深度强化学习综述

刘 全[+] 翟建伟 章宗长 钟珊 周 倩 章 鹏 徐 进

[1)](苏州大学计算机科学与技术学院 江苏 苏州 215006)

[2)](软件新技术与产业化协同创新中心 南京 210000)

摘 要 深度强化学习是人工智能领域的一个新的研究热点．它以一种通用的形式将深度学习的感知能力与强化学习的决策能力相结合，并能够通过端对端的学习方式实现从原始输入到输出的直接控制．自提出以来，在许多需要感知高维度原始输入数据和决策控制的任务中，深度强化学习方法已经取得了实质性的突破．该文首先阐述了 3 类主要的深度强化学习方法，包括基于值函数的深度强化学习、基于策略梯度的深度强化学习和基于搜索与监督的深度强化学习；其次对深度强化学习领域的一些前沿研究方向进行了综述，包括分层深度强化学习、多任务迁移深度强化学习、多智能体深度强化学习、基于记忆与推理的深度强化学习等．最后总结了深度强化学习在若干领域的成功应用和未来发展趋势．

关键词 人工智能；深度学习；强化学习；深度强化学习

中图法分类号 TP18

论文引用格式：

刘全,翟建伟,章宗长,钟珊,周倩,章鹏,徐进,深度强化学习综述,2017，Vol.40,在线出版号 No.1

LIU Quan, ZHAI Jian-Wei, ZHANG Zong-Zhang, ZHONG Shan, ZHOU Qian, ZHANG Peng, XU Jin, A Survey on Deep Reinforcement Learning, 2017,Vol.40,Online Publishing No.1

A Survey on Deep Reinforcement Learning

LIU Quan ZHAI Jian-Wei ZHANG Zong-Zhang ZHONG Shan ZHOU Qian ZHANG Peng XU Jin

[1)](School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006)

[2)](Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210000)

Abstract Deep reinforcement learning (DRL) is a new research hotspot in the artificial intelligence community. By using a general-purpose form, DRL integrates the advantages of the perception of deep learning (DL) and the decision making of reinforcement learning (RL), and gains the output control directly based on raw inputs by the end-to-end learning process. DRL has made substantial breakthroughs in a variety of tasks requiring both rich perception of high-dimensional raw inputs and policy control since it was proposed. In this paper, we systematically describe three main categories of DRL methods. Firstly, we summarize value-based DRL methods. The core idea behind them is to approximate the value function by using deep neural networks which have strong ability of perception. We introduce an epoch-making value-based DRL method called Deep Q-Network (DQN) and its variants. These variants are divided into two categories: improvements of training algorithm and improvements of model architecture. The first category includes Deep Double Q-Network (DDQN), DQN based on advantage learning technique, and DDQN with proportional prioritization. The second one includes Deep Recurrent Q-Network (DRQN) and a method based on Dueling Network architecture. In general, value-based DRL methods are good at dealing with large-scale problems with discrete action spaces. We then summarize policy-based DRL methods. Their powerful idea is to use deep neural networks to parameterize the policies and

本课题得到国家自然科学基金项目(61472262, 61303108, 61373094, 61502323, 61502329)、苏州市应用基础研究计划工业部分（SYG201422, SYG201308）资助．刘全（**通讯作者**），男，1969 年生，博士，教授，博士生导师，中国计算机协会(CCF)高级会员，主要研究方向为强化学习、深度强化学习和自动推理.E-mail:quanliu@suda.edu.cn. 翟建伟，男，1992 年生，硕士研究生，主要研究方向为强化学习、深度学习和深度强化学习．章宗长，男，1985 年生，博士，副教授，计算机学会会员，主要研究领域为部分感知的马尔可夫决策过程、强化学习和多 agent 系统．钟珊，女，1983 年生，博士研究生，主要研究方向为机器学习和深度学习．周倩，女，1992 年生，硕士研究生，主要研究方向为强化学习.章鹏，男，1992 年生，硕士研究生，主要研究方向为连续空间强化学习．徐进，男，1991 年生，硕士研究生，主要研究方向为连续空间深度强化学习．

optimization methods to optimize the policies. In this part, we firstly highlight some pure policy gradient methods, then focus on a series of policy-based DRL algorithms which use the actor-critic framework e.g., Deep Deterministic Policy Gradient (DDPG), followed by an effective method named Asynchronous Advantage Actor-Critic (A3C) with the benefit of reducing the training time dramatically. Compared to value-based methods, policy-based DRL methods have a wider range of successful applications in complex problems with continuous action spaces. We lastly introduce a DRL method based on search and supervision known as AlphaGo. Its core idea is to improve the efficiency of optimizing policies by introducing extra supervision and policy search techniques. Then this paper summarizes some cutting-edge research directions of DRL, including hierarchical DRL methods which can decompose an ultimate goal in RL into some sub-goals, multi-task and transfer DRL methods which can take full advantage of correlations between multiple tasks and transfer useful information to new tasks, multi-agent DRL methods which have the ability of cooperation and communication between multiple agents, DRL based on memory and reasoning which can be applied to some high-level cognitive heuristic tasks, and methods that balance between exploration and exploitation; Next, we summarize some successful applications in different fields such as games, robotics, computer vision, natural language processing and parameter optimization. Finally, we end up with discussing some potential trends in DRL's future development.

Keywords artificial intelligence; deep learning; reinforcement learning; deep reinforcement learning

1 引言

近年来，深度学习（Deep Learning, DL）作为机器学习领域一个重要的研究热点[1]，已经在图像分析[2-3]、语音识别[4-5]、自然语言处理[6-7]、视频分类[8]等领域取得了令人瞩目的成功. DL 的基本思想是通过多层的网络结构和非线性变换，组合低层特征，形成抽象的、易于区分的高层表示，以发现数据的分布式特征表示[9]. 因此 DL 方法侧重于对事物的感知和表达. 强化学习（Reinforcement Learning，RL）作为机器学习领域另一个研究热点，已经广泛应用于工业制造[10]、仿真模拟[11]、机器人控制[12]、优化与调度[13-14]、游戏博弈[15-16]等领域. RL 的基本思想是通过最大化智能体（agent）从环境中获得的累计奖赏值，以学习到完成目标的最优策略[17]. 因此 RL 方法更加侧重于学习解决问题的策略. 随着人类社会的飞速发展，在越来越多复杂的现实场景任务中，需要利用 DL 来自动学习大规模输入数据的抽象表征，并以此表征为依据进行自我激励的 RL，优化解决问题的策略. 由此，谷歌的人工智能研究团队 DeepMind 创新性地将具有感知能力的 DL 和具有决策能力的 RL 相结合，形成了人工智能领域新的研究热点，即深度强化学习（Deep Reinforcement Learning，DRL）. 此后，在很多挑战性领域中，DeepMind 团队构造并实现了人类专家级别的 agent. 这些 agent 对自身知识的构建和学习都直接来自原始输入信号，无需任何的人工编码和领域知识. 因此 DRL 是一种端对端（end-to-end）的感知与控制系统，具有很强的通用性. 其学习过程可以描述为：（1）在每个时刻 agent 与环境交互得到一个高维度的观察，并利用 DL 方法来感知观察，以得到抽象、具体的状态特征表示；（2）基于预期回报来评价各动作的价值函数，并通过某种策略将当前状态映射为相应的动作. （3）环境对此动作做出反应，并得到下一个观察. 通过不断循环以上过程，最终可以得到实现目标的最优策略. DRL 原理框架如图 1 所示.

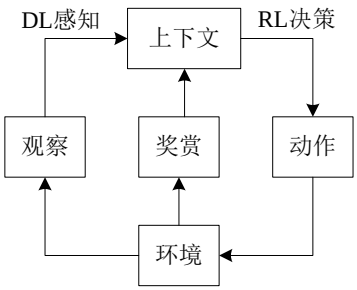

图 1 DRL 原理框架图

目前 DRL 技术在游戏[18-20]、机器人控制[21-23]、参数优化[24-25]、机器视觉[26-27]等领域中得到了广泛的应用，并被认为是迈向通用人工智能（Artificial General Intelligence，AGI）的重要途径. 本文对 DRL 的研究历程和发展现状进行了详细的阐述. 整体架构如图 2 所示.

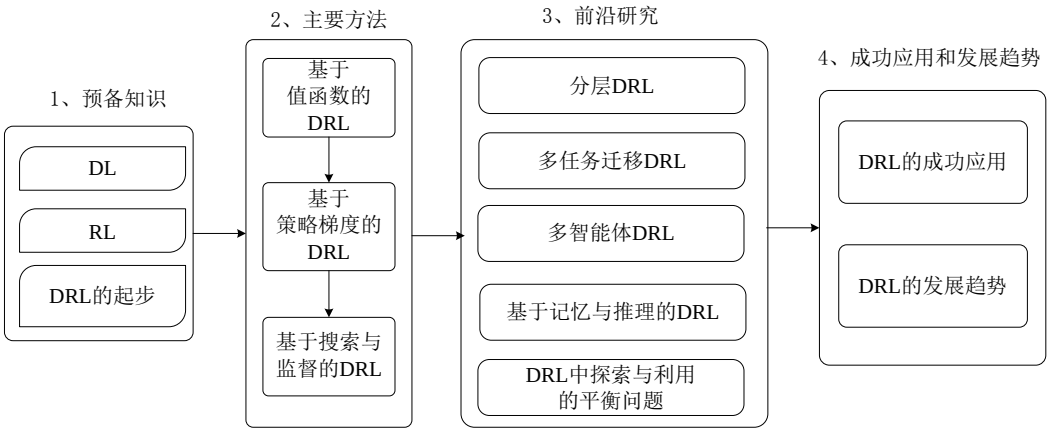

图 2 本文的整体架构

2 预备知识

2.1 深度学习

DL 的概念源于人工神经网络（Artificial Neural Network，ANN）．含多隐藏层的多层感知器（Multi-Layer Perceptron，MLP）是 DL 模型的一个典型范例．DL 模型通常由多层的非线性运算单元组合而成．其将较低层的输出作为更高一层的输入，通过这种方式自动地从大量训练数据中学习抽象的特征表示，以发现数据的分布式特征[28]．与浅层网络相比，传统的多隐藏层网络模型有更好的特征表达能力，但由于计算能力不足、训练数据缺乏、梯度弥散等原因，使其一直无法取得突破性进展．直到 2006 年，深度神经网络的研究迎来了转机．Hinton 等人[29]提出了一种训练深层神经网络的基本原则：先用非监督学习对网络逐层进行贪婪的预训练，再用监督学习对整个网络进行微调．这种预训练的方式为深度神经网络提供了较理想的初始参数，降低了深度神经网络的优化难度．此后几年，各种 DL 模型被相继提出．包括堆栈式自动编码器[30-31]（Stacked Auto-Encoder，SAE）、限制玻尔兹曼机[32-33]（Restricted Boltzmann Machine，RBM）、深度信念网络[33-34]（Deep Belief Network，DBN）、循环神经网络[35-36]（Recurrent Neural Network，RNN）等．

随着训练数据的增长和计算能力的提升，卷积神经网络（Convolutional Neural Network，CNN）开始在各领域中得到广泛应用．Krizhevsky 等人[2]在 2012 年提出了一种称为 AlexNet 的深度卷积神经网络，并在当年的 ImageNet 图像分类竞赛中，大幅度降低了图像识别的 top-5 错误率．此后，卷积神经网络朝着以下 4 个方向迅速发展：

（1）增加网络的层数．在 2014 年，视觉几何组（Visual Geometry Group，VGG）的 Simonyan 等人[37]提出了 VGG-Net 模型，进一步降低了图像识别的错误率．He 等人[38]提出了一种扩展深度卷积神经网络的高效方法；

（2）增加卷积模块的功能．Lin 等人[39]利用多层感知卷积层替代传统的卷积操作，提出了一种称为 Network in Network（NIN）的深度卷积网络模型．Szegedy 等人[40]在现有网络模型中加入一种新颖的 Inception 结构，提出了 NIN 的改进版本 GoogleNet，并在 2014 年取得了 ILSVRC 物体检测的冠军；

（3）增加网络层数和卷积模块功能．He 等人[41]提出了深度残差网络（Deep Residual Network，DRN），并在 2015 年取得了 ILSVRC 物体检测和物体识别的双料冠军．Szegedy 等人[42]进一步将 Inception 结构与 DRN 相结合，提出了基于 Inception 结构的深度残差网络（Inception Residual Network，IRN）．此后，He 等人[43]提出了恒等映射的深度残差网络（Identify Mapping Residual Network，IMRN），进一步提升了物体检测和物体识别的准确率；

（4）增加新的网络模块．向卷积神经网络中加入循环神经网络[44]（Recurrent Neural Network，RNN）、注意力机制[45]（Attention Mechanism，AM）等结构．

2.2 强化学习

RL 是一种从环境状态映射到动作的学习，目

标是使 agent 在与环境的交互过程中获得最大的累积奖赏[17]. 马尔可夫决策过程（Markov Decision Process，MDP）可以用来对 RL 问题进行建模. 通常将 MDP 定义为一个四元组(S, A, ρ, f)，其中：

（1）S 为所有环境状态的集合. $s_t \in S$ 表示 agent 在 t 时刻所处的状态；

（2）A 为 agent 可执行动作的集合. $a_t \in A$ 表示 agent 在 t 时刻所采取的动作；

（3）$\rho: S \times A \to R$ 为奖赏函数. $r_t \sim \rho(s_t, a_t)$ 表示 agent 在状态 s_t 执行动作 a_t 获得的立即奖赏值；

（4）$f: S \times A \times S \to [0,1]$ 为状态转移概率分布函数. $s_{t+1} \sim f(s_t, a_t)$ 表示 agent 在状态 s_t 执行动作 a_t 转移到下一状态 s_{t+1} 的概率.

在 RL 中，策略 $\pi: S \to A$ 是状态空间到动作空间的一个映射. 表示为 agent 在状态 s_t 选择动作 a_t，执行该动作并以概率 $f(s_t, a_t)$ 转移到下一状态 s_{t+1}，同时接受来自环境反馈的奖赏 r_t. 假设未来每个时间步所获的立即奖赏都必须乘以一个折扣因子 γ，则从 t 时刻开始到 T 时刻情节结束时，奖赏之和定义为：

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \tag{1}$$

其中 $\gamma \in [0,1]$，用来权衡未来奖赏对累积奖赏的影响.

状态动作值函数 $Q^{\pi}(s,a)$ 指的是在当前状态 s 下执行动作 a，并一直遵循策略 π 到情节结束，这一过程中 agent 所获得的累积回报表示为：

$$Q^{\pi}(s,a) = \mathrm{E}\left[R_t \middle| s_t = s, a_t = a, \pi\right] \tag{2}$$

对于所有的状态动作对，如果一个策略 π^* 的期望回报大于或等于其他所有策略的期望回报，那么称策略 π^* 为最优策略. 最优策略可能不只一个，但它们共享一个状态动作值函数：

$$Q^*(s,a) = \max_{\pi} \mathrm{E}\left[R_t \middle| s_t = s, a_t = a, \pi\right] \tag{3}$$

式（3）被称为最优状态动作值函数，且最优状态动作值函数遵循贝尔曼最优方程（Bellman optimality equation）. 即：

$$Q^*(s,a) = \mathrm{E}_{s' \sim S}\left[r + \gamma \max_{a'} Q(s',a') \middle| s, a\right] \tag{4}$$

在传统的 RL 中，一般通过迭代贝尔曼方程求解 Q 值函数：

$$Q_{i+1}(s,a) = \mathrm{E}_{s' \sim S}\left[r + \gamma \max_{a'} Q_i(s',a') \middle| s, a\right] \tag{5}$$

其中，当 $i \to \infty$ 时，$Q_i \to Q^*$. 即通过不断地迭代会使状态动作值函数最终收敛，从而得到最优策略：$\pi^* = \mathrm{argmax}_{a \in A} Q^*(s,a)$. 然而对于实际问题来说，通过迭代式（5）求解最优策略显然是不可行的，因为在大状态空间下，用迭代贝尔曼方程求解 Q 值函数的方法计算代价太大. 针对此问题，在 RL 算法中，通常使用线性函数逼近器来近似表示状态动作值函数，即 $Q(s,a|\theta) \approx Q^*(s,a)$. 此外，也可以用深度神经网络等非线性函数逼近器去近似表示值函数或策略. 然而将 RL 与深度神经网络相结合可能会出现算法不稳定等问题[46]，这一直阻碍着 DRL 的发展与应用.

2.3 深度强化学习的起步

DRL 兴起之前已经开展了一些前期工作，但由于训练数据和计算能力的欠缺，这些工作仅利用深度神经网络对高维度输入数据降维，以便于传统的 RL 算法对其进行处理. Riedmiller 等人[47]最先使用一个多层感知器来近似表示 Q 值函数，并提出了神经拟合 Q 迭代（Neural Fitted Q Iteration，NFQ）算法. Lange 等人[48]结合 DL 模型和 RL 方法，提出了一种深度自动编码器（Deep Auto-Encoder，DAE）模型. 然而 DAE 只适用于以视觉感知为输入信号且状态空间维度较小的控制问题. Abtahi 等人[49]用深度信念网络作为传统 RL 中的函数逼近器，极大地提高了 agent 的学习效率，并成功地应用于车牌图像字符分割任务中. Lange 等人[50]又进一步提出了深度拟合 Q 学习算法（Deep Fitted Q-Learning，DFQ），并将该算法应用于车辆控制中. Koutnik 等人[51]将神经演化（Neural Evolution，NE）方法与 RL 算法相结合，应用于一款视频赛车游戏中，实现了对赛车的自动驾驶.

3 基于值函数的深度强化学习

3.1 深度 Q 网络

Mnih 等人[18-19]将卷积神经网络与传统 RL 中的 Q 学习[52]算法相结合，提出了深度 Q 网络（Deep Q-Network, DQN）模型. 该模型用于处理基于视觉感知的控制任务，是 DRL 领域的开创性工作.

3.1.1 模型结构

DQN 模型的输入是距离当前时刻最近的 4 幅预处理后的图像. 该输入经过 3 个卷积层和 2 个全

连接层的非线性变换，最终在输出层产生每个动作的 Q 值．图 3 表示 DQN 的模型架构．

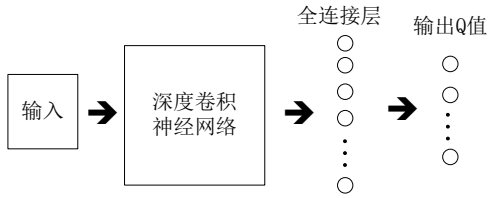

图 3　DQN 的模型结构

3.1.2　训练算法

图 4 描述了 DQN 的训练流程．为缓解非线性网络表示值函数时出现的不稳定等问题，DQN 主要对传统的 Q 学习算法做了 3 处改进．

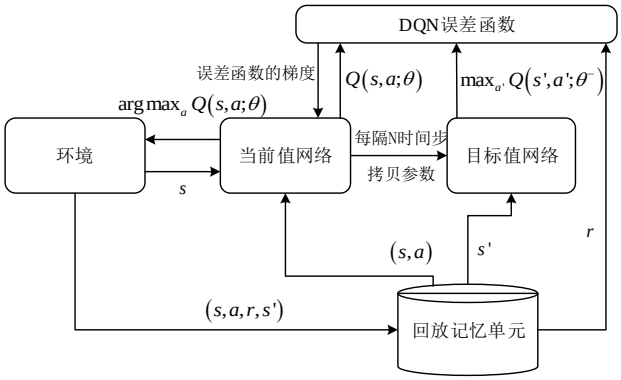

图 4　DQN 的训练流程

（1）DQN 在训练过程中使用经验回放机制[53]（experience replay），在线处理得到的转移样本 $e_t=(s_t,a_t,r_t,s_{t+1})$．在每个时间步 t，将 agent 与环境交互得到的转移样本存储到回放记忆单元 $D=\{e_1,\ldots e_t\}$ 中．训练时，每次从 D 中随机抽取小批量的转移样本，并使用随机梯度下降（Stochastic Gradient Descent, SGD）算法更新网络参数 θ．在训练深度网络时，通常要求样本之间是相互独立的．这种随机采样的方式，大大降低了样本之间的关联性，从而提升了算法的稳定性．

（2）DQN 除了使用深度卷积网络近似表示当前的值函数之外，还单独使用了另一个网络来产生目标 Q 值．具体地，$Q(s,a|\theta_i)$ 表示当前值网络的输出，用来评估当前状态动作对的值函数；$Q(s,a|\theta_i^-)$ 表示目标值网络的输出，一般采用 $Y_i=r+\gamma\max_{a'}Q(s',a'|\theta_i^-)$ 近似表示值函数的优化目标，即目标 Q 值．当前值网络的参数 θ 是实时更新的，每经过 N 轮迭代，将当前值网络的参数复制给目标值网络．通过最小化当前 Q 值和目标 Q 值之间的均方误差来更新网络参数．误差函数为：

$$L(\theta_i)=\mathrm{E}_{s,a,r,s'}\left[(Y_i-Q(s,a|\theta_i))^2\right] \quad (6)$$

对参数 θ 求偏导，得到以下梯度：

$$\nabla_{\theta_i}L(\theta_i)=\mathrm{E}_{s,a,r,s'}\left[\left(Y_i-Q(s,a|\theta_i)\right)\nabla_{\theta_i}Q(s,a|\theta_i)\right] \quad (7)$$

引入目标值网络后，在一段时间内目标 Q 值是保持不变的，一定程度上降低了当前 Q 值和目标 Q 值之间的相关性，提升了算法的稳定性．

（3）DQN 将奖赏值和误差项缩小到有限的区间内，保证了 Q 值和梯度值都处于合理的范围内，提高了算法的稳定性．实验表明，DQN 在解决诸如 Atari 2600 游戏等类真实环境的复杂问题时，表现出与人类玩家相媲美的竞技水平[19]，甚至在一些难度较低的非战略性游戏中，DQN 的表现超过了有经验的人类玩家．在解决各类基于视觉感知的 DRL 任务时，DQN 使用了同一套网络模型、参数设置和训练算法，这充分说明 DQN 方法具有很强的适应性和通用性．

3.2　深度 Q 网络训练算法的改进

3.2.1　深度双 Q 网络

在 DQN 中使用 $Y_i=r+\gamma\max_{a'}Q(s',a'|\theta_i^-)$ 近似表示值函数的优化目标时，每次都选取下一个状态中最大 Q 值所对应的动作．选择和评价动作都是基于目标值网络的参数 θ^-，这会引起在学习过程中出现过高估计 Q 值的问题．

Hasselt 等人[54]基于双 Q 学习算法[55]（double Q-learning），提出了深度双 Q 网络（Deep Double Q-Network，DDQN）算法．在双 Q 学习中有两套不同的参数：θ 和 θ^-．其中 θ 用来选择对应最大 Q 值的动作，θ^- 用来评估最优动作的 Q 值．两套参数将动作选择和策略评估分离开，降低了过高估计 Q 值的风险．因此 DDQN 使用当前值网络的参数 θ 来选择最优动作，使用目标值网络的参数 θ^- 来评估该最优动作．目标 Q 值的形式如下：

$$Y_i^{DDQN}=r+\gamma Q\left(x',\arg\max_a Q(s',a|\theta_i),\theta_i^-\right) \quad (8)$$

DDQN 在其他方面都与 DQN 保持一致．实验表明，DDQN 能够估计出更加准确的 Q 值，在一些 Atari 2600 游戏中可获得更稳定有效的策略．

3.2.2　基于优势学习的深度 Q 网络

根据 3.2.1 节可知，降低 Q 值的评估误差可以提升性能．Bellemare 等人[56]在贝尔曼方程中定义新的操作符，来增大最优动作值和次优动作值之间的差异，以缓和每次都选取下一状态中最大 Q 值对应动作所带来的评估误差．具体的改进如下：

基于采样得到的样本计算均方误差 $\Delta Q(s,a)^2$，其中误差项定义为：

$$\Delta Q(s,a)=r+\gamma V(s')-Q(s,a) \tag{9}$$

根据优势学习[57]（Advantage Learning，AL）定义两种新的操作符，并将这两种操作符运用到上式中，分别得到 AL 误差项和一致性优势学习（Persistent Advantage Learning，PAL）误差项．其中 AL 误差项定义为：

$$\Delta_{\mathrm{AL}}Q(s,a)=\Delta Q(s,a)-\alpha[V(s)-Q(s,a)] \tag{10}$$

为了定义 PAL 误差项，构造出下式：

$$\Delta_{\mathrm{AL}}Q'(s,a)=\Delta Q(s,a)-\alpha[V(s')-Q(s',a)] \tag{11}$$

得到 PAL 误差项的具体形式：

$$\Delta_{\mathrm{PAL}}Q(s,a)=\max\{\Delta_{\mathrm{AL}}Q(s,a),\Delta_{\mathrm{AL}}Q'(s,a)\} \tag{12}$$

实验表明，用 AL 和 PAL 误差项来替代贝尔曼方程中的误差项，可以有效地增加最优和次优动作对应值函数之间的差异，从而获得更加精确的 Q 值．即在 DQN 中加入 AL 和 PAL 误差项，可以有效地减小评估 Q 值时的偏差，促进学习效果的进一步提升，在许多 Atari 2600 游戏中取得了更好的表现．其中，采用 PAL 误差项时，最优和次优动作对应值函数之间的差异更大，Q 值的评估也更加精确．

3.2.3 基于优先级采样的深度 Q 网络

DQN 为了消除转移样本 $e_t=(s_t,a_t,r_t,s_{t+1})$ 之间的相关性，使用经验回放机制在线地存储和使用 agent 与环境交互得到的历史样本．在每个时刻，经验回放机制从样本池中等概率地抽取小批量的样本用于训练．然而等概率采样并不能区分不同样本的重要性，同时由于样本池 D 的存储量有限，某些样本还未被充分利用就已经被舍弃．针对该问题，Schaul 等人[58]在 DDQN 的基础上提出了一种基于比例优先级采样的深度双 Q 网络（double deep Q-network with proportional prioritization）．该方法用基于优先级的采样方式来替代均匀采样，提高一些有价值样本的采样概率，从而加快最优策略的学习．具体的改进如下：

该抽样方法将每个样本的时间差分（Temporal Difference, TD）误差项作为评价优先级的标准．该误差为：$r+\gamma\max_{a'}Q(s',a'|\theta^-)-Q(s,a|\theta)$，并且其绝对值越大，对应样本被采样的概率越高．在抽样过程中该方法使用随机比例化（stochastic prioritization）和重要性采样权重（importance-sampling weights）两种技巧．其中，随机比例化操作不仅能充分利用较大 TD 误差项对应的样本，而且保证了抽取样本的多样性．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．实验表明，基于该抽样方式的深度双 Q 网络可以提升训练速度，并在很多 Atari 2600 游戏中获得了更高的分数．

另外，Lakshminarayanan 等人[59]使用动态跳帧的方式来替代 DQN 中每个时刻重复 k 次的动作，提出了动态跳帧的 DQN（Dynamic Frame Skip Deep Q-Network，DFDQN）算法．实验表明，DFDQN 在一些 Atari 2600 游戏中取得了更好的性能；Hasselt 等人[60]使用一种称为 Pop-Art 的动态归一化操作来替代传统 DQN 中的区间裁剪方法．在不流失重要状态信息的前提下，统一了不同任务中目标 Q 值的量级，提高了 agent 在很多 Atari 2600 游戏中的表现；Vincent 等人[61]在 DQN 中使用自适应的折扣因子和学习率，加速了深度网络收敛的速度．

3.3 DQN 模型结构的改进

对 DQN 模型的改进一般是通过向原有网络中添加新的功能模块来实现的．例如，可以向 DQN 模型中加入循环神经网络结构，使得模型拥有时间轴上的记忆能力．本节主要介绍两种重要的 DQN 模型的改进版本，分别是基于竞争架构的 DQN 和深度循环 Q 网络（Deep Recurrent Q-Network，DRQN）．

3.3.1 基于竞争架构的 DQN

在很多基于视觉感知的 DRL 任务中，受不同动作的影响，状态动作对的值函数是不同的．然而在某些状态下，值函数的大小是与动作无关的．利用上述思想，Wang 等人[62]设计了一种竞争网络结构（dueling network），并将其加入到 DQN 网络模

型中. 如图 5，该网络结构与 DQN 模型的不同之处在于：DQN 将 CNN 提取的抽象特征经过全连接层后，直接在输出层输出对应动作的 Q 值，而引入竞争网络结构的模型则将 CNN 提取的抽象特征分流到两个支路中，其中一路代表状态值函数，另一路代表依赖状态的动作优势函数（advantage function）. 通过该种竞争网络结构，agent 可以在策略评估过程中更快地识别出正确的行为.

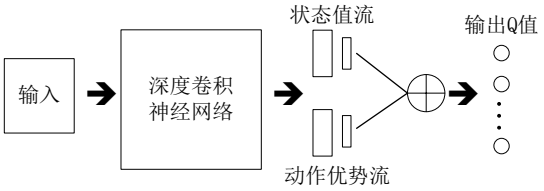

图 5 基于竞争架构的 DQN 模型结构

具体地，状态值函数表示为$\hat{V}(s|\theta,\beta)$，动作优势函数表示为$\hat{A}(s,a|\theta,\alpha)$. 通过一种聚合操作将状态值流和动作优势流相结合：

$$Q(s,a|\theta,\alpha,\beta)=\hat{V}(s|\theta,\beta)+\hat{A}(s,a|\theta,\alpha) \quad (13)$$

其中，α、β和θ分别代表状态值流、动作优势流和模型剩余部件的参数. 然而在实际操作中，一般要将动作优势流设置为单独动作优势函数值减去某状态下所有动作优势函数的平均值. 该技巧不仅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. 实验表明，在 DQN 中加入竞争网络可以使得值函数的估计更加精确. 在频繁出现 agent 采取不同动作但对应值函数相等的情形下，竞争架构的 DQN 模型性能提升最为明显.

3.3.2 深度循环 Q 网络

在传统的 RL 方法中，状态信息的部分可观察性一直是个亟待解决的难题. DQN 通过堆叠离当前时刻最近的 4 幅历史图像组成输入状态，有效缓解了状态信息的部分可观察问题，却增加了网络的计算和存储负担. 针对此问题，Hausknecht 等人[63]利用循环神经网络结构来记忆时间轴上连续的历史状态信息，提出了 DRQN 模型.

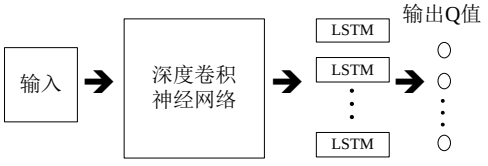

图 6 DRQN 模型结构

如图 6 所示，DRQN 将 DQN 中第 1 个全连接层的部件替换成了 256 个长短期记忆单元（Long Short-Term Memory，LSTM）. 此时模型的输入仅为当前时刻的一幅图像，减少了深度网络感知图像特征所耗费的计算资源. 实验表明，在部分状态可观察的情况下，DRQN 表现出比 DQN 更好的性能. 因此 DRQN 模型适用于普遍存在部分状态可观察问题的复杂任务.

随着 DL 领域中各种新颖网络模块的提出，未来 DRL 模型会朝着结构多样化、模块复杂化的方向发展. 例如，可以利用深度残差网络所具备的强大感知能力来提高 agent 对复杂状态空间的表征效果；另外，可以在模型中加入视觉注意力机制[45]（Visual Attention Mechanism，VAM），使得 agent 在不同状态下将注意力集中到有利于做出决策的区域，从而加速学习的进程.

4 基于策略梯度的深度强化学习

策略梯度是一种常用的策略优化方法，它通过不断计算策略期望总奖赏关于策略参数的梯度来更新策略参数，最终收敛于最优策略[64]. 因此在解决 DRL 问题时，可以采用参数为θ的深度神经网络来进行参数化表示策略，并利用策略梯度方法来优化策略. 值得注意的是，在求解 DRL 问题时，往往第一选择是采取基于策略梯度的算法. 原因是它能够直接优化策略的期望总奖赏，并以端对端的方式直接在策略空间中搜索最优策略，省去了繁琐的中间环节. 因此与 DQN 及其改进模型相比，基于策略梯度的 DRL 方法适用范围更广，策略优化的效果也更好.

4.1 深度策略梯度的起源与发展

策略梯度方法是一种直接使用逼近器来近似表示和优化策略，最终得到最优策略的方法. 该方法优化的是策略的期望总奖赏：

$$\max_{\theta} \mathrm{E}\left[R \mid \pi_{\theta}\right] \quad (14)$$

其中$R=\sum_{t=0}^{T-1} r_t$表示一个情节内所获得的奖赏总和. 策略梯度最常见的思想是增加总奖赏较高情节出现的概率. 策略梯度方法的具体过程如下：

假设一个完整情节的状态、动作和奖赏的轨迹为：$\tau=(s_0,a_0,r_0,s_1,a_1,r_1,\ldots,s_{T-1},a_{T-1},r_{T-1},s_T)$. 则策

略梯度表示为如下的形式：

$$g = R\nabla_\theta \sum_{t=0}^{T-1} \log\pi\left(a_t|s_t;\theta\right) \tag{15}$$

利用该梯度调整策略参数：

$$\theta \leftarrow \theta + \alpha g \tag{16}$$

其中，α 是学习率，控制着策略参数更新的速率. 式（15）中的 $\nabla_\theta \sum_{t=0}^{T-1} \log\pi\left(a_t|s_t;\theta\right)$ 梯度项表示能够提高轨迹 τ 出现概率的方向，乘上得分函数 R 之后，可以使得单个情节内总奖赏越高的 τ 越“用力拉拢”概率密度. 即如果收集了很多总奖赏不同的轨迹，通过上述训练过程会使得概率密度向总奖赏更高的轨迹方向移动，最大化高奖赏轨迹 τ 出现的概率.

然而在某些情形下，每个情节的总奖赏 R 都不为负，那么所有梯度 g 的值也都是大于等于 0 的. 此时在训练过程中遇到每个轨迹 τ，都会使概率密度向正的方向“拉拢”，很大程度减缓了学习速度. 这会使得梯度 g 的方差很大. 因此可以对 R 使用某种标准化操作来降低梯度 g 的方差. 该技巧使得算法能提高总奖赏 R 较大的轨迹 τ 的出现概率，同时降低总奖赏 R 较小的轨迹 τ 的出现概率. 根据上述思想，Williams 等人[65]提出了 REINFORCE 算法，将策略梯度的形式统一为：

$$g = \nabla_\theta \sum_{t=0}^{T-1} \log\pi\left(a_t|s_t;\theta\right)\left(R - b\right) \tag{17}$$

其中，b 是一个与当前轨迹 τ 相关的基线，通常设置为 R 的一个期望估计，目的是减小 R 的方差. 可以看出，R 超过基准 b 越多，对应的轨迹 τ 被选中的概率越大. 因此在大规模状态的 DRL 任务中，可以通过深度神经网络参数化表示策略，并采用传统的策略梯度方法来求解最优策略.

此外，优化策略的另一种思路是增加“好”的动作出现的概率. 在 RL 中一般是通过优势函数评价动作的好坏，因此可以利用优势函数项来构造策略梯度：

$$g = \nabla_\theta \sum_{t=0}^{T-1} \hat{A}_t \log\pi\left(a_t|s_t;\theta\right) \tag{18}$$

其中，$\hat{A}_t$ 表示状态动作对 $\left(s_t,a_t\right)$ 优势函数的一个估计，通常构造成如下形式：

$$\hat{A}_t^\gamma = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots - V\left(s_t\right) \tag{19}$$

其中，$\gamma \in [0,1]$ 表示折扣因子. 此时带折扣的奖赏之和 $r_t + \gamma r_{t+1} + \gamma r_{t+2} + \cdots$ 相当于式（17）中的 R，带折扣的状态值函数 $V\left(s_t\right)$ 相当于式（17）中的基准 b. 当 $\hat{A}_t^\gamma > 0$ 时，会增加对应动作被选择的概率，而 $\hat{A}_t^\gamma < 0$ 时，会减小对应动作被选择的概率.

另外，Hafner 等人[66]使用值函数来估计带折扣的奖赏和，进一步地缩小了梯度项的方差. 此时一步截断的 $\hat{A}_t^\gamma$ 表示为：

$$\hat{A}_t^\gamma = r_t + \gamma V\left(s_{t+1}\right) - V\left(s_t\right) \tag{20}$$

类似地，两步截断的 $\hat{A}_t^\gamma$ 表示为：

$$\hat{A}_t^\gamma = r_t + \gamma r_{t+1} + \gamma^2 V\left(s_{t+2}\right) - V\left(s_t\right) \tag{21}$$

然而使用值函数估计带折扣的奖赏和，也会产生一定的估计偏差. 为了缩小方差的同时还能保证偏差较小，Schulman 等人[67]提出了广义优势函数（generalized advantage function）：

$$\hat{A}_t^\gamma = \delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \ldots + (\gamma\lambda)^{T-t-1}\delta_{T-1} \tag{22}$$

其中，$\delta_t = r_t + \gamma V\left(s_{t+1}\right) - V\left(s_t\right)$. λ 是一个调节因子，范围大小为 $0 < \lambda < 1$. 当 λ 接近于 0 时，$\hat{A}_t^\gamma$ 是低方差、高误差的；当 λ 接近于 1 时，$\hat{A}_t^\gamma$ 是高方差、低误差的. 基于广义优势函数的策略梯度方法的不足之处在于：在利用式（16）全局优化策略的过程中，很难确定一个合理的步长参数 α 来保证学习的稳定性. 针对此问题，Schulman 等人[68]提出了一种被称为区域信赖的策略最优化（Trust Region Policy Optimization，TRPO）方法. TRPO 的核心思想是：强制限制同一批次数据上新旧两种策略预测分布的 KL 差异，从而避免导致策略发生太大改变的参数更新步. 为了将应用范围扩展到大规模状态空间的 DRL 任务中，TRPO 算法使用深度神经网络来参数化策略，在只接收原始输入图像的情况下实现了端对端的控制. 实验表明，TRPO 在一系列 2D 场景下的机器人控制和 Atari 2600 游戏任务中都表现优异. 此后，Schulman 等人[67]又尝试将广义优势函数与 TRPO 方法相结合，在一系列 3D 场景下的机器人控制任务中取得了突破.

此外，深度策略梯度方法的另一个研究方向是通过增加额外的人工监督来促进策略搜索. 例如著名的 AlphaGo 围棋机器人，先**使用监督学习从人类专家的**棋局**中预测人类的**走子行为，再用策略梯度方法针对赢得围棋比赛的真实目标进行精细的策略参数调整[20]. 然而在某些任务中是缺乏监督数据的，比如现实场景下的机器人控制，可以通过引导

式策略搜索[69]（guided policy search）方法来监督策略搜索的过程．在只接受原始输入信号的真实场景中，引导式策略搜索实现了对机器人的操控．

4.2　基于行动者评论家的深度策略梯度方法

4.1 节中深度策略梯度方法的基本思想是通过各种策略梯度方法直接优化用深度神经网络参数化表示的策略．这类方法在每个迭代步，都需要采样批量大小为 N 的轨迹 $\{\tau_i\}_{i=1}^{N}$ 来更新策略梯度．然而在许多复杂的现实场景中，很难在线获得大量训练数据．例如在真实场景下机器人的操控任务中，在线收集并利用大量训练数据会产生十分昂贵的代价，并且动作连续的特性使得在线抽取批量轨迹的方式无法达到令人满意的覆盖面．以上问题会导致局部最优解的出现．针对此问题，可以将传统 RL 中的行动者评论家（Actor-Critic，AC）框架拓展到深度策略梯度方法中．图 7 展示了基于 AC 框架的深度策略梯度方法的学习结构．

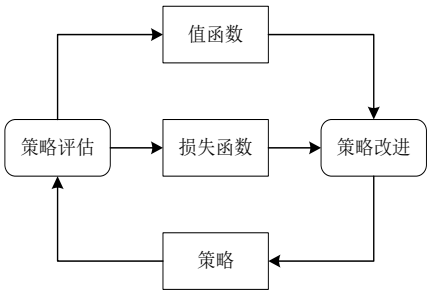

图 7　基于 AC 框架的深度策略梯度方法的学习结构

下面阐述一种重要的基于 AC 框架的深度策略梯度算法．Lillicrap 等人[21]利用 DQN 扩展 Q 学习算法的思路对确定性策略梯度[70]（Deterministic Policy Gradient，DPG）方法进行改造，提出了一种基于 AC 框架的深度确定性策略梯度（Deep Deterministic Policy Gradient，DDPG）算法，该算法可用于解决连续动作空间上的 DRL 问题．DDPG 分别使用参数为 θ^{μ} 和 θ^{Q} 的深度神经网络来表示确定性策略 $a=\pi(s|\theta^{\mu})$ 和值函数 $Q(s,a|\theta^{Q})$．其中，策略网络用来更新策略，对应 AC 框架中的行动者；值网络用来逼近状态动作对的值函数，并提供梯度信息，对应 AC 框架中的评论家．在 DDPG 中，目标函数被定义为带折扣的奖赏和：

$$J(\theta^{\mu})=\mathrm{E}_{\theta^{\mu}}\left[r_1+\gamma r_2+\gamma^2 r_3+\ldots\right] \tag{23}$$

然后，采用随机梯度下降方法来对目标函数进行端对端的优化．Silver 等人[70]证明了目标函数关于 θ^{μ} 的梯度等价于 Q 值函数关于 θ^{μ} 的期望梯度：

$$\frac{\partial J(\theta^{\mu})}{\partial \theta^{\mu}}=\mathrm{E}_{s}\left[\frac{\partial Q(s,a|\theta^{Q})}{\partial \theta^{\mu}}\right] \tag{24}$$

根据确定性策略 $a=\pi(s|\theta^{\mu})$ 可得：

$$\frac{\partial J(\theta^{\mu})}{\partial \theta^{\mu}}=\mathrm{E}_{s}\left[\frac{\partial Q(s,a|\theta^{Q})}{\partial a}\frac{\partial \pi(s|\theta^{\mu})}{\partial \theta^{\mu}}\right] \tag{25}$$

通过 DQN 中更新值网络的方法来更新评论家网络，此时梯度信息为：

$$\frac{\partial L(\theta^{Q})}{\partial \theta^{Q}}=\mathrm{E}_{s,a,r,s'\sim D}\left[\left(y-Q(s,a|\theta^{Q})\right)\frac{\partial Q(s,a|\theta^{Q})}{\partial \theta^{Q}}\right] \tag{26}$$

其中，$y=r+\gamma Q(s',\pi(s'|\hat{\theta}^{\mu})|\hat{\theta}^{Q})$，$\hat{\theta}^{\mu}$ 和 $\hat{\theta}^{Q}$ 分别表示目标策略网络和目标值网络的参数．DDPG 使用经验回放机制从 D 中获得训练样本，并将由 Q 值函数关于动作的梯度信息从评论家网络传递给行动者网络．并依据式（25）沿着提升 Q 值的方向更新策略网络的参数．

实验表明，DDPG 不仅在一系列连续动作空间的任务中表现稳定，而且求得最优解所需要的时间步也远远少于 DQN．与基于值函数的 DRL 方法相比，基于 AC 框架的深度策略梯度方法优化策略效率更高、求解速度更快．

然而在有噪声干扰的复杂环境下，策略一般都具有一定的随机性．DDPG 使用确定性的策略梯度方法．对于随机环境的场景，该方法并不适用．针对此问题，Heess 等人[71]提出了一种适用于连续动作空间任务的通用框架，称为随机值梯度（Stochastic Value Gradient，SVG）方法．SVG 使用“再参数化”（re-parameterization）[72]的数学技巧来学习环境动态性的生成模型，将确定性策略梯度方法扩展为一种随机环境下的策略优化过程．Balduzzi 等人[73]基于相容的值函数逼近器（compatible function approximation）理论，提出了值梯度反向更新（Value-Gradient Backpropagation，GProp）方法．Peng 等人[74]融合多个策略网络和对应的值网络，提出了一种基于混合型行动者评论家指导（Mixture of Actor Critic Experts，MACE）的深度策略梯度方法．该方法在自适应机器人控制任

务中取得了实质性的进展．MACE 相比于单个 AC 框架指导的深度策略梯度方法，有着更快的学习速度．随后，Heess 等人[75]使用循环神经网络，进一步扩展了 DPG 和 SVG 算法的适用范围，提出了循环确定性策略梯度（Recurrent Deterministic Policy Gradient，RDPG）和循环随机值梯度（Recurrent Stochastic Value Gradient，RSVG）方法．RDPG 和 RSVG 可以处理一系列部分可观察场景下连续动作的控制任务．Hausknecht 等人[76]进一步将深度策略梯度方法扩展到了参数化的连续动作空间问题中．此后，Schulman 等人[77]提出了一种形式化的随机计算图模型（stochastic computation graphs），开展了同时包含随机性和确定性操作的复杂深度策略梯度的研究．

4.3 异步的优势行动者评论家算法

不同类型的深度神经网络为 DRL 中策略优化任务提供了高效运行的表征形式．为了缓解传统策略梯度方法与神经网络结合时出现的不稳定性，各类深度策略梯度方法（如 DDPG、SVG 等）都采用了经验回放机制来消除训练数据间的相关性．然而经验回放机制存在两个不足之处：（1）agent 与环境的每次实时交互都需要耗费很多的内存和计算力；（2）经验回放机制要求 agent 采用离策略（off-policy）方法来进行学习，而离策略方法只能基于旧策略生成的数据进行更新．针对这些问题，Mnih 等人[78]根据异步强化学习（Asynchronous Reinforcement Learning，ARL）的思想，提出了一种轻量级的 DRL 框架，该框架可以使用异步的梯度下降法来优化网络控制器的参数，并可以结合多种 RL 算法．其中，异步的优势行动者评论家算法（Asynchronous Advantage Actor-Critic，A3C）在各类连续动作空间的控制任务上表现的最好．

具体地，A3C 算法利用 CPU 多线程的功能并行、异步地执行多个 agent．因此在任意时刻，并行的 agent 都将会经历许多不同的状态，去除了训练过程中产生的状态转移样本之间的关联性．因此这种低消耗的异步执行方式可以很好地替代经验回放机制．

A3C 算法在训练时降低了对硬件的要求．深度策略梯度算法十分依赖计算能力很强的图形处理器（Graphics Processing Unit，GPU），而 A3C 算法在实际的操作过程中只需要一个标准的多核 CPU．由表 1 可知，A3C 算法通过应用多线程技术，降低了模型对硬件的需求，在训练时间更少的情况下，A3C 算法在 Atari 2600 游戏任务上的平均性能有明显提升．而且 A3C 算法能够只根据原始的视觉输入学习到行走 3D 迷宫的有效策略．此外，A3C 算法还可以广泛应用于各种连续动作空间问题．综上所述，A3C 算法能够广泛应用于各种 2D、3D 离散和连续动作空间的任务，并且在这些任务中都取得了最佳的效果．这说明 A3C 是目前最通用和最成功的一种 DRL 算法．当然，将 A3C 与近期的一些深度策略梯度算法相结合可能会进一步提升其性能．

表 1 不同 DRL 模型在 57 个 Atari 游戏上的平均训练耗时及游戏性能的提升

模型	训练条件	训练时间/天	平均性能提升
DQN	GPU	8	121.9%
Gorila	100 台主机	4	215.2%
DDQN	GPU	8	332.9%
Dueling DDQN	GPU	8	343.8%
Prioritized DQN	GPU	8	463.6%
A3C, FF	CPU	1	344.1%
A3C, FF	CPU	4	496.8%
A3C, LSTM	CPU	4	623.0%

5 基于搜索与监督的深度强化学习

除了基于值函数的 DRL 和基于策略梯度的 DRL 之外，还可以通过增加额外的人工监督来促进策略搜索的过程，即为基于搜索与监督的 DRL 的核心思想．蒙特卡洛树搜索[79]（Monte Carlo Tree Search，MCTS)作为一种经典的启发式策略搜索方法，被广泛用于游戏博弈问题中的行动规划．因此在基于搜索与监督的 DRL 方法中，策略搜索一般是通过 MCTS 来完成的．本章所介绍的 AlphaGo 围棋算法[20]将深度神经网络和 MCTS 相结合，并取得了卓越的成就．

5.1 结合深度神经网络和 MCTS

在 AI 领域中，由于围棋存在状态空间巨大且精确评估棋盘布局、走子困难等原因，开发出一个能够精通围棋游戏的 agent，一直被认为是最有挑战性的难题．直到 Silver 等人将 CNN 与 MCTS 相结合，提出了一种被称作为 AlphaGo 的围棋算法，在

一定程度上解决了这一难题．AlphaGo 的主要思想有两点：（1）使用 MCTS 来近似估计每个状态的值函数；（2）使用基于值函数的 CNN 来评估棋盘的当前布局和走子．AlphaGo 完整的学习系统主要由以下 4 个部分组成：

（1）策略网络（policy network）．又分为监督学习的策略网络和 RL 的策略网络．策略网络的作用是根据当前的局面来预测和采样下一步走棋．

（2）滚轮策略（rollout policy）．目标也是预测下一步走子，但是预测的速度是策略网络的 1000 倍．

（3）估值网络（value network）．根据当前局面，估计双方获胜的概率．

（4）MCTS．将策略网络、滚轮策略和估值网络融合进策略搜索的过程中，以形成一个完整的系统．

首先，在训练 AlphaGo 的第 1 阶段，通过围棋对弈服务器 KGS 上的带标签的对局数据，来训练监督学习的策略网络 P_σ，最终目标是模拟当前棋盘状态 s 下人类玩家的走法 a：

$$\Delta\sigma \propto \frac{\partial \log P_\sigma(a|s)}{\partial \sigma} \tag{27}$$

其中，σ 表示监督学习的策略网络的参数．策略网络是一个 13 层的深度卷积网络，具体的训练方式为梯度下降法．训练结束后，在测试集上使用所有输入特征，预测人类专业棋手走子动作的准确率为 57.0%．只使用棋盘位置和历史走子记录作为输入时，预测的准确率也达到了 55.7%．另外，使用局部特征匹配与线性回归的方式来训练滚轮策略网络，预测人类专业棋手走子动作的准确率为 24.2%．

其次，在训练 AlphaGo 的第 2 阶段通过策略梯度方法来训练 RL 的策略网络 P_ρ，以进一步提高策略网络的走子能力，最终最大化整局棋的期望奖赏：

$$\Delta\rho \propto \frac{\partial \log P_\rho(a_t|s_t)}{\partial \rho} z_t \tag{28}$$

其中，ρ 表示 RL 的策略网络的参数，z_t 表示一局棋最终所获得的收益，胜为+1，负为-1．具体的训练方式是：随机选择先前迭代轮的策略网络和当前的策略网络 P_ρ 相互对弈，并利用策略梯度方法来更新参数，最终得到增强的策略网络．P_ρ 在结构上与 P_σ 是完全相同的．通过训练后，增强的 P_ρ 在对抗 P_σ 时胜率超过了 80%，对抗 Pachi（一种围棋算法）时达到 85%的胜率．

然后，在训练 AlphaGo 的第 3 阶段，主要关注的是对当前局面的价值评估．在训练时，通过最小化估值网络输出 $v_\theta(s)$ 和收益 z 之间的均方误差来训练估值网络：

$$\Delta\theta \propto \frac{\partial v_\theta(s)}{\partial \theta}(z - v_\theta(s)) \tag{29}$$

估值网络所采用的网络结构与策略网络类似，两者的不同之处在于：估值网络在输出层只输出单一的预测值 $v_\theta(s)$，用于估计黑棋或白棋获胜的概率，而策略网络输出的是可能走子动作的一个概率分布．

最后，AlphaGo 将 MCTS 算法与策略网络、估值网络相结合，并通过超前的搜索来选择走子动作．在每个时间步 t，从状态 s_t 中选择一个走子动作 a_t：

$$a_t = \text{argmax}_a(Q(s_t,a) + u(s_t,a)) \tag{30}$$

其中，$u(s_t,a)$ 表示额外的奖励，目的是在鼓励探索的前提下最大化走子动作的值：

$$u(s,a) \propto \frac{P(s,a)}{1+N(s,a)} \tag{31}$$

其中，$P(s,a) = P_\sigma(a|s)$，表示用策略网络的输出作为先验概率；$N(s,a)$ 表示状态动作对的访问次数．$u(s,a)$ 与先验概率成正比，与访问次数成反比．随后，当遍历 L 步后到达一个叶节点 S_L 时，综合估值网络 $v_\theta(s_L)$ 和滚轮策略网络两种评估方式来获得叶子节点的值：

$$V(s_L) = (1-\lambda)v_\theta(s_L) + \lambda z_L \tag{32}$$

其中，z_L 表示棋局终止时所获的奖赏．而后，更新状态动作对的访问次数和对应的动作值：

$$\begin{aligned} N(s,a) &= \sum_{i=1}^{N} 1(s,a,i) \\ Q(s,a) &= \frac{1}{N(s,a)} \sum_{i=1}^{N} 1(s,a,i) V(s_L^i) \end{aligned} \tag{33}$$

其中，$1(s,a,i)$ 与状态动作对 (s,a) 是否在第 i 次模拟中被访问到有关，具体为被访问到时值设置为 1，没被访问到时值设置为 0；s_L^i 表示第 i 次模拟时的叶子状态节点．一旦搜索完成，agent 从根节点的位置开始选择访问次数最多的走子动作．

训练完成后的 AlphaGo 先后战胜了一位欧洲冠军和一位世界冠军棋手，充分证明了基于 DRL 算

法的计算机围棋算法已经达到了人类顶尖棋手的水准．AlphaGo 的成功对于通用人工智能的发展具有里程碑式的意义．

6 分层深度强化学习

在一些复杂的 DRL 任务中，直接以最终目标为导向来优化策略，效率很低．因此可以利用分层强化学习（Hierarchical Reinforcement Learning，HRL）将最终目标分解为多个子任务来学习层次化的策略，并通过组合多个子任务的策略形成有效的全局策略[80]．本章将主要介绍 3 种具有代表性的分层 DRL 算法．

6.1 基于时空抽象和内在激励的分层深度强化学习

在一些复杂的目标导向型任务中，稀疏反馈的问题一直阻碍着 agent 性能的提升．现有的各种 DRL 模型（DQN、DRQN 等）在面对操作难度很大的 Montezuma's Revenge 游戏时，并不能表现出任何的智能行为．这是由于在学习过程中，agent 得到的反馈信号极少，导致其对某些重要状态空间的探索很不充分．若要在此类复杂的环境中进行有效的学习，agent 必须感知出层次化时空抽象（temporal abstraction）的知识表达，并在此基础上通过某些内在激励来促进其探索[81]．Kulkarni 等人[82]基于以上思想，提出了一种层次化的 DQN 算法（hierarchical Deep Q-Network，h-DQN）．h-DQN 是一种基于时空抽象和内在激励的分层 DRL 算法，通过在不同的时空尺度上设置子目标来层次化值函数．顶层的值函数用于确定 agent 的决策，以得到下一个内在激励的子目标，而底层的值函数用于确定 agent 的行动，以满足顶层的子目标．

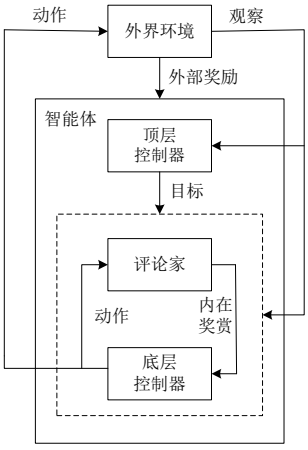

图 8 层次化的 DQN 模型结构图

如图 8 所示，h-DQN 模型是通过两个层次的模块来进行决策的：

顶层控制器（meta-controller）在上一个子目标完成或者到达终止状态之后，接受来自环境的输入，并通过最大化未来累计外部奖赏 $R_{mc}^{t}=\sum_{t'=t}^{t+N} f_{t'}+\gamma \max_{g'} Q_{mc}^{*}\left(s_{t+N}, g'\right)$ 的期望来确定新的子目标．其中，N 代表完成当前子任务所需要的步长，g' 表示在状态 s_{t+N} 时确定的新的子目标，$f_{t'}$ 表示从环境中得到的外部奖赏信号，Q_{mc}^{*} 表示顶层控制器在 t 时刻的最优值函数：

$$Q_{mc}^{*}(s,g)=\max_{\pi_g} \mathrm{E}\left(R_{mc}^{t} \middle| s_t=s, g_t=g, \pi_g\right) \quad (34)$$

其中，π_g 表示在状态 s 下子目标 g 的一个概率分布．

在当前子目标完成或者到达终止状态之前，底层模块（controller）都是根据当前的状态和 N 个时间步内固定的子目标来决定 agent 的动作，并通过最大化未来累计内部奖赏 $R_{c}^{t}=r_t+\gamma \max_{a_{t+1}} Q_{c}^{*}\left(s_{t+1}, a_{t+1}; g\right)$ 的期望来优化底层模块：

$$Q_{c}^{*}\left(s,a \middle| g\right)=\max_{\pi_{ag}} \mathrm{E}\left(R_{c}^{t} \middle| s_t=s, a_t=a, g_t=g, \pi_{ag}\right) \quad (35)$$

其中，R_{c}^{t} 中的 Q_{c}^{*} 表示底层模块在 t 时刻的最优值函数，r_t 表示从环境中得到的外部奖赏，g 表示在状态 s 时 agent 的子目标．而式（35）中的 π_{ag} 表示在当前状态和子目标下 agent 可采取动作的分布．在训练过程中，h-DQN 通过不同时空尺度上的批量梯度下降法（Batch Gradient Descent，BGD）来更新两个模块的参数．其中，顶层控制器每隔 N 时刻，存储转移序列 $\left(s_t, g_t, f_t, s_{t+N}\right)$ 到样本池 D_{mc} 中．不同的是，底层模块在当前子目标完成或者到达终止状态之前的每个时刻 t，都要将转移序列 $\left(s_t, a_t, g_t, r_t, s_{t+1}\right)$ 存储到样本池 D_c 中．上述训练过程体现了不同时空尺度的特性．

与 DQN 的参数更新方式类似，h-DQN 也是通过非线性的深度卷积神经网络来近似表示 Q 值函数 $Q^{*}(s,g)\approx Q\left(s,g \middle| \theta\right)$．两个层次模块的值函数 $Q\in\{Q_{mc}, Q_c\}$ 可以分别通过最小化各自的损失函数来训练．其中，低层次的值函数 Q_c 对应的损失函数形式为：

$$L_c\left(\theta_{c,i}\right)=\mathrm{E}_{(s,a,g,r,s')\sim D_c}\left[\left(y_{c,i}-Q_c(s,a \mid \theta_{c,i}, g)\right)^2\right] \quad (36)$$

其中，$y_{c,i}=r+\gamma \max_{a'} Q_c\left(s', a' \middle| \theta_{c,i-1}, g\right)$ 表示目标值

函数，$\theta_{c,i-1}$ 表示目标值网络的参数．确定完损失函数之后，即可通过对当前网络参数求偏导得到梯度信息，从而更新参数．高层次的值函数 Q_{mc} 的更新方式与之类似．

实验表明，h-DQN 模型可以在存在严重稀疏反馈问题的 DRL 任务中，保持高效地探索，从而提升了 agent 在面对复杂任务时的性能表现．针对 h-DQN 模型，还可以开展的未来工作如下：

（1）通过 CNN 提取的抽象状态缺乏结构化和复合化的表示．因此可以将深度生成式模型[83]（Deep Generative Model，DGM）与 h-DQN 相结合来区分输入图像中不同的构成因子（如物体、姿态、位置信息等），以获得结构化和复合化的表示；

（2）将循环神经网络模块加入 h-DQN，以缓解环境中存在的部分可观察和延迟反馈的问题；

（3）采用情节式记忆模型[84]来扩展 h-DQN 的适用范围．

6.2　基于内部 Option 的分层深度强化学习

h-DQN 模型需要在不同时空尺度上人工设置一些中间目标来促进 agent 的探索．然而人工设定中间目标的方式也限制了 h-DQN 的适用场景．针对此问题，Krishnamurthy 等人[85]提出了一种基于内部 Option 的深度 Q 学习（deep intra-option Q-learning）模型．该模型结合时空抽象和深度神经网络，自动地完成子目标的学习，同时可以在给定抽象状态和扩展动作集的前提下获得当前任务的一个层次化描述．该分层 DRL 方法省去了特定的内在激励和人工设定中间目标这两个环节，加速了 agent 的学习进程，并增强了模型在其他任务上的泛化能力．

受动力学系统启发，基于内部 Option 的深度 Q 学习模型利用一种被称作 PCCA+的聚类算法[86]．该算法可以在状态空间中寻求亚稳定的区域并将其与抽象状态相关联．这种关联性信息可以产生对应问题的学习技巧（也称为 Option）．例如，在著名的出租车问题中[87]，其中的一组抽象状态是出租车起始和乘客所处的位置，那么此时 Option 就是从出租车起始位置到乘客所处位置的一系列动作组合．很明显，生成的 Option 相当于 h-DQN 模型中设定的中间目标，省去了复杂的人工设定中间目标的过程，并使得学习到的 Option 与具体的学习任务无关．因此在相同的状态空间下，该模型具有很强的泛化性．

6.3　深度后续强化学习

一般地，在只给定原始输入观察和奖赏值的情况下，通过基于模型（model-based）或者模型无关（model-free）的 DRL 算法可以学习到鲁棒的值函数．后续状态表示法（Successor Representation，SR）为学习值函数提供了第 3 种选择．SR 将值函数分解为两个部分：后续状态映射图（successor map）和立即奖赏指示器（reward predictor）．后续状态映射图表示在给定当前状态下到达未来某一状态占有率的期望．立即奖赏指示器表示从状态到奖赏值的映射．在 SR 中，这两个部分以内积的形式构成值函数．基于上述理论知识，Kulkarni 等人[88]将 SR 的应用范围扩展到大规模状态空间的 DRL 问题中，提出了深度后续强化学习（Deep Successor Reinforcement Learning，DSRL）．

在 SR 中，后续状态映射图代表给定当前状态 s 下到达未来某一状态 s' 的占有率的期望：

$$M(s,s',a)=\mathrm{E}\left[\sum_{t=0}^{\infty}\gamma^t 1[s_t=s']\middle|s_0=s,a_0=a\right] \quad (37)$$

其中当括号项里的表达式为真时，1[□]取值为 1，否则取值为 0．依据贝尔曼方程的构造形式可得：

$$M(s,s',a)=1[s_t=s']+\gamma\mathrm{E}[M(s_{t+1},s',a_{t+1})] \quad (38)$$

因此在 SR 中，Q 值函数可以表示为式（38）与立即奖赏的内积：

$$Q^{\pi}(s,a)=\sum_{s'\in S}M(s,s',a)R(s') \quad (39)$$

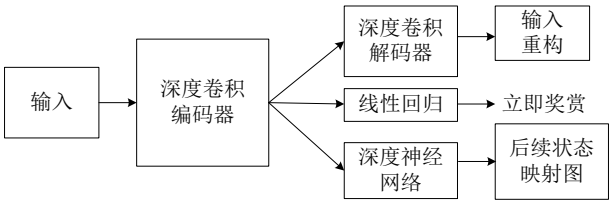

图 9　DSRL 模型的结构

对于大规模 DRL 问题，直接用上式（39）表示和学习 Q 值函数是不切实际的．因此 DSRL 使用了多个深度神经网络来近似表示各项模块．如图 9 所示，DSRL 首先使用了一个参数为 θ 的深度卷积编码器将状态 s 近似表示为 D 维的特征 ϕ_s；然后基于 ϕ_s 使用另一个参数为 α 的深度神经网络 u_ζ 来近似表示后续状态映射图 $m_{sa}\approx u_\zeta(\phi_s,a)$，并使用线性回归来近似立即奖赏 $R(s)\approx\phi_s\cdot\mathbf{w}$，其中 $\mathbf{w}\in\mathrm{P}^D$ 表

示权值向量．另外，由于在很多问题中奖赏是十分稀疏的，因此 DSRL 算法使用参数为θ^-深度卷积解码器来训练一种基于内在奖励的指示器$R_i(s)=g_{\theta^-}(\phi_s)$．最终，DSRL 中的 Q 值函数可以表示为两大部件的内积：

$$Q^{\pi}(s,a)\approx m_{sa}\cdot\mathbf{w} \tag{40}$$

DSRL 将 Q 值函数分解为后续状态映射图和立即奖赏指示器的内积，这种新颖的值函数构造方式使得 agent 对突出奖赏（distal reward）的变化值更加敏感，这样 agent 就可以在随机策略的情形下分解出更有价值的子目标．例如在出租车问题中，接到乘客就能获得突出奖赏．DSRL 为分层 DRL 提供了一个理想的分解子目标的方法．通过阶段性地分解子目标和学习子目标策略，DSRL 一定程度上增强了对未知状态空间的探索力度，使得 agent 更加适应那些存在延迟反馈的任务．

7 多任务迁移深度强化学习

在传统 DRL 方法中，每个训练完成后的 agent 只能解决单一任务．然而在一些复杂的现实场景中，需要 agent 能够同时处理多个任务，此时多任务学习和迁移学习就显得异常重要．在 RL 领域，Wilson 等人[89]利用层次化的混合贝叶斯模型来提供关于新任务的先验知识，使得 agent 能更好适应新的任务场景；Li 等人[90]针对部分可观察的随机多任务场景，提出区域化的策略表示（regionalized policy representation）用于刻画 agent 在不同任务场景下的行为．该方法利用狄利克雷过程（Dirichlet process）中包含的聚类性质来共享相似任务间的训练情节，并在不同任务间传递有价值的信息．与单任务学习模式相比，这种多任务 RL 的方法在格子世界导航和多目标分类任务上都取得了更突出的表现；E. Taylor 等人[91]提出了一种在不同任务之间迁移值函数的方式；Fern ́andez 等人[92]使用一种反映 agent 当前与过去状态动作对关系的映射，使得过去学习到的策略能够及时迁移到新的任务中；Wang 等人[93]总结出 RL 中的迁移分为两大类：行为上的迁移和知识上的迁移，这两大类迁移也被广泛应用于多任务 DRL 算法中．

7.1 行为模拟的多任务迁移深度强化学习

DQN 在解决多个游戏任务时，保持着一致的网络结构和参数设置．不过该方法也存在着局限性：每个网络一次只能解决一种游戏任务．因此可以尝试训练出一次完成多种任务的网络．该网络必须要充分挖掘多个任务之间的相似性，并可以在相似的任务之间泛化知识．Parisotto 等人[94]提出了一种基于行为模拟（actor-mimic）的多任务迁移 DRL 方法，它可以基于一组相关的源任务预训练一个深度策略网络．该方法的基本思想是通过监督信号的指导，使得单一的策略网络学会各自任务相对应的策略，并将学习到的知识迁移到相似的新任务中．下面介绍基于行为模拟的多任务迁移 DRL 方法的具体工作原理．

（1）为了训练出一个可以同时解决多个任务的策略网络，设定多个源任务为$S_1,\ldots,S_N$，对应的指导网络为$E_1,\ldots,E_N$．为了便于理解，用于多任务的策略网络可看作是学习网络（student network），$E_1,\ldots,E_N$ 可看作是指导网络（expert network)．actor-mimic 方法的基本思想是强制学习网络去模拟指导网络在每个状态下所选择的行为．该方法根据输出 Q 值的波尔兹曼分布（Boltzmann distribution）将每个指导网络转换成一个策略网络：

$$\pi_{E_i}(a|s)=\frac{e^{\tau^{-1}Q_{E_i}(s,a)}}{\sum_{a'\in A_{E_i}}e^{\tau^{-1}Q_{E_i}(s,a')}} \tag{41}$$

其中τ表示温度因子，A_{E_i}表示指导网络E_i的动作空间．对于源任务S_i中的每一个状态s，actor-mimic 根据多任务网络策略与指导网络策略之间的交叉熵（cross entropy）定义一个策略回归目标函数：

$$L_p^i(\theta)=\sum_{a\in A_{E_i}}\pi_{E_i}(a|s)\log\pi_{AMN}(a|s;\theta) \tag{42}$$

其中，$\pi_{AMN}(a\,|\,s;\theta)$表示用于模拟指导者行为的多任务策略网络．指导者网络的输出策略是一个稳定的监督训练信号，不断指导多任务网络的行为向指导网络的行为靠拢．

（2）通过优化特征回归目标函数可以获得专家网络的进一步指导．$h_{AMN}(s)$表示多任务网络，$h_{E_i}(s)$表示第i个指导网络最后一个隐藏层的激活值．$f_i(h_{AMN}(s))$定义为第i个特征回归网络，表示在状态s下，根据$h_{AMN}(s)$预测特征值$h_{E_i}(s)$．网络f_i可以通过如下的特征回归损失函数来进行训练：

$$L_{FR}^i(\theta,\theta_{f_i})=\left\|f_i(h_{AMN}(s;\theta);\theta_{f_i})-h_{E_i}(s)\right\|_2^2 \tag{43}$$

其中θ表示多任务网络的参数，θ_{f_i}表示第i个特征回归网络的参数．通过反向传播算法将误差从特征回归网络回馈给多任务网络，迫使多任务网络完成对指导网络特征信息的预测．训练完成后，指导网络的特征信息都包含在多任务网络的特征中．

（3）将策略回归和特征回归目标函数相结合来定义整体行为模拟的目标函数：

$$L_{AM}^{i}\left(\theta,\theta_{f_i}\right)=L_{p}^{i}\left(\theta\right)+\beta*L_{FR}^{i}\left(\theta,\theta_{f_i}\right) \tag{44}$$

其中β是控制两个不同目标函数相对权重的缩放因子．直观上，可以将策略回归目标函数当作老师（指导网络），指导学生（多任务策略网络）如何执行动作．特征回归目标函数可以类比为一个老师指导学生去理解为什么选择执行此动作．通过L_{AM}^{i}对参数求偏导，再利用梯度下降法训练出一个擅长解决多任务的 agent．

7.2　基于策略蒸馏的多任务迁移深度强化学习方法

Rusa 等人[95]提出一种新颖的多任务迁移 DRL 方法，称作策略蒸馏（policy distillation）．该方法根据学习网络和指导网络 Q 值的偏差来确定 Q 值回归目标函数，引导学习网络逼近指导网络的值函数空间．

具体地，策略蒸馏是一种从指导模型T向学习模型S迁移知识的方法．指导模型T用于产生训练数据集$D^{T}=\{(s_i,\mathbf{q}_i)\}_{i=0}^{N}$，其中每个样本包含一个简短的观察序列$s_i$和未标准化的 Q 值向量．首先，最简单的迁移 Q 值函数的方式是将固定策略下全部的 Q 值直接迁移到学习模型中．因此可以使用均方误差损失函数来训练学习模型S的参数：

$$L_{D^T}^{MSE}\left(\theta_S\right)=\sum_{i=1}^{|D|}\left\|\left(\mathbf{q}_i^{T}-\mathbf{q}_i^{S}\right)|\theta_s\right\|_2^2 \tag{45}$$

其中，$\mathbf{q}^{T}$表示指导网络T的 Q 值向量，$\mathbf{q}^{S}$表示学习网络S的 Q 值向量．由于不同任务中 Q 值函数的量级相差很多，直接迁移 Q 值函数可能会使学习过程很不稳定，并且所需的计算资源也较多．

另外一种迁移Q值函数的方式是只从T中将最大 Q 值所对应的动作$a_{i,best}=\mathrm{argmax}_a\left(\mathbf{q}_{\mathrm{i}}\right)$迁移到$S$中．此时使用负对数似然损失（Negative Log Likelihood，NLL）函数来预测相同的最优动作值，以训练学习模型S的参数：

$$L_{D^T}^{NLL}\left(\theta_S\right)=-\sum_{i=1}^{|D|}\log P\left(a_i=a_{i,best}|x_i,\theta_s\right) \tag{46}$$

不过由于很多动作的值函数可能是相差无几的，只迁移最大 Q 值对应的动作也是不充分的．

因此可以通过 Hinton 等人[96]提出的蒸馏（distillation）方法来迁移 Q 值函数，利用 KL 散度（Kullback-Leibler Divergence，KLD）定义损失函数：

$$L_{D^T}^{KL}\left(\theta_S\right)=\sum_{i=1}^{|D|}\mathrm{softmax}\left(\frac{\mathbf{q}_i^{T}|\theta_s}{\tau}\right)\ln\frac{\mathrm{softmax}\left(\frac{\mathbf{q}_i^{T}|\theta_s}{\tau}\right)}{\mathrm{softmax}\left(\mathbf{q}_i^{S}|\theta_s\right)} \tag{47}$$

其中 softmax 函数表示多元回归操作，可以将任意数值元素组成的D维向量转换成另一个D维向量．转换后的向量中各维度元素大小范围为$(0,1)$，并且所有元素之和为 1．另外，τ表示温度调节因子，升高τ使得更多的次要知识从T迁移到S．实验表明，基于 KL 散度的迁移方式训练的学习模型在解决 DRL 问题时表现出的效果最优．

图 10 描述了多任务的策略蒸馏过程．首先，使用多个训练完成的单任务 DQN 模型来产生输入状态、任务 id 和目标输出，并将其存储到各自的回放单元中．其中，任务 id 用来标识不同任务的指导模型．不同任务有着不同的动作集合和独立的输出层，因此在训练和评估模型的过程中必须使用 id 区分出不同的任务．然后，每个情节都按次序分别从各自的回放单元中采样n个训练样本，并基于这些训练样本来构造损失函数，以指导学习模型的训练．

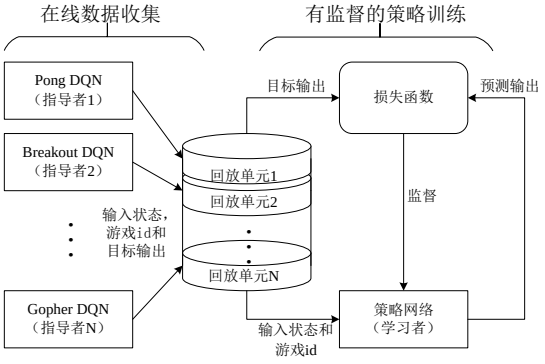

图 10 多任务的策略蒸馏过程

7.3　基于渐进式神经网络的迁移深度强化学习

7.1 和 7.2 小节分别介绍了两种用于解决 DRL 问题的迁移学习方法．然而这两种迁移学习方法都存在一定的局限性：在迁移知识之前，都需要耗费

大量的训练样本来指导模型的训练．虽然训练样本的获取对于视频游戏任务是没有难度的，但一些真实场景下的机器人控制任务就很难在线获取大量的训练数据．在这类消耗资源较多的场景中，试错学习会造成较大的损失．总之，上述两种迁移 DRL 的方法还不能将知识迁移到真实场景中．针对上述问题，Rusa 等人[97]提出了一种基于渐进式神经网络（progressive neural networks）的迁移 DRL 方法. 该渐进式神经网络可以通过逐层存储迁移知识和提取有价值特征，解决从仿真环境中迁移知识到真实环境的难题．

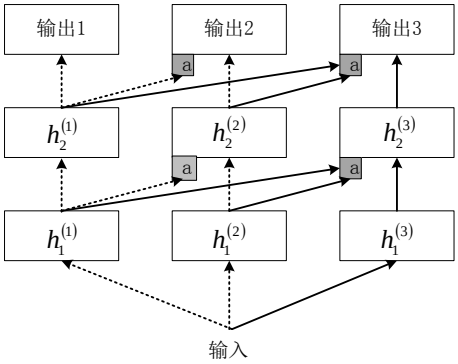

图 11 一个简单的 3 列渐进式网络结构

图 11 描述了一个简单的渐进式神经网络．其中标记为 a 的灰白盒子表示适配器（adapters），作用是保持前列的隐藏层激活值与原始输入的维度一致．渐进式神经网络的构成过程可以描述为：

（1）在第 1 列构造 1 个深度神经网络来训练某一任务；

（2）为了训练第 2 个任务，通过以下方式构造第 2 列深度神经网络：固定第 1 列神经网络的参数，并将其网络中各个隐藏层的激活值通过适配器处理之后连接到第 2 列神经网络的对应层，以作为额外输入；

（3）为了训练第 3 个任务，通过以下方式构建第 3 列神经网络：固定前两列神经网络的参数，前两列网络各个隐藏层的激活值通过适配器处理之后，组合连接到当前神经网络的对应层以作为额外的输入．另外，图 11 中所有的神经网络均通过 A3C 算法来训练参数．

渐进式神经网络在一系列序列任务中，通过逐层推进的方式来存储迁移知识和提取有价值特征，完成了对知识的迁移．这种基于渐进式神经网络的迁移 DRL 方法的优势在于[98]：针对新的任务，在训练时保留了之前训练模型的隐藏层状态，层次性地组合之前网络中每一隐藏层的有用特征，使得迁移学习拥有一个长期依赖的先验知识．

另外，Schaul 等人[98]提出了一种同时泛化状态和目标空间的通用值函数逼近器（Universe Value Function A pproximators，UVFAs）．利用 UVFAs 可以将已学习到的知识迁移到那些环境动态性相同，但目标不同的新任务中．Tessler 等人[99]结合深度技巧网络（Deep Skill Networks，DSNs）提出了层次化 DRL 网络（Hierarchical Deep Reinforcement Learning Network，H-DRLN）．该网络通过学习可重复利用的技巧和历史扩展的 Option 来完成相似任务间知识的迁移，形成针对最终目标的完整策略．

8 多 agent 深度强化学习

在面对一些真实场景下的复杂决策问题时，单 agent 系统的决策能力是远远不够的. 例如在拥有多玩家的 Atari 2600 游戏中，要求多个决策者之间存在相互合作或竞争的关系．因此在特定的情形下，需要将 DRL 模型扩展为多个 agent 之间相互合作、通信及竞争的多 agent 系统．

8.1 深度强化学习中多 agent 的合作与竞争

在多 agent RL 算法中，大多数情况下采取为每个 agent 单独分配训练机制的学习方式．例如，采取相互独立的 Q 学习算法来训练每个 agent．上述分布式的学习架构降低了实施学习的难度和计算的复杂度．对于大规模状态空间的 DRL 问题，用 DQN 算法替代 Q 学习算法来单独训练每个 agent，就可以构造出一个简易的多 agent 的 DRL 系统．Tampuu 等人[100]利用上述思想，并根据不同目标动态调整奖赏模式，提出了一种多 agent 之间可以相互合作与竞争的 DRL 模型．

为了说明该方法可以实现多个 agent 之间的竞争与合作．选取了经典的 Pong 游戏作为验证该方法有效性的平台．实验中，通过不同的奖励模式来验证模型具备的不同功能．具体地，当奖励模式设置为赢方+1 分，输方-1 分时，系统最终学到多个 agent 完全相互竞争的策略. 而如果将奖赏模式设置为不论每次结果如何赢方输方都为-1 分时，多个 agent 就可以学到一种完全相互协作的策略. 那么可以设置这样一个奖赏模式：赢方获得 $\rho \in [-1,1]$ 的奖励，输方获得-1 的惩罚．当 ρ 越大时，学习到的策

略越偏向于体现 agent 之间的竞争关系；而 ρ 越小时，学习到的策略越偏向于体现 agent 之间的合作关系．因此将 ρ 的大小设置到 $[-1,1]$ 内，系统就可以学会多个 agent 之间相互竞争与合作的策略．综上，该方法互不干扰地为每个 agent 单独训练自身的 Q 值函数，并针对不同任务调整奖赏函数模式，实现了一种通用的多 agent 相互合作与竞争的 DRL 方法．

8.2 基于通信协议的分布式深度循环 Q 网络

在面对一类需要多 agent 之间相互沟通的推理式任务时，通常 DQN 模型并不能学习到有效的策略．针对此问题，Foerster 等人[101]提出了一种称为分布式深度循环 Q 网络（Deep Distributed Recurrent Q-Networks，DDRQN)的模型，解决了状态部分可观察的多 agent 通信与合作的挑战性难题．

DDRQN 中采取为每个 agent 单独分配 DRQN 训练模块的方式来构建多 agent 系统．此时 Q 值函数的表示形式为 $Q^m\left(o_t^m,h_{t-1}^m,a^m;\theta_i^m\right)$．其中，$o_t^m$ 表示 t 时刻编号为 m 的 agent 的观察，h_{t-1}^m 表示 $t-1$ 时刻编号为 m 的 agent 对应的 LSTM 隐藏层状态，a^m 表示编号为 m 的 agent 对应的动作，θ_i^m 表示第 i 轮迭代编号为 m 的 agent 对应的网络参数．该方法为每个 agent 单独分配一个 Q 值网络，所消耗的计算和存储资源太大．实验表明，对于状态部分可观察下的多 agent 问题，这种训练方式提供的基于记忆的沟通信息也是远远不够的．因此 DDRQN 进行了 3 处改进：

（1）在任何时刻，对每个 agent 都在输入中增加上一时刻的动作信息，使得每个 agent 可以近似地估计状态动作历史序列；

（2）多 agent 之间共享网络参数，但每个 agent 的策略是基于自身历史信息产生的．通过这种参数共享的方式，大幅度减少了网络中可学习参数的数目，加快了学习的速度；

（3）改进后的 DDRQN 模型对应的 Q 值函数的表示形式为 $Q\left(o_t^m,h_{t-1}^m,m,a_{t-1}^m,a_t^m;\theta_i\right)$．其中，$m$ 表示当前处理的 agent 的索引，a_{t-1}^m 表示状态动作历史序列中的一部分，a_t^m 表示根据当前 Q 值网络的估计值所选出的动作．

通过 DDRQN 模型，解决了经典的红蓝帽子问题．实验表明，经过训练的 DDRQN 模型最终在多 agent 之间达成了一致的通信协议．这使得 DRL 算法成功地学习到一种通信协议，对于解决多 agent 的协作式任务具有较深远的意义．因此未来可以通过 DDRQN 模型来对物联网和移动智能设备上的通信协议进行学习和优化，以使其能够更好地适应不同的应用场景．

9 基于记忆与推理的深度强化学习

传统的基于视觉感知的 DRL 方法在解决更高层次的认知启发式任务（cognition-inspired tasks）时，其表现比起人类还相差甚远．即在解决一些高层次的 DRL 任务时，agent 不仅需要很强的感知能力，也需要具备一定的记忆与推理能力，才能学习到有效的决策．因此赋予现有 DRL 模型主动记忆与推理的能力就显得十分重要．

近年来外部存储的神经网络模型研究取得了实质性的进展．Graves 等人[102]提出了一种被称为神经图灵机的记忆结构（Neural Turing Machines, NTM），该结构在读写数据的同时，通过随机梯度下降方式来更新记忆结构的参数，优化记忆的内容．通过增加 NTM，使得神经网络模型具备完成复制、反转、加减法等一些简单任务的能力，说明了深度神经网络模型有了初步的记忆与推理能力．此后，Sukhbaatar 等人[103]又基于 NTM 提出了一种应用于问答系统和语言建模任务上的记忆网络模型，进一步提升了网络的长期记忆能力．因此在现有的 DRL 模型中加入这些外部记忆模块可以赋予网络一定的长期记忆、主动认知、推理等高层次的能力．另外，近年来认知神经科学的发展也一定程度推动了人工智能领域的发展．人们正在模拟人类大脑的辅助学习系统[104]，以构造一个可以自主记忆、学习和决策的 agent．

9.1 基于记忆网络的深度强化学习模型

由于传统的 DRL 模型不具备记忆、认知、推理等高层次的能力，因此在面对状态部分可观察和延迟奖赏的情形时，DQN 和 DRQN 等模型表现出的性能远远比不上人类．Junhyuk 等人[105]通过在传统的 DRL 模型中加入外部的记忆网络部件，并通过学习使模型拥有了一定的记忆和推理能力．根据是否加入 RNN 部件和反馈控制机制，可以分为以下几种模型：记忆深度 Q 网络（Memory Q-Network，MQN）、记忆深度循环 Q 网络（Recurrent Memory Q-Network，RMQN）、基于反馈控制机制的记忆深度循环 Q 网络（Feedback Recurrent Memory

Q-Network，FRMQN）.

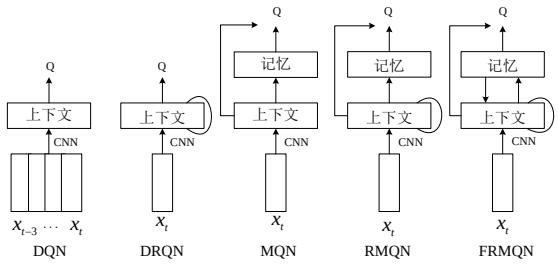

图 12 几种经典模型的结构对比图

图 12 刻画了几种典型的 DRL 模型，其中 x_t 表示 t 时刻的原始观察图像. 从图中可以看出，MQN、RMQN 和 FRMQN 与传统的 DQN、DRQN 模型的区别主要在于是否添加了记忆网络的结构. 传统的 DQN 直接基于 CNN 感知的上下文特征来逼近动作 Q 值，此时上下文向量表示从输入图像中提取的有价值信息；DRQN 则是通过 LSTM 构成的循环网络来存储一定长度内的时序信息，以缓解了状态部分可观察问题，此时的上下文向量应改为 LSTM 单元中的隐藏层状态向量；MQN 模型加入记忆模块，使得 agent 拥有类似于人类大脑的记忆、认知与推理能力. 该记忆模块可以通过 CNN 提取的特征向量来确定相关记忆的地址，并通过训练不断调整输入状态对应记忆的位置信息. 即该记忆模块可以动态获取与当前时刻输入信息相关的那部分记忆，从而更好地帮助网络逼近动作值函数；MRQN 类似于 DRQN，在 MQN 基础上加入由 LSTM 部件构成的循环神经网络，进一步提升模型在时间轴上的记忆能力；FRMQN 则同时加入循环神经网络和反馈控制机制. 通过反馈机制该模型在构成当前 LSTM 单元上下文向量 $\mathbf{h}_t$ 的输入时，增加了上一时刻检索的记忆信息 $\mathbf{o}_{t-1}$，使得 FRMQN 随着时间的推移逐步拥有越来越强大的自主推理能力. 当前时刻构成 $\mathbf{h}_t$ 的输入应包括：当前时刻编码得到的特征 $\mathbf{e}_t$、上一时刻记忆的信息 $\mathbf{o}_{t-1}$、当前时刻 LSTM 单元中的上下文向量 $\mathbf{h}_t$ 和记忆细胞状态 $\mathbf{c}_t$. 最终，FRMQN 模型通过整合当前检索的记忆 $\mathbf{o}_t$ 和上下文向量 $\mathbf{h}_t$ 来逼近动作值函数：

$$\mathbf{q}_t = \varphi^q\left(\mathbf{h}_t, \mathbf{o}_t\right) \tag{48}$$

其中，$\mathbf{q}_t \in \mathrm{P}^a$ 表示动作值函数的逼近，φ^q 表示一个多层的感知器. 具体的操作为：$\mathrm{g}_t = f\left(\mathbf{W}^h\mathbf{h}_t + \mathbf{o}_t\right)$，$\mathbf{q}_t = \mathbf{W}^q\mathbf{g}_t$，其中 $f(\square)$ 为矫正线性函数[106].

表 2 各网络模型在不同场景下的迷宫中成功到达目标位置的概率

模型	熟悉场景	陌生场景
DQN	62.9%（±3.4%）	60.1%（±3.4%）
DRQN	49.7%（±0.2%）	49.2%（±0.2%）
MQN	99.0%（±0.2%）	69.3%（±1.5%）
RMQN	82.5%（±2.5%）	62.3%（±1.5%）
FRMQN	100.0%（±0.0%）	91.8%（±1.0%）

通过上述分析可知，FRMQN 模型不仅拥有很好的视觉感知能力，在加入 LSTM 和记忆网络之后也具备了一定的记忆与推理功能. 另外，通过反馈控制机制，FRMQN 整合过去存储的有价值的记忆和当前时刻的上下文状态，评估动作值函数，做出正确的决策. 这种整合了感知、记忆、推理和反馈功能的网络结构初步模拟了人类的主动认知与推理能力，并完成了一些高层次的认知任务. 由表 2 可知：在一些训练过程中经常遇到的启发式认知任务上，经过训练后的 MQN、RMQN 和 FRMQN 模型与 DQN 和 DRQN 模型相比，能够取得更好的表现. 在一些未经过训练的任务中，FRMQN 模型表现出了很强的泛化能力. 综上所述，未来 DRL 模型正向模块复杂化、结构多样化、层次深入化的方向发展，目的是更好地模拟人类的主动认知与推理能力. 相对人类所具有的智能水平，目前 DRL 模型仍处于很低级的阶段，但随着计算和存储能力的不断提升以及模型的主动认知和推理能力的不断增强，通用人工智能会得到进一步发展.

9.2 模型无关的情节式控制器

由认知神经科学可知，哺乳动物的学习系统包括两个部分：一部分用于缓慢学习结构化的知识；另一部分用于快速存储感知到的信息，并通过大脑中的海马体结构回放存储信息. 这两部分构成一个完整的学习结构化知识的系统. 基于上述理论，Blundell 等人[84]设计出一种模型无关的情节式控制器（Model-Free Episode Control，MFEC）. 该控制器可以快速存储和回放状态转移序列，并将回放的序列整合到结构化知识的系统中，使得 agent 在面对一些较复杂的时序决策任务时，能够在更短的时间周期内达到人类玩家的水平.

现有的 DRL 算法都需要 agent 与环境经过上千

万次的交互才能到达人类的表现水平．这些方法都需要大量的训练数据，并通过大量的梯度更新步去完善最终的策略参数，因此学习的效率很低. MFEC则提供了一种辅助方法，**能够不断回放一系列动作中**带来**奖励最高的那些**状态转移序列，有效提高了agent 的学习速度．对于大规模状态空间的 DRL 问题，为了减小计算负担，可以通过一定的方式来泛化未访问到的状态动作空间. 因此 MFEC 通过一种非参数化的最佳近邻方法来泛化那些从未被访问过的状态动作对的值函数：

$$\hat{Q}^{EC}(s,a)=\begin{cases}\dfrac{1}{k}\sum_{i=1}^{k}Q^{EC}\left(s^i,a\right), Q(s,a)\notin Q^{EC}\\ Q^{EC}(s,a), \quad \text{其他}\end{cases} \tag{49}$$

其中 $\{s_1,s_2,...,s_k\}$ 表示与状态 s 距离最近的 k 个状态的集合，$Q^{EC}(s,a)$ 表示在状态 s 下执行动作 a 所获得的最大回报.

在 MFEC 中状态动作值函数 $Q^{EC}(s,a)$ 不再是未来累计折扣奖赏的一个估计值，而是在特定状态动作对下一个潜在的最大回报值的估计，并且该估计值是由过去经历的转移序列 (s_t,a_t,r_t,s_{t+1}) 构造的. 因此 MFEC 通过反向经验回放，使 agent 拥有初步的情节式记忆．实验表明，基于 MFEC 方法的 DRL 模型不仅可以在 Atari 2600 游戏中学习到有效策略，还可以在一些 3D 场景下的复杂任务中表现出与人类玩家相媲美的性能.

10　深度强化学习中的探索与利用

在基于视觉感知的 DRL 任务中，agent 在与未知环境交互时面临着如何平衡探索与利用的难题．尤其在一些接近真实场景的复杂问题中，缺乏有效的探索会导致 agent 无法充分利用环境中的许多关键信息，学习不到有效的控制策略．随着 DRL 的快速发展，各种有效的、可扩展的探索方法也相继被提出.

10.1　利用深度预测模型来激励探索

在传统的 DRL 方法中，通常通过 $\varepsilon-greedy$ 策略来平衡 agent 的探索与利用. 然而当 agent 面临一些较复杂的决策任务时，仅通过简单的启发式策略来探索环境中的未知信息是远远不够的．另外，常规的汤普森采样（Thompson sampling）[107]、玻尔兹曼探索（Boltzmann exploration）[108]和贝叶斯探索奖励（Bayesian Exploration Bonuses，BEB）[109]等激励探索方法并不适用于大规模状态空间的 DRL 任务. 针对上述问题，Stadie 等人[110]利用训练过程中不断完善的深度预测模型来评估状态的新颖度，来分配不同状态下的探索奖励.

具体地，构造编码后的状态特征和深度预测模型输出的状态特征之间的均方误差项：

$$e(s_t,a_t)=\left\|\sigma(s_{t+1})-M_\phi(\sigma(s_t),a_t)\right\|_2^2 \tag{50}$$

其中，$\sigma(s_t)$ 和 $\sigma(s_{t+1})$ 表示将高维度的输入状态编码转化为低维度的特征表示，$\sigma(\cdot)$ 表示一个总计 8 层的自动编码器，$M_\phi:\sigma(S)\times A\to\sigma(S)$ 表示参数为 ϕ 的环境动态性预测模型，该模型用于预测下一状态的特征，其网络结构为 3 层的全连接网络．将之前所有时刻计算出的误差项归一化：

$$\hat{e}(s_t,a_t)=\frac{e(s_t,a_t)}{\max_{a_t}e(s_t,a_t)} \tag{51}$$

根据归一化后的误差项，得到一个衡量状态新颖度的函数：

$$N(s_t,a_t)=\frac{\hat{e}(s_t,a_t)}{t*C} \tag{52}$$

其中，C 表示一个延迟常量．将此新颖度函数加到奖赏函数后，得到：

$$R_{Bonus}(s,a)=R(s,a)+\beta\left(\frac{\hat{e}(s_t,a_t)}{t*C}\right) \tag{53}$$

由上式可知，误差项 $\hat{e}(s_t,a_t)$ 越大，对应状态 s_t 的新颖度越高，说明 agent 对于该状态的认知不足，需要分配更多的探索奖励来鼓励策略再次访问该状态.

实验表明，将基于深度预测模型的激励探索方法应用到 DRL 模型中，不仅提高了 agent 学习的速度，而且提升了 agent 在复杂游戏任务中的性能.

10.2　通过引导型 DQN 进行深度探索

针对常规探索方法[107-109]不能适用于大规模空间 DRL 任务的问题，Osband 等人[111]提出了引导型深度 Q 网络算法（bootstrapped DQN）．在学习过程中，该算法利用多个分流网络来随机化值函数，临时扩展对状态空间的探索范围．图 13 简单描述

了 bootstrapped DQN 模型的结构.

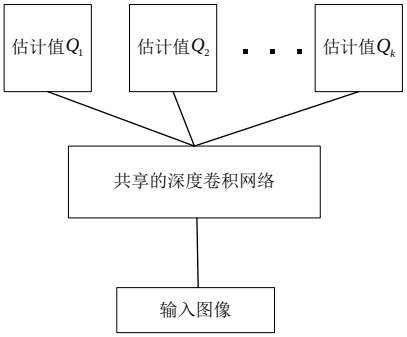

图 13 bootstrapped DQN 模型的结构

如图 13 所示，bootstrapped DQN 算法在通过深度卷积网络处理之后，在线、并行地产生 k 个引导式 Q 值函数的估计值 $Q_1,\ldots,Q_k$，并通过各自的 TD 误差产生对量化值函数不确定性估计的临时扩展. 某个动作对应的值函数不确定性越高，探索该动作带来的效益也越高. 通过这种分布式的深度探索方法，充分保证了 agent 对各种不同策略的探索，产生多样化的样本，使环境的动态信息更好地泛化到未知的状态空间中.

实验表明：一方面通过引导型 DQN 进行深度探索提高了 agent 在面对复杂的 DRL 任务时的学习速度，并在许多 Atari 2600 游戏中表现优异；另一方面，引导型 DQN 的性能明显优于利用深度预测模型来激励探索的启发式方法. 由于 bootstrapped DQN 在网络模型中分流了多个值函数的支路，增加了网络的计算负担.

10.3 基于状态“伪”访问次数的内在激励

在 RL 中，一般是通过经验样本的访问次数来衡量状态（或状态动作对）的不确定性. 当某个状态被访问的次数越少时，其新颖度越高，此时需要给该状态分配更多的探索奖励. 然而某些复杂的 DRL 任务的状态空间十分庞大，此时基于经验的激励探索方法并不适用. 这是因为 agent 无法访问到复杂环境（例如著名的 Montezuma's revenge）中的所有状态. 如何定量表示这些从未被访问过的状态的新颖度，成为解决这类复杂问题的突破口.

Bellemare 等人[112]使用序列密度模型生成各未知状态的“伪”访问次数（pseudo-count）. 例如，一个乘客每天早上出发前，都要观察 3 个因素：天气（$x^1\in$ *Rain*，*Sun*），时间（$x^2\in$ *Early*，*Late*），拥挤程度（$x^3\in$ *Busy*，*Quiet*）. 假设该乘客有 10 个观察：$s_1=(Sun,Late,Quiet)$，$s_i=(Rain,Early,Busy)$，其中 $i=2,\ldots,10$. 现在要衡量从未被观察过的状态 $s_{novel}=(Sun,Late,Busy)$ 的不确定性.

通过如下所表示的序列密度模型来近似估计 s_{novel} 被访问到的概率：

$$\rho_n(s)=\prod_{i=1}^{k}\mu(s^i;s^i_{1:n}) \tag{54}$$

其中，$\mu(\square;s^i_{1:n})$ 是有关状态表示中第 i 个因素的边缘经验分布. 尽管状态 s_{novel} 的访问次数是 $N_n(s_{novel})=0$，但通过序列密度模型依旧可以为该状态分配非零的概率 $\rho_n(s_{novel})=0.1^2\times0.9>0$. 基于此概率生成一个合理的“伪”访问次数来替代基于经验统计的访问次数 N_n. 通过序列密度模型可得：

$$\rho'_n(s)=\Pr_\rho(S_{n+2}=s_{novel}\mid S_1\ldots S_n=s_{1:n},S_{n+1}=s_{novel}) \tag{55}$$

上式表示下一观察是 s_{novel} 的情况下，再次观察到 s_{novel} 的概率大小. 通过简单的推理，可以得到状态 s_{novel} 的“伪”访问次数：

$$\hat{N}_n(s)=\frac{\rho_n(s)(1-\rho'_n(s))}{\rho'_n(s)-\rho_n(s)} \tag{56}$$

例子中 $s_{novel}=(Sun,Late,Busy)$ 出现的概率为 $\rho'_n(s_{novel})=(2/11)^2(10/11)\approx0.03$. 因此根据上式求得 s_{novel} 的“伪”访问次数为 $\hat{N}_n(s_{novel})=0.416$.

通过状态的“伪”访问次数来定义探索奖励：

$$R^+_n(s,a)=\beta\left(\hat{N}_n(s)+0.01\right)^{-1/2} \tag{57}$$

其中 $\beta=0.05$. 将该形式的探索奖励添加到奖赏函数中：$R'(s,a)=R(s,a)+R^+_n(s,a)$，并利用蒙特卡洛回报与 Q 值之间的差值来定义一种新的误差项：

$$\Delta Q_M(s_t,a_t)=\sum_{k=0}^{\infty}\gamma^k R'(s_{t+k},a_{t+k})-Q(s_t,a_t) \tag{58}$$

将 DDQN 算法的误差项 $\Delta Q_{DDQN}(s_t,a_t)$ 和新的误差项相结合：

$$\Delta Q(s_t,a_t)=(1-\eta)\Delta Q_{DDQN}(s_t,a_t)+\eta\Delta Q_M(s_t,a_t) \tag{59}$$

其中，η 表示调节因子，用于控制两个误差项的相对权重. 通过式（59）来构造误差函数，并用梯度下降法来训练 agent. 实验表明，在训练中加入基于状态“伪”访问次数的内在激励，能显著提高 agent 在面对高难度任务时的探索力度，并获得了优异的

性能.

另外，Junhyuk 等人[26]提出了一种基于高斯核（Gaussian kernel）的预测模型. 该模型可预测训练过程中不同视频帧被访问的频率. 在视频游戏任务中，agent 每执行一个动作，都会转移到下一个视频帧. 通过这种方式，可以大幅度降低 agent 对未知、新颖状态的不确定性，从而提升模型解决复杂 DRL 问题的能力. Houthooft 等人[113]提出了变分信息最大化的探索方法（Variational Information Maximizing Exploration，VIME）. 该方法通过贝叶斯神经网络（Bayesian neural networks）中的变分推理（variational inference）来近似环境动态模型的后验概率，并将该动态模型所带来的信息增益作为内在的奖赏，用来激励 agent 对未知领域的探索. 实验表明，在许多连续动作的控制问题中，VIME 方法都具有比传统的启发式探索方法更好的性能. Kulkarni 等人[82]基于一种内在驱动（intrinsic motivation）方法提出了 h-DQN 模型. 在一定程度上，该模型也可以提高 agent 在复杂 DRL 任务中的探索效率.

11 深度强化学习的应用

在 DRL 发展的最初阶段，DQN 算法主要被应用于 Atari 2600 平台中的各类 2D 视频游戏中. 随后，研究人员分别从算法和模型两方面对 DQN 进行了改进，使得 agent 在 Atari 2600 游戏中的平均得分提高了 300%，并在模型中加入记忆和推理模块，成功地将 DRL 应用场景拓宽到 3D 场景下的复杂任务中. AlphaGo 围棋算法结合深度神经网络和 MCTS，成功地击败了围棋世界冠军. 此外，DRL 在机器人控制、计算机视觉、自然语言处理和医疗等领域的应用也都取得了一定的成功.

11.1 深度强化学习在机器人控制领域的应用

在 2D 和 3D 的模拟环境中，基于策略梯度的 DRL 方法（TRPO、GAE、SVG、A3C 等）实现了对机器人的行为控制. 另外，在现实场景下的机器人控制任务中，DRL 也取得了若干研究成果. Levine 等人[69]利用深度 CNN 来近似表示策略，并采用引导式策略搜索来指导机器人完成一些简单的操作. Zhang 等人[114]基于内部存储的引导式策略搜索完成了一些机器人的操作和导航任务. Zhang 等人[115]还利用 DQN 技术对 3 个关节的机械手臂进行端对端的控制. Levine 等人[116]使用深度神经网络模型来预测机器人的动作，并在基于手眼协调（hand-eye coordination）的机器人抓取任务中取得了不错的效果. Finn 等人[117]将 DRL 与逆最优控制（inverse optimal control）方法相结合，完成了一些现实场景下对机器人行为的操控. Lenz 等人[118]使用一种在线、实时的模型预测控制方法，完成了机器人的食品加工任务.

然而在真实场景下机器人的训练数据十分缺乏，上述工作几乎都是通过一些启发式的 DRL 方法来训练单个机器人，以完成相对比较简单的任务. 随着并行计算能力的提升，多个机器人的协同学习逐渐成为主流. Gu 等人[119]利用多线程技术来收集不同机器人的训练数据，通过训练不断地将网络参数传递给每个机器人，以用于下一轮的采样. 这种并行采样及训练的方式，在一定程度上缓解了真实场景下缺失训练数据的问题，并在没有任何人工干涉的情况下让机器人学会了复杂的开门任务. Yahya 等人[120]提出了异步的引导式策略搜索算法（synchronous guided policy search）. 该算法过程可以描述为：多个机器人在面对不同的场景时，分别利用局部策略来优化各自的行为. 然后每个机器人并行地将各自的训练数据上传到服务器端，并在服务器上监督学习全局的策略网络，更好地优化各自机器人的局部策略. 通过这种多 Agent 协同学习的方式，大大缩短了训练的时间，并在一些真实场景下的机器人操纵任务上拥有了更好的泛化能力. 得益于云计算技术的日益成熟，基于服务器端的多机器人协同学习逐渐成为一种发展趋势. 随着计算能力和训练数据量的不断提升，融合了 DRL 方法的智能机器人，必然会在生产和生活中扮演更加重要的角色.

11.2 深度强化学习在计算机视觉领域的应用

基于视觉感知的 DRL 模型可以在只输入原始图像的情况下，输出当前状态下所有可能动作的预测回报. 因此可以将 DRL 模型应用到基于动作条件的视频预测（action-conditional video prediction）任务中. Junhyuk 等人[26]通过 DRL 模型控制动作的输入，完成了高维度视频图像的长期预测任务. 另外，Caicedo 等人[27]结合使用预训练后的 CNN 和 DQN 模型，并通过简单的动作变换来识别候选区域中目标对象的正确位置，完成了一系列目标定位（object localization）的任务. Zhu 等人[121]构造出了基于残差网络的深度孪生行动者评论家模型

（deep siamese actor-critic model）．针对不同的任务，该模型可以同时接收观察图像和目标图像作为输入，并通过 A3C 算法来训练网络参数．

11.3 深度强化学习在自然语言处理领域的应用

最近，用于对话生成（dialogue generation）的神经网络模型[122-123]取得了不错的进展，这些模型可以自动地生成应答的语句．然而这些网络模型存在明显的缺陷：它们只考虑如何根据当前语境生成下一时刻的响应，而忽略了该响应对未来对话产生的后果．因此可以利用 DRL 方法来衡量对话生成中的一些指标．这种结合了 DRL 的对话生成模型具有两项基本要求：（1）引入由人工定义的奖惩机制，从而能够更好地模拟和完成对话；（2）充分衡量对话过程中生成响应的长期影响．基于上述思想，Li 等人[124]分别对对话生成中的易被响应程度、信息性和语义连贯性这 3 个评价指标构造出具体的奖励函数．首先，易于响应是维持一段长久对话的关键因素之一，它保证了一段对话可以更好地向前发展．当对话中出现一些意义不大的响应时，会阻碍对话的进一步发展．此时，对于对话中出现的无意义回答，可以通过一个负的奖励来对其进行惩罚．该种奖惩机制通过自我强化学习，大幅度降低了对话系统产生无意义回答的概率．另外，如果某些应答能开辟出新的话题，那么这段对话就是可持续发展的．因此可以采用信息性来衡量前后两句对话的相似度，并且前后两句对话的相似度越低，在学习系统中设置的奖赏值越大．为了保证信息性而设置的奖励很容易使得系统产生与当前语义无关的各种响应，因此必须将响应和先前对话状态的互信息设置为另一个奖励因子，以保证系统能够产生语义连贯的对话．结合上述 3 个指标可以建立一种较准确的评价对话质量的标准，并利用 DRL 中的策略梯度方法训练对话模型，最终使模型生成更具连贯性、交互性和持续响应的一系列对话．

另外，Guo 等人[125]以目标句子和能产生最大效益的解码序列的相似度来确定当前时刻获得的奖赏值，通过一种带有 LSTM 记忆单元的 DQN 模型成功地解决了文本分析（text analytic）和文本生成（text generation）等问题．Satija 等人[126]将神经机器翻译机（neural machine translation）和 DQN 模型相结合，实现了实时的机器翻译．Narasimhan 等人[127]首次将 DRL 模型用于一种文本类游戏中．

目前，DRL 方法已经成功地应用于文本分析、对话生成、机器翻译、文本游戏等领域，表明 DRL 在自然语言处理（Natural Language Processing，NLP）领域存在广泛的应用前景．

11.4 深度强化学习在参数优化中的应用

在传统的神经网络中，一般通过梯度下降法来优化网络的参数．然而训练中反复调整学习率需要耗费大量的人力和物力资源．因此在训练之前，如果能通过某种学习机制，根据具体问题自动确定相应的学习率，将极大地提升模型的训练效率．针对此问题，Hansen 等人[24]使用 DQN 模型来控制优化超参数的过程，提出了一种基于 Q 值的梯度下降（Q-gradient descent）方法．该方法可以根据不同的任务自动学习相应的学习率．Andrychowicz 等人[25]提出了一种 agent 自我学习模型．该模型通过训练一个神经网络来学习优化其它神经网络的参数．此外，谷歌利用 DRL 算法来优化数据中心服务器群的参数设置，并节省了 40%的电力能源．综上，DRL 在优化参数方向的应用暂时还处于初步阶段，但可以预见，未来通过 DRL 模型自动学习超参数的方法必定会广泛应用于各类优化任务中．

11.5 深度强化学习在博弈论领域的应用

求解博弈论问题一直是人工智能领域的难题，早期在状态规模不大的博弈任务中，基于先验领域知识的求解方法取得了一定的成功．然而在一些复杂的博弈问题中，通过人工事先构造出相关任务的抽象表达，利用该表达来求解纳什均衡（Nash equilibrium）是相当困难的．

DRL 的不断发展为求解博弈论问题开辟了一条新的道路．深度卷积网络具有自动学习高维输入数据抽象表达的功能，可以有效解决复杂任务中领域知识表示和获取的难题．目前，利用 DRL 技术来发展博弈论已经取得了不错的研究成果．Heinrich 等人[128]提出了一种称为神经虚拟自我对局（Neural Fictitious Self-Play，NFSP）的博弈方法．在没有任何先验知识的前提下，该方法将 DRL 和虚拟自我对局技术向结合，并使用端对端的方式从自我博弈中学习求解问题的近似纳什均衡．实验表明，NFSP 通过自我学习成功掌握了玩德州扑克游戏的技巧，其表现已经接近人类专家的水平．其他的相关工作包括 Heriberto 等人[129]利用 DRL 模型来优化策略型对话式 agent 的行为，并在一种经典的对话式桌游中表现优异．

12 结束语

DRL 作为当前人工智能领域最热门的研究方向之一，已经吸引了越来越多学术界和工业界人士对其进行不断地研究与发展．本文详述了 DRL 当前的研究现状和发展趋势，介绍了基于值函数、策略梯度、搜索与监督 3 大类 DRL 方法．这 3 类 DRL 方法可以成功解决众多具有挑战性的问题，比如视频游戏、围棋和机器人的操纵等．然后介绍了几个 DRL 前沿的研究方向，包括分层 DRL、多任务迁移 DRL、多 agent 的 DRL、基于记忆与推理的 DRL 以及 DRL 中探索与利用的平衡问题．从中可以发现 DRL 技术正向更加通用、灵活、智能的方向发展，这体现在：（1）通过分层 DRL 方法可以将复杂、困难的整体任务分解为若干规模较小的子任务；（2）多任务迁移 DRL 的研究，通过训练单独的模型来完成多个任务变得可能；（3）多 agent 的 DRL 模型已经能够初步应对一些需要合作、竞争与通信的难题；（4）通过向 DRL 模型中加入外部的记忆组件，使得 agent 具有了初步的主动认知与推理能力；（5）DRL 模型正在尝试模拟人类大脑辅助学习系统的工作方式，以构造一个可以自主记忆、学习和决策的 agent；（6）DRL 模型在复杂场景中的探索效率正逐步提高，并在一些高难度的任务中取得了不错的表现；最后，本文介绍了 DRL 技术几个重要的实际应用．

综上所述，各类 DRL 方法的成功主要得益于大幅度提升的计算能力和训练数据量．本质上，这些 DRL 算法还不具备如人类般的自主思考、推理与学习能力．为了进一步接近通用人工智能的终极目标，未来 DRL 会朝着如下的几个方向发展：（1）更加趋于通过增量式、组合式的学习方式来训练 DRL 模型；（2）无监督的生成模型（generative model）将会在 DRL 方法中扮演更加重要的角色；（3）开发出完备和高效的计算图模型（computational graph），以方便地向 DRL 网络中加入注意力机制、记忆单元、反馈控制等辅助结构；（4）在 DRL 模型中整合不同种类的记忆单元，如 LSTM、内存堆栈记忆、NTM 等模型，使得 agent 的记忆功能更加趋于完善，以提高其主动推理与认知的能力；（5）进一步加强认知神经科学对 DRL 的启发，使 agent 逐渐掌握如人类大脑所拥有的记忆、聚焦、规划和学习等功能；（6）迁移学习将会被更多地应用到 DRL 方法中，以缓解真实任务场景中训练数据缺乏的问题；（7）借助于云服务器端的多 agent 协同学习将成为一种新趋势；（8）迁移学习、协同学习和目标驱动等方法使 DRL 模型的通用性更好．可以预见的是，随着 DRL 理论和方法研究的不断深入，人类将会在不久的将来实现 DeepMind 提出的“解决智能，并用智能解决一切”的理想目标．

参考文献

[1] Yu Kai, Jia Lei, Chen Yu-Qiang, Xu Wei. Deep learning: yesterday, today, and tomorrow. Journal of Computer Research and Development, 2013, 50(9): 1799-1804 (in Chinese)
(余凯, 贾磊, 陈雨强, 徐伟. 深度学习的昨天、今天和明天. 计算机研究与发展, 2013, 50(9): 1799-1804)

[2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks//Proceedings of the 26th Annual Conference on Neural Information Processing Systems. Nevada, USA, 2012: 1097-1105

[3] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115(3): 211-252

[4] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks//Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing. Vancouver, Canada, 2013: 6645-6649

[5] Li Ya-Xiong, Zhang Jian-Qiang, Pan Deng, Hu Dan. A study of speech recognition based on RNN-RBM language model. Journal of Computer Research and Development, 2014, 51(9): 1936-1944 (in Chinese)
(黎亚雄, 张坚强, 潘登, 胡惮. 基于RNN-RBM语言模型的语音识别研究. 计算机研究与发展, 2014, 51(9): 1936-1944)

[6] Cho K, Merrienboer B V, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation//Proceedings of Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, 2014: 1724-1734

[7] Yang Zhao, Tao Da-Peng, Zhang Shu-Ye, Jin Lian-Wen. Similar handwritten Chinese character recognition based on deep neural networks with big data. Journal on Communications, 2014, 35(9): 184-189 (in Chinese)
(杨钊, 陶大鹏, 张树业,等. 大数据下的基于深度神经网的相似汉字识别. 通信学报, 2014, 35(9): 184-189)

[8] Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks//Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition. Columbus, USA, 2014: 1725-1732

[9] Sun Zhi-Jun, Xue Lei, Xu Yang-Ming, Wang Zheng. Overview of deep learning. Application Research of Computers, 2012, 29(8): 2806-2810 (in Chinese)
(孙志军, 薛磊, 许阳明, 王正. 深度学习研究综述. 计算机应用研究, 2012, 29(8): 2806-2810）

[10] Gao Yang, Zhou Ru-Yi, Wang Hao, Cao Zhi-Xin. Study on an average reward reinforcement learning algorithm. Chinese Journal of Computers, 2007, 30(8): 1372-1378 (in Chinese)
(高阳, 周如益, 王皓, 曹志新. 平均奖赏强化学习算法研究. 计算机学报, 2007, 30(8): 1372-1378)

[11] Fu Qi-Ming, Liu Quan, Wang Hui, Xiao Fei, Yu Jun, Li Jiao. A novel off policy Q(λ) algorithm based on linear function approximation. Chinese Journal of Computers, 2014, 37(3): 677-686 (in Chinese)
(傅启明, 刘全, 王辉, 肖飞, 于俊, 李娇. 一种基于线性函数逼近的离策略q(λ)算法. 计算机学报, 2014, 37(3): 677-686)

[12] Kober J, Peters J. Reinforcement learning in robotics: a survey. International Journal of Robotics Research, 2013, 32(11): 1238-1274

[13] Wei Ying-Zi, Zhao Ming-Yang. A reinforcement learning-based approach to dynamic job-shop scheduling. Acta Automatica Sinica, 2005, 31(5): 765-771 (in Chinese)
(魏英姿, 赵明扬. 一种基于强化学习的作业车间动态调度方法. 自动化学报, 2005, 31(5): 765-771)

[14] Ipek E, Mutlu O, Martinez J F, et al. Self-optimizing memory controllers: a reinforcement learning approach. Computer Architecture, 2008, 36(3): 39-50

[15] Tesauro G. TD-Gammon, a self-teaching backgammon program, achieves master-level play. Neural Computation, 1994, 6(2): 215-219

[16] Kocsis L, Szepesvári C. Bandit based Monte-Carlo planning//Proceedings of the 17th European Conference on Machine Learning. Berlin, Germany, 2006: 282-293

[17] Sutton R S, Barto A G. Reinforcement learning: an introduction. Cambridge: MIT press, 1998

[18] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning//Proceedings of Workshops at the 26th Neural Information Processing Systems 2013. Lake Tahoe, USA, 2013:201-220

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

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

[21] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. Computer Science, 2016, 8(6): A187

[22] Duan Y, Chen X, Houthooft R, et al. Benchmarking deep reinforcement learning for continuous control//Proceedings of the 32nd International Conference on Machine Learning. New York, USA, 2016: 1329-1338

[23] Gu S, Lillicrap T, Sutskever I, et al. Continuous deep q-learning with model-based acceleration//Proceedings of the 32nd International Conference on Machine Learning. New York, USA, 2016: 2829-2838

[24] Hansen S. Using deep q-learning to control optimization hyperparameters. arXiv preprint arXiv:1602.04062, 2016

[25] Andrychowicz M, Denil M, Gomez S, et al. Learning to learn by gradient descent by gradient descent//Proceedings of the Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 3981-3989

[26] Oh J, Guo X, Lee H, et al. Action-conditional video prediction using deep networks in atari games//Proceedings of the Neural Information Processing Systems. Montreal, Canada, 2015: 2863-2871

[27] Caicedo J C, Lazebnik S. Active object localization with deep reinforcement learning//Proceedings of the International Conference on Computer Vision. Santiago, Chile, 2015: 2488-2496

[28] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436-444

[29] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527-1554

[30] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders//Proceedings of the 25th Annual International Conference on Machine Learning. New York, USA, 2008: 1096-1103

[31] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 2010, 11(6): 3371-3408

[32] Sutskever I, Hinton G E, Taylor G W. The recurrent temporal restricted boltzmann machine//Proceedings of the 27th Annual Conference on Neural Information Processing Systems. Vancouver, Canada, 2009: 1601-1608

[33] Hinton G. A practical guide to training restricted Boltzmann machines. Momentum, 2010, 9(1): 926

[34] Lee H, Grosse R, Ranganath R, et al. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations//Proceedings of the 26th International Conference on Machine Learning. Montreal, Canada, 2009: 609-616

[35] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network

based language model//Proceedings of the Conference of International Speech Communication Association. Chiba, Japan, 2010: 1045-1048

[36] Gregor K, Danihelka I, Graves A, et al. DRAW: A recurrent neural network for image generation//Proceedings of the International Conference on Machine Learning. Lille, France, 2015: 1462-1471

[37] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014

[38] He K, Zhang X, Ren S, et al. Delving deep into rectifiers: surpassing human-level performance on imagenet classification//Proceedings of the IEEE International Conference on Computer Vision. Boston, USA, 2015: 1026-1034

[39] Lin M, Chen Q, Yan S. Network in network//Proceedings of the International Conference on Learning Representations. Banff, Canada, 2014

[40] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA, 2015: 1-9

[41] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 770-778

[42] Szegedy C, Ioffe S, Vanhoucke V. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261, 2016

[43] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks//Proceedings of the European Conference on Computer Vision. Amsterdam, Netherlands, 2016: 630-645

[44] Vinyals O, Toshev A, Bengio S, et al. Show and tell: a neural image caption generator//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA, 2015: 3156-3164

[45] Xu K, Ba J, Kiros R, Courville A, Salakhutdinov R, Zemel R, Bengio Y. Show, attend and tell: neural image caption generation with visual attention//Proceedings of the International Conference on Machine Learning. Lille, France, 2015: 2048-2057

[46] Tsitsiklis J N, Van R B. An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control, 1997, 42(5): 674-690

[47] Riedmiller M. Neural fitted q iteration-first experiences with a data efficient neural reinforcement learning method//Proceedings of the Conference on Machine Learning. Berlin, German, 2005: 317-328

[48] Lange S, Riedmiller M. Deep auto-encoder neural networks in reinforcement learning//Proceedings of the 7th International Joint Conference on Neural Networks. Barcelona, Spain, 2010: 1-8

[49] Abtahi F, Fasel I. Deep belief nets as function approximators for reinforcement learning. Frontiers in Computational Neuroscience, 2011, 5(1): 112-131

[50] Lange S, Riedmiller M, Voigtlander A. Autonomous reinforcement learning on raw visual input data in a real world application//Proceedings of the 9th International Joint Conference on Neural Networks. Brisbane, Australia, 2012: 1-8

[51] Koutník J, Schmidhuber J, Gomez F. Online evolution of deep convolutional network for vision-based reinforcement learning//Proceedings of the International Conference on Simulation of Adaptive Behavior. New York, USA, 2014: 260-269

[52] Watkins C J C H. Learning from delayed rewards. Robotics & Autonomous Systems, 1989, 15(4): 233-235

[53] Lin L J. Reinforcement learning for robots using neural networks. USA: Defense Technical Information Center, DTIC Technical Report: ADA261434, 1993

[54] Van H V, Guez A, Silver D. Deep reinforcement learning with double q-learning//Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 2094-2100

[55] Hasselt H V. Double q-learning//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2010: 2613-2621

[56] Bellemare M G, Ostrovski G, Guez A, et al. Increasing the action gap: new operators for reinforcement learning//Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 1476-1483

[57] Baird III L C. Reinforcement learning through gradient descent. Carnegie Mellon University, USA, 1999

[58] Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay//Proceedings of the 4th International Conference on Learning Representations. San Juan, Puerto Rico, 2016:322-355

[59] Lakshminarayanan A S, Sharma S, Ravindran B. Dynamic frame skip deep q network//Proceedings of the Workshops at the International Joint Conference on Artificial Intelligence. New York, USA, 2016

[60] Hasselt H V, Guez A, Hessel M, et al. Learning functions across many orders of magnitudes//Proceedings of the Advances in Neural Information Processing Systems. Barcelona, Spain, 2016:80-99

[61] François-Lavet V, Fonteneau R, Ernst D. How to discount deep reinforcement learning: towards new dynamic strategies// Proceedings of the Workshops at the Advances in Neural Information Processing Systems. Montreal, Canada, 2015:107-116

[62] Wang Z, Freitas N D, Lanctot M. Dueling network architectures for deep reinforcement learning//Proceedings of the International Conference on Machine Learning. New York, USA, 2016: 1995-2003

[63] Hausknecht M, Stone P. Deep recurrent q-learning for partially

observable MDPs. arXiv preprint arXiv:1507.06527, 2015

[64] Sutton R S, Mcallester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation//Proceedings of the Advances in Neural Information Processing Systems. Denver, USA, 1999: 1057-1063

[65] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 1992, 8(3-4): 229-256

[66] Hafner R, Riedmiller M. Reinforcement learning in feedback control. Machine Learning, 2011, 84(1-2): 137-169

[67] Schulman J, Moritz P, Levine S, et al. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438, 2015

[68] Schulman J, Levine S, Moritz P, et al. Trust region policy optimization//Proceedings of the International Conference on Machine Learning. Lugano, Switzerland, 2015: 1889-1897

[69] Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies. Journal of Machine Learning Research, 2016, 17(39): 1-40

[70] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms//Proceedings of the International Conference on Machine Learning. Beijing, China, 2014: 387-395

[71] Heess N, Wayne G, Silver D, et al. Learning continuous control policies by stochastic value gradients//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2015: 2944-2952

[72] Rezende D J, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models//Proceedings of the International Conference on Machine Learning. Beijing, China, 2014: 1278-1286

[73] Balduzzi D, Ghifary M. Compatible value gradients for reinforcement learning of continuous deep policies. Computer Science, 2015, 8(6): A187

[74] Peng X B, Berseth G, van de Panne M. Terrain-adaptive locomotion skills using deep reinforcement learning. ACM Transactions on Graphics, 2016, 35(4): 81

[75] Heess N, Hunt J J, Lillicrap T P, et al. Memory-based control with recurrent neural networks//Proceedings of the Workshops of Advances in Neural Information Processing Systems. Montreal, Canada, 2015:301-312

[76] Hausknecht M, Stone P. Deep reinforcement learning in parameterized action space. arXiv preprint arXiv:1511.04143, 2015

[77] Schulman J, Heess N, Weber T, et al. Gradient estimation using stochastic computation graphs//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2015: 3528-3536

[78] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning//Proceedings of the International Conference on Machine Learning. New York, USA, 2016: 1928-1937

[79] Coulom R. Efficient selectivity and backup operators in Monte-Carlo tree search//Proceedings of the International Conference on Computers and Games. Berlin, Germany, 2006: 72-83

[80] Barto A G, Mahadevan S. Recent advances in hierarchical reinforcement learning. Discrete Event Dynamic Systems, 2003, 13(4): 341-379

[81] Sutton R S, Precup D, Singh S. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 1999, 112(1): 181-211

[82] Kulkarni T D, Narasimhan K R, Saeedi A, et al. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation//Proceedings of the Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 3675-3683

[83] Rezende D J, Mohamed S, Danihelka I, et al. One-shot generalization in deep generative models//Proceedings of the 33rd International Conference on Machine Learning. New York, USA, 2016: 1521-1529

[84] Blundell C, Uria B, Pritzel A, et al. Model-free episodic control. arXiv preprint arXiv:1606.04460, 2016

[85] Krishnamurthy R, Lakshminarayanan A S, Kumar P, et al. Hierarchical reinforcement learning using spatio-temporal abstractions and deep neural networks. arXiv preprint arXiv:1605.05359, 2016

[86] Weber M, Rungsarityotin W, Schliep A. Perron cluster analysis and its connection to graph partitioning for noisy data. Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2004

[87] Dietterich T G. Hierarchical reinforcement learning with the MAXQ value function decomposition. J. Artif. Intell. Res, 2000, 13: 227-303

[88] Kulkarni T D, Saeedi A, Gautam S, et al. Deep successor reinforcement learning. arXiv preprint arXiv:1606.02396, 2016

[89] Wilson A, Fern A, Ray S, et al. Multi-task reinforcement learning: a hierarchical Bayesian approach//Proceedings of the International Conference on Machine Learning. Corvallis, USA, 2007: 1015-1022

[90] Li H, Liao X, Carin L. Multi-task reinforcement learning in partially observable stochastic environments. Journal of Machine Learning Research, 2009, 10(3): 1131-1186

[91] Taylor M E, Stone P. Behavior transfer for value-function-based reinforcement learning//Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems. Utrecht, Netherlands, 2005: 53-59

[92] Fernández F, Veloso M. Probabilistic policy reuse in a reinforcement learning agent//Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems. Istanbul, Turkey, 2015: 720-727

[93] Wang Hao, Gao Yang, Chen Xing-Guo. Transfer of reinforcement learning: the state of the art. Chinese Journal of Electronics, 2008, 36(s1): 39-43 (in Chinese)
(王皓, 高阳, 陈兴国. 强化学习中的迁移: 方法和进展. 电子学报, 2008, 36(s1): 39-43)

[94] Parisotto E, Ba J L, Salakhutdinov R. Actor-mimic: deep multitask and transfer reinforcement learning//Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2016:156-171

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

[96] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network//Proceedings of the Workshops of Advances in Neural Information Processing Systems. Montreal, Canada, 2014:200-208

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

[98] Schaul T, Horgan D, Gregor K, et al. Universal value function approximators//Proceedings of the 32nd International Conference on Machine Learning. Lugano, Switzerland, 2015: 1312-1320

[99] Tessler C, Givony S, Zahavy T, et al. A deep hierarchical approach to lifelong learning in Minecraft. arXiv preprint arXiv:1604.07255, 2016

[100] Tampuu A, Matiisen T, Kodelja D, et al. Multi-Agent cooperation and competition with deep reinforcement learning. arXiv preprint arXiv:1511.08779, 2015

[101] Foerster J N, Assael Y M, Freitas N D, et al. Learning to communicate to solve riddles with deep distributed recurrent q-networks. arXiv preprint arXiv:1602.02672, 2016

[102] Graves A, Wayne G, Danihelka I. Neural turing machines. arXiv preprint arXiv:1410.5401, 2014

[103] Sukhbaatar S, Weston J, Fergus R. End-to-end memory networks//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2015: 2440-2448

[104] Lake B M, Ullman T D, Tenenbaum J B, et al. Building machines that learn and think like people. arXiv preprint arXiv:1604.00289, 2016

[105] Oh J, Chockalingam V, Singh S, et al. Control of memory, active perception, and action in Minecraft//Proceedings of the International Conference on Machine Learning. New York, USA, 2016: 2790-2799

[106] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines//Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel, 2010: 807-814

[107] Chapelle O, Li L. An empirical evaluation of thompson sampling//Proceedings of the Advances in Neural Information Processing Systems. Granada, Spain, 2011: 2249-2257

[108] Kocsis L, Szepesvári C. Bandit based monte-carlo planning//Proceedings of the European Conference on Machine Learning. Berlin, German, 2006: 282-293

[109] Kolter J Z, Ng A Y. Near-Bayesian exploration in polynomial time//Proceedings of the 26th Annual International Conference on Machine Learning. Montreal, Canada, 2009: 513-520

[110] Stadie B C, Levine S, Abbeel P. Incentivizing exploration in reinforcement learning with deep predictive models. arXiv preprint arXiv:1507.00814, 2015

[111] Osband I, Blundell C, Pritzel A, et al. Deep exploration via bootstrapped DQN//Proceedings of the Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 4026-4034

[112] Bellemare M G, Srinivasan S, Ostrovski G, et al. Unifying count-based exploration and intrinsic motivation//Proceedings of the Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 1471-1479

[113] Houthooft R, Chen X, Duan Y, et al. Curiosity-driven exploration in deep reinforcement learning via Bayesian neural networks. arXiv preprint arXiv:1605.09674, 2016

[114] Zhang M, McCarthy Z, Finn C, et al. Learning deep neural network policies with continuous memory states//Proceedings of the International Conference on Robotics and Automation. Stockholm, Sweden, 2016: 520-527

[115] Zhang F, Leitner J, Milford M, et al. Towards vision-based deep reinforcement learning for robotic motion control//Proceedings of the Australasian Conference on Robotics and Automation. Canberra, Australia, 2015

[116] Levine S, Pastor P, Krizhevsky A, et al. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. arXiv preprint arXiv:1603.02199, 2016

[117] Finn C, Levine S, Abbeel P. Guided cost learning: deep Inverse optimal control via policy optimization//Proceedings of the International Conference on Machine Learning. New York, USA, 2016: 49-58

[118] Lenz I, Knepper R, Saxena A. Deepmpc: learning deep latent features for model predictive control//Proceedings of the Robotics Science and Systems. Rome, Italy, 2015:201-209

[119] Gu S, Holly E, Lillicrap T, Levine S. Deep reinforcement learning for robotic manipulation. arXiv preprint arXiv:1610.00633, 2016

[120] Yahya A, Li A, Kalakrishnan M, et al. Collective robot reinforcement learning with distributed asynchronous guided policy search. arXiv preprint arXiv:1610.00673, 2016

[121] Zhu Y, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning. arXiv preprint arXiv:1609.05143, 2016

[122] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2014: 3104-3112

[123] Sordoni A, Galley M, Auli M, et al. A neural network approach to context-sensitive generation of conversational responses// Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies. Denver, USA, 2015: 196-205

[124] Li J W, Monroe W, Ritter A, et al. Deep reinforcement learning for dialogue generation//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Austin, USA, 2016: 1192-1202

[125] Guo H. Generating text with deep reinforcement learning//Proceedings of the Workshops of Advances in Neural Information Processing Systems. Montreal, Canada, 2015:1-9

[126] Satija H, Pineau J. Simultaneous machine translation using deep reinforcement learning//Proceedings of the Workshops of International Conference on Machine Learning. New York, USA, 2016:110-119

[127] Narasimhan K, Kulkarni T, Barzilay R. Language understanding for text-based games using deep reinforcement learning//Proceedings of the Conference on Empirical Methods on Natural Language Processing. Lisbon, Portugal, 2015: 1-11

[128] Heinrich J, Lanctot M, Silver D. Fictitious self-play in extensive-form games//Proceedings of the International Conference on Machine Learning. Lugano, Switzerland, 2015: 805-813

[129] Cuayáhuitl H, Keizer S, Lemon O. Strategic dialogue management via deep reinforcement learning//Proceedings of the Workshops of Advances in Neural Information Processing Systems. Montreal, Canada, 2015

LIU Quan, born in 1969, Ph.D., professor, Ph.D. supervisor. His main research interests include reinforcement learning, deep reinforcement learning and automated reasoning.

ZHAI Jian-Wei, born in 1992, Master student. His main research interests include reinforcement learning, deep learning and deep reinforcement learning.

ZHANG Zong-Zhang, born in 1985, Ph.D., associate professor. His research interests include POMDPs, reinforcement learning and multi-agent systems.

ZHOU Qian, born in 1992, Master candidate. Her main research interest is reinforcement learning.

ZHONG Shan, born in 1983，Ph.D. candidate, lecturer. Her research interests include machine learning and deep learning.

ZHANG Peng, born in 1992, Master student. His main research interest is reinforcement learning in continuous space.

XU Jin, born in 1991, Master student. His main research interest is deep reinforcement learning in continuous space.

Background

Deep reinforcement learning (DRL), as a combination of the advantages of rich perception of high-dimensional raw inputs in deep learning and sequential decision making under uncertainty in reinforcement learning, has achieved remarkable successes in a variety of domains, such as Atari games, robotic control, parametric optimization, natural language processing and computer vision, medical treatment, and so on. Our paper describes three main categories of DRL methods, summarizes some cutting-edge research directions of DRL, discusses many practical DRL applications, and highlights some future trends in the field, with the hope of providing a valuable reference in its future development.

This paper is partially supported by National Natural Science Foundation of China (61272005, 61303108, 61373094, 61472262, 61502323, 61502329), Natural Science Foundation of Jiangsu (BK2012616), High School Natural Foundation of Jiangsu (13KJB520020,16KJB520041), Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172014K04), Suzhou Industrial application of basic research program part (SYG201422, SYG201308). These projects aim to enrich the reinforcement-learning theory and develop efficient approximate algorithms to expand the power and applicability of reinforcement learning on large scale problems.