

文章编号: 2095-2163(2020)06-0025-06

中图分类号: TP301.6

文献标志码: A

# 神经网络架构搜索综述

耿飞, 王春楠, 王宏志

(哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150001)

**摘要:** 神经网络架构搜索是近年来兴起的自动化机器学习的重要组成部分。神经网络架构搜索旨在通过人为进行一些限定, 让机器能够在一定范围内自动对输入的深度学习任务进行自动模型构建和性能评估, 并最终产生一个针对该研究问题的神经网络架构, 作为整个算法的产出, 从而达到降低人工干预, 实现机器自动化, 获取高性能模型的目的。本文对近年来国内外的研究进行总结分析, 旨在给初入本领域的科研人员一个可以参照的指导性文章, 帮助科研人员快速入门并深入研究。

**关键词:** 神经网络架构搜索; 自动化机器学习; 深度学习

## Neural Architecture Search: A survey

GENG Fei, WANG Chunnan, WANG Hongzhi

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

**【Abstract】** Neural architecture search is an important part of the long-term rise of automated machine learning. Neural network architecture search gradually defines some artificially, so that the machine can automatically carry out model and performance evaluation of the input deep learning task within a certain range, and finally aim at a neural network architecture that studies the problem as the entire algorithm output, to reduce manual intervention and achieve the purpose of machine automation to obtain high-performance models. This article summarizes and analyzes the research domestic and foreign in recent years, and aims to give the first-time researchers a guide article that can be referenced to help researchers quickly get started and conduct in-depth research.

**【Key words】** Neural Architecture Search; Auto Machine Learning; Deep Learning

## 0 引言

深度学习的成功,很大程度上归功于端到端的学习方式和其特征工程的自动化。然而随着数据量的日益增加和数据的复杂化,需要人们手工设计的深度学习模型结构愈发复杂,然而神经网络架构的设计本身就是一个很困难的问题,并且设计工作并没有强有力的理论支撑和直觉的指导依据,所以自动生成网络结构模型能够在很大程度上解放人力,并有可能给人类手工设计模型提供思路和指导意见。本课题旨在用机器设计取代人工设计获得新模型,最终得到效果优于人类手工设计的神经网络结构。

随着2016年Google DeepMind团队的围棋AI——AlphaGo的大获成功,强化学习(RL)被越来越多的研究者所重视,Google团队也于该年年底将强化学习应用到了另一个研究问题——神经网络架构搜索NAS(Neural Architecture Search)上来,其自动搜索到的网络模型超越了同时期同类手工设计网络。作为自动化机器学习(AutoML)目前最为重要的研究分枝,有越来越多的机构学者致力于研究此

问题。

如图1所示,神经网络架构搜索的基本流程:搜索策略从预定义的搜索空间 $\mathcal{A}$ 中选择具体结构 $A$ ,该结构被性能评估模块评估,并将对 $A$ 的性能估计返回到搜索策略,指导下一次的结构选取。最终算法产生一个满足预设性能要求的模型 $A^*$ ,作为问题的输出。



图1 神经网络架构搜索

Fig. 1 Neural architecture search

## 1 国外研究综述

在时间维度上,MIT和谷歌大脑几乎在2016年底同一时间提出了基于强化学习(RL)的神经网络架构搜索算法,超过人类手工设计模型,但搜索耗时巨大,上千的GPU天训练时间让独立研究者和小型机构望而却步。2017年初,Google的论文Large-Scale Evolution of Image Classifiers采用进化算法来解决神经网络结构搜索问题<sup>[1]</sup>。

**作者简介:** 耿飞(1995-),男,硕士研究生,主要研究方向:神经网络架构搜索、深度学习;王春楠(1995-),女,博士研究生,主要研究方向:神经网络架构搜索、自动机器学习;王宏志(1978-),男,博士,教授,博士生导师,主要研究方向:大数据、数据库、机器学习等。

收稿日期: 2020-03-09

哈尔滨工业大学主办 ● 学术研究与应用

为了解决资源消耗过大的问题,很多研究机构从不同角度提出了优化方法。在 Google 的 ENAS 文章中,One-shot 方法被提出,所有子模型共享权重,搜索性能大大提升,效率提升千倍<sup>[2]</sup>。CMU 和 Google DeepMind 合作提出的 DARTS 方法,通过将网络结构与 softmax 进行结合,将搜索空间变为连续,进而可以用梯度方式优化目标问题,可微的搜索方法首次被提出<sup>[3]</sup>。

根据问题的定义,各国研究者主要从以下 3 个方面对神经网络架构搜索问题进行了探究:

(1) 搜索空间 (Search Space)。搜索空间定义了原则上可以表示哪些模型结构。针对不同的学习任务,搜索空间的先验知识也不同。然而,这容易引入人为偏见,可能会导致计算机生成的结构与人类设计结构类似,导致并不能发现新颖的结构。

(2) 搜索策略 (Search Strategy)。搜索策略详细说明了如何探索搜索空间。如何从搜索空间中确定这次要挑选的结构 A 和通过已有的结构和对应的评估结果,如何进行下一次的结构挑选。

(3) 性能评估策略 (Performance Estimation Strategy)。给定结构如何对结构进行性能评估,最为简单直接的办法就是用该结构所代表的网络模型从头至尾完成完整的训练过程,获得结果(例如图像分类的准确率)。性能评估更看重的网络模型之间的相对排序,而不是评估结果的绝对值。

### 1.1 搜索空间 (Search Space)

搜索空间决定了神经网络搜索框架可以发现的神经网络模式和范围。一个相对简单的搜索空间是链结构神经网络的空间。链结构神经网络结构 A 可以写成  $n$  层序列,其中第  $i$  层  $L_i$  从层  $i-1$  接收其输入,并且其输出用作层  $i+1$  的输入,即  $A = L_n \cdot \dots \cdot L_1 \cdot L_0$ 。然后通过以下方式对搜索空间进行参数化:(1) (最大) 层数  $n$  (可能无界)。(2) 每层可以执行的操作类型,例如:池化、卷积及一些高级卷积,如深度可分离卷积或扩张卷积;(3) 与操作相关的超参数。例如:滤波器的数量,内核大小和卷积层的步幅,或简单地用于全连接网络的模块数量。

近期关于 NAS 的工作结合了现有的设计元素,这些元素来自手工的结构,例如:skip-connection,可以构建复杂的多分支网络。在这种情况下,层  $i$  的输入可以形式地描述为组合先前层输出的函数  $g_i(L_{i-1}^{out}, \dots, L_0^{out})$ 。采用这种设计可以显著提高网络复杂度。这些多分支架构的特殊情况是:(1) 链结构网络(通过设置  $g_i(L_{i-1}^{out}, \dots, L_0^{out}) = L_{i-1}^{out}$ );(2) 残余

网络,其中前一层输出求和( $g_i(L_{i-1}^{out}, \dots, L_0^{out}) = L_{i-1}^{out} + L_j^{out}, j < i$ );(3) DenseNets, 其中前一层输出被连接( $g_i(L_{i-1}^{out}, \dots, L_0^{out}) = \text{concat}(L_{i-1}^{out}, \dots, L_0^{out})$ )。

目前主流的方式是搜索由人工设计的模块 (Cell 或 Block),然后将模块进行堆叠,而不是去搜索整个架构<sup>[4-5]</sup>。Zoph 等人优化两种不同类型的模块:保持输入维度的正常模块 (Normal Cell),减少特征图尺寸的降维模块 (Reduction Cell)。通过以预定义的方式堆叠这些模块来构建最终的神经网络。与上面讨论的相比,这个搜索空间有两个主要优点:

(1) 由于模块相对整个神经网络较小,因此搜索空间的大小显著减小。

(2) 通过调整模型中使用的模块数量,可以更容易地将模块转移到其他数据集。例如:在小数据集使用 8 个模块堆叠,大数据将该模块堆叠 20 次。

因此,后来这种基于模块的搜索空间成功地应用于许多工作。然而,当使用基于模块的搜索空间时,会出现新的问题,即如何设计模块结构:应该使用多少个模块以及如何连接它们来构建实际模型?例如:Zoph 等人从模块构建一个连续模型,其中每个模块接收前两个模块的输出作为输入,而 Cai 等人采用众所周知的手工设计架构的高级结构,如 DenseNet,并在这些模型中使用它们的模块。原则上,可以通过简单地用模块替换层来任意组合模块,多分支空间便可简单实现这个功能。理想情况下,模块应作为 NAS 的一部分自动优化。除此之外,复杂的模块设计可以增加模型生成的复杂度,从而对模型搜索算法的要求有所降低。

搜索空间的选择在很大程度上决定了优化问题的难度:即使对于具有固定模块的单个模块的搜索空间的情况,优化问题仍然是(1)不连续。(2) 相对较高的维度(因为更复杂的模型往往表现更好,导致设计更复杂的模块)。

### 1.2 搜索策略 (Search Strategy)

许多不同的搜索策略可用于探索神经架构的空间,包括随机搜索、贝叶斯优化、进化方法、强化学习 (Reinforcement learning, RL) 和基于梯度的方法。

为了将 NAS 问题转化为强化学习 (RL) 问题,可以将神经架构的生成视为代理动作,其中动作空间与搜索空间相同。代理的奖励基于对未见数据的训练架构的性能估计。不同的 RL 方法在如何表示代理的策略以及如何优化它们方面存在差异:Zoph 和 Le 使用递归神经网络 (RNN) 来顺序地对字符串进行采样,而字符串又对应于神经结构的编码;贝克

等人使用 Q-learning 来训练一个顺序选择算子类型和相应超参数的策略。这些方法的另一种观点是作为顺序决策过程,其中策略采样动作按照顺序生成网络结构,环境的“状态”包含到目前为止采样的动作的评估,并且奖励仅在最终完成动作之后获得。由于在此顺序过程中,没有发生与环境的交互(没有观察到外部状态,也没有中间奖励),将架构抽样过程解释为单个动作的顺序生成更直观,这将 RL 问题简化为多臂老虎机问题。

使用 RL 的替代方案是使用进化算法来优化神经结构的选取方法。第一种设计神经网络的方法可以追溯到近三十年:Miller 等使用遗传算法生成新结构并使用反向传播来优化其权重。许多研究者使用遗传算法,来优化神经结构及其权重。然而,当扩展到具有数百万权重的深度神经结构时,基于 SGD 的权重优化方法优于进化的方法。因此最近的神经进化方法,使用基于梯度的方法来优化权重,使用进化算法来优化神经结构本身<sup>[25-26]</sup>。进化算法维护一组模型,即一组(可能是训练过的)网络架构。在每个进化步骤中,对来自群体的至少一个模型进行采样,并通过对已有种群进行交叉变异生成新的子代。在 NAS 中,进化操作包括添加或移除层,改变层的超参数,添加跳过连接,以及改变训练超参数。在对后代进行训练之后,评估它们的适应性(例如,在验证集上的准确率)并将它们添加到种群中。

进化方法主要区别在如何选择初始种群,更新种群和产生后代方面存在差异。例如,Real 等人以及刘等人使用锦标赛选择<sup>[29]</sup>来抽样父母;Elsken 等人采用来自多目标帕累托前沿的样本父母;Real 等人从人口中删除最差的个体,发现去除最老的个体是有益的;刘等人不删除个体。为了生成后代,大多数方法随机初始化子网络,而 Elsken 等人采用继承的方式,即知识(以学习权重的形式)通过使用网络态射,从父网络传递给其子网络;Real 等人也让后代继承其父母不受突变影响的所有参数;虽然这种继承不严格继承性能,但与随机初始化相比,它也可以加速学习过程。此外,它们还允许改变学习速率,这可以被视为在 NAS 期间优化学习速率调度的方式。

Real 等人进行了 RL 进化和随机搜索(RS)的对比研究,得出的结论是 RL 和进化在最终测试准确性方面表现同样出色。随着时间的推移,进化具有更好的性能并且找到更小的模型。两种方法在实验中始终表现优于 RS,但其差别并不十分明显:RS

在 CIFAR-10 上实现了约 4% 的测试误差,而 RL 和进化达到约 3.5%。刘等人的差异甚至更小,其报告 CIFAR 10 测试误差为 3.9%,ImageNet RS 测试误差为 21.0%,而基于进化的方法分别为 3.75% 和 20.3%。

与上述无梯度优化方法相比,刘等人提出转变搜索空间到连续范围内,以实现基于梯度的优化:不要在特定层选定单个操作  $o_i$  (例如,卷积或池化),而是从一组  $\{o_1, \dots, o_m\}$  中计算凸组合。更具体地,给定输入  $x$ ,输出  $y$  被计算为:

$$y = \sum_{i=1}^m \lambda_i o_i(x). \quad (1)$$

其中:  $\lambda_i \geq 0$ ;  $\sum_{i=1}^m \lambda_i = 1$ ; 系数  $\lambda_i$  用于参数化网络架构。刘等人通过对权重的训练数据和诸如  $\lambda$  的结构参数的验证数据交替梯度下降步骤来优化网络权重和网络架构。最终,通过为每个层选择  $i = \operatorname{argmax}_i \lambda_i$  的操作  $i$  来获得离散架构。Ahmed 和 Torresani 也采用基于梯度的神经架构优化,但他们只考虑分别优化层的超参数和连接方式。

### 1.3 性能评估策略 (Performance Estimation Strategy)

搜索策略旨在寻找最大化某些性能测量的神经结构 A,例如测试图像分类的准确性。为了指导搜索过程,这些策略需要估计他们选出的结构 A 的性能。最简单的方法是在训练数据集上训练 A 并评估其在验证数据上的表现。然而,训练每个结构都要从头开始评估,这样的方式经常会带来数千 GPU 天的计算消耗(1 GPU 天定义为一个 GPU 运行一天的计算消耗)

为了减少这种计算负担,可以基于完全训练之后的实际性能的较低保真度来估计性能(也表示为代理度量)。这种较低的保真度包括较短的训练时间,对数据子集的训练,较低分辨率图像,或每层较少的滤波。虽然这些低保真近似降低了计算成本,但它们也会在估计中引入偏差,因为性能通常会被低估。只要搜索策略仅依赖于对不同体系结构进行排名,并且相对排名保持稳定,性能低估问题便不会对架构搜索产生明显影响。然而,最近的结果表明,当近似评估与完全评估之间的差异太大时,这种相对排名可能会发生显著变化<sup>[8]</sup>。

另一种估计架构性能的可能方法建立在学习曲线外推的基础上。Domhan 等人建议推断初始学习曲线,并终止预测表现不佳的曲线,以加快结构搜索

过程;贝克等人也考虑通过结构超参数来预测那些部分学习曲线是最有成长空间的;Liu 等人还提出了训练用于预测新架构性能的替代模型,不采用学习曲线推断,而是支持基于结构/模块特性预测性能,并推断出比训练期间看到的规模更大的结构/模块。预测神经网络结构性能的主要挑战在于如何合理利用有限的计算资源,仅仅通过较少的评估网络来获得其相对准确的性能排序。

加速性能估计的另一种方法是基于之前已经训练过的其他架构的权重来初始化新架构的权重。实现这一目标的方法,被称为网络态射<sup>[9]</sup>。该方法允许修改结构,同时保持网络所代表的功能不变。这允许连续增加网络容量并保持高性能而无需从头开始训练,这些方法的一个优点是它们允许搜索空间没有结构大小的固有上限;另一方面,严格的网络态射只能使架构更大,从而可能导致过于复杂的架构。这可以通过采用允许缩小架构的近似网络态射来减小模型的大小达到模型压缩的目的<sup>[10]</sup>。

One-Shot 结构搜索是另一种加速性能评估的十分有前景的方法,它将所有架构视为超图(HyperNet 或 SuperNet)的不同子图,并在具有该超图的边的架构之间共享权重<sup>[2,11]</sup>。只需要训练整个超网络的权重,子图结构可以通过继承超网络的权重。在没有任何单独训练的情况下,评估结构的性能。因为不需要从头训练(仅评估验证数据的性能),One-shot 模型大大加快了架构的性能评估。这种方法通常会产生很大的偏差,因为它严重低估了网络的实际性能。尽管如此,因为估计的性能与实际性能密切相关,它可以可靠地对排名架构进行排序。不同的 one-shot 方法在训练方式上有所不同;ENAS 学习了一个 RNN 控制器,它从搜索空间中对架构进行采样,并根据强化学习获得的近似梯度训练 one-shot 模型。DARTS 通过在 one-shot 模型的超网的每条边上放置候选操作(卷积算子等)的混合,来获得连续的搜索空间,一次性优化 one-shot 模型的所有权重;Bender 等人只训练 one-shot 模型一次,采样一个结构时,相当于对超网络其它部分进行 dropout 操作,这样获得的模型表现排序已经足够。虽然 ENAS 和 DARTS 在训练期间优化了神经网络结构的分布,但 Bender 等人的方法可以看作使用固定分布。

以上表明,权重优化和结构挑选的组合可能是 one-shot 模型的需要。

One-shot 模型一般限制是预先定义的超图,超

图基本就限定了子图的生成模式。此外,在架构搜索期间要求整个超图缓存在 GPU 中,这相应地限制相较小的超图和搜索空间,因此通常与基于小区的搜索空间结合使用。基于权重共享的方法根本上减少了 NAS 所需的计算资源(从几千 GPU 天到若干 GPU 天)。如果架构的采样分布与 one-shot 模型一起优化,目前还不能很好地解释它们引入搜索的偏差。从根本上来说,就是 one-shot 模型和子图模型的性能一致性问题,就是在超图上的子图模型的排序和子图模型单独从头训练的模型性能排序是否真实一致,目前还没有得到根本上的解决。

## 2 国内研究综述

近来,神经网络架构搜索越来越被国内外研究者重视。谷歌、MIT、Facebook、Uber、亚马逊早早加入战场;国内的商汤、旷视、小米、阿里、华为等各大公司也都紧随其后,在结构搜索方面有若干成果。国内中科院、上海交通大学等也都有一些研究成果问世。

中科大和微软发表的论文《Neural Architecture Optimization》提出另一种基于梯度的方法。先将网络结构嵌入(embedding)到一个连续的空间,这个空间中的每一个点对应一个网络结构。在这个空间上可以定义准确率的预测函数,以它为目标函数进行基于梯度的优化,找到更优网络结构的嵌入表征。优化完成后,再将这个嵌入表征映射回网络结构。这类方法的优点之一就是搜索效率高,对于 CIFAR-10 和 PTB,结合一些权重共享这样的加速手段,消耗可以少于 1 GPU/天。

在神经架构搜索过程中,最为耗时的其实就是对于候选模型的训练。而初版的神经架构自动搜索方法因为对每个候选模型都是从头训练的,因此会相当耗时。一个直观的想法是有没有办法让训练好的网络尽可能重用。上海交大和伦敦大学的论文《Reinforcement Learning for Architecture Search by Network Transformation》中,将网络态射与神经网络搜索结合。所谓网络态射就是将网络进行变形,同时保持其功能不变。这样带来的好处是变形后可以重用之前训练好的权重,而不用重头开始训练。

最近,中科院的论文《You Only Search Once: Single Shot Neural Architecture Search via Direct Sparse Optimization》提出了 DSO-NAS 方法,其特点是只搜一次。它始于一个完全连接的块,在操作间引入缩放因子,同时添加稀疏正则化来去除无用的连接,也就是去除不重要的操作,得到最优结构。文

中提出可以在一个优化问题中同时学习网络结构和参数。

旷视科技最近提出了一种使用均匀分布采样的单路径 One-Shot 模型,它的中心思想是使用一种简化的超网络,通过均匀路径采样的方法进行训练<sup>[12]</sup>。一旦训练好超网络,就使用进化算法来有效的搜索一个表现较好的结构,而无需进一步调整。目前该方法达到了很好的水平。

小米 AutoML 团队近期发表的 FairNas 在 One-shot 模型中为了能在超网络上更公平的比较结构,对每个操作算子进行严格公平训练,也较为明显的提升了超网络的训练拟合效果。

### 3 国内外研究简析

#### 3.1 搜索空间

搜索空间是模型生成的基石,搜索空间确定了模型生成的范围,模型选取的复杂程度。搜索空间的确定依然需要依靠人类先验知识,尽管这可能会阻碍机器去发现更有创新性的结构。

搜索空间的设计是一个十分具有难度的问题,在模型自动生成问题中,一个好的搜索空间,基本就奠定了最终搜索到的模型的性能结果的均值。例如:在一个很优秀的搜索空间中,即使只是采用随机采样的方式选取结构,仍然可能会比在一个较差的搜索空间中使用十分优秀的搜索算法所得到的模型要好得多。

目前绝大多数搜索空间都是基于 cell 方式堆叠,基于 cell 的搜索空间具备两大优势:(1)有效缩减搜索空间大小。(2)更容易进行模型迁移。

首先,基于 cell 的搜索空间可以指数级缩减搜索空间大小,例如:对于一个 8 层网络,一个 cell 包含两层,每层 4 种选择,搜索空间会从  $4^8$  缩减到  $4^2$ ;其次,基于 cell 的搜索空间所生成的模型能够很简单的进行模型的扩展迁移,例如:在一个小型数据集上训练搜索到了一个 8 个 cell 堆叠的模型,为了迁移到更大的数据集,或者获得更好的训练效果,可以将 8 个 cell 扩展到 20 个 cell,这样网络的深度变深,网络的表征能力也就会变强。

搜索空间设计的两个基本原则:

(1)能够包含若干现有的手工设计模型,在搜索结果的反馈中能够更方便的比较手工设计模型和自动生成模型的优劣。

(2)网络中有较为明显的优劣之别,就是整个搜索空间中的结构应该是比较容易比较的。

除了具有代表性的几个搜索空间,绝大多数的

研究者,大都是基于已有的搜索空间进行实验探索,要么采用一样的搜索空间,要么基于其它的搜索空间进行微调(加入或删除几个卷积算子、限制边的连接方式、搜索激活函数等等)。

作为初期的研究者,可以尝试将实验,实现在一个较为简单的搜索空间上,如果获得了比较满意的结果后,在把算法等迁移到更为复杂的搜索空间上,这是一种节约时间和计算资源的方式之一。

#### 3.2 搜索方法

目前主流的搜索方法大概可以分为 4 类:

(1)基于强化学习。最初由谷歌公司在 2017 年于《Neural Architecture Search With Reinforcement Learning》一文中提出,此种方法通常通过一个控制器(一般是一个 RNN 神经网络)负责生成模型的结构和相应超参数,利用这个结构在数据集上进行训练,得到准确率,然后进行反馈。如此反复直到得到最佳的结果。该方法获得好的模型的前提是进行大量的模型训练,时间代价很大,普通研究者是无法承受的。

(2)基于进化算法。初始产生若干结构,下一次选取的结构由已有结构进行交叉变异得到,这种方式的好处是符合自然发展规律,并且有着一个大家都共识的观点:结构相似的模型,在最终的表现结果上也是相似的。进化算法通常和其他方法联合使用,作为一个工具。

(3)基于可微方法。可微方法首次在 DARTS 论文中提出,论文中将结构和参数共同优化,并且将结构通过 softmax 映射到连续空间,使得可以通过求解梯度来进行下一个结构的选取,有得到全局最优的可能。但该方法由于通过梯度选取模型,所以无法控制生成模型的大小和 Flops 数量等。

(4)基于 one-shot 方法。该方法首次由 Google 在 ENAS 中提出,该方法将所有神经网络模型视作有向无环图(DAG)。首先固定节点,所有模型都放在一个超网络(Hypernet 或 Supernet)中,超网络中的一个子图表示一个神经网络模型。对整个超网络进行训练,通过采样子图,来获得一个结构,通过评估选出最好的模型。更多的时候 one-shot 模型都是和其它方法结合使用的(例如 RL、进化算法等)。但目前来说,one-shot 模型仍然没能从根本上解决超网络上模型的表现和最终模型效果的一致性问题。

#### 3.3 评估策略

评价一个网络模型的好坏是算法能够选出真正

优秀模型的途径,最直观也最准确的评估方式就是将模型从头开始训练,训练完全之后查看模型对测试数据集的准确性,但是这种方式往往是最为费时费力的。在搜索算法中,往往要比较成千上万的模型,才能够得出一个较好的局部最优解,然而将成千上万个模型都从头至尾训练完是不经济也不现实的,只有 Google 这样的公司才具备这样的条件。一般会寻求评估结果的准确性和所需算力的折衷。

低保真是其中的一种方式,比如测试时只取测试集合的一个小的子集进行测试,亦或使用低分辨率的图像等等。

很多时候,可以根据已有的效果评估来对搜索过程进行剪枝,如果搜索到的若干较差的结构有某些共同的特点,那之后就避免不去搜索带有这些特点的结构;亦或若干较优的结构具有某些共同的特点,之后就专注于搜索具有这些特点的结构。

基于 One-shot 的方式,由于训练了一个超网络,整个搜索空间上的所有结构都共用这个超网络的模型参数,所以在进行模型评估的时候十分便捷,只需直接使用超网络的参数即可,不需要进行额外的模型训练,所以基于 One-shot 的方法在最近也越来越多的被大家使用。基于 One-shot 模型还可以做到基于可微模型做不到的提前预判,对于带评估的结构,如果其模型参数或者 Flops 数不符合要求,那么可以直接不去进行计算,直接换另外一个采样结构即可。

#### 4 结束语

本文针对神经网络架构搜索这个研究方向,首先给出了目前最普遍接受的问题定义,根据国内外不同的研究现状进行了综述,在最后针对最主要的三个研究内容:搜索空间、搜索策略、评估策略给出了简析。通过本文可以帮助初入本研究方向的研究

人员对神经网络架构搜索了解一个大致的全貌,帮助研究人员更快的进行深入研究。

#### 参考文献

- [1] REAL E, MOORE S, SELLE A, et al. Large-scale evolution of image classifiers [C]//Proceedings of the 34th International Conference on Machine Learning—Volume 70. JMLR. org, 2017: 2902–2911.
- [2] PHAM H, GUAN M, ZOPH B, et al. Efficient neural architecture search via parameter sharing [J]. arXiv preprint arXiv:1802.03268, 2018.
- [3] LIU H, SIMONYAN K, YANG Y. Darts: Differentiable architecture search [J]. arXiv preprint arXiv:1806.09055, 2018.
- [4] SHIN R, PACKER C, SONG D. Differentiable neural network architecture search [C]//International Conference on Learning Representations Workshop, 2018.
- [5] BAKER B, GUPTA O, NAIK N, et al. Designing neural network architectures using reinforcement learning [J]. arXiv preprint arXiv:1611.02167, 2016.
- [6] MIKKULAINEN R, LIANG J, MEYERSON E, et al. Evolving deep neural networks [M]//Artificial Intelligence in the Age of Neural Networks and Brain Computing. Academic Press, 2019: 293–312.
- [7] GOLDBERG D E, DEB K. A comparative analysis of selection schemes used in genetic algorithms [M]//Foundations of genetic algorithms. Elsevier, 1991, 1: 69–93.
- [8] ZELA A, KLEIN A, FALKNER S, et al. Towards automated deep learning: Efficient joint neural architecture and hyperparameter search [J]. arXiv preprint arXiv:1807.06906, 2018.
- [9] WEI T, WANG C, RUI Y, et al. Network morphism [C]//International Conference on Machine Learning. 2016: 564–572.
- [10] ELSKEN T, METZEN J H, HUTTER F. Efficient multi-objective neural architecture search via lamarckian evolution [J]. arXiv preprint arXiv:1804.09081, 2018.
- [11] BENDER G, KINDERMANS P J, ZOPH B, et al. Understanding and simplifying one-shot architecture search [C]//International Conference on Machine Learning. 2018: 550–559.
- [12] GUO Z, ZHANG X, MU H, et al. Single Path One-shot Neural Architecture Search with Uniform Sampling [J]. arXiv preprint arXiv:1904.00420, 2019.

(上接第24页)

- [9] 钱卫星,黄丽亚. 采用二维 Otsu 直方图斜分快速算法实现方式的改进[J]. 华侨大学学报(自然科学版),2016,37(1):88–91.
- [10] WANG W, DUAN L, WANG Y. Fast image segmentation using two-dimensional Otsu based on estimation of distribution algorithm [J]. Journal of Electrical and Computer Engineering, 2017: 1–12.
- [11] 田怀文,刘晓敏,柯小甜,等. 基于双界线的 Otsu 阈值分割法及其快速算法[J]. 计算机应用研究,2016,33(12):3872–3875.
- [12] GUO W Y, WANG X F, XIA X Z. Two-dimensional Otsu's thresholding segmentation method based on grid box filter, Optik 125,2014,5234 – 5240.
- [13] 张新明,孙印杰,郑延斌. 二维直方图准分的 Otsu 图像分割及其快速实现[J]. 电子学报,2011,39(8):1778–1784.
- [14] 化春键,马金科,陈莹. 基于差异哈希算法的改进非局部均值去噪算法[J/OL]. [2020-01-03]. 激光与光电子学进展,1–12.
- [15] 易三莉,张桂芳,贺建峰,等. 基于最大类间方差的熵图像分割[J]. 计算机工程与科学,2018,40(10):1874–1881.
- [16] ZHOU X J, YANG C H, GUI W H. State transition algorithm. Journal of Industrial and Management Optimization, 2012, 8(4): 1039–1056
- [17] 吴禄慎,程伟,胡赞. 应用改进布谷鸟算法优化多阈值图像分割[J/OL]. [2020-04-02]. 吉林大学学报(工学版):1–13
- [18] 邢致恺,贾鹤鸣,宋文龙. 基于莱维飞行樽海鞘群优化算法的多阈值图像分割[J/OL]. [2020-03-30]. 自动化学报:1–15