多智能体深度强化学习的若干关键科学问题

孙长银 [1] 穆朝絮 [2]

摘 要 强化学习作为一种用于解决无模型序列决策问题的方法已经有数十年的历史, 但强化学习方法在处理高维变量问题时常常会面临巨大挑战. 近年来, 深度学习迅猛发展, 使得强化学习方法为复杂高维的多智能体系统提供优化的决策策略、在充满挑战的环境中高效执行目标任务成为可能. 本文综述了强化学习和深度强化学习方法的原理, 提出学习系统的闭环控制框架, 分析了多智能体深度强化学习中存在的若干重要问题和解决方法, 包括多智能体强化学习的算法结构、环境非静态和部分可观性等问题, 对所调查方法的优缺点和相关应用进行分析和讨论. 最后提供多智能体深度强化学习未来的研究方向, 为开发更强大、更易应用的多智能体强化学习控制系统提供一些思路.

关键词 强化学习, 深度强化学习, 多智能体, 学习系统, 智能控制, 决策优化

引用格式 孙长银, 穆朝絮. 多智能体深度强化学习的若干关键科学问题. 自动化学报, 2020, **46**(7): 1301−1312

DOI 10.16383/j.aas.c200159

Important Scientific Problems of Multi-Agent Deep Reinforcement Learning

SUN Chang-Yin[1] MU Chao-Xu[2]

Abstract Reinforcement learning has been used to solve sequence decision problems without models for decades. However, it often faces great challenges in dealing with high-dimensional problems. In recent years, with the rapid development of deep learning, it promotes that reinforcement learning can provide the optimized strategy for complex and high-dimensional multi-agent systems to efficiently perform the target tasks in challenging environments. This paper reviews on the principles of reinforcement learning and deep reinforcement learning, puts forward the closed-loop control framework of learning systems, and investigates the existing important problems and corresponding methods for the deep reinforcement learning of multi-agent systems, including multi-agent reinforcement learning algorithmic framework, non-static environment, partially observability, and so on. The merits and drawbacks of these investigated methods are analyzed, and some related applications are summarized. This paper also provides some new insights into various research directions of multi-agent reinforcement learning, and related ideas for better application development in the future.

Key words Reinforcement learning, deep reinforcement learning, multi-agent, learning system, intelligent control, decision optimization

Citation Sun Chang-Yin, Mu Chao-Xu. Important scientific problems of multi-agent deep reinforcement learning. *Acta Automatica Sinica*, 2020, **46**(7): 1301−1312

多智能体系统由多个具有一定传感、计算、执行能力的个体组成, 个体通过网络与其他智能体通信, 相互协作完成任务. 每个智能体具有一定的独立性和自主性, 能够自主学习、推理和规划并选择适当的策略解决子问题. 通过多个具备简单智能的个体相互协作实现复杂的智能, 多智能体系统在降低单个智能体复杂程度的同时, 有效提高了整个系统的鲁棒性、可靠性和灵活性[1−2]. 近年来, 随着通信和网络技术的快速发展, 多智能体系统在交通运输、工业生产等多个领域都有广泛和深入的应用. 面对越来越多的大规模复杂问题, 单智能体集成的解决方案将面临各种资源和条件的限制. 如何开发具有群体智能的多智能体系统, 高效优化的完成任务, 是人工智能和自动化领域面临的新的挑战[3−4].

伴随着计算和存储能力的大幅提升, 深度学习在人工智能领域获得了巨大的成功. 在此背景下, 产生了由深度学习和强化学习结合的深度强化学习 (Deep reinforcement learning, DRL)[5]. 深度强化学习将感知、学习、决策融合到同一框架, 实现了从原始输入到决策动作 “端到端” 的感知与决策,

收稿日期 2020-03-25 录用日期 2020-05-07
Manuscript received March 25, 2020; accepted May 7, 2020
科技部人工智能专项重大项目 (2018AAA0101400), 国家自然科学基金创新研究群体 (61921004), 国家自然科学基金 (61942301) 资助
Supported by Artificial Intelligence Major Project of the Ministry of Science and Technology of China (2018AAA0101400), National Natural Science Foundation of China for Creative Research Groups (61921004), National Natural Science Foundation of China (61942301)
本文责任编委 贺威
Recommended by Associate Editor HE Wei
1. 东南大学自动化学院 南京 210096 2. 天津大学电气自动化与信息工程学院 天津 300072
1. School of Automation, Southeast University, Nanjing 210096
2. School of Electrical and Information Engineering, Tianjin University, Tianjin 300072

并在游戏领域取得了令人兴奋的成绩. Google DeepMind 团队开发的 AlphaGo 系列围棋程序, 击败了人类顶级围棋选手[6−8]; 提出的深度 Q 网络 (Deep Q-network, DQN), 在多种 Atari 游戏中成功超越人类专业玩家. OpenAI 研发了能够在 Dota2 这一比围棋更复杂的游戏中击败人类专业玩家的游戏机器人[9]. 此外, 深度强化学习在无人驾驶[10]、机器人控制[11]、交通运输调度[12]、电力系统优化[13]、分布式传感网络[14] 以及金融和社会学等领域还有大量的应用研究[15]. 更为重要的是, 深度强化学习可能成为一种解决复杂问题的有效方法, 极大地推动人工智能和自动化技术的发展[16−17].

多智能体深度强化学习 (Multi-agent deep reinforcement learning, MADRL) 将深度强化学习的思想和算法用于多智能体系统的学习和控制中, 是开发具有群体智能的多智能体系统的重要方法. 然而, 深度强化学习方法扩展到多智能体系统, 面临诸多方面的挑战. 本文综述了强化学习和深度强化学习方法的原理, 分析了多智能体深度强化学习算法结构、环境非静态性、部分可观性等重要问题和研究进展, 对多智能深度强化学习方法的应用情况也进行了简要概述. 最后, 讨论了多智能体深度强化学习未来的研究方向和研究思路.

1 强化学习理论

受到生物学习规律的启发, 强化学习以试错机制与环境进行交互, 通过最大化累积奖赏的方式来学习和优化, 最终达到最优策略. 在强化学习中, 定义决策者或学习者为 “学习机”, 将学习机之外的事物定义为 “环境”, 系统与环境相融[18]. 学习机和环境之间的交互过程可以由三个要素来描述, 分别是: 状态 s、动作 a、奖励 r. 学习机根据初始状态 s_0, 执行动作 a_0 并与环境进行交互, 得到奖励 r_1 并获得更新的状态 s_1. 在时间步 t, 根据当前状态 s_t 和奖励 r_t, 学习机提供当前动作 a_t. 接着, 系统状态由 s_t 转变为 s_{t+1}, 与环境交互反馈奖励 r_{t+1}. 强化学习基本原理如图 1 所示.

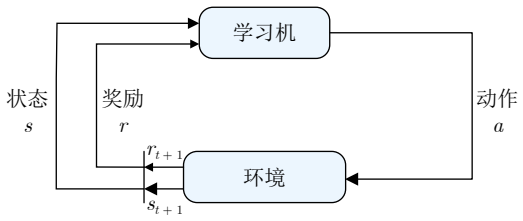

图 1 强化学习的基本原理

Fig. 1 Basic principles of reinforcement learning

一般来说, 强化学习强调和环境的交互, 表示为一系列状态、动作和奖励的序列: $s_0, a_0, r_1, s_1, a_1, r_2, \cdots, s_{n-1}, a_{n-1}, r_n$. 尽管 n 可以趋向于无穷, 但实际上通常定义一个末端状态 $s_n = s_T$ 来对 n 进行限制. 这一串从起始状态开始到末端状态结束的状态、动作和奖励序列称为一个学习周期 (Episode) 或训练周期. 策略通常表示为 π, 是从状态 s 到动作 a 的一个映射. 如果对所有的状态, 在状态为 s 时采取动作 a 的概率 $P(a|s) = 1$, 则这个策略为确定性策略. 反之, 如果对于状态 s, 在该状态下采取动作 a 的概率 $P(a|s) < 1$, 则该策略为随机策略. 在两种情况的任一情况下, 都可以定义策略 π 为一组状态备选动作的概率分布. 在当前时间步, 学习机与环境交互和试错学习, 迭代优化当前策略 π_t, 使新一步的策略 π_{t+1} 优于当前步的策略 π_t. 这个过程被称为 “策略更新”, 在强化学习过程中反复执行, 直到学习机不能寻找到一个更好的策略为止.

在学习机与环境的交互中, 学习机在每个时间步 t 都会得到一个反馈奖励 r_t, 直到末端状态 s_T. 然而每步奖励 r_t 并不能代表长期的奖励收益. 为了表达学习机长期的收益, 引入时间步 t 的回报 G_t:

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{T-t} r_T = \sum_{i=0}^{T-t} \gamma^i r_{t+i} \tag{1}$$

其中, γ 为折扣因子且满足 $0 < \gamma \leq 1$. 当 γ 接近 1 时, 学习机表现为更加重视长期回报. 反之, 当 γ 接近 0 时, 学习机更加重视短期回报. 在实际中, γ 更倾向于被设置接近 1, 使其更关注长期回报.

策略的优劣通常采用值函数来进行表示. 用于评判状态 s 下策略优劣的状态值函数表示为:

$$V_\pi(s) = \mathrm{E}[G_t | s_t = s, \pi] \tag{2}$$

根据该式可以求得最优策略:

$$\pi^* = \arg\max_\pi V_\pi(s) \tag{3}$$

另一种形式的值函数用于评判状态 s 下执行动作 a 的优劣程度, 称为状态–动作值函数, 也称为 Q 函数:

$$Q_\pi(s, a) = \mathrm{E}[G_t | s_t = s, a_t = a, \pi] \tag{4}$$

此时最优策略表示为:

$$\pi^* = \arg\max_a Q_{\pi^*}(s, a) \tag{5}$$

下面给出蒙特卡洛法、时间差分法和策略梯度法三类强化学习算法, 分别从基于值函数和基于策略的角度进行优化.

1.1 蒙特卡洛法

蒙特卡洛法通过重复生成训练周期并且记录在

每个状态或每个状态–动作对的平均回报值的方法来拟合值函数, 状态值函数的计算方法如下:

$$V_{\pi}^{MC}(s)=\lim_{j\to+\infty}\mathrm{E}[G^{j}(s_t)|s_t=s,\pi] \tag{6}$$

式中, $G^{j}(s_t)$表示在第 j 个训练周期中, 在状态 s_t 下观测到的回报值. 类似地, 还可以计算状态–动作值函数:

$$Q_{\pi}^{MC}(s,a)=\lim_{j\to+\infty}\mathrm{E}[G^{j}(s_t,a_t)|s_t=s,a_t=a,\pi] \tag{7}$$

为了使蒙特卡洛方法可以更有效的探索, 在策略更新中常采用 ϵ-贪婪的方法进行探索. 虽然蒙特卡洛法不需要任何系统状态转移概率的信息, 但为保证这种方法能够最终收敛, 还需要满足两个条件: 1) 足够多的训练周期; 2) 每个状态和状态下的每个动作都应被达到和执行过一定次数.

1.2　时间差分学习法

时间差分学习与蒙特卡洛法相同, 从环境交互的经验中学习, 且不需要模型. 但时间差分学习不是等到一个训练周期结束之后再进行更新, 而是在每个时间步上利用时间差分 (Temporal difference) 的方式进行更新, 因此可以达到更快的收敛效果. 状态值函数的更新方式为:

$$V(s_t)\leftarrow\alpha V(s_t)+(1-\alpha)(r_{t+1}+\gamma V(s_{t+1})) \tag{8}$$

其中, α 为更新速率, 满足 $0<\alpha<1$. 时间差分学习采用上一次的估计值来更新当前状态值函数, 这种方法也称作自举法 (Bootstrapping). 在大多数情况中, 自举法的学习速度要快于非自举方法. 时间差分学习方法旨在获得值函数, 当面临控制决策问题时, 状态–动作值函数对于动作的选择更具有指导意义. 基于状态–动作值函数使用时间差分学习的算法主要分为 SARSA 学习和 Q 学习. SARSA 学习算法是一种同策略 (On-policy) 的学习算法, 即评估策略和实际执行策略是同一个策略, 采用如下方法进行状态–动作值函数的估计:

$$\begin{aligned}Q(s_t,a_t)\leftarrow&\alpha Q(s_t,a_t)+(1-\alpha)(r_{t+1}+\\&\gamma Q(s_{t+1},a_{t+1}))\end{aligned} \tag{9}$$

其中, a_{t+1} 是下一步实际执行的动作. Q 学习也称为行为依赖启发式动态规划 (Action-dependent heuristic dynamic programming, ADHDP)[19]. 与 SARSA 算法对 Q 函数的更新方式不同, Q 学习使用贝尔曼最优性原理使当前值函数直接趋近于最优策略的值函数, 更新方法如下:

$$\begin{aligned}Q(s_t,a_t)\leftarrow&\alpha Q(s_t,a_t)+(1-\alpha)(r_{t+1}+\\&\gamma\max_{a'}Q(s_{t+1},a'))\end{aligned} \tag{10}$$

这里 a' 是在状态 s_{t+1} 下使 Q 函数最大的动作. 由于 Q 学习中值函数评估的策略与实际执行策略不同, 因此是一种异策略 (Off-policy) 学习算法. 通过适当设计 Q 函数和更新方法, Q 学习可以在未知模型信息条件下获得线性系统的最优策略.

1.3　策略梯度法

蒙特卡洛法和时间差分法都是基于值函数的方法, 实际使用中常采用表格来存储状态或状态–动作值函数, 因而对于具有较大动作空间的复杂问题是比较低效的. 策略梯度方法不依赖值函数, 直接将策略 π 参数化为 $\pi(s|\theta)$, 然后计算出关于策略性能指标的梯度. 根据梯度方向, 调整策略参数, 最终得到最优策略[20]. 参数化策略可以分为随机性策略 $\pi(s|\theta)=P[a|s,\theta]$ 和确定性策略 $a=\mu(s|\theta)$, 并设置策略目标函数 $J(\theta)$ 对参数化策略进行评价. 对于随机性策略, 当前状态 s 的动作 a 服从参数为 θ 的某个概率分布. 而对于确定性策略, 每个状态对应的动作是确定的. 根据策略梯度定理, 随机性策略梯度表示为[21]:

$$\nabla_{\theta}J(\theta)=\mathrm{E}_{s,a\sim\pi}[\nabla_{\theta}\ln\pi(s|\theta)Q_{\pi}(s,a)] \tag{11}$$

确定性策略梯度表示为[22]:

$$\nabla_{\theta}J(\theta)=\mathrm{E}_{s,a\sim\mu}[\nabla_{\theta}\mu(s|\theta)\nabla_{a}Q_{\mu}(s,a)|_{a=\mu(s|\theta)}] \tag{12}$$

梯度计算时, 需要真实的状态–动作值函数 $Q_{\pi}(s,a)$ 或 $Q_{\mu}(s,a)$, 然而实际上该函数是未知的. 一种方法是使用一定步数的回报值作为状态–动作值函数的估计. 另一种方法是使用执行器–评价器结构[23−24], 使用评价器 (Critic) 以拟合状态–动作值函数, 使用执行器 (Actor) 表示策略. 评价器表示为参数 w 的函数 $Q(s,a|w)$, 并使用时间差分方法更新. 时间差分误差 δ_t 表示为:

$$\delta_t=r_t+\gamma Q(s_{t+1},a_{t+1}|w)-Q(s_t,a_t|w) \tag{13}$$

评价器参数 w 的更新公式为:

$$w\leftarrow w+\alpha\delta_t\nabla_{w}Q(s_t,a_t|w) \tag{14}$$

将学习得到的评价器函数 $Q(s,a|w)$ 代替真实的值函数 $Q_{\pi}(s,a)$ 或 $Q_{\mu}(s,a)$, 代入策略梯度公式完成对策略的更新. 另外, 执行器–评价器结构也可以采用同策略或者异策略两种形式进行实施.

2　深度强化学习

深度强化学习融合了深度学习的感知能力和强化学习的决策能力, 用于解决高维决策问题[25−27]. 图 2 是深度强化学习的基本原理.

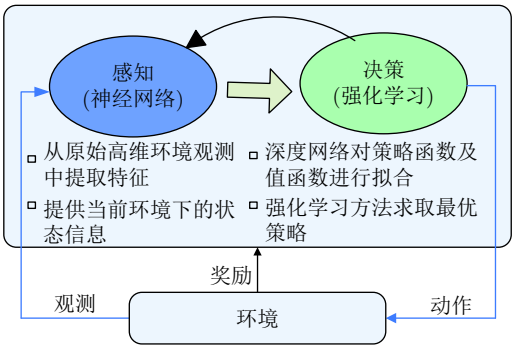

图 2 深度强化学习原理图

Fig. 2 Schematic diagram of deep reinforcement learning

2.1 深度 Q 网络

2015 年, Mnih 等将卷积神经网络与 Q 学习技术相结合, 提出了深度 Q 网络, 完成了由屏幕原始像素到控制输出的端到端感知与决策过程, 并且在 Atari 视频游戏中达到甚至超过了人类的水平. 具体地, 深度 Q 网络在每个时间步对当前状态所有动作的状态–动作值进行估计, 当学习完成后, 只需要在每一步选择最大状态–动作值对应的动作为最优策略[28].

深度 Q 网络采用深度神经网络拟合状态–动作值函数, 其网络参数表示为 w, 通过训练逼近最优策略对应的状态–动作值函数. 从数学角度, 深度 Q 网络采用梯度下降的方法最小化如下代价函数 $L(w)$:

$$L(w) = \mathrm{E}[(r + \gamma \max_{a'} Q(s', a'|w) - Q(s, a|w))^2] \quad (15)$$

其中, a 和 s 是当前时刻动作和状态, a' 和 s' 是下一时刻动作和状态, r 是奖励信号. 当采用神经网络进行值函数拟合时, 样本的相关性会带来训练过程的不稳定[29]. 为了降低样本的相关性, Mnih 等引入了参数为 w' 的目标网络, 目标网络的参数每 N 步与 Q 网络同步一次. 另外, 将与环境交互的样本 (s, a, r, s') 都保存在经验回放池 $\mathcal{D}$ 中, 在每次更新时, 从经验回放池中随机抽取多个交互样本进行批量式训练. 由此, 代价函数 (15) 表示的训练过程可以进一步写为:

$$\begin{cases} \nabla_w L(w) = \mathrm{E}_{s,a\sim\mathcal{D}}[(y - Q(s, a|w))\nabla_w Q(s, a|w)] \\ y = r + \gamma \max\limits_{a'} Q(s', a'|w') \\ w' \leftarrow w \quad \text{for every } N \text{ steps} \end{cases} \quad (16)$$

深度 Q 网络已经较好地解决强化学习中的维数灾问题, 在后续的研究中, 研究人员基于深度 Q 网络提出了各种改进方案.

Hasselt 等提出双深度 Q 网络算法 (Double deep Q-network, DDQN), 是对深度 Q 网络进行改进的一种重要算法[30−31]. 该算法将动作选择和动作评价使用两个 Q 网络分开执行, 以避免对状态–动作值的过高估计. 具体地, 使用深度 Q 网络选择状态–动作值最大的动作, 同时使用目标网络评价该动作, 其代价函数 $L_D(w)$ 可以写为:

$$L_D(w) = \mathrm{E}[(r + \gamma Q(s', \arg\max_{a'} Q(s', a'|w)|w') - Q(s, a|w))^2] \quad (17)$$

经验回放在深度 Q 网络中可以有效打破数据之间的关联, 具有很重要的作用, 但那些稀有的重要样本却常常没有得到重视. 完全随机选择样本并不是一个最优的选择, 一些重要的、与最终目标相关的样本应当更经常地被采样并用于训练, 而常见的样本则无需经常回放. 基于此发现, Schaul 等提出了优先经验回放算法[32], 使得第 k 个经验样本被采样的概率依赖于其优先度函数 p_k:

$$p_k = \left|r_k + \gamma Q(s'_k, \arg\max_{a'} Q(s'_k, a'|w)|w') - Q(s_k, a_k|w)\right| \quad (18)$$

其中, a_k 和 s_k 是第 k 个经验样本的动作和状态, r_k 和 s'_k 第 k 个经验样本的奖励信号和下一时刻状态.

深度 Q 网络的策略评估过程常常会遇到大量冗余策略的情况. 例如在一种情形下, 常会出现两种以上的动作选择, 而这些动作并不会导致不同的结果. 由此, Wang 等提出了一种竞争网络结构[33], 包括两个共存的网络. 一个参数为 ϕ 的网络用于估计状态值函数 $V(s|\phi)$, 另一个参数为 φ 的网络用于估计优势状态–动作值函数 $A(s, a|\varphi)$, 这两个网络通过下式进行值函数的拟合:

$$Q(s, a) = V(s|\phi) + (A(s, a|\varphi) - \max_{a'} A(s, a'|\varphi)) \quad (19)$$

深度 Q 网络可以使用部分历史数据作为输入, 以解决对历史数据有一定依赖的任务, 但不能有效处理长期历史数据依赖问题. 文献 [34] 对于这类部分可观马尔科夫决策过程, 将卷积神经网络之后的全连接层改为递归神经网络. 这种对于深度 Q 网络的改进算法称为深度递归 Q 网络 (Deep recurrent Q-network, DRQN). 该方法在一些依赖历史数据的游戏中取得了远超过深度 Q 网络的效果. 进一步, Lample 等在 DRQN 的网络结构上额外加入了游戏特征, 用 DRQN 结构在 Doom 游戏环境中超过了平均人类玩家的水平[35]. 另一种重要改进是在 DRQN 结构的基础上引入了注意力机制, 提出深度注意力递归 Q 网络 (Deep attention recurrent Q-network, DARQN)[36]. 这种结构使策略网络更加关注重要特征, 从而使用较小的网络规模可以完成相同的任务, 有效提高了训练速度.

2.2　深度策略梯度

深度 Q 网络通常只应用于离散动作空间的问题, 在选择每一步动作时, 需要找到使状态–动作值函数最大的动作. 如果在连续动作空间上, 需要在每一步进行迭代优化, 耗费大量的计算时间. 针对这个问题, Lillicrap 等将 DQN 的经验回放机制和目标网络机制与确定性策略梯度算法 (Deterministic policy gradient, DPG) 相结合, 提出了一种使用执行器-评判器结构的深度强化学习算法, 即深度确定性策略梯度 (Deep deterministic policy gradient, DDPG) 算法, 有效弥补了 DQN 只能用于离散动作空间的问题[37].

DDPG 使用了执行器-评判器结构, 执行器网络和评判器网络分别表示为 $\mu(s|\theta)$ 和 $Q(s,a|w)$, θ 和 w 分别为其网络参数. 两个网络分别有其对应的目标网络, 其参数分别为 θ' 和 w'. 在状态空间的探索方面, 由于 DDPG 算法是异策略的方法, 因此可以通过构建一个额外加入噪声项 ρ 的探索策略 $\widehat{\mu}(s_t|\theta)=\mu(s_t|\theta)+\rho$ 来进行探索. 最终, DDPG 的执行器网络和评判器网络的更新公式为:

$$\begin{cases}\delta=r+\gamma Q(s',\mu(s'|\theta')|w')-Q(s,a|w)\\ w\leftarrow w+\alpha_w\delta\nabla_w Q(s,a|w)\\ \theta\leftarrow\theta+\alpha_\theta\nabla_\theta\mu(s|\theta)\nabla_a Q(s,a|w)|_{a=\mu(s|\theta)}\end{cases}\tag{20}$$

DDPG 算法简洁易用, 可以很容易应用到高维的连续状态和动作空间上. 但 DDPG 在应用中却存在着训练低效的问题, 需要大量的训练样本和较长的训练时间才能收敛到稳定的策略.

DQN 和 DDPG 都使用了经验回放机制, 在高维复杂问题中需要使用大量的存储和计算资源. 针对该问题, Mnih 等提出了另一种思路来代替经验回放机制, 即创建多个智能体, 在不同线程上的相同环境中进行并行学习. 每个智能体使用不同的探索策略并进行参数的更新, 从而减少了经验数据在时间上的关联, 因此不需要通过经验回放机制也能够实现稳定的学习. 该方法结合 SARSA 学习、Q 学习以及执行器–评价器结构可以有多种实施方法, 其中使用执行器–评价器结构的异步执行的方式具有最好的效果, 被称为 A3C (Asynchronous advantage actor-critic) 算法[38]. 该算法包括一个全局执行器–评价器网络和多个对应于每个线程的执行器–评价器网络. 两种网络结构相同, 均为双输出的神经网络结构, 网络的一个输出表示策略, 另一个输出表示状态值函数. 全局策略和值函数分别表示为 $\pi(s|\theta)$ 和 $V(s|\phi)$, 每个线程的策略和值函数分别表示为 $\pi(s|\theta')$ 和 $V(s|\phi')$, 其中 θ,θ',ϕ 和 ϕ' 为网络的参数. 每执行 n 步或者达到某个终止状态时进行一次网络更新, 首先计算每个线程的值函数梯度和策略梯度, 然后将它们分别相加, 对全局的网络参数进行更新, 随后再复制给每个线程的网络. 另外, 在执行网络的参数梯度中加入了策略的熵正则化项 $\nabla_{\theta'}H(\pi(s_t|\theta'))$, 其中 $H(\pi(s_t|\theta'))$ 是熵[39], 可以增强算法在状态空间中的搜索效果, 避免策略过早收敛于某个确定的次优策略.

在上述深度学习算法的应用中, 尽管采用了很多方法来保证其训练的稳定性, 但往往无法保证其策略的性能总是向更好的方向更新. 对于该问题, Schulman 等提出了一种保证单调改进的 TRPO (Trust region policy optimization) 算法[40]. TRPO 算法通过引入由散度定义的置信区域约束, 来选取合适的更新补偿, 保证策略总向着更好的方向更新, 并在机器人游泳、跳跃、行走等任务的仿真环境中表现出良好的性能.

2.3　学习系统闭环控制

基于强化学习和深度强化学习的系统, 可以考虑是未来智慧系统的雏形. 人工智能与被控系统结合, 构建具有类脑智能的智慧系统, 是系统控制的高级目标. 古人云 “学而时习之” 、“温故而知新”, 无论是经典控制理论中最核心的 “反馈” 概念, 还是在上述学习算法中体现出来的 “执行–评价” 过程、“经验回放” 思想等, 都关注了对既往累积知识的使用和再学习. 图 3 概括了学习系统闭环控制框架, 统称为 “习件” (Relearnware), 包含与环境交互和感知, 基于输入和感知知识的学习, 自身累积知识的温习和反馈过程, 以及智能系统的更新进化.

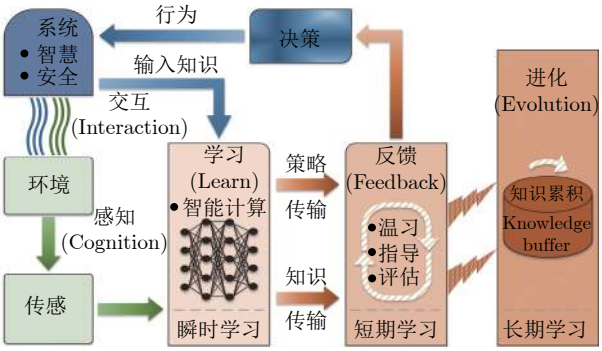

图 3　学习系统闭环控制框架

Fig. 3　Relearnware: closed-loop control framework of learning systems

“习件” 的思想体现了系统交互 (Interaction)、感知 (Cognition)、学习 (Learn)、反馈 (Feedback)、进化 (Evolution) 能力. 具体来说, 交互能力增强了系统在动态、开放、复杂环境中的适应性. 感知能力

增强了系统对有效信息的获取, 能够有效应对耦合、相关、不完整、非结构化信息. 学习能力是基于当前的交互、感知等输入信息, 通过智能计算获取决策的能力. 反馈能力有效对系统累积知识进行温习和回顾, 对学习决策进行指导和评估; 反馈机制的增加, 有助于提高系统学习的效率和优化学习决策, 同时提高决策的安全性, 是形成高效安全可信智能系统的重要步骤. 进化是学习系统基于自身累积的知识和核心学习算法, 形成有效的智能进化范式, 实现具有类脑智能的智慧系统.

3 多智能体深度强化学习

多智能体系统中每个智能体的策略不只取决于自身的策略和环境的反馈, 同时还受到其他智能体行为和合作关系的影响[41]. 例如, 若智能体对环境有完全的观测能力, 则每个智能体可以在时间步 t 获得全局状态, 并且通过自身的策略选择动作. 当智能体由于实际条件限制只有局部观测能力时, 每个智能体只能利用其局部观测值通过自身策略选择动作. 若智能体之间是合作关系时, 所有智能体为着一个相同的目标而努力, 在时间步 t 每个智能体获得的奖励是相同的. 当智能体之间相互竞争, 或者分组竞争时, 每个智能体则会得到不同的奖励值. 多种不同的具体情况使得多智能体强化学习更加复杂. 本节首先介绍几种常见的算法结构, 然后探讨环境非静态性、部分可观性、通信设计、算法稳定性与收敛性等几类重点问题.

3.1 学习算法结构

独立式学习和集中式学习是将单智能体强化学习方法直接推广到多智能体系统中的两种思路. 独立式学习方法对每个智能体分别使用强化学习算法, 而将其他智能体看作环境的一部分. 独立式 Q 学习算法 (Independent Q-learning, IQL) 是一个典型的例子, 在学习过程中, 每个智能体获得其局部观测, 并且向着最大化整体奖励值的方向调整每个智能体的策略[42], 即每个智能体独立的执行 Q 学习算法. 由于每个智能体在学习的过程中, 其他智能体的策略同时发生变化, 打破了环境静态性的假设, 该方法在离散状态–动作空间下的小规模问题上具有一定的效果[43], 对复杂问题无法获得理想的效果. 集中式学习方法将所有智能体的状态和动作集中在一起, 构成一个扩张的状态和动作空间, 并直接使用单智能体的强化学习算法[44−45]. 但随着智能体数量的增加, 会导致状态和动作空间过大, 以至于无法进行有效的探索和训练. 因此, 近期对于多智能体强化学习的研究, 一般都寻求一种分布式的方法, 以避免过大的状态和动作空间[46].

“集中式训练–分布式执行” 是当前常用的一种多智能体强化学习算法结构. 在训练时, 所有的智能体采用集中式结构进行训练, 每个智能体可以通过无限制开放的信道获得其他智能体的信息. 在训练结束之后则执行分布式策略, 每个智能体只能通过自身的观测和有限信道传来的其他智能体的信息进行动作的选择[47]. 由于多智能体强化学习常在模拟环境中进行训练, 因此智能体之间的通信不受实际硬件条件的限制, 而且易于获得额外的环境状态, 便于这种集中式训练–分布式执行的结构的实际应用. 因此, 这种结构也被认为是多智能体强化学习领域的典型学习结构之一.

Lowe 等提出了多智能体深度确定性策略梯度算法 (MADDPG), 将经典的 DDPG 算法扩展到多智能体领域, 使用了集中式训练–分布式执行的结构, 每个智能体均有执行器网络和评价器网络[48]. 在训练中, 评判器网络可以获取全局信息, 并对执行器网络的更新提供指导. 在测试中, 执行器网络根据其局部信息进行动作的选择. 此外, 该方法还引入了额外的网络用于预测其他所有智能体的策略, 并在多种合作和竞争的任务中取得了较好的效果. 基于 MADDPG 算法, 还有多种类似的拓展和补充性的工作[49−50]. Foerster 等提出了一种反拟多智能体策略梯度 (Counterfactual multi-agent policy gradients, COMA) 算法, 将一个智能体的奖励表示为当前状态下的整体奖励与该智能体替换动作之后的整体奖励之差[51]. 与 MADDPG 方法不同, COMA方法使用了一个全局的评判函数对当前的全部动作和状态进行评价, 提高了训练中信息共享效率和智能体之间的协作能力. 由于全局评判函数的使用, 该方法同样使用了集中式训练–分布式执行的结构. COMA 的不足之处是只能用于离散的动作空间, 而不能像 MADDPG 一样可用于连续动作空间.

尽管集中式训练–分布式执行的结构具有诸多优势, 但是随着智能体数量的增加, 集中式训练中评价器网络规模会快速增长, 因而无法处理大规模多智能体的学习问题. 针对这类问题, 带有信息共享的完全分布式学习结构更加有效[52−53]. 在这种结构中, 多个智能体通过稀疏的网络拓扑进行信息共享, 共享的内容主要有原始观测, 表示策略的参数或者梯度, 表示值函数的参数或者梯度, 以及以上几种内容的组合. 此外, 信息也可以是智能体通过学习得到的通信策略产生的内容, 对于这种通信方式, 将在后面进行详细的综述.

3.2 环境非静态性

在单智能体强化学习中, 仅需要根据自身动作和环境交互即可完成学习任务. 而当环境中存在多个智能体时, 每个智能体不仅观测其自身的动作对环境造成的影响, 同时也会观测其他智能体的动作对环境的影响[54]. 更重要的是, 每个智能体在环境中同时也进行学习, 改变自身的策略, 进而导致了从每个智能体的角度出发, 环境都具有非静态性.

在这种情况下, 通过学习改变其中一个智能体的策略会影响其他智能体最优策略的选取, 同时对于智能体值函数估计也会不准确. 换句话说, 当前情况下的最优策略, 随着其他智能体学习和策略的更新, 在未来的情况下将不再是最优策略[55], Q 学习等应用于传统单智能体的强化学习方法, 在多智能体环境中将不能保证算法的收敛性. 由于上述原因, 无论独立式 Q 学习算法或者近年来提出的经验回放深度 Q 网络算法[56], 均不适用于非静态环境的问题.

为解决多智能体强化学习中非静态环境问题, 基于 DQN 算法提出了多种改进方案. Abdallah 等基于重复更新 Q 学习 (Repeat update Q-learning, RUQL) 算法, 提出深度重复更新 Q 网络 (Deep repeated update Q-network, DRUQN)[57−58], 通过与选择动作概率成反比的方法, 来更新动作值避免策略的偏差. 基于松耦合 Q 学习方法[59], 深度松耦合 Q 网络 (Deep loosely coupled Q-network, DLCQN) 引入独立程度的概念, 通过观测信息和负值奖励信息为每个智能体调整独立程度, 智能体可以在不同情况中通过学习来决定独立行动还是与其他智能体进行合作. Diallo 等将 DQN 扩展为多智能体并行 DQN, 并展示该方法可以在非静态环境中收敛[60]. Foerster 等提出在多智能体环境下使用经验回放机制的 DQN 算法, 主要是给经验加入额外信息来辅助多智能体的训练过程[61], 包括两种具体解决方法: 1) 使用重要性采样方法来剔除过时数据; 2) 通过在经验中加入更多信息来确定经验池中回放样本的“年龄”. 类似的方法还有 Palmer 等提出的宽松 DQN (Lenient DQN, LDQN) 算法, 用以解决多智能体同时学习而导致的策略不稳定问题[62], 并在多智能体协同运输任务中与滞回 DQN (Hysteretic DQN, HDQN) 算法进行了对比, 表明 LDQN 算法在随机奖励环境中能够收敛到比 HDQN 算法更好的控制策略[63]. Zheng 等将上述宽松条件机制与经验定期回放机制结合, 提出了加权 DDQN (Weighted DDQN) 算法, 以应对多智能体环境中的非静态环境问题, 对随机奖励的两个智能体, 通过仿真验证了 WDDQN 相对于 DDQN 具有更好的性能[64].

3.3 部分可观性

在多数任务中, 每个智能体并不能得到全部环境信息, 而只能对部分环境信息进行观测, 这类问题可以使用部分可观马尔科夫决策过程 (Partially observable Markov decision process, POMDP) 进行建模和研究[65]. 针对部分可观测问题和 POMDP 模型, 已经有一些解决方案. Hausknecht 等提出了深度递归 Q 网络 (Deep recurrent Q-network, DRQN) 算法[34], 使用 DRQN 方法的单智能体能够在部分可观的环境中以鲁棒的方式学习并改进策略. 与传统的 DQN 算法不同, DRQN 通过递归神经网络近似 $Q(o, a)$, 即观测值 o 和动作值 a 的状态–动作值函数, 同时 DRQN 将网络的隐层状态视为环境的内部状态, 将隐层状态也包含在状态–动作值函数中, 然后再使用与 DQN 类似的方法进行值函数的更新.

Foerster 等将 DRQN 算法扩展为深度分布式递归 Q 网络算法 (Deep distributed recurrent Q-network, DDRQN), 用以处理多智能体部分可观测和 POMDP 问题[66]. DDRQN 算法主要有三个特点: 1) 将每个智能体上一时间步的动作作为本时间步的输入状态的一部分; 2) 在学习过程中所有智能体共享同一个 Q 网络; 3) 相比于 DQN 算法, 不使用经验回放机制. DDRQN 通过共享 Q 网络的方法, 可以大大减少网络参数的数量, 提高学习速度. 但该方法的一个重要局限在于假设所有的智能体动作集是相同的, 因此 DDRQN 方法不能应用于异构多智能体优化控制问题中.

Hong 等提出深度策略推理递归 Q 网络 (Deep policy inference recurrent Q-network, DPIRQN), 也使用了递归神经网络以应对部分可观性的问题[67]. DPIRQN 通过引入辅助任务和额外学习目标, 对其他智能体的策略进行学习. 在训练中, 自适应调整更加重视对其他智能体策略的学习, 还是更加重视对自身策略的优化. 这种算法使得每个智能体的值函数一定程度上依赖其他智能体的策略, 减小了环境的非静态性对学习带来的不利影响, 可同时应用于多智能体合作和竞争两种任务中.

3.4 基于学习的通信

在有些分布式的学习结构中, 智能体之间通过通信网络共享观测数据、策略参数、策略梯度等信息, 最终完成智能体之间的合作. 与这种指明通信内容的方法不同, 另一种用于多智能体强化学习的通信方式是基于学习的通信方式. 智能体通过学习算法, 逐渐学习一种通信策略. 智能体的通信策略可以根据当前状态决定什么时候发送信息, 发送什

么种类的信息, 发送信息的内容以及接收信息的目标智能体.

文献 [68] 最早给出了这种基于学习的通信方式, 多智能体通过 Q 学习确定给其他智能体发送信息的内容并完成离散状态和动作空间下的合作追捕问题. 近年来, 基于学习的通信结合值函数拟合方法的研究在多智能体强化学习领域得到了很大的发展. Foerster 等基于集中式训练–分布式执行结构, 提出了智能体间强化学习 (Reinforced inter-agent learning, RIAL) 方法和智能体间可微学习 (Differentiable inter-agent learning, DIAL) 方法, 引入了智能体基于学习的通信策略[69]. 智能体选择控制动作来改变自己的状态, 同时也选择通信动作来影响其他智能体的动作. 在 RIAL 方法中, 通过在深度 Q 网络中引入循环神经网络, 解决部分可观察性问题. 在训练中, 所有的智能体共享同一个深度 Q 网络来得到控制动作和通信动作的值. 在测试中, 每个智能体将训练得到的深度 Q 网络复制到本地, 并独立进行控制动作和通信动作的选择, 从而完成分布式的执行. DIAL 方法在深度 Q 网络中建立一条可微信道, 不再使用离散的通信动作, 可以在训练中将一个智能体的梯度信息推送到与其连接的智能体中, 大大增强了学习中的反馈作用, 提高了训练的效果. Sukhbaatar 等使用了类似的通信方法, 提出了一种多智能体强化学习通信网络, 称 CommNet 模型[70], 同样建立了可微信道, 并使用反向传播算法进行训练. 不同的是, 所有智能体共享同一个信道, 每个智能体接收到的是特定范围内所有智能体发送的通信消息的数值之和. 该方法在十字路口模拟调度和网格地图模拟战斗等任务中进行了测试, 取得了很好的效果.

3.5 算法稳定性与收敛性

在多智能体深度强化学习领域, 使用深度网络表示值函数和策略, 给多智能体系统的控制和决策带来了更为通用的方法, 使其能够应用于更多复杂的环境. 然而, 随着智能体数量的增加, 多智能体系统的联合状态–动作空间呈指数增长, 深度网络的复杂性也快速增加, 极大增加了深度强化学习算法的探索难度, 甚至使算法最终无法收敛. 总的来说, 多智能体深度强化学习方法的稳定性和收敛性问题, 既受到深度学习方法本身的限制, 也受到多智能体系统和其所在环境的限制, 至今仍是一个开放性的难题.

当强化学习算法用于多智能体一致性问题时, 常常会遇到算法的稳定性和收敛性问题. 在这种问题中, 每个智能体只能获得本地的观测, 同时通过通信网络获得相邻智能体的信息, 当值函数等的拟合采用线性函数或一般神经网络时, 可以得到一些理论上的稳定性和收敛性结果. 文献 [71] 使用执行器–评价器算法结构, 使得所有智能体的一致性误差最小, 给出了一致性误差的理论上界, 并且讨论了在已知系统动态的情况下得到最优控制器的可行性. 文献 [72] 针对多智能体强化学习问题提出了一种分布式执行器–评价器算法, 该方法假设所有的智能体都在本地保持对全局最优策略的估计, 并且独立更新本地的值函数. 通过引入额外的一致性处理方法, 使所有的智能体最终渐近收敛于全局最优策略, 同时进行了算法收敛性分析.

4 多智能体深度强化学习的应用

多智能体深度强化学习方法在多个领域有广泛的应用前景, 如无人驾驶、智能仓储、生产调度、资源访问控制等领域. 下面讨论几个具有广阔应用前景, 尚需进一步发展的应用领域.

4.1 社区能源管理和共享问题

多智能体强化学习方法近年来被引入社区能源管理和共享问题中[73−74]. 相比于随机能源共享方法, 采用多智能体深度强化学习方法, 在社区能源平衡调度方面具有明显的优势. Prasad 等在包含多个绿色建筑物的零能耗社区中, 将每一栋绿色建筑物抽象成一个深度强化学习的智能体, 设计奖励函数与整个社区中的能源净消耗量有关, 通过学习执行合理的动作与其他绿色建筑物共享能源, 使所有建筑物在一年内的总耗电量小于其可再生能源的发电量[75]. 但该方法仅应用于最多十个建筑物的社区能源共享调度上, 没有测试更大规模的社区, 也没有考虑电价变动带来的影响.

4.2 任务分配与调度

任务分配和任务调度问题, 通常需要通过多次迭代规划来获得最优解, 而复杂任务的分配和规划问题, 采用经典的规划方法往往难以获得可行的方案, 如复杂环境导航等问题[76−77]. Lin 等基于执行器-评判器结构和深度 Q 学习算法, 提出使用多智能体强化学习方法研究大规模车队高效调度问题[78]. 论文将车辆建模为智能体, 使用网格对区域进行描述, 通过地理信息嵌入的方式建立智能体之间明确的合作关系, 仿真表明该方法用于车队调度可以减少交通拥塞, 提高运输效率. Noureddine 等使用合作式多智能体深度强化学习方法研究任务分配问题[79], 使多个智能体能够在一个疏松耦合的分布式环境中请求其他智能体的帮助, 通过多个智能体之间的交互最终达到高效的任务分配.

4.3 机器人集群控制

机器人集群控制, 是目前多智能体深度强化学习方法的应用研究热点. Hüttenrauch 等将机器人集群系统建模为分布式 POMDP, 并使用执行器–评判器结构对机器人集群系统进行协同控制[80]. 该方法通过视频信息描述整个机器人集群的状态, 并作为一个全局信息用于估计系统的值函数. 每个机器人在环境中的观测范围有限, 通过合作方式, 可以完成协同搜救和装配等复杂的任务. Kurek 等基于 DQN 算法, 对每个智能体使用不同的 Q 网络和独立的经验回放池, 研究异构机器人合作问题[81]. 尽管该方法能够在游戏环境中有效提高机器人合作的得分, 然而其训练速度远远落后于同构机器人的情况. 期望在不久的将来, 可以看到多智能体强化学习方法在机器人集群控制中相关的实际应用.

4.4 社会学与博弈

社会学中的一些问题, 如具有代表性的囚徒困境的例子等, 反映个体最佳选择和团体最佳选择的博弈. 近年来, 多智能体强化学习的方法也被用于一些社会学问题的研究中. Leibo 等提出一种连续社会困境 (Sequential social dilemma, SSD) 概念, 并建立了 SSD 模型. 该问题无法使用一般的规划和进化的方法对均衡点进行求解, 使用独立 DQN 的学习方法可以模拟博弈中智能体的决策方式, 从而寻找到 SSD 的均衡点[15]. Perolat 等对于公共池塘资源 (Common-pool resource, CPR) 占用问题[82], 使用多个独立学习的 DQN 智能体在 CPR 环境进行学习, 通过不断试错和调整每个智能体的奖励方式, 最终得到 CPR 占用问题的最优解.

5 未来研究方向

5.1 复杂任务的 MADRL

多智能体深度强化学习方法, 具有强大的理解、决策和协调能力, 被期望是解决复杂任务问题的有效方法. 然而, 这些方法尚未在多智能体环境中进行全面的研究. 比如, 逆强化学习 (Inverse reinforcement learning) 作为模仿学习的方法之一, 在单智能体深度强化学习中是有效的[83]. 模仿学习和逆强化学习方法可以减少学习时间并提高策略的有效性, 有巨大的应用潜力[84–85]. 但逆强化学习假设关于未知奖励函数的策略是最优的, 并且需要从演示中推断出奖励函数. 将逆强化学习方法延伸到 MADRL 领域需要表示和建模能够共同演示任务的多位专家以及专家的交流和推理. 面对具体复杂任务, 深入融合目标任务、学习方法和通信规则, 设计出符合特定任务要求的高效智能算法, 是未来多智能体深度强化学习方法重要的发展方向之一.

5.2 基于模型的 MADRL

无模型深度强化学习方法能够解决单智能体和多智能体中的许多问题, 但是, 此类方法通常应用于确定的、静态的任务, 且需要大量样本和较长的学习时间才能获得良好的性能. 对于不确定和动态任务, 基于模型的多智能体深度强化学习方法已经在样本效率、可转移性和通用性等方面展现出有效性. 尽管最近在单智能体中研究了一些基于模型的深度强化学习方法[86–90], 但这些方法尚未在多智能体中得到广泛研究. 所以, 可以在基于模型的多智能体强化学习方向做更多的研究探索. 此外, 结合基于模型的方法和无模型方法, 设计多智能体深度强化学习方法, 也是尚未被充分研究的领域.

5.3 通信受限的 MADRL

大型系统中异构智能体协调与协作一直是多智能体强化学习领域的主要挑战. 在具有许多异构智能体的环境中, 由于个体具有共同的行为, 例如动作、领域知识和目标, 因此可以通过集中训练和分散执行, 来实现异构个体的控制[91–92]. 在异构个体之间通信困难, 或者同构个体之间通信受限的情况下, 如何设计深度强化学习算法中的目标函数、奖励策略、学习和通讯机制等[93], 实现通信受限下的多智能体高效协调与协作, 提供最佳决策方案并最大程度地完成任务, 是非常值得研究的问题.

5.4 人机交互的 MADRL

深度强化学习一定程度赋予了机器自主理解、学习和决策的能力, 但是, 在复杂和对抗环境中, 需要将人的智能与机器智能结合在一起[94]. 传统的“人在回路”设置中, 智能体会在一段时间内自主执行其分配的任务, 然后停止并等待人工命令, 此后以这种限速方式循环操作. 在循环中, 智能体可以自动执行任务, 直到任务完成为止, 而扮演监督角色的人员保留干预执行操作的能力[95]. 当循环快速进行, 外界环境发生突变时引入人工干预, 机器可能无法及时作出反应. 面对这类问题, 如何基于多智能体深度强化学习方法, 适时引入人的判断和经验, 整合人和机器的智能, 提高人与机器交互的能力, 也是未来值得研究的方向.

6 结论

本文阐述了强化学习和深度强化学习的基本原理与研究现状, 总结提出了包含交互、感知、学习、

反馈和进化的学习系统闭环控制框架, 强调了反馈在学习控制中的作用. 分析了多智能体深度强化学习的算法结构和存在的主要困难, 考察了在零能耗社区的能源共享、任务分配调度、机器人集群控制等相关应用领域的研究进展. 多智能体强化学习领域的理论研究日渐深入, 需要付出更多的时间和努力来探索多智能体强化学习理论的应用载体和相关技术, 并与具体任务相结合, 切实推进人工智能理论和技术的发展.

References

1 Rubenstein M, Cornejo A, Nagpal R. Programmable self-assembly in a thousand-robot swarm. *Science*, 2014, **345**(6198): 795−799

2 Wang Y D, He H B, Sun C Y. Learning to navigate through complex dynamic environment with modular deep reinforcement learning. *IEEE Transactions on Games*, 2018, **10**(4): 400−412

3 Zheng Nan-Ning. On challenges in artificial intelligence. *Acta Automatica Sinica*, 2016, **42**(5): 641−642
(郑南宁. 人工智能面临的挑战. 自动化学报, 2016, **42**(5): 641−642)

4 Nguyen T T, Nguyen N D, Nahavandi S. Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications. *IEEE Transactions on Cybernetics*, 2020

5 Zhao Dong-Bin, Shao Kun, Zhu Yuan-Heng, Li Dong, Chen Ya-Ran, Wang Hai-Tao, et al. Review of deep reinforcement learning and discussions on the development of computer Go. *Control Theory & Applications*, 2016, **33**(6): 701−717
(赵冬斌, 邵坤, 朱圆恒, 李栋, 陈亚冉, 王海涛, 等. 深度强化学习综述: 兼论计算机围棋的发展. 控制理论与应用, 2016, **33**(6): 701−717)

6 Zhou Zhi-Hua. AlphaGo special session: an introduction. *Acta Automatica Sinica*, 2016, **42**(5): 670
(周志华. AlphaGo 专题介绍. 自动化学报, 2016, **42**(5): 670)

7 Silver D, Huang A, Maddison C J, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016, **529**(7587): 484−489

8 Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of go without human knowledge. *Nature*, 2017, **550**(7676): 354−359

9 Berner C, Brockman G, Chan B, Cheung V, Dębiak P, Denniso C, et al. Dota 2 with large scale deep reinforcement learning. arXiv: 1912.06680, 2019.

10 Hung S M, Givigi S N. A Q-learning approach to flocking with UAVs in a stochastic environment. *IEEE Transactions on Cybernetics*, 2017, **47**(1): 186−197

11 Schwab D, Zhu Y F, Veloso M. Zero shot transfer learning for robot soccer. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018). Stockholm, Sweden: ACM, 2018. 2070−2072

12 Wang Yun-Peng, Guo Ge. Signal priority control for trams using deep reinforcement learning. *Acta Automatica Sinica*, 2019, **45**(12): 2366−2377
(王云鹏, 郭戈. 基于深度强化学习的有轨电车信号优先控制. 自动化学报, 2019, **45**(12): 2366−2377)

13 Rahman M S, Mahmud M A, Pota H R, Hossain M J, Orchi T F. Distributed multi-agent-based protection scheme for transient stability enhancement in power systems. *International Journal of Emerging Electric Power Systems*, 2015, **16**(2): 117−129

14 He J, Peng J, Jiang F, Qin G R, Liu W R. A distributed Q learning spectrum decision scheme for cognitive radio sensor network. *International Journal of Distributed Sensor Networks*, 2015, **2015**: 7

15 Leibo J Z, Zambaldi V, Lanctot M, Marecki J, Graepel T. Multi-agent reinforcement learning in sequential social dilemmas. In: Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems. Sao Paulo, Brazil: ACM, 2017. 464−473

16 Wu Guo-Zheng. Analysis of the status and trend of the development of China's automation discipline from F03 funding of NSFC. *Acta Automatica Sinica*, 2019, **45**(9): 1611−1619
(吴国政. 从 F03 项目资助情况分析我国自动化学科的发展现状与趋势. 自动化学报, 2019, **45**(9): 1611−1619)

17 Hernandez-Leal P, Kartal B, Taylor M E. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 2019, **33**(6): 750−797

18 Mu C X, Ni Z, Sun C Y, He H B. Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **28**(3): 584−598

19 Mu C, Zhao Q, Sun C, Gao Z. A novel Q-learning algorithm for optimal tracking control of linear discrete-time systems with unknown dynamics. *Applied Soft Computing*, 2019, **82**: 1−13

20 Wang Y D, Sun J, He H B, Sun C Y. Deterministic policy gradient with integral compensator for robust quadrotor control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019

21 Sutton R S, McAllester D, Singh S, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: Proceedings of the 12th International Conference on Neural Information Processing Systems. Denver, USA: MIT Press, 1999. 1057−1063

22 Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. Deterministic policy gradient algorithms. In: Proceedings of the 31st International Conference on Machine Learning. Beijing, China: ACM, 2014. 387−395

23 Wei Q L, Wang L X, Liu Y, Polycarpou M M. Optimal elevator group control via deep asynchronous actor-critic learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2020

24 Dong L, Zhong X N, Sun C Y, He H B. Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **28**(7): 1594−1605

25 Arulkumaran K, Deisenroth M P, Brundage M, Bharath A A. Deep reinforcement learning: a brief survey. *IEEE Signal Processing Magazine*, 2017, **34**(6): 26−38

26 Li Y X. Deep reinforcement learning: an overview. arXiv: 1701.07274, 2017.

27 Nguyen N D, Nguyen T, Nahavandi S. System design perspective for human-level agents using deep reinforcement learning: a survey. *IEEE Access*, 2017, **5**: 27091−27102

28 Nguyen T T. A multi-objective deep reinforcement learning framework. arXiv: 1803.02965, 2018.

29 Tsitsiklis J N, van Roy B. Analysis of temporal-difference learning with function approximation. In: Proceedings of the 9th International Conference on Neural Information Processing Systems. Denver, USA: MIT Press, 1996. 1075−1081

30 Van Hasselt H. Double Q-learning. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems. Vancouver, Canada: MIT Press, 2010. 2613−2621

31 Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. arXiv: 1509.06461, 2015.

32 Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. arXiv: 1511.05952, 2015.

33 Wang Z Y, Schaul T, Hessel M, van Hasselt H, Lanctot M, de Freitas N. Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning. New York, USA: ACM, 2016. 1995−2003

34 Hausknecht H, Stone P. Deep recurrent Q-learning for partially observable MDPs. arXiv: 1507.06527, 2017.

35 Lample G, Chaplot D S. Playing FPS games with deep reinforcement learning. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco, USA: AIAA, 2017.

36 Sorokin I, Seleznev A, Pavlov M, Fedorov A, Ignateva A. Deep attention recurrent Q-network. arXiv: 1512.01693, 2015.

37 Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. arXiv: 1509.02971, 2015.

38 Mnih V, Badia A P, Mirza M, Graves A, Harley T, Lillicrap T P, et al. Asynchronous methods for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning. New York, USA: ACM, 2016. 1928−1937

39 Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv: 1801.01290, 2018.

40 Schulman J, Levine S, Abbeel P, Jordan M I, Moritz P. Trust region policy optimization. In: Proceedings of the 32nd International Conference on Machine Learning. Lille, France: ACM, 2015. 1889−1897

41 Jadid O A, Hajinezhad D. A review of cooperative multi-agent deep reinforcement learning. arXiv: 1908.03963, 2019.

42 Tan M. Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the 10th International Conference on Machine Learning. Amherst, USA: ACM, 1993. 330−337

43 Matignon L, Laurent G J, Le Fort-Piat N. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 2012, **27**(1): 1−31

44 Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, Aru J, et al. Multiagent cooperation and competition with deep reinforcement learning. arXiv: 1511.08779, 2015.

45 Usunier N, Synnaeve G, Lin Z M, Chintala S. Episodic exploration for deep deterministic policies: an application to starcraft micromanagement tasks. arXiv: 1609.02993, 2016.

46 Cui L L, Wang X W, Zhang Y. Reinforcement learning-based asymptotic cooperative tracking of a class multi-agent dynamic systems using neural networks. *Neurocomputing*, 2016, **171**: 220−229

47 Kraemer L, Banerjee B. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 2016, **190**: 82−94

48 Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, USA: MIT Press, 2017. 6379−6390

49 Ryu H, Shin H, Park J. Multi-agent actor-critic with generative cooperative policy network. arXiv: 1810.09206, 2018.

50 Chu X X, Ye H J. Parameter sharing deep deterministic policy gradient for cooperative multi-agent reinforcement learning. arXiv: 1710.00336, 2017.

51 Foerster J N, Farquhar G, Afouras T, Nardelli N, Whiteson S. Counterfactual multi-agent policy gradients. arXiv: 1705.08926, 2017.

52 Zhang K Q, Yang Z R, Liu H, Zhang T, Basar T. Fully decentralized multi-agent reinforcement learning with networked agents. In: Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden: ACM, 2018. 5872−5881

53 Jiang J C, Dun C, Huang T J, Lu Z Q. Graph convolutional reinforcement learning. arXiv: 1810.09202, 2018.

54 Wang Q L, Psillakis H E, Sun C Y. Cooperative control of multiple agents with unknown high-frequency gain signs under unbalanced and switching topologies. *IEEE Transactions on Automatic Control*, 2019, **64**(6): 2495−2501

55 Hernandez-Leal P, Kaisers M, Baarslag T, de Cote E M. A survey of learning in multiagent environments: dealing with non-stationarity. arXiv: 1707.09183, 2017.

56 Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, et al. Human-level control through deep reinforcement learning. *Nature*, 2015, **518**(7540): 529−533

57 Abdallah S, Kaisers M. Addressing the policy-bias of Q-learning by repeating updates. In: Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems. Saint Paul, USA: ACM, 2013. 1045−1052

58 Abdallah S, Kaisers M. Addressing environment non-stationarity by repeating Q-learning updates. *The Journal of Machine Learning Research*, 2016, **17**(1): 1582−1612

59 Yu C, Zhang M J, Ren F H, Tan G Z. Emotional multiagent reinforcement learning in spatial social dilemmas. *IEEE Transactions on Neural Networks and Learning Systems*, 2015, **26**(12): 3083−3096

60 Diallo E A O, Sugiyama A, Sugawara T. Learning to coordinate with deep reinforcement learning in doubles pong game. In: Proceedings of the 16th IEEE International Conference on Machine Learning and Applications. Cancun, Mexico: IEEE, 2017. 14−19

61 Foerster J N, Nardelli N, Farquhar G, Afouras T, Torr P H S, Kohli P. Stabilising experience replay for deep multi-agent reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia: ACM, 2017. 1146−1155

62 Palmer G, Tuyls K, Bloembergen D, Savani R. Lenient multi-agent deep reinforcement learning. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems. Stockholm, Sweden: ACM, 2018. 443−451

63 Omidshafiei S, Pazis J, Amato C, How J P, Vian J. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In: Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia: ACM, 2017. 2681−2690

64 Zheng Y, Meng Z P, Hao J Y, Zhang Z Z. Weighted double deep multiagent reinforcement learning in stochastic cooperative environments. In: Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence. Nanjing, China: ACM, 2018. 421−429

65 Mu C X, Zhao Q, Sun C Y. Optimal model-free output synchronization of heterogeneous multi-agent systems under switching topologies. *IEEE Transactions on Industrial Electronics*, 2019

66 Foerster J N, Assael Y M, de Freitas N, Whiteson S. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. arXiv: 1602.02672, 2016.

67 Hong Z W, Su S Y, Shann T Y, Chang Y H, Lee C Y. A deep policy inference Q-network for multi-agent systems. In: Proceedings of the 17th Conference on Autonomous Agents and Multiagent Systems. Stockholm, Sweden: Springer, 2018. 1388−1396

68 Kasai T, Tenmoto H, Kamiya A. Learning of communication codes in multi-agent reinforcement learning problem. In: Proceedings of 2008 IEEE Conference on Soft Computing in Industrial Applications. Muroran, Japan: IEEE, 2008. 1−6

69 Foerster J N, Assael Y M, de Freitas N, Whiteson S. Learning to communicate with deep multi-agent reinforcement learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain: ACM, 2016. 2137−2145

70 Sukhbaatar S, Szlam A, Fergus R. Learning multiagent communication with backpropagation. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain: ACM, 2016. 2252−2260

71 Zhang H G, Jiang H, Luo Y H, Xiao G Y. Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method. *IEEE Transactions on Industrial Electronics*, 2017, **64**(5): 4091−4100

72 Zhang Y, Zavlanos M M. Distributed off-policy actor-critic reinforcement learning with policy consensus. arXiv: 1903.09255, 2019.

73 Wei Q L, Liu D R, Lewis F L, Liu Y, Zhang J. Mixed iterative adaptive dynamic programming for optimal battery energy control in smart residential microgrids. *IEEE Transactions on Industrial Electronics*, 2017, **64**(5): 4110−4120

74 Yang X D, Wang Y D, He H B, Sun C Y, Zhang Y B. Deep reinforcement learning for economic energy scheduling in data center microgrids. In: Proceedings of the 2019 IEEE Power & Energy Society General Meeting. Atlanta, USA: IEEE, 2019. 1−5

75 Prasad A, Dusparic I. Multi-agent deep reinforcement learning for zero energy communities. arXiv: 1810.03679, 2018.

76 Xu Xin. *Reinforcement Learning and Approximate Dynamic Programming*. Beijing: Science Press, 2010
(徐昕. 增强学习与近似动态规划. 北京: 科学出版社, 2010)

77 Wan Z Q, Jiang C, Fahad M, Ni Z, Guo Y, He H B. Robot-assisted pedestrian regulation based on deep reinforcement learning. *IEEE Transactions on Cybernetics*, 2020, **50**(4): 1669−1682

78 Lin K X, Zhao R Y, Xu Z, Zhou J Y. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London, UK: ACM, 2018. 1774−1783

79 Ben Noureddine D, Gharbi A, Ben Ahmed S. Multi-agent deep reinforcement learning for task allocation in dynamic environment. In: Proceedings of the 12th International Conference on Software Technologies. Madrid, Spain: SciTePress, 2017. 17−26

80 Hüttenrauch M, Šošić A, Neumann G. Guided deep reinforcement learning for swarm systems. arXiv: 1709.06011, 2017.

81 Kurek M, Jaśkowski W. Heterogeneous team deep Q-learning in low-dimensional multi-agent environments. In: Proceedings of the 2016 IEEE Conference on Computational Intelligence and Games (CIG). Santorini, Greece: IEEE, 2016. 1−8

82 Perolat J, Leibo J Z, Zambaldi V, Beattie C, Tuyls K, Graepel T. A multi-agent reinforcement learning model of common-pool resource appropriation. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, USA: ACM, 2017. 3643−3652

83 Piot B, Geist M, Pietquin O. Bridging the gap between imitation learning and inverse reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **28**(8): 1814−1826

84 Hadfield-Menell D, Russell S J, Abbeel P, Dragan A. Cooperative inverse reinforcement learning. In: Proceedings of the 30th Conference on Neural Information Processing Systems. Barcelona, Spain: ACM, 2016. 3909−3917

85 Hadfield-Menell D, Milli S, Abbeel P, Russell S, Dragan A D. Inverse reward design. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, USA: ACM, 2017. 6765−6774

86 Levine S, Finn C, Darrell T, Abbeel P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 2016, **17**(1): 1334−1373

87 Nagabandi A, Kahn G, Fearing R S, Levine S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In: Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane, Australia: IEEE, 2018. 7559−7566

88 Gu S X, Lillicrap T P, Sutskever I, Levine S. Continuous deep Q-learning with model-based acceleration. In: Proceedings of the 33rd International Conference on Machine Learning. New York, USA: ACM, 2016. 2829−2838

89 Finn C, Levine S. Deep visual foresight for planning robot motion. In: Proceedings of the 2017 IEEE International Conference on Robotics and Automation. Singapore: IEEE, 2017. 2786−2793

90 Serban I V, Sankar C, Pieper M, Pineau J, Bengio Y. The bottleneck simulator: a model-based deep reinforcement learning approach. arXiv: 1807.04723, 2018.

91 Rashid T, Samvelyan M, de Witt C S, Farquhar G, Foerster J, Whiteson S. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. arXiv: 1803.11485, 2018.

92 Foerster J N, Chen R Y, Al-Shedivat M, Whiteson S, Abbeel P, Mordatch I. Learning with opponent-learning awareness. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems. Stockholm, Sweden: ACM, 2018. 122−130

93 Yuan X, Dong L, Sun C Y. Solver-critic: a reinforcement learning method for discrete-time constrained-input systems. *IEEE Transactions on Cybernetics*, 2020

94 He W, Li Z J, Chen C L P. A survey of human-centered intelligent robots: issues and challenges. *IEEE/CAA Journal of Automatica Sinica*, 2017, **4**(4): 602−609

95 Nahavandi S. Trusted autonomy between humans and robots: toward human-on-the-loop in robotics and autonomous systems. *IEEE Systems, Man, and Cybernetics Magazine*, 2017, **3**(1): 10−17

孙长银 东南大学自动化学院教授. 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. 本文通信作者.
E-mail: cysun@seu.edu.cn
(**SUN Chang-Yin** Professor at the School of Automation, Southeast University. His research interest covers intelligent control and optimization, reinforcement learning, neural networks, and data-driven control. Corresponding author of this paper.)

穆朝絮 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.
E-mail: cxmu@tju.edu.cn
(**MU Chao-Xu** Professor at the School of Electrical and Information Engineering, Tianjin University. Her research interest covers reinforcement learning, adaptive and learning systems, nonlinear control and optimization.)