软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2022,33(1):129−149 [doi: 10.13328/j.cnki.jos.006306]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

神经结构搜索的研究进展综述*

李航宇[1], 王楠楠[1], 朱明瑞[1], 杨 曦[1], 高新波[2]

[1](综合业务网理论及关键技术国家重点实验室 (西安电子科技大学), 陕西 西安 710071)

[2](重庆邮电大学 计算机科学与技术学院, 重庆 400065)

通信作者: 王楠楠, E-mail: nnwang@xidian.edu.cn

摘 要: 近年来, 深度神经网络 (DNNs) 在许多人工智能任务中取得卓越表现, 例如计算机视觉 (CV)、自然语言处理 (NLP). 然而, 网络设计严重依赖专家知识, 这是一个耗时且易出错的工作. 于是, 作为自动化机器学习 (AutoML) 的重要子领域之一, 神经结构搜索 (NAS) 受到越来越多的关注, 旨在以自动化的方式设计表现优异的深度神经网络模型. 全面细致地回顾神经结构搜索的发展过程, 进行了系统总结. 首先, 给出了神经结构搜索的研究框架, 并分析每个研究内容的作用; 接着, 根据其发展阶段, 将现有工作划分为 4 个方面, 介绍各阶段发展的特点; 然后, 介绍现阶段验证结构搜索效果经常使用的数据库, 创新性地总结该领域的规范化评估标准, 保证实验对比的公平性, 促进该领域的长久发展; 最后, 对神经结构搜索研究面临的挑战进行了展望与分析.

关键词: 神经结构搜索; 自动化机器学习; 深度学习; 神经网络; 规范化评估

中图法分类号: TP311

中文引用格式: 李航宇, 王楠楠, 朱明瑞, 杨曦, 高新波. 神经结构搜索的研究进展综述. 软件学报, 2022, 33(1): 129–149. http://www.jos.org.cn/1000-9825/6306.htm

英文引用格式: Li HY, Wang NN, Zhu MR, Yang X, Gao XB. Recent Advances in Neural Architecture Search: A Survey. Ruan Jian Xue Bao/Journal of Software, 2022, 33(1): 129–149 (in Chinese). http://www.jos.org.cn/1000-9825/6306.htm

Recent Advances in Neural Architecture Search: A Survey

LI Hang-Yu[1], WANG Nan-Nan[1], ZHU Ming-Rui[1], YANG Xi[1], GAO Xin-Bo[2]

[1](State Key Laboratory of Integrated Services Networks (Xidian University), Xi'an 710071, China)

[2](College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: In recent years, deep neural networks (DNNs) have achieved outstanding performance on many AI tasks, such as computer vision (CV) and natural language processing (NLP). However, the network design relies heavily on the expert knowledge, which is time-consuming and error-prone. As a result, as one of the important sub-fields of automated machine learning (AutoML), the neural architecture search (NAS) has been paid more and more attention to, aiming to automatically design deep neural networks with superior performance. In this study, the development process of NAS is reviewed in detail and systematically summarized. Firstly, the overall research framework of NAS is given, and the function of each research content is analyzed. Next, according to the development stage in NAS field, the existing methods are divided into four aspects, and the characteristic of each stage is introduced in detail. Then, the datasets are introduced which are often used to verify the effect of NAS methods at this stage, and the normalized evaluation criteria in NAS field are innovatively summarized, so as to ensure the fairness of experimental comparison and promote the long-term development of this field. Finally, the challenges of NAS research are proposed and discussed.

Key words: neural architecture search (NAS); automated machine learning (AutoML); deep learning; neural network; normalized evaluation

深度学习 (deep learning)[1]已成为现阶段人工智能领域发展的重要推动力. 不同于传统手工设计特征, 深度神经网络以一种端到端的方式, 自动提取数据深层表征, 已在多个人工智能学科领域内取得卓越表现, 例如计算机视

* 基金项目: 国家重点研发计划 (2018AAA0103202); 国家自然科学基金 (61922066, 61876142, 62036007)
收稿时间: 2020-11-04; 修改时间: 2021-01-08; 采用时间: 2021-01-26; jos 在线出版时间: 2021-02-07

觉 (computer vision)、自然语言处理 (natural language processing)、语音识别 (speech recognition)、智能机器人 (intelligent robot) 等. 尽管深度学习在上述领域内取得成功, 研究人员还是面临着神经网络设计困难的挑战. 尤其是当前手工设计的神经网络结构越来越复杂, 不利于更多研究人员和从业人员使用深度学习. 于是, 研究者开始寻求一种自动化方式, 实现自主设计神经网络的目标, 即神经结构搜索 (neural architecture search, NAS).

自动化机器学习 (automated machine learning, AutoML) 是一种自动化的数据驱动方法, 并做出一系列决策. 仅需要使用者提供数据, 自动化机器学习技术能够自动获取最佳训练方案, 极大地降低机器学习技术的应用难度. 作为自动化机器学习的重要子领域之一, 神经结构搜索旨在以一种自动化的方式, 解决高难度的复杂神经网络设计问题. 具体上, 根据专家预先定义的搜索空间 (search space), 神经结构搜索算法在一个庞大的神经网络集合中评估结构性能并寻找到表现最佳的网络结构. 自动化结构搜索的结果往往是专家手工设计过程中未考虑的, 能够取得更加优异的性能表现, 尤其在一些硬件资源受限的应用场景中, NAS 往往能取得惊人的效果. 神经结构搜索在超参数选择的过程中扮演着关键角色, 而且具有重要的理论意义和应用价值. 面向一种特殊的神经网络结构超参数, 神经结构搜索联合优化理论和机器学习理论, 有效地解决神经网络模型的调参问题, 降低神经网络的使用成本与实现成本, 促使模型设计的智能化与神经网络应用的大众化.

近年来, 神经结构搜索成为人工智能领域中的热点方向之一. 根据 automl.org 列举的文献情况, NAS 文章发表时间与数量分布如图 1 所示. 自 2015 年起, 关于 NAS 的文章数量呈现指数增长的趋势. 回顾神经结构搜索技术的发展, 本文对神经结构搜索的已有重点研究工作进行全面综述. 在 NAS 算法发展的初期, NAS 算法通常采用采样重新训练的策略, 即从预先定义好的搜索空间中采样数量庞大的网络结构, 分别对每个采样结构重新训练并评估性能, 以获取表现最佳的神经网络. 这是广大研究者公认的真正意义上的一种神经结构搜索方法, 实验结果的优越性也表明其有效性. 然而, 对于 Cifar-10 数据集, 这类方法需要应用 800 个图形处理单元, 持续近一个月才能完成对最佳结构的搜索. 因此, 这种采样重新训练策略对计算资源的需求过大, 不利于 NAS 领域的发展与落地应用. 于是, 为了降低搜索阶段的资源消耗, 神经结构搜索领域内应用最广的一种加速方式: 权重共享策略 (weight-sharing strategy), 即尽可能地利用已经训练好的模型, 避免重新训练. 目前这种权重共享的搜索策略已经成为神经网络结构搜索的主流方向. 简而言之, 首先将预先设定的搜索空间表示为已经训练好的超级网络 (super-network), 然后在保留原始权重的同时, 直接对采样的子结构 (sub-architectures) 进行性能评估, 不需要重新进行模型训练.

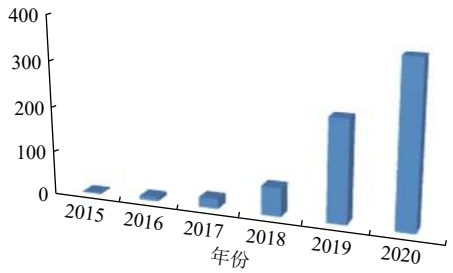

图 1 NAS 文章发表时间与数量分布

自 2018 年起, 已经有多篇神经结构搜索的研究综述[2−5]对该任务进行了介绍. Elsken 等人[2]给出了较全面的神经结构搜索领域的研究内容, 并从搜索空间、搜索策略和性能评估 3 个维度对 NAS 方法进行分类介绍. Xie 等人[4]深入分析基于权重共享的神经结构搜索方法, 并给出现阶段存在的优化缺陷与解决方案, 是目前 NAS 领域最全面的研究型综述. 然而, 回顾并反思现阶段的 NAS 发展, 最严重的问题就是实验评估中的不公平比较, 以及评估数据的局限性进一步限制神经结构搜索算法的通用性能, 这两个角度目前尚未在上述综述论文中得到分析, 我们将详细分析并给出相应的解决方案.

为了给读者提供清晰直观的 NAS 发展经历, 本文创新性地根据其发展阶段, 将现有工作划分为 4 个阶段, 即早期、快速发展期、应用期和反思期. 我们认为这种划分方式能够对今后研究 NAS 的工作人员提供很好的研究

基础, 更好地了解本领域的技术发展. 本文重点分析现阶段 NAS 算法在实验评估环节的缺陷, 建设性地提出规范化评估手段, 公正客观地对比不同方法, 推动该领域的良好发展与落地. 最后, 我们根据自身的研究基础, 概括 NAS 领域的现有问题与挑战, 提出若干点未来可能的研究方向, 帮助新的从业人员快速着手神经结构搜索研究.

本文第 1 节总结神经结构搜索的研究框架, 并分别介绍其中的 3 个重要模块, 即搜索空间 (search space)、搜索策略 (search strategy) 和性能评估 (performance estimation). 第 2 节从 4 个阶段 (即早期、快速发展期、应用期和反思期) 依次介绍现有的 NAS 算法, 重点分析代表性工作. 第 3 节介绍 NAS 领域常用的评估数据集, 分析现有工作在实验评估方面的不足, 创新地给出一系列规范化评估策略. 第 4 节讨论神经结构搜索任务中存在的挑战, 对未来值得关注的研究方向进行初步探讨. 最后, 总结全文, 并简单介绍我们今后的研究内容. 图 2 是本文的综述框架.

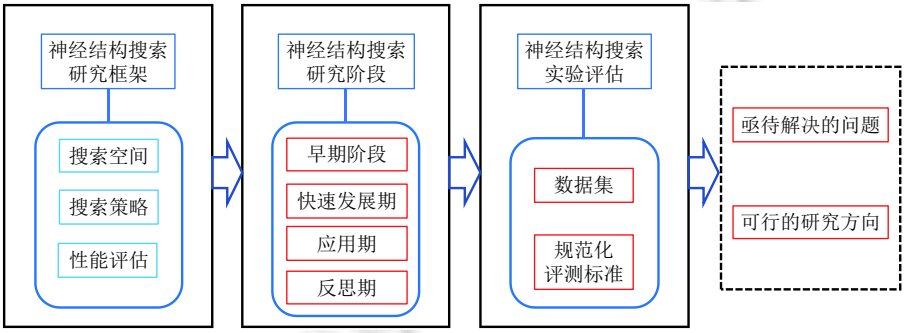

图 2 神经结构搜索研究综述框架

1 神经结构搜索的概述与研究框架

1.1 研究框架与算法流程

过去, 性能优越的神经网络结构往往需要专家手工设计, 存在一定的局限, 主要归结于客观原因和主观原因. 就客观原因而言, 专家知识总是存在一定的不足, 需要专家不断的尝试才能设计出性能优越的网络结构, 时间成本过高; 就主观原因而言, 专家也会因为个人因素影响网络设计, 人为地在模型中添加主观因素. 因此, 一种自动化、不受主观干扰的网络设计方式成为深度学习领域亟待解决的重要问题之一. 近年来, 神经结构搜索在学术界和工业界受到越来越多的关注, 其目的就是寻找到在某一数据集上取得最佳效果的模型, 例如识别任务中的准确率 (accuracy)[6,7]和图像分割中的均交并比 (MIoU)[8,9]等.

神经结构搜索研究框架如图 3 所示. 具体上, 神经结构搜索的早期方法主要是基于强化学习或进化算法, 指导神经网络的搜索. 大体的过程如图 3 中的早期 NAS 架构所示, 首先定义好搜索空间, 通过搜索策略采样得到一个网络结构, 进行性能评估并反馈给搜索策略, 重复上述过程, 直至得到一个表现最佳的神经网络结构. 近年来, 基于权重共享 (weight-sharing) 的结构搜索方法[4]受到广泛关注, 在这类方法中, 搜索策略和性能评估是高度相关的, 即往往是一体的. 于是, 现阶段的 NAS 架构主要是在搜索空间定义好的基础上, 进行结构搜索与优化, 继而得到性能最优的网络结构. 一个标准的神经结构搜索算法步骤如算法 1 所示.

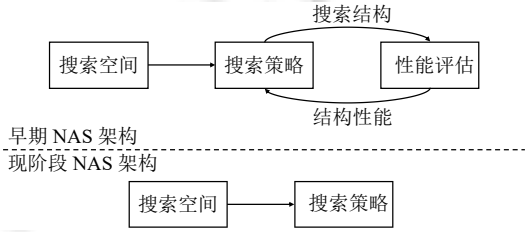

图 3 神经结构搜索研究框架

算法 1. 神经结构搜索的算法步骤

输入:搜索空间 $\mathcal{N}$, 训练数据 $\mathcal{D}_{\text{train}}$, 验证数据 $\mathcal{D}_{\text{valid}}$, 性能评估函数 $Eval_{\text{valid}}(\langle \mathbb{S}, \omega^*(\mathbb{S})\rangle)$;
初始化: 结构概率分布及网络性能状态等;
执行以下循环, 直至算法收敛, 循环结束
1 根据当前的结构概率分布采样网络结构 $\mathbb{S} \in \mathcal{N}$;
2 基于训练数据, 训练网络 $\mathbb{S}$, 并得到网络权重;
3 根据性能评估函数, 计算网络性能, 例如准确率等;
4 如果网络性能超过最佳状态, 则执行
5 根据上述训练得到的网络权重, 更新最佳模型;
6 结束
7 利用网络评估性能, 更新结构概率分布;
输出: 性能最优的子结构.

总而言之, 神经结构搜索的过程是在训练集上更新网络权重 $\omega(\mathbb{S})$, 在验证集上搜索性能最佳的结构 $\mathbb{S}^*$. 早期 NAS 算法的优化目标函数如式 (1) 所示, 首先在训练集上更新网络权重 $\omega(\mathbb{S})$, 并得到最优的网络权重 $\omega^*(\mathbb{S})$; 然后直接在验证集上计算性能评估函数, 其中 $\langle \mathbb{S}, \omega^*(\mathbb{S})\rangle$ 表示为网络结构 $\mathbb{S}$ 及其最优网络权重 $\omega^*(\mathbb{S})$; 最终得到最佳结构 $\mathbb{S}^*$. 然而, 基于权重共享的连续可微分 NAS 算法的优化目标是在训练集上更新网络权重 ω, 在验证集上更新结构系数 α. 优化目标函数如式 (2) 所示, 可微分的 NAS 方法以一种连续的方式, 计算验证损失, 避免传统的采样结构性能评估带来的高昂计算消耗.

$$\begin{cases} \max\limits_{\mathbb{S}} Eva\mathcal{L}_{\text{valid}}(\langle \mathbb{S}, \omega^*(\mathbb{S})\rangle) \\ \text{s.t. } \omega^*(\mathbb{S}) = \arg\min_{\omega} \mathcal{L}_{\text{train}}(\omega(\mathbb{S})) \end{cases} \tag{1}$$

$$\begin{cases} \min\limits_{\alpha} \mathcal{L}_{\text{vaild}}(\omega^*(\alpha), \alpha) \\ \text{s.t. } \omega^*(\alpha) = \arg\min_{\omega} \mathcal{L}_{\text{train}}(\omega, \alpha) \end{cases} \tag{2}$$

接下来, 我们简单介绍 3 个模块的定义与功能, 比较常用搜索空间的差异, 方便读者查阅.

1.2 搜索空间

搜索空间 (search space) 决定可搜索结构的范围, 包含一系列可用的计算操作, 如标准卷积 (standard convolution)、池化 (pooling) 和跳跃连接 (skip connection) 等. 由此可见, 搜索空间的规模极大限制搜索策略的表现. 尤其在理想状态下, 我们希望搜索空间越大越好, 只有这样才能评估更多结构的性能表现. 然而, 过大的搜索空间不利于神经结构搜索算法的收敛, 无法保证得到性能最优的网络结构. 因此, 需要精心考虑搜索空间的规模, 并选择最适合的结构集合.

目前, 搜索空间从搜索方式的角度主要分为两种, 即全局搜索空间和基于结构单元 (cell) 的搜索空间. 如图 4 所示, 全局搜索空间主要包含链式结构 (1) 及其衍生结构 (2,3), 决定网络结构的整体情况, 但灵活性有限. 为了提高每个操作的非线性表征能力, 基于结构单元的搜索空间是对每一个结构单元的计算操作进行搜索, 每个结构单元由若干个节点 (即特征图) 组成, 节点间的连线是需要搜索的计算操作. 有向无环图是一种具体的结构单元. 目前所有的 NAS 算法都在这两种搜索空间中进行结构搜索, 例如 MobileNet 搜索空间是由链式的全局空间和基于 Mobile 单元的空间构成; DARTS 搜索空间是一种由衍生的全局链式结构和基于结构单元的空间构成. 由此可见, 不同的搜索空间组合将产生多种不同的搜索空间, 且彼此之间相差较大, 对搜索策略的影响也很大. 根据 Xie 等人[4], 当前主要以 5 类搜索空间为代表, 分别为 EvoNet 搜索空间[10]、NAS-RL 搜索空间[11]、NASNet 搜索空间[12]、DARTS 搜索空间[13]和 MobileNet 搜索空间[14]. 表 1 总结所有搜索空间的描述与大小比较.

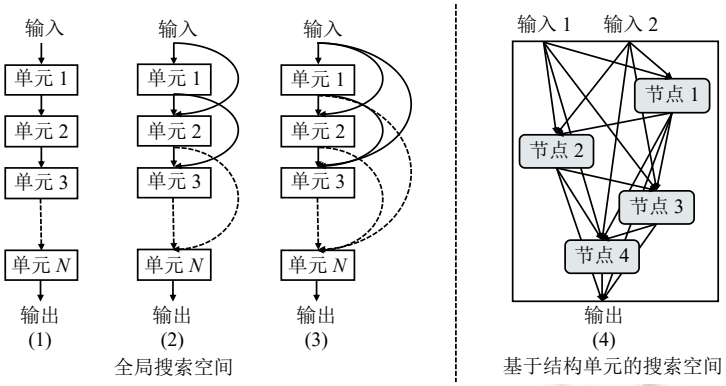

图 4 两种搜索空间的样例

表 1 目前所有搜索空间的对比情况

搜索空间	全局方式	局部方式	空间描述	空间大小
EvoNet	任意	任意	真正意义上的搜索空间, 候选操作任意选择, 例如特定位置插入卷积层、更改卷积核大小、跳跃连接等	无穷大
NAS-RL	紧密连接 (dense connect)	超参数	局部空间内共包含5个超参数: 卷积核的高、卷积核的宽、步长的高、步长的宽、卷积数量. 为简单起见, 所有超参数均从预先设定的集合中选择, 例如卷积数量范围是{24,36,48,64}	$C^N\times 2^{N(N-1)/2}$
NASNet	双链连接 (bi-chain style)	结构单元	每个结构单元内包含N个隐藏节点和两个输入节点, 且每个隐藏节点仅接受两个先前节点的输出作为输入. 候选操作分为两部分: (1) C_1个候选操作, 例如dil_conv_3×3, sep_conv_5×5等; (2) 汇总操作, 例如求和、拼接等	$(C_1^2\times C_2)^N\times\binom{2}{2}\times\binom{3}{2}\times\cdots\times\binom{N+1}{2}$
DARTS	双链连接 (bi-chain style)	结构单元	空间内共包含8个候选操作: zero, max_pool_3×3, avg_pool_3×3, skip_connect, sep_conv_3×3, sep_conv_5×5, dil_conv_3×3, dil_conv_5×5	1.1×10^{18}
MobileNet	链式结构	结构单元	基于MobileNet的局部单元结构, 候选操作包括: 可分离卷积的类型、跳跃连接的类型、通道数、卷积核大小、扩张比率和层数	$N(C^1+C^2+\cdots+C^L)$
其他	—	—	针对某些特定任务设计的搜索空间, 例如目标检测[15−22]、语义分割[23−26]、图像生成[27−30]等	—

1.3 搜索策略

搜索策略 (search strategy) 旨在寻找到一个神经网络, 并能够最大化性能指标, 例如在未知数据上的识别准确率. 现阶段的搜索策略主要包含 5 种方法, 分别为随机搜索、贝叶斯优化、进化算法、强化学习和基于梯度的方法. 从历史上看, 前 4 种搜索方式在一定程度上实现了神经结构搜索, 但无法满足大多数研究人员的要求. 于是, 基于梯度的方法是目前的主流方向, 革命性地将离散的结构搜索方式建模为一种连续松弛的搜索方式. 尤其是, 同以往在某个特定层固定操作相比, 这种基于梯度的方法能够计算一系列操作的凸组合, 灵活性更高. 关于具体分析, 读者可参考之前的综述文献[2,3]. 本文为了整个研究的完整性, 仅简单地介绍搜索策略的概念. 详细的方法介绍将在第 2 节中, 以 NAS 算法发展阶段的形式给出.

1.4 性能评估

搜索策略寻找到一个神经网络, 我们需要评估该网络的性能表现. 很容易想到的是, 在训练数据中重新训练该网络, 并在验证集上评估表现. 然而, 重新训练所有结构极大地提高计算需求, 例如首个基于强化学习的结构搜索方法[11]需要数千个 GPU 时完成整个结构搜索. 可想而知, 这种评估方式是不利于 NAS 领域的实际应用.

为了降低计算需求, 最好的解决办法是利用权重共享的概念. 因为大多数结构之间具有相似的操作, 所以多个相似结构间保持相同的网络权重, 可以有效避免重新训练带来的计算消耗. 值得注意的是, 在基于权重共享的优化方法中, 最重要的概念就是超级网络 (super-network). 具体地, 原始的搜索空间被建模为超级网络, 根据搜索策略采样得到的网络为子结构 (sub-architectures), 而且相似的子结构分享相同的网络权重.

通过权重共享采样子结构能够一定程度地加速搜索过程, 然而直接从超级网络中复制的权重不能表示其在验证集中表现. 因此, 研究人员开始转向可微分的神经结构搜索算法. 具体上, 直接以一种连续可微分的方式同时优化网络权重 (network weight) 和结构系数 (architecture parameter), 并在搜索的最后阶段直接得到最优的子结构, 不需要单独的结构采样操作, 进一步降低计算成本.

因此, 考虑到搜索策略和性能评估的任务相关性, Xie 等人[4]将这两个部分合二为一, 统一表述为“搜索策略”. 现有的 NAS 研究综述都是根据 NAS 研究框架构成来区分现有方法. 然而, 为了更好地以时间线回顾整个 NAS 领域的发展, 在下一节中, 我们创新性地以 4 个发展阶段划分现有 NAS 工作.

2 神经结构搜索的发展阶段

手工设计神经网络受多种因素影响, 例如不同层间的连接方式、卷积类型与大小、网络深度等. 这不仅需要网络设计专家具有丰富的机器学习知识, 还严重依赖专家在实践过程中总结的经验规律. 因此, 自 2015 年开始, 神经结构搜索得到广泛关注. 到目前为止, 神经结构搜索能够自动化地设计出性能优越的神经网络结构, 在图像识别等任务上取得显著效果. 这种成功是值得纪念的, 而且回顾 NAS 的整个发展阶段对于今后的发展有着重要意义. 图 5 是神经网络搜索研究的重要里程碑工作, 我们将从 4 个阶段分别介绍 NAS 算法的发展, 着重介绍每一阶段内具有代表性的成果. 这里我们需要说明的是, 考虑到现有 NAS 方法中存在的诸多问题, 且近两年不断有工作在反思神经结构搜索, 我们希望第 4 阶段中的反思能够促使 NAS 技术在未来的发展中实现更大的突破.

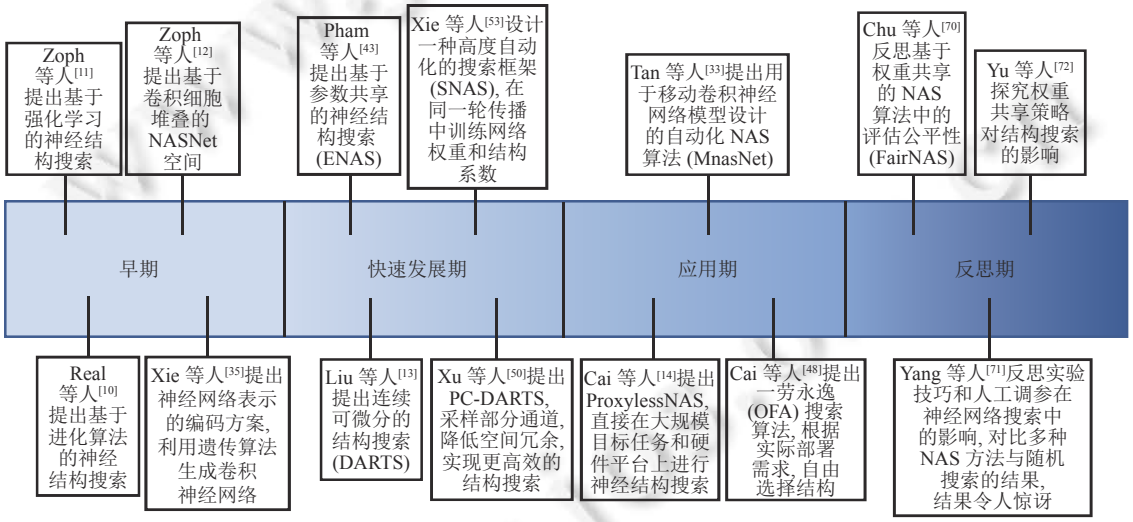

图 5 神经结构搜索研究的重要里程碑工作

2.1 早期阶段

为了实现神经结构搜索的目标, 最容易想到的方法就是列举所有可能的子结构, 分别训练并评估其性能. 然而, 这种搜索方法耗时严重, 对计算需求过大. 故而这种单独采样评估的方式仅仅是种设想, 不满足实际情况. 为了提高结构搜索的效率, 在研究早期, 研究人员开始考虑利用神经网络结构间的相互关系, 即某个子结构的性能表现更优, 其相邻结构被搜索、采样到的概率也应该更大. 具体上, 这类方法首先构建一种概率密度函数, 赋予搜索空间内的所有子结构被采样的可能性. 然后, 采样到的子结构性能指标帮助算法得到奖励 (reward), 并更新概率密度

函数. 本小节重点回顾早期阶段中有代表性的基于强化学习和基于进化/遗传算法的神经结构搜索方法.

近年来, 被人们熟知的第一个神经结构搜索的工作, Zoph 等人[11]利用强化学习 (reinforcement learning) 的思想, 通过一个控制器 (controller) 在搜索空间中以一定概率分布采样到子结构 (child network), 继而在训练数据集上进行训练, 并在验证集上测试得到性能表现. 为了控制器继续优化得到更优的网络结构, 上一步得到的评估准确率需反馈回控制器, 如此反复上述步骤直到得到性能最优的网络结构. 具体地, 如图 6 所示, 预测网络仅包含卷积层, 利用循环神经网络 (recurrent neural network, RNN) 预测卷积层内的一系列超参数, 例如卷积数量、卷积核的高和宽、步长等. 此外, 循环神经网络中的每一个预测输出继续作为下一层的输入.

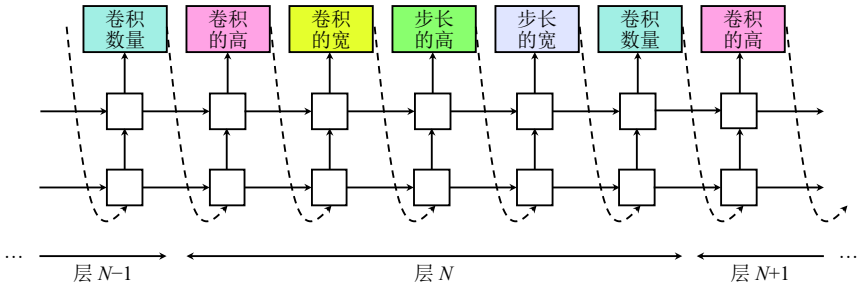

图 6　控制器循环神经网络采样简单的卷积神经网络的过程

尽管这种基于强化学习的搜索方式在 Cifar-10 数据集上取得良好效果, 但这种搜索方式不适用于 ImageNet 等大规模数据集. 在沿用现有思路的基础上, 如图 7 所示, Zoph 等人[12]创新性地设计合适的搜索空间, 即 NASNet 搜索空间, 并采用卷积单元 (convolution cell) 堆叠的思想. 其中, 正常单元 (normal cell) 不改变特征图的大小, 并计算输入特征; 缩小单元 (reduction cell) 对输入特征图进行下采样操作, 降低特征图的空间分辨率, 并计算表征. 因此, 控制器只需要分别预测两种结构单元, 堆叠后得到最终的子结构. 后面, 一系列基于强化学习[31−34]的结构搜索方法被提出. 其中, 有很多代表性的方法. 例如, 为了提高这类算法的搜索效率, Liu 等人[34]提出一种新的优化策略, 从增加复杂度的角度搜索网络结构, 缩小搜索空间并利用预测函数来预测网络精度.

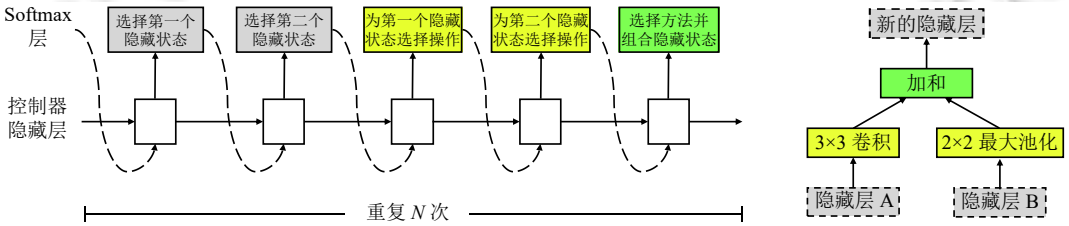

图 7　用于递归构建卷积结构单元块的控制器模型架构

上述方法利用强化学习理论搜索神经网络结构, 另外有大量的研究工作[10,35−41]考虑进化/遗传算法在 NAS 中的作用. Real 等人[10]首次提出类似生物进化的方法, 自动化地设计深度神经网络, 即在随机生成的模型中, 对性能表现优越的模型进行变异, 逐步淘汰性能表现差的网络结构, 最后得到表现最佳的模型. 实验结果也验证这种进化思想在神经结构搜索中的有效性, 并在图像分类任务上实现不低于人工精心设计结构的性能表现. Xie 等人[35]利用传统的遗传算法, 自动化地生成卷积神经网络. 具体上, 他们首次提出一种神经网络结构表示的编码方案, 初始化种群, 对种群进行选择选择、变异和交叉, 从而抛弃评估性能较差的网络结构, 并产生新的卷积神经网络. Liu 等人[36]提出一种分层表示的神经结构搜索空间, 通过进化算法进行空间搜索. Real 等人[37]继续探索 NASNet 搜索空间, 创新性地对遗传算法中的锦标赛选择法进行改进, 将其变为基于年龄的选择法 (aging evolution), 使遗传算法更加关注年轻个体.

除了将强化学习和进化算法分别作为神经结构算法的优化策略之外, Chen 等人[42]创新性地将两种理论进行

合并, 优化神经结构搜索算法. 具体上, 他们提出增强进化神经结构搜索算法 (RE-NAS), 这是一种针对神经结构搜索的增强突变的进化方法. 该方法将增强变异整合到神经结构搜索的进化算法中, 介绍一种变异控制器学习轻微改变的效果并做出变异行为, 引导模型种群有效进化. 此外, 子结构在进化过程中从父辈模型中继承参数, 大大降低计算资源的消耗.

2.2 快速发展期

尽管早期的一系列神经结构搜索方法取得了较好的效果, 但对计算资源的消耗是巨大的, 影响众多从业人员的研究工作, 严重制约领域的良好发展. 于是, 加速搜索成为解决 NAS 问题的关键研究方向之一. 如何加速结构搜索是一个需要深思熟虑的问题. 反思早期 NAS 方法, 最严重的问题就是需要二次重新训练采样的子结构. 这种重新训练带来的不仅是搜索时间的加长, 更是计算资源需求的增长. 因此, 权重共享 (weight-sharing) 策略渐渐地在神经结构搜索领域内得到肯定.

何谓权重共享? 简单来说, 就是针对一个已经训练好的原始大规模神经网络, 子结构直接继承原始网络的模型权重, 继而显著降低模型的运算量. 最具代表性的权重共享方法是基于 One-Shot 的神经结构搜索, 即构建搜索空间成超级网络 (super-network), 采样得到所有可能的子结构 (sub-architecture), 并且不同结构之间共享相关联的模块权重 (module weights). 从图形学的角度考虑, 超级网络可视作有向无环图. 换句话说, 原始的搜索空间表示为单个有向无环图 (directed acyclic graph, DAG), 即搜索空间中所有可能子结构的叠加结果. 如图 8 所示, 节点 (node) 表示局部计算, 边 (edge) 表示信息流. 图中的 6 个节点包含多种单向的有向无环图, 而虚线标出的 DAG 是最终选择的子图.

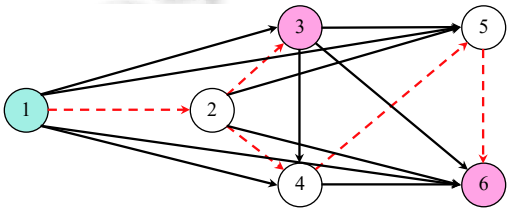

图 8 由有向无环图表示的搜索空间

众所周知, Pham 等人[43]首次提出基于参数共享的神经结构搜索 (ENAS) 方法, 是一种快速且低耗的自动模型设计方法. ENAS 中, 控制器通过在大型计算图中搜索最佳子图, 同时子模型间的权重共享大大降低计算开销. 这项工作的重要贡献在于, 通过强制子结构间的参数共享来避免重新训练每个子模型带来的高昂计算, 从而提升神经结构搜索的效率. 具体上, 图 9 给出控制器和有向无环图设计递归神经网络单元的过程. 以 4 个节点为例, ENAS 的控制器是一个循环神经网络, 用于决定边 (edge) 是否需要激活和有向无环图中每个节点需要执行的计算类型. 很容易看出, 对于每一组节点 (节点 i 和节点 $j, i<j$) 都会有对应的权重矩阵 $W_{j,i}^{(h)}$, 因此在整个的循环单元中, 这一组权重是被共享的.

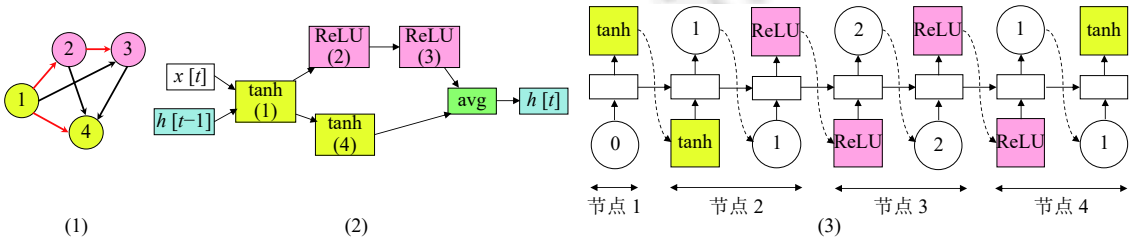

图 9 一个包含 4 节点的循环单元样例

另外, 基于这种权重共享策略的神经结构算法在不断更新, 很多研究者们提出一些新的角度[44−49]. Brock 等人[45]提出训练辅助网络 (HyperNet), 动态生成不同结构的模型权重. 尽管这种自动生成的权重在性能上不如常规训练学习到的权重, 但在训练初期, 使用生成模型权重的结构间相对性能在一定程度上映射出最优状态时的相对性能.

因此这种搜索方法可以仅通过一轮训练对大量的子结构进行重要性排序. Bender 等人[46]深入分析权重共享在神经结构搜索中的作用, 并发现未实现好的搜索结果, 超网络和强化学习控制器不是必须的. 具体上, 他们预先训练好包含所有可能候选操作的大规模 One-Shot 模型, 不断剔除一些操作, 测量其对模型测试的影响. 实验结果表明 NAS 算法专注于搜索产生良好性能的操作上, 剔除不重要操作对模型性能影响很小, 而剔除重要操作对模型测试性能的影响很大. Guo 等人[47]构建一个简化版的超级网络——单路径超级网络 (single path supernet), 按照均匀的路径采样策略进行训练, 使所有子结构及其权重获得平等且充分的训练.

虽然上述方法以一种有效方式将搜索空间建模为超级网络, 并在一定程度上提升算法效率, 可仍然需要一个独立组块对超级网络进行采样, 以得到性能优越的子结构; 而且优化方法上依旧采用进化算法或强化学习的思想. 为了消除搜索过程中的采样步骤, 研究人员开始思考是否能够以一种连续可微分的搜索方式, 反向传播结构系数, 同时利用权重共享策略, 实现权重共享的可微分神经结构搜索方法. 尤其是, Liu 等人[13]以全新的角度解决神经结构搜索问题, 率先提出可微分结构搜索 (differentiable architecture search, DARTS). 不同于过去方法在候选操作的离散集合中进行搜索, DARTS 将整个搜索空间松弛到连续空间 (即 DARTS 搜索空间), 从而通过梯度下降的方法, 根据验证集上的性能表现, 对结构系数进行优化. 如算法 2 所示, DARTS 提出近似迭代优化的策略, 其中网络权重 ω 和结构系数 α 分别在训练集和验证集中交替优化, 且优化更新后的 ω 作为参数, 进一步优化 α.

算法 2. DARTS —可微分的神经结构搜索算法

创建一个混合候选操作 $\bar{o}^{(i,j)}$, 其中每条边被 $\alpha^{(i,j)}$ 参数化;

当算法不收敛, 执行

1. 计算梯度 $\nabla_{\alpha}\mathcal{L}_{\text{valid}}(\omega-\xi\nabla_{\omega}\mathcal{L}_{\text{train}}(\omega,\alpha),\alpha)$, 更新结构系数 α;

(当 $\xi=0$ 时, 计算一次近似即可)

2. 计算梯度 $\nabla_{\omega}\mathcal{L}_{\text{train}}(\omega,\alpha)$, 更新网络权重 ω;

基于学习到的结构参数 α 得到最终的结构.

此外, DARTS 的搜索空间继续跟随前人的搜索空间策略, 即搜索一个计算单元 (computation cell) 作为最终结构的构建模块 (building block). 每个结构单元都是一个有向无环图, 即由 N 个节点组成的有序序列. 其中, 每个节点 $x^{(i)}$ 是一个潜在表征 (latent representation), 例如卷积中的特征图; 每个有向边 (directed edge) 表示某些操作 $o^{(i,j)}$, 用于转换 $x^{(i)}$. 不同过往的采样策略, DARTS 连续化搜索空间, 将特定操作的种类选择松弛为所有候选操作的 Softmax 函数 (如式 (3) 所示), 于是结构参数就统一在计算图中, 且计算验证损失后就可以反向传播并计算结构系数, 即搜索空间松弛后, 神经结构搜索的目的就是学习一组连续变量 $\alpha=\alpha^{(i,j)}$. 搜索的过程就是优化结构系数的过程, 最后保留结构系数最大的操作就是搜索到的神经结构.

$$\bar{o}^{(i,j)}(x)=\sum_{o\in\mathcal{O}}\frac{\exp\left(\alpha_o^{(i,j)}\right)}{\sum_{o'\in\mathcal{O}}\exp\left(\alpha_{o'}^{(i,j)}\right)}o(x) \tag{3}$$

其中, $\alpha^{(i,j)}$ 表示一对节点 (i,j) 的操作混合权值 (operation mixing weights). 图 10 是 DARTS 算法的概述.

可微分的神经结构搜索不需要单独评估搜索过程中产生的大量网络结构, 效率更高, 可谓是近年来神经结构搜索领域的卓越贡献之一. 后续有非常多的工作[50−59]都是基于可微分的思路, 不断完善并改进 NAS 算法. 尤其为了解决 DARTS 方法中存在的低效、不稳定等问题, Xu 等人[50]提出局部通道连接 (partial channel connections, PC) 的 DARTS, 通过采样一小部分的超级网络来降低网络空间的冗余性, 在不影响性能的情况下执行更加有效的结构搜索过程. PC-DARTS 解决了 DARTS 中内存开销大的问题, 能够以更大的批量进行训练, 且速度更快、训练稳定性更高. Zela 等人[51]研究 DARTS 算法的稳定性问题, 认为可微分的结构搜索算法在一些新的数据集上没有表现出稳健的结果, 原因在于 DARTS 对验证集过度拟合. 同时为了避免搜索过程中陷入一种锐利最小化 (sharp minimum) 的情况, 提出早停 (early stopping) 和正则化 (regularization) 的思想. Dong 等人[55]提出全新的 GDAS, 进一步提升 DARTS 的有效性. 具体地, DARTS 优化所有候选操作的权重, 训练时间过长, 于是 GDAS 只需要训练采

样后的候选操作, 缩短训练时间; DARTS 同时优化不同的候选操作, 使它们产生对抗性, 不同的候选操作可能产生相反的值, 求和后可能出现特征消失现象, 从而导致两个相连节点间信息流的破坏, 影响优化过程, 所以 GDAS 仅优化有向无环图中的一部分. 考虑到 DARTS 搜索和评估过程中存在的深度差异 (depth gap), Chen 等人[56]提出一种渐进式搜索 (P-DARTS) 的方法, 使搜索结构的深度在训练过程中逐渐增长. 为了解决随之而来的计算量大和稳定性差的问题, P-DARTS 提出使用搜索空间近似和正则化的策略. 此外, Xie 等人[53]以全新角度提出一种高效且高度自动化的搜索框架, 即随机神经结构搜索 (SNAS), 该框架在保持 NAS 工作流程完整并可微分的前提下, 在同一轮反向传播中训练神经网络权重和结构系数.

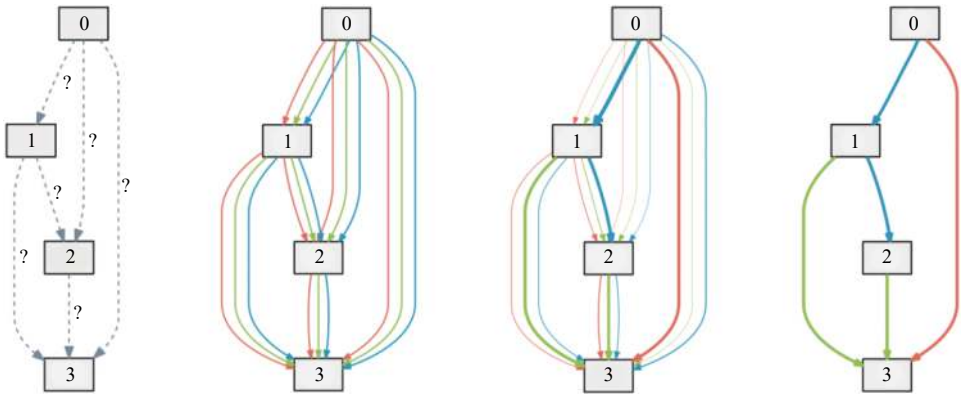

图 10 DARTS 的概述

2.3 应用期

在前两个阶段中, 现有的神经结构搜索算法基本都在重新设计新的搜索空间, 初步实现自动化网络设计的目标. 此外, 现有搜索方法得到的网络结构虽然取得较好的性能表现, 但是不适合模型的实际部署. 以 DARTS 为例, 该方法能够在很小的模型参数量的前提下实现很高的模型精度, 可是前向传播是很慢的, 远远不能达到实际部署的要求. 借鉴前人手工设计的轻量模型[60−62], 研究人员开始考虑引入相关的先验知识, 提升神经结构搜索算法的实际部署性能. 由于移动设备 (如手机等) 上可用的计算资源有限, 设计一个资源受限的移动模型是具有挑战性的. 基于 MobileNet 搜索空间, 一系列研究[14,33,48,63−68]重点解决神经结构搜索在手机等硬件设备上的部署限制.

Tan 等人[33]提出一种用于移动卷积神经网络模型设计的自动化神经结构搜索算法 (MnasNet). 如图 11 所示, MnasNet 主要的贡献在于延迟感知的多目标奖励机制和全新的搜索空间. 具体地, MnasNet 将网络设计表示为多目标优化问题, 并考虑卷积神经网络模型的准确率和推理延迟, 直接在实际移动设备上运行模型来计算实际延迟; 为了增强结构的多样性, MnasNet 在一种分解层次搜索空间进行结构搜索, 允许层在架构上不同, 而且保持灵活性和搜索空间大小之间的适当平衡.

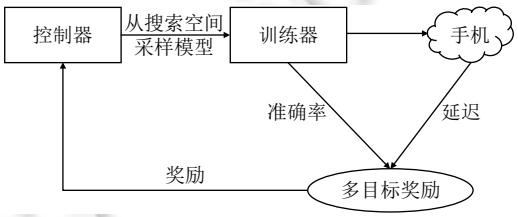

图 11 面向移动平台的神经结构搜索概述

此外, 神经结构搜索算法能够自动设计网络结构, 由于早期方法计算量大, 难以在大型任务上执行搜索过程. 于是可微分的神经结构搜索解决了搜索耗时的问题, 但是仍存在 GPU 内存消耗大的问题. 如图 12 所示, 过去的方

法只能在代理任务 (proxy task) 上搜索, 例如在较小的数据集上训练、使用较小的块 (block)、减小训练次数等. 这种基于代理的方法导致小数据集上搜索的网络模型无法在目标任务上达到最优的性能表现. 针对这种缺陷, Cai 等人[14]提出一种不使用代理任务的方法 (ProxylessNAS), 直接在大规模的目标任务和硬件平台上进行神经结构搜索, 解决神经结构算法中 GPU 内存占用高和计算耗时长的问题. 具体地, ProxylessNAS 为神经结构搜索提供了一种新的路径剪枝 (path-level pruning) 方法, 创新性地展示 NAS 与模型压缩之间的关系, 通过二值量化的策略降低 GPU 内存消耗; 此外, ProxylessNAS 是一种基于延迟正则化损失 (latency regularization loss) 的梯度方法, 约束硬件指标, 使其能够在特定的硬件上处理特定的神经网络.

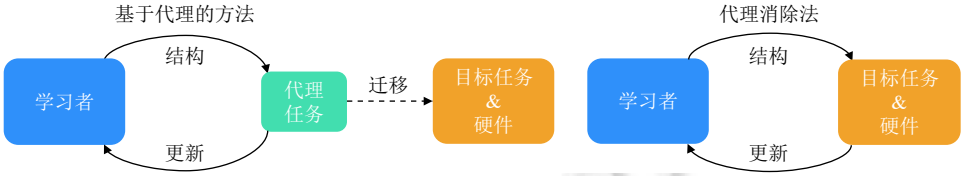

图 12 优化方式的差异

下面介绍的一些后续方法, 在一定程度上推动神经结构算法在实际应用中的部署. Wu 等人[63]利用可微分神经结构搜索设计硬件感知的卷积结构 (FBnet). Dai 等人[64]提出一种给定资源约束下的神经网络设计方法, 专注于开发硬件特性和计算资源, 适应目标延迟或资源限制. 同 MnasNet[33]类似, ChamNet[64]也是在优化框架中设计了平台感知的神经结构搜索, 并考虑准确率和延迟等资源限制. Stamoulis 等人[65]创新性地提出超核 (super kernel) 概念, 统一表示卷积核为 3×3 和 5×5 的两种卷积, 使网络成为单路径结构, 进一步提升搜索速度. 深度学习模型在实际部署需要适应不同的硬件平台, 且重新训练的时间成本多大. 为了解决 NAS 模型的实际部署问题, Cai 等人[48]提出一种一劳永逸 (once for all, OFA) 的结构搜索算法, 同时处理多种部署场景. 具体上, 该方法将模型训练和结构搜索过程进行分离, 训练一个支持多种不同结构设置 (深度、宽度、卷积核大小和空间分辨率等) 的 OFA 网络. 根据实际部署场景, 选择 OFA 中合适的子结构, 不需要额外的训练过程.

2.4 反思期

在如今的人工智能领域, 自动化神经结构搜索受到越来越多的关注. 相信在不久的将来, 手工设计神经网络会被网络自动设计方法替代. 由于神经结构搜索技术的不断革新, 在应用时期, 目前的诸多成果在手机等移动设备中成功落地, 意义深远. 毫无疑问, 神经结构搜索算法在图像识别和周边领域的开发进程和研究突破中发挥着重要作用. 但是, 这并不代表神经结构搜索算法是完美的. 研究人员在搜索过程中经常会遇到一个问题: 算法在搜索过程中的不稳定性. 尤其在 DARTS[13]算法搜索网络结构的过程中, 搜索的子结构大部分操作会被跳跃连接占据, 这样虽然有助于算法的收敛, 但由于在训练集上的过度拟合, 搜索的网络结构在验证集上的评估表现往往会很差, 无法达到预期目标. 除此之外, 还有很多搜索结构过程中遇到的问题需要研究人员的关注. 因此, 近两年出现越来越多的工作[69−72]专注于研究神经结构搜索算法中存在的通用问题, 并反思过去方法中经常忽略的弊端, 例如 One-Shot 方法中的采样不公平现象、不同方法的不公平对比等.

One-Shot 的思想在快速发展期被提出, 主张网络权重是可以共享的, 即从头到尾训练一个超级网络, 每个网络模型都是超级网络的采样子结构. 这一思想以大幅提升神经结构搜索效率的优势, 不需要重新训练每个采样子结构即可得知其表征特性, 成为目前神经结构搜索领域的主流方向之一. 然而, 这一思想有一个假设前提, 即假设权重共享是有效的, 模型表征能力能够通过这种共享方式快速地得到验证. 所以, 这种假设是否成立直接影响后面一系列方法的有效性. 往往一些表现差的模型并非是因为自身表征能力不够, 只是训练不得当或不充分导致最终的结果比较差. 在训练过程中, 所有采样模型需要给予相同的机会和条件来提升其表征能力. 考虑到这一通用缺陷, Chu 等人[70]反思基于权重共享的神经结构搜索中评估公平性的问题, 提出 FairNAS 方法, 公平地采样和训练所有采样子结构, 发挥各组成模块的潜能. 为了保证评估的公平性, FairNAS 方法提出严格公平 (strict fairness) 的要求, 即超级网络在每次迭代更新的过程中, 每层可选择的候选操作的参数都要充分训练.

此外, 近年来 NAS 算法发展迅速, 可是关于哪种算法最好、哪种搜索方法最好, 其实一直没有得到研究人员的肯定与共识. 不同的神经结构搜索方法之间的对比是极不公平的, 也就是说, 不同的方法很难在相同的参数设置在进行公平比较. 通常, 一些训练技巧 (如 Cutout、DropPath、AutoAugment 等)、人工调参和精心设计过拟合等都可以使算法在某些数据集上取得优异表现. 于是 Yang 等人[71]在 5 个数据集上对比 8 个具有开源代码的神经结构搜索算法 (DARTS[13]、StacNAS[69]、P-DARTS[56]、MANAS[73]、CNAS[74]、NSGANET[75]、ENAS[43]、NAO[52]) 的性能表现. 结果发现, 相比随机采样的结果, 上述方法在性能上提升很小, 甚至在某些情况下要差于随机搜索的结果; 方法之间的性能差异不大, 真正起作用的是搜索空间, 搜索策略和优化起到的作用很小; 在 Cifar-10 数据集上的性能表现相近, 但在其他数据集上的结果差异很大, 说明这些方法在 Cifar-10 数据集上存在过拟合, 泛化能力很差. 通过这一系列的对比, 我们能够发现现阶段神经结构搜索算法的显著问题, 往往优异的结果不是依靠搜索算法得到的, 而是借助各种训练技巧, 这种不公平的对比值得反思. 类似地, Yu 等人[72]也考虑搜索算法与随机搜索在性能的差异, 结果也表明随机搜索往往会取得同搜索算法相同的效果, 而且权重共享策略使一些子结构重要性下降到无法反映其真实性能的程度, 降低搜索过程的有效性.

总而言之, 神经结构搜索理论从诞生到如今一系列的显著成果, 我们能够肯定 NAS 算法的关键作用, 但也需要回顾前人的工作并做出反思, 不断推动该领域的良好发展. 表 2 总结了常用的神经结构搜索算法在 Cifar-10 基准数据集上分类错误率、参数量和搜索耗时等结果 (均出自论文的原始结果), 以帮助读者随时查阅实验结果并作对比. 此外, 我们给出各种搜索方法应用的优化方法与策略, 促使同类方法之间的公平比较.

表 2 现有方法在 Cifar-10 数据集上的结果

方法	测试错误率(%)	参数量(百万)	搜索耗时(GPU时)	搜索方法描述
ResNet[6]	4.62	10.2	—	手工设计
DenseNet-BC[7]	3.46	25.6	—	手工设计
NAS-RL[11]	3.65	37.4	24 000	基于强化学习的方法
NASNet[12]	2.65	3.3	2 000	基于强化学习的方法
AmoebaNet-A[37]	3.34±0.06	3.2	3 150	基于进化算法的方法
AmoebaNet-B[37]	2.55±0.05	2.8	3 150	基于进化算法的方法
PNAS[34]	3.41±0.09	3.2	225	基于序列模型优化的方法
NAONet[53]	3.53	3.1	0.4	基于连续优化的方法
ENAS[43]	2.89	4.6	0.5	权重共享+基于强化学习的方法
SNAS[53]	2.85±0.02	2.8	1.5	基于强化学习的可微分方法
BayesNAS[76]	2.81±0.04	3.4	0.2	权值共享+基于贝叶斯优化的方法
ProxylessNAS[14]	2.08	5.7	4.0	基于梯度和路径剪枝的方法
NASP[77]	2.83±0.09	3.3	0.1	基于近端迭代的可微分方法
DARTS(1st-order)[13]	3.00±0.14	3.3	0.4	基于权重共享的可微分方法
DARTS(2nd-order)[13]	2.76±0.09	3.3	1.0	基于权重共享的可微分方法
GDAS[55]	2.93	3.4	0.3	基于权重共享的可微分方法
P-DARTS[56]	2.50	3.4	0.3	基于权重共享的可微分方法
PC-DARTS[50]	2.57±0.07	3.6	0.1	基于权重共享的可微分方法
R-DARTS[51]	2.95±0.21	—	1.6	基于权重共享的可微分方法

3 评测数据库与规范化标准

为了评估神经结构搜索的相关算法, 我们往往需要在一些公开数据集上进行实验, 并通过公平统一的规范化评测标准来评估所提出搜索算法的性能. 在本节中, 我们对神经结构搜索的相关常用数据集和规范评测标准进行总结.

3.1 评测数据集

近年来神经结构搜索问题在学术界和工业界中的关注逐渐提高, 对相关算法的评估也是亟需考虑的问题, 其中最重要的就是算法训练与测试过程中使用的数据集. 我们回顾并总结现有 NAS 算法中常用的数据集. 针对目前使用数据集的单一性, 我们列举多个任务难度大、相对复杂的图像数据集, 使得 NAS 算法能够在更多场景中得到评估. 表 3 中给出这些可用数据集的用途与下载链接.

表 3 评测数据集的用途与下载方式

数据集	任务	下载链接
Cifar-10/100[78]	物体分类	http://www.cs.toronto.edu/~kriz/cifar.html
ImageNet[79]	物体识别	http://www.image-net.org/
Penn Treebank[80]	自然语言处理	http://www.fit.vutbr.cz/~imikolov/rnnlm/
WikiText[81]	自然语言处理	https://www.salesforce.com/products/einstein/ai-research/the-wikitext-dependency-language-modeling-dataset/
RAF-DB[82]	人脸表情识别	http://www.whdeng.cn/RAF/model1.html
Flowers102[83]	细粒度分类	http://www.robots.ox.ac.uk/~vgg/data/flowers/102/
MIT67[84]	场景识别	http://web.mit.edu/torralba/www/indoor.html
COCO[85]	目标检测	https://cocodataset.org/#download
DIV2K[86]	图像超分辨率重建	https://data.vision.ee.ethz.ch/cvl/DIV2K/

(1) 常用数据集

目前, 为了评测神经结构搜索算法的性能, 在图像识别任务中, 常用的评测数据集主要包括 Cifar-10/100 和 ImageNet; 在自然语言处理任务中, 常用的评测数据集主要是 Penn Treebank 和 WikiText-2.

Cifar-10/100 数据集[78]是 8 000 万小图像数据集的标记子集. Cifar-10 数据集包含 60 000 幅大小为 32×32 的图像, 即 50 000 张训练图像和 10 000 张测试图像. 所有图像被划分为 10 个类别, 每个类包含 6 000 幅图像, 分别是 airplane、automobile、bird、cat、deer、dog、frog、horse、ship 和 truck. 该数据集划分为 5 个训练批次和 1 个测试批次, 每个批次包含 10 000 张图像. 测试批次包含从每个类别中随机选择的 1 000 张图像. 训练批次包含剩余的所有图像, 但一些训练批次可能存在类别分布不均衡的现象, 即某个类别的样本数多于另一个类别的样本数. 同 Cifar-10 数据集相似, Cifar-100 数据集被划分为 100 个类别, 每个类包含 600 个图像, 其中 500 张训练图像和 100 张测试图像. 值得注意的是, Cifar-100 数据集的类别标签含有两类, 即细标签 (所属类别, 100 类) 和粗标签 (所属超类, 20 类).

ImageNet 数据集[79]是目前图像识别领域中最常用的基准数据集. 它是根据 WordNet 层次结构组织的图像数据集. 目前已有几千万的图片被手工标注, 至少一百万的图像提供边界框. ImageNet 数据集包含了 14 197 122 幅图片, 涵盖 21 841 个类别, 每个类别大约有 100 到 1 000 张图像. 自 2010 年之后, 每年度的 ImageNet 大规模视觉识别挑战赛 (ILSVRC) 使用 ImageNet 的一个子集, 共有约 120 万张训练图像, 50 000 张验证图像, 以及 150 000 张测试图像, 共 1 000 个类别. 同 Cifar 数据集相比, ImageNet 包含的数据量与类别更多, 分辨率更高, 识别难度也更高.

Penn Treebank 数据集[80]是自然语言处理领域常用的语料库, 对语料进行标注, 标注内容包括词性标注和句法分析. 预料来源为 1989 年华尔街日报, 包括 1 M 的词汇量, 共 2 499 篇文章.

WikiText 英语词数据库[81]是一个包含 1 亿个词汇的英文词库数据, 这些词汇是从 Wikipedia 的优质文章和标杆文章中提取得到, 包括 WikiText-2 和 WikiText-103 两个子集, 相比 Penn Treebank (PTB) 数据库中的词汇数量, WikiText-2 数据集是 PTB 的 2 倍, WikiText-103 数据集是 PTB 的 110 倍. 每个词汇同时保留该词汇源自的原始文章, 非常适合长时依赖 (longterm dependency) 自然语言建模的场景需求.

(2) 其他数据集

在上述的常用数据集中, 现有的神经结构搜索算法取得惊人的效果. 然而, 我们认为 NAS 算法应该在更多、

更加复杂的任务中得到评估. 结合我们过往的研究经历, 我们尝试在人脸表情识别中验证 NAS 算法的有效性, 可得到的效果差强人意. 所以, NAS 需要在不同数据库中评估, 验证算法的鲁棒性. 下面我们列举几种多个视觉任务中常用的数据集, 并给出详细介绍.

RAF 数据集[82]是目前人脸表情识别中最常用的数据集, 拥有从互联网中下载的大约 3 万幅多样的人脸图像. 在众包标注的参与下, 每幅图像都由大约 40 名注释者独立标注. 该数据库中的人脸图像在被使者的年龄、性别和种族、头部姿态、光照条件、遮挡等方面有很大变化. RAF-DB 共包含 29 672 张真实面部图像, 被划分为 7 个类别. 特殊的是, 该数据库中有两个不同的子集, 包含单标签子集 (7 类基础情绪) 和双标签子集 (12 类复合情绪). 在现阶段的人脸表情识别任务中, 常使用该数据库中的单标签子集. 值得注意的是, 人脸表情识别是一种复杂的计算机视觉任务, 不易达到高精度的准确率, 对评估神经结构搜索算法的性能有很大的挑战.

Flowers102 数据集[83]是用于评估细粒度图像分类算法的数据库之一, 由 102 种产自英国的花卉组成, 每类有 40–258 张花卉图像. 这些图像有很大的尺寸、姿态和光照变化. 此外, 有些类别在类别中的变化很大, 而部分类别又十分相似.

MIT67 数据集[84]是麻省理工学院公开的室内场景识别数据集. 室内场景识别是高水平视觉领域中一个具有挑战性的开放性问题. 该数据库包含 67 个室内类别, 共计 15 620 张图像. 不同类别的图像数量不同, 但每个类别至少有 100 张图像.

COCO 数据集[85]是由微软发布的大型图像数据集, 专为对象检测、分割、人体关键点检测、语义分割和字幕生成而设计, 包括 MS-COCO 2014, MS-COCO 2015 和 MS-COCO 2017. MS-COCO 2014 和 MS-COCO 2017 提供训练集、验证集和测试集, MS-COCO 2015 仅有测试集. MS-COCO 2014 包括 82 783 张训练图像、40 504 张验证图像和 40 775 张测试图像. MS-COCO 2017 包含 118 000 张训练图像和 5 000 张验证图像. 为评估目标检测算法, 该数据集提供 80 个类别, 超过 200 000 张图片. 每张图片的标签包括目标检测框的位置以及所属类别, 背景复杂, 具有丰富的小目标, 评估标准更加严格.

DIV2K 数据集[86]是一个高质量的图像数据集, 包含 1 000 张 2K 分辨率的高清图像, 其中 800 张训练图像、100 张验证图像和 100 张测试图像, 近年来被广泛应用在图像超分辨率重建任务中.

3.2 规范化评测标准

对于从事深度学习的研究人员来说, 设计一个性能优越的神经网络是一件很繁琐的工作, 使得 AI 从业者对自动化神经结构搜索算法产生极大兴趣. 目前, 尽管学术界和工业界在神经结构搜索算法方面取得重大进展, 但相较于其他成熟的机器学习领域, NAS 算法的实验评估质量令人堪忧, 因此亟需一套客观的规范化评测标准, 公平地对比现有的神经结构搜索算法, 促进该领域的持续发展. 早期, Lindauer 等人[87]已经给出一系列神经结构搜索中的研究实践要求. 在此基础上, 我们结合最新的关于神经结构搜索的研究, 创新性地总结出若干点规范化评测标准, 希望给读者提供一个清晰的实验设置要求.

(1) 公平的实验细节设置

众所周知, 深度学习中的诸多训练细节和技巧对最终结果有着很大影响. 尤其是, 神经结构搜索算法在搜索阶段中采用的优化器、正则化方法、学习率变化、数据增强的技巧、权重衰减系数等超参数需要在介绍实验结果前详细给出, 便于读者复现出原论文中的实验效果. 换句话说, 只有采用相同的训练策略, 两种搜索算法才能够进行公平对比, 否则无法验证搜索策略的有效性. 另外, 实验平台与应用的深度学习开源框架在一定程度上也影响着网络结构的性能表现. 具体来说, GPU 显卡的计算能力各不相同, 直接影响结构搜索的耗时结果. 目前大部分研究人员都是采用 NVIDIA Tesla V100 作为实验平台, 并报告相应的延迟和搜索时间等结果.

(2) 随机搜索 (random search) 和消融实验 (ablation study) 的必要性

在实验评估环节中, 基准实验是实验评估内容中的重要部分之一. 神经结构搜索中最重要的基准实验就是随机搜索, 搜索算法的性能表现必须要同随机搜索作比较. 目前很多的神经结构搜索研究避免与随机搜索做出对比, 无法保证实验环节的严谨性与实验结果的有效性. 尤其是, 很多研究[71,72,88]开始反思随机搜索与先进搜索算法间的性能差异, 经对比后发现, 随机搜索的性能有时甚至超过搜索算法的性能. 在一些好的初始化条件下, 随机搜索

到的网络结构在测试数据上的表现会更好. 因此, 我们认为 NAS 实验环节中必须增加同随机搜索的对比, 更好地凸显搜索算法的竞争力.

另外, 在机器学习领域, 特别是复杂的深度神经网络中, 需要采用消融实验描述网络中每个模块的作用, 更好地理解网络结构的行为. 神经结构搜索算法的评估更是如此, 为了验证提出搜索策略的有效性, 非常有必要完成消融实验, 研究单个算法模块的重要性. 关于具体的消融步骤, 我们认为反思期中 Yang 等人的工作[71]是非常有意义的, 建议读者可以借鉴相关的实验设置, 拆分系统模块并评估每个模块的性能.

(3) 神经结构搜索实验的基准要求

神经结构搜索实验的基准要求主要包括预训练、数据划分、搜索空间的选择等. 以 DARTS[13]为代表的神经结构搜索算法, 需要人为提前拆分 Cifar-10 数据集中的训练数据为训练集和验证集, 以便于分别更新网络权重和结构系数. 所以, 为保证两阶段更新系数的过程中使用的数据相同, 需要研究人员确定数据的划分标准, 明确训练和验证两部分的范围. 尤其是目前很多任务中使用的数据库不再划分验证集 (如 RAF-DB[82]), 更需要在实验设置中明确数据划分的依据. 此外, 在前面的章节中, 我们已经介绍多种搜索空间. 基于不同搜索空间的方法之间不具有对比性, 例如基于 DARTS 搜索空间[13]的方法只能同 DARTS 系列[13,50,51,56]的方法进行比较. 尤其对于一些特定任务设计的搜索空间, 需要重新在新的搜索空间中评估现有的搜索方法.

为了提高 NAS 算法的可重复性, 降低搜索算法的计算需求, 同时公平地衡量 NAS 算法的真实性, 有研究提出公共神经结构搜索的基准结构数据集[89,90], 能够在毫秒时间内从数据集中查找并评估模型的质量. 这种策略的好处在于公平比较每种神经结构搜索算法得到的网络结构, 同时加速搜索、实时跟踪算法的性能. 我们相信这种结构数据基准在将来的研究中会发挥越来越大的作用, 并成为 NAS 领域内的研究共识.

(4) 结果报告方式的统一性

受神经结构搜索过程中的不稳定性影响, NAS 算法搜索到的结构与性能表现总是随机的. 例如, 根据 Li 等人的反思[54], 在相同的训练数据上运行两次搜索算法, 不一定得到相同的结果. 受益于近年来基于权重共享的神经结构搜索算法大大降低搜索耗时成本, 我们有必要报告多次实验后的结果, 不能仅报告最优的性能表现, 例如在不同的随机种子设置下, 多个搜索到的网络结构取得的平均性能表现.

4 挑战与展望

神经结构搜索任务是机器学习领域中一个非常有挑战性的问题, 拥有非常高的学术价值与广泛的应用前景. 尽管近年来不断有效果显著的神经结构搜索算法被提出, 然而目前的神经结构搜索方法中仍存在诸多问题, 需要学术界和工业界共同关注、不断研究并解决. 可以说, 目前的神经结构搜索任务仍处于初级发展阶段, 该领域仍然有一些研究问题亟待解决.

(1) 神经结构搜索算法在更多任务中评估表现

目前优秀的神经结构搜索算法大多在 Cifar-10/100 或 ImageNet 数据集上进行性能评估, 即集中解决视觉任务中的图像识别问题, 很少关注在其他任务上的实验效果, 例如对抗学习 (adversarial learning)、视频处理 (video processing)、图网络 (graph network) 和超分辨率重建 (super-resolution) 等研究内容. 尤其是同 Cifar 数据集相比较, 其他任务的复杂度会更大, 相应地对搜索空间和搜索策略的要求也越高. 此外, 一套通用型的神经结构搜索系统是有重要意义的. 根据任务类型, 该系统能够自适应地设计出性能优越的网络结构, 真正意义上实现机器的全自动化, 消除人工设计的影响. 因此, 我们相信有必要扩大神经结构搜索算法的应用范围, 增强其通用性.

(2) 不均衡性对可微分结构搜索的影响

众所周知, 以 DARTS[13]为代表的连续可微分结构搜索算法存在严重的优化误差问题, 尤其是两阶段的优化步骤加剧这一消极影响. 原因在于, 两阶段优化的目标 (网络权重ω和结构系数α) 在数量上是极不均衡的. 常见的是, 网络的可学习权重数量通常是数百万, 然而结构系数仅仅有数百, 例如 DARTS 需要学习的结构系数只有 224 个. 这种不均衡特性必然对搜索阶段产生不利的影响. 此外, 在数据样本不均衡的背景下, 机器学习算法的表现会很差. 由此我们联想到不均衡样本是否也会对结构搜索产生影响. 尤其是在样本和优化目标的双重不均衡作

用下, 搜索算法得到的网络结构是否还是最优的, 需要今后的研究深入反思这一问题. 例如, 我们可以分别探究两部分不均衡的影响, 首先利用重新加权或重新采样的方式, 缓解样本不均衡对结构搜索性能的影响; 然后, 设计自适应权重系数, 强化网络搜索过程中对结构系数的优化; 最后通过联合学习的方式, 解决双重不均衡问题. 此外, 造成不均衡性的重要元素之一的结构系数α是否真实反映选择操作的重要性仍需要进一步探究. 尤其从目前实验结果可以看出, 每种操作对应的结构系数之间差别不大, 很难准确评估操作的重要性, 继而影响网络结构性能的评估. 我们认为解决这一问题可以从性能评估的角度重新定义操作的重要性度量, 将准确率等指标作为反馈, 指标数值越高代表操作更重要, 即搜索到的神经网络结构更优.

(3) 人工设计的影响依旧存在

近年来, 关于神经结构搜索方法的研究受到越来越多的关注, 各类神经结构搜索算法搜索出的深度神经网络已经逐渐在性能上超越了人工设计的神经网络. 然而, 现阶段的 NAS 算法还未达到全自动的要求, 而且主要针对图像识别任务, 无法做到特定任务上的自适应深度神经结构搜索的目标. 另外, 神经结构搜索算法中重要组成部分之一的搜索空间是人工提前设定的, 算法仅仅能够在空间内选择可能的计算操作, 无法选择空间内没有的计算单元. 以 DARTS 搜索空间[13]为例, 候选操作集合中仅包含 8 个操作, 专家提前将卷积核尺寸限定为 5×5 和 7×7, 极大地降低搜索空间的自适应能力. 同时, 搜索结构的层数, 即结构单元数也是人为设定好的. 这种预先设置的方式也限制了搜索空间的规模. 尽管小的搜索空间可以降低搜索成本, 但也束缚搜索算法的自动搜索性能. Xie 等人[4]提出理想状态下的搜索空间大小应超过 10^{1000}, 目前的搜索空间远远没有达到这一目标. 我们认为人工干预依旧影响着 NAS 领域, 尤其是现阶段的搜索空间不但被人为限定, 还限制了搜索算法发现性能更优的网络结构. 因此, 亟需一种解决办法扩大搜索空间规模, 消除人工设计的影响. 例如, 不再预先设定卷积核尺寸或引入一种自适应系数, 对现有空间内的候选操作进行线性或非线性组合, 形成新的计算操作, 扩大搜索空间.

(4) 小规模代理任务限制搜索算法性能

目前, 主流的神经结构搜索算法大多采用小规模代理任务搜索的方式, 即在小规模代理数据集上通过 NAS 算法进行网络结构搜索, 在搜索过程中利用代理任务上的评估性能估计在实际任务中的性能, 并将搜索到的神经网络直接迁移到实际任务中. 这种基于小规模代理任务搜索评估的策略很大程度上缓解计算资源高昂的弊端, 能够以较低的搜索成本实现神经结构搜索的目标. 然而, 这种方式的成功必须建立在代理任务和实际任务之间具有极高相似度的前提下, 否则搜索的神经网络仅能在小规模代理任务中实现优越的性能表现. 为了实现通用型的 NAS 算法, 任务间的差异性是必然存在的. 因此, 我们认为这种采用小规模代理任务搜索评估是不够理想的, 需要直接在实际目标任务中搜索网络结构并评估其性能, 并且要解决因此带来的计算开销问题.

(5) 开拓研究基于神经结构搜索的自动化模型压缩技术

诸多计算机视觉任务因深度神经网络得到空前发展, 并达到前所未有的高度, 但是模型的复杂度、高额的存储空间等限制技术落地到手机等硬件平台. 为了加速算法在真实场景的应用, 模型压缩能够最大限度地降低神经网络模型的计算空间和运行时间. 当前来说, 神经网络模型部署的一般步骤主要是网络设计 (主要依靠专家人工设计, 部分利用 NAS 技术)、模型剪枝和模型量化. 很容易想到一个问题, 神经结构搜索算法得到的大尺寸网络结构经过模型压缩后是否依旧保持最优性. 因此, 我们认为有必要研究模型压缩和神经结构搜索之间的模型关系, 并设计出基于神经结构搜索的自动化模型压缩算法, 实现真正意义上的全自动化人工智能. 例如, 联合模型剪枝与候选操作选择, 实现自动化模型压缩.

5 总 结

神经结构搜索是自动化机器学习中的一个重要研究内容, 旨在替代传统模式的人工设计神经网络方法. 经过近 6 年的快速发展, NAS 展示其成为深度学习标准工具的潜力, 可应用于不同领域的人工智能任务. 目前, 神经结构搜索对一些计算机视觉任务做出了重大贡献, 但也存在着严重缺陷, 阻碍了其更广泛的应用. 本文全面地回顾神经结构搜索方法. 根据神经结构搜索的发展历程, 我们划分为 4 个阶段, 分别为早期、快速发展期、应用期和反思期. 然后我们介绍现阶段验证 NAS 算法性能时常用的几种数据集, 建设性给出其他评估数据集, 使 NAS 算法在更

多视觉任务中得到评估. 除此之外, 我们认为现阶段 NAS 算法的实验评估是极不公平的, 严重影响该领域的良好发展. 为保证实验评估的公平性, 我们创新性地总结一系列规范化评估手段. 最后, 我们分析神经结构搜索领域当前仍面临的一些挑战, 如双重不均衡的影响等. 在未来的研究工作中, 我们将考虑神经结构搜索在人脸表情合成、超分辨率重建和模型压缩中的应用, 分别从搜索空间和搜索策略角度探究 NAS 算法在上述应用中的性能, 提高深度学习应用系统的智能性与实时性.

References:

[1] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436–444. [doi: 10.1038/nature14539]

[2] Elsken T, Metzen JH, Hutter F. Neural architecture search: A survey. Journal of Machine Learning Research, 2019, 20(55): 1–21.

[3] Wistuba M, Rawat A, Pedapati T. A survey on neural architecture search. arXiv: 1905.01392, 2019.

[4] Xie LX, Chen X, Bi KF, *et al*. Weight-sharing neural architecture search: A battle to shrink the optimization gap. arXiv: 2008.01475, 2020.

[5] Ge DH, Li HS, Zhang L, Liu RY, Shen PY, Miao QG. Survey of lightweight neural network. Ruan Jian Xue Bao/Journal of Software, 2020, 31(9): 2625–2653 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5942.htm [doi: 10.13328/j.cnki.jos.005942]

[6] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

[7] Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 2261–2269. [doi: 10.1109/CVPR.2017.243]

[8] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 3431–3440. [doi: 10.1109/CVPR.2015.7298965]

[9] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Proc. of the 18th Int'l Conf. on Medical Image Computing and Computer-assisted Intervention. Munich: Springer, 2015. 234–241. [doi: 10.1007/978-3-319-24574-4_28]

[10] Real E, Moore S, Selle A, Saxena S, Suematsu YL, Tan J, Le QV, Kurakin A. Large-scale evolution of image classifiers. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: PMLR, 2017. 2902–2911.

[11] Zoph B, Le QV. Neural architecture search with reinforcement learning. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[12] Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. In: Proc. 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8697–8710. [doi: 10.1109/CVPR.2018.00907]

[13] Liu HX, Simonyan K, Yang YM. DARTS: Differentiable architecture search. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[14] Cai H, Zhu LG, Han S. ProxylessNAS: Direct neural architecture search on target task and hardware. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[15] Chen YK, Yang T, Zhang XY, Meng GF, Xiao XY, Sun J. DetNAS: Backbone search for object detection. In: Proc. of the 33rd Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2019. 6642–6652.

[16] Ghiasi G, Lin TY, Le QV. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 7029–7038. [doi: 10.1109/CVPR.2019.00720]

[17] Tan MX, Pang RM, Le QV. EfficientDet: Scalable and efficient object detection. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 10778–10787. [doi: 10.1109/CVPR42600.2020.01079]

[18] Xu H, Yao LW, Li ZG, Liang XD, Zhang W. Auto-FPN: Automatic network architecture adaptation for object detection beyond classification. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 6648–6657. [doi: 10.1109/ICCV.2019.00675]

[19] Yao LW, Xu H, Zhang W, Liang XD, Li ZG. SM-NAS: Structural-to-modular neural architecture search for object detection. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence, 32nd Conf. on Innovative Applications of Artificial Intelligence, the 10th Symp. on Educational Advances in Artificial Intelligence. New York: AAAI, 2020. 12661–12668. [doi: 10.1609/aaai.v34i07.6958]

[20] Guo JY, Han K, Wang YH, Zhang C, Yang ZH, Wu H, Chen XH, Xu C. Hit-detector: Hierarchical trinity architecture search for object detection. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 11402–11411. [doi: 10.1109/CVPR42600.2020.01142]

[21] Chen B, Ghiasi G, Liu HX, Lin TY, Kalenichenko D, Adam H, Le QV. MnasFPN: Learning latency-aware pyramid architecture for object detection on mobile devices. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE,

2020. 13604–13613. [doi: 10.1109/CVPR42600.2020.01362]

[22] Du XZ, Lin TY, Jin PC, Ghiasi G, Tan MX, Cui Y, Le QV, Song XD. SpineNet: Learning scale-permuted backbone for recognition and localization. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 11589–11598. [doi: 10.1109/CVPR42600.2020.01161]

[23] Shaw A, Hunter D, Landola F, Sidhu S. SqueezeNAS: Fast neural architecture search for faster semantic segmentation. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision Workshop. Seoul: IEEE, 2019. 2014–2024. [doi: 10.1109/ICCVW.2019.00251]

[24] Liu CX, Chen LC, Schroff F, Adam H, Hua W, Yuille AL, Li FF. Auto-DeepLAB: Hierarchical neural architecture search for semantic image segmentation. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 82–92. [doi: 10.1109/CVPR.2019.00017]

[25] Lin PW, Sun P, Cheng GL, Xie SR, Li X, Shi JP. Graph-guided architecture search for real-time semantic segmentation. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 4202–4211. [doi: 10.1109/CVPR42600.2020.00426]

[26] Weng Y, Zhou TB, Li YJ, Qiu XY. NAS-unet: Neural architecture search for medical image segmentation. IEEE Access, 2019, 7: 44247–44257. [doi: 10.1109/ACCESS.2019.2908991]

[27] Gong XY, Chang SY, Jiang YF, Wang ZY. AutoGAN: Neural architecture search for generative adversarial networks. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 3223–3233. [doi: 10.1109/ICCV.2019.00332]

[28] Gao C, Chen YP, Liu S, Tan ZX, Yan SC. AdversarialNAS: Adversarial neural architecture search for GANs. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 5679–5688. [doi: 10.1109/CVPR42600.2020.00572]

[29] Zhou P, Xie LX, Zhang XP, Ni BB, Tian Q. Searching towards class-aware generators for conditional generative adversarial networks. arXiv: 2006.14208, 2020.

[30] Wang HC, Huan J. AGAN: Towards automated design of generative adversarial networks. arXiv: 1906.11080, 2019.

[31] Baker B, Gupta O, Naik N, Raskar R. Designing neural network architectures using reinforcement learning. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[32] Zhong Z, Yan JJ, Wu W, Shao J, Liu CL. Practical block-wise neural network architecture generation. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 2423–2432. [doi: 10.1109/CVPR.2018.00257]

[33] Tan MX, Chen B, Pang RM, Vasudevan V, Sandler M, Howard A, Le QV. MnasNET: Platform-aware neural architecture search for mobile. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 2815–2823. [doi: 10.1109/CVPR.2019.00293]

[34] Liu CX, Zoph B, Neumann M, Shlens J, Hua W, Li LJ, Li FF, Yuille A, Huang J, Murphy K. Progressive neural architecture search. In: Proc. of the 15th European Conf. on Computer Vision. Munich: Springer, 2018. 19–35. [doi: 10.1007/978-3-030-01246-5_2]

[35] Xie LX, Yuille A. Genetic CNN. In: Proc. 2017 IEEE Int'l Conf. on Computer Vision. Venice: IEEE, 2017. 1388–1397. [doi: 10.1109/ICCV.2017.154]

[36] Liu HX, Simonyan K, Vinyals O, Fernando C, Kavukcuoglu K. Hierarchical representations for efficient architecture search. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[37] Real E, Aggarwal A, Huang YP, Le QV. Regularized evolution for image classifier architecture search. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence, the 31st Conf. on Innovative Applications of Artificial Intelligence, the 9th Symp. on Educational Advances in Artificial Intelligence. Honolulu: AAAI, 2019. 4780–4789. [doi: 10.1609/aaai.v33i01.33014780]

[38] Stanley KO, Clune J, Lehman J, Miikkulainen R. Designing neural networks through neuroevolution. Nature Machine Intelligence, 2019, 1(1): 24–35. [doi: 10.1038/s42256-018-0006-z]

[39] So DR, Liang C, Le QV. The evolved transformer. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 5877–5886.

[40] Miikkulainen R, Liang J, Meyerson E, Rawal A, Fink D, Francon O, Raju B, Shahrzad H, Navruzyan A, Duffy N, Hodjat B. Evolving deep neural networks. In: Kozma R, Alippi C, Choe Y, Morabito FC, eds. Artificial Intelligence in the Age of Neural Networks and Brain Computing. London: Academic Press, 2019. 293–312. [doi: 10.1016/B978-0-12-815480-9.00015-3]

[41] Suganuma M, Shirakawa S, Nagao T. A genetic programming approach to designing convolutional neural network architectures. In: Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence. Stockholm: IJCAI.org, 2018. 5369–5373. [doi: 10.24963/ijcai.2018/755]

[42] Chen YK, Meng GF, Zhang Q, Xiang SM, Huang C, Mu LS, Wang XG. Renas: Reinforced evolutionary neural architecture search. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4782–4791. [doi: 10.1109/CVPR.2019.00492]

[43] Pham H, Guan MY, Zoph B, Le QV, Dean J. Efficient neural architecture search via parameter sharing. In: Proc. of the 35th Int'l Conf.

on Machine Learning. Stockholm: PMLR, 2018. 4095–4104.

[44] Cai H, Chen TY, Zhang WN, Yu Y, Wang J. Efficient architecture search by network transformation. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence, the 30th Innovative Applications of Artificial Intelligence, the 8th AAAI Symp. on Educational Advances in Artificial Intelligence. New Orleans: AAAI, 2018.

[45] Brock A, Lim T, Ritchie JM, Weston N. SMASH: One-shot model architecture search through hypernetworks. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[46] Bender G, Kindermans PJ, Zoph B, Vasudevan V, Le Q. Understanding and simplifying one-shot architecture search. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 550–559.

[47] Guo ZC, Zhang XY, Mu HY, Heng W, Liu ZC, Wei YC, Sun J. Single path one-shot neural architecture search with uniform sampling. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 544–560. [doi: 10.1007/978-3-030-58517-4_32]

[48] Cai H, Gan C, Wang TZ, Zhang ZK, Han S. Once-for-all: Train one network and specialize it for efficient deployment. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[49] Elsken T, Metzen JH, Hutter F. Simple and efficient architecture search for convolutional neural networks. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[50] Xu YH, Xie LX, Zhang XP, Chen X, Qi GJ, Tian Q, Xiong HK. PC-darts: Partial channel connections for memory-efficient architecture search. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[51] Zela A, Elsken T, Saikia T, Marrakchi Y, Brox T, Hutter F. Understanding and robustifying differentiable architecture search. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[52] Luo RQ, Tian F, Qin T, Chen EH, Liu TY. Neural architecture optimization. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montreal: NeurIPS, 2018. 7827-7838. [doi: 10.5555/3327757.3327879]

[53] Xie SR, Zheng HH, Liu CX, Lin L. SNAS: Stochastic neural architecture search. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[54] Li L, Talwalkar A. Random search and reproducibility for neural architecture search. In: Proc. of the 34th Conf. on Uncertainty in Artificial Intelligence. Tel Aviv: AUAI Press, 2020. 367–377.

[55] Dong XY, Yang Y. Searching for a robust neural architecture in four gpu hours. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 1761–1770. [doi: 10.1109/CVPR.2019.00186]

[56] Chen X, Xie LX, Wu J, Tian Q. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 1294–1303. [doi: 10.1109/ICCV.2019.00138]

[57] Nayman N, Noy A, Ridnik T, Friedman I, Jin R, Zelnik-Manor L. XNAS: Neural architecture search with expert advice. In: Proc. of the 33rd Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2019. 1975–1985.

[58] Liang HW, Zhang SF, Sun JC, He XQ, Huang WR, Zhuang KC, Li ZG. Darts+: Improved differentiable architecture search with early stopping. arXiv: 1909.06035, 2019.

[59] Bi KF, Xie LX, Chen X, Wei LH, Tian Q. GOLD-NAS: Gradual, one-level, differentiable. arXiv: 2007.03331, 2020.

[60] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv: 1602.07360, 2016.

[61] Zhang XY, Zhou XY, Lin MX, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 6848–6856. [doi: 10.1109/CVPR.2018.00716]

[62] Howard AG, Zhu ML, Chen B, Kalenichenko D, Wang WJ, Weyand T, Andreetto M, Adam H. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv: 1704.04861, 2017.

[63] Wu BC, Dai XL, Zhang PZ, Wang YH, Sun F, Wu YM, Tian YD, Vajda P, Jia YQ, Keutzer K. FBNet: Hardware-aware efficient ConVnet design via differentiable neural architecture search. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 10726–10734. [doi: 10.1109/CVPR.2019.01099]

[64] Dai XL, Zhang PZ, Wu BC, Yin HX, Sun F, Wang YH, Dukhan M, Hu YQ, Wu YM, Jia YQ, Vajda P, Uyttendaele M, Jha NK. ChamNet: Towards efficient network design through platform-aware model adaptation. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 11390–11399. [doi: 10.1109/CVPR.2019.01166]

[65] Stamoulis D, Ding RZ, Wang D, Lymberopoulos D, Priyantha B, Liu J, Marculescu D. Single-path NAS: Designing hardware-efficient ConvNets in less than 4 hours. In: Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. Würzburg: Springer, 2019. 481–497. [doi: 10.1007/978-3-030-46147-8_29]

[66] Tan MX, Le QV. EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proc. of the 36th Int'l Conf. on Machine

Learning. Long Beach: PMLR, 2019. 6105–6114.

[67] Tan MX, Le QV. MixConv: Mixed depthwise convolutional kernels. In: Proc. of the 30th British Machine Vision Conf. Cardiff: BMVA, 2019.

[68] Fang JM, Sun YZ, Zhang Q, Li Y, Liu WY, Wang XG. Densely connected search space for more flexible neural architecture search. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 10625–10634. [doi: 10.1109/CVPR42600.2020.01064]

[69] Li GL, Zhang X, Wang ZT, Tan M, Feng JS, Li ZG, Zhang T. Hierarchical neural architecture search via operator clustering. arXiv: 1909.11926.

[70] Chu XX, Zhang B, Xu RJ, Li JX. FairNAS: Rethinking evaluation fairness of weight sharing neural architecture search. arXiv: 1907.01845, 2019.

[71] Yang A, Esperanca PM, Carlucci FM. NAS evaluation is frustratingly hard. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[72] Yu KC, Sciuto C, Jaggi M, Musat C, Salzmann M. Evaluating the search phase of neural architecture search. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[73] Carlucci FM, Esperança PM, Singh M, Gabillon V, Yang A, Xu H, Chen ZW, Wang J. MANAS: Multi-agent neural architecture search. arXiv: 1909.01051, 2019.

[74] Weng Y, Zhou TB, Liu L, Xia CL. Automatic convolutional neural architecture search for image classification under different scenes. IEEE Access, 2019, 7: 38495–38506. [doi: 10.1109/ACCESS.2019.2906369]

[75] Lu ZC, Whalen I, Dhebar Y, Deb K, Goodman E, Banzhaf W, Boddeti VN. NSGA-Net: Neural architecture search using multi-objective genetic algorithm (extended abstract). In: Proc. of the 29th Int'l Joint Conf. on Artificial Intelligence. Yokohama: IJCAI.org, 2020. 4750–4754. [doi: 10.24963/ijcai.2020/659]

[76] Zhou HP, Yang MH, Wang J, Pan W. BayesNAS: A bayesian approach for neural architecture search. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 7603–7613.

[77] Yao QM, Xu J, Tu WW, Zhu ZX. Efficient neural architecture search via proximal iterations. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence, the 32nd Conf. on Innovative Applications of Artificial Intelligence, the 10th Symp. on Educational Advances in Artificial Intelligence. New York: AAAI, 2020. 6664–6671. [doi: 10.1609/aaai.v34i04.6143]

[78] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Technical Report: University of Toronto. 2009.

[79] Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. Imagenet: A large-scale hierarchical image database. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255. [doi: 10.1109/CVPR.2009.5206848]

[80] Kombrink S, Mikolov T, Karafiát M, Burget L. Recurrent neural network based language modeling in meeting recognition. In: Proc. of the 12th Annual Conf. of the Int'l Speech Communication Association. Florence: ISCA, 2011.

[81] Merity S, Xiong CM, Bradbury J, Socher R. Pointer sentinel mixture models. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[82] Li S, Deng WH, Du JP. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 2584–2593. [doi: 10.1109/CVPR.2017.277]

[83] Nilsback ME, Zisserman A. Automated flower classification over a large number of classes. In: Proc. of the 2008 6th Indian Conf. on Computer Vision, Graphics & Image Processing. Bhubaneswar: IEEE, 2008. 722–729. [doi: 10.1109/ICVGIP.2008.47]

[84] Quattoni A, Torralba A. Recognizing indoor scenes. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 413–420. [doi: 10.1109/CVPR.2009.5206537]

[85] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft COCO: Common objects in context. In: Proc. of the 13th European Conf. on Computer Vision. Zurich: Springer, 2014. 740–755. [doi: 10.1007/978-3-319-10602-1_48]

[86] Timofte R, Agustsson E, van Gool L, *et al*. NTIRE 2017 challenge on single image super-resolution: Methods and results. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition Workshops. Honolulu: IEEE, 2017. 1110–1121. [doi: 10.1109/CVPRW.2017.149]

[87] Lindauer M, Hutter F. Best practices for scientific research on neural architecture search. Journal of Machine Learning Research, 2020, 21: 1–18.

[88] Xie SN, Kirillov A, Girshick R, He KM. Exploring randomly wired neural networks for image recognition. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 1284–1293. [doi: 10.1109/ICCV.2019.00137]

[89] Ying C, Klein A, Christiansen E, Real E, Murphy K, Hutter F. NAS-bench-101: Towards reproducible neural architecture search. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 7105–7114.

[90] Dong XY, Yang Y. NAS-bench-201: Extending the scope of reproducible neural architecture search. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

附中文参考文献:

[5] 葛道辉, 李洪升, 张亮, 刘如意, 沈沛意, 苗启广. 轻量级神经网络架构综述. 软件学报, 2020, 31(9): 2625–2653. http://www.jos.org.cn/1000-9825/5942.htm [doi: 10.13328/j.cnki.jos.005942]

李航宇 (1994－), 男, 博士生, 主要研究领域为机器学习, 模式识别和计算机视觉.

杨曦 (1988－), 女, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为机器学习, 模式识别和计算机视觉.

王楠楠 (1986－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为机器学习, 模式识别和计算机视觉.

高新波(1972－), 男, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为机器学习, 模式识别, 计算机视觉和计算智能.

朱明瑞 (1992－), 男, 博士, 讲师, 主要研究领域为机器学习, 模式识别和计算机视觉.