小型微型计算机系统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2025-0234
2026年5月 第5期
Vol.47 No.5 2026

面向资源受限环境的渐进式多保真度神经架构搜索

王恩良[1,2],胡晔凯[2,3],陈文欣[2,3],孙知信[2,3]

[1](南京邮电大学 物联网学院,南京 210003)
[2](南京邮电大学 国家邮政局邮政行业技术研发中心(物联网技术),南京 210003)
[3](南京邮电大学 宽带无线通信技术教育部工程研究中心,南京 210003)
E-mail:sunzx@njupt.edu.cn

摘 要: 神经架构搜索(NAS)是深度学习自动化的关键技术,但其高昂的计算成本严重限制了实际应用.传统方法需要对每个候选架构进行完整训练,导致搜索过程耗时且资源密集.本文提出渐进式多保真度神经架构搜索方法(PMF-NAS),通过三阶段渐进策略实现高效架构搜索.PMF-NAS在全局探索阶段使用低保真度快速评估识别高潜力区域,在区域搜索阶段采用中等保真度在缩小的空间内细化搜索,在精细优化阶段对最优候选进行高保真度验证.该方法的核心是基于早期训练特征的性能预测器,能够准确预测架构最终性能,避免大量无效计算.同时引入自适应资源分配机制,根据架构潜力和不确定性动态调整评估投入.实验表明,PMF-NAS在单GPU环境下可在8~9小时内完成搜索,同时在多个数据集上达到最优或接近最优的准确率.文本为资源受限环境下的神经架构搜索提供了实用解决方案,降低了NAS技术的应用门槛,有望推动其在更广泛领域的应用.

关 键 词: 神经架构搜索;多保真度评估;渐进式搜索;性能预测;资源优化
中图分类号: TP391 **文献标识码:** A **文 章 编 号:** 1000-1220(2026)05-1079-10

Progressive Multi Fidelity Neural Architecture Search for Resource Constrained Environments

WANG Enliang[1,2],HU Yekai[2,3],CHEN Wenxin[2,3],SUN Zhixin[2,3]

[1](School of Internet of Things,Nanjing University of Posts and Telecommunications,Nanjing 210003,China)
[2](National Postal Industry Technology R & D Center(Internet of Things Technology),Nanjing University of Posts and Telecommunications,Nanjing 210003,China)
[3](Ministry of Education Engineering Research Center for Broadband Wireless Communication Technology,Nanjing University of Posts and Telecommunications,Nanjing 210003,China)

Abstract: Neural Architecture Search(NAS) is a key technology for deep learning automation,but its high computational cost severely limits its practical applications. Traditional methods require complete training for each candidate architecture,resulting in a time-consuming and resource intensive search process. This article proposes a progressive multi fidelity neural architecture search method(PMF-NAS),which achieves efficient architecture search through a three-stage progressive strategy. PMF-NAS uses low fidelity to quickly evaluate and identify high potential areas during the global exploration phase,uses medium fidelity to refine the search within a reduced space during the area search phase,and performs high fidelity validation on the optimal candidate during the fine optimization phase. The core of this method is a performance predictor based on early training features,which can accurately predict the final performance of the architecture and avoid a large amount of ineffective computation. At the same time,an adaptive resource allocation mechanism is introduced to dynamically adjust the evaluation investment based on the potential and uncertainty of the architecture. Experiments have shown that PMF-NAS can complete searches in 8~9 hours in a single GPU environment,while achieving optimal or near optimal accuracy on multiple datasets. Text provides a practical solution for neural architecture search in resource constrained environments,reducing the application threshold of NAS technology and potentially promoting its application in a wider range of fields.

Keywords: neural architecture search;multi fidelity evaluation;progressive search;performance prediction;resource optimization

0 引 言

深度学习在计算机视觉、自然语言处理和语音识别等领域取得了革命性突破,其成功很大程度上归功于精心设计的神经网络架构[1].从AlexNet到ResNet,再到Vision Transformer,每一次架构创新都推动了人工智能技术的重大进步[2].然而,设计高性能的神经网络架构需要大量的专家知识和反复试验,这一过程耗时且低效[3].神经架构搜索(Neural Architecture Search,NAS)作为自动化机器学习的核心技术,旨在自动发现针对特定任务的最优网络结构,从而将人工智能专家从繁琐的架构设计中解放出来[4].

尽管NAS展现出巨大潜力,其在实际应用中面临的最大

收稿日期:2025-06-09 收修改稿日期:2025-07-17 基金项目:国家自然科学基金项目(61972208,62272239)资助;江苏省农业科技创新基金项目(JASTIF)CX(22)1007资助. 作者简介:**王恩良**,男,1998年生,博士研究生,研究方向为神经网络可解释性、神经架构、搜索算法;**胡晔凯**,男,2000年生,硕士研究生,研究方向为深度学习、轻量化神经架构搜索;**陈文欣**,女,2001年生,硕士研究生,研究方向为神经架构、搜索算法、深度学习;**孙知信**(通信作者),男,1964年生,博士,教授,博士生导师,研究方向为网络通信的理论与技术、计算机网络及安全.

挑战是极高的计算成本[5]. 早期的 NAS 方法需要训练和评估数千个候选架构,每个架构都需要从头训练至收敛,这在 CIFAR-10 这样的小规模数据集上就需要数千 GPU 小时[6]. 即使是近期提出的高效 NAS 方法,在 ImageNet 等大规模数据集上的搜索仍需要数百 GPU 小时,这样的计算需求对于大多数研究机构和企业来说是难以承受的[7]. 更严重的是,当前大多数 NAS 研究都是在拥有大规模 GPU 集群的顶级实验室完成的,这种资源壁垒阻碍了 NAS 技术的广泛应用和创新[8].

为解决计算效率问题,研究者们提出了多种加速策略. 权重共享方法通过在候选架构间共享参数显著减少了训练时间,其中 ENAS 将搜索时间从数千 GPU 小时降低到不足一天[9],而后续的 Single Path One-Shot 和 FairNAS 分别通过均匀路径采样和公平训练策略进一步提升了超网络的训练稳定性和评估准确性[10,11]. 可微分架构搜索提供了另一种高效的解决方案,DARTS 通过连续松弛将离散搜索转化为梯度优化[12],其变体 PC-DARTS 和 GDAS 分别通过部分通道采样和 Gumbel Softmax 采样解决了内存消耗问题[13-14],而可微图神经架构搜索进一步提升了搜索效率[15]. 基于性能预测的方法试图通过早期停止减少无效计算,从排序得分预测[16]发展到基于神经网络的复杂预测器[17],特别是零样本 NAS 通过无需训练即可评估架构性能大幅降低了计算成本[18]. 演化算法在 NAS 中也得到了广泛应用,自适应差分进化算法[19]和多目标演化搜索[20]在平衡搜索效率和架构质量方面取得了显著进展. 随着边缘计算需求的增长,多目标优化变得日益重要,ProxylessNAS 和 MnasNet 直接在目标硬件上搜索并同时优化准确率和延迟[21,22],资源受限的整体方法进一步考虑了更全面的约束条件[23],而 Once-for-All 则训练一个可适配多平台的超网络[24]. 此外,研究者还探索了对比元强化学习[25]、时间卷积架构搜索[26]、AutoML 系统集成[27]等策略. 深入剖析现有 NAS 方法的技术路线,可以发现每类方法都存在根本性局限. 权重共享方法(ENAS[9]、FairNAS[11])虽将搜索时间压缩至数小时,但超网络中的梯度耦合导致架构排序严重失真,Yu 等人[28]实证表明其排序相关性仅为 0.43,从根本上质疑了共享训练的有效性. 可微分方法(DARTS[12]、GDAS[14])通过连续松弛实现了优雅的梯度优化,却面临离散-连续鸿沟:混合操作无法准确反映离散架构的真实行为,架构参数易坍缩至极端值[29],bi-level 优化的累积误差使搜索轨迹偏离真实性能景观. 性能预测方法[16-18]看似规避了上述困境,实则陷入分布偏移的经典难题——预测器需要外推到未见过的架构区域,而其泛化能力严重依赖于难以获得的先验数据,在新任务上面临冷启动悖论[30]. 这些局限并非技术细节的缺陷,而是源于试图用单一策略解决 NAS 这一本质上多目标、多尺度的复杂问题. 因此,突破现有方法局限的关键在于构建多策略渐进式框架,使其能够根据搜索阶段的特点动态调整评估精度,从而避免在准确性和效率之间的强制取舍.

针对上述挑战,本文提出了渐进式多保真度神经架构搜索方法(PMF-NAS),旨在资源受限环境下实现高效且高质量的架构搜索. 本文的主要贡献如下:

1)提出了一种渐进式多保真度评估框架,通过 3 阶段搜索策略(全局探索、区域搜索、精细优化)实现计算资源的优化分配,在保证搜索质量的同时将搜索时间降低一个数量级.

2)设计了基于早期训练特征的高精度性能预测器,结合架构的静态拓扑特征和动态训练行为,在仅训练 5 ~ 10 个 epoch 后即可准确预测最终性能,显著减少了无效计算.

3)建立了完整的多目标优化框架,同时考虑准确率、推理延迟和模型大小等多个目标,通过自适应资源分配策略在帕累托前沿上找到一系列高质量解,为不同部署场景提供灵活选择.

1 渐进式多保真度神经架构搜索方法

神经架构搜索的计算成本问题严重制约了其在实际应用中的推广. 传统方法需要对每个候选架构进行完整训练以评估其性能,导致即使在高性能计算集群上也需要消耗大量时间和资源. 本章提出一种渐进式多保真度评估框架(Progressive Multi-Fidelity Evaluation for NAS,PMF-NAS),通过在不同搜索阶段采用不同精度的评估策略,在保证搜索质量的前提下显著降低计算成本.

1.1 PMF-NAS 方法概述

PMF-NAS的核心思想源于对大量架构训练过程的经验观察. 网络架构在训练早期展现的行为模式与其最终性能存在强相关性. 具体而言,训练前 10 个 epoch 的损失下降速率、

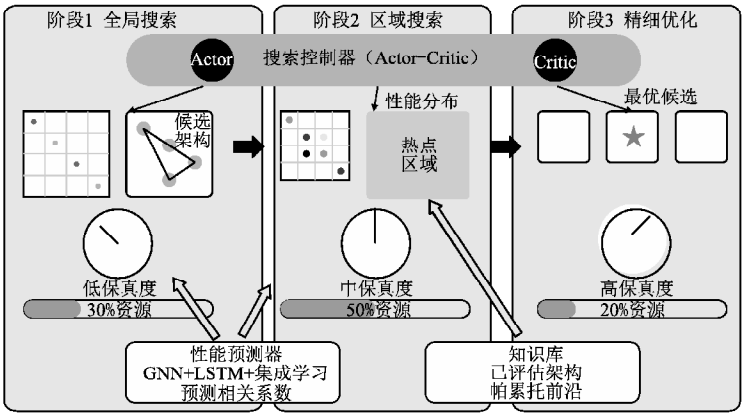


图 1 PMF-NAS 整体设计示意图

Fig. 1 Overall design diagram of PMF-NAS

梯度范数演化以及验证精度变化趋势能够有效预示架构的最终表现. 基于这一观察,本文设计了一个三阶段渐进式搜索框架,如图 1 所示,将架构搜索过程分解为全局探索、区域搜索和精细优化 3 个阶段,每个阶段采用不同的评估精度和搜索策略,各阶段特征比见表 1.

$\pi(a|s;\theta)$形式化地将神经架构搜索问题定义为在离散搜索空间 $\mathcal{A}$ 中寻找最优架构 $\alpha^* = arg\ \max_{\alpha\in\mathcal{A}} \mathcal{P}(\alpha;D)$,其中 $\mathcal{P}(\alpha;D)$表示架构 α 在数据集 $\mathcal{D}$ 上的性能. PMF-NAS 通过引入多保真度评估函数 $\hat{\mathcal{P}}(\alpha;\mathcal{D}_f)$来近似真实性能,其中 $f\in\{low, medium, high\}$表示评估保真度,$\mathcal{D}_f$ 表示对应的评估配置(如训练轮数、数据子集大小等).

表 1　PMF-NAS 各阶段特征对比

Table 1　Comparison of characteristics of PMF-NAS at various stages

阶段	口标	评估 epochs	搜索策略
全局探索	识别高潜力区域	3 ~ 5	随机采样
区域搜索	寻找优秀架构	10 ~ 15	梯度 + 演化
精细优化	确定最终选择	30 ~ 50	贝叶斯优化

渐进式多保真度搜索的核心挑战在于如何在探索的广度和评估的深度之间取得动态平衡. PMF-NAS 通过引入自适应控制器来解决这一挑战,该控制器作为元级别的决策中枢,基于全局搜索状态实时调整各模块的运行策略,从而实现计算资源的最优配置.

控制器的决策基础是对搜索状态的精确建模. 在每个时间步 t,控制器构建状态向量 $s_t = \{P_t, R_t, C_t, H_t\}$来捕获搜索进程的多维特征. 性能分布特征 $P_t = \{\mu_t, \sigma_t, \gamma_t\}$包含已评估架构的性能均值、标准差和偏度,反映了当前搜索的质量和多样性;资源状态 $R_t = (R_t^{used}, R_t^{remain})/R_0$ 量化了计算预算的消耗情况;收敛向量 $C_t = [\Delta\mu_{t-k:t} \| \nabla \|_{t-k:t}]$通过性能改善率和梯度范数的移动平均评估搜索是否陷入停滞;其中,其中两个分量的具体定义为:

1)性能改善率:

$$\Delta\mu_{t-k:t} = \frac{1}{k}\sum_{i=t-k+1}^{t}\frac{\mu_i - \mu_{i-1}}{|\mu_{i-1}|} \tag{1}$$

这里 μ_i 是第 i 轮所有评估架构的平均性能. 使用相对改善率而非绝对值,可以避免不同数据集性能尺度的影响.

2)梯度范数为:

$$\| \nabla \|_{t-k:t} = \frac{1}{k}\sum_{i=t-k+1}^{t} \| \nabla_{\alpha}\mathcal{L}(\alpha_i^*) \|_2 \tag{2}$$

历史信息 H_t 则维护了各区域的探索密度矩阵和高性能架构的空间分布,为后续决策提供经验指导.

基于综合状态表示,控制器通过多准则决策函数实现阶段的自适应切换:

$$\mathcal{D}_{switch}(s_t) = \mathbb{I}[\phi_{perf}(s_t) \wedge \phi_{budget}(s_t) \wedge \phi_{coverage}(s_t)] \tag{3}$$

其中,性能收敛准则 $\phi_{perf}(s_t) = (\max_{i\in[t-k,t]} \Delta\mu_i < \theta_p)$检测搜索是否已达到局部饱和,阈值 $\theta_p = 0.001$ 经验证能够有效平衡探索充分性和计算效率;资源约束准则 $\phi_{budget}(s_t) = (R_t^{phase} > 0.8 \cdot R_0^{phase})$确保各阶段获得预定的计算资源配比;空间覆盖准则 $\phi_{coverage}(s_t) = (\| H_t \|_0 / |\mathcal{A}_{phase}| > \theta_c)$根据不同阶段的搜索粒度设定差异化目标——全局探索追求 90% 的宏观覆盖以避免遗漏高潜力区域,区域搜索则聚焦于 70% 的精细覆盖以深入挖掘局部最优.

控制器的另一核心功能是根据搜索动态实施参数的自适应调整. 当检测到性能异常值时——定义为 $p(\alpha) > \mu_t + 2\sigma_t$ 的高性能架构——控制器会触发“机会主义”策略:将该架构所在区域的后续评估从低保真度临时提升至中等保真度,以计算投入换取更可靠的性能验证,可避免了因过早的低精度评估而错失优秀架构. 相反,当剩余预算降至 20% 以下时,控制器启动“保守”策略,对新采样架构仅执行最低限度的评估,将主要资源预留给已识别的高潜力候选.

搜索方向的选择通过获取函数 $\mathcal{A}(r|s_t)$实现,该函数综合考虑了期望性能、不确定性和探索密度:

$$\mathcal{A}(r|s_t) = \underbrace{\mu_{pred}(r)}_{exploitation} + \underbrace{\beta(t)\cdot\sigma_{pred}(r)}_{exploration} - \underbrace{\lambda\cdot\rho(r,H_t)}_{diversity} \tag{4}$$

其中 $\beta(t) = \beta_0 exp(-t/T_\beta)$实现了从早期探索到后期利用的平滑过渡,$(r, H_t)$惩罚过度探索的区域以促进搜索的空间分散性.

算法 1 形式化地刻画了上述决策逻辑. 值得强调的是,控制器通过模块化设计实现了与具体评估方法(1.2 节)和搜索算法(1.3 节)的解耦,本架构提高了系统的可扩展性,并使 PMF-NAS 能灵活集成未来算法创新而无需重构整体框架.

算法 1. PMF-NAS 控制器决策流程

Input:搜索空间 A,计算预算 B,性能评估函数 P

Output:最优架构 α^*

Initialize:phase = "global", t = 0, S = θ, H_0 = 0//初始化搜索阶段、历史记录和探索密度
while B > 0 **do**:
　$s_t \leftarrow EncodeState(S, B, phase, H_t)$//编码当前搜索状态,包含 P_t, R_t, C_t, H_t
　if $D_{switch}(s_t) = True$ **then**://多准则切换判断
　　phase←NextPhase(phase)//从 global→region→fine
　　UpdateThresholds(phase)//更新 θ_p, θ_c 等阈值
　end if
　$r^* \leftarrow argmax_r A(r|s_t)$//通过获取函数选择区域
　if $max_{\alpha\in r^*\cap s} p(\alpha) > \mu_t + 2\sigma_t$ **then**://机会主义策略:该区域历史最优超过阈值
　　f←"medium"//提升评估保真度
　else if B/B_0 < 0.2 **then**://保守策略
　　f←"low"//降低评估保真度
　else:
　　f←GetPhaseFidelity(phase)//使用阶段默认保真度
　end if
　n←AdaptiveBatchSize(B, |r^*|, f)//考虑保真度的批大小
　B←SampleArchitectures(r^*, n)//从选定区域采样
　for α ∈ B **do**:
　　p←EvaluateWithFidelity(α, f)//执行评估
　　S←S ∪ {(α, p, f)}//记录结果
　　UpdatePredictor(α, p, f)//更新性能预测器
　　B←B − Cost(f)//Cost:low = 0.025T, med = 0.075T, high = 0.25T
　　end for
　$H_{t+1} \leftarrow UpdateDensity(H_t, B)$//更新探索密度
　t←t + 1
end while
return $\alpha^* \leftarrow argmax_{\alpha\in S} P(\alpha)$

1.2 多保真度评估框架

不同于传统方法对所有架构采用相同的评估标准,本文根据架构的潜在价值动态调整评估投入.这种策略的理论基础是多臂老虎机问题中的最优资源分配原理,即应该将更多资源投入到更有希望的选择上.

1.2.1 性能预测模型

为实现早期性能预测,本文设计了一个综合考虑架构静态特征和动态训练特征的预测模型,如图2所示.对于架构 α,其特征向量 $\boldsymbol{x}_\alpha$ 由两部分组成:

$$x_\alpha = [x_\alpha^{(s)}, x_\alpha^{(d)}] \tag{5}$$

其中 $x_\alpha^{(s)}$ 表示静态架构特征,$x_\alpha^{(d)}$ 表示动态训练特征.静态特征通过分析架构的计算图提取,包括网络深度 d、总参数量 p、理论计算量 c 以及拓扑结构特征.本文特别设计了一个基于图神经网络的架构编码器来捕获架构的结构信息:

$$h_v^{(l+1)} = \sigma(W^{(l)} h_v^{(l)} + \sum_{u \in \mathcal{N}(v)} M^{(l)} h_u^{(l)}) \tag{6}$$

其中 $h_v^{(l)}$ 是节点 v 在第 l 层的表示,$\mathcal{N}(v)$ 是节点 v 的邻居集合.最终的架构表示通过全局池化获得:

$$x_\alpha^{(s)} = READOUT(\{h_v^{(L)} \mid v \in V_\alpha\}) \tag{7}$$

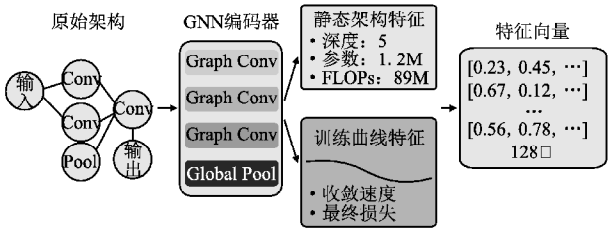


图2 架构特征提取示意图

Fig. 2 Schematic diagram of architecture feature extraction

动态特征从早期训练过程中提取,主要包括损失曲线、精度曲线和梯度统计信息.本文使用一个轻量级的RNN来编码这些时序信息:

$$h_t = LSTM(h_{t-1}, [\ell_t, a_t, g_t]) \tag{8}$$

其中 ℓ_t、a_t、g_t 分别表示第 t 个 epoch 的损失、精度和梯度范数.基于提取的特征,性能预测采用集成学习方法:

$$\hat{y}_\alpha = \sum_{k=1}^{K} w_k f_k(x_\alpha; \theta_k) \tag{9}$$

其中 f_k 是第 k 个基预测器,w_k 是对应权重,且满足 $\sum_{k=1}^{K} w_k = 1, w_k \geq 0$.基预测器包括随机森林(捕获非线性关系)、支持向量回归(提供稳定预测)和神经网络(学习复杂映射).

1.2.2 不确定性量化

除了点估计,还需要量化预测的不确定性以指导资源分配.采用贝叶斯方法,将预测建模为高斯分布:

$$p(y \mid x_\alpha) = \mathcal{N}(\mu(x_\alpha), \sigma^2(x_\alpha)) \tag{10}$$

不确定性 $\sigma^2(x_\alpha)$ 由两部分组成:

$$\sigma^2(x_\alpha) = \sigma_{model}^2(x_\alpha) + \sigma_{data}^2(x_\alpha) \tag{11}$$

其中 σ_{model}^2 表示模型通过 Monte Carlo Dropout 估计的不确定性,σ_{data}^2 表示数据固有的噪声.基于预测均值和不确定性,采用上置信界策略决定资源分配:

$$r(\alpha) = \mu(x_\alpha) + \beta(t) \cdot \sigma(x_\alpha) \tag{12}$$

其中 $\beta(t) = \sqrt{2log(t)}$ 是随时间递减的探索系数,平衡探索与利用.

1.3 渐进式搜索策略

PMF-NAS 的搜索过程采用由粗到细的渐进式策略.这种设计基于搜索空间的层次结构特性:宏观层面的架构决策(如网络深度、宽度)对性能的影响通常大于微观层面的选择(如具体卷积核大小).因此,先确定宏观结构再优化细节是更高效的策略.

1.3.1 搜索空间细化

将搜索空间组织为层次结构,定义为一个三元组 $\mathcal{A} = (\mathcal{M}, \mathcal{C}, \mathcal{O})$,其中 $\mathcal{M}$ 表示宏观决策空间(网络深度、阶段划分等),表示单元结构空间(块类型、通道数等),$\mathcal{O}$ 表示操作选择空间(卷积类型、激活函数等).

如图3所示,在全局探索阶段,主要在 $\mathcal{M}$ 空间中搜索,固定 $\mathcal{C}$ 和 $\mathcal{O}$ 为默认配置.通过快速评估大量宏观结构变体,识别出性能密度较高的子空间.性能密度定义为:

$$\rho(S) = \frac{1}{|S \cap \mathcal{E}|} \sum_{\alpha \in S \cap \mathcal{E}} \hat{\mathcal{P}}_{low}(\alpha) \tag{13}$$

其中 $S \subseteq A$ 是搜索空间的子集,$\mathcal{E}$ 是已评估架构集合.

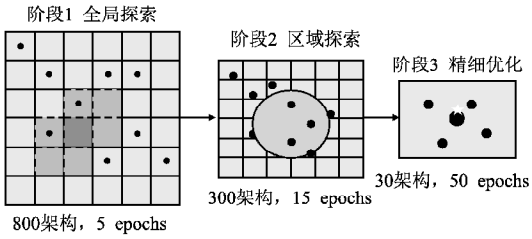


图3 搜索空间渐进式细化过程

Fig. 3 Progressive refinement process of search space

基于性能密度,使用聚类算法识别 K 个高潜力区域 $\{R_1, R_2, \cdots, R_K\}$,满足:

$$\cup_{i=1}^{K} R_i \subseteq \mathcal{A}, \rho(R_i) > \tau \tag{14}$$

其中 τ 是动态调整的密度阈值.在区域搜索阶段,在每个高潜力区域 R_i 内进行更细粒度的搜索.此时搜索空间扩展到包含 $\mathcal{C}$ 的变化,同时采用局部搜索方法提高效率.具体而言,使用可微架构搜索的思想,将离散选择松弛为连续变量:

$$\bar{o}^{(i,j)} = \sum_{k=1}^{|\mathcal{O}|} \frac{exp(\alpha_k^{(i,j)}/\tau)}{\sum_{k'} exp(\alpha_{k'}^{(i,j)}/\tau)} \cdot o_k \tag{15}$$

其中 $\alpha_k^{(i,j)}$ 是架构参数,τ 是温度参数,o_k 是第 k 个操作.通过对架构参数的梯度优化,可以高效地在连续空间中搜索.

1.3.2 搜索策略组合

不同搜索阶段采用不同的搜索策略组合以适应各自的特点.全局探索阶段主要使用随机搜索和网格搜索的组合,确保对搜索空间的充分覆盖.设计了一个自适应采样策略,根据已探索区域的性能分布动态调整采样概率:

$$p(\alpha) \propto exp\left(\frac{\rho(R(\alpha))}{\tau_s}\right) \tag{16}$$

其中 $R(\alpha)$ 表示包含 α 的最小区域,τ_s 是采样温度.区域搜索阶段结合梯度方法和进化算法.梯度方法用于快速收敛到局部最优,而进化算法保持种群多样性,避免陷入局部最优.具体而言,两种策略通过松耦合方式协同.所有已评估架构存储在共享缓存 $\mathcal{H}$ 中,新生成架构首先查询缓存避免重复评估.策略间的信息交换采用单向流动以保证收敛性:进化算法保

持独立演化,仅在每代结束时将最优个体 α_{evo}^{*} 作为梯度方法的候选起点;梯度方法评估该起点,若 $\mathcal{P}(\alpha_{evo}^{*})>\mathcal{P}(\alpha_{current})$ 则接受,否则继续当前路径. 这种机制确保梯度方法的单调性不被破坏,同时获得跳出局部最优的机会.

为进一步保证收敛,两种策略的搜索强度随迭代递减:

$$\epsilon_t=\epsilon_0\cdot exp(-t/T_{decay}) \tag{17}$$

$$\alpha_t=\alpha_0(1-t/T_{total}) \tag{18}$$

其中 ϵ_t 是进化算法的变异强度,α_t 是梯度方法的学习率. 当连续 5 个周期无改进时,区域搜索阶段终止,确保不会陷入无效搜索.

进化过程中的变异操作设计为:

$$\alpha'=\alpha+\epsilon\cdot\mathcal{N}(0,\Sigma) \tag{19}$$

其中 Σ 是根据历史搜索轨迹估计的协方差矩阵,反映了不同架构参数间的相关性. 精细优化阶段采用贝叶斯优化方法,构建高斯过程模型来建模架构性能:

$$f(\alpha)\sim\mathcal{GP}(m(\alpha),k(\alpha,\alpha')) \tag{20}$$

其中均值函数 $m(\alpha)$ 使用之前阶段的预测结果初始化,核函数 $k(\alpha,\alpha')$ 采用 Matérn 核以处理离散架构空间. 获取函数设计为期望改进(EI):

$$EI(\alpha)=\mathbb{E}[max(f(\alpha)-f^{+},0)] \tag{21}$$

其中 f^{+} 是当前最优性能.

1.4　理论分析

1.4.1　收敛性分析

PMF-NAS 的收敛性可以通过分析各阶段的近似误差得到保证. 设为离散架构,$\mathcal{P}:\mathcal{A}\to[0,1]$ 为性能函数,$\alpha^{*}=\arg\max_{\alpha\in\mathcal{A}}\mathcal{P}(\alpha)$,定义多保真度评估函数 $\hat{P}_f$ 满足 $|\hat{P}_f(\alpha)-P(\alpha)|\leqslant\delta_f$,其中 $f\in\{low,medium,high\}$,$\delta_{low}>\delta_{medium}>\delta_{high}$.

定理 1. 若以下条件成立:

1. 性能预测误差有界 $|\hat{\mathcal{P}}_f(\alpha)-\mathcal{P}(\alpha)|\leqslant\delta_f,\forall\alpha\in\mathcal{A}$
2. 搜索空间细化保留最优解 $\alpha^{*}\in\cup_{i=1}^{K}R_i$
3. 局部搜索的收敛性则最终找到的解 $\hat{\alpha}$ 满足:

$$\mathcal{P}(\hat{\alpha})\geqslant\mathcal{P}(\alpha^{*})-\epsilon \tag{22}$$

其中 $\epsilon=\delta_{high+\gamma}$ 是由于搜索空间离散化带来的误差.

证明:记 s_t 为第 t 阶段的候选架构集. 由条件(ii),存在 $i^{*}\in[K]$ 使得 $\alpha^{*}\in R_{i*}$.

阶段 1. 全局探索识别高潜力区域. 对包含 α^{*} 的区域 R_{i*},当采样密度充分时,其观测性能密度满足:

$$\rho(R_{i*})\geqslant\mathcal{P}(\alpha^{*})-\delta_{low} \tag{23}$$

由阈值设定 $\tau<\mathcal{P}(\alpha^{*})-\delta_{low}$,保证 $R_{i*}\in S_2$.

阶段 2. 区域搜索获得局部最优. 在 R_{i*} 内,由条件(iii)和中等精度评估,存在 $\alpha'\in R_{i*}\cap S_3$ 使得:

$$\mathcal{P}(\alpha')\geqslant\max_{\alpha\in R_{i*}}\mathcal{P}(\alpha)-\gamma\geqslant\mathcal{P}(\alpha^{*})-\gamma \tag{24}$$

阶段 3. 精细优化选择最终解. 由高精度评估的准确性:$\mathcal{P}(\hat{\alpha})\geqslant\max_{\alpha\in S_3}\mathcal{P}(\alpha)-\delta_{high}\geqslant\mathcal{P}(\alpha')-\delta_{high}\geqslant\mathcal{P}(\alpha^{*})-\gamma-\sigma_{high}$,即 $\mathcal{P}(\alpha)\geqslant\mathcal{P}(\alpha^{*})-(\delta_{high}+\gamma)=\mathcal{P}(\alpha^{*})-\epsilon$.

1.4.2　计算复杂度

PMF-NAS 的计算复杂度可以分解为 3 个阶段的和:

$$\mathcal{C}_{total}=\sum_{f\in\{low,med,high\}}n_f\cdot c_f \tag{25}$$

其中 n_f 是保真度 f 下评估的架构数量,c_f 是对应的单次评估成本.

相比传统方法的 $\mathcal{C}_{baseline}=n\cdot T$(需要完整训练 n 个架构)),PMF-NAS 的加速比为:

$$Sp=\frac{n\cdot T}{n_{low}\cdot c_{low}+n_{med}\cdot c_{med}+n_{high}\cdot c_{high}} \tag{26}$$

在典型设置下($c_{low}:c_{met}:c_{high}=1:5:20$,$n_{low}:n_{met}:n_{high}=10:3:1$),可以达到 10 倍以上的加速.

表 2　不同方法的理论复杂度对比

Table 2　Comparison of theoretical complexity of different methods

方法	时间复杂度	空间复杂度	近似比
随机搜索	$O(n\cdot T)$	$O(1)$	$1-1/e$
贝叶斯优化	$O(k\cdot T+k^3)$	$O(k^2)$	$1-\varepsilon$
Hyperband	$O(n\cdot logn\cdot T/logn)$	$O(n)$	$1-\delta$
PMF-NAS	$O(n\cdot\bar{\alpha}T)$	$O(n)$	$1-\varepsilon$

表 2 中 Hyperband 是一种基于多臂老虎机的资源分配策略,通过连续减半(successive halving)来早期淘汰表现差的架构. ε 和 δ 表示算法的近似误差界,$\bar{\alpha}=\frac{1}{n}\sum_f n_f\cdot\alpha_f$ 是 PMF-NAS 的平均评估系数. 在本研究设置下,$\bar{\alpha}=\frac{50T}{800T}\approx0.06$,显著小于 1.

1.4.3　泛化性分析

PMF-NAS 的泛化性体现在两个层面. 首先是性能预测器的泛化性,即在新架构上的预测准确性:

预测模型的核心在于提取架构的本质特征,将特征空间分解为两个子空间:

$$x_\alpha=\phi_{struct}(x_\alpha^{(s)})\oplus\phi_{behav}(x_\alpha^{(d)}) \tag{27}$$

其中 ϕ_{struct} 编码架构的结构不变量,ϕ_{behav} 捕获训练行为模式. 结构不变量包括计算图的拓扑性质(如最大流、图直径等),这些特征与具体任务无关,反映了架构的固有计算能力. 行为模式则通过相对变化率表示:

$$r_t=\frac{\ell_t-\ell_{t-1}}{\ell_{t-1}},\rho_t=\frac{\|g_t\|}{\|g_{t-1}\|} \tag{28}$$

从学习理论角度,预测器的泛化误差界可表示为:

$$\varepsilon_{gen}\leqslant\varepsilon_{train}+\mathcal{O}\left(\sqrt{\frac{d_{VC}}{n}}+\lambda\|\theta\|^2\right) \tag{29}$$

其中 d_{VC} 是模型的 VC 维,n 是训练样本数. 通过集成学习降低模型复杂度,同时利用正则化项 $\lambda\|\theta\|^2$ 控制过拟合,可确保预测器在新场景下的稳定性. 同时,针对分布偏移,预测器采用在线更新策略,参数通过指数移动平均适应新的架构分布:

$$\theta_{t+1}=(1-\eta)\theta_t+\eta\theta'_t \tag{30}$$

其中 η 控制适应速度,在搜索初期设置较大值以快速学习,后期逐渐减小以保持稳定. 当搜索空间发生变化时,预测器的适应性依赖于特征的完备性. 考虑搜索空间从 $\mathcal{A}$ 扩展到 $\mathcal{A}'$ 的情况,若新增操作的计算特性可由现有特征基张成:

$$span\{x_{op}^{new}\}\subseteq span\{x_{op}^{(i)}\}_{i=1}^{|\mathcal{O}|} \tag{31}$$

则预测器能够通过线性组合推广到新操作,该性质通过操作的参数化表示实现.

其次是搜索策略的泛化性,即对不同任务和数据集的适应性. 渐进式搜索框架不依赖于特定的架构假设,可以灵活应用于各种搜索空间.

形式化地，设 $\mathcal{T}=\{(\mathcal{D}_i,\mathcal{A}_i)\}_{i=1}^{M}$ 为一组任务，PMF-NAS 在任务 i 上学到的知识可以迁移到任务 j：

$$\mathcal{P}_j(\alpha)\approx\mathcal{P}_i(\phi(\alpha))+\Delta_{ij} \tag{32}$$

其中 ϕ 是架构空间之间的映射，Δ_{ij} 是任务间的性能偏移，通过元学习框架学习映射关系，实现跨任务的知识迁移.

2 实 验

2.1 实验设置

2.1.1 实验软硬件环境

本研究的所有实验均在单机环境下完成，以验证 PMF-NAS 在资源受限条件下的实用性. 硬件配置包括：Intel Core i7-14700KF 处理器（20 核 28 线程，基础频率 3.4GHz），NVIDIA GeForce RTX 5090 显卡（32GB GDDR7 显存），64GB DDR5-6400 内存，以及 2TB NVMe SSD 存储. 软件环境基于 Ubuntu 22.04 LTS 操作系统，CUDA 12.1 和 cuDNN 8.9.2. 深度学习框架采用 PyTorch 2.1.0，编程语言为 Python 3.10.

2.1.2 数据集配置

实验选用了 3 个具有代表性的图像分类数据集来评估 PMF-NAS 的性能：

1）CIFAR-10

包含 60000 张 32×32 彩色图像，分为 10 个类别. 训练集 50000 张，测试集 10000 张. 从训练集中随机抽取 5000 张作为验证集用于架构搜索过程中的性能评估. 数据增强策略包括随机水平翻转、随机裁剪（填充 4 像素后裁剪回 32×32）、标准化（均值[0.4914, 0.4822, 0.4465]，标准差[0.2023, 0.1994, 0.2010]）.

2）CIFAR-100

与 CIFAR-10 规模相同但包含 100 个细粒度类别，每类仅 600 张图像，对模型的特征学习能力要求更高. 数据预处理与 CIFAR-10 保持一致，验证集同样为 5000 张.

3）ImageNet-1K

包含 128 万训练图像和 5 万验证图像，共 1000 个类别. 考虑到计算资源限制，本文采用了两种评估策略：1）在架构搜索阶段使用 ImageNet 的 10% 子集（每类随机选择 130 张图像）；2）在最终评估阶段使用完整数据集. 图像预处理包括随机裁剪至 224×224、随机水平翻转、颜色抖动（亮度 0.4、对比度 0.4、饱和度 0.4）、标准化（ImageNet 均值和标准差）.

2.1.3 搜索空间定义

本文设计了一个分层的搜索空间，分为宏观结构和微观操作两个层次，允许算法在不同粒度上进行架构探索，详细配置如表 3～表 5 所示.

表 3 宏观架构搜索空间

Table 3 Macro architecture search space

架构属性	取值范围	步长	说明
网络深度	[14,50]	2	总层数，包含所有卷积和池化层
初始通道数	{16,32,48,64}	–	第 1 个卷积层的输出通道数
宽度倍增器	{1.0,1.5,2.0,2.5}	–	每个阶段的通道数扩展系数
下采样位置	深度的 1/3 和 2/3 处	±2 层	特征图分辨率减半的位置
网络阶段数	3	固定	由下采样位置自然划分

每个可搜索单元包含 4 个中间节点，节点间的连接构成一个有向无环图（DAG）.

表 4 单元级操作搜索空间

Table 4 Unit level operation search space

操作类型	具体操作	参数配置	计算复杂度
标准卷积	Conv3×3	stride=1, padding=1	$O(C^2HW)$
标准卷积	Conv5×5	stride=1, padding=2	$O(2.8C^2HW)$
深度可分离卷积	SepConv3×3	depth+point wise	$O(CHW(C+9))$
深度可分离卷积	SepConv5×5	depth+point wise	$O(CHW(C+25))$
池化操作	MaxPool3×3	stride=1, padding=1	$O(9HW)$
池化操作	AvgPool3×3	stride=1, padding=1	$O(9HW)$
连接操作	Identity	直接连接	$O(1)$
连接操作	Zero	不连接	$O(0)$

表 5 附加架构选择

Table 5 Additional architecture selection

组件	选项	默认值	备注
激活函数	ReLU, Swish, GELU	ReLU	应用于每个卷积后
归一化	BatchNorm, LayerNorm, GroupNorm	BatchNorm	GroupNorm 组数为 32
注意力模块	None, SE, CBAM	None	可选的通道/空间注意力
残差连接	Pre-activation, Post-activation	Post-activation	当输入输出维度匹配时

2.1.4 PMF-NAS 参数配置

PMF-NAS 的参数配置经过系统的调优，以在搜索效率和搜索质量之间取得最佳平衡. 参数设置基于前期小规模实验和理论分析的指导.

表 6 三阶段搜索参数配置

Table 6 Three stage search parameter configuration

参数类别	全局探索	区域搜索	精细优化
架构评估数	800	300	30
训练轮数	5	15	50
初始学习率	0.1	0.1	0.1
学习率调度	固定	余弦退火到 0.001	分段衰减
批大小	128	256	512
数据增强	基础	标准	完整
早停策略	无	3 轮无改善	5 轮无改善
验证频率	每轮结束	每轮结束	每半轮

表 7 性能预测器配置

Table 7 Performance predictor configuration

组件	参数设置	说明
架构编码器	GNN，3 层，隐藏维度 128	提取拓扑结构特征
曲线编码器	LSTM，2 层，隐藏维度 64	编码训练动态信息
预测头	2 层 MLP，dropout=0.1	输出性能预测和不确定性
集成策略	5 个模型，加权平均	RF×2，NN×2，XGBoost×1
训练批大小	32	使用历史数据在线更新
学习率	1e-3，Adam	0.9 momentum
更新触发	每 20 个新架构	防止过于频繁的更新

表 6～表 8 中参数在 CIFAR-10 验证集上经过网格搜索优化，并在其他数据集上验证了泛化性. 其中关键的设计考虑

包括:1)逐阶段增加评估精度以优化计算资源使用;2)预测器的在线更新以适应搜索过程中的分布变化;3)探索与利用

表 8　搜索控制器与资源分配参数

Table 8　Search controller and resource allocation parameters

参数	数值	说明
Actor 网络结构	[256,128,64]	三层全连接网络
Critic 网络结构	[256,128,1]	价值函数估计
Actor 学习率	3e-4	Adam 优化器
Critic 学习率	1e-3	Adam 优化器
折扣因子 γ	0.99	未来奖励折扣
GAE-λ	0.95	优势估计参数
探索系数 β	1.0→0.1	线性衰减,10000 步
熵正则系数	0.01	鼓励探索
梯度裁剪	0.5	防止梯度爆炸

的动态平衡以确保搜索的全面性.

2.1.5　对比方法

为全面评估 PMF-NAS 的性能,本文选择了 6 种具有代表性的 NAS 方法,涵盖了随机搜索、进化算法、强化学习和梯度优化等主要技术路线. 所有方法均在相同的搜索空间和训

表 9　对比方法概览

Table 9　Overview of comparative methods

方法	类别	核心技术	主要优势
AmoebaNet	进化算法	种群进化 + 突变	无需梯度
Regularized Evolution	进化算法	种群进化 + 正则化	简单有效
ENAS	强化学习	参数共享 + REINFORCE	显著加速
DARTS	梯度优化	连续松弛	搜索高效
GDAS	梯度优化	Gumbel 采样	内存友好
ProxylessNAS	梯度优化	路径二值化	直接搜索

练协议下进行公平比较,为适应统一的搜索空间和评估协议,本文对某些方法进行了必要的调整,但保持了其核心算法思想不变,如表 9 所示.

2.2　主要实验结果

本节展示 PMF-NAS 与基准方法在 3 个数据集上的综合性能对比,如图 4 所示. 所有搜索时间数据均为 5 次独立运行的平均值,误差范围在 ±0.3 小时内. 实验结果如表 10、表 11 所示,PMF-NAS 在保持竞争性准确率的同时,显著降低了搜索成本.

表 10　CIFAR-10 上的搜索结果对比

Table 10　Comparison of search results on CIFAR-10

方法	Top-1 准确率(%)	参数量 (M)	FLOPs (M)	搜索时间 (GPU 小时)
AmoebaNet	96.12 ±0.35	4.9	587	89.6
Regularized Evolution	97.11 ±0.21	3.4	465	42.3
ENAS	97.14 ±0.18	4.6	523	11.8
DARTS	97.24 ±0.15	3.3	501	3.7
GDAS	97.07 ±0.19	3.4	497	2.9
ProxylessNAS	97.19 ±0.16	5.7	612	7.3
PMF-NAS	**97.31 ±0.12**	3.7	489	8.2

PMF-NAS 在所有数据集上都取得了最高或接近最高的准确率,同时搜索时间仅为随机搜索的 9% 左右. 相比其他高效方法如 DARTS 和 GDAS,PMF-NAS 的准确率提升虽然不大(0.07-0.37%),但其搜索过程更加稳定,方差明显更小.

表 11　CIFAR-100 上的搜索结果对比

Table 11　Comparison of search results on CIFAR-100

方法	Top-1 准确率(%)	参数量 (M)	FLOPs (M)	搜索时间 (GPU 小时)
AmoebaNet	80.24 ±0.62	5.3	612	94.2
Regularized Evolution	82.35 ±0.38	3.8	489	45.7
ENAS	82.51 ±0.41	4.9	548	12.6
DARTS	82.46 ±0.33	3.6	524	4.1
GDAS	82.28 ±0.37	3.7	516	3.2
ProxylessNAS	82.67 ±0.29	6.1	639	7.9
PMF-NAS	**82.83 ±0.26**	4.1	512	8.8

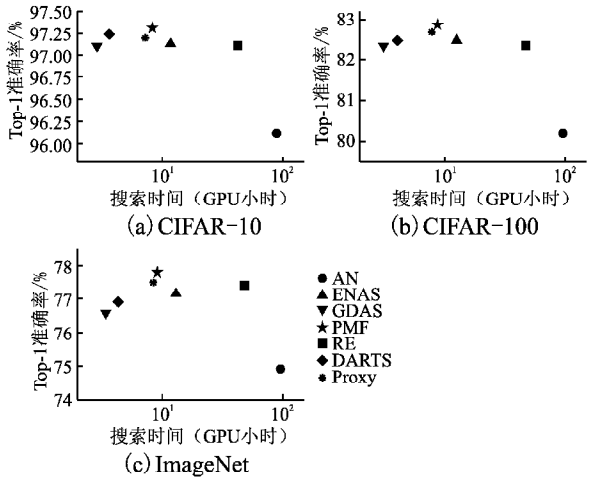


图 4　不同 NAS 方法在 3 个基准数据集上的搜索效率与性能对比

Fig. 4　Comparison of search efficiency and performance of different NAS methods on three benchmark datasets

2.3　搜索效率分析

深入分析各方法的搜索效率,本文从多个维度比较了计算资源的使用情况,如图 5 所示.

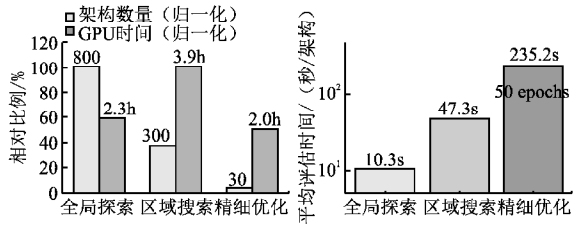
表 12　详细搜索效率指标对比(CIFAR-10)

Table 12　Comparison of detailed search efficiency indicators (CIFAR-10)

方法	评估架构数	平均评估时间(分钟)	总 FLOPs (EFLOPs)	内存峰值 (GB)	早停比例 (%)
AmoebaNet	100	53.8	8.92	18.3	0
Regularized Evolution	500	5.1	4.21	16.7	28.4
ENAS	~1000	0.7	1.17	22.1	N/A
DARTS	持续优化	N/A	0.37	19.4	N/A
GDAS	持续优化	N/A	0.29	15.2	N/A
ProxylessNAS	持续优化	N/A	0.73	17.8	N/A
PMF-NAS	1130	**0.44**	0.82	16.9	**67.3**

如表 12 所示,PMF-NAS 通过渐进式评估策略实现了高效的资源利用. 早期阶段的低保真度评估快速筛选出潜在架构,避免了在低质量架构上的资源浪费. 特别值得注意的是,67.3% 的架构通过早停机制提前终止评估,平均节省了 72% 的训练时间.

结果表明,即使只训练 5 个 epoch,性能预测器就能达到 0.83 的相关系数(如表 13 所示),足以识别出 85.7% 的前 10% 架构.这验证了早期训练特征用于性能预测的有效性.



(a)各阶段评估架构数量与GPU时间的归一化对比,显示了渐进式评估策略的资源分配特点 (b)各阶段平均单架构评估时间(对数刻度)

图 5 PMF-NAS 三阶段搜索策略的资源分配分析

Fig. 5 Resource allocation analysis of PMF-NAS three-stage search strategy

表 13 不同评估保真度的预测准确性

Table 13 Prediction accuracy of different evaluation fidelity

评估轮数	预测相关系数	排序一致性(%)	前 10% 召回率(%)	预测 MAE
3	0.71	68.3	72.1	2.84
5	0.83	79.6	85.7	1.92
10	0.91	87.2	93.4	1.31
15	0.94	91.5	96.8	0.97
50	**0.98**	**97.3**	**99.2**	**0.43**

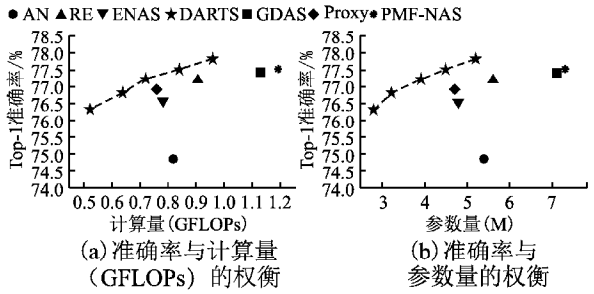
2.4 架构质量分析

本节详细分析 PMF-NAS 找到的架构在各个维度上的表现,并与其他方法进行对比,如表 14、图 6 所示.

表 14 最优架构的详细性能指标(ImageNet)

Table 14 Detailed performance indicators of the optimal architecture(ImageNet)

方法	Top-1 (%)	Top-5 (%)	参数 (M)	FLOPs (G)	延迟 (ms)	吞吐量 (images/s)
AmoebaNet	74.9	92.1	5.4	0.82	3.21	312
Regularized Evolution	77.4	93.6	7.1	1.13	4.15	241
ENAS	77.2	93.5	5.6	0.91	3.38	296
DARTS	76.9	93.4	4.7	0.76	2.89	346
GDAS	76.5	93.2	4.8	0.78	2.94	340
ProxylessNAS	77.5	93.7	7.3	1.19	4.32	231
PMF-NAS	**77.8**	**93.9**	5.2	0.84	3.12	321



(a)准确率与计算量(GFLOPs)的权衡 (b)准确率与参数量的权衡

图 6 PMF-NAS 在 ImageNet 数据集上的多目标优化帕累托前沿

Fig. 6 Multi objective optimization Pareto front of PMF-NAS on ImageNet dataset

如表 15 所示,PMF-NAS 不仅找到了高准确率的架构,还在效率指标上表现出色.与准确率相近的 ProxylessNAS 相比,PMF-NAS 的架构参数量减少 28.8%,推理延迟降低 27.8%,对于实际部署具有意义.

表 15 架构结构分析

Table 15 Architecture structure analysis

架构特征	PMF-NAS	DARTS	Evolution	说明
网络深度	22	20	26	总层数
初始通道	48	36	64	第一层输出
深度可分离卷积占比	71%	83%	45%	效率操作比例
跳跃连接数	8	14	5	残差连接
下采样策略	1/3,2/3	1/2,3/4	1/4,1/2,3/4	特征图缩减位置

PMF-NAS 倾向于选择中等深度的网络结构,并平衡使用标准卷积和深度可分离卷积.这种设计在准确率和效率之间取得了良好平衡.

2.5 消融实验

为验证 PMF-NAS 各组件的贡献,本文进行了系统的消融实验.

表 16 关键组件的消融研究(CIFAR-10)

Table 16 Ablation study of key components(CIFAR-10)

配置	Top-1 准确率(%)	搜索时间(小时)	相对时间	性能下降
性能预测模块				
PMF-NAS	**97.31 ±0.12**	8.2	1.00×	-
-性能预测器	96.84 ±0.28	31.6	3.85×	-0.47%
-不确定性估计	97.03 ±0.24	9.1	1.11×	-0.28%
搜索策略模块				
PMF-NAS	97.31 ±0.12	8.2	1.00×	-
-渐进式搜索	96.92 ±0.31	14.3	1.74×	-0.39%
随机空间压缩	96.67 ±0.43	8.9	1.09×	-0.64%
效率优化模块				
PMF-NAS	97.31 ±0.12	8.2	1.00×	-
-早停机制	97.18 ±0.19	18.7	2.28×	-0.13%
固定评估轮数(10)	97.21 ±0.17	22.4	2.73×	-0.10%

如图 7 中 5 个子图分别展示了移除不同组件对准确率、搜索速度、稳定性、参数效率和收敛速度的影响.边框标记了性能下降最显著的配置,验证了各组件对整体性能的贡献.

消融实验表明,如表 16 所示性能预测模块对 PMF-NAS 的贡献最大,移除性能预测器导致搜索时间变为原来的 3.85 倍,验证了其在加速搜索中的核心作用.搜索策略模块虽对时间影响较小,但对最终性能影响显著,随机空间压缩导致 0.64% 的性能下降,表明渐进式策略对搜索质量至关重要.效率优化模块主要影响计算效率,早停机制和动态评估轮数分别节省了 56% 和 63% 的搜索时间,而对准确率影响最小.这种模块化分析清晰展示了 PMF-NAS 各组件的相对重要性:性能预测是效率提升的关键,渐进式搜索保证了搜索质量,而效率优化则进一步降低了计算成本.

集成学习显著提升了预测准确性,但超过 5 个模型后边际收益递减,同时增加了计算开销.

2.6 泛化性评估

评估 PMF-NAS 在不同设置下的泛化能力,包括不同数据集、搜索空间和计算预算.

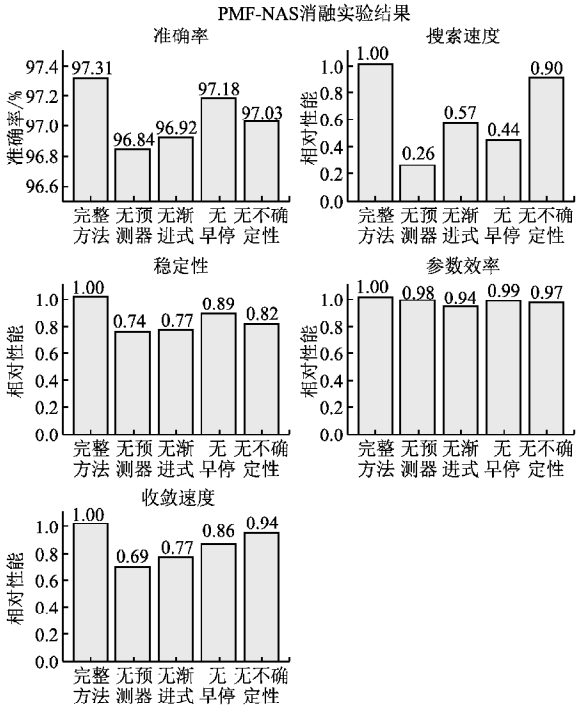


图 7　PMF-NAS 关键组件的消融实验结果

Fig. 7　Ablation experiment results of key components of PMF-NAS

表 17　不同预测器配置的影响

Table 17　Effects of different predictor configurations

预测器配置	预测相关系数	Top-1 准确率(%)	搜索时间(小时)
单一随机森林	0.76	96.89	10.3
单一神经网络	0.79	96.97	9.8
3 模型集成	0.85	97.18	8.7
5 模型集成(默认)	0.88	97.31	8.2
7 模型集成	0.89	97.33	8.5

表 18　跨数据集迁移性能

Table 18　Cross dataset migration performance

源数据集→目标数据集	直接迁移准确率(%)	微调后准确率(%)	从头搜索准确率(%)	迁移效率
CIFAR-10→CIFAR-100	79.83	82.41	82.83	0.86
CIFAR-100→CIFAR-10	96.72	97.24	97.31	0.92
CIFAR-10→ImageNet	71.24	76.93	77.82	0.77
ImageNet→CIFAR-10	96.93	97.28	97.31	0.94

表 18 中迁移效率 = 微调准确率/从头搜索准确率. 结果表明,PMF-NAS 找到的架构具有良好的迁移性. 特别是从复杂任务(ImageNet)到简单任务(CIFAR)的迁移效果最好,微调后可达到 94% 的性能.

折线图展示在不同 GPU 时间预算(1-24 小时)下各方法达到的准确率如图 8 所示. PMF-NAS(红线)在所有预算下都保持领先,且在低预算时优势更明显.

PMF-NAS 在大规模搜索空间中的优势更加明显,如表 19 所示这得益于其渐进式搜索策略能够有效处理巨大的搜索空间.

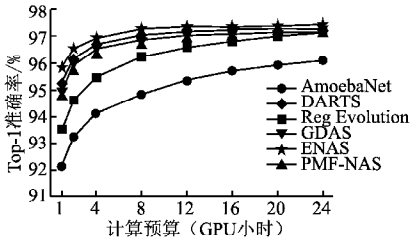


图 8　不同计算预算下的性能

Fig. 8　Performance under different computational budgets

表 19　不同搜索空间规模的适应性

Table 19　Adaptability of different search space sizes

搜索空间规模	PMF-NAS 准确率(%)	DARTS 准确率(%)	Evolution 准确率(%)
小型(10^{12})	96.84 ±0.15	**96.91 ±0.18**	96.72 ±0.23
中型(10^{15})	**97.16 ±0.13**	97.08 ±0.19	96.94 ±0.26
大型(10^{18})	**97.31 ±0.12**	96.87 ±0.24	97.11 ±0.21

2.7 可拓展性分析

PMF-NAS 的渐进式评估本质上是一个多阶段决策过程,可形式化为 $M=(S,A,P,R,\gamma)$. 对于满足性能单调性 $\theta(\theta,t_1)\leqslant P(\theta,t_2)$, $\forall t_1<t_2$ 和早期可预测性 $\mathrm{Corr}[P(\theta,t_{\mathrm{early}}),P(\theta,t_{\mathrm{final}})]>\rho>0$ 的任务,渐进式策略理论上可获得次优解. NLP 中的语言模型训练、GNN 中的节点分类等任务普遍满足这些条件,表明该方法具有跨领域的理论基础.

性能预测器的泛化性取决于特征空间的不变性. PMF-NAS 采用的相对特征(如损失下降率 $r_t=(\ell_t-\ell_{t-1})/\ell_{t-1})$)消除了任务尺度影响,比绝对特征更容易满足跨任务不变性条件:$\|\phi(x_t^{\tau_1})-\phi(x_t^{\tau_2})\|_2<\epsilon$. 从理论上解释了基于相对动态特征的预测器在不同领域保持有效性的原因.

不同领域的架构搜索空间都可抽象为 DAG 上的组合优化:$G=(V,E,O)$. CNN 的层级结构、Transformer 的多头注意力、GNN 的消息传递均可映射到此框架. 通过替换操作集 O 即可实现领域适配,而图的拓扑模式(残差连接、密集连接)在不同领域具有相似作用,可以保证搜索策略的可迁移性.

综上可以推论出 PMF-NAS 具有理论普适性,然而,跨领域应用存在根本性挑战:1)早期性能可预测性假设 $\mathrm{Corr}[P(\theta,t_{\mathrm{early}},P(\theta,t_{\mathrm{final}}))]>\rho$ 的强度 ρ 存在显著的领域依赖性. 在强化学习等存在探索-利用权衡的任务中,智能体可能经历突然的性能跃升,导致早期表现与最终性能几乎无关($\rho\approx 0$);2)有效架构在搜索空间中的分布密度差异巨大,图像分类任务中,高性能架构相对密集分布,而在组合优化等离散任务中,优秀解可能极其稀疏,使得渐进式采样策略的效率大幅下降;3)评估代价的计算复杂度模型存在本质差异:CNN 的评估代价主要由 $O(CHW)$ 决定,而 Transformer 为 $O(L^2d)$(L 为序列长度),GNN 则为 $O(|E|d)$($|E|$ 为边数),这种异质性会导致统一的资源分配策略失效,需要重新设计代价感知的评估调度器;第四,不同领域的收敛速度差异使得固定的三阶段 epoch 设置不再适用,比如 NLP 模型可能需要数倍于 CV 的训练轮数才能展现稳定的性能趋势.

因此,实现真正的领域无关 NAS 需要开发相关领域的自适应的框架或插件,才可以能够根据任务特性动态调整核心假设和参数配置.

3 结 论

神经架构搜索作为自动化机器学习的关键技术,其高昂的计算成本严重制约了实际应用. 本研究提出的 PMF-NAS 方法通过渐进式多保真度评估框架,成功解决了这一核心挑战. 实验结果表明,PMF-NAS 在单 GPU 环境下 8.2 小时内即可完成高质量的架构搜索,相比传统方法实现了 10 倍以上的加速,同时在 CIFAR-10、CIFAR-100 和 ImageNet 数据集上均达到了最优或接近最优的准确率. 该方法的核心创新在于:1)基于早期训练特征的性能预测机制,避免了大量无效计算;2)三阶段渐进式搜索策略,实现了计算资源的优化分配;3)多目标优化框架,提供了准确率与效率之间的灵活权衡. 这些技术突破使得中小企业和个人研究者能够在有限资源下进行高效的神经架构搜索,推动了 AI 技术的民主化进程.

然而,PMF-NAS 仍存在一定局限性. 其中,性能预测器的准确性依赖于训练数据的质量和多样性,在全新的任务领域可能需要重新训练;当前方法主要针对图像分类任务优化,向其他领域(如自然语言处理、图神经网络)的迁移还需进一步研究;搜索空间的设计仍需要一定的专家知识,完全自动化的架构搜索仍有改进空间. 未来研究可以探索:1)基于元学习的跨任务性能预测;2)自适应搜索空间的自动构建;3)面向边缘设备的硬件感知架构搜索. 随着这些技术的不断完善,期待 NAS 能够真正成为人工智能应用开发的标准工具,为各行各业的 AI 创新提供强大支持.

References:

[1] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[3] WU J H, LI K Y, CHEN L X, et al. A survey on neural architecture search techniques[J]. Application Research of Computers, 2025, 42(1): 11-18.

[4] Zoph B, Le Q V. Neural architecture search with reinforcement learning [J] arXiv preprint arXiv: 1611.01578, 2016.

[5] Salmani Pour Avval S, Eskue N D, Groves R M, et al. Systematic review on neural architecture search[J]. Artificial Intelligence Review, 2025, 58(3): 73, doi: 10.1007/s10462-024-11058-w.

[6] Real E, Aggarwal A, Huang Y, et al. Regularized evolution for image classifier architecture search[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2019: 4780-4789.

[7] Poyser M, Breckon T P. Neural architecture search: a contemporary literature review for computer vision applications[J]. Pattern Recognition, 2024, 147: 110052.

[8] Salehin I, Islam M S, Saha P, et al. AutoML: a systematic review on automated machine learning with neural architecture search[J]. Journal of Information and Intelligence, 2024, 2(1): 52-81.

[9] Pham H, Guan M, Zoph B, et al. Efficient neural architecture search via parameters sharing[C]//International Conference on Machine Learning, 2018: 4095-4104.

[10] Guo Z, Zhang X, Mu H, et al. Single path one-shot neural architecture search with uniform sampling[C]//European Conference on Computer Vision, 2020: 544-560.

[11] Chu X, Zhang B, Xu R. FairNAS: rethinking evaluation fairness of weight sharing neural architecture search[C]//Proceedings of the IEEE International Conference on Computer Vision, 2021: 12239-12248.

[12] Liu H, Simonyan K, Yang Y. DARTS: differentiable architecture search[J]. arXiv preprint arXiv: 1806.09055, 2018.

[13] Xu Y, Xie L, Zhang X, et al. PC-DARTS: partial channel connections for memory-efficient architecture search[J]. arXiv preprint arXiv: 1907.05737, 2019.

[14] Dong X, Yang Y. Searching for a robust neural architecture in four gpu hours[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 1761-1770.

[15] YANG J, LIN Y J, ZHOU P. Differentiable graph neural architecture search based on relational features[J]. Journal of Beijing University of Aeronautics and Astronautics, 2025: 1-17, doi: 10.13700/j.bh.1001-5965-2024-0794.

[16] JIANG P C, XUE Y. Evolutionary neural architecture search method based on ranking score prediction[J]. Chinese Journal of Computers, 2024, 47(11): 2522-2535.

[17] Luo R, Tian F, Qin T, et al. Neural architecture optimization[C]//Advances in Neural Information Processing Systems, 2018: 7816-7827.

[18] Li G, Hoang D, Bhardwaj K, et al. Zero-shot neural architecture search: challenges, solutions, and opportunities[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 7618-7635.

[19] Han X, Xue Y, Wang Z, et al. SaDENAS: a self-adaptive differential evolution algorithm for neural architecture search[J]. Swarm and Evolutionary Computation, 2024, 91: 101736.

[20] Song X, Lv Z, Fan J, et al. Evolutionary multi-objective spiking neural architecture search for image classification[J]. IEEE Transactions on Evolutionary Computation, 2025, doi: 10.1109/TEVC.2025.3528471.

[21] Cai H, Zhu L, Han S. ProxylessNAS: direct neural architecture search on target task and hardware[J]. arXiv preprint arXiv: 1812.00332, 2018.

[22] Tan M, Chen B, Pang R, et al. MnasNet: platform-aware neural architecture search for mobile[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 2820-2828.

[23] Lupión M, Cruz N C, Ortigosa E M, et al. A holistic approach for resource-constrained neural network architecture search[J]. Applied Soft Computing, 2025: 112832, doi: 10.1016/j.asoc.2025.112832.

[24] Cai H, Gan C, Wang T, et al. Once-for-all: train one network and specialize it for efficient deployment[C]//International Conference on Learning Representations, 2020, doi: 10.48550/arXiv.1908.09791.

[25] Xu Z, Wu J. Contrastive meta-reinforcement learning for heterogeneous graph neural architecture search[J]. Expert Systems with Applications, 2025, 260: 125433.

[26] REN P Z, LIANG X D, CHANG X J, et al. Neural architecture search on temporal convolutions for complex action recognition[J]. Journal of Computer Research and Development, 2025, 62(8): 1862-1874.

[27] Salehin I, Islam M S, Saha P, et al. AutoML: a systematic review on automated machine learning with neural architecture search[J]. Journal of Information and Intelligence, 2024, 2(1): 52-81.

[28] Yu K, Sciuto C, Jaggi M, et al. Evaluating the search phase of neural architecture search [J]. arXiv preprint arXiv: 1902.08142, 2019.

[29] Zela A, Siems J, Zimmer L, et al. Understanding and robustifying differentiable architecture search[J]. arXiv preprint arXiv: 1909.09656, 2019.

[30] Ning X, Zheng Y, Zhao T, et al. A generic graph-based neural architecture encoding scheme for predictor-based NAS[C]//European Conference on Computer Vision, 2021: 189-204.

附中文参考文献:

[3] 武家辉,李科研,陈丽新,等. 神经架构搜索技术研究综述[J]. 计算机应用研究,2025,42(1):11-18.

[15] 杨 军,尚颖婕,周 鹏. 基于关系特征的可微图神经架构搜索[J]. 北京航空航天大学学报,2025:1-17,doi:10.13700/j.bh.1001-5965.2024.0794.

[16] 蒋鹏程,薛 羽. 基于排序得分预测的演化神经架构搜索方法[J]. 计算机学报,2024,47(11):2522-2535.

[26] 任鹏真,梁小丹,常晓军,等. 基于时间卷积神经架构搜索的复杂动作识别[J]. 计算机研究与发展,2025,62(8):1862-1874.