第 34 卷第 12 期 控 制 理 论 与 应 用 Vol. 34 No. 12
2017 年 12 月 Control Theory & Applications Dec. 2017

DOI: 10.7641/CTA.2017.70808

深度强化学习进展: 从AlphaGo到AlphaGo Zero

唐振韬, 邵 坤, 赵冬斌†, 朱圆恒

(中国科学院 自动化研究所 复杂系统管理与控制国家重点实验室, 北京 100190; 中国科学院大学, 北京 100190)

摘要: 2016年初, AlphaGo战胜李世石成为人工智能的里程碑事件. 其核心技术深度强化学习受到人们的广泛关注和研究, 取得了丰硕的理论和应用成果. 并进一步研发出算法形式更为简洁的AlphaGo Zero, 其采用完全不基于人类经验的自学习算法, 完胜AlphaGo, 再一次刷新人们对深度强化学习的认知. 深度强化学习结合了深度学习和强化学习的优势, 可以在复杂高维的状态动作空间中进行端到端的感知决策. 本文主要介绍了从AlphaGo到AlphaGo Zero的深度强化学习的研究进展. 首先回顾对深度强化学习的成功作出突出贡献的主要算法, 包括深度Q网络算法、A3C算法、策略梯度算法及其他算法的相应扩展. 然后给出AlphaGo Zero的详细介绍和讨论, 分析其对人工智能的巨大推动作用. 并介绍了深度强化学习在游戏、机器人、自然语言处理、智能驾驶、智能医疗等领域的应用进展, 以及相关资源进展. 最后探讨了深度强化学习的发展展望, 以及对其他潜在领域的人工智能发展的启发意义.

关键词: 深度强化学习; AlphaGo Zero; 深度学习; 强化学习; 人工智能

中图分类号: TP273 **文献标识码**: A

Recent progress of deep reinforcement learning: from AlphaGo to AlphaGo Zero

TANG Zhen-tao, SHAO Kun, ZHAO Dong-bin†, ZHU Yuan-heng

(The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China;
University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In the early 2016, the defeat of Lee Sedol by AlphaGo became the milestone of artificial intelligence. Since then, deep reinforcement learning (DRL), which is the core technique of AlphaGo, has received widespread attention, and has gained fruitful results in both theory and applications. In the sequel, AlphaGo Zero, a simplified version of AlphaGo, masters the game of Go by self-play without human knowledge. As a result, AlphaGo Zero completely surpasses AlphaGo, and enriches humans' understanding of DRL. DRL combines the advantages of deep learning and reinforcement learning, so it is able to perform well in high-dimensional state-action space, with an end-to-end structure combining perception and decision together. In this paper, we present a survey on the remarkable process made by DRL from AlphaGo to AlphaGo Zero. We first review the main algorithms that contribute to the great success of DRL, including DQN, A3C, policy-gradient, and other algorithms and their extensions. Then, detailed introduction and discussion on AlphaGo Zero are given and its great promotion on artificial intelligence is also analyze. The progress of applications with DRL in such areas as games, robotics, natural language processing, smart driving, intelligent health care, and related resources are also presented. In the end, we discuss the future development of DRL, and the inspiration on other potential areas related to artificial intelligence.

Key words: deep reinforcement learning; AlphaGo Zero; deep learning; reinforcement learning; artificial intelligence

1 引言(Introduction)

深度强化学习(deep reinforcement learning: DRL)结合了深度神经网络和强化学习的优势, 可以用于解决智能体在复杂高维状态空间中的感知决策问题[1−3]. 在游戏、机器人、推荐系统等领域, 深度强化学习已经取得了突破性进展. 2016 年, 基于深度强化学习和蒙特卡罗树搜索的AlphaGo击败了人类顶尖职业棋手, 引起了全世界的关注[4]. 近日, DeepMind在《Nature》上公布了最新版AlphaGo论文, 介绍了迄今为止最强的围棋人工智能(artificial intelligence,

收稿日期: 2017−11−06; 录用日期: 2017−12−21.
†通信作者. E-mail: dongbin.zhao@ia.ac.cn; Tel.: +86 10-82544764.
本文责任编委: 方勇纯.
国家自然科学基金项目 (61603382, 61573353, 61533017)资助.
Supported by the National Natural Science Foundation of China (61603382, 61573353, 61533017).

AI): AlphaGo Zero[5]. AlphaGo Zero不需要人类专家知识, 只使用纯粹的深度强化学习技术和蒙特卡罗树搜索, 经过3天自我对弈就以100比0击败了上一版本的AlphaGo. AlphaGo Zero证明了深度强化学习的强大能力, 也必将推动以深度强化学习为代表的人工智能领域的进一步发展.

本文主要介绍深度强化学习领域的最新研究进展和AlphaGo Zero的发展历程. 主要结构如下: 首先简要介绍强化学习和深度学习的基本概念; 然后重点介绍基于值函数和基于策略梯度的深度强化学习主要算法进展; 由此引出AlphaGo Zero的原理和特点, 分析AlphaGo Zero与早期版本的改进与不同; 随后介绍深度强化学习在游戏、机器人、自然语言处理、智能驾驶、智能医疗等领域的最新应用成果; 最后作出总结与思考.

2 深度强化学习算法进展(Progress of deep reinforcement learning algorithms)

在人工智能领域, 感知和决策能力是衡量智能的关键性指标. 近几年深度学习和强化学习的发展使得直接从原始的数据中提取高水平特征进行感知决策变成可能[6]. 深度学习起源于人工神经网络. 早期研究人员提出了多层感知机的概念, 并且使用反向传播算法优化多层神经网络, 但是由于受到梯度弥散或爆炸问题的困扰和硬件资源的限制, 神经网络的研究一直没有取得突破性进展. 最近几年, 随着计算资源的性能提升和相应算法的发展, 深度学习在人工智能领域取得了一系列重大突破, 包括图像识别[7]、语音识别[8]、自然语言处理[9]等. 深度学习由于其强大的表征能力和泛化性能受到众多研究人员的关注, 相关技术在学术界和工业界都得到了广泛的研究与应用.

强化学习是机器学习中的一个重要研究领域, 它以试错的机制与环境进行交互, 通过最大化累积奖赏来学习最优策略[10]. 强化学习智能体在当前状态s_t下根据策略π来选择动作a_t. 环境接收该动作并转移到下一状态s_{t+1}, 智能体接收环境反馈回来的奖赏r_t并根据策略选择下一步动作. 强化学习不需要监督信号, 可以在模型未知的环境中平衡探索和利用, 其主要算法有蒙特卡罗强化学习, 时间差分(temporal difference: TD)学习, 策略梯度等[11−12].

强化学习由于其优秀的决策能力在人工智能领域得到了广泛应用. 然而, 早期的强化学习主要依赖于人工提取特征, 难以处理复杂高维状态空间下的问题. 随着深度学习的发展, 算法可以直接从原始的高维数据中提取出特征. 深度学习具有较强的感知能力, 但是缺乏一定的决策能力; 而强化学习具有较强的决策能力, 但对感知问题束手无策. 因此, 将两者结合起来, 优势互补, 能够为复杂状态下的感知决策问题提供解决思路[1].

2.1 深度Q网络及其扩展(Deep Q network and its extensions)

值函数作为强化学习领域的一个基本概念而得到了广泛的应用. 其中, 时间差分学习和Q学习是分别用于求解状态值函数和动作值函数的经典算法. 基于值函数的深度强化学习是一个重要的研究方向.

2015年, DeepMind团队提出了深度Q网络(deep Q network, DQN), 网络框架如图1所示[13]. DQN只使用游戏的原始图像作为输入, 不依赖于人工提取特征, 是一种端到端的学习方式. DQN创新性地将深度卷积神经网络和Q学习结合到一起, 在Atari视频游戏上达到了人类玩家的控制效果. 通过经验回放技术和固定目标Q网络, DQN有效解决了使用神经网络非线性动作值函数逼近器带来的不稳定和发散性问题, 极大提升了强化学习的适用性. 经验回放增加了历史数据的利用率, 同时随机采样打破了数据间的相关性, 与目标Q网络的结合进一步稳定了动作值函数的训练过程. 此外, 通过截断奖赏和正则化网络参数, 梯度被限制到合适的范围内, 从而可以得到更加鲁棒的训练过程.

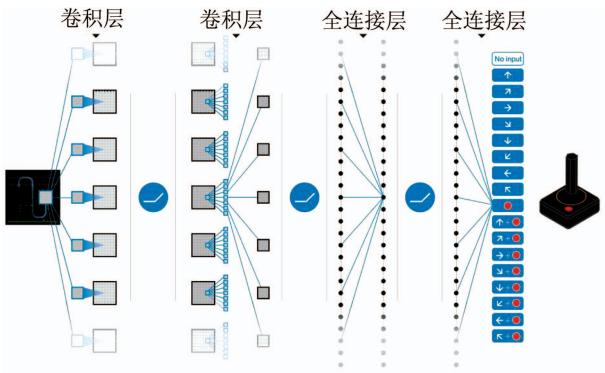


图 1 DQN网络结构图[13]

Fig. 1 The network architecture of DQN[13]

DQN训练过程中使用相邻的4帧游戏画面作为网络的输入, 经过多个卷积层和全连接层, 输出当前状态下可选动作的Q值, 实现了端到端的学习控制. DQN采用带有参数θ的卷积神经网络作为函数逼近器, 并且定期从经验回放池中采样历史数据更新网络参数, 具体的更新过程为

$$\theta_{i+1}=\theta_i+E_{(s,a,r,s')}[(r+\gamma\max_{a'}Q(s',a';\theta_i^-)-Q(s,a;\theta_i))\nabla_{\theta_i}Q(s,a;\theta_i)], \tag{1}$$

其中: s是当前状态, s'是下一时刻状态, a是当前动作, a'是下一时刻动作, r是奖赏信号, γ是折扣因子, θ_i是训练网络的参数, θ_i^-是目标网络的参数.

作为深度强化学习领域的重要开创性工作, DQN的出现引发了众多研究团队的关注. 在文献[1]中, 介绍了DQN早期的主要改进工作, 包括大规模分

布式 DQN[14]、双重 DQN[15]、带优先级经验回放的DQN[16]、竞争架构 DQN[17]、引导 DQN[18]以及异步DQN[19]等. 这些工作从不同角度改进DQN的性能.

此后, 研究人员又陆续提出了一些DQN的重要扩展, 继续完善 DQN 算法. Zhao 等基于在策略 (on-policy) 强化学习, 提出了深度 SARSA(state-action-reward-state-action)算法[20]. 实验证明在一些Atari视频游戏上, 深度 SARSA 算法的性能要优于 DQN. Anschel等提出了平均DQN, 通过取Q值的期望以降低目标值函数的方差, 改善了深度强化学习算法的不稳定性[21]. 实验结果表明, 平均DQN在ALE测试平台上的效果要优于DQN和双重DQN. He等在DQN的基础上提出一种约束优化算法来保证策略最优和奖赏信号快速传播[22]. 该算法极大提高了DQN的训练速度, 在ALE平台上经过一天训练就达到了DQN 和双重DQN经过十天训练的效果. 作为DQN的一种变体, 分类DQN算法从分布式的角度分析深度强化学习[23]. 与传统深度强化学习算法中选取累积奖赏的期望不同, 分类DQN将奖赏看作一个近似分布, 并且使用贝尔曼等式学习这个近似分布. 分类DQN算法在Atari视频游戏上的平均表现要优于大部分基准算法. 深度强化学习中参数的噪声可以帮助算法更有效地探索周围的环境, 加入参数噪声的训练算法可以大幅提升模型的效果, 并且能更快地教会智能体执行任务. 噪声DQN 在动作空间中借助噪声注入进行探索性行为, 结果表明带有参数噪声的深度强化学习将比分别带有动作空间参数和进化策略的传统强化学习效率更高[24]. 彩虹(Rainbow)将各类DQN的算法优势集成在一体, 取得目前最优的算法性能, 视为DQN算法的集大成者[25]. DQN算法及其主要扩展如表1所示.

表 1　DQN及其扩展历程

Table 1　Timeline of DQN and its extensions

时间	研究单位	算法名称
2015	Google DeepMind	DQN[13]
2015	Google DeepMind	分布式DQN[14]
2016	Google DeepMind	双重DQN[15]
2016	Google DeepMind	优先级经验回放DQN[16]
2016	Google DeepMind	竞争架构DQN[17]
2016	Standford & DeepMind	引导DQN[18]
2016	Google DeepMind	异步DQN[19]
2017	Technion	平均DQN[21]
2017	Illinois Urbana-Champaign	约束优化DQN[22]
2017	Google DeepMind	分类DQN[23]
2017	Google DeepMind	噪声DQN[24]
2017	Google DeepMind	Rainbow[25]

2.2　A3C及其扩展(A3C and its extensions)

深度强化学习领域另一个重要算法是异步优势actor-critic (asynchronous advantage actor-critic, A3-C)[19], 模型结构如图2所示.

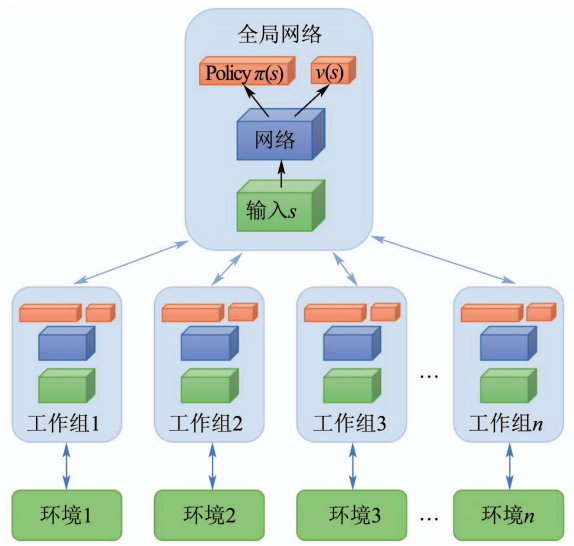


图 2　A3C 模型结构图[1]

Fig. 2　The model architecture of A3C[1]

与 DQN 采用 Q 学习不同, A3C 采用了 actor-critic (AC)这一强化学习算法. actor-critic是一个时序差分算法, critic给出状态s_t价值函数的估计$V(s_t;\theta)$, 对动作的好坏进行评价, 而actor根据状态输出策略$\pi(a_t|s_t;\theta)$, 以概率分布的方式输出. 相比于传统AC算法, A3C基于多线程并行的异步更新算法, 结合优势函数训练神经网络, 大幅度提升AC强化学习算法的样本利用效率. A3C使用多步奖赏信号来更新策略和值函数. 每经过$t_{\max}$步或者达到终止状态, 进行更新. A3C在动作值Q的基础上, 使用优势函数作为动作的评价. 优势函数A是指动作a在状态s下相对其他动作的优势. 采用优势函数A来评估动作更为准确. 在策略参数θ_p、价值参数θ_v、共享参数θ作用下, 损失函数为

$$\nabla_{\theta_p}\log\pi(a_t|s_t;\theta_p)A(s_t,a_t;\theta,\theta_v), \tag{2}$$

其中$A(s_t,a_t;\theta,\theta_v)$是优势函数:

$$A(s_t,a_t;\theta,\theta_v)=R_t-V(s_t;\theta_v), \tag{3}$$

R_t是累积奖赏:

$$R_t=\sum_{i=0}^{k-1}\gamma^i r_{t+i}+\gamma^k V(s_{t+k};\theta_v). \tag{4}$$

A3C中非输出层的参数实现共享, 并且通过一个卷积层和softmax函数输出策略分布π, 以及一个线性网络输出值函数V. 此外, A3C还将策略π的熵加入到损失函数中来鼓励探索, 防止模型陷入局部最优. 完整的损失函数为

$$\nabla_{\theta_p}\log\pi(a_t|s_t;\theta_p)(R_t-V(s_t;\theta_v))+\beta\nabla_{\theta_p}H(\pi(s_t;\theta_p)), \tag{5}$$

[1]https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-8-asynchronous-actor-critic-agents-a3c-c88f72a5e9f2/.

其中: H为熵, β为熵的正则化系数. 策略网络参数θ的更新公式为

$$\theta \leftarrow \theta + \nabla_{\theta_p} \log \pi(a_i|s_i;\theta_p)(R_t - V(s_i;\theta_v')), \quad (6)$$

价值网络参数θ_v的更新公式为

$$\theta_v \leftarrow \theta_v + \partial(R_t - V(s_i;\theta_v'))^2\, \partial\theta_v'. \quad (7)$$

A3C算法采用异步训练的思想, 启动多个训练环境进行采样, 并直接使用采集样本进行训练. 相比DQN 算法, A3C 算法不需要使用经验池存储历史样本, 节省存储空间, 提高数据的采样效率, 以此提升训练速度. 与此同时, 采用多个不同训练环境采集样本, 样本的分布也更加均匀, 更有利于神经网络的训练. A3C算法在以上多个环节上做出了改进, 使得其在Atari游戏上的平均成绩是DQN算法的4倍.

A3C算法由于其优秀的性能, 很快成为了深度强化学习领域新的基准算法. 传统的A3C算法中每一个异步智能体拥有一个独立的模型, 随后一起同步地更新模型. Wu等提出了批量A3C算法, 每个智能体在同一个模型中做出行动, 最后进行批量地更新[26]. 批量A3C可以提高数据的利用效率, 加快模型的收敛. 基于批量A3C算法的游戏AI最终在VizDoom比赛中获得了最佳名次. 传统A3C使用的是中央处理器(central processing unit, CPU)的多线程进行异步训练, 没有充分利用图形处理器(graphic processing unit, GPU)的并行计算. Babaeizadeh等提出了基于CPU和GPU混合架构的GPU-A3C(GA3C)[27]. 通过引入一种队列系统和动态调度策略, GA3C能有效利用GPU的计算能力, 大幅提升了原始A3C的训练速度.

Jaderberg等在A3C的基础上做了进一步扩展, 提出了非监督强化辅助学习(unsupervised reinforcement and auxiliary learning, UNREAL) 算法[28]. UNREAL算法在训练A3C的同时, 训练多个辅助任务来改进算法, 其中包含了两类辅助任务, 第一种是控制任务, 包括像素控制和隐层激活控制, 另一种是回馈预测任务.

UNREAL算法本质上是通过训练多个面向同一个最终目标的任务来提升动作网络的表达能力和水平, 这样提升了深度强化学习的数据利用率, 在A3C算法的基础上对性能和速度进行进一步提升. 实验结果显示, UNREAL在Atari游戏上取得了人类水平8.8倍的成绩, 并且在第一视角的3D迷宫环境Labyrinth上也达到了87%的人类水平.

Wang 等基于长短时记忆网络 (long short-term memory, LSTM), 提出堆栈LSTM-A3C算法, 通过与元强化学习的结合, 在不同任务间拥有良好的泛化能力[29]. 从传统的A3C到后续的UNREAL以及堆栈LSTM-A3C算法得到了越来越广泛的研究, 其主要扩展如表2所示.

表 2 A3C及其扩展历程

Table 2 Timeline of A3C and its extensions

时间	研究单位	算法名称
2016	Google DeepMind	A3C[19]
2017	Facebook	批量A3C[26]
2017	Nvidia	GA3C[27]
2017	Google DeepMind	UNREAL[28]
2017	Google DeepMind	堆栈LSTM-A3C[29]

DQN和A3C等深度强化学习算法都可用于离散动作空间, 各自都可以提升深度强化学习性能的某个方面. 而且它们构建在同一个框架上, 能够被整合起来. 实验结果证明了这些算法很大程度上是互补的. 表3给出了深度强化学习算法在ALE平台上的性能比较[25], 其中no-ops表示智能体在训练开始后的一定步数内不采取动作, 以获取一些训练数据. human-start表示智能体在训练开始后先使用人类玩家的游戏数据初始化, 再使用强化学习训练. Rainbow在Atari视频游戏基准测试平台的数据效率和最终结果上都达到了业界最佳水平.

表 3 深度强化学习算法在ALE平台上的性能比较[25]

Table 3 The performance comparison of deep reinforcement learning algorithms on ALE[25]

算法	no-ops/%	human-start/%
DQN	79	68
双重DQN	117	110
优先级的双重DQN	140	128
竞争架构双重DQN	151	117
A3C	—	116
噪声DQN	118	102
分布式DQN	164	125
Rainbow	**223**	**153**

2.3 策略梯度深度强化学习及其扩展 (Policy-based deep reinforcement learning and its extensions)

基于值函数的深度强化学习主要应用于离散动作空间的任务. 面对连续动作空间的任务, 基于策略梯度的深度强化学习算法能获得更好的决策效果.

目前的大部分actor-critic算法都是采用在策略的强化学习算法. 这意味着无论使用何种策略进行学习, critic部分都需要根据当前actor的输出作用于环境产生的反馈信号才能学习. 因此, 在策略类型的actor-critic算法是无法使用类似于经验回放的技术提升学习效率的, 也由此带来训练的不稳定和难以收敛性. Lillicrap等提出的深度确定性策略梯度算法 (deep deterministic policy gradient, DDPG), 将DQN算法在离散控制任务上的成功经验应用到连续控制任务的

研究[30]. DDPG是无模型、离策略(off-policy)的actor-critic 算法, 使用深度神经网络作为逼近器, 将深度学习和确定性策略梯度算法有效地结合在一起. DDPG源于确定性策略梯度(determinist policy gradient, DPG)算法[31]. 确定性策略记为$\pi_\theta(s)$, 表示状态S和动作A在参数θ的策略作用下得到$S \mapsto A$. 期望奖赏$J(\pi)$如下所示:

$$J(\pi_\theta)=\int_S d^\pi(s)\int_{S'} f(s,\pi(s),s')r(s,\pi(s),s')\mathrm{d}s'\mathrm{d}s, \tag{8a}$$

$$\nabla_\theta J(\pi_\theta)=\int_S d^\pi(s)\nabla_\theta\pi_\theta(s)\nabla_a Q^\pi(s,a)|_{a=\pi_\theta(s)}\mathrm{d}s, \tag{8b}$$

其中: f为状态转移概率密度函数, π_θ为策略函数, 上式的参数皆为连续型变量. 由于确定性策略的梯度分布是有界的, 随着迭代次数的增长, 随机性策略梯度(stochastic policy gradient, SPG)分布的方差会趋于0, 进而得到确定性策略. 将随机性与确定性策略梯度作比较, SPG 算法需要同时考虑状态和动作空间, 然而DPG 算法只需要考虑状态空间. 这样使得DPG算法的学习效率要优于SPG算法, 尤其是在动作空间的维度较高时, DPG算法的优势更为明显.

DDPG是在DPG的基础上结合actor-critic算法扩展而来, 该算法充分借鉴了DQN的成功经验即经验回放技术和固定目标Q网络, 将这两种技术成功移植到策略梯度的训练算法中. DDPG中的actor输出$\pi_\theta(s)$和critic输出$Q_w(s,a)$都是由深度神经网络所得. critic部分的参数更新算法和DQN 类似, 而actor部分的参数更新则是通过DPG算法所得:

$$\mathrm{E}_{\pi'_\theta}\{\nabla_a Q_w(s,a)|_{s=s_t,a=\pi_\theta(s_t)}\nabla_\theta\pi_\theta(s)|_{s=s_t}\}, \tag{9}$$

其中的期望值对应相应的行为策略. 在更新过程中, DDPG采用经验回放技术, 使用探索策略从环境中采样状态转移样本, 将样本储存到记忆池中, 每次更新时从记忆池中均匀采样小批量样本. 由于DDPG需要应用于连续性控制的任务, 因此相比于DQN的固定目标Q网络, DDPG的固定目标Q网络的更新算法要更加平滑. 不同于DQN直接将训练网络权值复制到目标网络中, DDPG则是采用类似惯性更新的思想对目标网络参数进行更新:

$$\theta' = \tau\theta + (1-\tau)\theta', \tag{10a}$$

$$w' = \tau w + (1-\tau)w'. \tag{10b}$$

探索策略π'是在确定性策略π_θ的基础上添加噪声过程N所得, 具体为$\pi'(s)=\pi_\theta(s)+N$. 因而在保证策略搜索稳定的前提下, 增加对未知区域的探索, 以避免陷入到局部最优的情形.

基于策略的强化学习算法需要有好的策略梯度评估器, 因而必须根据对应的策略参数得到相应的期望奖赏的梯度. 但是大多数的策略梯度算法难以选择合适的梯度更新步长, 因而实际情况下评估器的训练常处于振荡不稳定的状态. Schulman等提出可信域策略优化(trust region policy optimization, TRPO) 处理随机策略的训练过程, 保证策略优化过程稳定提升, 同时证明了期望奖赏呈单调性增长[32]. TRPO中策略π的更新公式如下所示:

$$\begin{aligned}&\max L_{\pi_{\pi_{\theta'}(\pi_\theta)}},\\ &\text{s.t. } \bar{D}_{\mathrm{KL}}(\pi_{\theta'}||\pi_\theta)\leqslant\delta,\end{aligned} \tag{11}$$

其中策略$\pi_{\theta'}$为优化前的策略函数. TRPO采用基于平均KL散度(Kullback-Leibler divergence: 也称相对熵)的启发式逼近器对KL散度的取值范围进行限制, 替换此前的惩罚项因子, 并在此基础上使用蒙特卡罗模拟的算法作用在目标函数和约束域上, 得到

$$\begin{aligned}&\max_\theta \mathrm{E}\{\frac{\pi_\theta(s,a)}{\pi_{\theta'}(s,a)}A_{\theta'}(s,a)\},\\ &\text{s.t. } \mathrm{E}\{D_{\mathrm{KL}}(\pi_{\theta'}(s,\cdot)||\pi_\theta(s,\cdot))\}\leqslant\delta.\end{aligned} \tag{12}$$

TRPO在每步的更新过程中必须满足KL散度的约束条件, 一般通过线性搜索实现. 使用线性搜索的原因在于该方法可以在训练过程中避免产生较大更新步长, 影响模型的训练稳定性. 由于深度神经网络通常需要计算大量的参数, TRPO算法使用共轭梯度算法计算自然梯度方向, 避免运算矩阵求逆的过程, 使算法在深度学习领域的应用复杂度降低.

DDPG和TRPO是基于策略梯度的深度强化学习主要算法, 研究人员后续又提出了一些改进算法. 无模型的强化学习算法在样本复杂度较高时, 难以选择合适的高维函数逼近器进行逼近, 这点严重限制了无模型算法的应用. Gu 等提出了标准化优势函数(normalized advantage functions, NAF), 将具有经验回放机制的Q学习算法应用到连续性任务, 并成功应用到机器人仿真控制问题, 将原本只能执行离散任务的Q学习扩展到了连续任务[33]. Wu等基于actor-critic算法框架提出了ACKTR算法 (actor-critic Kronecker-factored trust region, ACKTR)[34]. ACKTR使用Kronecker因子分解, 结合可信域自然梯度法, 以逼近可信域曲线进行学习. 该算法可完成离散和连续两类控制任务. 与之前的在策略actor-critic算法比较, ACKTR算法的平均样本效率可提升2到3倍. Wang等汲取其他深度强化学习算法的优势, 提出了带经验回放的actor-critic 算法 (actor-critic with experience replay, ACER)[35]. ACER采用随机竞争型网络, 根据偏差相关性进行采样, 并且使用高效的可信域策略优化算法, 提升了算法性能. Schulman等提出基于通用优势估计算法(generalized advantage estimation, GAE), 通过价值函数作用, 减少策略梯度方差, 提升模型的训练稳定性[37].

基于策略梯度的深度强化学习和离策略深度强化学习都有各自的优势, 两者结合也是深度强化学习的一个主要方向. O'Donoghue等提出了结合策略梯度和离策略强化学习的策略梯度 Q 学习算法 (policy gradient Q, PGQ), 从而更好地利用历史经验数据[38]. 作者证明了熵正则化策略梯度时, 在贝尔曼方程的不动点处, 动作值函数可以看作是策略对数回归. PGQ学习算法基于值函数的估计, 组合了熵正则化的策略梯度更新和Q学习算法. 实验表明, PGQ在Atari视频游戏上的效果优于 DQN 和 A3C. Nachum 等分析了softmax时序一致性的概念, 概括了贝尔曼方程一致性在离策略Q学习中的应用, 提出路径一致性学习(path consistency learning, PCL)算法 [39]. PCL在基于值函数和基于策略的强化学习之间建立了一种新的联系, 在基准测试上超过A3C和DQN. 无模型深度强化学习算法在很多模拟仿真领域取得了成功, 但由于巨大的采样复杂度难以应用于现实世界. Gu等提出Q–Prop算法, 结合策略梯度算法的稳定性和离策略强化学习算法的采样效率来提高深度强化学习算法性能[40]. 实验结果显示Q–Prop比TRPO, DDPG具有较高的稳定性和采样效率. 相比于值函数算法, 传统策略梯度算法的实现和调参过程都比较复杂. Schulman等提出的近似策略优化(proximal policy optimization, PPO)算法简化了实现过程和调参行为, 而且性能上要优于现阶段其他策略梯度算法[41]. PPO主要使用随机梯度上升, 对策略采用多步更新的算法, 表现出的稳定性和可靠性和TRPO相当.

ACKTR是以actor-critic框架为基础, 引入TRPO使算法稳定性得到保证, 然后加上Kronecker因子分解以提升样本的利用效率并使模型的可扩展性得到加强. ACKTR 相比于TRPO在数据利用率和训练鲁棒性上都有所提升, 因而训练效率更高. PPO和TRPO一样以可信域算法为基础, 以策略梯度算法作为目标更新算法, 但PPO相比于TRPO, 只使用一阶优化算法, 并对代理目标函数简单限定约束, 实现过程更为简便但表现的性能更优. 基于策略的深度强化学习发展历程如表4所示.

表 4 基于策略的深度强化学习历程

Table 4 Timeline of policy-based deep reinforcement learning

时间	研究单位	算法名称
2014	Google DeepMind	DPG[31]
2015	Google DeepMind	DDPG[30]
2015	UC Berkeley	TRPO[32]
2015	Google DeepMind	SVG[36]
2017	University of Toronto	ACKTR[34]
2017	Google DeepMind	PGQ[38]
2017	Google Brain	PCL[39]
2017	University of Cambridge	Q–Prop[40]
2017	OpenAI	PPO[41]

表5给出了6种典型的深度强化学习的算法特点以及在Atari游戏的表现性能比较. 需要指出, 表现性能具体参考了文献[13, 17, 19, 32, 34, 41]的实验结果, 根据6种算法在相同40款Atari游戏的得分情况后计算所得. 具体计算方法是以DQN在Atari游戏的得分表现作为基准, 计算其他算法在同款游戏的得分增长率, 最终以各个游戏的得分增长率的平均值作为衡量标准.

表 5 典型的深度强化学习算法特点及性能比较

Table 5 Characteristic and performance comparison of classical deep reinforcement learning algorithms

算法	算法特点	Atari游戏表现
DQN	经验回放技术, 异步更新目标网络	100%(DQN表现作为基准)
Dueling DQN	竞争型网络结构, 提升网络更新效率	151.72%
A3C	异步多线程优势函数作用网络更新	163.07%
TRPO	理论保证单调提升, 但训练耗时较长	实验游戏数量较少, 且表现性能较差
ACKTR	使用K–FAC因式分解, 降低梯度计算复杂度, 提升算法样本利用率	353.87%
PPO	具有TRPO算法的稳定性和可靠性, 算法复杂度较低	46.26%

2.4 其他深度强化学习算法(Other deep reinforcement learning algorithms)

除了上述深度强化学习算法, 还有深度迁移强化学习、分层深度强化学习、深度记忆强化学习以及多智能体深度强化学习等算法.

2.4.1 深度迁移强化学习算法(Deep transfer reinforcement learning algorithms)

传统深度强化学习算法每次只能解决一种游戏任务, 无法在一次训练中完成多种任务. 迁移学习和强化学习的结合也是深度强化学习的一种主要思路.

Parisotto等提出了一种基于行为模拟的深度迁移强化学习算法[42]. 该算法通过监督信号的指导, 使得单一的策略网络学习各自的策略, 并将知识迁移到新任务中. Rusa等提出策略蒸馏(policy distillation)深度迁移强化学习算法[43]. 策略蒸馏算法中分为学习网络和指导网络, 通过这两个网络Q值的偏差来确定目标函数, 引导学习网络逼近指导网络的值函数空间. 此后, Rusa等又提出了一种基于渐进神经网络(progressive neural networks, PNN)的深度迁移强化学习算法[44]. PNN是一种把神经网络和神经网络连起来的算法. 它在一系列序列任务中, 通过渐进的方式来存储知识和提取特征, 完成了对知识的迁移. PNN最终实现多个独立任务的训练, 通过迁移加速学习过程, 避免灾难性遗忘. Fernando 等提出了路径网络 (PathNet)[45]. PathNet可以说是PNN的进阶版. PathNet把网络中每一层都看作一个模块, 把构建一个网络看成搭积木, 也就是复用积木. 它跟PNN非常类似, 只是这里不再有列, 而是不同的路径. PathNet将智能体嵌入到神经网络中, 其中智能体的任务是为新任务发现网络中可以复用的部分. 智能体是网络之中的路径, 其决定了反向传播过程中被使用和更新的参数范围. 在一系列的Atari强化学习任务上, PathNet都实现了正迁移, 这表明PathNet在训练神经网络上具有通用性应用能力. PathNet也可以显著提高A3C算法超参数选择的鲁棒性. Schaul等提出了一种通用值函数逼近器(universal value function approximators, UVFAs)来泛化状态和目标空间[50]. UVFAs可以将学习到的知识迁移到环境动态特性相同但目标不同的新任务中.

2.4.2 分层深度强化学习算法(Hierarchical deep reinforcement learning algorithms)

分层强化学习可以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. Kulkarni等提出了分层DQN (hierarchical deep Q-network, h–DQN) 算法[46]. h–DQN基于时空抽象和内在激励分层, 通过在不同的时空尺度上设置子目标对值函数进行层次化处理. 顶层的值函数用于确定宏观决策, 底层的值函数用于确定具体行动. Krishnamurthy等在h–DQN的基础上提出了基于内部选择的分层深度强化学习算法[47]. 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, 避免了特定的内在激励和人工设定中间目标, 加速了智能体的学习进程, 同时也增强了模型的泛化能力. Kulkarni等基于后续状态表示法提出了深度后续强化学习 (deep successor reinforcement learning, DSRL)[48]. DSRL通过阶段性地分解子目标和学习子目标策略, 增强了对未知状态空间的探索, 使得智能体更加适应那些存在延迟反馈的任务. Vezhnevets等受封建(feudal)强化学习算法的启发, 提出一种分层深度强化学习的架构FeUdal网络(FuNs)[49]. FuNs框架使用一个管理员模块和一个工人模块. 管理员模块在较低的时间分辨率下工作, 设置抽象目标并传递给工人模块去执行. FuNs框架创造了一个稳定的自然层次结构, 并且允许两个模块以互补的方式学习. 实验证明, FuNs有助于处理长期信用分配和记忆任务, 在Atari视频游戏和迷宫游戏中都取得了不错的效果.

2.4.3 深度记忆强化学习算法(Deep memory reinforcement learning algorithms)

传统的深度强化学习模型不具备记忆、认知、推理等高层次的能力, 尤其是在面对状态部分可观察和延迟奖赏的情形时. Junhyuk等通过在传统的深度强化学习模型中加入外部的记忆网络部件和反馈控制机制, 提出反馈递归记忆Q网络(feedback recurrent memory Q-network, FRMQN))[51]. FRMQN模型具备了一定的记忆与推理功能, 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, 评估动作值函数并做出决策. FRMQN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. 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.

Blundell等设计出一种模型无关的情节控制算法(model-free episode control, MFEC)[52]. MFEC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系统中, 使得智能体在面对一些复杂的决策任务时, 能快速达到人类玩家的水平. MFEC通过反向经验回放, 使智能体拥有初步的情节记忆. 实验表明, 基于MFEC算法的深度强化学习不仅可以在Atari游戏中学习到有效策略, 还可以处理一些三维场景的复杂任务. Pritzel等在MFEC的基础上进一步提出了神经情节控制 (neural episodic control, NEC), 有效提高了深度强化学习智能体的记忆能力和学习效率 [53]. NEC能快速吸收新经验并依据新经验来采取行动. 价值函数包括价值函数渐变状态表示和价值函数快速更新估计两部分. 大量场景下的研究表明, NEC的学习速度明显快于目前最先进的通用深度强化学习智能体.

2.4.4 多智能体深度强化学习算法(Multi-agent deep reinforcement learning algorithms)

在一些复杂场景中, 涉及到多智能体的感知决策问题, 这时需要将单一模型扩展为多个智能体之间相互合作、通信及竞争的多智能体深度强化学习系统. Foerster等提出了一种称为分布式深度递归Q网络(deep distributed recurrent Q-networks, DDRQN) 的模型, 解决了状态部分可观测状态下的多智能体通信与合作的挑战性难题 [54]. 实验表明, 经过训练的DDRQN模型最终在多智能体之间达成了一致的通信协

议, 成功解决了经典的红蓝帽子问题.

让智能体学会合作与竞争一直以来都是人工智能领域内的一项重要研究课题, 也是实现通用人工智能的必要条件. Lowe等提出了一种用于合作–竞争混合环境的多智能体actor-critic算法(multi-agent deep deterministic policy gradient, MADDPG)[55]. MADDPG对DDPG强化学习算法进行了延伸, 可实现多智能体的集中式学习和分布式执行, 让智能体学习彼此合作和竞争. 在多项测试任务中, MADDPG的表现都优于DDPG.

2.5 深度强化学习算法小结 (Summary of deep reinforcement learning algorithms)

基于值函数概念的DQN及其相应的扩展算法在离散状态、离散动作的控制任务中已经表现了卓越的性能, 但是受限于值函数离散型输出的影响, 在连续型控制任务上显得捉襟见肘. 基于策略梯度概念的, 以DDPG, TRPO等为代表的策略型深度强化学习算法则更适用于处理基于连续状态空间的连续动作的控制输出任务, 并且算法在稳定性和可靠性上具有一定的理论保证, 理论完备性较强. 采用actor-critic架构的A3C算法及其扩展算法, 相比于传统DQN算法, 这类算法的数据利用效率更高, 学习速率更快, 通用性、可扩展应用性更强, 达到的表现性能更优, 但算法的稳定性无法得到保证. 而其他的如深度迁移强化学习、分层深度强化学习、深度记忆强化学习和多智能体深度强化学习等算法都是现在的研究热点, 通过这些算法能应对更为复杂的场景问题、系统环境及控制任务, 是目前深度强化学习算法研究的前沿领域.

3 从AlphaGo到AlphaGo Zero(From AlphaGo to AlphaGo Zero)

3.1 AlphaGo概述(Introduction of AlphaGo)

人工智能领域一个里程碑式的工作是由DeepMind在2016年初发表于《Nature》上的围棋AI: AlphaGo[4]. AlphaGo的胜利对整个围棋领域AI的研究产生了极大的促进作用. 达到人类围棋职业选手顶尖水平的围棋AI如腾讯的绝艺、日本的DeepZenGo等, 都深受AlphaGo的影响. AlphaGo的问世将深度强化学习的研究推向了新的高度. 它创新性地结合深度强化学习和蒙特卡罗树搜索, 通过策略网络选择落子位置降低搜索宽度, 使用价值网络评估局面以减小搜索深度, 使搜索效率得到了大幅提升, 胜率估算也更加精确. 与此同时, AlphaGo使用强化学习的自我博弈来对策略网络进行调整, 改善策略网络的性能, 使用自我对弈和快速走子结合形成的棋谱数据进一步训练价值网络. 最终在线对弈时, 结合策略网络和价值网络的蒙特卡罗树搜索在当前局面下选择最终的落子位置.

AlphaGo成功地整合了上述算法, 并依托强大的硬件支持达到了顶尖棋手的水平. 文献[1]介绍了AlphaGo的技术原理, 包括线下学习和在线对弈的具体过程. 分析了AlphaGo成功的原因以及当时存在的问题. 此后, DeepMind对AlphaGo做了进一步改进, 并先后战胜了李世石、柯洁以及60多位人类顶尖围棋选手, 显示出了自己强大的实力.

3.2 AlphaGo Zero概述 (Introduction of AlphaGo Zero)

在AlphaGo的基础上, DeepMind进一步提出了AlphaGo Zero[5]. AlphaGo Zero的出现, 再一次引发了各界对深度强化学习算法和围棋AI的关注与讨论. AlphaGo Fan(和樊麾对弈的AlphaGo)和AlphaGo Lee(和李世石对弈的AlphaGo)都采用了策略网络和价值网络分开的结构, 其中策略网络先模仿人类专业棋手的棋谱进行监督学习, 然后使用策略梯度强化学习算法进行提升. 在训练过程中, 深度神经网络与蒙特卡罗树搜索算法相结合形成树搜索模型, 本质上是使用神经网络算法对树搜索空间的优化.

AlphaGo Zero与之前的版本有很大不同, 如表6所示.

表 6 各版本AlphaGo对比

Table 6 Comparison among all versions of AlphaGo

	AlphaGo Zero	AlphaGo Master	AlphaGo Lee	AlphaGo Fan
神经网络	1个共享网络	策略、价值、走子网络	策略、价值、走子网络	策略、价值、走子网络
Elo	5,185	4,858	3,739	3,144
运行阶段硬件需求	单机4块TPU	单机4块TPU	48块TPU+176块GPU	48 块TPU+176 块GPU
训练时间	40天(36小时Elo超越Lee)	未说明	数月	未说明
专家棋谱作用	未使用	AlphaGo Lee产生的棋谱	KGS数据集	KGS数据集

1) 神经网络权值完全随机初始化. AlphaGo Zero不利用任何人类专家的经验或数据, 随机初始化神经网络的权值进行策略选择, 随后使用深度强化学习进行自我博弈和提升.

2) 无需先验知识. AlphaGo Zero不再需要人工设计特征, 而是仅利用棋盘上的黑白棋子的摆放情况作为原始数据输入到神经网络中, 以此得到结果.

3) 神经网络结构复杂性降低. AlphaGo Zero将原

先两个结构独立的策略网络和价值网络合为一体, 合并成一个神经网络. 在该神经网络中, 从输入层到中间层的权重是完全共享的, 最后的输出阶段分成了策略函数输出和价值函数输出.

4) 舍弃快速走子网络. AlphaGo Zero不再使用快速走子网络替换随机模拟, 而是完全将神经网络得到的结果替换为随机模拟, 从而在提升学习速率的同时, 增强了神经网络估值的准确性.

5) 神经网络引入残差结构. AlphaGo Zero的神经网络采用基于残差网络结构的模块进行搭建, 用更深的神经网络进行特征表征提取, 从而在更加复杂的棋盘局面中进行学习.

6) 硬件资源需求更少. 以前Elo[2]评分最高的AlphaGo Fan需要1920块CPU和280块GPU才能完成执行任务, AlphaGo Lee则减少到176块GPU和48块TPU, 而到现在的AlphaGo Zero只需要单机4块TPU便可完成, 如图3所示.

7) 学习时间更短. AlphaGo Zero仅用3天的时间便达到AlphaGo Lee的水平, 21天后达到AlphaGo Master水平, 棋力快速提升, 如图4所示.

从影响因素的重要程度而言, AlphaGo Zero棋力提升的关键因素可以归结为两点, 一是使用基于残差模块构成的深度神经网络, 不需要人工制定特征, 通过原始棋盘信息便可提取相关表示特征; 二是使用新的神经网络构造启发式搜索函数, 优化蒙特卡罗树搜索算法, 使用神经网络估值函数替换快速走子过程, 使算法训练学习和执行走子所需要的时间大幅减少.

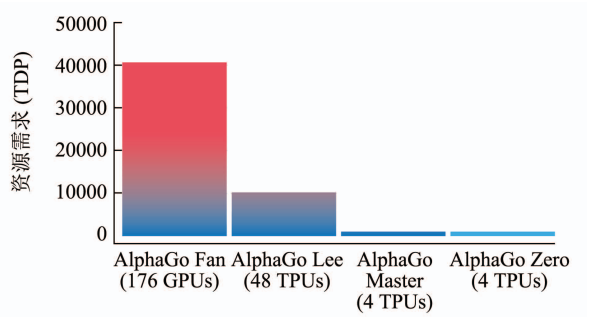
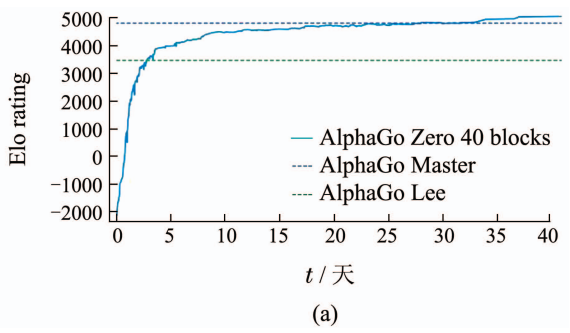


图 3　各个版本AlphaGo的硬件资源需求[5]

Fig. 3　Hardware resource requirements of different versions of AlphaGo[5]



[2]棋类游戏常用的评分系统, 命名源于其创始人Arpad Elo.

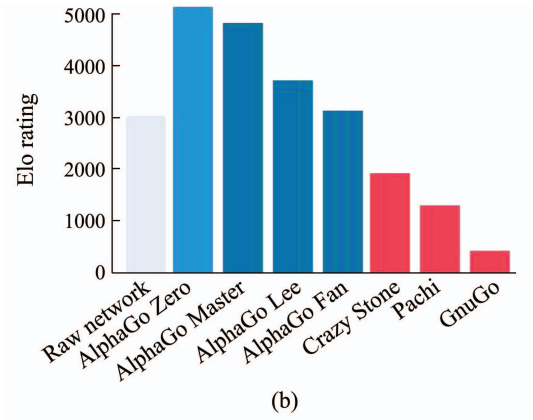


图 4　AlphaGo Zero的训练过程[5]

Fig. 4　The training process of AlphaGo Zero[5]

作为AlphaGo Zero关键技术之一的深度残差网络, 由何恺明等在2016年提出[56]. 深度残差网络是真正意义上的“深度学习”, 与其他深度神经网络模型相比, 深度残差网络能进行成百乃至上千层的网络学习, 并且在多项极具挑战性的识别任务, 如ImageNet和微软COCO等比赛中均取得当下最佳成绩, 体现深度网络之深对特征表征提取的重要性. 深度残差网络由多层“残差单元”堆叠而成, 其通式表达为

$$y_l = h(x_l) + \boldsymbol{F}(x_l, \boldsymbol{W}_l), \tag{13a}$$

$$x_{l+1} = \boldsymbol{f}(y_l), \tag{13b}$$

其中: $\boldsymbol{W}_l$是神经网络权值, y_l是中间输出, x_l和x_{l+1}分别是第l个单元的输入和输出, $\boldsymbol{F}$是一个残差函数, h是恒等映射, f为常用ReLU函数的激活函数. 残差网络与其他常见的卷积型前向神经网络的最大不同在于多了一条跨层传播直连接通路, 使得神经网络在进行前向传播和后向传播时, 传播信号都能从一层直接平滑地传递到另一指定层. 残差函数引入批归一化(batch normalization, BN)作优化, 使神经网络输出分布白化, 从而使数据归一化来抑制梯度弥散或是爆炸现象[57].

3.3　AlphaGo Zero 的深度神经网络结构 (Deep neural network architecture in AlphaGo Zero)

AlphaGo Zero的深度神经网络结构有两个版本, 分别是除去输出部分的39(19个残差模块)层卷积网络版和79(39个残差模块)层卷积网络版. 两个版本的神经网络除了中间层部分的残差模块个数不同, 其他结构大致相同.

神经网络的输入数据为19×19×17的张量, 具体表示为本方最近8步内的棋面和对方最近8步内的棋面以及本方执棋颜色. 所有输入张量的取值为{0,1}, 即二元数据. 前16个二维数组型数据直接反映黑白双方对弈距今的8个回合内棋面, 以1表示本方已落子状态, 0表示对方已落子或空白状态. 而最后1个的19×

19二维数组用全部元素置0 表示执棋方为白方, 置1表示执棋方为黑方.

由AlphaGo Zero的网络结构图(图5)可见: 输入层经过256个3×3、步长为1的卷积核构成的卷积层, 经过批归一化处理, 以ReLU作为激活函数输出; 中间层为256个3×3、步长为1的卷积核构成的卷积层, 经过两次批归一化处理, 由输入部分产生的直连接信号作用一起进入到ReLU激活函数.

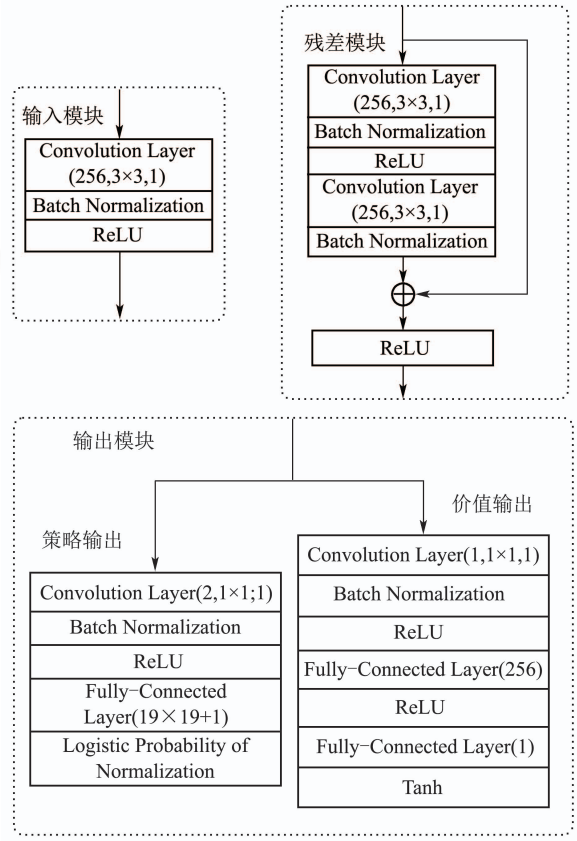


图 5 AlphaGo Zero神经网络结构的3个主要模块

Fig. 5 The three main modules of AlphaGo Zero's neural network architecture

输出部分分为两个部分: 一部分称为策略输出, 含2个1×1卷积核、步长为1的卷积层, 同样经过批归一化和ReLU激活函数作处理, 再连接神经元个数为19^2(棋盘交叉点总数)+1(放弃走子: pass move)=362个线性全连接层. 使用对数概率对所有输出节点作归一化处理, 转换到[0, 1]之间; 另一部分称为估值输出, 含1个1×1卷积核、步长为1的卷积层, 经批归一化和ReLU激活函数以及全连接层, 最后再连接一个激活函数为Tanh 的全连接层, 且该层只有一个输出节点, 取值范围$[-1, 1]$.

输入模块、输出模块及残差模块的具体示意图如图5所示, 图5中各模块代表一个模块单元的基本组成部分、模块结构及相关参数.

3.4 AlphaGo Zero中的蒙特卡罗树搜索(MCTS in AlphaGo Zero)

假设当前棋面为状态s_t, 深度神经网络记作f_θ, 以f_θ的策略输出和估值输出作为蒙特卡罗树搜索的搜索方向依据, 取代原本蒙特卡罗树搜索所需要的快速走子过程. 这样既有效降低蒙特卡罗树搜索算法的时间复杂度, 也使深度强化学习算法在训练过程中的稳定性得到提升.

如图6所示, 搜索树的当前状态为s, 选择动作为a, 各节点间的连接边为$e(s,a)$, 各条边e存储了四元集为遍历次数$N(s,a)$、动作累计值$W(s,a)$, 动作平均值$Q(s,a)$, 先验概率$P(s,a)$. 与AlphaGo以往版本不同, AlphaGo Zero将原来蒙特卡罗树搜索所需要的4个阶段合并成3个阶段, 将原来的展开阶段和评估阶段合并成一个阶段, 搜索过程具体为选择阶段、展开与评估阶段、回传阶段. 最后通过执行阶段选择落子位置.

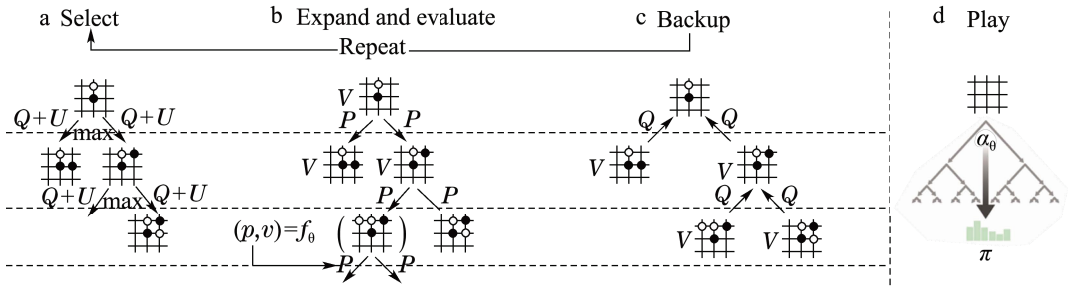


图 6 AlphaGo Zero蒙特卡罗树搜索流程[5]

Fig. 6 The MCTS process of AlphaGo Zero[5]

3.4.1 选择阶段(Select)

假定搜索树的根节点为s_0, 从根节点s_0到叶子节点s_l需要经过的路径长度为L, 在路径L上的每步t中, 根据当前时刻的搜索树的数据存储情况, a_t由下式所得, 选择值对应当前状态s_t的最大动作值作为搜索路径.

$$a_t = \arg\max_a (Q(s_t,a) + U(s_t,a)), \tag{14a}$$

$$U(s_t,a)=c_{\text{puct}}P(s_t,a)\frac{\sqrt{\sum_b N(s_t,b)}}{1+N(s_t,a)}, \tag{14b}$$

$$P(s_t,a)=(1-\epsilon)P(s_t,a)+\epsilon\eta, \tag{14c}$$

其中: c_{puct}是重要的超参数, 平衡探索与利用间的权重分配, 当c_{puct}较大时, 驱使搜索树向未知区域探索, 反之则驱使搜索树快速收敛; $\sum_b N(s_t,b)$表示经过状态s_t的所有次数; $P(s_t,a)$为深度神经网络$f_\theta(s_t)$的策略输出对应动作a的概率值, 并且引入噪声η服从Dirchlet(0.03)分布, 惯性因子$\epsilon=0.25$, 从而使神经网络的估值鲁棒性得到增强. 值得一提, 蒙特卡罗树搜索的超参数c_{puct}是通过高斯过程优化得到, 并且39个残差模块版本与19个残差模块版本的神经网络所用的超参数并不一样, 较深网络的超参数是由较浅网络再次优化后所得.

3.4.2　展开与评估阶段(Expand and evaluate)

在搜索树的叶子节点, 进行展开与评估. 当叶子节点处于状态s_l时, 由神经网络f_θ得到策略输出p_l和估值输出v_l. 然后初始化边$e(s_l,a)$中的四元集: $N(s_l,a)=0, W(s_l,a)=0, Q(s_l,a)=0, P(s_l,a)=p_l$. 在棋局状态估值时, 需要对棋面旋转$n\times 45°, n\in\{0,1,\cdots,7\}$或双面反射后输入到神经网络. 在神经网络进行盘面评估时, 其他并行线程皆会处于锁死状态, 直至神经网络运算结束.

3.4.3　回传阶段(Backup)

当展开与评估阶段完成后, 搜索树中各节点连接边的信息都已经得到. 此时需要将搜索后所得最新结构由叶子节点回传到根节点上进行更新. 访问次数$N(s_t,a_t)$、动作累计值$W(s_t,a_t)$、动作平均值$Q(s_t,a_t)$具体的更新方式为

$$N(s_t,a_t)=N(s_t,a_t)+1, \tag{15a}$$

$$W(s_t,a_t)=W(s_t,a_t)+v_t, \tag{15b}$$

$$Q(s_t,a_t)=\frac{W(s_t,a_t)}{N(s_t,a_t)}, \tag{15c}$$

其中v_t为神经网络$f_\theta(s_t)$的估值输出. 从式中可见, 随着模拟次数的增加, 动作平均值$Q(s_t,a_t)$会逐渐趋于稳定, 且从数值形式上与神经网络的策略输出p_t没有直接关系.

3.4.4　执行阶段(Play)

经过1600次蒙特卡罗树搜索, 树中的各边存储着历史信息, 根据这些历史信息得到落子概率分布$\pi(a|s_0)$, $\pi(a|s_0)$是由叶子节点的访问次数经过模拟退火算法得到, 具体表示为

$$\pi(a|s_0)=\frac{N(s_0,a)^{\frac{1}{\tau}}}{\sum_b N(s_0,b)^{\frac{1}{\tau}}}, \tag{16}$$

其中模拟退火参数τ初始为1, 在前30步走子一直为1, 然后随着走子步数的增加而减小趋向于0. 引入了模拟退火算法后, 极大地丰富围棋开局的变化情况, 并保证在收官阶段能够作出最为有利的选择.

在执行完落子动作后, 当前搜索树的扩展子节点及子树的历史信息会被保留, 而扩展子节点的所有父节点及信息都会被删除, 在保留历史信息的前提下, 减少搜索树所占内存空间. 并最终以扩展节点作为新的根节点, 为下一轮蒙特卡罗树搜索作准备. 值得注意的是, 当根节点的估值输出v_θ小于指定阈值v_{resign}, 则作认输处理. 即此盘棋局结束.

3.5　AlphaGo Zero的训练流程(Training process of AlphaGo Zero)

AlphaGo Zero的训练流程可以分为4个阶段, 如图7所示.

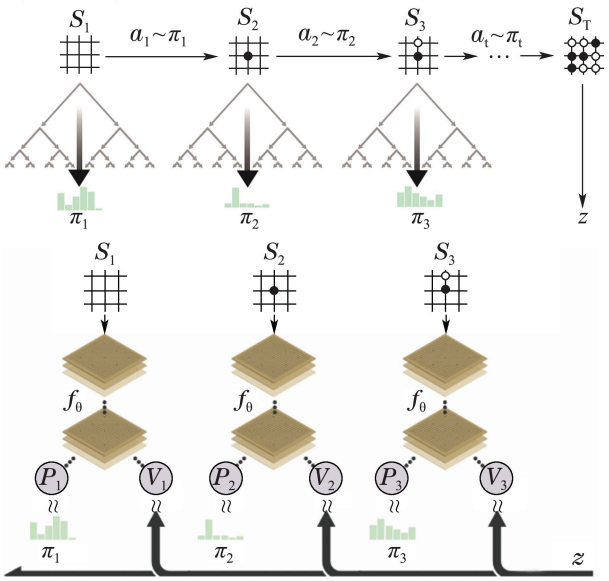


图 7　AlphaGo Zero自我对弈流程[5]

Fig. 7　The self-play process of AlphaGo Zero[5]

第1阶段　假设当前棋面状态为x_t, 以x_t作为数据起点, 得到距今最近的本方历史7步棋面状态和对方历史8步棋面状态, 分别记作$x_{t-1}, x_{t-2}, \cdots, x_{t-7}$和$y_t, y_{t-1}, \cdots, y_{t-7}$. 并记本方执棋颜色为$c$, 拼接在一起, 记输入元$s_t$为$\{x_t, y_t, x_{t-1}, y_{t-1}, \cdots, c\}$. 并以此开始进行评估.

第2阶段　使用基于深度神经网络f_θ的蒙特卡罗树搜索展开策略评估过程, 经过1600次蒙特卡罗树搜索, 得到当前局面x_t的策略π_t和参数θ下深度神经网络$f_\theta(s_t)$输出的策略函数p_t和估值v_t.

第3阶段　由蒙特卡罗树搜索得到的策略π_t, 结合模拟退火算法, 在对弈前期, 增加落子位置多样性, 丰富围棋数据样本. 一直持续这步操作, 直至棋局终了, 得到最终胜负结果z.

第4阶段 根据上一阶段所得的胜负结果z与价值v_t使用均方和误差, 策略函数p_t和蒙特卡罗树搜索的策略π_t使用交叉信息熵误差, 两者一起构成损失函数. 同时并行反向传播至神经网络的每步输出, 使深度神经网络f_θ的权值得到进一步优化.

深度神经网络的输出和损失函数分别为

$$(p_t, v_t) = f_\theta(s_t), \tag{17a}$$

$$l = (z - v_t)^2 - \pi_t^T \log p_t + c||\theta||^2. \tag{17b}$$

3.6 AlphaGo Zero成功的讨论(Discussion on the success of AlphaGo Zero)

AlphaGo Zero的成功证明了在没有人类经验指导的前提下, 深度强化学习算法仍然能在围棋领域出色地完成这项复杂任务, 甚至比有人类经验知识指导时, 达到更高的水平. 在围棋下法上, AlphaGo Zero比此前的版本创造出了更多前所未见的下棋方式, 为人类对围棋领域的认知打开了新的篇章. 就某种程度而言, AlphaGo Zero展现了机器“机智过人”的一面.

现在从以下几个方面对AlphaGo和AlphaGo Zero进行比较.

1) 局部最优与全局最优.

虽然AlphaGo和AlphaGo Zero都以深度学习作为核心算法, 但是核心神经网络的初始化方式却不同. AlphaGo是基于人类专家棋谱使用监督学习进行训练, 虽然算法的收敛速度较快, 但易于陷入局部最优. AlphaGo Zero则没有使用先验知识和专家数据, 避开了噪声数据的影响, 直接基于强化学习以逐步逼近至全局最优解. 最终AlphaGo Zero的围棋水平要远高于AlphaGo.

2) 大数据与深度学习的关系.

传统观点认为, 深度学习需要大量数据作支撑, 泛化性能才会更好. 但是, 数据的采集和整理需要投入大量的精力才能完成, 有时候甚至难以完成. 而AlphaGo Zero另辟蹊径, 不需要使用任何外部数据, 完全通过自学习产生数据并逐步提升性能. 自学习产生的数据可谓取之不尽、用之不竭. 并且伴随智能体水平的提升, 产生的样本质量也会随之提高. 这些恰好满足了深度学习对数据质与量的需求.

3) 强化学习算法的收敛性.

强化学习的不稳定性和难以收敛性一直是被研究者所诟病之处, 而AlphaGo Zero则刷新了人们对强化学习的认知, 给出了强化学习稳定收敛、有效探索的可能性. 那便是通过搜索算法, 对搜索过程进行大量模拟, 根据期望结果的奖赏信号进行学习, 使强化学习的训练过程保持稳定提升的状态. 但目前相关理论支持仍不完善, 还需要开展更多工作进行研究.

4) 算法的“加法”和“减法”.

研究AlphaGo Zero的成功会发现以往性能优化的研究都是在上一个算法的基础上增添技巧或外延扩展, 丰富之前的研究, 归结为做加法的过程. 而AlphaGo Zero却与众不同, 是在AlphaGo的基础上作减法, 将原来复杂的3个网络模型缩减到一个网络, 将原来复杂的蒙特卡罗树搜索的4个阶段减少到3个阶段, 将原来的多机分布式云计算平台锐减到单机运算平台, 将原来需要长时间训练的有监督学习方式彻底减掉. 每一步优化都是由繁到简、去粗取精的过程. 使AlphaGo摆脱了冗余方法的束缚, 轻装上阵, 在围棋领域成为一代宗师. 相信这样的“减法”思维定将在未来产生更加深远的影响, 创造出更多令人赞叹的新发明、新技术.

目前来看, AlphaGo中神经网络的成功主要还是基于卷积神经网络, 但是下围棋是一个动态持续的过程, 因此引入递归神经网络是否能对AlphaGo的性能有所提升也是一个值得思考的问题. AlphaGo Zero所蕴含的算法并非是石破天惊、复杂无比, 相反这里面的很多算法都早已被前人提出及实现. 但是以前, 这些算法尤其是深度强化学习等算法, 通常只能用来处理规模较小的问题, 在大规模问题上难以做到无师自通. AlphaGo Zero的成功则刷新了人们对深度强化学习算法的认识, 并对深度强化学习领域的研究更加充满期待. 深度学习与强化学习的进一步结合相信会引发更多的思想浪潮. 深度学习已经在许多重要的领域被证明可以取代人工提取特征得到更优结果. 而深度学习在插上了强化学习的翅膀后更是如虎添翼, 甚至有可能颠覆传统人工智能领域, 进一步巩固和提升机器学习在人工智能领域的地位.

4 深度强化学习应用进展(Application progress of deep reinforcement learning)

近两年来, 深度强化学习在游戏、机器人、自然语言处理、智能驾驶和智能医疗等诸多领域得到了更加广泛的应用推广.

4.1 游戏(Games)

传统游戏AI主要是基于专家知识库和推理系统实现的. 随着机器学习领域的不断发展, 逐渐有基于人工特征、神经网络、蒙特卡罗树搜索等算法出现, 但受特征工程的制约所取得的水平有限[58–59]. 近几年, 基于端到端的深度强化学习在游戏上取得了广泛的应用成果, 包括Atari视频游戏、棋类游戏、第一人称射击游戏、即时战略游戏等[60]. 基于深度强化学习的算法不需要人工提取特征便可完成游戏任务, 在个别游戏中甚至超越了人类顶尖玩家.

目前, 许多公司或组织开放了深度强化学习算法的测试平台, 方便研究者或工程师对自己的深度强化学习算法性能进行测试. 最早提供标准测试平台的

是Bellemare等于2013年开放的街机游戏测试环境(arcade learning environment, ALE)[61]. 在ALE平台上, 研究人员进行了一系列算法研究, 极大推动了深度强化学习从一个新兴的领域走向标准化与成熟. 在2016年, OpenAI的Brockman等预见到深度强化学习的发展迫切需要一个统一的标准平台用于算法的测试和比较, 发布了整合多款强化学习测试环境的OpenAI Gym[62], 其成为首个将强化学习的绝大部分测试环境集成在一起的强大测试平台[3]. 随着深度强化学习逐渐向视频游戏领域方向发展, OpenAI在Gym的基础上开发出了更加全面的测试平台Universe. Universe为视频游戏提供更多接口, 方便研究者对视频游戏使用深度强化学习的算法展开研究. 身为深度强化学习领域的奠基者之一的DeepMind公司也不甘落后, 开放了自己的内部测试平台DeepMind Lab, 该平台主要提供3D迷宫游戏作为测试基准, 鼓励研究人员使用深度强化学习算法提升智能体路径规划、目标导航、物体识别等能力[63]. 其他类型的平台包括赛车驾驶游戏Torcs[64]、多智能体协作和导航的MineCraft[65]、第一人称射击的VizDoom[66]和即时战略游戏星际争霸II的SC2LE[67]等.

在星际争霸的局部对抗任务中, Peng等提出一种多智能体actor-critic模型[68]. 通过自动编组和构建全局和个体奖赏, 实现了多个单元间的协调作战, 并使用双向RNN网络实现了端到端的策略学习. 同样在星际争霸微操任务中, Usunier等通过贪心推理打破单元每步动作的复杂性, 使用零阶优化强化学习算法解决探索问题, 并且通过混合参数随机性和简单梯度下降直接在策略空间探索[69]. 这种算法很好地解决了星际争霸微操中非完全信息多智能体的对抗博弈问题. Shao等通过高效的状态表示降低了星际争霸局部对抗任务的复杂度, 使用内在激励的资格迹强化学习算法实现了多个智能体的协同决策, 并战胜了内置AI[70].

目前, 深度强化学习仍未完全攻克游戏智能领域. 本质上, AlphaGo Zero解决的是启发式搜索的问题, 并没有展现出类似于DQN在Atari视频游戏中那样普遍适用的泛化性能. 基于深度强化学习的蒙特卡罗树搜索虽然在回合制游戏上已经取得了成功, 但是由于搜索算法与生俱来的搜索时间与空间的开销, 也许对回合制类游戏影响不大, 但是对实时类游戏的影响却是巨大的.在如同星际争霸这类实时游戏中, 如何解决好时间开销与游戏连续性的矛盾则是一个值得深思的问题. 2017年10月31日, 人类职业玩家以4:0的压倒性优势轻松战胜了星际争霸I游戏的顶级AI, 其中包括FaceBook公司使用机器学习算法所做的CherryPi.

相较于其他类型游戏, 星际争霸类的实时战略游戏是由实时性需求、态势感知与估计、非完全信息博弈和多智能体协同等多个问题构成的复杂性系统问题. 基于深度强化学习算法的DeepStack在非完全信息博弈的典型游戏“一对一无限注德州扑克”已具备职业玩家的水平[71], DeepStack的成功会给非完全信息博弈问题的解决带来启发. 在需要多智能体协同配合完成的中小规模层次的实时作战任务, 基于主–从级结构的多智能体深度强化学习算法中取得了令人满意的效果[72]. 然而传统方法在这些问题的表现结果则十分有限. 由此可知, 随着游戏AI的研究不断深入, 从简单的Atari到复杂的星际争霸, 传统算法逐渐难以满足复杂游戏任务的需求. 因而需要更多类似深度强化学习的算法, 向复杂的游戏任务发起挑战.

4.2 机器人(Robotics)

传统的强化学习很早便应用于机器人控制领域, 如倒立摆系统平衡、二级倒立摆平衡等非线性连续控制任务. Zhu等使用自适应动态规划算法研究这些问题, 并取得了令人满意的效果[73–75].

然而传统强化学习算法难以处理高维状态空间的决策问题, 深度强化学习为这一问题提供了解决思路. Schulman等人提出了TRPO算法, 在理论上保证强化学习算法可以单调优化, 并成功应用于机器人控制的仿真任务[32]. Levine等以卷积神经网络作为策略特征表示, 提出指导性策略搜索算法(guided policy search, GPS), 将策略搜索转化为监督学习, 以视觉图像作为输入样本, 实现直接端到端的从眼到手的机械臂操作控制 [76]. 为了应对机器人导航问题中的奖赏值稀疏问题, Mirowski等引入两项辅助任务学习以丰富损失函数项[77]. 其中一项辅助任务是对低维深度图像进行无监督重构, 有助于提升避障和短期路径轨迹规划的能力; 另外一项辅助任务对局部轨迹进行自监督闭环分类. 基于LSTM网络在不同时间跨度上根据动态环境因素进行学习, 最终使机器人具备在复杂的三维环境中实现由原始传感器像素输入的端到端导航的能力.

目前, 深度强化学习已经在机器人的仿真控制、运动控制、室内室外导航、同步定位和建图等方向产生重要的影响. 通过端到端的决策与控制, 深度强化学习简化了机器人领域算法的设计流程, 降低了对数据进行预处理的需求.

4.3 自然语言处理(Natural language processing)

自然语言处理领域的研究一直被视为人工智能研究的热门领域, 不同于计算机视觉、图形图像这类直观模式识别问题, 自然语言是一种具有推理、语境、情感等人为性因素的更高层次的问题, 是当今尚待攻克的重要研究领域. 现阶段的深度强化学习算法已经在

[3] https://universe.openai.com.

对话问答系统、机器翻译、文本序列生成方面取得突破性研究进展. 在问答系统领域, Su等提出一种在线的深度强化学习框架, 根据高斯过程模型制定奖赏函数, 并且使用明确的用户评价作为奖赏信号反馈, 达到减少手动标注样本数据的开销和清除用户反馈的噪声信息的目标[78]. 在机器翻译领域, 有时要将两种语言互相进行翻译, 以此验证算法的翻译性能. 受此启发, He等提出双向学习机制建立双向互译模型, 采用策略梯度算法, 使用语言模型的近似程度作为奖赏信号[79]. 实验结果表明, 在使用较少数据集的前提下, 双向互译模型的翻译效果仍然能达到使用完全数据集进行单向翻译所训练模型的水平. 在文本序列生成领域, Yu等提出基于策略梯度算法的序列生成对抗式网络(sequence generative adversarial nets, SeqGAN)[80], 将对抗神经网络和强化学习有机结合在一起. 与之前基于知识库的文本序列生成算法相比, SeqGAN的文本序列生成质量得到明显提升.

现阶段的自然语言领域研究由于语言数据采集处理困难、人力资源成本投入大、算法评测标准存在一定的主观性等问题的挑战, 传统的算法已经表现出乏力的态势, 而深度强化学习领域正逐步往这个领域渗透, 相信在不远的未来, 深度强化学习能为自然语言处理的研究做出更大的贡献.

4.4 智能驾驶(Smart driving)

智能驾驶系统的决策模块需要先进的决策算法保证安全性、智能性、有效性. 目前传统算法的解决思路是以价格昂贵的激光雷达作为主要传感器, 依靠人工设计的算法从复杂环境中提取关键信息, 根据这些信息进行决策和判断. 该算法缺乏一定的泛化能力, 不具备应有的智能性和通用性. 深度强化学习的出现有效地改善了传统算法泛化性不足的问题, 能给智能驾驶领域带来新的思路.

深度强化学习由数据驱动, 不需要构造系统模型, 具有很强的自适应能力. 普林斯顿大学的Chen等使用深度学习算法, 根据摄像头采集的图像数据预测目标的距离, 同时输出操作指令[81]. 斯坦福大学的Zhu等使用暹罗网络结构, 同时输入当前视角图像和目标物体图像, 并且使用残差网络模型提取特征. 通过A3C算法进行训练, 成功控制小车在虚拟场景和现实场景中到达指定地点[82]. 国内的Zhao等使用深度强化学习算法和注意力机制, 实现了智能驾驶领域车辆的高精度分类[83]. Zhu基于TORCS的真实物理变量, 使用高斯过程强化学习算法PILCO离线训练控制器, 实现车道保持. 同时以图像为输入, 使用深度学习算法感知环境信息, 预测本车距离车到中央线距离、偏航角、道路曲率等. 最终将RL的控制策略和DL的特征预测结合, 实现基于图像的车道保持.

现阶段深度强化学习在智能驾驶领域的研究大多在基于仿真环境下进行, 在实车上的应用较为缺乏. 如何在真实道路环境和车辆上应用深度强化学习算法构建智能驾驶系统仍是一个开放性问题.

4.5 智能医疗(Intelligent healthcare)

医疗与人们的生活息息相关. 随着机器学习算法的不断进步和发展, 人们将先进的科技手段引入到医疗领域中, 以期达到人类专家水平, 缓解医疗资源紧张的问题. 谷歌的Gulshan等使用深度卷积神经网络对13万个视网膜照片进行训练, 最终表现的水平和单个眼科医生的水平相当[84]. 斯坦福大学的Esteva等同样采用了深度卷积神经网络, 对皮肤损伤照片进行训练, 判断水平达到了皮肤病学家的分类水平[85]. 埃默里大学的Nemati等应用深度强化学习对重症监护病人的肝素剂量进行建模[86], 使用判别式隐马尔可夫模型和Q网络对少量的相关数据进行学习, 从而探索到适合的最优策略. 麻省理工学院的Aniruddh等通过建立连续状态空间模型, 表示败血症病人不同时间节点上的生理状态, 使用深度Q网络算法, 找到适应败血症患者当前状态的最佳治疗方案[87].

目前的深度学习虽然已经在医疗的某些领域达到了专业医师的水平. 但深度学习通常需要大量的数据样本, 才能使模型的泛化性得到保证, 然而医疗数据具有私密性、隐私性和珍稀性的特点, 因此要获取足够的医疗数据通常需要大量的人力物力. 深度强化学习则能有效应对深度学习的这一需求, 在只需要少量初始样本的前提下, 通过强化学习的算法, 产生大量的经验模拟数据, 应用到模型学习, 以此达到较高的专业水准. AlphaGo Zero的成功, 证明了深度强化学习算法在没有大量先验知识的前提下, 仍能以端到端的形式完成围棋这项复杂任务. 相信AlphaGo Zero的成功会给予智能医疗领域更多新的启发.

5 深度强化学习资源进展(Progress of deep reinforcement learning resource)

近些年来, 随着谷歌DeepMind公司在《Nature》杂志上发表了Atari游戏AI[13], AlphaGo[4] 和AlphaGo Zero[5], 深度强化学习的研究在学术圈引起了广泛的关注. 2017年, IEEE Transactions on Neural Networks and Learning Systems (TNNLS)组织了关于深度强化学习与自适应动态规划的专刊, IEEE Transactions on Computational Intelligence and AI in Games (TCIAIG)组织了关于深度强化学习与游戏的专刊, Neural Networks组织了深度强化学习的专刊. 相关的书籍与网上在线学习资源, 也为广大的人工智能领域的研究者们学习了解深度强化学习提供了方便有效的学习路径. 书籍具体包括: 经典的强化学习与自适应动态规划[88–90]、深度学习[91]等. 网上学习资源包括:

Levine等的深度强化学习教程[4]、李飞飞的卷积神经网络课程[5]、Socher的自然语言处理领域的深度学习[6]、Silver的强化学习教程[7]. 更多资源可参见文献[2].

6 深度强化学习的发展展望(Development prospect of deep reinforcement learning)

随着硬件平台不断的更新换代, 计算资源及算力的大幅提升, 使原来需要大量训练时间的算法能够缩减到较短的时间周期. 如Alpha Zero使用5000块I代TPU和64块II代TPU完成自我对弈数据的产生和神经网络的训练, 用了不到2个小时就击败了日本将棋的最强程序Elmo, 用了4个小时打败了国际象棋最强程序Stockfish, 仅用了8个小时就超过了AlphaGo Lee的对弈水平[92]. 深度强化学习算法的贡献不言而喻, 但不能忽视算法背后所需要的强大算力资源. 要想更快提升算法的训练效率, 不能一味依靠硬件资源的支撑, 更需要对数据的利用训练效率展开更加深入细致的研究.

AlphaGo Zero和Alpha Zero算法的训练曲线图皆呈现稳定上升的走势, 说明深度强化学习是能够稳定提升的. Alpha系列算法的成功很大程度上归功于蒙特卡罗树搜索所做的贡献. 但是蒙特卡罗树搜索通常需要进行大量反复的完整的过程模拟, 这在简单环境下较易实现, 但是如果迁移到复杂的实时状态环境中, 便难以使用蒙特卡罗树搜索算法模拟相应状态. 当前, 深度强化学习的训练稳定性提升的理论保证和算法探索还需要投入更多的研究力量.

当下, 大部分深度强化学习算法是基于单个智能体行为控制任务的前提下所做的研究, 在需要不同属性的多智能体协同配合完成的决策性任务(如实时战略游戏、多人在线对抗游戏、多智能体信息交互等)的表现仍差强人意, 目前的相关工作已经展开[93–94], 并引起了社会各界的广泛关注. 可以预计, 基于多智能体协作的深度强化学习算法会成为将来的研究的重点之一.

7 结束语(Conclusion remarks)

本文介绍了AlphaGo出现以来的深度强化学习进展, 包括基于值函数的DQN及其扩展, 基于actorcritic的A3C及其扩展, 基于策略梯度的DDPG, TRPO, 以及其他类型的深度强化学习算法. 这些算法都在不同层面对深度强化学习进行了完善, 为AlphaGo Zero的出现奠定了坚实的基础. 继而, 通过对AlphaGo Zero技术原理的分析, 认识到深度强化学习在围棋AI领域取得的巨大成就. 在具体应用方面, AlphaGo的出现使深度强化学习在游戏、机器人、自然语言处理等领域的推广发展也非常迅速. 相信AlphaGo Zero的成功会进一步促进以深度强化学习为基础的其他人工智能领域的发展.

AlphaGo之父David Silver认为, 监督学习能产生当时性能最优的模型, 而强化学习却可以超越人类已有的知识得到更进一步的提升. 只使用监督学习算法确实可以达到令人惊叹的表现, 但是强化学习算法才是超越人类水平的关键. AlphaGo Zero的成功有力的证明了强化学习实现从无到有的强大学习能力, 但是这并不意味着通用人工智能领域问题得到了解决. AlphaGo Zero的出现只是证明在围棋这类特定应用环境的成功, 但要将这样的成功经验扩展到通用领域, 仍尚需时日, 因而通用人工智能问题的研究及解决仍然任重道远.

从文中统计的深度强化学习进展来看, 近两年的主要工作是由 Google DeepMind, Facebook, OpenAI等公司、以及一些国外名校也紧随其后. 这方面的研究仍然受到设备、数据、人才、资金等方面的制约, 国内好的成果仍然非常有限. 正如在综述[1]中提到的, 深度强化学习的先进基础理论算法、广泛的日常生活应用、以及潜在的军事领域扩展, 正在加大我国与国外的差距. 2017年初, 中国工程院院刊提出了“人工智能2.0”的发展规划, 并引起国家层面的关注和重视, 希望藉此可以大力发展以深度强化学习为基础的人工智能理论、算法和应用的研究.

致谢　感谢清华大学的周彤教授、华南理工大学的胡跃明教授提供的宝贵指导意见. 感谢李栋和卢毅在智能驾驶和智能医疗方面提供的建议和帮助. 感谢张启超、张旗、陈亚冉、李浩然和李楠楠提供的宝贵意见和对全文修改的帮助.

参考文献(References):

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

[2] LI Y. *Deep reinforcement learning: an overview* [EB/OL]. arXiv preprint, arXiv: 1701.07274, 2017.

[3] ARULKUMARAN K, DEISENROTH M P, BRUNDAGE M, et al. A brief survey of deep reinforcement learning [J]. *IEEE Signal Processing Magazine*, 2017, 34(6): 26 – 38.

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

[4] http://rll.berkeley.edu/deeprlcourse/.

[5] http://cs231n.stanford.edu/.

[6] http://cs224d.stanford.edu/.

[7] http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html.

[5] SILVER D, SCHRITTWIESERH J, SIMONYAN K, et al. Mastering the game of Go without human knowledge [J]. *Nature*, 2017, 550(7676): 354 – 359.

[6] LECUN Y, BENGIO Y, HINTON G E. Deep learning [J]. *Nature*, 2015, 521(7553): 436 – 444.

[7] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks [J]. *Communications of the ACM*, 2017, 60(6): 84 – 90.

[8] GRAVERS A, MOHAMED A, HINTON G E. Speech recognition with deep recurrent neural networks [C] //*Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vancouver: IEEE, 2013: 6645 – 6649.

[9] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks [C] //*Advances in Neural Information Processing Systems (NIPS)*. Montréal: MIT Press, 2014: 3104 – 3112.

[10] SUTTON R S, BARTO A G. *Reinforcement Learning: An Introduction* [M]. Massachusetts: MIT Press. 1998.

[11] LITTMAN M L. Reinforcement learning improves behaviour from evaluative feedback [J]. *Nature*, 2015, 521(7553): 445 – 451.

[12] ZHU Yuanheng, ZHAO Dongbin. Probably approximately correct reinforcement learning solving continuous-state control problem [J]. *Control Theory & Applications*, 2016, 33(12): 1603 – 1613.
(朱圆恒, 赵冬斌. 概率近似正确的强化学习算法解决连续状态空间控制问题 [J]. 控制理论与应用, 2016, 33(12): 1603 – 1613.)

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

[14] NAIR A, SRINIVASAN P, BLACKELLl S, et al. Massively parallel methods for deep reinforcement learning [C] //*Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Lille: [s.n.], 2015.

[15] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning [C] //*Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*. Phoenix: AAAI, 2016: 2094 – 2100.

[16] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. San Juan: ACM, IEEE, 2016.

[17] WANG Z, SCHAUL T, HESSEL M, et al. Dueling network architectures for deep reinforcement learning[C] //*Proceedings of the 33rd International Conference on Machine Learning (ICML)*. New York: [s.n.], 2016: 1995 – 2003.

[18] OSBAND I, BLUNDELL C, PRITZEL A, et al. Deep exploration via bootstrapped DQN [C] //*Advances in Neural Information Processing Systems (NIPS)*. Barcelona: MIT Press, 2016: 4026 – 4034.

[19] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning [C] //*Proceedings of the 33rd International Conference on Machine Learning (ICML)*. New York: [s.n.], 2016: 1928 – 1937.

[20] ZHAO D B, WANG H T, SHAO K, et al. Deep reinforcement learning with experience replay based on SARSA [C] //*Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI)-Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. Athens, Greece: IEEE, 2016.

[21] ANSCHEL O, BARAM N, SHIMKIN N. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning[C] //*Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney: [s.n.], 2017: 176 – 185.

[22] HE F S, LIU Y, SCHWING A G, et al. Learning to play in a day: Faster deep reinforcement learning by optimality tightening [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[23] BELLEMARE M G, DABNEY W, MUNOS R. A Distributional perspective on reinforcement learning [C] //*Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney: [s.n.], 2017: 449 – 458.

[24] FORTUNATO M, AZAR M G, PIOT B, et al. *Noisy networks for exploration* [EB/OL]. arXiv preprint arXiv: 1706.10295, 2017.

[25] HESSEL M, MODAYIL J, VAN HASSELT H, et al. *Rainbow: combining improvements in deep reinforcement learning* [EB/OL]. arXiv preprint arXiv:1710.02298, 2017.

[26] WU Y, TIAN Y. Training agent for first-person shooter game with actor-critic curriculum learning [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[27] BABAEIZADEH M, FROSIO I, TYREE S, et al. GA3C: GPU-based A3C for deep reinforcement learning [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[28] JADERBERG M, MNIH V, CZARNECKI W M, et al. Reinforcement learning with unsupervised auxiliary tasks [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[29] WANG J X, KURTH-NELSON Z, TIEUMALA D, et al. Learning to reinforcement learn [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[30] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. San Juan: ACM, IEEE, 2016.

[31] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms [C] //*Proceedings of the 31st International Conference on Machine Learning (ICML)*. Beijing: [s.n.], 2014: 387 – 395.

[32] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization [C] //*Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Lille: [s.n.], 2015: 1889 – 1897.

[33] GU S, LILLICRAP T, SUTSKEVER I, et al. Continuous deep Q-learning with model-based acceleration [C] //*Proceedings of the 33rd International Conference on Machine Learning (ICML)*. New York: [s.n.], 2016: 2829 – 2838.

[34] WU Y, MANSIMOV E, LIAO S, et al. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation [EB/OL]. arXiv preprint arXiv: 1708.05144, 2017.

[35] WANG Z, BAPST V, HEESS N, et al. Sample efficient actor-critic with experience replay [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[36] HEESS N, WAYNE G, SILVER D, et al. Learning continuous control policies by stochastic value gradients [C] //*Advances in Neural Information Processing Systems (NIPS)*. Montréal: [s.n.], 2015: 2944 – 2952.

[37] SCHULMAN J, MORITZ P, LEVINEe S, et al. High-dimensional continuous control using generalized advantage estimation [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. San Juan: ACM, IEEE, 2016.

[38] O'DONOGHUE B, MUNOS R, KAVUKCUOGLU K, et al. PGQ: combining policy gradient and Q-learning [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[39] NACHUM O, NOROUZI M, XU K, et al. Bridging the gap between value and policy based reinforcement learning [C] //*The Annual Conference on Neural Information Processing Systems (NIPS)*. Long Beach: [s.n.], 2017.

[40] GU S, LILLICRAP T, GHAHRAMANI Z S, et al. Q-prop: sample-efficient policy gradient with an off-policy critic [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[41] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. *Proximal policy optimization algorithms* [EB/OL]. arXiv preprint arXiv: 1707.06347, 2017.

[42] PARISOTTO E, BA J L, SALAKHUTDINOV R. actor-mimic: deep multitask and transfer reinforcement learning [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. San Juan: ACM, IEEE, 2016.

[43] RUSU A A, COLMENAREJO S G, GULCEHRE C, et al. *Policy distillation* [EB/OL]. arXiv preprint arXiv: 1511.06295, 2015.

[44] RUSU A A, RABINOWITZ N C, DESJARDINS G, et al. *Progressive neural networks* [EB/OL]. arXiv preprint arXiv: 1606.04671, 2016.

[45] FERNANDO C, BANARSE D, BLUNDELL C, et al. *Pathnet: evolution channels gradient descent in super neural networks* [EB/OL]. arXiv preprint arXiv:1701.08734, 2017.

[46] KULKARNI T D, NARASIMHAN K, SAEEDI A, et al. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation [C] //*Advances in Neural Information Processing Systems (NIPS)*. Barcelona: [s.n.], 2016: 3675 – 3683.

[47] KRISHNAMURTHY R, LAKSHMINARAYANAN A S, KUMAR P, et al. Hierarchical reinforcement learning using spatio-temporal abstractions and deep neural networks [C] //*Proceedings of the 33rd International Conference on Machine Learning (ICML)*. New York: [s.n.], 2016.

[48] KULKRRNI T D, SAEEDI A, GAUTAM S, et al. *Deep successor reinforcement learning* [EB/OL]. arXiv preprint arXiv: 1606.02396, 2016.

[49] VEZHNEVETS A S, OSINDERO S, SCHAUL T, et al. *Feudal networks for hierarchical reinforcement learning* [EB/OL]. arXiv preprint arXiv: 1703.01161, 2017.

[50] SCHAUL T, HORGAN D, GREGOR K, et al. Universal value function approximators [C] //*Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Lille: [s.n.], 2015: 1312 – 1320.

[51] OH J, CHOCKALINGAM V, SINGH S, et al. Control of memory, active perception, and action in minecraft [C] //*Proceedings of the 33rd International Conference on Machine Learning (ICML)*. New York: [s.n.], 2016: 2790 – 2799

[52] BLUNDELLl C, URIA B, PRITZEL A, et al. *Model-free episodic control* [EB/OL]. arXiv preprint arXiv: 1606.04460, 2016.

[53] PRITZEL A, URIA B, SRINIVASAN S, et al. *Neural episodic control* [EB/OL]. arXiv preprint arXiv: 1703.01988, 2017.

[54] FOERSTER J N, ASSAEL Y M, DE FREITAS N, et al. *Learning to communicate to solve riddles with deep distributed recurrent Q-networks* [EB/OL]. arXiv preprint arXiv: 1602.02672, 2016.

[55] LOWE R, WU Y, TAMAR A, et al. *Multi-agent actor-critic for mixed cooperative-competitive environments* [EB/OL]. arXiv preprint arXiv: 1706.02275, 2017.

[56] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas: [s.n.], 2016: 770 – 778.

[57] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C] //*Proceedings of the International Conference on Machine Learning (ICML)*. Lille: [s.n.], 2015: 448 – 456.

[58] ZHAO D B, ZHANG Z, DAI Y J. Self-teaching adaptive dynamic programming for Gomoku [J]. *Neurocomputing*, 2012, 78(1): 23 – 29.

[59] TANG Z T, ZHAO D B, SHAO K, et al. ADP with MCTS Algorithm for Gomoku [C] // *Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI)-Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. Athens, Greece: [s.n.], 2016.

[60] JUSTESEN N, BONTRAGER P, TOGELIUS J, et al. *Deep learning for video game playing* [EB/OL]. arXiv preprint arXiv: 1708.07902, 2017.

[61] BELLEMARE M G, NADDAF Y, VENESS J, et al. The arcade learning environment: an evaluation platform for general agents [J]. *Journal of Artificial Intelligence Research (JAIR)*, 2013, 47: 253 – 279.

[62] BROCKMAN G, CHEUNG V, PETTERSSON L, et al. *OpenAI gym* [EB/OL]. arXiv preprint arXiv: 1606.01540, 2016.

[63] BEATTIE C, LEIBO J Z, TEPLYASHIN D, et al. *Deepmind lab* [EB/OL]. arXiv preprint arXiv: 1612.03801, 2016.

[64] WYMANN B, ESPIEE, GUIONNEAU C, et al. *Torcs, the open racing car simulator* [EB/OL]. Software available at http://torcs. sourceforge. net, 2000.

[65] SHORT D. Teaching scientific concepts using a virtual world—minecraft [J]. *Teaching Science-the Journal of the Australian Science Teachers Association*, 2012, 58(3): 55.

[66] KEMPKA M, WYDMUCH M, RUNC G, et al. Vizdoom: a doom-based ai research platform for visual reinforcement learning [C] //*Procedings of the IEEE Conference on Computational Intelligence and Games (CIG)*. Greece: IEEE, 2016: 1 – 8.

[67] VINYALS O, EWALDS T, BARTUNOV S, et al. *Starcraft II: a new challenge for reinforcement learning* [EB/OL]. arXiv preprint arXiv: 1708.04782, 2017.

[68] PENG P, YUAN Q, WEN Y, et al. *Multiagent bidirectionally-coordinated nets for learning to play StarCraft combat games* [EB/OL]. arXiv preprint arXiv: 1703.10069, 2017.

[69] USUNIER N, SYNNAEVE G, LIN Z, et al. Episodic exploration for deep deterministic policies: an application to StarCraft micromanagement tasks [C] //*Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon: ACM, IEEE, 2017.

[70] SHAO K, ZHU Y H, ZHAO D B, Cooperative reinforcement learning for multiple units combat in StarCraft [C] //*Proceedings of IEEE Symposium Series on Computational Intelligence(SSCI)–Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Hawaii: [s.n.],

[71] MORAVČÍK M, SCHMID M, BURCH N, et al. Deepstack: expert-level artificial intelligence in heads-up no-limit poker [J]. *Science*, 2017, 356(6337): 508 – 513.

[72] KONG X, XIN B, LIU F, et al. *Revisiting the master-slave architecture in multi-agent deep reinforcement learning* [EB/OL]. arXiv preprint arXiv:1712.07305, 2017.

[73] ZHU Y H, ZHAO D B. Comprehesive comparison of online ADP algorithms for continuous-time optimal control [J]. *Artificial Intelligence Review*, 2017, DOI: 10.1007/s10462-017-9548-4.

[74] ZHU Y H, ZHAO D B, YANG X, et al. Policy iteration for Hinfinity optimal control of polynomial nonlinear systems via sum of squares programming [J]. *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2016.2643687.

[75] ZHU Y H, ZHAO D B, HE H B, et al. Event-triggered optimal control for nonlinear constrained-input systems with partially unknown dynamics via adaptive dynamic programming [J]. *IEEE Transactions on Industrial Electronics*, DOI: 10.1109/TIE.2016.2597763.

[76] LEVINE S, FINN C, DARRELL T, et al. End-to-end training of deep visuomotor policies [J]. *Journal of Machine Learning Research*, 2016, 17(39): 1 – 40.

[77] MIROWSKI P, PASCANU R, VIOLA F, et al. *Learning to navigate in complex environments* [EB/OL]. arXiv preprint arXiv: 1611.03673, 2016.

[78] SU P H, GASIC M, MRKSIC N, et al. On-line active reward learning for policy optimisation in spoken dialogue systems [C] //*Proceedings of the 51st Annual meeting of the Association for Computational Linguisticsmeeting (ACL)*. Berlin: [s.n.], 2016: 2431 – 2441.

[79] HE D, XIA Y, QIN T, et al. Dual learning for machine translation [C] //*Advances in Neural Information Processing Systems (NIPS)*. Barcelona: [s.n.], 2016: 820 – 828.

[80] YU L, ZHANG W, WANG J, et al. SeqGAN: sequence generative adversarial nets with policy gradient [C] //*Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*. San Francisco: AAAI, 2017: 2852 – 2858.

[81] CHEN C, SEFF A, KORNHAUSER A, et al. Deepdriving: learning affordance for direct perception in autonomous driving [C] //*Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Santiago: IEEE, 2015: 2722 – 2730.

[82] ZHU Y, MOTTAGHI R, KOLVE E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning [C] //*Proceedings of the International Conference on Robotics and Automation (ICRA)*, Singapore: IEEE, 2017: 3357 – 3364.

[83] ZHAO D B, CHEN Y R, LV L. Deep reinforcement learning with visual attention for vehicle classification [J]. *IEEE Transactions on Cognitive and Developmental Systems*, 2016, DOI: 10.1109/TCDS.2016.2614675.

[84] GULSHAN V, PENG L, CORAM M, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs [J]. *Jama*, 2016, 316(22): 2402 – 2410.

[85] ESTEVA A, KUPREL B, NOVOA R A, et al. Dermatologist-level classification of skin cancer with deep neural networks [J]. *Nature*, 2017, 542(7639): 115 – 118.

[86] NEMATI S, GHASSEMI M M, CLIFFORD G D. Optimal medication dosing from suboptimal clinical examples: a deep reinforcement learning approach [C] //*Proceedings of the 38th Annual International Conference of the Engineering in Medicine and Biology Society*. Florida: IEEE, 2016: 2978 – 2981.

[87] RAGHU A, KOMOROWSKI M, CELI L A, et al. *Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach* [EB/OL]. arXiv preprint arXiv:1705.08422, 2017.

[88] SUTTON R S, BARTO A G. *Reinforcement Learning: An Introduction (2nd Edition, in preparation)* [M]. Massachusetts: MIT Press. 2017

[89] POWELL W B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality* [M]. New Jersey: John Wiley & Sons, 2007.

[90] LEWIS F L, LIU D. *Reinforcement learning and approximate dynamic programming for feedback control* [M]. New Jersey: John Wiley & Sons, 2013.

[91] GOODFELLOW I, BENGIO Y, COURVILLE A. *Deep Learning* [M]. Massachusetts: MIT Press, 2016.

[92] SILVER D, HUBERT T, SCHRITTWIESER J, et al. *Mastering chess and shogi by self-play with a general reinforcement learning algorithm* [EB/OL]. arXiv preprint arXiv: 1712.01815, 2017.

[93] FOERSTER J, ASSAEL Y, DE FREITAS N, et al. Learning to communicate with deep multi-agent reinforcement learning [C] //*Advances in Neural Information Processing Systems (NIPS)*. Barcelona: [s.n.], 2016: 2137 – 2145.

[94] FOERSTER J, FARQUHAR G, AFOURAS T, et al. *Counterfactual multi-agent policy gradients* [EB/OL]. arXiv preprint arXiv: 1705.08926, 2017.

作者简介:

唐振韬 (1992–), 男, 博士研究生, 研究方向为强化学习、深度学习等, E-mail: tangzhentao2016@ia.ac.cn;

邵 坤 (1991–), 男, 博士研究生, 研究方向为强化学习、深度学习等, E-mail: shaokun2014@ia.ac.cn;

赵冬斌 (1972–), 男, 博士, 研究员, 研究方向为深度强化学习、自适应动态规划、智能交通、机器人、过程控制等, E-mail: dongbin.zhao@ia.ac.cn;

朱圆恒 (1989–), 男, 博士, 副研究员, 研究方向为深度强化学习、自适应动态规划等, E-mail: yuanheng.zhu@ia.ac.cn.