

## 基于自适应平滑度策略的三维模型分类神经架构搜索

周鹏<sup>1</sup>, 杨军<sup>1,2</sup>

(1. 兰州交通大学电子与信息工程学院, 甘肃 兰州 730070;

2. 兰州交通大学测绘与地理信息学院, 甘肃 兰州 730070)

**摘要:** 针对人工设计三维模型分类网络架构过度依赖专家经验且泛化能力较差的问题, 提出了一种自适应平滑度策略的神经架构搜索方法。首先, 使用改进候选操作选择策略和连续松弛化方法将离散的搜索空间连续化, 并利用权重共享机制提高搜索效率。其次, 在损失函数中添加自适应平滑度策略的正则化, 由温度参数控制损失函数的平滑程度。最后, 使用指数归一化方法计算损失函数, 以避免损失值溢出。在三维点云数据集和蛋白质间相互作用数据集上的实验结果表明, 在相同的训练样本和迭代次数下, 自适应平滑度策略的神经架构搜索方法的分类准确率更高, 性能更稳定。

**关键词:** 正则化; 神经架构搜索; 搜索空间; 点云; 分类

**中图分类号:** TP391.41

**文献标志码:** A

**doi:** 10.11959/j.issn.2096-6652.202417

## Neural architecture search for 3D model classification based on adaptive smoothness strategy

ZHOU Peng<sup>1</sup>, YANG Jun<sup>1,2</sup>

1. School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

2. Faculty of Geomatics, Lanzhou Jiaotong University, Lanzhou 730070, China

**Abstract:** Aiming at the problem of poor generalization ability in hand-crafted architectures that overly rely on expert experience, a neural network architecture search method with an adaptive smoothness strategy was proposed. Firstly, an improved candidate operation selection strategy and a continuous relaxation method were used to convert discrete search space into continuous space, and a weight-sharing mechanism was employed to enhance search efficiency. Secondly, a regularization operation with an adaptive smoothness strategy was added to the loss function, whose smoothness degree was controlled by a temperature parameter. Finally, the loss function was calculated using an exponential normalization method to avoid loss value overflow. Experimental results on 3D point cloud datasets and protein-protein interaction datasets showed that the proposed method achieved higher classification accuracy and more stable performance under the same training samples and iterations.

**Key words:** regularization, neural architecture search, search space, point cloud, classification

### 0 引言

三维模型分类已广泛应用于无人驾驶、机器人

导航、虚拟现实和三维重建等领域<sup>[1-2]</sup>, 成为计算机视觉和计算机图形学的重要研究课题。传统人工设计网络架构的方法过度依赖专家经验, 难以获得

收稿日期: 2024-01-02; 修回日期: 2024-05-14

通信作者: 杨军, yangj@mail.lzjtu.cn

基金项目: 国家自然科学基金项目 (No.42261067); 甘肃省自然科学基金项目 (No.22JR11RA157)

**Foundation Items:** The National Natural Science Foundation of China(No.42261067), The Natural Science Foundation of Gansu Province(No.22JR11RA157)

最优的超参数，需要针对不同应用场景进行重新设计。生成式人工智能是实现通用人工智能的重要方式之一<sup>[3]</sup>，能够发挥深度学习的优势。同时，自动搜索神经架构也成为当前的一个研究热点。

针对高维空间参数自动优化问题，神经架构搜索（neural architecture search, NAS）由搜索策略在搜索空间中测试和评估大量架构，利用最大化适应度函数选择满足给定问题的目标架构，实现网络参数自动优化。基于NAS的图像分类算法在二维图像领域取得了较好的分类准确率<sup>[4]</sup>，但无法直接应用于三维模型。文献[5]提出了一种顺序贪婪架构搜索（sequential greedy architecture search, SGAS）算法，该算法基于有向无环图（directed acyclic graph, DAG）的边定义子问题，贪婪选择每条边上的操作，对剩余的子问题进行迭代处理，从而解决搜索过程和验证过程准确率存在差异的问题。文献[6]提出资源平衡的神经架构搜索（resource balance neural architecture search, RBNAS）算法，该算法将超网与候选操作之间的关系优化为先合作后竞争的关系，从而平衡网络架构的分类准确率和复杂性，但由于剪枝算法对计算资源需求较大，该算法的效率有待提升。文献[7]提出考虑边特征的图神经网络架构搜索（edge feature graph neural architecture search, EFGNAS）算法，但由于需要同时计算边特征和节点特征，该算法在深层网络和大规模数据集上的效率不高。以上方法从选择策略、资源平衡和特征提取等方面对三维模型神经架构搜索方法进行了研究，但仍存在如下问题：一是稳定性不高，基于可微的架构搜索方法在搜索过程中会使无参数操作成为子结构中的大部分操作，一定程度上加速了算法的收敛，但会在训练集上过拟合，搜索得到的网络架构在验证集上效果不佳，产生严重的优化误差问题，尤其是复杂数据模型的处理耗时长且出错率高，降低了算法的稳定性；二是泛化能力不足，在特定数据集和应用下搜索得到的最优分类网络架构，在不同的场景需求和不同数据集下，会表现出不同的搜索效率和分类准确率，算法的泛化能力无法得到保证。

为此，本文提出自适应平滑度策略算法（adaptive smoothness strategy algorithm, ASSA），从浅层到深层依次选择不同的特征核生成局部特征，利用正则化优化方法构建网络架构，解决了搜索过程中存在的优化误差和泛化能力不足的问题。本文的

贡献点如下。

（1）改进候选操作选择算法，使候选操作选择过程可微，允许在反向传播时使用梯度更新网络架构参数，减少无参数操作对网络性能的影响。

（2）在损失函数中添加自适应平滑度策略的正则化操作，由温度参数控制损失函数的平滑程度，使损失值曲线趋于平滑，增强模型的泛化能力。

## 1 相关研究工作

早期的神经架构搜索算法主要基于强化学习<sup>[8]</sup>和进化算法<sup>[9]</sup>，存在计算量大、效率不高的问题。为使搜索过程连续化，避免低效的采样，连续可微分架构搜索（differentiable architecture search, DARTS）<sup>[10]</sup>逐渐受到关注。可微分架构搜索算法交替优化超网权重参数和架构参数，无须单独评估搜索过程中产生的大量网络架构，解决了搜索耗时的问题，但该方法得到的子网与超网存在性能差距，搜索结果不稳定且泛化能力弱。

在增强模型稳定性的研究中，文献[11]采用早停策略，但人工设定的早停准则存在不确定性，可能会错过最优的网络架构。文献[12]提出渐进式可微分架构搜索（progressive differentiable architecture search, P-DARTS）算法，渐进式增加网络结构的深度，缓解了DARTS算法在搜索和评估过程中的深度差异问题。为消除跳跃连接过多的影响，文献[13]提出公平的搜索策略，将搜索阶段候选操作的竞争关系转化为合作关系，但需要根据特定任务进行模型调整和优化。文献[14]通过添加辅助连接来消除跳跃连接的不良影响，确保所有操作的竞争更加公平，使模型的稳定性得到一定提升。文献[15]将Hessian特征值作为衡量网络性能的重要指标，在超网训练过程中采用正则化规则抑制Hessian特征值的生长，在特定数据集上提高了模型的稳定性，但泛化能力仍有待提升。

在提升泛化能力的研究中，文献[16]设计了域间自适应的神经架构（adapting neural architectures between domains, AdaptNas），研究了待搜索网络架构的泛化边界，通过域自适应增强神经架构的泛化能力，最小化不同域间泛化能力的差距。文献[17]通过混合来自不同域的多个小规模数据集，生成一个大规模数据集，从而提高神经架构的泛化能力。文献[18]利用正则化算法对网络架构参数的运算结果进行约束，避免了架构参数的方差过大，该方法

在二维图像分类任务中取得了较好的效果，但在三维模型分类中泛化能力不足。

综上所述，目前的研究在神经架构搜索的稳定性和泛化能力方面取得了一定的成果，但主要集中在改变搜索和离散化过程、添加额外的参数规避无参数操作及多域之间的自适应等方面，缺少同时提升稳定性和泛化能力的研究。因此，针对网络架构参数的正则化优化算法，具有较高的研究价值。

## 2 正则化架构参数策略

### 2.1 候选操作选择策略的搜索空间连续化

以DARTS为基础的可微分架构搜索算法在训练数据集和验证数据集上交替更新架构参数和架构权重。该方法存在两个问题：第一，由于训练和评估数据集的差异，搜索得到的架构在评估阶段表现不佳，这会带来“退化的搜索-评价”问题；第二，其采用离散函数进行混合运算操作，操作的结果是独热编码（one-hot），返回值是集合中使函数值最大化对应的元素，在计算候选操作时不可导，因而无法采用随机梯度下降法进行优化。

本文通过改进候选操作选择策略，使搜索空间连续化，并使用权重共享机制，以堆叠方式搜索网络架构。每个搜索单元是包含 $N$ 个节点的DAG，节点表示潜在的网络层，特征信息通过DAG的边从节点 $x_i$ 传递到 $x_j$ ，每条边包含若干个候选操作，节点间的混合操作权重由可学习的架构参数 $\alpha^{(i,j)}$ 加权运算得到，从而将离散操作选择问题转换为可微参数优化问题。节点 $x_i$ 、节点 $x_j$ 之间的边上的操作作用混合运算函数 $\bar{O}^{(i,j)}$ 表示：

$$\bar{O}^{(i,j)}(x) = \sum_{k=1}^{|\mathcal{O}|} \zeta_k^{(i,j)} O_k(x) \quad (1)$$

$$\zeta_k^{(i,j)} = \frac{\exp(\alpha_k^{(i,j)})}{\sum_{k'=1}^{|\mathcal{O}|} \exp(\alpha_{k'}^{(i,j)})} \quad (2)$$

式中， $x$ 是输入特征， $\bar{O}$ 表示混合运算， $\mathcal{O}$ 表示候选操作集， $O_k$ 表示第 $k$ 个候选操作， $\zeta$ 表示归一化指数函数运算， $\alpha^{(i,j)}$ 表示与节点 $(i, j)$ 相关的架构参数， $\alpha_k^{(i,j)}$ 是与候选操作 $k$ 相关联的架构参数。

本文采用Gumbel-Softmax<sup>[19]</sup>方法将Gumbel分布与Softmax()函数操作结合，可实现离散采样的可微分近似，并在采样过程中进行梯度反向传播，使梯度可导且候选操作选择过程可微，在反向传播

时用梯度更新网络架构参数，从而减少无参数操作对网络性能的影响。最终的网络架构由混合操作选择权重最高的候选操作得到，候选操作选择策略优化如下：

$$O_k^{(i,j)} = \text{Gumbel} - \text{Softmax}_{k \in \mathcal{O}} \alpha_k^{(i,j)} \quad (3)$$

### 2.2 基于自适应平滑度策略的正则化优化算法

当样本特征多而样本数相对较少时，模型易陷入过拟合，可采用正则化的方法处理这个问题。正则化算法在损失函数中添加正则化项，调整权值以减少过拟合，从而提高模型泛化能力。本文针对三维模型分类设计了网络架构参数的正则化算法，在损失函数中引入正则化项对网络架构参数进行调整，以提高神经架构自动搜索方法的稳定性和泛化能力。

本文采用基于双层优化的搜索算法搜索得到最优子网，验证阶段的主要任务是优化网络架构参数 $\alpha$ ，训练阶段的主要任务是优化网络权重参数 $\omega$ ，通过梯度下降法降低训练损失和验证损失。在可微架构搜索方法中，双层优化很好地平衡了模型准确率和效率，通过逐步递减验证损失 $\nabla_{\alpha} \mathcal{L}_{\text{val}}(\omega - \zeta \nabla_{\omega} \mathcal{L}_{\text{tr}}(\omega, \alpha), \alpha)$ 更新网络架构参数；同时，通过逐步递减训练损失值 $\nabla_{\omega} \mathcal{L}_{\text{tr}}(\omega, \alpha)$ 更新网络权重参数 $\omega$ 。其中， $\nabla$ 表示梯度计算， $\nabla_{\alpha}$ 和 $\nabla_{\omega}$ 分别表示通过梯度下降法优化 $\alpha$ 和 $\omega$ ， $\mathcal{L}_{\text{val}}$ 和 $\mathcal{L}_{\text{tr}}$ 分别表示验证损失和训练损失， $\zeta$ 表示归一化指数函数运算。由于优化过程需要耗费较多的计算资源，准确地评估架构存在一定难度，可采用以下近似优化方案：

$$\omega_k^{t+1} = \omega_k^t - \eta_{\omega} \nabla_{\omega k} \mathcal{L}_{\text{tr}}(\omega_k^t, \alpha_k^t) \quad (4)$$

$$\alpha_k^{t+1} = \alpha_k^t - \eta_{\alpha} \nabla_{\alpha k} \mathcal{L}_{\text{val}}(\omega_k^t, \alpha_k^t) \quad (5)$$

其中， $\mathcal{L}_{\text{tr}}(\omega_k^t, \alpha_k^t)$ 表示边 $t$ 上的训练损失， $\mathcal{L}_{\text{val}}(\omega_k^t, \alpha_k^t)$ 表示边 $t$ 上的验证损失， $\omega_k^{t+1}$ 表示优化后的权重参数， $\alpha_k^{t+1}$ 表示优化后的架构参数， $\eta_{\omega}$ 和 $\eta_{\alpha}$ 分别表示参数 $\omega$ 和 $\alpha$ 的初始学习率。优化 $\omega$ 会加快训练准确率上升， $\alpha$ 决定最终的训练准确率<sup>[6]</sup>，因此本文讨论的是网络架构参数 $\alpha$ 的正则化算法。DARTS算法及在此基础上的研究工作<sup>[2-7]</sup>使用的L2正则化算法，对每个权重的惩罚相对均匀，在权重衰减过程中会使大部分特征的权重接近但不为零，可能导致较小权重的特征影响模型性能，尤其是在处理三维点云数据时会降低计算效率。因此，在梯度自适应算法中，L2正则化并非最优，可考虑重新定义

正则化算法。在式(5)的基础上,对架构参数 $\alpha$ 进行正则化后表示为:

$$\bar{\alpha}_k^{t+1} \approx \alpha_k^t - \eta_\alpha (\nabla_{\alpha_k} \mathcal{L}_{\text{val}} + \mu T(\alpha_k^t)) \quad (6)$$

其中,  $\mu$  是一个非负的正则化调整系数, 用来平衡训练损失和验证损失。 $\bar{\alpha}_k^{t+1}$  表示  $\alpha$  经过正则化后的值, 映射函数  $T$  决定了  $\bar{\alpha}_k^{t+1}$  与  $\alpha_k^t$  之间的变化, 同时对  $\zeta$  影响较大。 $T$  必须满足以下两点: (1)  $T$  的值应趋于稳定, 不受  $\alpha$  值大幅度变化的影响, 从而避免无效正则化和优化效果不高的问题; (2)  $T$  要能反映  $\alpha$  的变化大小, 并对  $\alpha$  变化过大的操作施加更多的惩罚。为此, 构造函数  $T$  如下:

$$T((\alpha_k + u_k)/\tau) = \frac{\exp((\alpha_k + u_k)/\tau)}{\sum_{k'=1}^{|\mathcal{O}|} \exp((\alpha_{k'} + u_{k'})/\tau)} \quad (7)$$

式中,  $u_k$  表示 Gumbel(0, 1) 的独立同分布采样。 $\tau$  是  $T$  函数大于零的温度参数, 控制  $T$  的平滑程度, 温度越高, 生成的分布越平滑, 越接近均匀分布; 温度越低, 生成的分布越接近离散的独热编码分布。在训练中, 可以逐渐降低温度, 以逐步逼近真实的离散分布。若  $(\alpha_k + u_k)/\tau$  过大, 对应的指数函数也随之变大, 此时容易出现上溢问题; 若  $(\alpha_k + u_k)/\tau$  过小, 对应的指数函数会出现下溢问题, 可能会导致分母为 0。为使损失回归正常范围, 避免上溢和下溢出现, 使用指数归一化方法计算交叉熵损失函数, 即对  $T(a)$  求对数, 如式(8)所示:

$$\begin{aligned} \log \left( \frac{\exp((\alpha_k + u_k)/\tau)}{\sum_{k'=1}^{|\mathcal{O}|} \exp((\alpha_{k'} + u_{k'})/\tau)} \right) &= \\ \log(\exp((\alpha_k + u_k)/\tau)) - \log \left( \sum_{k'=1}^{|\mathcal{O}|} \exp((\alpha_{k'} + u_{k'})/\tau) \right) &= \\ (\alpha_k + u_k)/\tau - \log \left( \sum_{k'=1}^{|\mathcal{O}|} \exp((\alpha_{k'} + u_{k'})/\tau) \right) \end{aligned} \quad (8)$$

LogSumExpTem 函数表示带有温度参数  $\tau$  的对数求和指数函数, 可简化为 LSET 函数。该函数是影响损失值的主要部分, 其定义如式(9)所示:

$$\text{LogSumExpTem} = \log \left( \sum_{k'=1}^{|\mathcal{O}|} \exp((\alpha_{k'} + u_{k'})/\tau) \right) \quad (9)$$

对于函数  $f(\delta) = \log(\delta + \lambda)$ , 根据线性近似的定义可知:  $f(\delta) = f(\delta - \lambda) + f'(\delta - \lambda) \cdot \lambda = \log \delta + \frac{1}{\delta} \cdot$

$\lambda$ , 设  $\Gamma_{k'} = (\alpha_{k'} + u_{k'})/\tau$ ,  $\Gamma_{k'} > 0$ , 有:

$$\begin{aligned} \text{LSET} &= \log \left( \sum_{k'=1}^{|\mathcal{O}|} \exp((\alpha_{k'} + u_{k'})/\tau) \right) = \\ \log \left( \sum_{k'=1}^{|\mathcal{O}|} \exp(\Gamma_{k'}) \right) &= \\ \log \left( \exp(\Gamma_{k''}) + \left( \sum_{k' \neq k''} \exp(\Gamma_{k'}) \right) / \exp(\Gamma_{k''}) \right) \end{aligned} \quad (10)$$

其中,  $k'$ 、 $k''$  表示候选操作集  $\mathcal{O}$  的任一元素, 设集合  $\mathcal{O}$  的元素个数为  $n$ , 对于  $\Gamma_{k'}$  来说:

$$\begin{aligned} \log(\exp(\max(\Gamma_{k'}))) &\leq \log \left( \sum_{k'=1}^{|\mathcal{O}|} \exp(\Gamma_{k'}) \right) \leq \\ \log(n \cdot \exp(\max(\Gamma_{k'}))) & \end{aligned} \quad (11)$$

可以得出:

$$\begin{aligned} \max(\Gamma_1, \dots, \Gamma_n) &\leq \text{LSET}(\Gamma_1, \dots, \Gamma_n) \leq \\ \max(\Gamma_1, \dots, \Gamma_n) &+ \log(n) \end{aligned} \quad (12)$$

由式(12)可知, LSET 函数是对  $\max()$  函数的一种平滑操作, 是改进了的 Softmax() 函数, 可实现对架构参数  $\alpha$  的正则化。由式(2)可知,  $\zeta$  的值受参数  $\alpha$  影响, 因此, 使用 LSET 函数可以优化  $\zeta$ 。

### 3 实验结果与分析

为验证算法的稳定性和泛化能力, 在 ModelNet 数据集和 PPI 数据集进行实验, 在每个数据集上进行 5 次实验。

#### 3.1 ModelNet 数据集实验结果分析

ModelNet 数据集是普林斯顿大学发布的三维模型数据集, 共有 662 种分类、127 915 个 CAD 模型。该数据集包含 3 个子集: ModelNet10 包含 10 个类别的带有方向标记的三维数据集合; ModelNet40 包含 40 个类别的三维数据集合; Aligned40 包含 40 个类别的对齐三维数据集合。

本文实验在 ModelNet10 数据集上训练和验证, 在 ModelNet40 数据集上评估。由于搜索初期的网络性能不稳定, 前 10 次迭代不进行搜索决策, 之后每 5 个迭代进行一次搜索决策。搜索阶段和评估阶段的初始学习率分别为 0.004 和 0.001, 迭代数分别为 50 和 250, 批量值均为 32, 采样点个数均为 1 024。

ASSA 在 ModelNet 数据集上的 5 次实验的分类性能如表 1 所示, 5 次实验的分类准确率较为接近, 网络性能稳定。其中, ASSA\_ModelNet\_1 的分类

准确率最高, 达到93.83%, 参数量最多, GPU时长最长; ASSA\_ModelNet\_2的分类准确率仅次于ASSA\_ModelNet\_1, 参数量最小, 但GPU时长并不是最小的。因此, 可得出结论: 分类准确率、参数量和GPU时长之间的关系紧密, 不能仅靠某一类参数衡量网络模型的性能。

表1 ASSA在ModelNet数据集上的分类性能

| 实验名称            | 分类准确率  | 参数量/MB | GPU时长/h |
|-----------------|--------|--------|---------|
| ASSA_ModelNet_1 | 93.83% | 4.15   | 13.85   |
| ASSA_ModelNet_2 | 93.57% | 3.81   | 9.82    |
| ASSA_ModelNet_3 | 92.79% | 4.10   | 11.33   |
| ASSA_ModelNet_4 | 92.80% | 3.91   | 6.55    |
| ASSA_ModelNet_5 | 92.26% | 3.95   | 6.59    |

图1为5次实验搜索得到的网络架构, 本文方法采用的候选操作与文献[7]相同。每个架构有2个输入, 它们是上一个网络架构的输出。 $c_{\{k\}}$ 表示本架构的输出, 节点之间连线上的标注是搜索出的候选操作。图1(a)、图1(b)和图1(c)的结构相同, 但是候选操作不完全相同, 其中图1(a)和图1(b)有3个候选操作不相同, 由表1可知ASSA\_ModelNet\_1和ASSA\_ModelNet\_2的分类准确率较为接近, 但ASSA\_ModelNet\_2的参数量更小, GPU时长更短。可见, 个别候选操作可能会影响

整体网络架构的性能, 同时也说明最优的网络架构不仅和候选操作有关, 还和候选操作的组合方式有关。

将本文搜索得到的网络架构同其他方法进行对比, 如表2所示。本文方法的分类准确率最优值为93.83%, 与人工设计模型KPConv相比, 本文方法的分类准确率提升了1.53%, 参数量减少了10.15 MB; 与SGAS(最佳值)算法相比较, 本文方法的分类准确率提升了1.49%, 参数量减少了4.34 MB, GPU时长减少了7.44 h; 与EFGNAS(最佳值)算法相比, 本文方法的分类准确率提升了0.41%, 参数量增加0.24 MB, GPU时长增加了8.4 h。本文方法的分类准确率优于其他算法, 但参数量和搜索时间有所增加, 因为正则化算法通过平滑函数操作减少了数据的抖动, 有助于提取数据的关键特征, 但正则化算法中设计的对数求和指数函数LSET需要的计算资源较大, 导致所消耗的GPU资源多于其他对比方法。

本文提出的ASSA算法在损失函数中添加了自适应平滑度策略, 对架构参数进行了正则化。将本文算法与SGAS算法、EFGNAS算法进行对比, 这两种对比算法都采用了神经架构搜索的方式, 数据集与本文算法相同。其中, 在ModelNet数据集对

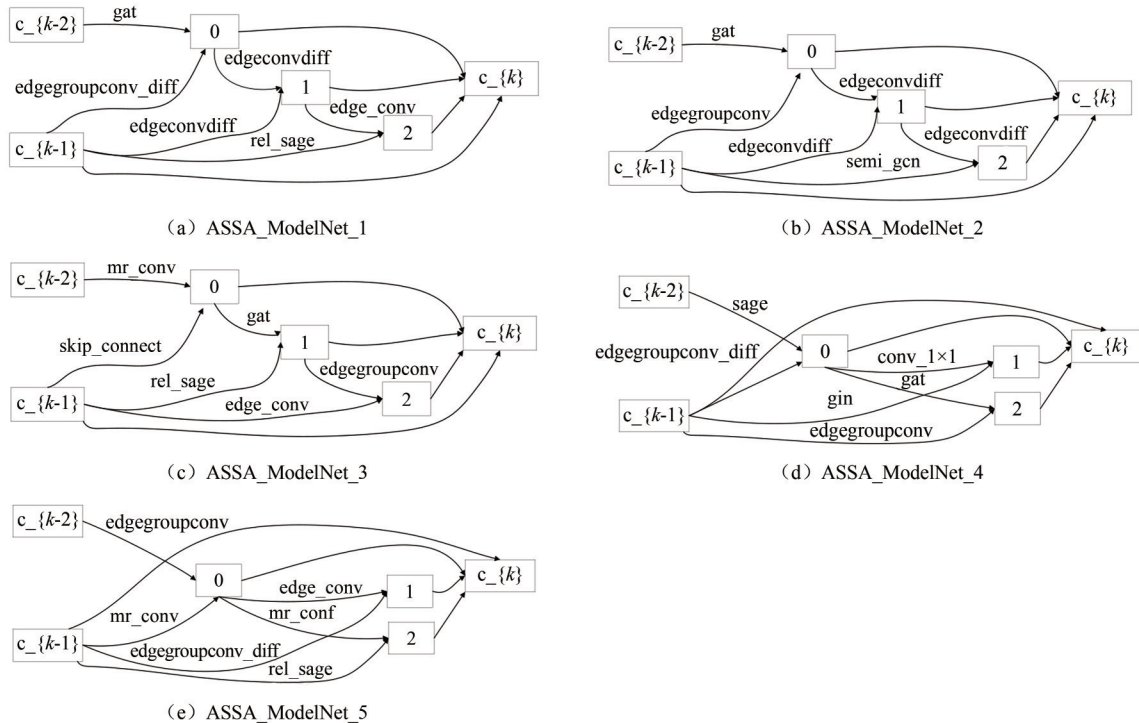


图1 ASSA在ModelNet数据集上搜索到的网络架构

表2 不同算法的点云分类性能

| 对比项                              | 分类准确率  | 参数量/MB | GPU时长/h |
|----------------------------------|--------|--------|---------|
| PointNet++ <sup>[20]</sup>       | 90.70% | 1.48   | 人工设计    |
| KPConv <sup>[21]</sup>           | 92.30% | 14.30  | 人工设计    |
| DGCNN <sup>[22]</sup>            | 92.20% | 1.84   | 人工设计    |
| SGAS <sup>[5]</sup> (平均值)        | 91.95% | 8.99   | 24.72   |
| SGAS <sup>[5]</sup> (最佳值)        | 92.34% | 8.49   | 21.36   |
| RBNAS <sup>[6]</sup> (平均值)       | 92.88% | 4.10   | 2.88    |
| RBNAS <sup>[6]</sup> (最佳值)       | 93.10% | 4.11   | 2.88    |
| EFGNAS <sup>[7]</sup> (平均值)      | 92.56% | 4.24   | 6.72    |
| EFGNAS <sup>[7]</sup> (最佳值)      | 93.42% | 3.91   | 5.52    |
| NoisyDARTS <sup>[23]</sup> (平均值) | 92.65% | 8.72   | 2.64    |
| NoisyDARTS <sup>[23]</sup> (最佳值) | 92.90% | 8.48   | 2.64    |
| ARGNP <sup>[24]</sup>            | 93.48% | 4.15   | 4.80    |
| ASSA(平均值)                        | 93.05% | 3.98   | 9.60    |
| ASSA(最佳值)                        | 93.83% | 4.15   | 13.92   |

比实验中, 本文算法与EFGNAS算法的候选操作相同; 在PPI数据集对比实验中, 本文算法与SGAS算法的候选操作相同。

本文方法和EFGNAS算法在ModelNet数据集上训练、验证和测试的分类性能如图2所示, 每种方法分别进行5次实验。由图2(a)可知, 在训练阶段的5次实验中, 本文方法的分类准确率提升较为稳定, 没有出现大幅波动。EFGNAS算法的分类准确率波动较大, 尤其是EFGNAS\_3、EFGNAS\_4、EFGNAS\_5。该波动影响了验证阶段的中后期, 验证阶段第40~50次迭代的分类准确率持续降低。可见, 本文算法的分类准确率通过正则化算法优化得到了平稳提升, 说明本文算法增强了网络模型的稳定性。

由图2(a)可知, 在前30次迭代中, ASSA的分类准确率低于EFGNAS算法, 但在第30~50次迭代中, ASSA分类准确率迅速提升。由图2(b)和图2(c)可知, 经过前10次迭代预热后, ASSA的网络性能趋于稳定, 开始进行决策选择; ASSA的分类准确率高出EFGNAS算法, 且其在验证阶段和测试阶段的变化基本一致。

为进一步验证本文方法在降低模型训练损失和提升训练精度方面的有效性, 与EFGNAS算法进行对比, 实验结果如图3、图4所示。可以看出EFGNAS算法的平均训练损失和平均验证损失的平滑度不一致且损失值较大, 本文方法的平均训练损失和平均验证损失的平滑度接近且损失值远低于EFGNAS算法。如图3所示, 两种算法的平均训练

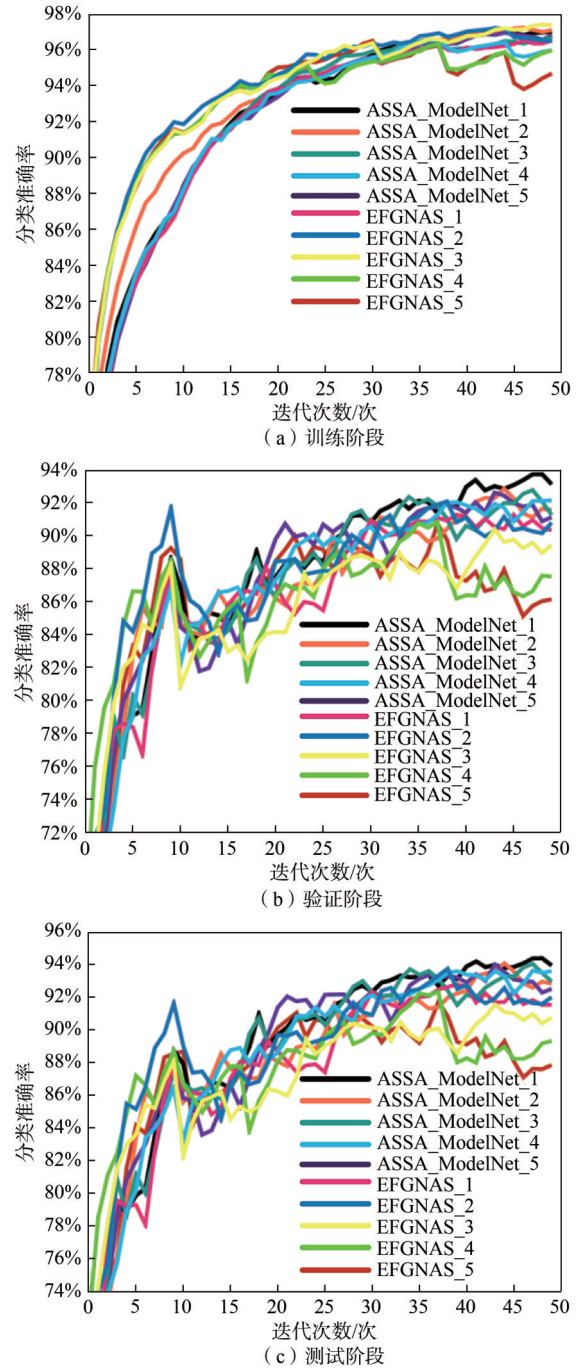


图2 ASSA和EFGNAS算法的训练、验证和测试分类准确率

损失随着迭代次数的增加平缓下降。如图4所示, EFGNAS算法的平均验证损失在前期迭代过程中的递减不规律, 这与初始阶段不稳定的网络性能有关, 而ASSA的损失值稳步递减。

### 3.2 PPI数据集实验结果分析

蛋白质相互作用 (protein-protein interaction, PPI) 数据集是一种生物图数据集, 包含蛋白质间的相互作用信息。该数据集共24张图, 每张图的

平均节点数为2 372个,每个节点有50个特征和121个标签,节点表示蛋白质,边表示蛋白质之间的相互作用。每张图对应不同的生物组织,包含位置基因集、基序基因集和免疫学特征,以基因本体集作为标签,在图神经网络中主要用于节点分类任务。PPI用于描述两种及以上蛋白质的结合过程,PPI网络可用来描述蛋白质间的复杂相互作用关系。本文实验在PPI数据集上搜索和评估用于节点分类任务的神经架构。

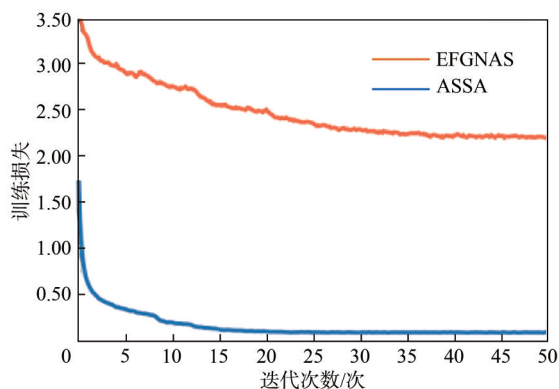


图3 ASSA和EFGNAS算法在ModelNet数据集上的平均训练损失

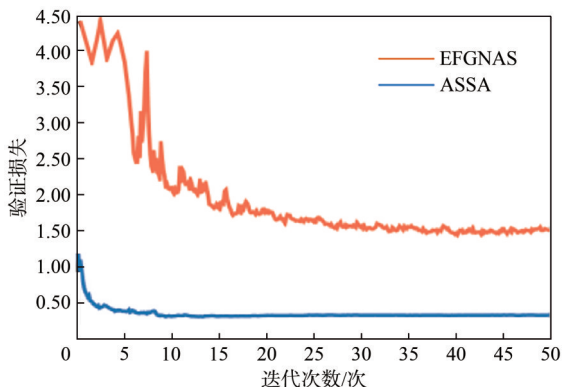


图4 ASSA和EFGNAS算法在ModelNet数据集上的平均验证损失

搜索和评估阶段的迭代数分别为50和2 000,学习率分别为0.005和0.002,初始通道数分别为32和256,批量值大小分别为6和1。ASSA在PPI数据集上5次实验的分类性能如表3所示,5次实验的分类准确率比较接近,最高值为99.46%,最低值为99.44%,模型性能稳定性强。然而,部分实验的分类准确率相同而参数量和GPU时长并不相同,但是差别不是很大。这进一步验证了分类准确率、参数量、GPU时长和之间的关系受到多种因素的影响。

在PPI数据集上搜索得到的网络架构如图5所示,各参数和候选操作的表示方法与图1相同,候

选操作的选取与SGAS算法一致。如表3所示,ASSA\_PPI\_2的准确率最高且参数量最少,GPU时长也仅次于ASSA\_PPI\_5。如图5(b)所示,ASSA\_PPI\_2出现了两次跳跃连接(skip\_connect),由此可见,合理的跳跃连接可以帮助神经网络更好地学习特征,提高模型的泛化能力,简化模型结构。

表3 ASSA在PPI数据集上的分类性能

| 实验名称       | 分类准确率  | 参数量/MB | GPU时长/h |
|------------|--------|--------|---------|
| ASSA_PPI_1 | 99.46% | 6.22   | 3.12    |
| ASSA_PPI_2 | 99.46% | 5.88   | 2.40    |
| ASSA_PPI_3 | 99.45% | 7.20   | 2.64    |
| ASSA_PPI_4 | 99.44% | 6.22   | 3.12    |
| ASSA_PPI_5 | 99.44% | 7.20   | 2.16    |

不同方法在PPI数据集上的分类性能如表4所示,与人工设计的网络架构DenseMRGCN-14<sup>[25]</sup>和自动搜索算法SGAS相比,本文方法的分类准确率略有提高,但参数量分别减少了47.54 MB和1.32 MB,但本文方法的初始通道数为SGAS方法的1/2,较多的初始通道虽然能更好地捕获输入数据的特征,但也会增加网络的复杂性和计算成本。本文方法的初始通道数较小,能够降低网络复杂性,在同样的迭代次数下其网络性能更优、参数量更小。由于PPI数据集的数据量较少,因此本文方法的GPU时长与SGAS算法方法相同。

表4 不同方法在PPI数据集上的分类性能

| 算法                               | 分类准确率  | 参数量/MB | GPU时长/h |
|----------------------------------|--------|--------|---------|
| DenseMRGCN-14 <sup>[25]</sup>    | 99.43% | 53.42  | 人工设计    |
| ResMRGCN-28 <sup>[25]</sup>      | 99.41% | 14.76  | 人工设计    |
| GraphSAGE (LSTM) <sup>[26]</sup> | 61.20% | 0.26   | 人工设计    |
| GAT <sup>[27]</sup>              | 97.30% | 1.81   | 人工设计    |
| SGAS <sup>[5]</sup>              | 99.43% | 7.20   | 0.02    |
| ASSA                             | 99.46% | 5.88   | 0.02    |

本文方法和SGAS算法在PPI数据集上的训练分类准确率和验证分类准确率分别如图6、图7所示,每种方法进行2次实验。为便于比较,实验设置的迭代次数与SGAS算法相同,均为50次。可以看出,在前10次迭代中,本文方法和SGAS算法在训练分类准确率和验证分类准确率上的差异较小,但本文方法的训练分类准确率均高于SGAS算法,而验证分类准确率相对较低;在迭代10次后,本文方法的训练分类准确率和验证分类准确率均有

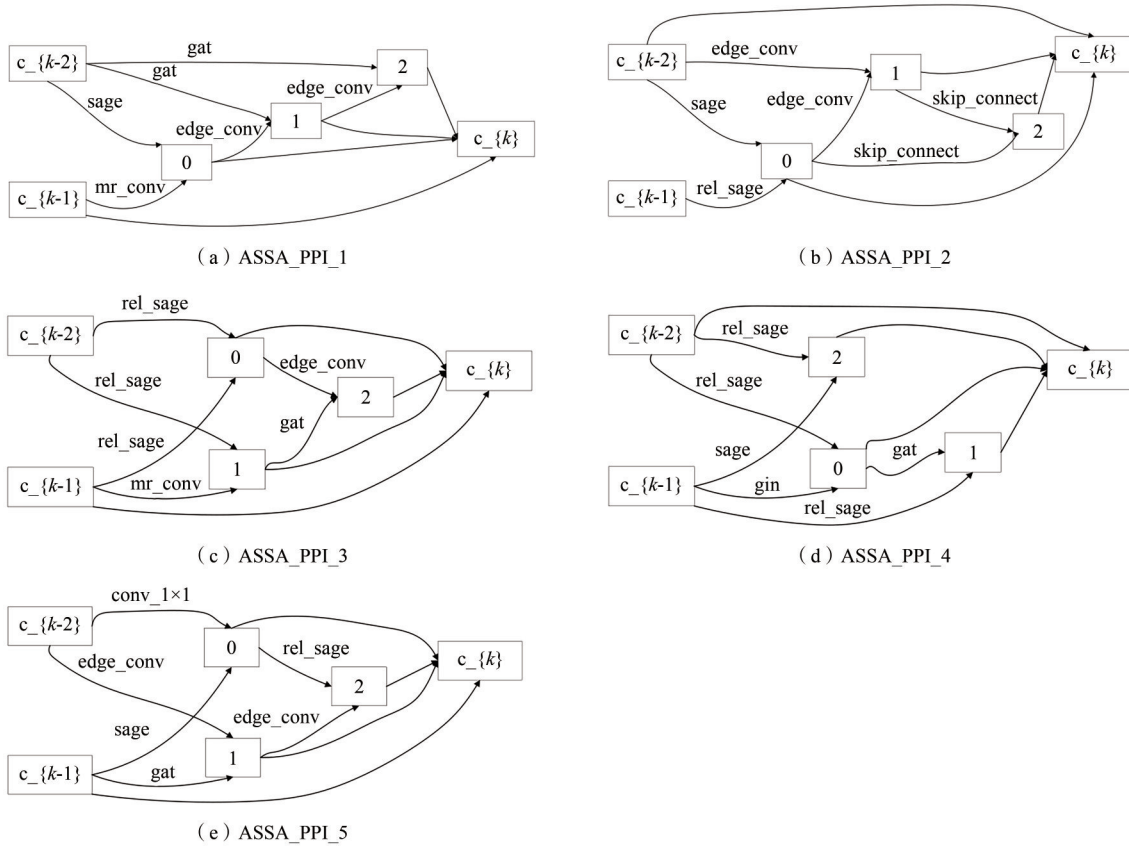


图5 ASSA在PPI数据集上搜索得到的网络架构

大幅提升，而SGAS算法的分类准确率呈现下降趋势。这是因为正则化优化算法中的温度参数可以动态调整，从而缩短训练损失和验证损失之间的差距，稳定网络模型的性能。

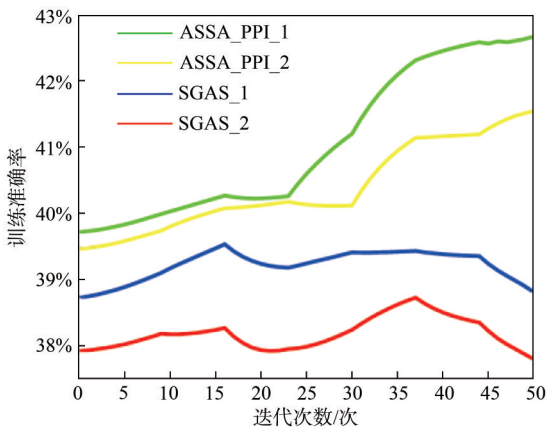


图6 ASSA和SGAS算法在PPI数据集上的训练分类准确率

#### 4 结束语

本文提出了基于自适应平滑度策略的三维模型分类网络架构自动设计方法，该方法对候选操

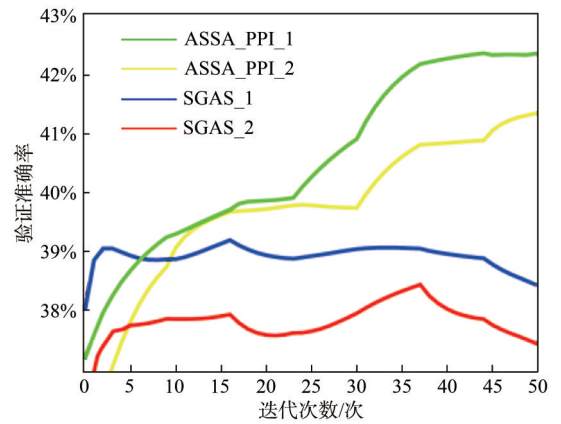


图7 ASSA和SGAS算法在PPI数据集上的验证分类准确率

作选择策略进行了优化，在损失函数中添加了基于自适应平滑度策略的正则化操作，使损失函数趋于平滑，从而提升了模型的稳定性和泛化能力。在ModelNet和PPI数据集的实验结果表明，在有限的计算资源下，本文方法获得了最佳的分类准确率，并在速度和准确率之间取得了良好的平衡。未来可进一步优化搜索时间，提高网络架构的搜索效率。

## 参考文献:

- [1] 郑伟斌, 练国富, 张学明, 等. 基于主成分分析和特征图匹配的点云配准方法[J]. 智能科学与技术学报, 2023, 5(4): 543-552.  
ZHENG W B, LIAN G F, ZHANG X M, et al. Point cloud registration method based on principal component analysis and feature map matching[J]. Chinese Journal of Intelligent Science and Technology, 2023, 5(4): 543-552.
- [2] 于航, 付彦伟, 姜柏言, 等. 基于少量图像的三维重建综述[J]. 智能科学与技术学报, 2022, 4(4): 544-559.  
YU H, FU Y W, JIANG B Y, et al. A survey of image-based few-shot 3D reconstruction[J]. Chinese Journal of Intelligent Science and Technology, 2022, 4(4): 544-559.
- [3] 缪青海, 王兴霞, 杨静, 等. 从基础智能到通用智能: 基于大模型的GenAI和AGI之现状与展望[J]. 自动化学报, 2024, 50(4): 674-687.  
MIAO Q H, WANG X X, YANG J, et al. From foundation intelligence to general intelligence: the state-of-art and perspectives of GenAI and AGI based on foundation models[J]. Acta Automatica Sinica, 2024, 50(4): 674-687.
- [4] 李建明, 陈斌, 孙晓飞. 熵启发的分级可微分网络架构搜索[J]. 哈尔滨工业大学学报, 2021, 53(8): 22-28.  
LI J M, CHEN B, SUN X F. Multi-level differentiable architecture search with heuristic entropy[J]. Journal of Harbin Institute of Technology, 2021, 53(8): 22-28.
- [5] LI G H, QIAN G C, DELGADILLO I C, et al. SGAS: sequential greedy architecture search[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 1617-1627.
- [6] 周鹏, 杨军. 采用神经网络架构搜索的三维模型分类[J]. 计算机辅助设计与图形学学报, 2022, 34(5): 722-733.  
ZHOU P, YANG J. 3D model classification based on neural architecture search[J]. Journal of Computer-Aided Design & Computer Graphics, 2022, 34(5): 722-73.
- [7] 周鹏, 杨军. 基于边缘特征的图神经网络架构搜索[J]. 兰州交通大学学报, 2022, 41(6): 44-53.  
ZHOU P, YANG J. Graph neural architecture search based on edge features[J]. Journal of Lanzhou Jiaotong University, 2022, 41(6): 44-53.
- [8] ZOPH B, LE Q V. Neural architecture search with reinforcement learning[EB]. arXiv preprint, 2016, arXiv: 1611.01578.
- [9] REAL E, MOORE S, SELLE A, et al. Large-scale evolution of image classifiers[C]//Proceedings of the 34th International Conference on Machine Learning. New York: ACM, 2017: 2902-2911.
- [10] LIU H X, SIMONYAN K, YANG Y M. DARTS: differentiable architecture search[EB]. arXiv preprint, 2018, arXiv: 1806.09055.
- [11] LIANG H W, ZHANG S F, SUN J C, et al. DARTS+: improved differentiable architecture search with early stopping[EB]. arXiv preprint, 2019, arXiv: 1909.06035.
- [12] CHEN X, XIE L X, WU J, et al. Progressive differentiable architecture search: bridging the depth gap between search and evaluation[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 1294-1303.
- [13] CHU X X, ZHOU T B, ZHANG B, et al. Fair DARTS: eliminating unfair advantages in differentiable architecture search[C]//Proceedings of the 2020 European Conference on Computer Vision. Cham: Springer, 2020: 465-480.
- [14] CHU X X, WANG X X, ZHANG B, et al. DARTS-: robustly stepping out of performance collapse without indicators[EB]. arXiv preprint, 2020, arXiv: 2009.01027.
- [15] ZELA A, ELSKEN T, SAIKIA T, et al. Understanding and robustifying differentiable architecture search[EB]. arXiv preprint, 2019, arXiv: 1909.09656.
- [16] LI Y X, YANG Z H, WANG Y H, et al. Adapting neural architectures between domains[C]//Proceedings of the 34th Conference on Neural Information Processing Systems. New York: ACM, 2020: 789-798.
- [17] LIU L Y, WEN Z W, LIU S W, et al. MixSearch: searching for domain generalized medical image segmentation architectures[EB]. arXiv preprint, 2021, arXiv: 2102.13280.
- [18] YE P, LI B P, LI Y K, et al.  $\beta$ -DARTS: beta-decay regularization for differentiable architecture search[C]//Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 10864-10873.
- [19] HERRMANN C, BOWEN R S, ZABIH R. Channel selection using gumbel softmax[C]//Proceedings of the 2020 European Conference on Computer Vision. Cham: Springer, 2020: 241-257.
- [20] QI C R, YI L, SU H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 5105-5114.
- [21] THOMAS H, QI C R, DESCHAUD J E, et al. KPConv: flexible and deformable convolution for point clouds[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 6410-6419.
- [22] WANG Y, SUN Y B, LIU Z W, et al. Dynamic graph CNN for learning on point clouds[J]. ACM Transactions on Graphics, 2019, 38(5): 146-160.
- [23] CHU X X, ZHANG B. Noisy differentiable architecture search[EB]. arXiv preprint, 2020, arXiv: 2005.03566.
- [24] CAI S F, LI L, HAN X Z, et al. Automatic relation-aware graph network proliferation[C]//Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 10853-10863.
- [25] LI G H, MÜLLER M, THABET A, et al. DeepGCNs: can GCNs go as deep as CNNs? [C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 9266-9275.
- [26] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs[EB]. arXiv preprint, 2017, arXiv: 1706.02216.
- [27] BASTIDAS A A, TANG H L. Channel attention networks[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE Press, 2019: 881-888.

## [作者简介]



周鹏 (1982-), 男, 兰州交通大学电子与信息工程学院博士生, 主要研究方向为三维模型处理、神经架构搜索。



杨军 (1973-), 男, 博士, 兰州交通大学教授、博士生导师, 主要研究方向为模式识别、三维模型的空间分析、遥感影像分析与处理等。