体在协作过程中都能收到对应于自身行动的奖励值，同时提高所有智能体共同的奖励值。Sunehag P 等人[62]将所有智能体的联合 Q 网络分解

为每个智能体单独的 Q 网络，提出 VDN 算法。Mao H 等人[63]提出了基于注意力机制的 MADDPG 算法，对其他智能体的策略进行自适应建模，以促进多智能体之间的协作，同时引入注意力机制来提升智能体建模的效率。Iqbal S 等人[64]在集中式评判器中采用自我注意力机制，使每个智能体都对其他智能体的观测和动作信息进行不同程度的关注，有效提升了算法的效率，并且可以扩展到大规模智能体的情况，同时引入了 SAC 算法来避免收敛到次优的策略，采用 COMA 算法思想解决多智能体信度分配的问题。

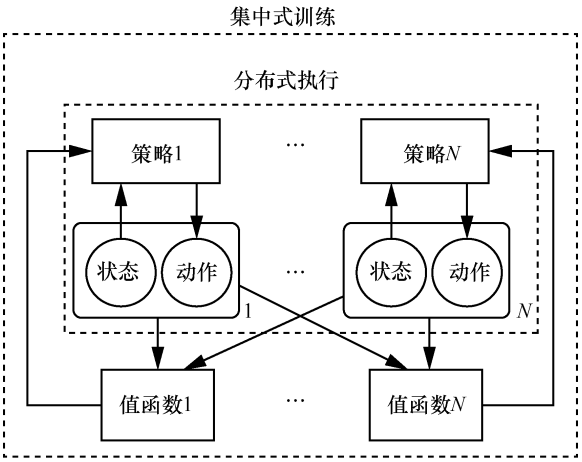


图 7　MADDPG 算法结构

多智能体通信一方面可以促进智能体之间的协作，另一方面，训练时智能体能够得到其他智能体的信息，从而缓解环境非静态问题。Foerster J N 等人[65]使用通信来促进智能体之间的协作，并提出了 RIAL 和 DIAL 两种通信方法。RIAL 的 Q 网络中不仅要输出环境动作，还要输出通信动作到其他智能体的 Q 网络中。DIAL 利用集中式学习的优势，直接在两个智能体的 Q 网络之间建立一个可微信道，促进智能体之间的双向交流。Sukhbaatar S 等人[66]提出了一种通信神经网络模型 CommNet，使得多智能体在协作的过程中能够连续通信。Jiang J 等人[67]提出了注意力通信模型 ATOC，通过注意力单元来选取智能体的通信对象，采用双向长短期记忆（long short term memory，LSTM）单元来收集通信智能体的信息，进而选取动作。Kim D 等人[68]考虑更实际的多智能体通信，即带宽有限以及智能体共享通信介质的情况，提出了 SchedNet 结构，采用基于权重调度的方法和调度向量来确定需要通信的智能体，解决通信资源有限以及智能体竞争通信的问题。Das A 等人[69]提出了一种目标通信结构 TarMAC，采用自注意力机制来计算智能体与其他智能体的通信权重，并根据权重来整合其他智能体的通信信息。

4.4　DRL 在推荐系统中的应用

现有的大多数方法把推荐过程考虑为一个静态过程，按照贪婪策略进行推荐，这些方法无法捕捉到用户的动态偏好；另外，传统方法主要考虑最大化短期贡献，忽略了推荐对象在长期回报中的贡献[70]。DRL 可以很好地应对这些挑战，首先，DRL 可以使用推荐系统与用户实时交互的反馈对推荐策略进行调整，直到最后收敛到一个满足用户动态偏好的最优策略；其次，DRL 的目标是最大化累积的长期奖励，因此利用 DRL 训练出的推荐系统可以在用户推荐中得到长期的回报[71]。例如，Zhao X 等人[72]采用一种基于 AC 结构的 DRL 方法，执行器根据用户的偏好生成推荐页面，评价器对生成的推荐页面进行评估，并且执行器根据评估的结果对推荐策略进行改进。Zheng G 等人[73]提出了一种 DRL 算法，用于新闻推荐，不仅考虑了新闻推荐中存在的适应性和长期性问题，而且在用户反馈中加入了用户的积极性，即用户多久会返回到推荐服务中，此外还提出了一种改进的探索方式来解决现有推荐方法中存在的推荐相似新闻倾向的问题。

5　结束语

现阶段，关于 DRL 的研究已经取得了较大的进步，但在算法上仍存在采样效率不足、奖励值设置困难、探索困境等问题。在应用方面，对 DRL 的研究主要集中在虚拟环境中，无模型 DRL 算法很难应用于现实环境中。这是因为 DRL 算法需要大量的采样数据进行训练，而现实中的样本很难通过试错进行获取。此外，DRL 算法还存在泛化能力不足、鲁棒性不强等问题，这也限制了 DRL 在实际生活中的应用。据此，未来对 DRL 的研究可以从以下方面展开。

- 提高算法的采样效率。现阶段提高采样效率的方式主要是使用经验回放技术，然而经验回放存在数据的实时性无法保证、关键经验数据无法被采样等问题。因此未来对 DRL 的研究应该着眼于提高数据的使用效率，在保证数据的实时性的同时，也需要提出一种更有效的数据采样方式。
- 设置更有效的奖励值。DRL 奖励值的设置

没有一个固定的标准，更多的是靠经验，因此如何有效地设置奖励函数是一项很大的挑战。逆强化学习通过策略或者专家示范来反推出奖励函数，可以作为设计奖励函数的一种方案。另外，也可以模仿学习，使智能体模仿人类的行动，进而得到奖励值。

• 提高算法的探索效率。为了找到最优策略，无模型的 DRL 依赖于很多的探索，而大多数现实的应用场景奖励很稀疏，智能体需要在环境中做很多无意义的探索。因此未来找到一个有效率的探索方式也是将 DRL 算法应用在现实环境中的关键所在。

• 提高算法的泛化能力。现阶段的 DRL 主要应用于静态、封闭和确定性的环境中，如围棋、游戏等，而现实世界的环境是动态、开放和不确定的。因此，如何提升 DRL 算法在实际应用场景的泛化能力将是一个巨大的挑战。

• 基于模型的学习方法。无模型的 DRL 算法需要大量的采样数据进行训练，而这些数据往往很难通过交互得到。因此可以考虑使用已有的现实环境中的数据建立环境模型，然后利用环境模型对智能体进行训练。

• 在应用方面，DRL 的决策能力可以帮助其应用于各种需要决策的领域，如金融、推荐系统、自动驾驶等。未来对 DRL 的应用研究不仅要考虑对算法的改进，还要考虑将人的智能和机器智能进行结合，采用人工干预来应对环境的不确定性和突发性情况，提高人机交互能力。

参考文献：

[1] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. Cambridge: MIT Press, 2018.

[2] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[3] 赵冬斌, 邵坤, 朱圆恒, 等. 深度强化学习综述: 兼论计算机围棋的发展[J]. 控制理论与应用, 2016, 33(6): 701-717.
ZHAO D B, SHAO K, ZHU Y H, et al. Review of deep reinforcement learning and discussions on the development of computer Go[J]. Control Theory & Applications, 2016, 33(6): 701-717.

[4] 万里鹏, 兰旭光, 张翰博, 等. 深度强化学习理论及其应用综述[J]. 模式识别与人工智能, 2019, 32(1): 67-81.
WAN L P, LAN X G, ZHANG H B, et al. A review of deep reinforcement learning theory and application[J]. Pattern Recognition and Artificial Intelligence, 2019, 32(1): 67-81.

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

[6] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

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

[8] BERNER C, BROCKMAN G, CHAN B, et al. Dota2 with large scale deep reinforcement learning[J]. arXiv preprint, 2019, arXiv:1912. 06680.

[9] VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in StarCraftII using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-354.

[10] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.
LIU Q, ZHAI J W, ZHANG Z Z, et al. A survey on deep reinforcement learning[J]. Chinese Journal of Computers, 2018, 41(1): 1-27.

[11] 刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述[J]. 计算机学报, 2019, 42(6): 1406-1438.
LIU J W, GAO F, LUO X L. Survey of deep reinforcement learning based on value function and policy gradient[J]. Chinese Journal of Computers, 2019, 42(6): 1406-1438.

[12] SUTTON R S. Learning to predict by the methods of temporal differences[J]. Machine Learning, 1988, 3(1): 9-44.

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

[14] VAN HASSELT H, GUEZ A, SILVER D, et al. Deep reinforcement learning with double Q-learning[C]//The 30th AAAI Conference on Artificial Intelligence. [S.l.:s.n.], 2016.

[15] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[C]//The 4th International Conference on Learning Representations. [S.l.:s.n.], 2016.

[16] WANG Z, SCHAUL T, HESSEL M, et al. Dueling network architectures for deep reinforcement learning[C]//The 33rd International Conference on Machine Learning. New York: ACM Press, 2016.

[17] NAIR A, SRINIVASAN P, BLACKWELL S, et al. Massively parallel methods for deep reinforcement learning[J]. arXiv preprint, 2015, arXiv:1507. 04296.

[18] SLIVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms[C]//The 31st International Conference on Machine Learning. New York: ACM Press, 2014.

[19] LILLICRAP P T, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[C]//The 4th International Conference on Learning Representations. [S.l.:s.n.], 2016.

[20] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]//The 33rd International Conference on Machine Learning. New York: ACM Press, 2016.

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

[22] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor[J]. arXiv preprint, 2018, arXiv:1801.01290.

[23] 沈宇, 韩金朋, 李灵犀, 等. 游戏智能中的 AI——从多角色博弈到平行博弈[J]. 智能科学与技术学报, 2020, 2(3): 205-213.
SHEN Y, HAN J P, LI L X, et al. AI in game intelligence—from multi-role game to parallel game[J]. Chinese Journal of Intelligent Science and Technology, 2020, 2(3): 205-213.

[24] BADIA A P, PIOT B, KAPTUROWSKI S, et al. Agent57: outperforming the atari human benchmark[J]. arXiv preprint, 2020, arXiv:2003.13350.

[25] KEMPKA M, WYDMUCH M, RUNC G, et al. Vizdoom: a doom-based AI research platform for visual reinforcement learning[C]//2016 IEEE Conference on Computational Intelligence and Games (CIG). Piscataway: IEEE Press, 2016: 1-8.

[26] LAMPLE G, CHAPLOT D S. Playing FPS games with deep reinforcement learning[C]//The 31st AAAI Conference on Artificial Intelligence. [S.l.:s.n.], 2017.

[27] DOSOVITSKIY A, KOLTUN V. Learning to act by predicting the future[J]. arXiv preprint , 2016, arXiv:1611.01779.

[28] PATHAK D, AGRAWAL P, EFROS A A, et al. Curiosity-driven exploration by self-supervised prediction[C]//The 34th International Conference on Machine Learning. New York: ACM Press, 2017.

[29] WU Y, ZHANG W, SONG K. Master-slave curriculum design for reinforcement learning[C]//The 28th International Joint Conference on Artificial Intelligence. New York: ACM Press, 2018: 1523-1529.

[30] VINYALS O, EWALDS T, BARTUNOV S, et al. StarcraftII: a new challenge for reinforcement learning[J]. arXiv preprint, 2017, arXiv:1708.04782.

[31] ZAMBALDI V, RAPOSO D, SANTORO A, et al. Relational deep reinforcement learning[J]. arXiv preprint, 2018, arXiv:1806. 01830.

[32] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. New York: ACM Press, 2017: 5998-6008.

[33] RASHID T, SAMVELYAN M, DE WITT C S, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning[J]. arXiv preprint, 2018, arXiv:1803.11485.

[34] YE D, LIU Z, SUN M, et al. Mastering complex control in MOBA games with deep reinforcement learning[C]//The 34th AAAI Conference on Artificial Intelligence. [S.l.:s.n.], 2020: 6672-6679.

[35] OH J, CHOCKALINGAM V, SINGH S, et al. Control of memory, active perception, and action in minecraft[C]//The 33rd International Conference on Machine Learning. New York: ACM Press, 2016.

[36] JADERBERG M, MNIH V, CZARNECKI W M, et al. Reinforcement learning with unsupervised auxiliary tasks[J]. arXiv preprint, 2016, arXiv:1611.05397.

[37] MIROWSKI P, PASCANU R, VIOLA F, et al. Learning to navigate in complex environments[J]. arXiv preprint, 2016, arXiv:1611.03673.

[38] WANG Y, HE H, SUN C. Learning to navigate through complex dynamic environment with modular deep reinforcement learning[J]. IEEE Transactions on Games, 2018, 10(4): 400-412.

[39] SHI H, SHI L, XU M, et al. End-to-end navigation strategy with deep reinforcement learning for mobile robots[J]. IEEE Transactions on Industrial Informatics, 2020, 16(4): 2393-2402.

[40] SAVINOV N, RAICHUK A, MARINIER R, et al. Episodic curiosity through reachability[C]//The 7th International Conference on Learning Representations. [S.l.:s.n.], 2019.

[41] ZHU Y, MOTTAGHI R, KOLVE E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning[C]//2017 IEEE international conference on robotics and automation (ICRA). Piscataway: IEEE Press, 2017: 3357-3364.

[42] TAI L, LIU M. Towards cognitive exploration through deep reinforcement learning for mobile robots[J]. arXiv preprint, 2016, arXiv:1610.01733.

[43] TAI L, PAOLO G, LIU M. Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Piscataway: IEEE Press, 2017: 31-36.

[44] WU Y, RAO Z, ZHANG W, et al. Exploring the task cooperation in multi-goal visual navigation[C]//The 28th International Joint Conference on Artificial Intelligence. [S.l.:s.n.], 2019: 609-615.

[45] ZHANG W, ZHANG Y, LIU N. Map-less navigation: a single DRL-based controller for robots with varied dimensions[J]. arXiv preprint, 2020, arXiv:2002.06320.

[46] MIROWSKI P, GRIMES M K, MALINOWSKI M, et al. Learning to navigate in cities without a map[C]//Advances in Neural Information Processing Systems. [S.l.:s.n.], 2018: 2419-2430.

[47] LI A, HU H, MIROWSKI P, et al. Cross-view policy learning for street navigation[C]//The IEEE International Conference on Computer Vision. Piscataway: IEEE Press, 2019: 8100-8109.

[48] HERMANN K M, MALINOWSKI M, MIROWSKI P, et al. Learning to follow directions in street view[C]//The 34th AAAI Conference on Artificial Intelligence. [S.l.:s.n.], 2020.

[49] CHANCÁN M, MILFORD M. CityLearn: diverse real-world environments for sample-efficient navigation policy learning[J]. arXiv preprint, 2020, arXiv:1910.04335.

[50] 孙长银, 穆朝絜. 多智能体深度强化学习的若干关键科学问题[J]. 自动化学报, 2020, 46(7): 1301-1312.
SUN C Y, MU C X. Important scientific problems of multi-agentdeep reinforcement learning[J]. Acta Automatica Sinica, 2020, 46(7): 1301-1312.

[51] OROOJLOOYJADID A, HAJINEZHAD D. A review of cooperative multi-agent deep reinforcement learning[J]. arXiv preprint, 2019, arXiv:1908.03963.

[52] OMIDSHAFIEI S, PAZIS J, AMATO C, et al. Deep decentralized multi-task multi-agent reinforcement learning under partial observability[C]//The 34th International Conference on Machine Learning. New York: ACM Press, 2017.

[53] MATIGNON L, LAURENT G J, LE FORT-PIAT N. Hysteretic Q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams[C]//2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE Press, 2007: 64-69.

[54] FOERSTER J, NARDELLI N, FARQUHAR G, et al. Stabilising experience replay for deep multi-agent reinforcement learning[C]//The 34th International Conference on Machine Learning. New York: ACM Press, 2017.

[55] PALMER G, TUYLS K, BLOEMBERGEN D, et al. Lenient multi-agent deep reinforcement learning[C]//The 17th International Conference on Autonomous agents and Multiagent Systems. New York: ACM Press, 2018.

[56] EVERETT R, ROBERTS S. Learning against non-stationary agents with opponent modelling and deep reinforcement learning[C]//2018 AAAI Spring Symposium Series. [S.l.:s.n.], 2018.

[57] JIN Y, WEI S, YUAN J, et al. Stabilizing multi-agent deep reinforcement learning by implicitly estimating other agents' behaviors[C]// 2020 IEEE International Conference on Acoustics, Speech and Signal

Processing. Piscataway: IEEE Press, 2020: 3547-3551.

[58] LIU X, TAN Y. Attentive relational state representation in decentralized multiagent reinforcement learning[J]. IEEE Transactions on Cybernetics, 2020.

[59] GUPTA J K, EGOROV M, KOCHENDERFER M. Cooperative multi-agent control using deep reinforcement learning[C]//The 16th International Conference on Autonomous Agents and Multiagent Systems. Cham: Springer, 2017: 66-83.

[60] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]//Advances in Neural Information Processing Systems. New York: ACM Press, 2017: 6379-6390.

[61] FOERSTER J, FARQUHAR G, AFOURAS T, et al. Counterfactual multi-agent policy gradients[C]//The 32nd AAAI Conferenceon Artificial Intelligence. [S.l.:s.n.], 2018.

[62] SUNEHAG P, LEVER G, GRUSLYS A, et al. Value-decomposition networks for cooperative multi-agent learning[J]. arXiv preprint, 2017, arXiv:1706. 05296.

[63] MAO H, ZHANG Z, XIAO Z, et al. Modelling the dynamic joint policy of teammates with attention multi-agent DDPG[C]//The 18th International Conference on Autonomous Agentsand Multiagent Systems. New York: ACM Press, 2019.

[64] IQBAL S, SHA F. Actor-attention-critic for multi-agent reinforcement learning[C]//International Conference on Machine Learning. [S.l.:s.n.], 2019: 2961-2970.

[65] FOERSTER J N, ASSAEL Y M, DE FREITAS N, et al. Learning to communicate with deep multi-agent reinforcement learning[C]//Advances in Neural Information Processing Systems. New York: ACM Press, 2016: 2137-2145.

[66] SUKHBAATAR S, SZLAM A, FERGUS R. Learning multiagent communication with back propagation[C]//Advances in Neural Information Processing Systems. New York: ACM Press, 2016: 2244-2252.

[67] JIANG J, LU Z. Learning attentional communication for multi-agent cooperation[C]//Advances in Neural Information Processing Systems. New York: ACM Press, 2018: 7254-7264.

[68] KIM D, MOON S, HOSTALLERO D, et al. Learning to schedule communication in multi-agent reinforcement learning[C]//The 7th International Conference on Learning Representations. [S.l.:s.n.], 2019.

[69] DAS A, GERVET T, ROMOFF J, et al. TarMAC: targeted multi-agent communication[C]//The 36th International Conference on Machine-Learning. [S.l.:s.n.], 2019.

[70] SHANI G, HECKERMAN D, BRAFMAN R I, et al. An MDP-based recommender system[J]. Journal of Machine Learning Research, 2005, 6(Sep): 1265-1295.

[71] ZHAO X, XIA L, TANG J, et al. Deep reinforcement learning for search, recommendation, and online advertising: a survey[J]. ACM SIGWEB Newsletter, 2019 (Spring): 1-15.

[72] ZHAO X, XIA L, ZHANG L, et al. Deep reinforcement learning for page-wise recommendations[C]//The 12th ACM Conference on Recommender Systems. New York: ACM Press, 2018: 95-103.

[73] ZHENG G, ZHANG F, ZHENG Z, et al. DRN: a deep reinforcement learning framework for news recommendation[C]//The 2018 World Wide Web Conference. New York: ACM Press, 2018: 167-176.

[作者简介]

刘朝阳（1996− ），男，天津大学电气自动化与信息工程学院博士生，主要研究方向为强化学习、多智能体强化学习。

穆朝絮（1984− ），女，博士，天津大学电气自动化与信息工程学院教授，主要研究方向为强化学习、自适应学习系统、非线性控制和优化。

孙长银（1975−），男，博士，东南大学自动化学院教授，中国自动化学会会士，中国自动化学会人工智能与机器人教育专业委员会主任。主要研究方向为智能控制与优化、强化学习、神经网络、数据驱动控制。担任 *IEEE Transactions on Neural Networks and Learning Systems*、*IEEE/CAA Journal of Automatica Sinica*、《自动化学报》《控制理论与应用》《智能科学与技术学报》等高质量学术期刊编委。2011 年获得国家杰出青年科学基金。“智能机器人感知与控制”江苏高等学校优秀科技创新团队带头人，2016 年全国优秀科技工作者，第三批国家“万人计划”科技创新领军人才，中国科学技术协会第九次全国代表大会代表，“自主无人系统协同控制理论及应用”国家自然科学基金委员会创新研究群体学术带头人，科学技术部科技创新 2030—“新一代人工智能”重大项目“人在回路的混合增强智能”首席科学家，江苏省前沿引领技术基础研究专项领衔科学家。