doi:10.11887/j.cn.202306014 http://journal.nudt.edu.cn

神经网络架构搜索研究进展与展望*

丁 丁[1],刘文哲[2],盛常冲[3],隋金坪[4],刘 丽[2]

(1. 国防科技大学 教研保障中心,湖南 长沙 410073;2. 国防科技大学 系统工程学院,湖南 长沙 410073;
3. 海军工程大学 电磁能技术全国重点实验室,湖北 武汉 430033;
4. 海军大连舰艇学院 作战软件与仿真研究所,辽宁 大连 116016)

摘 要:神经网络架构搜索旨在针对不同任务,自动化地搜索得到性能最优的神经网络结构,是深度学习、计算机视觉技术结合当前现实需求应运而生的一大重要科学问题。对近年来神经网络架构搜索研究进行梳理、归类和评述;阐述神经网络架构搜索的定义和意义,全方位剖析当前研究所面临的难点与挑战;以此为基础,对主流的搜索策略进行阐述和归纳;探讨研究潜在的问题及未来颇具潜力的研究方向,以期推动该领域的进一步发展。

关键词:深度学习;神经网络架构搜索;自动机器学习;强化学习;搜索空间设计;搜索策略;进化算法

中图分类号:TP183 **文献标志码**:A **开放科学(资源服务)标识码(OSID)**:

文章编号:1001-2486(2023)06-100-32

听语音 聊科研 与作者互动

State of the art and prospects of neural architecture search

DING Ding[1], *LIU Wenzhe*[2], *SHENG Changchong*[3], *SUI Jinping*[4], *LIU Li*[2]

(1. Center for Teaching and Research Support, National University of Defense Technology, Changsha 410073, China;
2. College of Systems Engineering, National University of Defense Technology, Changsha 410073, China;
3. National Key Laboratory of Electromagnetic Energy, Naval University of Engineering, Wuhan 430033, China;
4. Operational Software and Simulation Institute, Dalian Navy Academy, Dalian 116016)

Abstract: Neural architecture search is a task that aims to automatically search for the optimal neural network structure for different tasks, which is of great importance and inevitability in the joint development of deep learning and computer vision to the current stage. A comprehensive review of the research on neural network search was provided. In specific, the definition and significance of neural architecture search were introduced, and the difficulties and challenges faced in relevant research were deeply analyzed. Based on this, the mainstream search strategies was elaborate and summarize; Finally, the potential problems and possible future research directions were summarized and discussed to promote further development in this field.

Keywords: deep learning; neural architecture search; automation of machine learning; reinforcement learning; search space design; search strategy; evolutionary algorithms

近年来,随着大数据的兴起与算力资源的丰富,以深度神经网络[1-2]为代表的人工智能技术得以飞速发展,已从最初的计算机视觉[3-6]和语音识别[7]领域,飞速延伸到如今自动驾驶[8]、癌症检测[9]、机器翻译[10]、虚拟游戏[11-14]、人脸识别[15]、地震预测[16]、药物发现[17]、推荐系统、机器人等大量科学和技术领域。仅在过去十年间,深度神经网络相关技术在诸多应用领域便取得了重大突破,在部分领域,深度神经网络甚至达到超越人类专家的水平。

作为一种数据表示学习技术,基于连接主义主张的深度神经网络的本质是,通过构建具有多个隐层的非线性变换,自动地从大量原始数据中学习丰富的、层次化的特征表达,以便于更好地建立从底层信号到高层语义的映射关系[18]。深度神经网络具有强大的特征学习能力,脱离了对特征工程(例如尺度不变特征转换(scale-invariant feature transform, SIFT)[19]、梯度直方图(histogram of oriented gradient, HOG)[20]等手工设计特征)的依赖,获得学术界和工业界的广泛

* **收稿日期**:2022-02-19
基金项目:国家自然科学基金资助项目(61872379)
作者简介:丁丁(1982—),女,湖南长沙人,高级工程师,博士,E-mail:dd2598752468@163.com;
隋金坪(通信作者),男,山东日照人,讲师,博士,E-mail:suijinping13@nudt.edu.cn

关注。深度神经网络的巨大成功，很大原因归功于主干神经网络架构的不断更新和发展，如图 1 所示，从最初的 AlexNet[4] 到后来的 ZFNet[21]、VGGNet[22]，以及 Inception[23]、ResNet[24]、DenseNet[25]、ResNeXt[26] 等。图 2 展示了近年来极具代表性的深度卷积神经网络结构在基于 ImageNet 的 ILSVRC 竞赛[27] 中图像分类任务上的性能对比，AlexNet 提出非线性激活函数 ReLU 和防止过拟合的 dropout 和数据增强方法，在竞赛中比上一年冠军的错误率下降了 10.9%；VGG16 使用多个较小的 3×3 卷积核代替卷积核较大的卷积层，借此通过增加网络的深度和通道数来提升网络最终的性能，指明卷积神经网络未来的一种发展方向；Inception V3[28] 设计由众多平行连接组成的 Inception 结构代替原本的卷积层，并且继承 VGGNet 的卷积核分解思想，进一步分解 3×3 卷积核为 1×3 和 3×1 不对称卷积，增加网络尺寸、设计卷积核和结构的多样性也成为卷积神经网络的未来发展方向；ResNet-101 引入残差网络结构，跳跃式的结构使得某一层的输出可以跨越

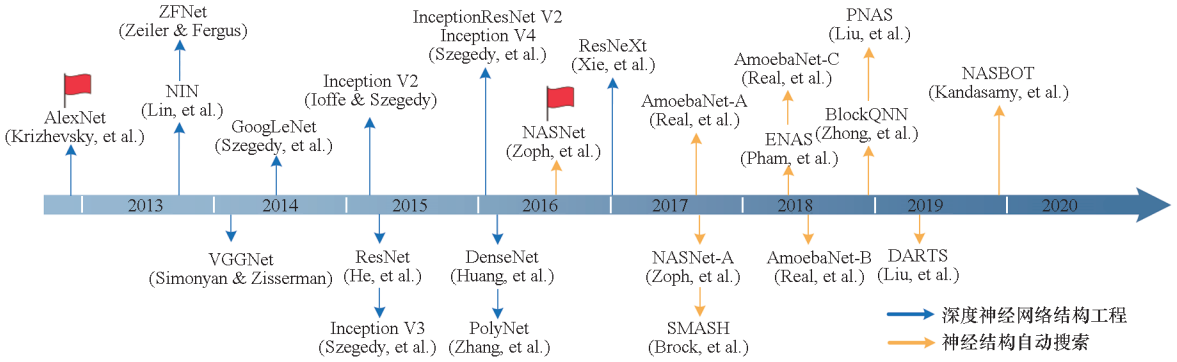


图 1　神经网络架构搜索研究中的代表性工作

Fig. 1　Representative work in the research of neural architecture search

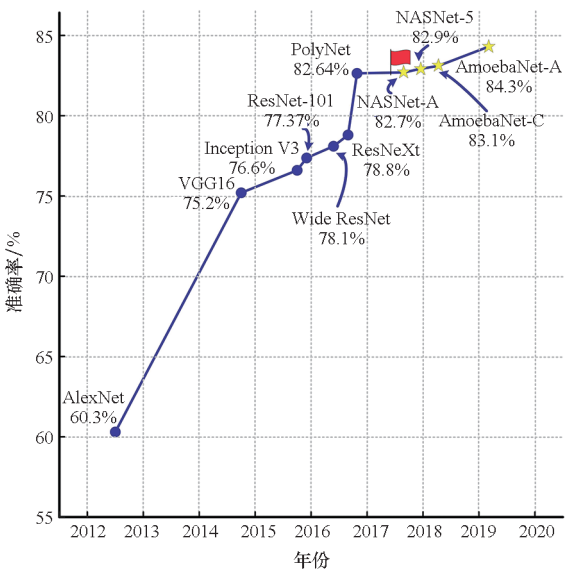


图 2　ImageNet 数据集上的代表性工作及其 TOP-1 准确率

Fig. 2　Representative works and their TOP-1 accuracy on ImageNet

多层作为之后某层的输入，解决随着网络层数的加深，难以训练的问题，拓展网络的层数至数千层；随着网络加深的边际效应递减，Wide ResNet[29] 进一步探究网络宽度对于性能的影响；PolyNet[30] 则朝着结构多样化的方向，提出可以灵活替换网络中不同部分的新结构 PolyInception。可见合理有效的深度神经网络结构是很多任务实现性能跃升的引擎。尽管这些神经网络结构取得了很大成功，但这并不代表它们是最优的，神经网络结构的设计仍有很大的提升空间。虽然深度神经网络脱离了对特征工程的依赖，但是其性能仍然依赖于各种超参数的调试，且设计出优秀的深度神经网络结构需要领域知识进行反复测试，成本很高。

经过了前几年的蓬勃发展时期，目前人工设计神经网络模型的研究也进入了瓶颈期。因此，研究者们另辟蹊径，更多地探究令神经网络自行搜索其最优结构的方法，即神经网络架构搜索（neural architecture search，NAS）技术。NAS 是一种自动设计人工神经网络的技术，其目的是自动学习一种神经网络拓扑结构，使其在目标任务上达到最优性能。NAS 研究具有重要的理论意义与实用价值。首先，神经结构搜索技术通过自动化搜索，可以快速找到性能更优的结构，以节省因超参数调试而耗费的大量时间和精力；其次，NAS 技术往往可以搜索得到目标任务上性能超越人类专家设计的网络结构，有助于研究神经网络可解释性等相关工作的进展；再次，NAS 技术不要求对于神经网络的深入理解以及相关应用领域的专

业知识,智能化的结构搜索让深度学习技术可以更广泛地为非专家所使用,令更多的用户受益于深度学习。

NAS 的想法并不是近期提出的,更早可以追溯到 Yao[31] 和 Stanley 等[32] 的研究工作,但是跟深度学习一样沉寂了很长一段时间。直到近期,因 Zoph 等[33],以及 Klein 等[34] 的工作而重新焕发活力。Zoph 等[33] 采用强化学习的方法,搜索到的神经网络结构在 CIFAR-10 数据集上达到 3.65% 的图像分类错误率,在多 40 个卷积核的基础上,其推断速度比当时专家设计的最佳结构 DenseNet[25] 仍快 5%,且错误率降低了 0.09%。随后,相关研究者将研究重点从人工设计深度神经网络结构,转为 NAS 算法研究,进而加速了 NAS 算法的发展[35-38]。图 1 展示一些具有代表性的 NAS 算法,作为 NAS 技术再次崛起的开端,NAS-RL[33] 基于强化学习原理,展开了对网络结构的搜索;后续 AmoebaNet[36] 提出用进化算法解决网络结构搜索问题,并基于 NAS 问题的特点引入年龄因子,偏向于对新发现模型的进一步探索;NASNet-A[35] 基于结构的多样性,按照是否降维特征图,设计了两种全新结构,并组成网络;之后 SMASH[39] 和 ENAS[40] 分别通过训练快速赋予候选结构权值的 HyperNet 和引入参数共享方法,避免从零开始训练候选网络,以加速网络搜索过程;PNAS[37] 的搜索顺序由易到难,提出了仅由一个全新的结构堆叠组成的神经网络;BlockQNN[41] 基于分布式异步 Q 学习框架和早停策略完成结构搜索任务; DARTS[38] 将整个网络搜索的时间缩短至消费级显卡的一个 GPU 日(GTX 1080Ti),大幅度降低研究门槛以及搜索成本,进一步推动了 NAS 技术的发展; NASBOT[42] 则尝试在网络结构搜索中引入贝叶斯方法。总而言之,NAS 方法已经在一些任务上取得比人工设计的神经网络结构更高的准确率,例如图像分类[35-37,39]、物体检测[43-44] 和图像语义分割[45-46] 等。

2019 年,Elsken 等[47] 对 NAS 算法进行了粗粒度总结,从搜索空间、搜索策略、性能评估策略三个方面对 2018 年之前的代表性工作进行了归纳。2020 年,Deng 等[48] 从实现网络压缩的角度,简要介绍了自动优化网络结构的 NAS 方法。根据调研和了解,国内目前还没有 NAS 方面的全面综述。鉴于 NAS 的重要研究意义以及其快速发展趋势,有必要对已有 NAS 方法进行全面的综述,对最具代表性的方法进行梳理、归纳、性能比较以及评述,对存在的问题进行探讨,并分析展望该领域未来可能的发展方向。

尽管 NAS 技术也可以应用到 RNN[33,38,40] 或者 transformer[49] 相关网络结构的自动搜索之中,但是现有工作还是较少,代表性不强。故本文主要介绍 NAS 在 ImageNet 数据集上搜索卷积神经网络结构解决分类问题的进展,辅以介绍 NAS 在目标检测、图像分割等领域的应用,以及搜索循环神经网络解决语言模型问题的现状。

本文对当前神经网络架构搜索的定义和意义,所面临的难点与挑战,主流的搜索策略进行了系统性地介绍、阐述和归纳,并于文末探讨了研究潜在的问题及未来可能的研究方向,以期进一步推动与此问题相关的研究。

1 卷积神经网络回顾

1.1 基本组成

1.1.1 基本算子

1)卷积层。卷积层是卷积神经网络(convolutional neural networks, CNN)的主要组成部分。卷积层的操作主要是将前一层特征映射到当前层中,固定尺寸的小窗口滑动取样得到的特征映射依次与若干二维的卷积核做卷积,得到加权和,组成具有若干通道数的、新的特征映射。即卷积层的每个输出只是某一固定窗口输入的加权和,并且不同通道的同一位置的固定窗口中的权值是共用的,该固定小窗口的大小也被称为感受野。其主要参数有卷积类型(普通卷积、不对称卷积、可分离卷积、空洞卷积等)、卷积核尺度(1×1、3×3、1×3、3×1、5×5、7×7、1×7、7×1 等)、卷积通道数(24、36、64 等)、滑动步长(1、2 等)等。

2)下采样层。下采样层的功能是降低特征映射的维度,并且分别施加在输入特征映射的每一个通道上,依次对固定尺寸的小窗口滑动取样得到特征映射,以某种方式,用更少的值代替。其主要功能是降低特征图尺寸、增大卷积核的感受野、保留显著特征和增强网络对于图像中小位移和畸变的鲁棒性。其主要参数有下采样的窗口大小(2×2、3×3 等)、滑动步长(1、2 等)、下采样方式(平均池化、最大值池化等)等。

3)全连接层。与卷积层中限定窗口大小以及共用权值的特征不同,全连接层的每一个输出是所有输入的加权和,即每一个输出均与所有的输入有连接,故其参数会比卷积层多,然而资源消耗却比卷积层低。其维度根据前后的卷积层等结

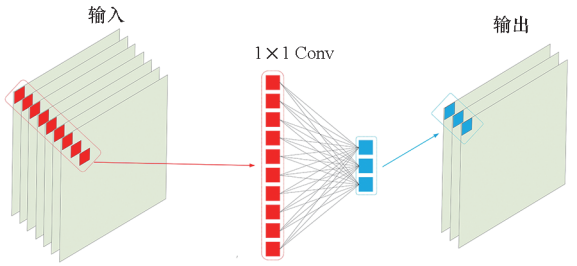
构的维度而确定。其主要参数是神经元数目，又称隐层节点数。一般来说神经元数目越多，模型越大，网络速度越慢，模型拟合能力越好，过拟合的风险也越大。

1.1.2　连接方式

神经网络依靠多种不同的链式结构连接上述的基本算子，组成众多不同的拓扑结构。除了顺序依次连接得到的链式结构之外，比较具有代表性的拓扑结构还有跳跃连接（skip connection）搭建的 bottleneck[24] 和 DenseNet 密集连接结构[25]、平行连接（parallel connectivity）搭建的 Inception 模块[23] 等多分支结构。

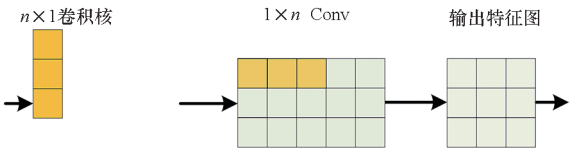
1.2　典型结构

随着深度神经网络技术的不断发展，人工设计网络也不断挖掘出具有指导性意见的领域知识，以供后续研究人员参考，如图 3 所示。如果需要改变特征映射的通道数，可以使用 1×1 卷积[23,50-51]，也称为 bottleneck 层，其通过一系列的线性加权实现升维、降维与跨通道的信息交互。同时可以在不损害模型表达能力的情况下，大大减少参数量与计算量，并且在不损害分辨率的前提下，利用后接的非线性激活函数，大幅增加非线性拟合能力[52-53]。如果需要设计轻量级网络，则可以考虑非对称卷积（asymmetric convolution），在保证感受野与普通卷积一致的前提下，大幅减少网络的参数量。以原本 $n\times n$ 卷积分解成 $1\times n$ 和 $n\times1$ 卷积为例，其参数量为原来的 $(1\times n+n\times1)/(n\times n)=2\times n/n^2=2/n$。类似的轻量级设计的结构还有深度可分离卷积[50]（depthwise separable convolution）。在较深的网络结构中往往采用跳跃连接[24] 以减轻梯度消失问题，使得更深的网络训练难度降低。其具体操作是将当前层的输入与多级之前的某层的输出相连，为跨级的前后层创建捷径，保证网络中层与层之间最大的信息流动和特征复用。如果需要图像在不同尺度上的信息，则可以考虑类似 Inception[23] 的结构，其并行连接不同尺度的多个卷积核并融合。其就是利用 1×1 卷积降低特征映射的维度，减少计算量；诸如 Inception 和 Xception[54] 等工作提出用基本子结构单元 cell 代替普通卷积来堆叠组成网络结构，这类 cell 基本单元也是由普通卷积组成，并且和普通卷积一样发挥卷积特征提取的功能，往往通过设计可以用更少的计算开销和参数实现更强大的特征表示能力。



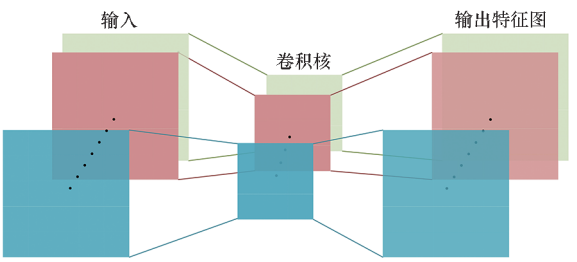
（a）1×1 卷积

（a）1×1 convolution



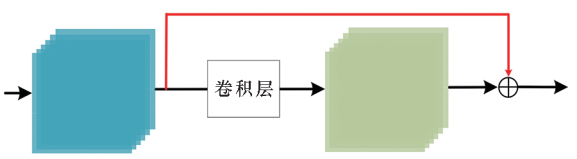
（b）$1\times n$ 和 $n\times1$ 卷积

（b）$1\times n$ & $n\times1$ convolution



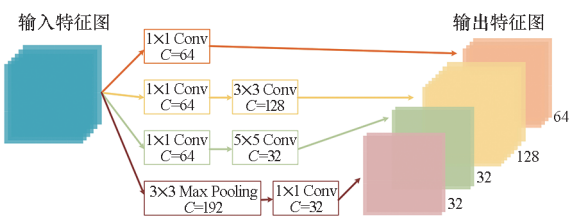
（c）深度可分离卷积

（c）Depthwise separable convolution



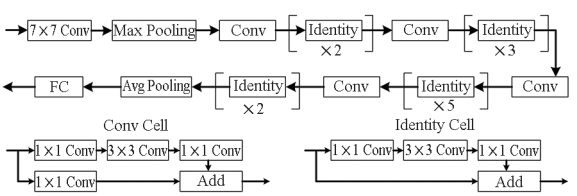
（d）跳跃连接

（d）Skip connection



（e）Inception 结构

（e）Inception architecture



（f）基本单元搭建网络

（f）Basic units building network

图 3　CNN 部分领域知识示意图

Fig. 3　Illustration of domain knowledge sample in CNN

除了结构设计上的专家经验知识之外，深度神经网络领域的其他方法也值得 NAS 借鉴。譬如，利用在源域上得到知识来快速提升在目标任务上性能的迁移学习方法，其具体算法是利用已有的预训练模型，在更大规模的数据集上训练好权重和参数的神经网络，然后在目标数据集上微调。其底层的原理是 CNN 一开始都会聚焦在检测类似曲线和边缘的特征，即底层的特征大部分是通用的，之后再是独特的更高层、更抽象特征的学习训练。其缓解了从头开始训练深度神经网络的大量计算资源消耗问题，并保证了在短时间内训练得到满意的效果，即更快更简单地使用较少训练图像快速将已学习的特征迁移到新任务；在网络设计过程中提出的网络态射（network morphism）[55] 概念，又称网络渐变，其定义是在神经网络修改过程中，一前一后有父子两个结构，如果两者功能相同，但是结构不同则互为渐变。基本思想是基于相似并且已经充分训练的网络结构的权值来初始化新生成网络（包括增删层数、改变通道数目、卷积核尺寸、增删子网络等操作）的权重，以大幅节省训练时间，并使两者具有相似的准确性。

以 ResNet-18 为例，简要阐释 NAS 工作中所需要搜索的主要目标：网络层数、层类型、层连接方式以及层形状，包括卷积核尺寸、通道数和卷积核类型等网络结构参数，如表 1 所示。ResNet-18 共有 18 个带有参数的层，包括 17 个卷积层以及 1 个全连接层。此外，还有 2 个池化层，以及若干激活函数和正则化函数。ResNet-18 层与层之间的连接不仅有最基础的序列连续连接，还有用于解决梯度消失的跨越多层的跳跃连接，并且连接的方式也不尽相同。特殊的连接方式还有 GoogleNet 中 Inception 模块中的平行连接，其平行连接不同尺寸的过滤器以处理在不同尺度上的信息。至于层内形状，ResNet-18 中卷积层的所有过滤器的尺寸多为 3×3，第一个卷积层 Conv1 的尺寸为 7×7，而通道数随着网络的深入按 64、128、256、512 增加。因为 ImageNet 的类别数为 1 000，最后的分类器 FC1 的维度为 1 000；池化层 Pool1 为窗口大小为 3×3 的最大池化（max pooling），Pool2 为全局的平均池化。而为了降低参数的数目，ResNet-50 以及更多层的 ResNet 系列网络中（101，152）频繁地使用 1×1 尺寸的卷积核。

基于上述深度神经网络的基本要素以及典型网络结构的介绍，NAS 可具体定义为：在给定的任务目标中，设计恰当的搜索空间，借助网络结构或算法自动化地搜索网络结构参数，并且不断地评估参数在网络中的性能，搭建满足条件的深度网络模型，以寻找出当前场景下性能最优的网络结构，以应对当下越发复杂、越发严苛的模型结构要求，旨在设计人工难以设计乃至解释的网络结构，降低对于神经网络使用者专业知识与设计经验的要求，以大众化深度学习技术，并节省研究人员大量的时间和精力。

表 1 搜索目标汇总

Tab. 1 Illustration of search object

搜索目标	选项
网络层数	18，50，101，152……
层类型	全连接层，卷积层，池化层，激活函数，正则化函数……
层连接方式	序列连接，跳跃连接，平行连接……
卷积核尺寸	1×1，3×3，5×5，7×7……
卷积核通道数	3，10，24，224，256，512……

2 NAS 问题描述与挑战

2.1 NAS 问题描述

因为由网络结构参数多样性导致的定义网络结构的参数量一般较大，神经网络架构搜索，或者说网络结构搜索，本质上可以看作是一个高维空间的最优参数搜索问题。具体来说，如图 4 所示，该搜索问题可以分解为三个子问题：设计搜索空间、确定搜索策略和性能评估策略。

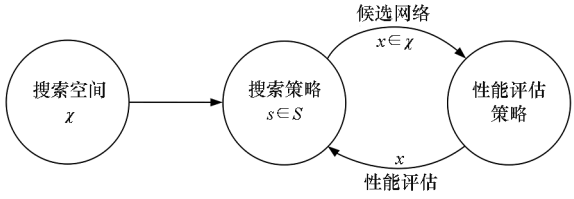


图 4 神经网络架构搜索技术的一般流程

Fig. 4 General work flow of neural architecture search

2.1.1 搜索空间

搜索空间指的是搜索中候选的神经网络结构集合，定义了优化问题的变量，即解的空间，旨在从中选择最优的结构。构成搜索空间的候选网络结构的质量和数量均能对后面的搜索过程产生影响。高质量的搜索空间能让后续的搜索算法更加容易筛选出高性能的网络结构，而相对较小的搜索空间则能减轻搜索的负担。因此，搜索空间的设计是神经网络结构搜索领域的重要一环。

2.1.2　搜索策略

搜索策略是指在搜索空间中对其候选网络结构的选择标准,比如最简单的随机搜索策略,就是从搜索空间中随机选择出一定数量的网络结构,通过后续的评估方法逐一评估筛选出其中性能最优的一个。典型的搜索策略还包括基于强化学习的搜索,基于进化理论的搜索和基于梯度的搜索等。一套好的搜索策略能从庞大的搜索空间中通过较少的迭代高效准确地定位出性能优越的网络结构,因此对搜索策略的研究也是必不可少的。

2.1.3　性能评估策略

性能评估策略则是将取样的网络结构进行一定的评估,指标包括精度、速度等,并且反馈必要的信息以指导后续的搜索。快速的网络性能评估能够缩短迭代的周期,从而提高搜索的效率。

2.2　NAS 难点与挑战

现有的神经网络架构搜索方法主要是对网络结构参数的搜索,存在以下几点挑战问题:

1)离散问题。在超参数优化领域,优化参数、正则化参数优化和神经网络架构搜索具有一定的相似性,但是后者难度和情形更加复杂。两相比较,神经网络架构搜索包含了对网络参数的搜索,即层与层之间的交互方式、卷积核数量和卷积核尺寸、网络层数和激活函数等。网络参数描述神经网络结构的参数,含有诸如拓扑结构、层类型、层内的离散型超参数等离散数据。上述问题导致网络结构的维度高,离散并且相互依赖,进而导致自动调优的难度也随之增大。

2)黑箱优化问题。同深度学习一样,神经网络架构搜索也是一种解释性较弱的黑箱问题,其需要在训练过程中评估网络模型的性能,评价函数没有统一的标准,算法不知道优化目标函数的具体形式,并且是用代理的数据对模型进行评价,再将模型转移到目标数据之上。目前来说,理论的前进晚于实践,急需理论上可解释性的进一步研究,以方便领域进一步的推广。

3)局部最优问题。目前来看,神经网络架构搜索的搜索策略机理只能保证模型是局部最优的,而不能保证是全局最优的。对于寻求全局最优模型的目标来说,还是具有一定的改进空间。

4)高计算复杂度问题。当前,基于强化学习和进化算法的一次优化结果的评价仍然十分耗时,大型的深度学习模型参数数以亿计,运行一次结果需要几周时间。以速度优化后的强化学习工作[35]为例,其花费的时间在奠基性工作[33]的基础上加速 7 倍后,仍需要超过 2 000 个 GPU 日;对于目前在 ImageNet 数据集上图像分类任务获得最高 TOP-1 准确率的 AmoebaNet 结构[36],其搜索也需要约 3 150 个 GPU 日。

基于强化学习和进化算法的搜索策略的资源消耗还是巨大,例如不断加速改进过的算法[56]仍需要在 GeForce GTX 1080 GPU 上运行 10 d,以完成小数据集上的搜索任务。此外,研究需求从只考虑精度的单目标,到综合考虑功耗、时延、计算复杂度、内存占用等指标的多目标,其研究难度势必增加不少。

除了上述的难点和挑战外,多目标优化、难以复现等因素对于神经网络架构搜索也有很大的影响。

3　神经网络架构搜索研究进展

本节将从 NAS 主要涉及的搜索空间、搜索策略和性能评估策略三个主要研究内容的研究现状进行详细的阐述与分析。

3.1　搜索空间

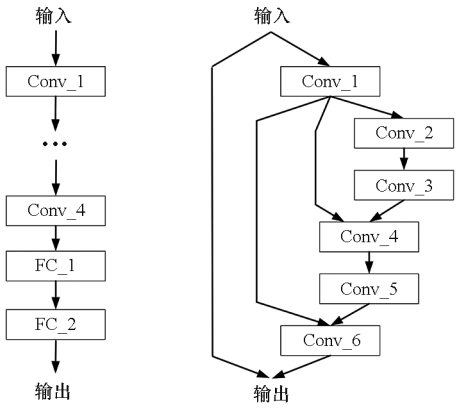
搜索空间指的是待选的可行网络模型集合,网络设计涉及的各个结构参数都可以视为搜索的对象。比如在卷积神经网络结构搜索中,搜索对象可以包括卷积核的大小、卷积操作的类型、卷积层的宽度、网络的深度、层运算类型、层与层之间的连接方式等。

伴随着搜索空间的不断发掘,搜索空间的选择很大程度上决定了优化问题的难度,搜索空间的巧妙设计可以加速优化整个搜索过程[44],因此部分研究人员认为搜索空间是整个网络搜索技术的核心:一个好的搜索空间就可以基本决定好的搜索结果[39,57-58]。例如,引入适合具体任务的先验知识能够减少搜索空间并简化搜索。需要指出的是,搜索空间的设计也从一定程度上引入了人为的臆断和偏差,限定了神经网络中某些层次上的拓扑结构,对找到新颖的、超越当前人类认识的新的结构会具有一定的妨碍。

3.1.1　常见的网络结构

基于搜索空间的重要意义,准确把握常用网络的结构就显得十分重要。大体看来,现阶段的神经网络结构可以分为链式(chain-like)结构[22]和多分支(multi-branch)结构[23-24],如图 5 所示,所有的网络结构都可以抽象成一个无孤立节点的有向无环图(directed acyclic graph, DAG),节点表示表征神经网络的层,边代表数据的流动。

链式结构多出现在传统的神经网络模型中,反映在有向无环图中,即一个节点的前驱节点只



(a) 链式结构 (b) 多分支结构
(a) Chain-like structure (b) Multi-branch structure

图5 链式结构与多分支结构示意图

Fig. 5 Illustration of chain-like and multi-branch structure

有一个,并且直接以前驱节点的输出值作为本节点的输入。对于该类架构的搜索参数大致上包含以下三点:

1)网络的最大层数。

2)每层执行的操作。除了第一层必须为输入层,最后一层必须是输出层外,可以是池化(pooling)、卷积(convolution)、激活(activation)这种基础操作,也可以是深度可分离卷积(depthwise separable convolution)、空洞卷积(dilated convolution)、反卷积(transposed convolution)等高级的卷积方式。

3)每层操作相关的超参数。例如对于普通的卷积操作就有卷积核数目(filter number)、核尺寸(kernel size)、计算步长(strides)等,又如空洞卷积中的扩张率(dilation rate)等。因为这一项取决于上一项操作类型的确定值,且层数是不一致的,故超参数选项是不一样的,也就导致候选网络表示的变长问题[33]。

当前研究[24,39]已经证明垂直的链式结构容易造成空间弥散,不利于梯度的传递以及网络向更深层次发展,于是多分支结构被提出以及广泛使用。多分支结构[23-24]是近年来随着深度神经网络兴起的架构,反映在有向无环图中,即有多个前驱节点,需要将前驱节点的值汇总后输入本节点,对应的汇总策略有相加和拼接,主要代表分别是引入跳跃连接的 ResNet 和引入密集连接(dense connectivity)的 DenseNet。跳跃连接是将当前层的输入与残差分支的输出相加,为前后层创建捷径,保证网络中层与层之间最大的信息流动。即

$$g_i(L_{i-1}^{\text{out}},\cdots,L_0^{\text{out}}) = L_{i-1}^{\text{out}} + L_j^{\text{out}}, j < i-1 \quad (1)$$

密集连接是将每一层与后面的多层相连接,因此后面层会获取前面多层的特征图。即

$$g_i(L_{i-1}^{\text{out}},\cdots,L_0^{\text{out}}) = concat(L_{i-1}^{\text{out}},\cdots,L_0^{\text{out}}) \quad (2)$$

因为不再像链式结构一样线性地增长,多分支结构需要搜索的参数第一项由网络的最大层数变成了网络的拓扑结构,包括网络的层数以及层的连接关系。第二、第三项与链式结构一致。多分支结构通过引入密集连接和跳跃连接鼓励特征复用以加强特征传播,在减少参数数量的同时解决了梯度消失问题,从而使更深的网络成为可能,并且多分支架构空间已经被证实可以实现更高精度的效果。但是与之而来的代价是深度加深和复杂拓扑结构所带来的指数级别增长的搜索空间。

3.1.2 基于基本单元的搜索空间

随着对网络搜索技术这一多维优化问题的研究深入,研究人员发现直接在全局搜索空间中对整个网络进行建模搜索存在计算成本的问题。其主要原因是现代神经网络通常由数百个卷积层组成,每个卷积层的类型和超参数又拥有众多选项,特别是当数据集较大时便形成了巨大的搜索空间,引起繁重的计算成本。同时,随着网络层数的加深和基本算子的增多,搜索空间会以指数形式增多,没有上限,甚至候选网络结构的数量趋于无穷。例如,当网络的层数达到 20 时,搜索空间则会包含超过10^{36}个不同的候选网络结构。而且通常设计的网络结构受限于特定的数据集或任务,很难转移到其他任务、设备,或者推广到具有不同输入数据大小的另一个数据集。考虑到神经网络架构搜索的高维度、连续和离散混合等诸多难点,如果可以实现搜索空间的降维,那么将会大大提升模型算法效果。

研究人员通过仔细观察发现很多具有代表性的成功多分支结构,如 GoogleNet、Xception 等,往往会包含重复并且有序排列的基本子结构单元,被称为“cell”,如图 6 所示。

为了提升搜索速度,限制搜索空间的大小,研究者转而搜索 cell 这类基本子结构单元,而不是直接搜索整个网络结构,实现了搜索空间与网络深度解耦。一般来说,一个 cell 中会包含若干个 block,而一个 block 的内部结构在整个搜索过程中会遵循预先设定的规则,一般来说会有 5 个离散的待搜索参数,即作为输入的 2 个隐状态、分别施加在 2 个输入上的 2 个基本操作以及 1 个组合上述操作运算结果的合并操作,以使子单元输出一个结果。该思想最先在 Zoph 等[35]提出的 NASNet 搜索空间得以展现,将整体的网络结构由搜索得到的基础单元 cell 重复构建,每个 cell 单

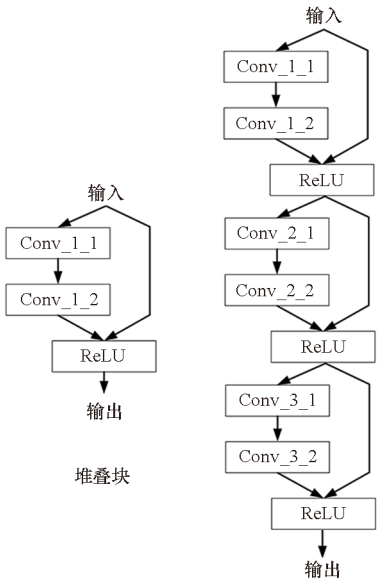


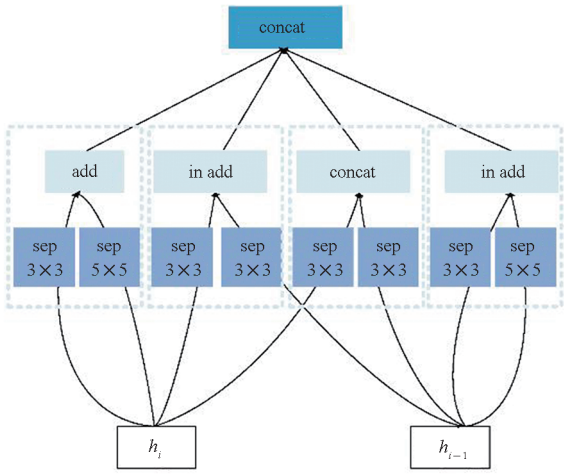
图 6　重复堆叠网络示意图

Fig. 6　Illustration of cell-based network

元都代表着某种特征转换,并且可以由一个小型有向无环图表示。Zoph 等设计的基本单元主要分为标准单元(normal cell)和还原单元(reduction cell)两类。如图 7 所示,基本操作中的“max”表示最大值池化操作,“sep”表示深度可分离卷积,“1×7 7×1”表示某种非对称卷积核操作,即先做 1×7 卷积操作,再做 7×1 卷积操作;合并操作可以是应用于 ResNet 等工作中的两个特征映射的逐位相加,也可以是应用于 DenseNet 等的通道数合并或级联。标准单元的卷积操作不改变特征映射的大小,还原单元的卷积操作则通过设定步长为 2,将输入特征映射的长宽各减少为原来的一半。最终的网络结构则由标准单元和还原单元组成。在 cell 单元的内部,每个子单元 block 的输入可以是单元内部的其他子单元的输出 h_i,也可以是外部其他单元的输出 h_{i-1} 以实现单元间的跳跃连接。

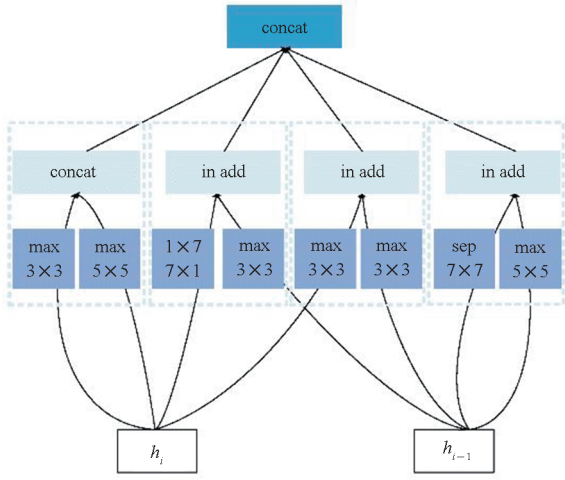
由此,搜索空间缩小到对两类基础单元结构的搜索,实验结果显示比文献[33]的工作提速 7 倍之多。一旦将搜索的目标转化为重复堆叠网络结构(cell-based network)中的基本单元,一方面可以减少优化变量数目,另一方面可以使相同的基本单元在不同任务之间进行迁移[35]。

Zhong 等[41]同时期提出的 BlockQNN 也是基于基础单元(block)堆叠而成的。通过设计网络结构代码(network structure code, NSC)来用五维向量表征每个基本单元,向量中每一位分别表示层索引、操作算子类型、核尺寸、上级操作(predecessor parameter)。与工作[35]的区别在于



(a) 标准单元

(a) Normal cell



(b) 还原单元

(b) Reduction cell

图 7　标准单元与还原单元示意图

Fig. 7　Illustration of normal cell and reduction cell

其不是通过专门搜索的还原单元实现维度缩减和数据降维,而是在基本单元之间引入池化操作;并且单元的定义更加松弛,即每个单元中操作的数目并不是固定的,且前向连接可为 $l-2$ 个。

Liu 等[59]进一步提升搜索空间的可拓展性,将搜索空间层次化表示。其将搜索空间划分为 L 层,第一层即为文献[35]中提到的基本操作;最后一层即为最终搜索得到的网络结构;l 层中的待搜索结构即基于 $l-1$ 层中的结构组成。Zoph 等的工作[35]可视为该工作中 $L=3$ 的一个特例。

PNAS[37]沿用文献[39,60]中设计代理模型(surrogate function)来实现自动化对候选网络进行评估,并且使得设计的网络不再局限于评估固定尺寸的候选网络。其在搜索空间的设计聚焦于逐步提升基本子单元中的 block 数目,即 cell 中特

征映射的数量。以实现快速训练代理模型,并且从易到难的网络结构搜索。

之后的很多研究工作,如文献[37 – 38,40,45,61],都是基于文献[35]做一些操作选择和单元堆叠策略的简单更改。不过,Tan 等[43]则认为基于少数复杂基本单元堆叠而成的网络结构[35]阻碍了结构的多样性发展,不利于保证搜索网络的性能。Tan 等[43]主张将模型分解为若干个(>2)的模块在局部的重复堆叠,并且分别搜索每个模块内的操作和连接,允许在不同的模块内出现不同的结构。

基于基本单元的搜索空间的提出也引发了新的问题:如何选择基本单元之上的宏观结构,即需要使用多少基本单元,以及单元之间如何连接组成网络。Liu 等[45]提出的分层搜索空间或者层次化搜索空间,就是针对基本单元的数量和基本单元之间的连接方式的设计问题提出的,其在不同的层面上设计了“基本单元”(motif):最底层是原始操作(meta operation)集合,包括卷积、池化等;中间层的基本单元是通过连接原始操作的有向无环图,类似于文献[35]中提出的基本单元概念;最高层则专注于宏观结构,即以某种方式连接中间层中的基本单元。该工作增强搜索空间的表达性,使得多种搜索策略都能取得期望的性能。

总而言之,与全局搜索空间相比,基于单元重复堆叠而设计的搜索空间拥有显著减小的规模,导致搜索的加速。并且仅仅通过调整一个模型中所使用单元的数量,就可以很容易地应用到其他的数据集、环境或者任务中。缺点是搜索空间自由度稍许降低,如不能修改内部卷积核的数量。该思想也可以继续迁移到残差块(residual blocks)或者循环神经网络中堆叠长短期记忆网络的结构搜索。

3.1.3 其他典型的搜索空间

搜索空间的设计也是依据具体的搜索策略作考量的,以下是在一些典型的网络结构搜索工作中出现的搜索空间。

1)过连接搜索空间。除了基本单元这种从小单元出发的搜索空间,也有为了适应特殊的搜索策略,从大结构出发,把整个搜索空间建模为一张包含众多候选结构的过连接的大型网络[38,40]。在搜索过程中学习连接的重要程度,并据此删减冗余连接,以得到期望的结构。

2)连续搜索空间。一般而言,网络结构的参数都是离散的。然而文献[38,62 – 63]将搜索空间连续化,使得目标空间可微,以期基于梯度信息有效地搜索。Liu 等[38]在搜索过程中将每处操作表征为所有候选操作的 Softmax 函数值和,即混合操作,于是整个搜索任务转换为优化由众多连续值组成的集合,即实现将离散搜索空间松弛到连续搜索空间。

3)非结构化搜索空间。该类型一般会出现在剪枝类的搜索策略[64 – 65]中,早期细粒度(fine-grained)、连接层面的剪枝研究[66 – 67]聚焦于卷积核内部冗余的连接,强制卷积权值或者特征映射稀疏化,而不改变整个网络的结构。该搜索空间如图 8 所示,高度个性化定制网络结构带来的缺点就是 GPU 处理不规则的权值速度慢,可能需要特殊的硬件来保证网络前向传播的速度。与之相对应的结构化搜索空间多出现在稀疏结构学习(sparse structure learning)和通道剪枝(channel pruning)[68 – 74]中,会把剪枝的重心放在卷积层中的卷积核尺寸、通道数或者整个卷积层等结构化信息,并且剪枝之后的结构整齐,可以方便地拓展和应用。

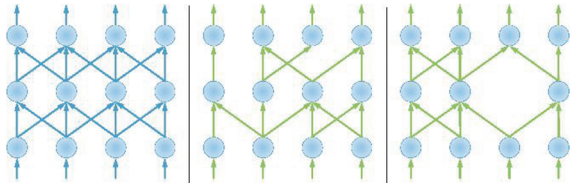


图 8 非结构化搜索空间示意图

Fig. 8 Illustration of non-structured search space

3.2 搜索策略

搜索策略一方面要快速找到性能良好的框架,另一方面也要避免过早地因为次优结构而停止搜索,好的搜索策略就是可以在以上准确和速度两点中取到平衡。下面依次介绍主流的基于强化学习、基于进化算法、基于梯度的搜索策略以及其他搜索策略。

3.2.1 基于强化学习的搜索策略

基于强化学习理论的实现[33]是神经网络架构搜索技术的奠基工作,这项工作给神经网络架构搜索技术带来发展契机,并且逐渐发展成热点问题,吸引众多学者和机构的研究。如图 9 所示,强化学习基于环境而应对并作出调整,以取得最大化的预期利益,其架构可以抽象成一个智能体选择动作,并且通过环境反馈的激励来不断优化智能体。在大多数情况下,强化学习近似于一种范式,只要提炼出强化学习的四要素,即智能体、动作、环境和激励,问题就可以用强化学习求

解,故网络结构搜索问题也可以通过强化学习算法解决。

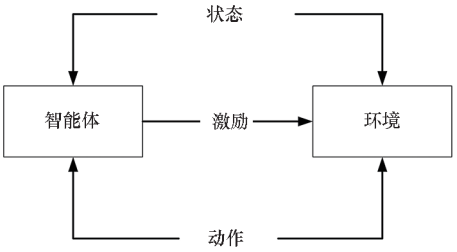


图 9　强化学习整体框架示意图

Fig. 9　Illustration of the overall framework of reinforcement learning

基于强化学习的搜索策略将候选结构的采样生成过程建模成马尔可夫决策过程,将候选网络的验证准确率作为激励,而候选网络采样生成过程的更新则通过强化算法实现,包括了 Q 学习[41,75]、基于策略的 REINFORCE[33] 与近端策略优化(proximal policy optimization, PPO)算法[35,43]等。

随着强化学习在神经网络架构搜索领域的应用不断巩固,该项技术也不断拓展应用于其他纵深领域:He 等[64]设计不同的激励函数来满足不同的搜索约束,并且将智能体的动作定义为指定每个卷积层的压缩率,达到自动化压缩网络模型的效果;Tan 等[43]综合准确率和计算时延组成多目标激励信号。

$$\max_m ACC(m) \times \left[\frac{LAT(m)}{T}\right]^{-0.07} \tag{3}$$

Wang 等[76]则在考虑计算时延的前提下制定网络量化策略,针对性地应对不同层的不同冗余情况,为网络的每一层搜索不同的权值位宽(bit),搜索得到量化后的混合精度神经网络模型;基于在图像分类和特征提取的优秀性能,Ghiasi 等[44]将基于强化学习的神经网络架构搜索技术应用于物体检测领域,提出 NAS-FPN 方法以搜索物体检测框架 RetinaNet 中用于多尺度检测的特征金字塔网络(feature pyramid network, FPN),并将激励信号设定为模型在验证集上采样结构的平均准确率(average precision, AP);Cubuk 等[77]则将技术迁移到同样是离散搜索问题的数据增强领域,提出 AutoAugment 用以实现自动化的数据增强策略制定。因为作为激励信号的验证集准确率仍然是一个不可微的值,控制器仍然是通过策略梯度算法更新。

3.2.2　基于进化算法的搜索策略

进化算法产生的灵感来自大自然的生物进化,是一种成熟的具有高鲁棒性和广泛适用性的全局优化方法。具有自组织、自适应、自学习的特点,能够不受问题性质的限制。进化算法整体框架如图 10 所示。

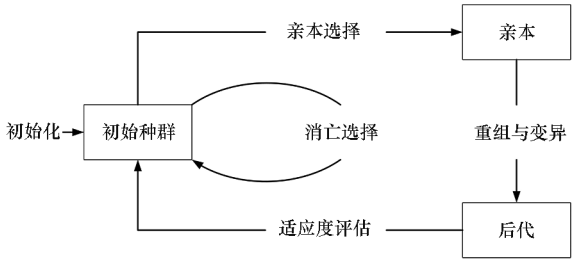


图 10　进化算法整体框架示意图

Fig. 10　Illustration of the overall framework of evaluation algorithm

首先随机生成一个种群,即 N 组解,又称为一个模型簇,在网络搜索问题中即为 N 个候选网络模型。之后开始循环以下的步骤:选择、交叉、变异,直到满足最终条件。在进化过程中使用得比较广泛的是由 Real 等[78]提出的联赛选择(tournament selection)方法,即随机从模型簇中选出两个个体模型,根据优胜劣汰识别模型,立刻从模型簇中剔除相对不适合的模型,即表示在此次进化中的消亡。更优的模型成为母体进行繁殖,即在母体的基础上变异发展形成新的模型簇,再重复之前的过程,直至特定轮次或指定性能。具体地,插入新层、修改参数、引入跳跃连接等视为突变,更多的变异类目下的选项可以参照表 2。

表 2　进化算法中变异的选项示例

Tab. 2　Illustration of mutation options in evaluation algorithm

选项
改变学习率
模型保持不变,但是继续训练一段时间
重置所有参数权值
插入卷积层(随机决定是否加 Batch-Normalization 和 ReLU)
移除卷积层
改变卷积层的步长
随机改变卷积层的通道数
加入跳跃连接层

在文献[78]的基础上,Real 等[36]想通过限制搜索选择来减少搜索空间,以提高算法可控性,于是通过限定群体的初始化模型必须遵从专家规则,譬如约束基本单元的外部堆叠模式,即不再从简单的模型开始进化,以去除搜索空间中所有可

能导致大型误差的结构,并且更易获得更高质量的模型。与此同时,作者提出的 AmoenaNet-A 修改了之前的联赛选择算法,通过引入一个“年龄”属性使得搜索更偏向于“年轻的基因型”(新产生的网络结构)。具体的实现方法是在每轮中选定准确率最高的作为母体产生子集,但是去除“年龄最大的”网络结构,而不是性能最差的网络结构。此方法有利于抵抗训练噪声,即避免某些模型可能仅仅因为运气好而达到了较高的准确率的情况。在文献[78]中,这类充当“噪声”的性能较差网络结构可能存在于整个进化过程中,留下大量后代,使得搜索空间不能得到充分的探索,得到的结构缺乏多样性,类比到遗传学上称之为“近亲繁殖”。作者还比较了进化算法、强化学习与随机搜索在执行网络结构搜索任务时的表现,实验结果显示进化算法在早期阶段搜索速度更快,适用于在计算资源有限的环境中不得不早停的情形,并且对数据集或者搜索空间的变化具备很强的稳定性。

后续一系列工作进一步拓展了进化算法在神经网络架构搜索领域的应用:Elsken 等[79]将网络态射技术应用到进化算法中的变异过程。Dai 等[80]将进化学习算法应用到多目标的优化任务中,满足了计算资源有限平台对于能量消耗和计算时延的约束;So 等[49]应用进化算法搜索解决序列问题,并且以目前性能最优且善于解决长期依赖问题的完全注意力网络 Transformer[81-82]作为初始种群,而不是使用完全随机模型对搜索进行初始化。实验结果显示在英德翻译、英法翻译等四个语言任务中表现均优于原版的特征处理器 Transformer;Liu 等在层次化的搜索空间上运行进化算法。不同于链式的连接方式,层次结构将前一步骤生成的单元结构作为下一步单元结构的基本组成部件。最底层为基本操作组件(如卷积、池化等),中间层为操作组件构成的单元模块,最高层是由单元模块堆叠而成的整体网络。搜索中仍旧采用联赛选择方法:每次选出当前种群中5%的个体,选择在验证集上的准确率最高的个体进行变异操作,产生新个体。但是整个搜索过程中不会从模型簇中剔除任何的结构,随着算法的进行,种群的规模会不断增大。实验结果显示在 CIFAR-10 数据集上取得了进化算法中的最佳模型结构。

总的来说,基于进化算法的搜索策略是一种无梯度的优化算法,优点是没有预设搜索的范围,鼓励在更大的搜索空间中搜索。因为一直保持一定规模的候选网络作为候选的变异母体,基于进化算法的搜索策略不会过早地聚焦于已搜索到的优秀模型。与强化学习相比,因为具有更少的元参数,形式更加简洁,且不需要训练本身具有很多权重的智能体或者控制器,在早期阶段搜索速度更快。与随机搜索相比,进化算法的优势是只有好的模型才会被选择进行变异,以进行探索,故进化算法也可以视为带有选择的随机搜索。缺点是当种群数量较小的时候,会陷入较差的局部最优。并且仍然会占用较大的计算资源,譬如一般需要初始种群数的 1/4 数量的机器实现分布式计算[78]。

3.2.3 基于梯度的搜索策略

基于强化学习和进化算法的策略本质上还是把目标函数看作黑盒,并在离散空间中搜索。这一做法导致需要独立地评价大量的结构,使得计算资源消耗巨大。为了解决计算成本上的问题,基于梯度的搜索策略应运而生,其主要想法是如果搜索空间连续,目标函数可微,那么基于梯度信息传递可以更有效地搜索网络结构。

Liu 等[38]首次将梯度方法应用到神经网络架构搜索技术中,提出 DARTS 方法。DARTS 方法沿用基于 cell 的搜索空间,将网络结构视为由若干个基础构建模块(基本计算单元)组成,而基本计算单元建模成一个由 N 个有序节点组成的有向非循环图,如图 11 所示,节点 $\boldsymbol{x}^{(i)}$ 表征特征映射,而连接的有向边(i,j)则表示某一操作 $o^{(i,j)}$ 用于转换 $\boldsymbol{x}^{(i)}$,例如不同尺寸(3×3、5×5、7×7)和不同类型(空洞卷积、深度可分离卷积、空间可分卷积)的卷积、不同类型的池化(平均池化、最大池化)操作、表示节点之间直连的操作以及表示节点之间不存在连接的零操作等。初始时每一个操作对应的用于表征保留概率的权重参数是相等的,且求和为1,并且该参数在训练过程中会不断更新,确定最终基本单元的连接和具体结构也由权重参数决定,故基本单元的搜索就被简化为学习每条有向边上候选操作的权重。

具体将搜索过程连续化的方法是借助 Softmax 函数所实现。用维数为$|O|$的向量 $\boldsymbol{\alpha}^{(i,j)}$ 存储节点对有向边(i,j)的所有候选操作的参数化权重,将所有候选操作使用 Softmax 函数按照权重进行混合,混合后的结果即作为该处待搜索操作的表征,如式(4)所示。

$$\bar{o}^{(i,j)}(\boldsymbol{x}) = \sum_{o\in O}\frac{\exp(\boldsymbol{\alpha}_o^{(i,j)})}{\sum_{o'\in O}\exp(\boldsymbol{\alpha}_{o'}^{(i,j)})}o(\boldsymbol{x}) \quad (4)$$

即某处操作(连接)均可以被该处的所有候

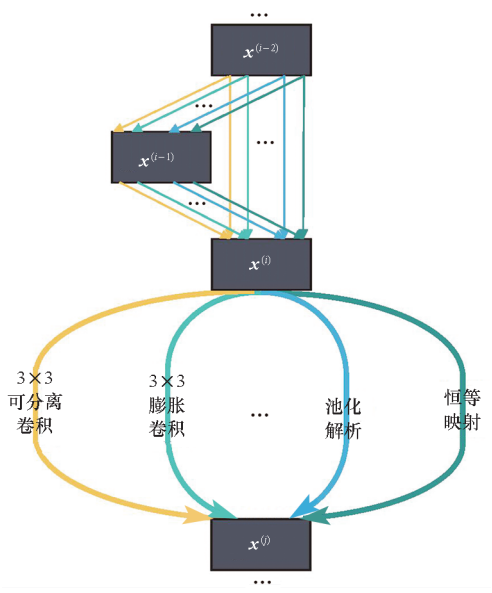


图 11　DARTS 方法中基本单元内搜索空间示意图

Fig. 11　Illustration of search space of basic unit in DARTS

选操作所表征，每条有向边视为混合的操作集，且权重保留在计算图中，以便计算验证集损失时可以后向传播计算权重的梯度。搜索空间变成由连续变量 $\boldsymbol{\alpha}=\{\boldsymbol{\alpha}^{(i,j)}\}$ 表征的连续空间。在搜索过程结束之后，根据

$$o^{(i,j)}=\arg\max_{o\in O}\boldsymbol{\alpha}_o^{(i,j)} \tag{5}$$

将权重最大的候选操作替换搜索过程中的混合操作表征，得到离散的网络结构。整个网络搜索问题是双层优化问题，即对编码模型结构的连续变量 $\boldsymbol{\alpha}$ 以及权重参数 $\boldsymbol{w}$ 的学习。学习过程中交替搜索结构参数 $\boldsymbol{\alpha}$ 和训练权值参数 $\boldsymbol{w}$，以标准梯度下降法迭代优化交替更新两个参数：训练阶段将数据分为训练集和验证集两部分，将验证集上的性能看作是拟合程度，用来优化验证集上的损失值 L_{val}，找到使 L_{val} 更小的网络结构 $\boldsymbol{\alpha}'$；再将训练集上的性能看作是激励信号，搜索内部权值 $\boldsymbol{w}'$ 使损失值 L_{train} 更低。收敛后即得到 $\boldsymbol{\alpha}^*$ 和 $\boldsymbol{w}^*=\arg\min_w L_{\text{train}}(\boldsymbol{w},\boldsymbol{\alpha}^*)$。搜索结束之后再通过式(5)离散化形成最终离散域中的结果网络。训练之后的计算单元可以通过堆叠构成卷积神经网络，或者通过循环连接构成循环网络。进而从头开始训练，并测试在测试集上的表现。

相比于早期的连续域的结构搜索方法，DARTS 方法不像先前工作需要在计算过程中显式地实例化所有候选的网络结构，而是蕴含于最初的超图(SuperNet)中，以节省计算内存；同时不受限于特定结构，可以用于搜索卷积和循环神经网络。实验结果也显示 DARTS 方法搜索一次仅需要一个算力较强的 GPU(如 1080Ti)运行一天即可得到具有竞争力的网络结构。不过缺点是搜索的灵活度较差，DARTS 方法中已经固定最终的网络整体由 8 个基础建造模块按顺序连接而成，而基础建造模块也限定为两类，且在序列中的位置也无法搜索，而是人为指定；在基础建造模块内部中节点的数目也是人为指定的。除此之外，DARTS 方法比较占用 GPU 运存(RAM)，若每处连接有 K 个候选操作，则运存占用就会相应达到原本训练网络的 K 倍之多；尽管 DARTS 方法搜索结构的效率快，但是并不能保证生成的为性能最优的结构，重复多次搜索的结果才更具有说服力。不过 DARTS 仍因为其平民化和高效的特点，是近年来热度最高的搜索策略，很多新工作都是基于 DARTS 而完成的。

受 DARTS 方法的启发，Luo 等[83]设计一个编码器模型将网络结构映射到一个连续的向量空间上使其更紧凑和平滑地表示，并在连续空间上建立一个用于性能预测的回归模型来逼近结构的最终性能，利用梯度方法在该连续空间进行优化，之后以配备了注意力机制的 LSTM 模型作为解码器以精准恢复出网络结构，最终完成结构搜索任务；Xie 等[84]为了能够更好地利用损失值以及其中所蕴含的梯度信息，用一组服从完全可分解联合分布的 one hot 随机变量表示搜索空间，变量相乘作为掩码来选择每条边的操作，以实现结构参数和神经元参数可以在同一轮后向传播中训练更新。

针对文献[38]中仍然使用固定长度编码搜索空间，即每个操作只能有两个输入且每个块中的节点数量是固定的，并且模块在网络结构中复用共享等弊端，Zhang 等[85]从一个候选操作间完全两两连接的网络出发，引入缩放因子来加权运算之间的信息流，并且作为表征保留概率的权重参数。接着加上稀疏正则化(sparse regularization)来删除构架中的某些连接，使得整个搜索过程中只有一个模型被训练和评估。其直接稀疏优化方法带来可微分、效率高的优点，可以直接应用到 ImageNet 之类的大数据集上；并且可以通过灵活地修改正则化项来实现包括浮点运算数(floating point operations，FLOPs)等搜索约束在内的多目标任务。

在 Wu 等[61]的工作中，同样是设计了一个包含所有候选操作的大型网络，并行地包含设定的候选基本单元。但是在前向传播的过程中，只有一个候选单元被采样，且取样执行是基于其 Softmax 函数处理后得到的激活概率。为了使损失值对于抽

样参数直接可微,作者直接利用 Gumbel Softmax 处理激活概率计算中的参数,从而实现到连续域的转换,进而通过梯度方法完成搜索。

Dong 等[86]指出 DARTS 方法因为每次迭代中都要更新所有参数,直接在完整的大型网络上更新太耗时,故针对表征搜索空间的有向无环图设计一个可微的采样器,每次只需优化采样到的子图,实现在训练集上优化有向无环图内每个子图的参数,在验证集上优化可微的神经网络采样器。ImageNet 上的实验结果显示,在网络准确率与 DARTS 持平的情况下,搜索时间快 5 倍以上。

Chen 等[87]提出了 DARTS 方法可能因为受限于显存会出现深度差距问题,即网络搜索技术选择的操作或基本单元在搜索阶段中作用于代理数据集上的较浅网络结构中变现好(例如应用于 CIFAR-10 上深度为 8 的网络),但是一旦拓展到真正使用在目标数据集的较深网络中则表现较差(例如通过堆叠得到应用于 ImageNet 上深度为 20 的网络)。作者参考 PNAS 对 NAS 技术的渐进式优化,提出了 PDARTS 方法,在训练过程中循序渐进地增加网络结构的深度,有效抑制了深度差距问题。针对层数增加所带来的计算量问题,根据搜索过程中的表现,动态地随着层数增加减少候选操作的数量;针对跳跃连接易于导致快速的梯度下降,进而倾向于被选中而占据结构中主导地位的问题,对搜索空间添加正则化约束,以控制跳跃连接出现的次数。

Bi 等[62]直接基于 DARTS 方法做优化,指出在运算过程中,表征结构的参数的梯度近似是制约搜索稳定性,阻碍梯度方法应用到更大搜索空间以及灵活适用到其他情况的关键因素。例如随着训练时间的增加,DARTS 及其引申方法会收敛到相似的结构,并且出现很多的跳跃连接,这种可训练参数较少的相等结构尽管在搜索阶段的代理数据集可以取得高的准确率,但是无法在大型数据集上保持性能,这也解释了为什么很多类 DARTS 方法会主张早停以防止网络性能的衰减。作者提出添加一个修正项来估计之前梯度计算中丢弃的某项,以修正误差并防止 DARTS 方法收敛到其他非最优点。

此外,Cai 等[88]指出先前的研究中使用的低保真方法,诸如训练较少轮次等,再重复堆叠同一高性能单元模块的方法得到在目标任务上的网络结构可能并不是最优的,而且依赖低保真代理(例如 FLOPs)的模型评估意味着无法在搜索过程中直接权衡延迟等硬件指标。于是作者将神经网络架构搜索与剪枝和量化这类典型的模型压缩方法结合,设计了路径归一化方法来加速网络搜索和 GPU 内存的占用,并且直接在目标任务和硬件平台上搜索。作者首先建立一个过度参数化网络,各个位置保留了所有可能的候选操作,并用 0、1 二值的结构参数表示网络中候选操作是否被激活,分别在训练集和测试集上交替学习更新权重参数和二值化的结构参数,权重参数采用标注梯度下降更新,而结构参数则引入类梯度方法或者 REINFORCE 方法进行更新。在每个结构参数更新的阶段,为了进一步提升搜索速度,网络只会选择两个候选操作的权值进行更新,其中一个被增强,另一个被削弱,而未被选择的操作的权值则保持不变。结构参数训练完毕之后,即可剔除过参数化网络中的冗余操作,最终得到一个轻量化网络结构。因为其直接在不同的设备上搜索,对比之后得到以下规律:因为 GPU 比 CPU 具有更高的并行数值计算能力,可以更好地利用大的 MBConv[50],所以特别是当特征映射较大时,CPU 倾向于选择小的 MBConv,而 GPU 偏好利用大的 MBConv,使得搜索网络整体短宽;当特征映射被下采样时,所有网络结构都倾向于选择更大的 MBConv,猜测可能是大结构有利于网络在下采样时保留更多的信息;影响网络轻量化的因素主要包括操作类型、输入输出特征映射的大小、卷积核大小和卷积操作步长等。

3.2.4 序列模型优化方法

序列参数优化(sequential parameter optimization, SPO)思想最早在 2010 年由 Hutter 等[89]提出,主张在有限的迭代轮次内,按照损失函数的期望值最小,同时方差最大的方向选择参数,即选择损失值小,并且最有可能更小的方向进行探索,寻找更优超参数,也被称之为期望改进方案(expected improvement criterion, EIC)。直到 2018 年,Liu 等[37]提出了神经网络架构搜索领域的基于序列模型优化方法。该方法是一种启发式的搜索策略,搜索的结构由简单到复杂,即按复杂度逐渐增大的顺序搜索网络。随着搜索的深入,不仅结构的复杂度在提升,每一步中的候选网络数目也在增多。该搜索策略最显著的优点是样本利用率高,训练尽可能少的网络以加速代理模型的训练以及整个搜索过程。在保证不损失性能的情况下,CIFAR-10 的搜索时间分别比典型的强化学习搜索策略[35]和进化算法搜索策略[39]快 2 倍和 5 倍。后续的工作[88]中指出,文献[37]中使用代理模型来评估结构性能会带来不可忽略的偏差,

转而通过权值共享降低了计算负荷,并且削弱代理模型的作用以更好地搜索。

3.2.5 单次模型搜索方法

单次模型搜索,也称一步法,顾名思义是仅需要完成一次网络训练即可基本完成搜索的任务。该方法主要是解决诸如早期神经网络架构搜索方法使用嵌套式优化,即从搜索空间采样出模型结构,接着从头训练权重所带来的计算成本问题。尽管后续为了解决计算量问题提出权重共享方法,但是该方法又将模型结构分布不断参数化,导致超网络权重之间以及权重和模型参数之间深度耦合,后续仍然需要精细地微调和优化。单次模型搜索具体的实现方式分为两类:一类是以 SMASH[39]、GHN[90]模型为代表的辅助网络设计;一类是一个从大网络开始做减法的神经网络模型搜索框架,基于强化学习的 ENAS[40]、包括 DSO-NAS[85]、ProxylessNAS[88]、FBNet[61]、SNAS[84]、DARTS[38]在内的大多数梯度搜索方法都可以归类为这一方法中。

基于快速获得网络结构在验证集上的性能以对网络性能排序的构想,Brock 等[39]提出的 SMASH 模型的大致流程是首先在候选网络中挑选网络结构,再用训练好的全卷积的辅助网络(HyperNet)负责不同网络结构的权值参数生成,即动态地实现结构编码到权重的映射,通过测试验证集上的准确率即可完成候选网络结构的相对排序,并最终确定最优的网络结构。整个搜索过程中,仅全卷积的辅助网络需预先用梯度下降法训练,以获得对变长的候选网络表示有足够的抽象理解能力。该算法避免了从头训练每个候选网络权重的情况,使得训练的时间大大减小。

另一单次网络搜索方法设计一个包括所有子结构的超网络,并且在整个搜索过程中,只训练一次,允许各结构共享其中的权重,即所有的子结构便可以直接从超网络继承获得权重,无须从头训练,仅需在验证数据集上评估性能即可。

Bender 等[91]摒弃了文献[39]中为网络结构动态产生权重的辅助网络 HyperNet 和文献[40]中从搜索空间取样的 RNN 控制器,设计并训练一个大型网络,使候选模型直接从该大型网络中复用权重,并且在验证集上的性能也能真实地反映网络结构的性能。所选的最好网络结构还需在网络结构搜索结束之后,从头训练得到权值,得到最终的网络模型。在神经网络结构中很多位置上包含有很多不同的操作选择,传统的搜索空间是随着操作数目呈指数增长的,而单次搜索模型的大小只随操作数目线性增长,究其原因就是相同的权重可以用来评估取样的不同网络结构,极大地降低了计算量。该大型网络由带动量的随机梯度下降法训练,以保证大型网络中的权值可以用来预测其中取样网络结构的性能。该工作还尝试解释大型网络中的权重为何可以在不同的结构中共享:因为大型网络在训练时就学会辨别有用的操作,并且往特定方向更新权重,所以保留特定操作组合时在验证集上的性能是最好的。

在 Saxena 等[92]提出的三维栅格结构的织物(fabrics)状网络中,三个维度分别表示层数、层尺度和特征映射。尽管整个结构包含了比单个网络更多的参数,但是整体的计算资源占用率会比测试候选网络更少。在图像分类和语义分割上的实验结果均验证了模型的可行性;后续的工作[93-94]基于文献[92]提出的 neural fabrics 分别提出了 MSDNet 与 RANet,均是对于不同任务难度的图像选择不同深度的网络结构,旨在解决模型前向推导时资源有限导致的即时分类与固定预算批分类的问题。优点是可以策略性地分配适当的计算资源,缺点是模型框架搜索方向单一,只能在深度上调整;Guo 等[95]提出简化后的单路径超网络,训练中不需要任何超参数来指导子结构的选择,按照均匀的路径采样方法进行训练,使得所有子结构以及权重获得充分而平等的训练。一旦该超网络完成训练,不同于之前的单次结构搜索工作[39,90-91]中使用的随机方法,它是结合进化算法选择最佳网络。又因为进化算法中的变异和交叉过程是可控的,所以可以产生满足诸如计算时延、能量消耗等多目标任务需求的合适网络结构。

这种从较大网络出发做减法的框架类似于 3.1 节中的剪枝方法,两者的区别在于剪枝专注于层内的修剪,只能改变层内的卷积核数目,但是无法在更大的层面上改变网络的结构。并且作为删减连接的依据多为权值绝对值的大小;而单次搜索模型则可以通过路径(path-level)的修剪更加灵活地改变模型的结构。且模块或操作选择的依据直接为数据集上的性能表现。

3.2.6 随机搜索策略

随机搜索,顾名思义是在搜索空间中随机取样而进行评估的搜索策略,是最简单也是最基础的搜索策略,往往被视为是比较低智能的算法,但是其性能起初也很难被超越[96]。不过因其可以很灵活地应用于不同的搜索空间中,且高度支持并行化的运算,进而大大地加速搜索进程,近年来

也通常用作 baseline。文献[36]已经证明随机搜索在小数据集上得到的网络准确率具有较强的竞争力,近年来,随机搜索通过进一步与合理设计的搜索空间[39,46]或者其他技术[57,90]相结合,仍然保持顶尖的性能。

Xie 等[26]基于 ResNet 和 DenseNet 的成功很大程度上可以归功于创新性的连接方式,提出了 RandWire 模型使用三种随机图模型实现结构的生成和搜索。最主要的贡献是实现随机生成网络这一更高层次的随机,打破人工设计所带来的强大先验信息的束缚,有利于发现人未曾设计出的全新结构。在该背景下的随机搜索也表现出与其他复杂搜索策略相当的竞争力。

Li 等[57]提出将随机搜索分别与早停(early-stopping)法和权值共享技术相结合,实验结果显示:在 PTB 和 CIFAR-10 数据集上,两种优化后的随机搜索均超过了现阶段领先的基于强化学习算法的 ENAS 方法[40]。作者提出,理论上网络搜索仍是一种特殊的超参数优化,而随机搜索是超参数优化的标准基线。故理论和实验均说明随机搜索仍是现阶段效率最高的搜索策略。

在 Zoph 等[35]提出对搜索空间的第一次优化后,就经过实验测试发现在更好的搜索空间中,基于均匀分布采样的随机算法仅稍逊于强化学习方法的性能;在 Liu 等[59]设计的更高级的分层搜索空间中,随机搜索取得和进化算法相当的性能;Yu 等[97]提出在同一个神经网络架构搜索评估框架的基础上,随机搜索的策略与其他典型搜索方法[38,40,81]在简单的任务中的表现不相上下,甚至更好。

基于随机搜索在神经网络架构搜索发展至今天越发展现出强势的竞争力,越来越多研究人员[57,97]指出目前主流的神经网络架构搜索研究并没有和随机搜索有充分以及公平的比较。Yu 等[97]也制作了随机搜索的 baseline 以供后续研究,并呼吁使用统一的评估方法的基础上,寻找更优的网络结构搜索策略。

3.2.7 蒙特卡罗树搜索

蒙特卡罗树搜索(Monte Carlo tree search, MCTS)是大型马尔可夫决策问题中找到最优策略的一个方法,已经成熟运用于众多领域的规划算法,最引人瞩目的就是围棋机器人 AlphaGo。该方法的主要思想是利用当前收集到的信息来引导搜索,以更好地遍历搜索空间。具体步骤是将搜索空间变成树形结构,树中的节点表示候选网络结构,之后在蒙特卡罗树搜索中循环选择、扩展、仿真和反向传播 4 个步骤,并在迭代循环的过程中不断拓展博弈树的规模。反向传播的目的是更新反向传播路径上所有节点的总模拟奖励以及总访问次数,即被访问节点被利用和被探索的信息,分别反映了节点的潜在价值和被探索的程度。起初,Negrinho 等[98]指出之前的神经网络架构搜索工作[33,75,78]均预先设定固定的搜索空间,难以修改和拓展,使它们不能作为结构搜索的一般性算法,遂将搜索空间转换为一棵搜索树,用蒙特卡罗树搜索解决结构搜索问题;之后,Wistuba[99]则在文献[98]的基础上改进蒙特卡罗搜索树算法,引入迁移学习以加速搜索,并在 CIFAR-10 数据集上取得了 6.08% 的进步;Su 等[100]针对蒙特卡罗树中经复杂运算得到的结点历史信息仅使用一次后就弃置的低效问题,通过提出结点通信机制,即在待搜索超图中同一深度的同一候选操作结点共享激励值,实现中间的计算结果保存在树结构中,以便后续的决策。

总的来说,蒙特卡罗树搜索方法能权衡对已有知识的利用和对未充分模拟节点的探索,并且可以随时结束搜索并返回结果,在超大规模博弈树的搜索过程中相对于传统的搜索算法有着时间和空间方面的优势。尽管上述的各个搜索策略特点鲜明,但是搜索策略之间并不是完全割裂的,比如某种搜索策略中就参考了其他搜索策略,甚至两两之间相互融合促进。例如 He 等[64]结合了强化学习和剪枝的思想,综合优化了模型压缩以及实际部署时计算加速性能;Cai 等[88]在设计模型中更新结构参数时尝试使用不同的搜索策略,梯度方法和强化学习均搜索到了符合预期的网络结构;更加典型的是单次模型搜索方法和梯度方法在思想上就有相似之处,实际的应用中也多次结合[38,61,84-88]。

表 3 对当前主流的神经网络搜索策略,如强化学习策略、进化算法策略、梯度算法策略等进行了优缺点分析和总结。

3.3 性能评估策略

性能评估阶段旨在得到候选网络结构的性能,以指导整个搜索过程。如上文所述,性能评估就是整个搜索过程中最耗时的阶段,评估部分的速度提升对整个网络搜索过程速度提升作用最明显。最简单的、也是最早期神经网络架构搜索方法[33,35]使用的性能评估方法就是将每一个候选网络在训练集上从头开始训练,并将训练结束之后在验证数据集上的准确率作为性能评估。由上文中可知,搜索空间中的候选网络数的数量级巨

表 3　搜索策略优缺点总结

Tab. 3　Summary of Advantages and Disadvantages in search strategy

搜索策略	亮点	不足	主要工作
强化学习	将神经网络结构搜索抽象提炼出强化学习的四要素,使用 RNN 作为控制器产生子网络,再对子网络进行训练和评估,得到其网络性能,采用强化学习算法直接更新控制器的参数,一直迭代以上过程直到达到目标	①数据利用率低,计算资源消耗高;②过度的探索可能会导致收敛变慢,而过度的利用会导致收敛到局部最优	NAS[33] MetaQNN[75] EAS[56] NASNet[35] BlockQNN[41] AutoAugment[77] HAQ[76] MnasNet[43] NAS-FPN[44]
进化算法	①随机生成一个种群(N 组候选网络结构),开始循环选择、交叉、变异步骤,直到满足最终条件;②搜索早期的速度快,对数据集具有稳定性;③不预设搜索的范围,鼓励更大的搜索空间	①较高的计算资源消耗;②当种群数量较小的时候,会陷入较差的局部最优	文献[78] 文献[59] AmoebaNet[36] LEMONADE[79] 文献[49] ChamNet[80] 文献[101] 文献[102] 文献[103] 文献[104]
梯度算法	①将离散的搜索空间连续化,使得优化的目标函数可微,将网络结构搜索转化为连续空间的优化问题,采用梯度下降法求解并进行搜索; ②搜索效率高,信息流动速度快,数量级加速搜索过程	搜索得到网络结构杂乱,没有明显规律	NAONet[83] Auto-DeepLab[45] GDAS[86] P-DARTS[87] 文献[105] 文献[106] 文献[107]
序列模型 SMBO	启发式搜索的策略,训练的模型从简单到复杂,即按照复杂度逐渐增大的顺序搜索网络结构		文献[108]
剪枝 Pruning	以一个全连接的结构出发,通过设计判断连接重要与否的准则,不断地移除冗余的连接以得到最后的模型	初始化需要性能较好的网络	文献[109] 文献[66]
单次架构搜索技术 One Shot	所有的候选网络都视为一个超级图的不同子图,并在此超级图中共享权重,搜索过程中仅有该超级图需要训练,即可得到不同候选结构的性能排序	①超级图的设计需要精细的专业知识,并且施加了人为的先验约束;②鲁棒性较差;③易导致好的网络模型没有获得充分训练	CNF[92] SMASH[39]
随机搜索	①在搜索空间里面随机采样得到网络结构,比较在数据集上的性能差异;②能灵活运用到不同的搜索空间中	搜索效果的保证倚重于搜索空间的合理设计	文献[59] GHN[90]

大,用上述的性能评估方法必然会导致过量的计算负担,对于 NAS 的实现不切实际。故众多工作展开了对性能评估阶段的速度提升的研究。本节着重介绍模型性能预测、权值共享、迁移学习、自适应调整、计算加速与简化等多种方法。搜索代价可以分为搜索过程中 GPU 运行时间和内存消耗两方面,因 GPU 运行时间方便统计且公开数据较完备,表 4 以 GPU 运行时间对比分析其数据不同速度优化方法的搜索代价(数据集为 ImageNet,运行平台为 Tesla V100)。

表 4 不同速度优化方法的搜索代价数据对比

Tab. 4 Comparison on search cost of different speed optimization method

方法	搜索策略	GPU 成本/h	准确率/%	
			Top-1	Top-5
AmoebaNet-A	EA	75 600	74.5	92.0
NASNet-A	RL	48 000	74.0	91.3
MnasNet	RL	40 000	74.0	91.8
Proxyless-R	G + P	200	74.6	92.2

3.3.1 模型性能预测

模型性能预测方法[37,110]通过设置一个性能预测函数,将候选网络的结构作为输入,直接输出验证集上的准确率来指导搜索,避免了冗余的训练。而这样一个性能预测函数则是基于搜索空间中若干候选网络的实际训练及验证结果得到的。

低保真方法,或者代理模型方法,指的是使用诸如训练更少的次数和时间[35, 41,43 - 44, 111 - 112]、仅选取部分原始数据[64,84,90, 113]、使用低分辨率的图像[114]、每一层使用更少的滤波器[35 - 36,115]等来评估模型的性能。尽管低保真方法引入了误差,一般会低估真实性能,但是因为在模型选优过程中并不需要绝对数值,只需要有不同网络间的性能评估相对数值即可排序选择了,故公认可以使用具有保序性质的低保真方法以降低计算开销。

具体实现低保真的方法有很多,以训练更少次数为例,Zhong 等[41]通过实验发现 FLOPs 和密度 Density(定义为有向非循环图中边数量比节点数量的值)与完整训练之后模型的准确率 ACC_{ES} 呈负相关,故提出修正强化学习的激励 R,引入与 FLOPs 和 Density 相关的参数 P_F 和 P_D。

$$R = ACC_{ES} - \mu\lg(P_F) - \rho\lg(P_D) \tag{6}$$

使激励和最终的准确率更加相关,弥合代理度量和真实准确率的差距。于是据此提出了早停策略,通过提前训练停止获得的准确性对于预测训练完备后的模型准确性仍具有较高的参考价值。

尽管如此,学界还是有其他不一样的声音:Zela 等[112]提出当低保真的训练次数与真实评估时的训练次数差距足够大时,对于性能的低估可能会影响到性能的排序;Falkner 等[116]也呼吁在神经网络架构搜索的研究中增强保真度,以保证搜索的效果。

除了低保真方法,一部分论文[34,60,117 - 118]提出学习曲线外插值模型。学习曲线表现形式为模型性能是关于训练时间或迭代次数的函数。该方法借鉴人类在网络模型训练一段时间后,根据各种指标曲线就能大体判断超参数组合是否达到预期。该方法即用训练前期观测到的指标数据进行插值预测最终性能,实现在评估一个模型时,采取提前终止策略,不必训练充分。早在 2014 年,Swersky 等[118]基于学习曲线大体上是向某一未知最终值而指数式衰减的前提假设,设计一个混合众多指数型衰减基函数的核函数来预测学习曲线,并且结合贝叶斯方法高斯过程,以预测超参数组合最终的性能。Domhan 等[117]也指出:实验已经证明理想的学习曲线是饱和函数。在以往早停策略的实验运用中,出发点是防止过拟合现象,其判断是否有必要继续训练的依据就是学习曲线是否已经达到了饱和点。而该工作中的早停法是为了节省计算资源,故选取 11 种饱和函数,赋予权重且合并后拟合学习曲线,以预测模型后面的性能,预测准确率低或者学习曲线差的网络结构暂停训练或直接放弃。实验结果在不同规模的卷积神经网络中均取得了较好性能。后续的工作中,Klein 等[34]优化了文献[117]中的似然性函数,通过去除引起偏差的某项,在不影响网络结构性能的前提下,使超参数配置和预测的学习曲线相关性更高,解决了学习曲线插值过程中的不稳定性问题;不同于文献[117]中对于所有网络结构都是仅完成部分的学习,Baker 等[60]的工作完整训练前 100 个候选网络,并且摒弃了文献[117]中复杂的马尔可夫蒙特卡罗(Markov chain Monte Carlo, MCMC)取样方法和人工设计的拟合学习曲线的基函数,结合前期的学习曲线与网络结构信息,超参数信息和时序上的验证精度信息共同来进行预测,实现早停进一步提升加速效果。

另一种性能预测方法借助了贝叶斯方法。贝叶斯方法[119]是超参数优化问题的常用手段,适用于评估代价昂贵和候选空间复杂巨大的情形。

Shahriari 等[120]将贝叶斯方法解决的问题形式化为

$$X^* = \arg\max_{x \in \chi} f(x) \tag{7}$$

在网络结构搜索问题中,χ 表示候选网络集合,问题转换为从 χ 中寻找 x 使得未知目标函数 f 取得全局最大值。贝叶斯方法的基本思路是对目标函数 f 猜想一个先验分布模型(一般为高斯分布),然后利用后续获得的信息,不断优化此猜想模型,使模型越来越接近真实的分布。网络结构搜索问题测试成本高而且影响维度多,于是 Kandasamy 等[42]应用贝叶斯方法中最常见的先验分布模型,即高斯分布,设计距离表征 OTMANN,并且在获取函数(acquisition function)中结合进化算法完成网络搜索。实验结果显示,在多层感知机和卷积神经网络上表现优异。多目标搜索问题中,Dai 等[80]在准确率和能量消耗的评估上都使用了高斯分布为先验模型的贝叶斯方法,使得算法可以搜索更少的候选网络并且较快收敛。White 等[119]设计一个神经网络模型来实现贝叶斯优化,仅训练 200 个结构即可在 1% 的误差内预测出其他网络结构在验证集上的准确率,并且最终搜索网络的性能优于其他典型性方法[38,57,84]。

3.3.2　权值共享

权值共享的主要思想是让训练好的网络尽可能多地被重用。经证实当权值共享之后,候选网络的性能评估可能不是真实准确的,但是足够提供优选排序的参考。基于单次搜索模型的搜索策略均借鉴了权值共享的思想,该种搜索策略已在 3.2.5 节中详细介绍了,故不在此赘述,现着重介绍权值共享技术和其他搜索策略的结合使用。

Pham 等[40]设计的 ENAS 模型让搜索中所有的子模型重用权重,克服了之前的计算瓶颈:即每一轮训练候选网络都从头开始训练至收敛,没有借助已有的训练结果。具体的算法实现是将网络搜索的采样过程转化为循环神经网络作为控制器,在一个大型有向非循环图里面采样子图的过程。该有向非循环图中的节点代表网络结构中的模块或操作,并且保存有各自的权值。只有模块被控制器采样之后,其权值才会被激活,激活的模块共同组成候选网络,故候选网络中的权值是从有向非循环图中继承而来,不需要从头训练即可以快速验证精度。训练过程中用反向传播交替优化有向非循环图中存储的权值信息和控制器中的结构参数,实验结果将搜索时间至少缩短至原本的 1‰,并且在 CIFAR-10 上的测试误差与 NASNet 不相上下。谢彪[121]基于 ENAS 进一步提出在预训练阶段的子图中进行结构近似的参数共享,并且探讨了不同超参数的参数共享方式,既保证训练的高效性,也避免参数共享带来的偏置问题。

权值共享技术还能和基于进化算法的搜索策略[111]相结合,譬如 Real 等[78]为了加速训练过程,其允许在进化算法过程中突变生成的子结构继承母结构的权重值。具体应对不同情况的处置如下:如果某一层的结构匹配,则保留母结构权值;对于改变学习率以及类似的突变,则可以保留母结构的所有权重值;对于重置权重值以及类似的突变,则不继承任何权重值;更多的突变,例如卷积核大小的突变等,则保留部分的权重值。

在权值共享技术的发展之下,网络态射概念也逐渐完善,网络态射技术将网络变形,即将参数为 w 神经网络 f^w 快速映射为具有参数 $\tilde{w}$ 的网络 $g^{\tilde{w}}$,同时保持功能不变的一项技术,并且具有多种映射范式,基本方向都是基于小型网络做加法。映射得到的网络可以重用之前训练好的权重,而不是从头开始训练。Elsken 等[122]运用网络态射技术于进化算法的突变过程,产生预训练过的网络,以节约从头开始训练一个网络的巨大花销。网络态射主导的突变方向主要包括使网络结构变深、变宽和添加跳跃连接。但是网络态射仅扩充结构而不会缩小结构的缺点,这使得以往的网络态射参与的结构搜索疏于对小型结构的探索,而偏爱较大网络。针对这一缺点,又有以下三项代表性的工作:Jin 等[123]将贝叶斯算法和网络态射相结合,贝叶斯算法用以指导在网络态射集合中选择最有可能的操作。并且通过设计一种树形结构,不仅拓展结构末端的叶节点,而且在中间节点的基础上拓展;Elsken 等[79]利用网络态射作为进化算法中产生子代的方法加速模型的训练,考虑到传统的网络态射只会增加网络的尺寸,提出类网络态射,具体的操作包括剔除某层、剪枝卷积核和替换某层;Cai 等[56,124]提出类似的“网络变换”概念,并且定义该变换为重用已有网络结构信息,而不是广义上对网络结构的任何更改。将网络变换技术作为强化学习算法中的动作,而状态为当前搜索的网络结构。以往的方法会在结构上叠加全新设计的一个新层,但是网络变换是在已有的层次上进行增减、加宽等操作,从而保证当前任务中已有的结构以及权值的重用。

3.3.3　迁移学习

网络结构的迁移能力本身就是评价性能的一

个重要指标。诸如文献[75,79]等尝试将CIFAR-10上搜索得到最优网络结构迁移到训练其他的数据集中,以检查搜索出网络结构的迁移能力,并且比较了迁移网络从头训练和从已有权值微调两种方式。现有其他工作[39-41]已经验正了在大数据集上学习训练得到的模型比在小数据集上学习得到的模型,在该小数据集上的性能更优。指出该类向下迁移方法,在大数据集上学习得到的网络迁移到不同的数据集或者任务中会得到意想不到的结果。

网络的迁移能力不由得引人思考,是否可以将从简单任务中学习得到的网络结构迁移到目标复杂任务中,以达到加速搜索的目的。神经网络架构搜索技术也可以借助网络模型的迁移能力,将某个数据集或者任务上已经训练得到的结构应用到目标数据集或任务上。一些工作[38,81]将CIFAR-10和Penn Treebank上学习得到的结构分别迁移到体量更大的CIFAR-100和WikiText-2数据集任务;文献[41,84-85]则索性将基于CIFAR-10和CIFAR-100数据集搜索得到的网络结构迁移到更大的ImageNet。这些工作均能媲美,甚至超越已有研究的性能。这是因为大量历史寻优数据的积累会极大程度上帮助新问题的寻优,故借助从之前任务中习得的知识,迁移已经训练好的模型权重或者结构参数对新的目标问题进行赋值,相当于从一个较完备的初值开始寻优,以缩短收敛的时间。

3.3.4 训练过程中的自适应调整

很多研究也不断尝试根据训练过程中的即时反馈,动态地调整搜索技巧以实现搜索过程的加速。

为了保证搜索得到的网络在评估阶段也具有同搜索阶段一样理想的准确率,Chen等[87]在搜索过程中让网络的深度逐渐加深,以缩小搜索阶段和评估阶段的网络深度不同带来的差距,当然与此同时也带来了沉重的计算开销。为了应对此问题,在选择搜索速度较快的梯度方法的基础上,随着搜索的深入和网络的加深,模型会主动丢弃之前搜索过程中评估较差的候选操作,使其不再占用计算资源。CIFAR-10上的实验结果也显示网络搜索仅需耗费0.3个GPU日就使搜索出来的结构达到了低至2.5%的错误率;类似的工作还有So等[49]基于文献[36],将进化算法和联赛选择算法应用到改进目前自然语言处理任务中最火热的特征提取器Transformer中,然而在语言数据集中并不能像之前视觉研究一样提取出合适的低保真数据集,为了加速搜索过程,动态地部署计算资源的分布,提出动态障碍(progressive dynamic hurdle, PDH)方法,允许训练前期表现更好的结构训练更多的步数,保证实验的可行性,成功地运用神经网络架构搜索技术自动化搜索得到性能更优的Transformer模型。

调整搜索空间中候选网络的搜索顺序也能实现搜索加速。Liu等[37]尝试启发式的搜索策略SMBO,采取递进的方式,考虑到简单的结构训练速度更快,按照模型简单到复杂的顺序进行搜索。具体借助LSTM模型,输入定义为描述网络结构的变长字符串,按照复杂度递增的顺序搜索,借助贝叶斯方法使用获取函数输出预测的验证精度,来评价候选的模型架构。之后基于训练阶段性能更好的结构继续拓展,再结合其他加速方法,整体搜索速度上相较文献[35]提升了8倍之多。与SMBO方法类似,但是搜索从复杂到简单的一个方法是Yan等[125]提出了适合分布熵的采样方法,其在搜索初期抽取更多样本鼓励探索,随着学习的进行,缩小候选网络结构的分布范围从而减少取样数。

3.3.5 计算加速和简化方法

在整个神经网络架构搜索技术的发展历程中,一直有在计算层面上使用技巧实现搜索过程的加速。除了3.1.2节中已详细阐述的基于基本单元搜索堆叠完整网络结构的方法,代表性工作包括并行计算、量化和查找表技术。

并行计算是神经网络架构搜索技术发展伊始就应用的技术,早在2017年,Zoph等[33]提出分布式训练以及异步参数更新方法,在实现多运算单元加速计算的同时,某一运算单元得到的参数可以公布给全局的运算单元使用,保证了该领域奠基工作得以顺利完成;类似的分布式异构框架也在之后的研究[39,41,78]中被广泛使用。该框架使多GPU设备之间可以高效合作,并且便于结合诸如早停等其他加速策略进一步减少计算消耗。

因为GPU内存占用随着候选网络的数量线性增长,为了解决内存爆炸的问题,量化方法也备受关注。例如,Cai等[88]的工作将神经网络架构搜索视为对过度参数网络进行路径剪枝,该网络中的各个位置保留了所有可能的操作。作者将结构中的参数全部0、1二值化,每次仅激活一条保留路径上的参数,再使用基于梯度的方法搜索,无须训练数以万计的候选网络,也无须额外的元控制器,遂将显存需求降为与训练网络模型相当的级别,结果显示搜索速度较文献[43]和文献[35]

分别提升了 200 和 240 倍。

对于某些优化的参数,例如计算时延,合理使用查找表也是一种高效的加速方法。因为在大部分的移动 CPU 或者数字信号处理器上运算操作是连续的,Wu 等[61]假设每个操作之间都是独立的,预先制作计算时延数据库,搜索时通过查找每个操作的时延,并且累加得到最终整个模型的时延,使得搜索出的网络模型兼顾速度和精度。文献[80]中的实验说明,对于运算操作查找仅需消耗 CPU 大概 1 s,而如果在真实物理设备上模拟则需要花费数分钟。尽管对于不同的设备有不同的查找表,但一旦查找表标准化之后,就可以重用给后续的研究提供基准化的数据,可以说一劳永逸。

另外一项值得关注的工作是 Ying 等[126]直接根据查找表的思路设计了第一个开源的网络搜索架构数据库——NAS-Bench-101。主要工作是基于 42.3 万余个卷积神经网络结构,使用图同构(graph isomorphism)技术拓展出 500 万个网络结构,并且同时编译以上结构在 CIFAR-10 上包括运行时间和准确率等性能。该数据集的工作耗费了数百小时的张量处理器(tensor processing unit, TPU)运算时间,方便后续的研究者以毫秒级的速度查询评估各种模型的质量。这种量级的加速方式一定程度上解决了神经网络架构搜索技术需要大量计算资源的问题,让更复杂的任务研究具有可行性,比如对比不同搜索方式的性能等。

4　数据集与算法性能对比

4.1　数据集

数据集在神经网络架构搜索研究中起着非常重要的作用,基准的数据集可以让搜索算法更容易复现,并且有利于搜索算法之间的性能比较。由于神经网络架构搜索主要还是基于主流视觉识别任务,故使用目前各个领域主流的数据集。下面按照具体的计算机视觉任务对常用的神经网络架构搜索数据集进行介绍和总结。

对于图像分类任务,也是目前神经网络架构搜索领域主流使用的数据集包括 MNIST、SVHN、CIFAR-10、CIFAR-100 和 ImageNet。MNIST 数据集是一个大型手写数字数据库,由 250 个不同人手写的数字构成,其中有包含 60 000 个示例的训练集和包含 10 000 个示例的测试集,图像尺寸均归一化为 28 ×28 的灰度图,是深度学习领域应用最广泛的数据集之一;SVHN 同样也是识别图像中阿拉伯数字的数据集,但是图像来自真实世界的门牌号码,由 Google 街景提供拍摄的门牌号图片。初始训练集包括 73 257 张图像,测试集中有 26 032 张图像,并且有一个包含 531 131 张图像的附加训练集。尽管与 MNIST 相似,但是包含更多标记数据,而且取样于更加困难的现实世界问题。CIFAR 数据集取自一个包含 8 000 万张 32 × 32 的自然彩色微小图像的数据集,CIFAR-10 和 CIFAR-100 均包含 60 000 张图像,且分为 50 000 张训练图像和 10 000 张测试图像。不同的是,CIFAR-10 分为完全相互排斥的 10 类,每类包含 6 000个图像;CIFAR-100 分为 20 个超类中的总共 100 类,每类包含 600 个图像。相比于之前的数据集,CIFAR 数据集是现实世界中真实的物体,不仅噪声大,而且物体的比例、特征都不尽相同。

针对特定领域的应用会用到细分领域的数据集,譬如花卉数据集 102 Oxford Flowers[117]、包含 37 种猫犬类的宠物数据集 Oxford-IIT Pets[77]、细粒度的飞行器数据集 FGVC Aircraft[77]、细粒度汽车数据集 Stanford Cars[77]、图像尺寸约为 300 × 200 的 Caltech-101[77]等也经常被使用,均为神经网络架构搜索在现实中的各个应用提供了实验基础。

对于目标检测和密集标记等任务,对数据集的要求也越来越高,并且制作难度更大。神经网络架构搜索研究在该领域主要使用以下的数据集:PASCAL VOC 2012 包含属于 20 类的 11 530 张图像,以及总共 27 450 个目标的标定,故平均每个图像中有 2.4 个目标;COCO 数据集包含 91 类目标、328 000 个影像和 2 500 000 个标签,其中图像主要截取自复杂的日常场景,包含目标及目标位置标定。与 PASCAL 相比具有更多的类和图像,且每类包含的图像多,有利于获得每类物体位于特定场景的能力;为了方便在自动驾驶领域的应用,Cityscape 数据集被提出,包含 5 000 张在城市环境中驾驶场景的图像;在更加细分的面部识别领域,则由 Part Labels 面部数据集填补空缺,其包含2 927张标注好头发、皮肤和背景区域的面部照片。

对于语言建模,即预测文本中下一个词的任务,大多借助以下两个代表性英文数据集:Penn Treebank Dataset[33]取材于 1989 年的《华尔街日报》中的 2 499 篇文章,包含超过百万词汇,标注内容包括词性标注以及句法分析;WikiText-2 则是由大约 200 万个从维基百科文章中提取的单词构成,体量是 PTB 数据集的两倍,而且语言更加接近现代日常实际情况。

4.2 现有方法性能对比

对于神经网络架构搜索技术,搜索策略的选择对搜索结果有至关重要的影响,但仍然存在很多其他的影响因素:从搜索空间来说,会导致搜索数据类型和搜索量不同,进而影响总体的识别率;从数据预处理的角度来说,即使采用了相同的搜索策略,同样会影响最终的搜索结果;性能评估策略和具体任务也会影响搜索的性能参数。为了使读者更直观地了解,本部分仅对比给出不同数据集以及搜索策略下,在图像分类任务中准确率和网络参数量的性能对比。

表5中选取了CIFAR-10、CIFAR-100、MNIST、SVHN和ImageNet五个数据集,其中"*"表示模型训练过程中对数据进行了数据增强。表格涵盖每个数据集下代表性的搜索方法以及图像分类任务的实验结果。搜索策略栏目中,manual表示手工设计的网络,random表示随机的搜索策略,RL表示基于强化学习(reinforcement learning)的搜索策略,EA表示基于进化算法(evolutionary algorithm)的搜索策略,gradient/G表示基于梯度的搜索策略,SMBO表示序列模型搜索策略,pruning/P表示基于剪枝的搜索策略,OS表示单次架构搜索策略;参数量栏目中,"—"表示参考文献没有收录,或者代码细节没有公开,导致无从查证计算的参数量。本文后续的表格也遵循以上说明。

5 神经网络架构搜索的应用

图像分类任务是神经网络架构搜索的起点,也是最基础而且目前研究最完备的高级识别任务,但是神经网络架构搜索在以下的领域也取得了值得关注的进步。

5.1 目标检测

目标检测是指在影像中定位物体空间位置的一类任务,是计算机视觉领域的关键问题。图像特征提取是实现目标检测任务的基础,神经网络架构搜索技术也大部分用于优化图像特征提取阶段的任务。典型性的工作[35,39]将网络结构搜索在图像分类任务中学习得到的更通用的图像特征迁移到目标检测任务中,即将在ImageNet中搜索得到的NASNet嵌入到Faster-RCNN架构中,并将应用于COCO数据集的结果显示在大模型和移动端设备中以实现更优的性能;类似地,Tan等[43]将神经网络架构搜索得到的模型作为特征提取器植入SSDLite目标检测框架中,综合考虑了实际应用中的移动设备的性能,即在准确率的基础上考虑了对时延的优化。实验结果显示,目标检测任务中的各项性能均超越MobileNet作为特征提取器的框架,相较SSD-300减少了7倍的参数量和42倍的乘积累加运算,增强了模型的灵活性和便携性,更加适应科技和社会的发展。

目前有相当一部分的目标检测网络框架由骨干网络和特征金字塔网络这两个主要组件构成。其中,采用特征金字塔结构是为了融合不同尺度的图像特征以解决多尺度的问题,而特征金字塔均由人工设计,并不一定是最优的结构,为了自动化地获得性能更优的特征金字塔结构,Ghiasi等[44]则聚焦于采用神经网络架构搜索的技术构建特征金字塔,实验结果显示搜索得到的金字塔网络可以很好地和包括MobileNet、ResNet、AmoebaNet等主流骨干网络共同完成目标检测任务,并且进一步提升准确率、降低设备时延。

表6展示了神经网络架构搜索技术在目标检测任务上的性能对比。因为不同的图像分辨率对网络结构在计算成本、预处理方式和特殊任务(例如小物体检测)上的要求不一样,所以将其作为参考列在表中。目标检测任务的评价指标由平均精度均值表示,默认为以与数据集标注交集并集比(intersection over union, IoU)大于0.5作为正确分类的指标而得到的准确率。$mAP[0.5, 0.95]$则表示在0.5~0.95区间中以0.05为间隔取的共10个不同IoU为指标下的mAP结果做平均,故$mAP[0.5, 0.95]$要求更加严格,数值比mAP更低。表6中Mini-val和Test-dev则为COCO数据集中的不同数据子集。

5.2 逐像素标注

逐像素标注(dense image prediction)任务精细到处理像素级的内容,相比于简单的图像分类仅需要输出一个标签,其需要精确到像素级,为图像中的每个像素分配一个标签,是计算机视觉中更高级的应用,进一步可以分为场景解析(scene parsing)、人物分割(person part segmentation)、语义分割等子任务,主要的数据集包括Part Labels、Cityscapes、PASCAL VOC 2012、ADE20K等。相较于之前的图像分类任务中广泛使用的降采样策略,或者使用低分辨率图像做性能评估,逐像素标注任务的特征给研究带来诸多困难:逐像素标注任务的输入尺寸分辨率更大,需要在图像多个层次上提取具体特征且保持高分辨率,故网络结构具有的变数更多,同时要求搜索技术更加高效以应对更高分辨率带来的复杂计算,故搜索难度将会更大。

表 5　不同数据集下代表性搜索策略算法在图像分类任务上的性能比较

Tab. 5　Performance comparison of representative search strategies on image classification tasks under different datasets

数据集	方法	搜索策略	准确率/%		参数量
			Top-1	Top-5	
CIFAR-10	ENAS[40]	RL	96.46		4.6×10^6
	NASNet-A(baseline)[35]	RL	96.59		3.3×10^6
	AmoebaNet-A ($N=6,F=36$)[36]	EA	96.66		3.2×10^6
	NAONet[83]	gradient	97.02		28.6×10^6
	PNASNet-5[37]	SMBO	96.59		3.2×10^6
CIFAR-10*	DenseNet ($L=100,k=24$)	manual	96.26		27.2×10^6
	GHN Top-Best, 1K($F=32$)[90]	random	97.16		5.7×10^6
	NASNet-A (7@2304)[35]	RL	97.60		27.6×10^6
	LEMONADE[79]	EA	97.42		13.1×10^6
	SNAS(single level + moderate constraint)[84]	gradient	97.15		2.8×10^6
	Proxyless-G[88]	G + P	97.92		5.7×10^6
	One-Shot Top ($F=128$)[91]	OS	96.10		41.3×10^6
CIFAR-100*	DenseNet[25]	manual	82.8		25.6×10^6
	ENAS[40]	RL	80.57		4.6×10^6
	Large-scale Evolution[78]	EA	77		40.4×10^6
	P-DARTS CIFAR-100[87]	gradient	84.08		3.6×10^6
	PNAS[37]	SMBO	80.47		3.2×10^6
	SMASH V2[30]	OS	79.4		16×10^6
MNIST	NIN	manual	99.53		—
	R-CNN	manual	99.69		—
	MetaQNN[75]	RL	99.56		9.67×10^6
MNIST*	DropConnect	manual	99.68		379×10^3
	CNF-sparse ($L=16$, $C=32$)[92]	OS	99.52		249×10^3
	CNF-dense ($L=8$, $C=64$)[92]	OS	99.67		5.3×10^6
SVHN	NIN	manual	97.65		—
	R-CNN	RL	98.23		—
	MetaQNN[75]	RL	97.72		10.38×10^6
	EAS (plain CNN, $depth=16$)[56]	RL	98.17		—
	EAS (plain CNN, $depth=20$)[56]	RL	98.27		—
ImageNet	ResNeXt-101 ($64\times4d$)	manual	80.9%	95.6%	83.6×10^6
	RandWire-WS(regular computation regime)	random	80.1%	94.8%	61.5×10^6
	NASNet-A ($N=6,F=168/6$@4032)	RL	82.2%	96.2%	88.9×10^6
	AmoebaNet-A ($N=6,F=448$)	EA	83.9%	96.6%	469×10^6
	P-DARTS (searched on CIFAR-10)	gradient	75.6%	92.6%	4.9×10^6
	PNASNet-5 ($N=4$, $F=216$)	SMBO	82.9%	96.2%	86.1×10^6
	Proxyless (GPU)	G + P	75.1%	92.5%	—
	One-Shot Top ($F=32$)	OS	75.2%	—	11.9×10^6

表6 NAS在目标检测任务的性能对比

Tab. 6 NAS performance comparison on object detection

数据集	方法	搜索策略	mAP/%		mAP [0.5,0.95]	分辨率
			Mini-val	Test-dev		
COCO	SSD-300	manual	—	23.2		320×320
	MnasNet-A1 + SSDLite	RL	—	23.0		320×320
	NAS-FPNLite MobileNet-V2	RL	—	24.2		320×320
	ShuffleNet (2x)	manual	24.5	—		600×600
	NASNet-A(4@1056)	RL	29.6	—		600×600
	FPNAmoebaNet@256	manual	—	43.4		1 280×1 280
	NAS-FPN R-50(7@384) + Dropbox	RL	—	46.6		1 280×1 280
	NAS-FPN AmoebaNet(7@384) + DropBlock	RL	—	48.3		1 280×1 280
PASCAL VOC 2007	Faster-RCNN with VGG16	manual	68.7		36.7	—
	AMC(base on Faster-RCNN with VGG16)	RL + P	68.8		37.2	—

Chen 等[46] 于2018年设计了一种基于神经网络架构搜索的分割模型,主要集中于设计适用于逐像素标记任务的搜索空间和快捷代理任务,例如候选网络中包含易于编码多尺度上下文信息的空洞卷积,以捕捉多尺度的信息。因为逐像素标注任务的特殊性,高分辨率图像对于传递多尺度信息至关重要,故不能通过降低训练图像的分辨率来加速搜索过程,于是作者在训练过程中使用预训练过的更小的骨干网络,保留之前训练集上产生的特征映射,并且使用早停策略使候选网络不必训练至收敛,将搜索耗时降至原来的1%以内。仅使用随机搜索的策略,在场景分割、人体分割和语义图像分割三项任务中的实验结果显示,网络搜索得到的网络结构性能均超过人工设计的神经网络,并且将所需的参数数量和计算资源减半。

然而随着搜索空间的增大,诸如强化学习、进化算法等智能搜索策略的使用可能会进一步提高搜索效率。于是 Liu 等[45] 为了使用梯度方法,设计了分层架构搜索空间,适配具备大量网络结构变体和对图像分辨率要求严格的密集图像预测任务,以期既控制内部单元级结构的逐层计算,又搜索外部网络级架构控制空间分辨率变化。对比以往的网络搜索技术在基本单元搜索出来之后就依照预设的模式堆叠,该方法实现了更广意义上的自动化搜索。同时通过梯度搜索策略开发出连续可微的公式,实现在两级分层架构上进行有效的搜索,搜索得到的最优结构 Auto-DeepLab-L 仅用低于50%的乘积累加操作就以8.6%超过原先最优结构的交集并集比(IoU),而搜索得到的轻量化模型也在测试集中达到80.9%的交集并集比,并且仅占用了 10.15×10^6 的参数量和 333.25×10^9 的乘积累加操作。神经网络架构搜索技术在 Part Labels 和 Cityscapes 数据集上的逐像素标注性能对比分别如图12和表7所示,主要的评价指标包括表征产生目标窗口与标记窗口的交叠率平均值的平均交并比(mean intersection over union, MIoU)、表征所有分类正确的像素数占像素总数的比例的像素准确率(pixel accuracy, PA)和降维之后的超像素准确率(super pixel accuracy, SPA)。

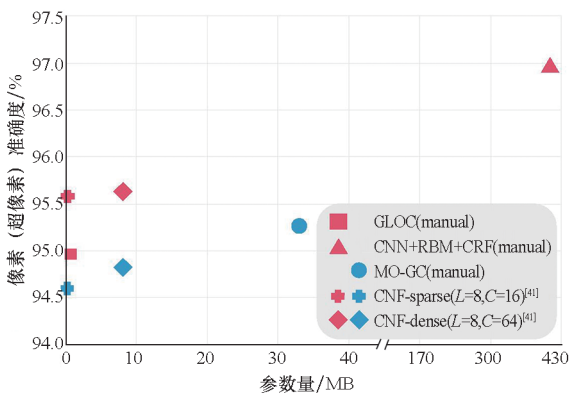


图12 NAS在Part Labels的数据分割任务性能对比

Fig. 12 Performance comparison of NAS in segmentation on Part Labels

5.3 多目标

现阶段深度神经网络性能不断提升,逐渐拥有对大数据高层特征提取的能力,同时随之而来

表 7　NAS 在 Cityscapes 的数据分割任务性能对比

Tab. 7　Performance comparison of NAS in segmentation on Cityscapes

方法	搜索策略	MIoU/%
ResNet-38	manual	80.6
PSPNet	manual	81.2
DeepLabV3 +	manual	82.1
DPC[46]	random	82.7
Auto-DeepLab-S[45]	gradient	80.9
Auto-DeepLab-L[45]	gradient	82.1
DCNAS[106]	gradient	83.6

的是深度网络模型的高存储、高功耗的弊端,需要巨大的计算开销和内存存储,阻碍了在如移动或嵌入式设备端等资源有限环境中的使用,例如自动驾驶车辆需要实时的行人检测,制约智能化移动嵌入式设备、现场可编程阵列等在线学习和识别任务。现阶段,很多神经网络架构搜索模型专注于优化模型的准确率,而忽略与底层硬件和设备适配,这些仅考虑高准确率的模型难以在资源有限的终端部署,这导致网络搜索可以取得好的实验结果,但并不一定适合实际部署,比如 DARTS[38] 能在很小的参数量下达到很好的精度,但是实际应用中,搜索得到网络的前向计算其实很慢。随着神经网络架构搜索的发展,研究人员从一开始只考虑精度的单目标到多目标,如同时考虑网络大小(参数量)、功耗、时延、计算复杂度(FLOPs/Multi-Adds)、内存占用等,实现多目标多任务应用的拓展。

期望搜索得到的网络结构特征可以概括为参数少、速度快、精度高。这一类的问题往往无法找到单一解让所有子目标同时最优,所以一般找帕累托最优解,即没有任何一个其他 X' 可以使得 X' 的所有优化目标值均小于 X 的优化目标值[43]。而且如果需要多方面权衡的话,搜索空间可能也会显著增大,进而增加搜索的难度。神经网络架构搜索技术在 ImageNet 数据集上对于不同多目标指标的性能对比详见表 8。

表 8　NAS 在多目标拓展上的性能对比

Tab. 8　NAS performance comparison on multi-target task

优化目标	运行环境	方法	搜索策略	目标值	准确率/%	
					Top-1	Top-5
参数量 Model Size		AlexNet[4]	manual	61×10^6	57.2	80.3
		MobileNet-V1[127]	manual	4.2×10^6	70.6	89.5
		AlexNet Compressed[66]	pruning	6.7×10^6	57.2	80.3
		VGG16 Compressed[66]	pruning	10.35×10^6	68.83	89.09
		MnasNet-A1[43]	RL	3.9×10^6	75.2	92.5
FLOPs		MobileNet-V1[127]	manual	569×10^6	70.9	89.5
		MobileNet-V2	manual	300×10^6	72.0	—
		AMC (0.5 × FLOPs)[64]	RL + P	285×10^6	70.5	89.3
		FBNet-A[61]	gradient	249×10^6	73.0	—
Latency	Google Pixel-1 CPU	MobileNet-V1[127]	manual	123 ms	70.9	89.5
		MnasNet[43]	RL	78 ms	75.2	92.5
		AMC (0.5 × Time)[64]	RL + P	63.3 ms	70.2	89.2
	BitFusion	MobileNet-V1[127]	manual	20.08 ms	70.82	89.85
		HAQ[76]	RL	11.09 ms	70.40	89.69
	Tesla V100	MobileNet-V2	manual	6.1 ms	72.0	91.0
		Proxyless[88]	G + P	5.1 ms	75.1	92.5
	Samsung Galaxy S8	MobileNet-V2	manual	21.7 ms	72.0	—
		FBNet-A[61]	gradient	19.8 ms	73.0	—
Energy	BitFusion	MobileNet-V1[127]	manual	31.03 mJ	70.82	89.95
		HAQ[76]	RL	16.30 mJ	70.37	89.40

1)模型尺寸。Han 等[66]基于文献[109]的工作通过剪枝的搜索策略,剔除冗余的连接,仅剪枝一步就可以分别将 VGG16 和 AlexNet 网络中的连接数减少到原来的 1/13 和 1/9,再结合网络压缩中的量化和编码进一步降低权值的存储要求。实验结果显示,在保持模型在 ImageNet 上的准确率的情况下,AlexNet 的存储量从原来的 240 MB 较小至 6.9 MB,VGG16 的存储量从 552 MB 优化至 11.3 MB,很大地促进了复杂网络在移动端的应用,特别是当移动端应用大小和网络带宽均受限的情况。并且随着模型大小的减小,实验结果表明时延和能耗方面也均得到了一定程度的优化。

2)FLOPs。浮点运算数,FLOPs 指的是模型的计算力消耗,一般可以视为计算时延的一个代理信号展开研究。He 等[64]主要结合强化学习和剪枝思想,提出 AMC 方法。以往细粒度的剪枝旨在剔除权向量中不重要的元素,尽管压缩率可观,但是所导致的不规则的压缩模式可能需要特制的硬件以加速网络运算。故作者选用粗粒度的结构化剪枝以剔除权向量中较完整的规则区域,譬如整个通道以及卷积核整行、整列等。为了精确地获得每一层网络效果最好的压缩率,作者采用强化学习中的深度确定性策略梯度法产生连续空间上的具体压缩率,并兼顾 FLOPs 和模型大小设计激励函数。实验结果显示,相比人工设计压缩的 VGG16 网络在 ImageNet 上的表现,作者的方法在 1/4 的 FLOPs 的条件下,将准确率提升了 2.7%;将 MobileNet 在 Google Pixel 1 上整整加速了近一倍。此外,也有工作直接将紧致设计的网络模块直接引入搜索空间中,Yan 等[101]、Xiong 等[128]和 Liu 等[129]分别引入具有更少参数和运算数的深度可分离卷积、倒置残差(inverted residuals)结构[50]和空洞卷积以完成轻量化网络的设计。

3)时延。计算时延是评价网络模型的一个重要指标,某些方法倾向于搭建具有复杂拓扑结构的网络,使得在实际使用中网络前向传播的时延也很长。因为该参数是不可微的,给搜索提出了更高的要求。而且很多工作[88]已经指出:在 GPU 上计算速度优化的网络结构在 CPU 或者移动设备上运算速度并不好,故针对计算资源有限设备的应用也亟待优化。Tan 等[43]直接提出时延的优化目标,指明之前的 FLOPs 只是一个代理参数,并且不能精确地反映现实世界中设备的计算时延。将神经网络结构搜索技术引入提升模型效率的领域,与典型的剪枝和量化相比,NAS 方法并不依附于现有的高性能网络结构。以往的研究为了在移动端部署,一般会减小网络的深度,并且使用深度卷积[127]和分组卷积[51]等开销较小的操作,尽管这样简化了搜索过程,但是也阻碍了层的多样性,进而伤害了网络的准确率和计算时延。作者设计了分解的层次搜索空间,将完整的卷积神经网络分解成独立的块,并且在每个块中独立地搜索候选操作和连接,使得不同的块中有不同的层次结构,而不是像之前的研究中仅仅搜索少部分复杂的基本单元,再简单地堆叠。实验结果显示,在 Google Pixel 1 上运行 ImageNet 的分类任务时,搜索得到的结构分别比 MobileNet-V2 和 NASNet 的准确率提升了 0.5% 和 1.2%,并且分别加速了 1.8 和 2.3 倍;Cai 等[88]提出的 ProxylessNAS 同样不像之前的研究一样重复地堆叠单元,而是允许每一个块都可以接受训练。鉴于先前研究先在小数据集上训练再迁移到大数据集上,无法权衡计算时延等硬件指标,研究人员利用神经网络架构搜索技术直接为待部署的硬件搜索最优的网络框架结构,包括 GPU、CPU 和移动设备,并且直接从目标的完整数据集中获得反馈信息,而不是使用低保真的代理信息。又因为计算时延是不可微分的优化目标,作者将计算时延建模为关于神经网络结构的连续函数,并视其为正则化损失项优化,解决了因为不同的硬件平台会有包括并行结构、缓存容量在内的诸多不同特性的问题,实现了针对不同硬件设计更适合的专门网络结构。但是 ProxylessNAS 对于每一个硬件平台都需要重头搜索,存在计算资源消耗的问题;后续 Cai 等[110]提出 OFA 方法,解耦训练和搜索过程,仅需训练一个网络,即可实现在多平台的直接部署。

$$E[lat_i] = \sum_j p_j^i \times F(o_j^i) \tag{8}$$

式中,$E[lat_i]$表示待搜索网络结构中第 i 个模块的估计时延,$F(\cdot)$表示时延预测模型,o_j 表示某一候选模块,p_j 即为选择该候选模块的概率。故可以通过$\frac{\partial E[lat_i]}{\partial p_j^i} = F(o_j^i)$实现梯度传递。结果显示,在 ImageNet 的分类任务中,在 Tesla V100 GPU 运行时,搜索得到的结构分别比 MobileNet-V2 的 Top-1 准确率提升了 3.1%,并且加速了 1.2 倍。计算时延的缩短将很大程度地推进实时视频分析等一系列任务的推广。

4)能耗。Dai 等[80]提出 ChamNet,在进化搜

索策略中综合考虑准确率、计算时延和能耗三方面因素,其中准确率和能耗均借助贝叶斯优化方法,而计算时延则直接查询一个包含超过 35 万条记录的查找表得到。在高通骁龙 835 CPU装置功率监控器上记录的实验数据显示,在降低 26% 能耗的同时提升 1.2% 的准确率,这一进步有利于帮助电池供电设备具有更长的运行时间。

随着对多目标的要求深入,目前的一大热门趋势是将经典的网络压缩方法和神经网络架构搜索技术相融合。以量化方法为例,Wang 等[76] 利用强化学习自动寻找在特定任务中的最优的量化策略,分别为网络中的每一层确定灵活精度的权值和激活值,得到一个经过优化的混合精度模型,并且性能超过拥有固定量化精度的网络结构。随着支持混合精度的芯片的持续开发,这项技术也将大展拳脚;Kim 等[130] 则直接专注于二值化网络的结构搜索,指出二值化网络和浮点型网络的差异也会导致网络搜索逻辑产生差别,譬如浮点型神经网络中效果较好的 Depthwise 卷积[127] 和 Pointwise 卷积[131] 可能在二值域中导致严重的梯度消失问题,以及归零层(zeroize layer)在二值域中会提升网络的准确率,却会在浮点域中起副作用等问题。作者通过对基本单元间增加跳跃连接、定义不同于浮点域中的搜索单元集合、设计多样性正则化抵消二值域中对某些单元的偏爱等手段实现了较好的二值网络搜索。

5.4　数据增强

数据增强是帮助提升图像分类器的准确率及增强泛化性能的一类技术,广泛地应用于深度学习中。具体的数据增强方法包括但不限于平移、旋转、镜像、剪切等。目前数据增强的瓶颈在于手工设计且欠缺通用性,例如在某一特定数据集上最优的数据增强方法可能并不适用于另一数据集,即对于特定的任务需要找到合适的数据增强手段;而且数据增强技术早年提出之后就鲜少有改进,譬如目前 ImageNet 上最主流的数据增强技术是 2012 年提出的。

Cubuk 等[77] 提出 AutoAugment,以强化学习为基础,针对不同数据集自动化地寻找更优的数据增强策略,在 ImageNet 和 CIFAR-10 上分别达到 83.5% 和 98.5% 的准确率,分别较之前的最优方法进一步提升了 0.4% 和 0.6%。实验结果再一次验证了不同数据集对应不同的最佳数据增强方法,譬如对于 CIFAR-10 数据集,AutoAugment 倾向于选择色彩补偿(equalize)、自动对比度(auto contrast)调整、明亮度(brightness)调整等颜色相关的变换。而对于 SVHN 数据集,翻转、旋转、裁剪等空间尺度上的变换能表现更好的性能。并且该模型具有较好的迁移能力,即在 ImageNet 上训练得到的增强策略在 Oxford Flowers、Caltech-101、Oxford-IIT Pets、FGVC Aircraft、Stanford Cars 等诸多数据集上均显著提升性能。

此后,Lim 等[132] 发现对于相对较小的数据集,都需要消耗大量的计算资源,故提出了 Fast AutoAugment,将增强图像视为训练数据集中的缺失图像,使用密度匹配(density matching)和贝叶斯思想,在不损失性能的情况下,加快数据增强策略的搜索速度:在同样的实验环境中,Fast AutoAugment 比 AutoAugment 针对 ImageNet 数据集的数据增强策略搜索快了 33 倍。展望未来,其他的搜索算法应用于这一领域可能带来更优的效果,并且将共同推动 AutoML 的发展。

5.5　语言模型

语言模型是预测文档中下一个单词或字符的一类任务,具有代表性的英文数据集有 Penn Treebank 和 WikiText-2。在深度学习中,语言模型一般是由以 LSTM 为代表的循环神经网络 RNN 处理的。循环神经网络的主要特点是当前单元中的隐藏状态 h_t 是由输入 i_t 和上一时刻中的隐藏状态 h_{t-1} 共同计算得到的,单元中包含的操作主要有 tanh、ReLU、sigmoid、补零(zeroize)、赋值(identity)等。

神经网络架构搜索技术不仅发展了卷积神经网络结构,也加速了循环神经网络领域的网络结构自动化设计:Zoph 等[33] 合理设计激励函数,运用强化学习的思想自动设计循环神经网络,在单词和字符预测中均取得了超越人工设计网络的性能,并且模型具有一定的迁移能力;So 等[49] 和 Liu 等[38] 分别基于进化算法和梯度方法搜索循环神经网络结构,并取得期望性能;Pham 等[40] 在文献[33]的基础上,利用权值共享技术大幅度地缩短了搜索耗时,仅一个 GPU 运行 10 h,即可得到期望的网络结构,并且该网络性能较之前研究进一步提升;后续一系列工作[57,81-84] 进一步优化搜索速度,促进语言领域结构搜索技术的应用。具体的 NAS 在语言模型上的性能如表 9 所示,困惑度(perplexity)刻画的是某个语言模型估计的一句话出现的概率,困惑度的值越小表示语言模型越好。

表9 语言模型任务性能对比

Tab. 9 Performance comparison on language modeling

数据集	方法	搜索策略	困惑度		参数量	搜索成本/GPU日
			验证	测试		
Penn Treebank	LSTM + DropConnect	manual	60.0	57.3	24×10^6	—
	Random searchWS[57]	random	57.8	55.5	23×10^6	2
	ENAS[40]	RL	60.8	58.6	24×10^6	0.5
	DARTS (first order)[38]	gradient	60.2	57.6	23×10^6	0.5
	DARTS (second order)[38]	gradient		55.7	23×10^6	1
	GDAS[86]	gradient		57.5	23×10^6	0.4
	NAONet-WS + weight tying + weight penalty[83]	gradient		56.6	27×10^6	0.4
WikiText-2	LSTM + skip connections	manual		65.9	24×10^6	—
	LSTM + 15 Softmax experts	manual		63.3	33×10^6	—
	ENAS[40]	RL	72.4	70.4	33×10^6	0.5
	DARTS[38]	gradient	71.2	69.2	33×10^6	1
	NAONet[83]	gradient	—	67.0	36×10^6	—

除了在本节上述任务中应用,近年来NAS技术也在动作分割(action segmentation)[133]、目标追踪(object tracking)[101]、对抗攻击(adversarial attack)[105]、姿态估计(pose estimation)[133-135]等任务上取得突破性进展。

6 对神经网络架构搜索研究的思考

近年来,受益于学界不断增长的关注度,神经网络架构搜索问题的研究也不断得到深入,性能不断提升,应用也不断得到拓展,但已有的神经网络架构搜索难以满足更高层次的实际应用需求,相关研究仍然任重道远。下面对神经网络架构搜索存在的问题及未来可能的研究方向进行总结。

1)当前神经网络架构搜索还不具备完全自行设计网络架构的能力。现阶段搜索空间的设计在一定程度上和神经网络架构搜索技术的自动化初衷相悖,未来应该提高搜索空间设计的自动化程度[39]。现有更高级的主流研究中,算法仍然仅基于人工限定较多的基本单元,再将基本单元堆叠得到网络结构作为搜索结果。以文献[35]为例,基本单元中每个子单元包含两个输入、两种操作和一个组合操作,共五个离散的待搜索参数,是否存在性能更佳的参数部署模板还未有定论;而粗略地分为标准单元和还原单元两类基本单元是否最优也亟待研究。因为理论上,既然基本单元之间可以被随意地连接,基本单元也应该自适应地改变。与此同时,虽然基本单元的表示方式极大地推动了相关技术的发展,但是基于基本单元的搜索空间设计主要是基于人类在图像分类方面的先验经验,并且不容易推广到语义分割、对象检测等其他领域,更加灵活的表示方式还亟待开发。由此引出未来关于神经网络架构搜索的另外一个关键发展方向即在更广泛的搜索空间寻找高效架构,这也对搜索策略和性能评估提出更高的要求,已经有研究人员[136-137]正准备提出元结构层面上的优化等其他方面的研究。

2)在同一标准限制下,急需挖掘更多的、有难度的基准,以方便方法之间的对比和评估。譬如在考虑多指标的情况下,搜索成本和计算时延均没有一个统一的运行环境或平台作为共同基准,不利于方法之间的比较和技术的后续发展;方法对于数据集的增强处理方法不尽相同且公开信息有限,而且不同数据集间的迁移能力也暂未有合理的度量方法。基准的制作完成对于合理公正地判断比较不同神经网络架构搜索算法的性能有极大的帮助,并且将最终推动整个技术的发展。

3)现有的神经网络架构搜索方法难以复现。原因包括:①一些方法需要数以月计的计算时间,对于大部分计算预算有限的个人研究人员而言显得遥不可及。②众多方法拥有不同的训练流程和搜索空间,使得各个方法之间很难相互借鉴。尽管目前已经提出了共同基准下的NAS-Bench-101数据集[126]比较不同神经网络架构搜索方法,但是仍然远远不够。③网络搜索的实验设置中存在

训练流程、正则化技巧等各类变数以及原始资料的缺失,很难准确再现其他搜索策略的性能。因为复现是科学进步的核心原则,Yu 等[97]提出在制作搜索方法的 baseline 时,应当更加严格地限定。故对于新提出的搜索方法,实验结果的稳健性需多花心思,并且详尽地表述关键的细节。

4)仍需拓展更多网络种类的设计。神经网络架构搜索目前主要集中于卷积神经网络的设计,而其他类型的网络也有参数和结构的搜索需求。譬如,包含生成网络和对抗网络的生成对抗网络 GAN、主要用于序列数据处理的 RNN 循环神经网络、由自注意力和前馈神经网络组成的 Transformer 网络等。不同网络类型具有的不同特点,例如循环神经网络在时间序列上共享参数,均对搜索提出了更大的要求。

5)仍需进一步拓展开源工具包及相关应用场景。当前 NAS 开源工具较少,仅有 AutoML 将 NAS 与传统的 ML 管道优化结合在一起,开发 Auto-Pytorch,联合稳健地优化网络架构和训练超参数,以实现全自动深度学习 (AutoDL)。除了在图像分类等领域上的应用之外,神经网络架构搜索的优点也有望推广到包括人脸识别、人体姿态估计、图像定位、图像修复、机器翻译以及数据融合(sensor fusion)等其他问题上。基于在图像分类领域的成就,未来神经网络架构搜索技术有望推动产业内一系列问题的进一步发展。

7　总结

神经网络架构搜索是深度学习发展到一定阶段所面临的一个必然问题,尽管 NAS 在诸多领域有着广泛的应用价值,也依然存在许多困难与挑战。本文从搜索空间、搜索策略和性能评估策略三个角度,系统性地阐述了神经网络架构搜索近年来的研究进展,详细梳理了主流搜索策略的优势和不足,并阐述了不同策略之间的内在联系。同时,本文以此为基础介绍了神经网络架构搜索技术的拓展应用,并展望了神经网络架构搜索及其相关领域未来的研究重点。

参考文献(References)

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

[2] SEJNOWSKI T J. The unreasonable effectiveness of deep learning in artificial intelligence[J]. Proceedings of the National Academy of Sciences, 2020, 117(48): 30033 – 30038.

[3] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014: 580 – 587.

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

[5] LIU L, CHEN J, FIEGUTH P, et al. From BoW to CNN: two decades of texture representation for texture classification[J]. International Journal of Computer Vision, 2019, 127(1): 74 – 109.

[6] LIU L, OUYANG W L, WANG X G, et al. Deep learning for generic object detection: a survey[J]. International Journal of Computer Vision, 2020, 128(2): 261 – 318.

[7] HINTON G, DENG L, YU D, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82 – 97.

[8] CHEN C Y, SEFF A, KORNHAUSER A, et al. DeepDriving: learning affordance for direct perception in autonomous driving[C]//Proceedings of IEEE International Conference on Computer Vision (ICCV), 2015.

[9] 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.

[10] WU Y H, SCHUSTER M, CHEN Z F, et al. Google's neural machine translation system: bridging the gap between human and machine translation[EB/OL]. (2016 – 10 – 08)[2022 – 02 – 01]. https://arxiv.org/abs/1609.08144.

[11] 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.

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

[13] SILVER D, HUBERT T, SCHRITTWIESER J, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play[J]. Science, 2018, 362(6419): 1140 – 1144.

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

[15] TAIGMAN Y, YANG M, RANZATO M, et al. DeepFace: closing the gap to human-level performance in face verification[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1701 – 1708.

[16] DEVRIES P M R, VIÉGAS F, WATTENBERG M, et al. Deep learning of aftershock patterns following large earthquakes[J]. Nature, 2018, 560(7720): 632 – 634.

[17] STOKES J M, YANG K, SWANSON K, et al. A deep learning approach to antibiotic discovery[J]. Cell, 2020, 181(2): 475 – 483.

[18] SCHMIDHUBER J. Deep learning in neural networks: an overview[J]. Neural Networks, 2015, 61: 85 – 117.

[19] LOWE D G. Object recognition from local scale-invariant features[C]//Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.

[20] FREEMAN W T, ROTH M. Orientation histograms for hand gesture recognition[C]//Proceedings of International Workshop on Automatic Face and Gesture Recognition, 1995.

[21] ZEILER M D, FERGUS R. Visualizing and understanding convolutional networks [C]//Proceedings of European Conference on Computer Vision, 2014: 818 – 833.

[22] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2015 – 04 – 10)[2022 – 02 – 01]. https://arxiv.org/abs/1409.1556.

[23] SZEGEDY C, LIU W, JIA Y Q, et al. Going deeper with convolutions [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[24] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[25] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[26] XIE S N, GIRSHICK R, DOLLÁR P, et al. Aggregated residual transformations for deep neural networks [C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[27] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211 – 252.

[28] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the Inception architecture for computer vision [C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[29] ZAGORUYKO S, KOMODAKIS N. Wide residual networks[C]// Proceedings of British Machine Vision Conference, 2016.

[30] ZHANG X C, LI Z Z, LOY C C, et al. PolyNet: a pursuit of structural diversity in very deep networks[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[31] YAO X. Evolving artificial neural networks[J]. Proceedings of the IEEE, 1999, 87(9): 1423 – 1447.

[32] STANLEY K O, MIIKKULAINEN R. Evolving neural networks through augmenting topologies [J]. Evolutionary Computation, 2002, 10(2): 99 – 127.

[33] ZOPH B, LE Q V. Neural architecture search with reinforcement learning[EB/OL]. (2017 – 02 – 15)[2022 – 02 – 01]. https://arxiv.org/abs/1611.01578.

[34] KLEIN A, FALKNER S, SPRINGENBERG J T, et al. Learning curve prediction with Bayesian neural networks[C]// Proceedings of International Conference on Learning Representations, 2017.

[35] ZOPH B, VASUDEVAN V, SHLENS J, et al. Learning transferable architectures for scalable image recognition[C]// Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018: 8697 – 8710.

[36] REAL E, AGGARWAL A, HUANG Y P, et al. Regularized evolution for image classifier architecture search [C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2019: 4780 – 4789.

[37] LIU C X, ZOPH B, NEUMANN M, et al. Progressive neural architecture search[C]//Proceedings of European Conference on Computer Vision, 2018.

[38] LIU H X, SIMONYAN K, YANG Y M. DARTS: differentiable architecture search[EB/OL]. (2019 – 04 – 23)[2022 – 02 – 01]. https://arxiv.org/abs/1806.09055.

[39] BROCK A, LIM T, RITCHIE J M, et al. SMASH: one-shot model architecture search through HyperNetworks [C]// Proceedings of the 6th International Conference on Learning Representations, 2018.

[40] PHAM H, GUAN M Y, ZOPH B, et al. Efficient neural architecture search via parameter sharing[EB/OL]. (2018 – 12 – 12) [2022 – 02 – 01]. https://arxiv.org/abs/1802.03268.

[41] ZHONG Z, YAN J J, WU W, et al. Practical block-wise neural network architecture generation[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[42] KANDASAMY K, NEISWANGER W, SCHNEIDER J, et al. Neural architecture search with Bayesian optimisation and optimal transport[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018.

[43] TAN M X, CHEN B, PANG R M, et al. MnasNet: platform-aware neural architecture search for mobile[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[44] GHIASI G, LIN T Y, LE Q V. NAS-FPN: learning scalable feature pyramid architecture for object detection [C]// Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[45] LIU C X, CHEN L C, SCHROFF F, et al. Auto-DeepLab: hierarchical neural architecture search for semantic image segmentation[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[46] CHEN L C, COLLINS M D, ZHU Y K, et al. Searching for efficient multi-scale architectures for dense image prediction[EB/OL]. (2018 – 09 – 11)[2022 – 02 – 01]. https://arxiv.org/abs/1809.04184.

[47] ELSKEN T, METZEN J H, HUTTER F. Neural architecture search: a survey [J]. The Journal of Machine Learning Research, 2019, 20(1): 1997 – 2017.

[48] DENG L, LI G Q, HAN S, et al. Model compression and hardware acceleration for neural networks: a comprehensive survey[J]. Proceedings of the IEEE, 2020, 108(4): 485 – 532.

[49] SO D, LE Q, LIANG C. The evolved transformer [C]// Proceedings of the 36th International Conference on Machine Learning, 2019.

[50] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]// Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[51] ZHANG X Y, ZHOU X Y, LIN M X, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices [C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[52] IANDOLA F N, HAN S, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size[EB/OL]. (2016 – 11 – 04) [2022 – 02 – 01]. https://arxiv.org/abs/1602.07360.

[53] LIN M, CHEN Q, YAN S C. Network in network[EB/OL]. (2014 – 03 – 04)[2022 – 02 – 01]. https://arxiv.org/abs/1312.4400.

[54] CHOLLET F. Xception: deep learning with depthwise separable convolutions[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[55] WEI T, WANG C H, RUI Y, et al. Network morphism[C]//Proceedings of the 33rd International Conference on Machine Learning, 2016.

[56] CAI H, CHEN T Y, ZHANG W N, et al. Efficient architecture search by network transformation [C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

[57] LI L, TALWALKAR A. Random search and reproducibility for neural architecture search[C]//Proceedings of the 35th Uncertainty in Artificial Intelligence Conference, 2020.

[58] RADOSAVOVIC I, JOHNSON J, XIE S N, et al. On network design spaces for visual recognition[C]//Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[59] LIU H X, SIMONYAN K, VINYALS O, et al. Hierarchical representations for efficient architecture search[EB/OL]. (2018-02-22)[2022-02-01]. https://arxiv.org/abs/1711.00436.

[60] BAKER B, GUPTA O, RASKAR R, et al. Accelerating neural architecture search using performance prediction[EB/OL]. (2017-11-08)[2022-02-01]. https://arxiv.org/abs/1705.10823.

[61] WU B C, DAI X L, ZHANG P Z, et al. FBNet: hardware-aware efficient ConvNet design via differentiable neural architecture search [C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[62] BI K F, HU C P, XIE L X, et al. Stabilizing DARTS with amended gradient estimation on architectural parameters[EB/OL]. (2020-05-04)[2022-02-01]. https://arxiv.org/abs/1910.11831.

[63] LIANG H W, ZHANG S F, SUN J C, et al. DARTS+: improved differentiable architecture search with early stopping[EB/OL]. (2020-10-20)[2022-02-01]. https://arxiv.org/abs/1909.06035.

[64] HE Y H, LIN J, LIU Z J, et al. AMC: AutoML for model compression and acceleration on mobile devices [C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[65] FEDOROV I, ADAMS R P, MATTINA M, et al. SpArSe: sparse architecture search for CNNs on resource-constrained microcontrollers[EB/OL]. (2019-05-28)[2022-02-01]. https://arxiv.org/abs/1905.12107.

[66] HAN S, MAO H Z, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[EB/OL]. (2016-02-15)[2022-02-01]. https://arxiv.org/abs/1510.00149.

[67] LOUIZOS C, WELLING M, KINGMA D P. Learning sparse neural networks through L_0 regularization[EB/OL]. (2018-06-22)[2022-02-01]. https://arxiv.org/abs/1712.01312.

[68] WANG Y L, ZHANG X L, XIE L X, et al. Pruning from scratch[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12273-12280.

[69] HUANG Z H, WANG N Y. Data-driven sparse structure selection for deep neural networks[C]//Proceedings of the European Conference on Computer Vision, 2018.

[70] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[EB/OL]. (2017-03-10)[2022-02-01]. https://arxiv.org/abs/1608.08710.

[71] HE Y, KANG G L, DONG X Y, et al. Soft filter pruning for accelerating deep convolutional neural networks [C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018: 2234-2240.

[72] HE Y H, ZHANG X Y, SUN J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of IEEE International Conference on Computer Vision (ICCV), 2017.

[73] LIU Z, SUN M J, ZHOU T H, et al. Rethinking the value of network pruning[EB/OL]. (2019-03-05)[2022-02-01]. https://arxiv.org/abs/1810.05270.

[74] LUO J H, WU J X, LIN W Y. ThiNet: a filter level pruning method for deep neural network compression [C]//Proceedings of IEEE International Conference on Computer Vision (ICCV), 2017.

[75] BAKER B, GUPTA O, NAIK N, et al. Designing neural network architectures using reinforcement learning[EB/OL]. (2017-03-22)[2022-02-01]. https://arxiv.org/abs/1611.02167.

[76] WANG K, LIU Z J, LIN Y J, et al. HAQ: hardware-aware automated quantization with mixed precision [C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[77] CUBUK E D, ZOPH B, MANÉ D, et al. AutoAugment: learning augmentation strategies from data[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[78] REAL E, MOORE S, SELLE A, et al. Large-scale evolution of image classifiers[C]//Proceedings of the 34th International Conference on Machine Learning, 2017.

[79] ELSKEN T, METZEN J H, HUTTER F. Efficient multi-objective neural architecture search via Lamarckian evolution[EB/OL]. (2019-11-26)[2022-02-01]. https://arxiv.org/abs/1804.09081.

[80] DAI X L, ZHANG P Z, WU B C, et al. ChamNet: towards efficient network design through platform-aware model adaptation[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[81] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding[EB/OL]. (2019-05-24)[2022-02-01]. https://arxiv.org/abs/1810.04805.

[82] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

[83] LUO R Q, TIAN F, QIN T, et al. Neural architecture optimization [C]//Proceedings of the 32nd International Conference on neural Information Processing Systems, 2018.

[84] XIE S R, ZHENG H H, LIU C X, et al. SNAS: stochastic neural architecture optimization search[EB/OL]. (2020-04-01)[2022-02-01]. https://arxiv.org/abs/1812.09926.

[85] ZHANG X B, HUANG Z H, WANG N Y, et al. You only search once: single shot neural architecture search via direct sparse optimization [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(9): 2891-2904.

[86] DONG X Y, YANG Y. Searching for a robust neural architecture in four GPU hours[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[87] CHEN X, XIE L X, WU J, et al. Progressive differentiable architecture search: bridging the depth gap between search and evaluation[C]//Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[88] CAI H, ZHU L G, HAN S. ProxylessNAS: direct neural architecture search on target task and hardware[EB/OL]. (2019-11-23)[2022-02-01]. https://arxiv.org/abs/1812.00332.

[89] HUTTER F, HOOS H H, LEYTON-BROWN K, et al. Time-bounded sequential parameter optimization[C]//International Conference on Learning and Intelligent Optimization, 2010.

[90] ZHANG C, REN M Y, URTASUN R. Graph HyperNetworks for neural architecture search[EB/OL]. (2020-12-18)[2022-02-01]. https://arxiv.org/abs/1810.05749.

[91] BENDER G, KINDERMANS P J, ZOPH B, et al. Understanding and simplifying one-shot architecture search[C]//Proceedings of the 35th International Conference on Machine Learning, 2018.

[92] SAXENA S, VERBEEK J. Convolutional neural fabrics[EB/OL]. (2017-01-30)[2022-02-01]. https://arxiv.org/abs/1606.02492.

[93] HUANG G, CHEN D, LI T H, et al. Multi-scale dense convolutional networks for efficient prediction[EB/OL]. (2018-06-07)[2022-02-01]. https://arxiv.org/abs/1703.09844v2.

[94] YANG L, HAN Y Z, CHEN X, et al. Resolution adaptive networks for efficient inference[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[95] GUO Z C, ZHANG X Y, MU H Y, et al. Single path one-shot neural architecture search with uniform sampling[C]//Proceedings of the European Conference on Computer Vision, 2020.

[96] BERGSTRA J, BENGIO Y. Random search for hyper-parameter optimization[J]. Journal of Machine Learning Research, 2012, 13: 281-305.

[97] YU K C, SCIUTO C, JAGGI M, et al. Evaluating the search phase of neural architecture search[EB/OL]. (2019-11-22)[2022-02-01]. https://arxiv.org/abs/1902.08142.

[98] NEGRINHO R, GORDON G. DeepArchitect: automatically designing and training deep architectures[EB/OL]. (2017-04-28)[2022-02-01]. https://arxiv.org/abs/1704.08792.

[99] WISTUBA M. Finding competitive network architectures within a day using UCT[EB/OL]. (2018-07-23)[2022-02-01]. https://arxiv.org/abs/1712.07420.

[100] SU X, HUANG T, LI Y X, et al. Prioritized architecture sampling with Monte-Carlo tree search[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[101] YAN B, PENG H W, WU K, et al. LightTrack: finding lightweight neural networks for object tracking via one-shot architecture search[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[102] 牛瑞丞. 基于进化算法的神经网络结构搜索[D]. 昆明: 云南大学, 2020.
NIU R C. Neural architecture search based on evolutionary algorithms[D]. Kunming: Yunnan University, 2020. (in Chinese)

[103] 梁峰, 董名, 田志超, 等. 面向轻量化神经网络的模型压缩与结构搜索[J]. 西安交通大学学报, 2020, 54(11): 106-112.
LIANG F, DONG M, TIAN Z C, et al. Model compression and structure search for lightweight neural network[J]. Journal of Xi'an Jiaotong University, 2020, 54(11): 106-112. (in Chinese)

[104] 胡文玥. 基于演化优化的神经网络结构搜索方法研究[D]. 上海: 华东师范大学, 2021.
HU W Y. The research of neural network structure search based on evolutionary optimization[D]. Shanghai: East China Normal University, 2021. (in Chinese)

[105] HOSSEINI R, YANG X Y, XIE P T. DSRNA: differentiable search of robust neural architectures[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[106] ZHANG X, XU H M, MO H, et al. DCNAS: densely connected neural architecture search for semantic image segmentation[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[107] 张宝丰. 基于梯度优化的神经网络结构搜索算法研究[D]. 南京: 南京邮电大学, 2021.
ZHANG B F. Research on neural architecture search based on gradient optimization[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2021. (in Chinese)

[108] 赵亮, 方伟. 一种快速渐进式卷积神经网络结构搜索算法[J]. 计算机工程, 2022, 48(12): 134-139, 149.
ZHAO L, FANG W. A fast and progressive convolutional neural architecture search algorithm[J]. Computer Engineering, 2022, 48(12): 134-139, 149. (in Chinese)

[109] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[C]//Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015.

[110] CAI H, GAN C, WANG T Z, et al. Once-for-all: train one network and specialize it for efficient deployment[EB/OL]. (2020-04-29)[2022-02-01]. https://arxiv.org/abs/1908.09791.

[111] YANG Z H, WANG Y H, CHEN X H, et al. CARS: continuous evolution for efficient neural architecture search[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[112] ZELA A, KLEIN A, FALKNER S, et al. Towards automated deep learning: efficient joint neural architecture and hyperparameter search[EB/OL]. (2018-07-18)[2022-02-01]. https://arxiv.org/abs/1807.06906.

[113] KLEIN A, FALKNER S, BARTELS S, et al. Fast Bayesian hyperparameter optimization on large datasets[J]. Electronic Journal of Statistics, 2017, 11(2): 4945-4968.

[114] CHRABASZCZ P, LOSHCHILOV I, HUTTER F. A downsampled variant of ImageNet as an alternative to the CIFAR datasets[EB/OL]. (2017-08-23)[2022-02-01]. https://arxiv.org/pdf/1707.08819.pdf.

[115] REAL E, AGGARWAL A, HUANG Y P, et al. Aging evolution for image classifier architecture search[C]//Proceedings of AAAI Conference on Artificial Intelligence,

2019.

[116] FALKNER S, KLEIN A, HUTTER F. BOHB: robust and efficient hyperparameter optimization at scale [EB/OL]. (2018 - 07 - 04)[2022 - 02 - 01]. https://arxiv.org/abs/1807.01774.

[117] DOMHAN T, SPRINGENBERG J T, HUTTER F. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves[C]//Proceedings of the 24th International Conference on Artificial Intelligence, 2015.

[118] SWERSKY K, SNOEK J, ADAMS R P. Freeze-thaw Bayesian optimization[EB/OL]. (2014 - 06 - 16)[2022 - 02 - 01]. https://arxiv.org/abs/1406.3896.

[119] WHITE C, NEISWANGER W, SAVANI Y. BANANAS: Bayesian optimization with neural architectures for neural architecture search[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(12): 10293 - 10301.

[120] SHAHRIARI B, SWERSKY K, WANG Z Y, et al. Taking the human out of the loop: a review of Bayesian optimization[J]. Proceedings of the IEEE, 2015, 104(1): 148 - 175.

[121] 谢彪. 基于强化学习和参数共享的神经网络结构搜索[D]. 成都: 西南财经大学, 2021.
XIE B. Neural network structure search based on reinforcement learning and parameter sharing[D]. Chengdu: Southwestern University of Finance and Economics, 2021. (in Chinese)

[122] ELSKEN T, METZEN J H, HUTTER F. Simple and efficient architecture search for convolutional neural networks [EB/OL]. (2017 - 11 - 13)[2022 - 02 - 01]. https://arxiv.org/abs/1711.04528.

[123] JIN H F, SONG Q Q, HU X. Auto-Keras: an efficient neural architecture search system [EB/OL]. (2020 - 12 - 18)[2022 - 02 - 01]. https://arxiv.org/abs/1806.10282.

[124] CAI H, YANG J C, ZHANG W N, et al. Path-level network transformation for efficient architecture search [C]// Proceedings of the 35th International Conference on Machine Learning, 2018.

[125] YAN Z C, DAI X L, ZHANG P Z, et al. FP-NAS: fast probabilistic neural architecture search[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[126] YING C, KLEIN A, CHRISTIANSEN E, et al. NAS-bench-101: towards reproducible neural architecture search[C]// Proceedings of the 36th International Conference on Machine Learning, 2019.

[127] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017 - 04 - 17)[2022 - 02 - 01]. https://arxiv.org/abs/1704.04861.

[128] XIONG Y Y, LIU H X, GUPTA S, et al. MobileDets: searching for object detection architectures for mobile accelerators[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[129] LIU J, LI C M, LIANG F, et al. Inception convolution with efficient dilation search [C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[130] KIM D, SINGH K P, CHOI J. Learning architectures for binary networks [C]//Proceedings of the European Conference on Computer Vision, 2020.

[131] HUA B S, TRAN M K, YEUNG S K. Pointwise convolutional neural networks[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[132] LIM S, KIM I, KIM T, et al. Fast AutoAugment[EB/OL]. (2019 - 05 - 25)[2022 - 02 - 01]. https://arxiv.org/abs/1905.00397.

[133] GAO S H, HAN Q, LI Z Y, et al. Global2local: efficient structure search for video action segmentation [C]// Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[134] CHEN Z R, HUANG Y, YU H Y, et al. Towards part-aware monocular 3D human pose estimation: an architecture search approach[C]// Proceedings of the European Conference on Computer Vision, 2020.

[135] XU L M, GUAN Y D, JIN S, et al. ViPNAS: efficient video pose estimation via neural architecture search [C]// Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[136] SHAW A, WEI W, LIU W Y, et al. Meta architecture search[C]// Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019.

[137] RADOSAVOVIC I, KOSARAJU R P, GIRSHICK R, et al. Designing network design spaces[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.