

基于排序得分预测的演化神经架构搜索方法

蒋鹏程 薛 羽

(南京信息工程大学软件学院 南京 210044)

摘 要 大量的实际应用场景已经很好地证明了神经网络的优异性能,而神经网络性能的主要决定因素在于其架构。目前,最先进的优秀架构需要人工设计,并且依赖大量的专家经验和反复的试错来验证性能。近年来不断发展的演化神经架构搜索(Evolutionary Neural Architecture Search, ENAS)能够在一定程度上减轻人工设计的负担。然而,即使 ENAS 方法能够自动地搜索到优秀架构,却因为其巨大的时间和计算资源消耗导致难以被广泛使用。代理模型能够较好地解决这一消耗过大的问题,但是现有的代理模型辅助的演化神经架构搜索并不能充分融合搜索和代理的过程,并且目前代理方法难以准确预测精度相近的网络架构的准确排序关系。同时,现有的代理模型普遍需要大量的架构信息作为训练数据才能获得较好的代理精度,这些特点都导致代理模型难以较好地辅助 ENAS,从而制约了 ENAS 的发展。本文中,我们提出了排序得分预测器辅助的演化神经架构搜索方法(Rank Score Predictor-assisted ENAS, RSP-ENAS)。在使用本文提出的面向排序得分预测的新型损失函数的情况下,作为得分预测器的多层感知器(Multi-Layer Perceptron, MLP)给出的种群中个体性能得分的排序与他们实际性能的顺序会尽可能保持一致。在使用本方法搜索的过程中,预测获得的得分可以直接被用于精英选择。在搜索阶段中,本文提出了一种两阶段的搜索方法,在搜索的前期使用小种群关注于代理数据集历史信息的积累,在后期着重使用代理模型预测大种群的适应度值。本文中的实验在 EvoXBench 平台上进行,并且能够在所有的基准数据集上都取得较好的结果,另外我们还在 ImageNet 数据集上进行了验证。和其他方法相比,本文的方法在 NASBench-101 空间上能够搜索到最优的架构,在 NASBench-201 空间的三个数据集上的正确率相较于其他最优方法分别取得了 0.35%、1.12%、0.55% 的进步。在 ImageNet 上使用真实数据集进行的实验中,我们的方法获得了 2.2% 的分类精度的提升。另外,在使用相同数据量的情况下,本文中提出的排序得分预测模型得出的排序结果相较于其他最优方法在 Kendall's Tau 系数上获得了 1.55% 的提升。此外,我们还对代理模型中使用的 One-hot 编码和提出的排序损失进行了验证,从而证明这两项模块对于整体算法的有效性。

关键词 演化计算;神经架构搜索;遗传算法;代理模型;排序预测;得分预测

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2024.02522

Evolutionary Neural Architecture Search with Predictor of Ranking-Based Score

JIANG Peng-Cheng XUE Yu

(School of Software, Nanjing University of Information Science and Technology, Nanjing 210044)

Abstract The exceptional performance of neural networks has been extensively validated across various practical applications, with architecture serving as the primary determinant of their efficacy. Presently, the state-of-the-art architectures necessitate manual design, heavily relying on expert experience and iterative trial-and-error methodologies for performance validation. In recent years, the emergence of Evolutionary Neural Architecture Search (ENAS) has alleviated the burden associated with manual design. However, despite the ability of ENAS methods to autonomously identify superior architectures, their widespread application remains impeded by the substantial time and computational resources required. Surrogate models can mitigate this excessive resource consumption to some extent. However, existing surrogate model-assisted evolutionary neural

收稿日期:2023-12-24;在线发布日期:2024-09-05。本课题得到国家自然科学基金(62376127, 61876089, 61876185, 61902281)资助。

蒋鹏程, 博士研究生, 主要研究方向为演化神经架构搜索、代理模型。E-mail: pc.jiang@qq.com。薛羽(通信作者), 博士, 教授, 中国计算机学会(CCF)会员, 主要研究领域为演化计算、神经架构搜索、人工智能、特征选择。E-mail: xueyu@nuist.edu.cn。

architecture searches fail to fully integrate the search and surrogate processes. Moreover, it is difficult for the current surrogate methods to accurately predict network architectural rankings with similar accuracies. Furthermore, existing surrogate models typically necessitate substantial amounts of architectural information as training data to attain satisfactory surrogate accuracy. These limitations hinder the effective assistance of surrogate models in ENAS, thereby constraining its advancement. In this paper, we propose a Rank Score Predictor-assisted Evolutionary Neural Architecture Search method (RSP-ENAS). By introducing a novel loss function specifically designed for rank score prediction, the Multi-Layer Perceptron (MLP) employed as a score predictor can optimally align the ranking of individual performance scores within the population with their actual performance order. During the search process utilizing this method, the predicted scores are directly applicable for elite selection. We introduce a two-stage search strategy in the search phase, initially focusing on accumulating historical information for the surrogate dataset from evaluating a small population and subsequently emphasizing the use of the surrogate model to predict fitness values for a larger population in the later stages. The experiments conducted in this study were performed on the EvoXBench platform, yielding superior results across all benchmark datasets. Additionally, we validated our approach on the ImageNet dataset. Compared to alternative methodologies, our approach successfully identifies the optimal architecture within the NASBench-101 space. On the three datasets within the NASBench-201 space, accuracy improvements of 0.35%, 1.12%, and 0.55% were achieved relative to other optimal methods. In experiments utilizing real datasets on ImageNet, our method demonstrated a 2.2% enhancement in classification accuracy. Moreover, with the same quantity of data, the ranking results generated by the proposed rank score prediction model exhibited a 1.55% improvement in Kendall's Tau coefficient when compared to other optimal approaches. We further validated the effectiveness of One-hot encoding and the proposed rank loss within the surrogate model, demonstrating the efficacy of these two components for the overall algorithm. This research underscores the potential of advanced surrogate models to enhance the efficiency and accuracy of neural architecture search processes. By reducing computational costs and improving the precision of architecture rankings, our RSP-ENAS method could significantly advance the practical application and accessibility of neural network design, potentially catalyzing more rapid advancements in the fields of machine learning and artificial intelligence. Future work may explore the surrogate models for less training data, which could yield even more substantial enhancements in neural architecture search methodologies.

Keywords evolutionary computation; neural architecture search; genetic algorithms; surrogate models; ranking prediction; score prediction

1 引言

深度学习是目前的一个重要研究热点^[1]. 在这一领域中, 神经网络相关的模型与技术是一个重要的研究分支^[2-4]. 这一技术在目前的很多领域中都展现出了非凡的性能, 包括图像分类^[5]、目标检测^[6]和语义分割^[7]等. 众所周知的是, 神经网络的表现能力很大程度上取决于它们的架构. 因此在不同的数据或者场景下, 神经网络的架构常常需

要被调整才能获得最好的效果^[8]. 然而, 能够适应不同场景的神经网络架构需要丰富的专家经验和对应的专业知识才能被设计出, 这样的条件在大部分情况下是难以被满足的.

神经架构搜索 (Neural Architecture Search, NAS) 是一种利用机器学习技术自动地构建神经网络的方法. 使用这种方法, 用户可以方便、快速地寻找到所需要的最合适的架构^[9]. 在 NAS 的领域中, 搜索过程通常涉及两个层次的操作. 首先, 在宏观层面上, 搜索策略依据候选网络在验证集上的性能表

现来筛选出最具潜力的网络架构. 这一宏观过程的目标是识别出能够在特定任务上达到最优性能的网络架构. 随后, 在微观层面上, 每个选定的网络架构都会经历一个精细化的训练阶段对网络权重进行调整, 目的是挖掘并实现该架构的最大潜能, 确保其在给定任务上达到最佳的性能表现. 简而言之, NAS 的过程可以被视为一个双层优化问题: 第一层是发现最佳的网络架构, 第二层是为这些架构找到最理想的权重配置. 通过这种分层次的搜索和训练, NAS 能够有效地提升神经网络的性能. 这一双层优化过程可以被简化为式(1). 其中, D_{train} 表示微观层面上用于精细化训练网络架构的数据集, D_{val} 表示用于验证网络架构性能的数据集, A 表示候选的神经网络架构, W 表示对应架构中的权重, W^* 表示经过在 D_{train} 上训练得到的神经网络权重. 通过对错误率的最小化优化, NAS 算法能够获得最终所需的神经网络架构.

$$\begin{aligned} & \min_A \text{Error}(A, W^*, D_{\text{val}}) \\ & \text{s. t. } W^* = \arg \min_W \text{Loss}(A, W, D_{\text{train}}) \end{aligned} \quad (1)$$

演化算法是人工智能领域中的一个重要分支, 可以有效地解决各种复杂的优化问题, 尤其是在组合优化^[10]、调度优化^[11]等领域具有明显优势. 演化算法在求解过程中模拟生物界“适者生存”的进化规则来求解问题, 从而能够稳定地获得更优的解. 近年来, 演化算法在求解自动深度学习问题方面取得了重要进展, 这一研究分支常被称为演化深度学习 (Evolutionary Deep Learning, EDL)^[12]. 在这一领域中, 演化算法被使用于数据处理优化^[13]、模型结构和超参数优化^[14]、模型训练优化^[15]、模型评估和集成优化^[16]等.

在 EDL 领域中, 一个重要分支方向是模型结构优化, 即使用演化算法搜索并优化神经网络的架构, 这个方法常被称为演化神经架构搜索 (Evolutionary Neural Architecture Search, ENAS). 这一方法的特点在于, 在宏观层面中, 研究者们采用遗传算法^[17]、粒子群优化算法^[18]、差分进化算法^[19]等基于演化的机器学习算法作为搜索策略, 从而筛选网络架构. 相比于其他种类的神神经架构搜索方法, 演化算法更适合优化神经网络架构这一离散的问题. 例如, 微分神经架构搜索方法使用连续化的方式表示架构, 在搜索阶段会获得大量无参数操作的病态网络架构; 强化学习使用奖励策略寻找最优的神经网络, 但是由于奖励机制的反馈不及时, 搜索效率低下^[8].

ENAS 方法虽然为设计神经网络架构这一任务省下了人力成本, 然而若是想要将这一方法投入实际场景的使用, 其时间消耗和资源消耗的瓶颈仍然存在. 例如, Sun 等人^[20]提出的 AE-CNN 需要 27 GPU 天, 他们提出的另一个搜索方法 CNN-GA 的搜索时间是 35 GPU 天^[21]. Lu 等人^[17]提出的 NAT 使用了大约 59 GPU 天. Chu 等人^[22]提出的 FairNAS 使用了 12 GPU 天. Zhang 等人^[23]提出的 Evo-OSNet 也需要 8.6 GPU 天的搜索时间. 由此可见, 演化神经网络架构搜索仍然需要不小的时间和资源开销. 在应用到实际场景时, 其仍然存在局限性.

众所周知的是, NAS 方法的时间消耗主要来自于神经网络个体中权重的训练, 因此, 缩短或者免除搜索中训练过程的时间消耗是缩短搜索时间的关键. 目前, 用于减少 NAS 算法的时间消耗的方法中, 权重共享与代理模型是两种被主要使用的方法. 权重共享方法的总体思路是维护一个权重池来初始化候选网络, 候选网络可以在未训练时就获得较好的分类精度. 例如, Lu 等人^[24]使用基于权重共享的方法, 利用 OFA 超网中的权重从搜索空间中搜索最优的网络架构, 从而使候选网络的个体的训练过程消耗时间更少. Huang 等人^[25]使用基于权重继承的方法, 使用父代中已经训练好的权重信息初始化部分子代架构的网络权重, 从而缩短子代网络训练的时间并辅助演化搜索的过程. 另一种方法是使用代理方法获得代理值来表示神经网络的性能优劣, 从而免除对候选网络的真实训练. 早期的研究中, 一些研究者通过减少训练数据集数量或者不充分训练的方式获得分类精度, 这种方法在当时被称为低保真度训练, 其本质是使用低保真度的分类精度作为真实的分类精度的代理值参与候选网络之间的比较. 在之后的研究中, 更复杂的机器学习模型被用于这一代理过程. 例如, Sun 等人^[26]使用随机森林进行端到端的性能预测, 从而大大缩短了 AECNN 的搜索时间. 另外, Lu 等人^[17, 24]将多个预测器进行了集成, 从而获得了能力更强的代理模型, 这种方式大大增加了代理模型的拟合能力, 但是在一定程度上增加了代理模型的资源消耗和耗时. 然而, Guo 等人^[27]指出直接预测网络架构绝对性能的方式存在局限性, 这种方法容易在相近真实精度的网络架构之间出现错误的预测排名. Xu 等人^[28]也同样指出直接使用回归预测的方式可能出现小幅度的误差, 从而导致排序出现较大的错误. 与直接预测绝对精

度不同的是, Wang 等人^[29]使用支持向量机作为代理模型辅助粒子群优化算法的架构搜索, 在搜索过程中将已经评估的个体作为经验知识训练二元比较器从而快速比较出两两之间的优劣. 由于使用类似于预测比较关系的间接代理方式必然存在预测错误的排序对(即排序的关系链中存在环路), 因此大多数情况下无法直接获取整体排序, 而需要依赖于计数器. 在代理评估大量个体时, 这一过程由于其接近 $O(n^2)$ 的时间复杂度会导致大量的时间消耗. 在代理评估少量个体时, 由于较少的比较对难以产生足够的计数项, 这种方法容易导致排序的混乱. 因此这种方法在 ENAS 中的代理效果有限.

在本文中, 我们提出了“RSP-ENAS”, 这种方法使用排序得分预测器辅助演化神经网络架构搜索, 以缩短网络评估时间并加快搜索过程. 这种方法一方面可以极大程度上减少传统方法中的评估时间, 另一方面可以获得比现有的其他代理方法更精准的排序预测. 这种利用排序信息训练的得分预测器使用搜索初期的真实评估结果训练一个多层感知器(Multi-Layer Perceptron, MLP), 并且通过设计的新型损失函数使预测值之间的排序与真实性能排序尽可能一致. 在早期搜索的过程中, 种群中只有部分个体被采用从而限制训练过程的时间消耗. 所有的评估结果被记录在档案中作为预测器的训练数据. 在使用代理模型预测得分的阶段, 由于预测个体得分的时间消耗非常小, 因此采用整个种群进行搜索. 在本文中, 我们采用 EvoXBench 框架, 在多个搜索空间和基准数据集上进行了实验, 都获得了优异的结果. 本文的主要贡献如下:

(1) 在 EvoXBench 框架的基础上, 我们对所有非二进制编码区域进行独热编码从而获得每个候选架构的完全二进制编码, 以更好地用于得分预测器.

(2) 我们设计了用于训练 MLP 代理模型的新型损失函数. 这一损失函数可以使 MLP 的预测值排名顺序尽可能与真实的排序一致.

(3) 我们设计了一个两阶段的演化算法框架并集成了上述代理模型, 通过在小种群上真实评估、在大种群上代理评估的方式高效地搜索神经网络架构.

本文第 2 节介绍本文的相关工作; 第 3 节详细介绍所提出的排序得分预测器辅助的演化神经架构搜索算法; 为了验证所提出的得分预测器的有效性和效率, 第 4 节给出实验设计和实验结果; 最后, 第 5 节给出本文的结论和今后的工作.

2 相关工作

在本节中, 我们将首先针对演化深度学习模型优化这一领域展开介绍. 其次, 我们将介绍 EvoXBench^[30] 框架中使用的编码策略, 这是本文中抽样和搜索的基础. 之后, 我们回顾了最先进的神经网络性能预测器, 并加以总结和概述.

2.1 演化深度学习模型优化

在传统深度学习中, 模型的优劣完全依赖于研究人员的经验和繁琐的试验, 这种寻找优秀深度学习模型的过程是十分低效的. 因此, 如何自动搜索得到合适的模型对于解决实际问题十分重要. 以神经网络这一深度学习模型为例, 神经网络的性能主要由其架构决定, 具体为神经网络的超参数和拓扑结构. 在超参数搜索方面, 研究者们采用粒子群优化算法^[31]、遗传算法^[20-21]、分布估计算法^[32]等方法搜索神经网络中卷积核尺寸、通道数量等超参数. 另一方面, 在神经网络拓扑结构搜索方面, 遗传算法^[33]、正则演化算法^[34]等也取得了较大的成功.

2.2 EvoXBench 中使用的搜索空间和编码策略

EvoXBench 框架中包含有 7 个搜索空间, 包括 NASBench-101^[35]、NASBench-201^[36]、NATS^[37]、DARTS^[38]、MoblieNetV3^[39]、ResNet50^[39] 和 Transformer^[40].

在 EvoXBench 框架中, NASBench-101、NASBench-201 和 DARTS 采用微观编码. 微观编码方式使用 Cell 模块作为编码单元, Cell 中的连接方式表示为有向无环图. 具体而言, Ying 等人^[35]使用基于图的表示方法设计了 NASBench-101 的搜索空间, 并训练了所有的候选网络构建了这一基准数据集. NASBench-101 中的网络架构编码长度为 26 位, 其中前 21 位是二进制表示的 7×7 的邻接矩阵上三角区域, 1 表示操作之间有连接, 0 表示没有连接. 这个邻接矩阵的第一列和最后一列表示连接到输入和输出的节点, 中间 5 个节点的操作类型使用编码的后 5 位数表示, 取值范围为 0~2, 分别代表“conv3 \times 3-bn-relu”、“maxpool3 \times 3”和“conv1 \times 1-bn-relu”. Dong 等人^[36]采用了 3 个 Cell Stage 和 2 个残差块的网络架构作为骨干网络. 他们在每个 Stage 中放置 5 个 Cell, 并且对于同一个网络中的所有 Cell 使用相同的拓扑, 这种搜索空间被命名为 NASBench-201. 其中, 每个 Cell 含有 4 个操作并且用边来表示, 每个操作具有 5 种可能的选项. 最后一

种微观搜索空间基准数据集是 Zela 等人^[38]改进后的 DARTS 搜索空间. 与原始的搜索空间相同的是, Zela 等人^[38]使用具有 2 个输入、4 个中间节点和 1 个输出的结构表示整个网络, 所有节点之间的操作边的总数为 8, 与原始的搜索空间不同的是, 每条边上的可选操作不包括“None”操作.

其他的 4 个搜索空间使用宏观编码方式, 这一方式可以直接使用超参数表示网络架构, 例如通道数、扩张率等. 在 NASBench-201 的骨干网络架构基础上, Dong 等人^[37]在每个 Stage 中只使用一个预设的 Cell 拓扑, 并设置所有 Cell 和残差块的通道数从 8 种数字中产生, 这种修改后的通道搜索空间被命名为 NATS. Cai 等人^[39]依据 ResNet 在设计过程中使用的超参数, 设计了 ResNet50 搜索空间, 其中输入图像的尺寸、每个卷积块的深度、扩张率等超参数被使用序号进行表示. Cai 等人^[39]另外针对 MoblieNetV3 的网络架构, 将输入图像的尺寸、卷积核的尺寸、卷积块的深度等超参数使用序号编码. 随着 ViT 在图像领域的巨大成功, Chen 等人^[40]首次将 Transformer 形式的搜索空间构建基准数据集, 其中每个网络的编码包括网络的深度、隐藏层的神经元个数、多头注意力机制中的头的数量等.

2.3 代理辅助的神经网络架构搜索

为了减少神经网络架构搜索过程中的巨大时间消耗, 很多研究者着眼于代理方法的研究. 目前针对 NAS 存在的代理方法各有不同, 本部分将这些方法大致分为 5 类.

2.3.1 训练曲线预测

对于给定的体系结构, 此方法预测它的训练曲线, 以及它的未来表现. 例如, Klein 等人^[41]采用贝叶斯神经网络预测卷积神经网络的训练曲线. Baker 等人^[42]采用了相似的方法, 不同的是他们建立了一个回归模型, 从网络结构、超参数和早期学习曲线得到对应的特征. 这些特征和训练曲线一起被用于预测最终的性能.

2.3.2 正确率/错误率预测

这种方法通过使用回归模型直接拟合历史信息的真实精度, 从而给出架构的分类正确率或者错误率. 例如, Dudziak 等人^[43]使用图卷积网络预测正确率. 巧妙的是, 他们发现直接学习正确率的规律比较困难, 因此他们先根据网络结构学习延迟预测器, 再将延迟预测器迁移到精度预测的任务上. Wen 等人^[44]在使用图卷积网络预测正确率的过程中进一步权衡抽样的个数和保留的个数, 从而进一步提升

效率和有效性. Sun 等人^[26]使用随机森林进行端到端的性能预测, 通过大量决策树的集成从而获得不错的预测正确率.

2.3.3 比较关系预测

对于一对候选架构, 该方法预测两者中哪一个具有更高的性能(即预测成对比较的结果). 例如, Chen 等人^[45]使用对比学习的方法训练出神经架构比较器, 根据比较器的结果进一步探索可能的高精度网络, 从而减小对训练数据的依赖. Wang 等人^[29]先使用支持向量机预测两个网络之间可能存在的比较关系, 之后使用粒子群优化算法探索搜索空间, 并在搜索过程中过滤掉一些架构以减少搜索时间.

2.3.4 大致范围预测

这种方法往往只能给出一个大致的优劣区分, 通过一边筛选网络一边进一步训练预测器得到最终的优秀架构. 例如, Wu 等人^[46]尝试使用随机森林找出大体上好的网络, 通过不断缩小范围从而获得最终搜索到的结构. 然而, 这种方法不能确定搜索到的网络哪个才是最好的架构, 需要对最后一次预测的多个可能的优秀网络进一步评估.

2.3.5 排序预测

这种方法尝试进一步利用排序信息从而能够提升网络架构之间性能排名的准确度. 例如 Zheng 等人^[47]发现, 训练前期就表现出色的模型, 往往在收敛以后会得到一个很好的结果. 他们通过候选网络架构的初期表现得到相对的性能排序, 从而加速搜索过程. Xu 等人^[28]提出了 RankLoss 添加到预测中作为辅助手段, 然而这种方法极大地依赖数据量. Gao 等人^[27]在此基础上添加了等级预测, 从而解决数据量不足的情况下对优势网络预测不准确的问题.

3 方 法

在本文中, 我们提出了一种使用得分预测器的演化神经架构搜索方法(RSP-ENAS), 从而减少评估过程的时间. 首先, 我们使用 EvoXBench 中的编码方案表示网络架构, 在训练预测器的过程中使用 One-hot 编码将所有非二进制编码区域转换为二进制编码. 其次, 在代理模型部分, 我们设计了特殊的损失函数用于在训练的过程中进一步关注预测值的排序与真实标签的排序一致性, 并使用一个小型的 MLP 作为得分预测器模型. 最后, 我们设计了一个使用代理模型辅助的遗传算法搜索框架, 在搜索过程中, 代理模型分为训练模式和预测模式. 在训练模

式中,种群中的部分个体被正常评估,在预测模式中,种群中所有的个体使用代理模型预测的得分作为适应度值。

3.1 编码

搜索阶段的网络架构编码采用第 2.1 节中介绍的 EvoXBench 框架里的编码方案,包括 7 种搜索空间,即 NASBench-101、NASBench-201、NATS、DARTS、MoblieNetV3、ResNet50 和 Transformer。考虑到在训练代理模型时,编码值之间的差距尺度会导致代理模型难以拟合训练数据,因此,在代理模型阶段,编码中的所有非二进制编码区域都会被转换为 One-hot 形式的二进制编码。举例来说,NASBench-101 空间将每个 Cell 表示为 26 位编码,其中

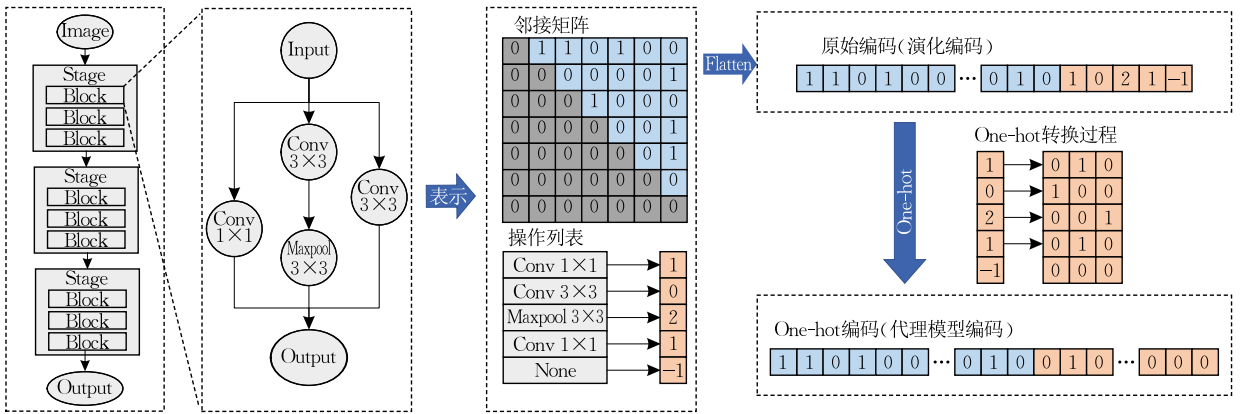


图 1 NASBench-101 搜索空间编码样例

表 1 EvoXBench 中搜索空间的概述

搜索空间	类型	数据集	空间大小	编码维度	
				原始	One-hot
NASBench-101	micro	CIFAR-10	423 K	26	36
NASBench-201	micro	CIFAR-100 Imagenet-16-120	15.6 K	6	30
NATS	macro	CIFAR-10 Imagenet-16-120	32.8 K	5	40
DARTS	micro	CIFAR-10	$\approx 10^{21}$	32	168
ResNet50	macro	ImageNet-1K	$\approx 10^{14}$	25	104
Transformer	macro	ImageNet-1K	$\approx 10^{14}$	34	86
MoblieNetV3	macro	ImageNet-1K	$\approx 10^{20}$	21	205

3.2 基于排序信息训练的得分预测器

本文使用了一种新的损失函数训练基于神经网络的代理模型,从而进一步关注训练数据的排序信息。在代理模型的训练过程中,面临的挑战之一是难以获取大量的架构样本作为训练数据。这一限制导致参数量大的代理模型易产生过拟合现象。鉴于此,为了确保代理模型既具有评估实时性又具有高效性,我们选择了小型 MLP 充当得分预测器。MLP

前 21 位表示邻接矩阵的上三角区域,取值范围是 $\{0,1\}$,后 5 位是按照序号表示的 3 种操作类型,取值范围是 $\{0,1,2\}$ 。如图 1 所示,这个搜索空间中的编码经过转换后具有 36 位,其中后 15 位是原先长度为 5 的类型编码经过 One-hot 转换后的二进制表示。图中的原始编码将被后续用于演化过程,而 One-hot 编码将被用于代理模型的训练和预测。在 EvoXBench 中,所有的搜索空间的编码长度被统计在表 1 中。其中,表中的“数据集”一列表示这个搜索空间中的网络是在对应的图像数据集下设计并训练的,“空间大小”一列表示搜索空间的大小,即搜索空间中的候选架构数量。编码的原始长度和 One-hot 格式的长度也被展示在了“编码维度”列中。

中含有的隐藏层数量和每个隐藏层含有的神经元数量随着使用的搜索空间中表示的编码长度动态调整。在前两个隐藏层中,神经元数量增长为输入数据维度的两倍。在最后的隐藏层中神经元数量等于输入数据的维度,最后一层的输出为一个预测值。所有用于训练代理模型的数据需要经过第 3.1 节中介绍的 One-hot 处理。之后,我们对所有数据都进行了标准化处理从而使这一数据具有搜索空间中的唯一表示(此操作仅仅对于 NASBench-101 空间生效)。具体方法为,我们先构建邻接矩阵并分析无效节点,然后清除所有无效节点的连接边,并将其序号放在所有有效节点的序号后面,同时修改代表操作种类的 One-hot 向量为零向量,在编码中的表现为全 0 行和全 0 列只会出现在非全 0 行和非全 0 列之后。

在训练的过程中,训练数据被按照批量大小读取,并得到由预测值组成的向量。通常情况下,损失函数是预测值和对应的正确值计算得到的均方根误差(MSE)。然而同样的均方根误差下,可能会存在不同的预测结果,但是这些结果从解决问题的角度

来看达到的效果完全不同. 例如, 假设有 3 个数据, 真实的正确率是 $[0.92, 0.91, 0.90]$, 一种预测值是 $[0.93, 0.89, 0.87]$, 另一种预测值是 $[0.89, 0.93, 0.91]$, 不难发现两种预测结果的 MSE 损失都是 0.0005, 然而前者预测的排序完全正确, 而后者存在排序错误的问题. 因此, 我们认为仅仅是拟合真实标签难以应对性能排序问题, 并且, 相比于直接预测精度, 预测得到正确的排序更重要.

本文提出了一种能够应对排序问题的损失函数. 假设一个批量大小为 n 的数据被表示为 $\{(\mathbf{X}_i, y_i)\}_{n_i}$, 其中 \mathbf{X}_i 是经过评估并记录的个体编码向量, y_i 是对应的真实正确率. 给定函数过程 H 表示 3.1 节中的 One-hot 编码转换, F 表示使用 MLP 的前向传播过程, 并引入一个新的单调递减的连续函数 $\phi(x)$ 用于建立原本的矢量距离与训练代理模型的损失值之间的负相关性, 并且我们选用非线性激活函数用于抽取训练过程中的非线性特征, 在本文中, 我们使用了 $\phi(x) = \log(1 + e^{-x})$. 我们提出的排序损失函数由两部分组成, 其中第一部分如式 (2) 所示. 其中, 我们使用 $\hat{y}_i = F(H(\mathbf{X}_i))$ 表示代理模型对第 i 个样本的预测得分以简化公式表示, 并引入 sign 信号函数用于判断其输入参数是否为正值.

$$L_1(\hat{Y}, Y) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \phi((\hat{y}_i - \hat{y}_j) \times \text{sign}(y_i - y_j)) \quad (2)$$

进一步地, 对于式 (2), 假设有 3 个个体, X_1 和 X_2 都优于 X_3 , 经过对 $L_1(\hat{Y}, Y)$ 的最小优化, X_1 和 X_2 产生的更新方向都是使预测值变小, 同时使 X_3 的预测值变大. 然而, 这并不能体现出这一步的更新相对距离. 为了解决这一问题, 我们提出的损失函数包含的另一部分为距离损失. 假设用 \hat{d}_{ij} 表示 X_i 和 X_j 之间的预测差值, 即 $\hat{d}_{ij} = |\hat{Y}_i - \hat{Y}_j|$. 用 d_{ij} 表示 Y_i 和 Y_j 之间的真实差值, 即 $d_{ij} = |Y_i - Y_j|$. 距离损失被表示为式 (3), 其中 \hat{D}_i 和 D_i 分别表示两两预测值之差组成的向量中的第 i 个元素.

$$L_2(\hat{Y}, Y) = \sum_{i=1}^{M-1} \sum_{j=i+1}^M \phi((\hat{D}_i - \hat{D}_j) \times \text{sign}(D_i - D_j)) \quad (3)$$

最终, 在训练过程中, 我们提出的排序损失函数为式 (4). 其中, α 是超参数, 用于控制两种损失函数的重要性, Sigmoid 函数用于控制系数的范围为正值.

$$L(\hat{Y}, Y) = L_1(\hat{Y}, Y) + \text{Sigmoid}(\alpha) \times L_2(\hat{Y}, Y) \quad (4)$$

3.3 排序得分预测器辅助的演化搜索框架

在 3.1 节和 3.2 节的基础上, 我们设计了一种基于遗传算法的搜索框架用于从搜索空间中寻找最优的网络架构, 并在整个演化的过程中使用两阶段

的搜索方式进行高效搜索. 我们已经在 2.1 节中介绍了不同搜索空间上的编码方式. 基于这一编码方式, 整个搜索框架分为 2 个层级, 如图 2 所示.

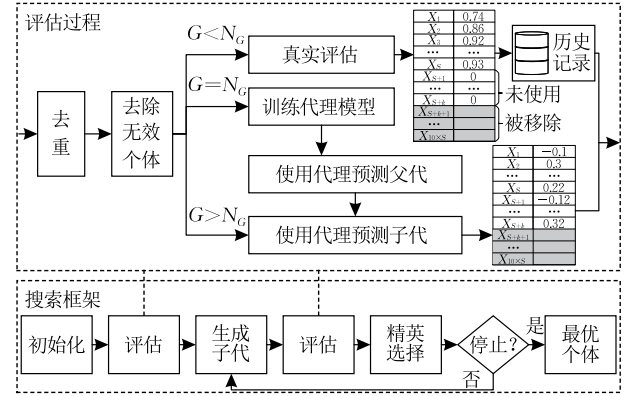


图 2 RSP-ENAS 的总体框架

第一个层级是宏观框架. 具体来说, 初始种群被随机生成, 经过评估后, 新个体通过父代交叉和变异操作被不断产生, 新个体被使用相同的评估方式获得适应度值. 我们选择了更适用于整数编码的均匀交叉算子^[17], 对于两个父代的个体, 我们生成一个与编码长度相同的 $flag_1$ 数组, 根据我们设定的交叉阈值, $flag_1$ 中大于阈值对应位置的父代基因位进行交换. 对于变异操作, 受 AmoebaNet 启发^[48], 我们选用了类似的随机均匀选择变异算子, 对于一个个体, 我们生成一个与编码长度相同的 $flag_2$ 数组, 根据我们设定的变异阈值, $flag_2$ 中大于阈值对应位置的基因位随机变换为其他在上界和下界之内的有效值. 新的子代被评估后, 和父代种群一起经过精英选择. 在这一过程中我们只保留下父代和子代种群中较好的部分个体. 在整个过程中, 早期由于个体需要被逐个查询真实的正确率, 在实际场景中难以做到这一点, 因此我们只使用种群中的一部分个体, 从而实现小种群搜索的过程, 未被使用的个体被排除在进化的过程外. 另外, 我们需要保证相同的一代中的重复个体要被跳过真实评估的过程, 并选择新的未被评估过的个体参与真实评估, 以确保能够获得尽可能多的新架构数据, 从而在之后更好地训练代理模型. 我们设定了一个阈值 N_G , 作为评估部分种群的过程完成的信号. 到达这一阈值后, 我们使用架构历史记录训练代理模型. 在之后的搜索过程中, 我们直接使用训练好的 MLP 预测每一个架构个体的得分. 由于预测的时间消耗较小, 因此整个大种群都被使用来获得适应度值. 第一层级整体架构搜索算法框架的伪代码如算法 1 所示.

算法 1. 架构搜索算法框架.

输入：每代种群中真实评估的架构数量 S ；搜索迭代次数 T ；所有迭代中真实评估的次数 N_G

输出：最优个体及其正确率

初始化 $10S$ 数量的个体作为初始种群 P ；

初始化一个 MLP 作为预测器 M ；

$G \leftarrow 0$ ； $History \leftarrow \{\}$ ；

$Evaluate(G, N_G, None, P, History, M)$ ；

WHILE $G < T$

 使用交叉、变异算子产生 $10S$ 数量的子代 Q ；

$Evaluate(G, N_G, P, Q, History, M)$ ；

 基于 $P.fitness$ 和 $Q.fitness$ 的结果进行精英选择，形成新的种群 P ；

$G \leftarrow G + 1$ ；

END WHILE

真实评估最优个体；

第二个层级是评估部分，如算法 2 所示。首先，有一些个体不满足编码规则因此被丢弃，并且重复的个体也会被去除，去除的个体在图 2 中标记为灰色区域，剩下的个体我们称其为有效个体。之后的过程包含三个分支，使用阈值 N_G 区分。在收集架构信息的阶段，即当前代数 $G < N_G$ 的真实评估的过程中，种群中的部分有效个体被赋予真实的正确率，并且这些结果被记录到历史中，其他未被使用或者不符合编码规则的个体被赋予正确率 0%，从而仅仅使用小种群参与演化，使其他无关个体被排除在演化过程外。在预测阶段的第一代，即代数 $G = N_G$ ，我们使用架构历史信息训练 MLP。从此之后，我们只需要使用预测器直接预测大种群中所有有效个体的评分，而不需要再获取真实的正确率。需要注意的是，在预测阶段的第一代，即代数 $G = N_G$ 时，预测器的预测范围和真实的正确率范围不在同一个数量范围，并且含义也完全不同，因此父代个体需要被使用预测器重新评估一次，从而能够和子代一起进行正确的精英选择过程。

算法 2. 评估过程 $Evaluate$.

输入：当前的迭代轮数 G ；父种群 P ；子种群 Q ；适应度值历史 $History$ ；得分预测器 M ；所有迭代中真实评估的次数 N_G

输出： $History$

标准化处理子种群 Q 中的个体编码；

去除子种群 Q 中的重复和无效个体；

IF $G < N_G$

 对 Q 中序号索引在前 S 个的个体真实评估，并记录在 $Q.fitness$ 中；

 设置 Q 中其他未被评估的个体的适应度值为 0，并记录在 $Q.fitness$ 中；

 在 $History$ 中添加 Q 中的前 S 个个体；

ELSE IF $G = N_G$

$Encoding \leftarrow$ 将 $History$ 中的每个个体转换为 One-hot 编码；

 使用 $Encoding$ 训练预测器 M ；

 将 P 和 Q 分别转换为 One-hot 编码，记为 P' 和 Q' ；

$Q.fitness \leftarrow M(Q')$ ；

$P.fitness \leftarrow M(P')$ ；

ELSE

 将 Q 转换为 One-hot 编码，记为 Q' ；

$Q.fitness \leftarrow M(Q')$ ；

END IF

4 实验

为了验证所提出的性能预测器的有效性和其在提升架构搜索效率方面的作用，我们精心设计并进行了一系列的实验。本节将详细介绍使用本框架进行的实验结果和分析，并且提供了和其他方法的对比。另外，我们还设计了相关的消融实验来证明本方法中使用的各个模块的有效性。

4.1 超参数设置和数据集

在本节中，我们将提供实验结果对应的所有超参数设置。在遗传算法中，种群大小被设置为 400，在需要真实评估的过程中，每次只进行 40 个个体的评估，其他 360 个个体直接被赋予正确率 0%。搜索的总代数为 50，其中前 10 代采用真实评估，后 40 代采用代理模型预测得分。在训练得分预测器的阶段，学习率被设置为 5×10^{-4} ，训练的批大小为 200 个架构样本。EvoXBench 中的 7 个搜索空间都被采用，其中 NASBench-201 和 NATS 还包含 3 个图像数据集上的基准数据，因此共有 11 组实验被完成。

4.2 在 EvoXBench 上的实验结果

我们在 EvoXBench 中所有的数据集上完成了共计 11 组实验，每次实验包含 5 次重复运行的结果。表 2 中展示的是在这些实验中取得的效果。其中，全局最优的列包含测试集上表现最好的个体对应的验证正确率和测试正确率。前 2 列表示的是 RSP-ENAS 在验证集上能够找到的最好的架构的正确率。我们给出了 5 次运行在验证集上的均值、标准差和最小值。根据每次在验证集上搜索的结果，我们获取了最优的架构的编码并在测试集上评估，测试集上的均值、标准差和最小值也同样在表中展示，记录在第 4 和 5 列。需要强调的是，表中验证集下的全局最优表示在测试集上的全局最优的个体在验证集上的表现，而非真正在验证集上最优的个体，因此我们使用“*”标记。另外，DARTS、MN3、ResNet50、Transformer 这 4 个搜索空间由于包含的候选架构过多，因此无法提供确切的最优测试精

度, EvoXBench 框架中提供了一个估计值, 我们在表 2 中使用“+”标记. 我们可以发现, 在一些搜索空间中, RSP-ENAS 能够寻找到测试集最优. 甚至在 Transformer 这一搜索空间中, RSP-ENAS 可以搜索到比框架估计的最高精度值还要高的网络架构. 在其他的数据集中, RSP-ENAS 寻找到的最小值也接近测试集最优. 但是, 由于基准数据集中存在测试集最优解的个体在验证集表现较差的情况, 因此我们的方法在一些空间上没有找到测试集上的最优架

构. 例如在 NASBench-101 上, RSP-ENAS 被误导找到了验证集上的最优解, 同时这个解在测试集上是排名第二. 另外在时间开销上, 每次实验中, 本方法一共使用了 16 400 个架构, 其中 400 个架构是真实评估的, 16 000 个架构是使用预测器直接预测得分的, 最后一代个体的预测得分值最高的 10 个个体也被再次真实评估. 由于预测得分阶段的时间开销几乎可以忽略不计, 因此相当于节省了 97.56% 的架构的评估时间.

表 2 EvoXBench 上所有搜索空间和基准数据集的正确率的实验结果

(单位: %)

搜索空间	验证集正确率			测试集正确率		
	均值±标准差	搜索最优	全局最优	均值±标准差	搜索最优	全局最优
NASBench-101	94.78±0.14	95.06	94.50*	94.10±0.12	94.23	94.32
NASBench-201(CIFAR-10)	91.57±0.09	91.68	91.61*	94.37±0.00	94.37	94.37
NASBench-201(CIFAR-100)	73.60±0.11	73.72	73.49*	73.35±0.23	73.51	73.51
NASBench-201(ImageNet-16-120)	46.42±0.32	46.87	46.77*	46.52±0.32	47.08	47.31
NATS(CIFAR-10)	90.45±0.07	90.50	90.71*	93.33±0.18	93.65	93.65
NATS(CIFAR-100)	70.43±0.21	70.58	70.92*	70.12±0.34	70.72	71.34
NATS(ImageNet-16-120)	45.35±0.68	46.27	46.73*	46.53±0.44	47.17	47.40
DARTS	94.90±0.08	95.01	/	94.36±0.18	94.47	94.73+
MNV3	83.38±0.07	83.45	/	79.14±0.10	79.22	80.13+
ResNet50	83.95±0.05	84.00	/	80.18±0.04	80.25	82.35+
Transformer	82.34±0.01	82.36	/	82.31±0.01	82.33	79.08+

4.3 与其他算法的对比

我们在 NASBench-201 搜索空间上进行了实验对比. 使用此搜索空间上的实验结果与其他方法进行比较的结果如表 3 所示. 在这一搜索空间中, 所有的网络架构在 CIFAR-10、CIFAR-100 和 ImageNet-16-120 上进行训练. 可以发现, 在这一搜索空间中,

RSP-ENAS 能够在三个基准数据集上进行搜索, 并在验证集和测试集上找到更优的体系结构. 相较于其他最新的方法, 我们的方法在 NASBench-201 空间的三个数据集上的正确率相较于其他最优方法分别取得了 0.35%、1.12%、0.55% 的提升.

表 3 NASBench-201 搜索空间下的搜索结果准确率对比

(单位: %)

对比方法	CIFAR-10		CIFAR-100		ImageNet-16-120	
	验证集	测试集	验证集	测试集	验证集	测试集
RANDOM ^[49]	91.07±0.26	93.86±0.23	71.46±0.97	71.55±0.97	45.03±0.91	45.28±0.97
RSPS ^[50]	87.60±0.61	91.05±0.66	68.27±0.72	68.26±0.96	39.73±0.34	40.69±0.36
SETN ^[51]	90.00±0.97	92.72±0.73	69.19±1.42	69.36±1.72	39.77±0.33	39.51±0.33
ENAS ^[52]	90.20±0.00	93.76±0.00	70.21±0.71	70.67±0.62	40.78±0.00	41.44±0.00
FairNAS ^[22]	90.07±0.57	93.23±0.18	70.94±0.94	71.00±1.46	41.90±1.00	42.19±0.31
ReNAS ^[28]	90.90±0.31	93.99±0.25	71.96±0.99	72.12±0.79	45.85±0.47	45.97±0.49
DARTS ^[53]	91.03±0.44	93.80±0.40	71.36±1.51	71.53±1.51	44.87±1.46	45.12±0.82
BOHB ^[54]	91.17±0.27	93.94±0.28	72.04±0.93	72.00±0.86	45.55±0.79	45.70±0.86
REA ^[48]	91.25±0.31	94.02±0.31	72.28±0.95	72.23±0.84	45.71±0.77	45.77±0.80
REINFORCE ^[55]	91.12±0.25	93.90±0.26	71.80±0.94	71.86±0.89	45.37±0.74	45.64±0.78
GDAS ^[56]	89.68±0.72	93.23±0.58	68.35±2.71	68.17±2.50	39.55±0.00	39.40±0.00
RSP-ENAS	91.57±0.09	94.37±0.00	73.6±0.11	73.35±0.23	46.42±0.32	46.52±0.32

为了说明本方法在真实网络架构搜索下的效果, 本方法使用 EvoXBench 中 MNV3 数据集使用的编码方案, 结合 Once-For-All 超网, 重复了上述实验, 在搜索过程中我们只使用 ImageNet 训练集中抽样的 10 000 个数据作为验证集以确保实验的严谨性和公

平性. 在搜索完成后, 我们选择得分最好的个体, 构建真实的网络架构并使用完整的 ImageNet 训练集训练 450 个 epoch, 并最后计算测试精度. 表 4 中展示的本实验的结果和其他现有 NAS 方法的对比, 可以看到的是我们搜索到的网络在分类精度上达到了最优.

表 4 使用 Once-For-All 的搜索结果准确率对比 (单位: %)

对比方法	Top-1	Top-5	参数量/M	FLOPs/M
DARTS ^[53]	73.3	91.3	4.7	574
ProxylessNAS ^[57]	75.1	92.5	7.1	465
PC-DARTS ^[58]	75.8	92.7	5.3	597
EnTranNAS ^[59]	75.7	92.8	5.5	637
SNAS ^[60]	72.7	90.8	4.3	533
RelativeNAS ^[61]	75.12	92.3	5.1	563
SLE-NAS ^[25]	75.7	92.5	4.5	412
CDARTS ^[62]	76.3	92.9	6.1	701
RSP-ENAS	78.5	93.9	6.1	503

4.4 消融实验

为了进一步揭示 RSP-ENAS 中使用的得分预测器的性能,我们使用随机抽样的方式从基准数据集中获取了 424 个架构样本数据用于训练预测器,并再次抽取了 5000 个新的架构样本用于验证. 验证的过程中,我们使用数据集上的测试数据和得分预

测器给出的分数计算出 Kendall's Tau 相关性系数 (KTau),表 5 中展示的相关数据是在 NASBench-101 搜索空间中进行 5 次实验给出的结果. 另外,我们也提供了在 7 个数据集上的实验结果绘制的真实和预测之间的关系图. 在图 3 中我们可以看到,得分预测器能够在这 7 个数据集上都获得非常好的预测准确性. 值得一提的是,我们可以发现在 NATS 上的点非常接近红线表示的准确值,并且都获得了 0.85 以上的 KTau. 经过我们的验证,直接使用 NATS 空间在三个数据集上训练后的代理模型在验证集精度和测试集精度的预测评分与真实标签的 KTau 分别为 0.8349、0.9009 和 0.8757. 这说明使用我们提出的得分预测器已经能够较好地代替原本的训练过程.

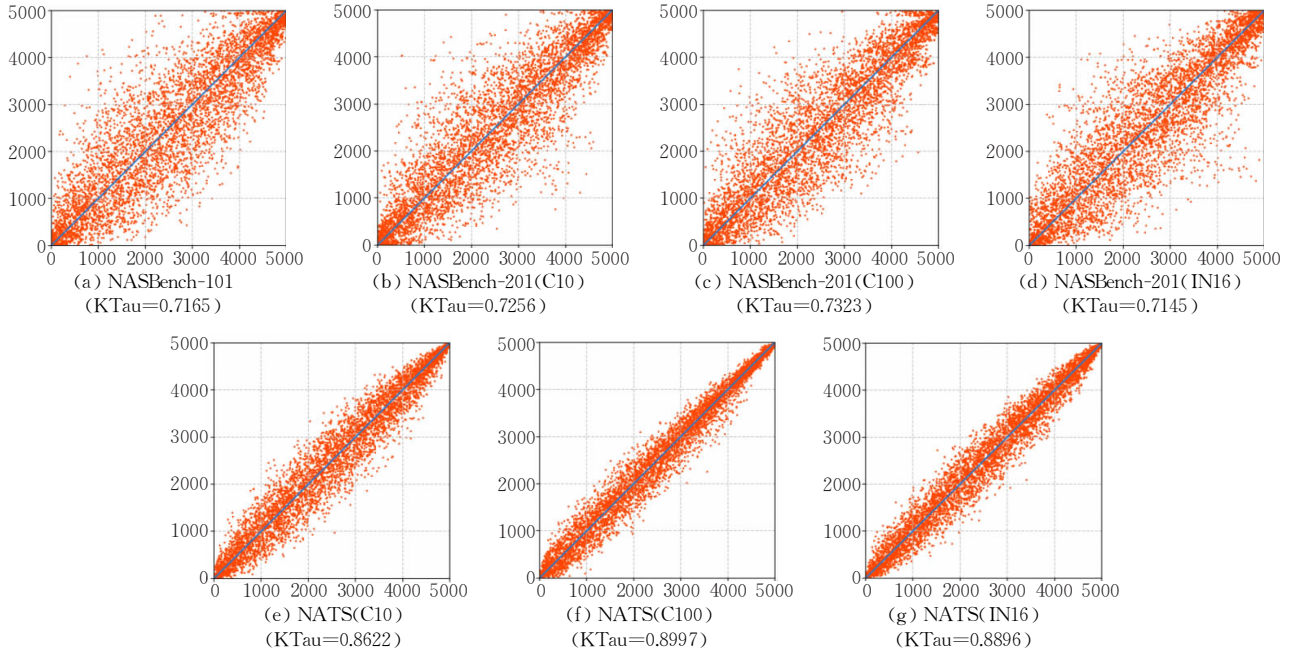


图 3 在 3 种搜索空间共计 7 个基准数据集上预测得分和测试集标签的关系图

表 5 NASBench-101 搜索空间下预测得分与测试集标签的 KTau 系数对比

对比方法	KTau	样本量/%
E2EPP ^[26]	0.5038	≈0.1
E2EPP ^[26]	0.5705 ± 0.0082	≈0.25
Peephole ^[63]	0.4556	≈0.1
Peephole ^[63]	0.4373 ± 0.0112	≈0.25
NPNAS ^[44]	0.6945	≈0.1
HAAP ^[64]	0.7010 ± 0.0022	≈0.1
HAAP ^[64]	0.7126 ± 0.0024	≈0.25
SSANA ^[65]	0.6541 ± 0.0078	≈0.25
ReNAS ^[28]	0.6574	≈0.1
NAO ^[66]	0.655	≈0.1
RSP-ENAS	0.7165	≈0.1

针对编码过程中的 One-hot 操作,我们进行了消融实验证明这一操作的必要性. 表 6 中展示了我们在使用原始编码和使用 One-hot 编码对于代理模型预测能力的影响,我们从 NASBench-101 抽取了 0.05%、0.10% 和 1% 的数据进行了实验,可以发

表 6 NASBench-101 搜索空间下使用不同的编码方式时预测得分和验证集标签、测试集标签的 KTau 系数

编码方法	0.05%		0.10%		1%	
	验证集	测试集	验证集	测试集	验证集	测试集
原始编码	0.5919	0.6001	0.6811	0.6898	0.8130	0.8079
One-hot 编码	0.6128	0.6120	0.6945	0.7165	0.8134	0.8074

现的是,采用 One-hot 编码的情况下,代理模型更容易地拟合并预测,从而能够获得更高的 K τ 结果.

最后,我们进行了消融分析以证明我们提出的方法相较于其他损失函数的优势,具体包括 MSE 损失函数和 MRL (Margin Ranking Loss) 损失函数^[67]. 另外我们还分析了本文提出的损失函数中 L_2 带来的影响. 为了同时分析本代理模型对于训练数据数量的敏感性,我们还针对 NASBench-101 抽取了 0.05%、0.10%、1% 和 10% 的架构样本进行实验.

表 7 NASBench-101 搜索空间下使用不同的架构样本数量的预测得分和验证集标签、测试集标签的 K τ 系数

损失函数对比	0.05%		0.10%		1%		10%	
	验证集	测试集	验证集	测试集	验证集	测试集	验证集	测试集
MSE	0.3098	0.3115	0.5121	0.5129	0.6296	0.6296	0.6594	0.6605
MRL	0.5654	0.5669	0.6555	0.6535	0.8098	0.8016	0.8319	0.8266
L_1	0.6078	0.6074	0.6736	0.6716	0.8121	0.8070	0.8330	0.8258
$L_1 + \alpha L_2$	0.6128	0.6120	0.6945	0.7165	0.8134	0.8074	0.8351	0.8308

5 结 论

本文的目标是开发一种高效神经网络得分预测器辅助的演化神经网络架构搜索方法. 该目标通过两个核心部分得以实现. 具体来说,我们提出了一种使用遗传算法在搜索空间中寻找最优网络并使用得分预测器辅助搜索,该遗传算法框架能够完全契合精英保留策略以便迅速地从大量个体中获得优势个体. 另外,对于得分预测器的设计,我们提出了由两个子损失构成的排序损失函数辅助训练过程,从而将注意力重点关注在排序的准确性上. 我们使用 EvoXBench 框架设计了充分的实验. 根据文中公布的结果可以发现,通过使用我们提出的得分预测器, RSP-ENAS 能够具有良好的搜索能力,并且没有产生较大的时间花费,与其他方法相比具有较大的优势. 通过实验证明,本文提出的得分预测器模型具有非常稳定的预测能力. 我们未来的工作重点是使用更少的数据量的情况下获得较好的得分预测器,从而进一步提升演化神经网络架构搜索的效率.

参 考 文 献

- [1] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, 521(7553): 436-444
- [2] Jiao Li-Cheng, Yang Shu-Yuan, Liu Fang, et al. Seventy years beyond neural networks: Retrospect and prospect. *Chinese Journal of Computers*, 2016, 39(8): 1697-1716(in

我们将 3 种损失函数在 4 种架构数据量下的预测结果与验证集、测试集的标签计算了 K τ 系数,并展示在了表 7 中. 可以发现的是,我们提出的损失函数能够在较少样本的情况下就产生较好的结果,并且在数据量增加后仍保持优势. 另外可以证明的是,我们在损失函数中使用的距离损失对预测的准确度具有一定的作用. 虽然使用 1% 的样本量能够获得非常高的 K τ 相关性,但是如此众多的样本数量会导致过长的真实评估的时间,因此出于折衷考虑,我们采用了 0.1% 的样本量.

Chinese)

(焦李成, 杨淑媛, 刘芳等. 神经网络七十年: 回顾与展望. *计算机学报*, 2016, 39(8): 1697-1716)

- [3] Zhou Fei-Yan, Jin Lin-Peng, Dong Jun. Review of convolutional neural network. *Chinese Journal of Computers*, 2017, 40(6): 1229-1251(in Chinese)
(周飞燕, 金林鹏, 董军. 卷积神经网络研究综述. *计算机学报*, 2017, 40(6): 1229-1251)
- [4] Dou Hui, Zhang Ling-Ming, Han Feng, et al. Survey on convolutional neural network interpretability. *Journal of Software*, 2024, 35(1): 159-184(in Chinese)
(窦慧, 张凌茗, 韩峰等. 卷积神经网络的可解释性研究综述. *软件学报*, 2024, 35(1): 159-184)
- [5] Li Xiao-Xu, Liu Zhong-Yuan, Wu Ji-Jie, et al. Total relation network with attention for few-shot image classification. *Chinese Journal of Computers*, 2023, 46(2): 371-384 (in Chinese)
(李晓旭, 刘忠源, 武继杰等. 小样本图像分类的注意力全关系网络. *计算机学报*, 2023, 46(2): 371-384)
- [6] Yang Xue, Yan Jun-Chi. Visual oriented object detection via feature alignment and Gaussian parameterization. *SCIENTIA SINICA Informationis*, 2023, 53(11): 2250-2265(in Chinese)
(杨学, 严骏驰. 基于特征对齐和高斯表征的视觉有向目标检测. *中国科学: 信息科学*, 2023, 53(11): 2250-2265)
- [7] Tian Xuan, Wang Liang, Ding Qi. Review of image semantic segmentation based on deep learning. *Journal of Software*, 2019, 30(2): 440-468(in Chinese)
(田萱, 王亮, 丁琪. 基于深度学习的图像语义分割方法综述. *软件学报*, 2019, 30(2): 440-468)
- [8] Liu Y, Sun Y, Xue B, et al. A survey on evolutionary neural architecture search. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(2): 550-570

- [9] Meng Zi-Yao, Gu Xue, Liang Yan-Chun, et al. Deep neural architecture search: A survey. *Journal of Computer Research and Development*, 2021, 58(1): 22-33(in Chinese)
(孟子尧, 谷雪, 梁艳春等. 深度神经架构搜索综述. *计算机研究与发展*, 2021, 58(1): 22-33)
- [10] Zhan Z-H, Shi L, Tan K C, Zhang J. A survey on evolutionary computation for complex continuous optimization. *Artificial Intelligence Review*, 2022, 55(1): 59-110
- [11] Liu S-C, Chen Z-G, Zhan Z-H, et al. Many-objective job-shop scheduling: A multiple populations for multiple objectives-based genetic algorithm approach. *IEEE Transactions on Cybernetics*, 2023, 53(3): 1460-1474
- [12] Zhan Z-H, Li J-Y, Zhang J. Evolutionary deep learning: A survey. *Neurocomputing*, 2022, 483: 42-58
- [13] Zhu Y, Yan Y, Zhang Y, Zhang Y. EHSO: Evolutionary hybrid sampling in overlapping scenarios for imbalanced learning. *Neurocomputing*, 2020, 417: 333-346
- [14] Wang Y-Q, Chen C-H, Zhang J, Zhan Z-H. Dropout topology-assisted bidirectional learning particle swarm optimization for neural architecture search//*Proceedings of the Genetic and Evolutionary Computation Conference Companion*. Boston, USA, 2022: 93-96
- [15] Li J-Y, Du K-J, Zhan Z-H, et al. Multi-criteria differential evolution//*Proceedings of the Genetic and Evolutionary Computation Conference Companion*. Lille, France, 2021: 183-184
- [16] Gong Y-J, Zhang J, Zhou Y. Learning multimodal parameters: A bare-bones niching differential evolution approach. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, 29(7): 2944-2959
- [17] Lu Z, Sreekumar G, Goodman E, et al. Neural architecture transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, 43(9): 2971-2989
- [18] Huang J, Xue B, Sun Y, et al. Particle swarm optimization for compact neural architecture search for image classification. *IEEE Transactions on Evolutionary Computation*, 2023, 27(5): 1298-1312
- [19] Huang J, Xue B, Sun Y, Zhang M. EDE-NAS: An eclectic differential evolution approach to single-path neural architecture search//*Proceedings of the Australasian Joint Conference on Artificial Intelligence*. Perth, Australia, 2022, 13728: 116-130
- [20] Sun Y, Xue B, Zhang M, Yen G G. Completely automated CNN architecture design based on blocks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 31(4): 1242-1254
- [21] Sun Y, Xue B, Zhang M, et al. Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*, 2020, 50(9): 3840-3854
- [22] Chu X, Zhang B, Xu R. FairNAS: Rethinking evaluation fairness of weight sharing neural architecture search//*Proceedings of the IEEE/CVF International Conference on computer vision*. Montreal, Canada, 2021: 12219-12228
- [23] Zhang H, Jin Y, Hao K. Evolutionary search for complete neural network architectures with partial weight sharing. *IEEE Transactions on Evolutionary Computation*, 2022, 26(5): 1072-1086
- [24] Lu Z, Deb K, Goodman E, et al. NSGANetV2: Evolutionary multi-objective surrogate-assisted neural architecture search//*Proceedings of the European Conference on Computer Vision*. Glasgow, UK, 2020: 35-51
- [25] Huang J, Xue B, Sun Y, et al. Split-level evolutionary neural architecture search with elite weight inheritance. *IEEE Transactions on Neural Networks and Learning Systems*, 2023: 1-15
- [26] Sun Y, Wang H, Xue B, et al. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 2020, 24(2): 350-364
- [27] Guo B, Chen T, He S, et al. Generalized global ranking-aware neural architecture ranker for efficient image classifier search//*Proceedings of the 30th ACM International Conference on Multimedia*. Lisboa, Portugal, 2022: 3730-3741
- [28] Xu Y, Wang Y, Han K, et al. ReNAS: Relativistic evaluation of neural architecture search//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Nashville, USA, 2021: 4409-4418
- [29] Wang B, Xue B, Zhang M. Surrogate-assisted particle swarm optimization for evolving variable-length transferable blocks for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2022, 33(8): 3727-3740
- [30] Lu Z, Cheng R, Jin Y, et al. Neural architecture search as multiobjective optimization benchmarks: Problem formulation and performance assessment. *IEEE Transactions on Evolutionary Computation*, 2024, 28(2): 323-337
- [31] Wang Y, Li J, Chen C, et al. Scale adaptive fitness evaluation-based particle swarm optimisation for hyperparameter and architecture optimisation in neural networks and deep learning. *CAAI Transactions on Intelligence Technology*, 2023, 8(3): 849-862
- [32] Li J-Y, Zhan Z-H, Xu J, et al. Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameters optimization in convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(5): 2338-2352
- [33] Lu Z, Whalen I, Dhebar Y, et al. NSGA-Net: Neural architecture search using multi-objective genetic algorithm (extended abstract)//*Proceedings of the 29th International Joint Conference on Artificial Intelligence*. Yokohama, Japan, 2020: 4750-4754
- [34] Wei C, Niu C, Tang Y, et al. NPENAS: Neural predictor guided evolution for neural architecture search. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(11): 8441-8455
- [35] Ying C, Klein A, Christiansen E, et al. NAS-Bench-101: Towards reproducible neural architecture search//*Proceedings of the International Conference on Machine Learning*. Vienna, Austria, 2019, 97: 7105-7114

- [36] Dong X, Yang Y. NAS-Bench-201: Extending the scope of reproducible neural architecture search//Proceedings of the International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020
- [37] Dong X, Liu L, Musial K, Gabrys B. NATS-Bench: Benchmarking NAS algorithms for architecture topology and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(7): 3634-3646
- [38] Zela A, Siems J N, Zimmer L, et al. Surrogate NAS benchmarks: Going beyond the limited search spaces of tabular NAS benchmarks//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2021
- [39] Cai H, Gan C, Wang T, et al. Once-for-All: Train one network and specialize it for efficient deployment//Proceedings of the International Conference on Learning Representations. New Orleans, USA, 2019
- [40] Chen M, Peng H, Fu J, Ling H. AutoFormer: Searching transformers for visual recognition//Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, Canada, 2021: 12250-12260
- [41] Klein A, Falkner S, Springenberg J T, Hutter F. Learning curve prediction with Bayesian neural networks//Proceedings of the International Conference on Learning Representations. Toulon, France, 2017
- [42] Baker B, Gupta O, Raskar R, Naik N. Accelerating neural architecture search using performance prediction//Proceedings of the International Conference on Learning Representations. Vancouver, Canada, 2018
- [43] Dudziak L, Chau T, Abdelfattah M, et al. BRP-NAS: Prediction-based NAS using GCNs//Advances in Neural Information Processing Systems. Virtual-only, 2020, 33: 10480-10490
- [44] Wen W, Liu H, Chen Y, et al. Neural predictor for neural architecture search//Proceedings of the European Conference on Computer Vision. Glasgow, UK, 2020: 660-676
- [45] Chen Y, Guo Y, Chen Q, et al. Contrastive neural architecture search with neural architecture comparators//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA, 2021: 9497-9506
- [46] Wu J, Dai X, Chen D, et al. Stronger NAS with weaker predictors//Advances in Neural Information Processing Systems. Virtual-only, 2021, 34: 28904-28918
- [47] Zheng X, Ji R, Tang L, et al. Multinomial distribution learning for effective neural architecture search//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Republic of Korea, 2019: 1304-1313
- [48] Real E, Aggarwal A, Huang Y, Le Q V. Regularized evolution for image classifier architecture search//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu, USA, 2019, 33: 4780-4789
- [49] Fan L, Wang H. Surrogate-assisted evolutionary neural architecture search with network embedding. *Complex & Intelligent Systems*, 2023, 9(3): 3313-3331
- [50] Li L, Talwalkar A. Random search and reproducibility for neural architecture search//Proceedings of the 35th Uncertainty in Artificial Intelligence Conference. Tel Aviv, Israel, 2020: 367-377
- [51] Dong X, Yang Y. One-shot neural architecture search via self-evaluated template network//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Republic of Korea, 2019: 3680-3689
- [52] Pham H, Guan M, Zoph B, et al. Efficient neural architecture search via parameter sharing//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018, 80: 4095-4104
- [53] Liu H, Simonyan K, Yang Y. DARTS: Differentiable architecture search//Proceedings of the International Conference on Learning Representations. Vancouver, Canada, 2018
- [54] Falkner S, Klein A, Hutter F. BOHB: Robust and efficient hyperparameter optimization at scale//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018, 80: 1437-1446
- [55] Zoph B, Le Q V. Neural architecture search with reinforcement learning//Proceedings of the International Conference on Learning Representations. Toulon, France, 2017
- [56] Dong X, Yang Y. Searching for a robust neural architecture in four GPU hours//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, USA, 2019: 1761-1770
- [57] Cai H, Zhu L, Han S. ProxylessNAS: Direct neural architecture search on target task and hardware//Proceedings of the International Conference on Learning Representations. New Orleans, Louisiana, USA, 2019
- [58] Xu Y, Xie L, Zhang X, et al. PC-DARTS: Partial channel connections for memory-efficient architecture search//Proceedings of the International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020
- [59] Yang Y, You S, Li H, et al. Towards improving the consistency, efficiency, and flexibility of differentiable neural architecture search//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA, 2021: 6663-6672
- [60] Xie S, Zheng H, Liu C, Lin L. SNAS: Stochastic neural architecture search//Proceedings of the International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020
- [61] Tan H, Cheng R, Huang S, et al. RelativeNAS: Relative neural architecture search via slow-fast learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(1): 475-489
- [62] Yu H, Peng H, Huang Y, et al. Cyclic differentiable architecture search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023, 45(1): 211-228
- [63] Deng B, Yan J, Lin D. Peephole: Predicting network performance before training. *arXiv preprint arXiv: 1712.03351*, 2017
- [64] Liu Y, Tang Y, Sun Y. Homogeneous architecture augmentation for neural predictor//Proceedings of the IEEE/CVF

International Conference on Computer Vision. Montreal, Canada, 2021: 12229-12238

- [65] Tang Y, Wang Y, Xu Y, et al. A semi-supervised assessor of neural architectures//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2020: 1807-1816

[66] Luo R, Tian F, Qin T, et al. Neural architecture optimization //Advances in Neural Information Processing Systems. Montréal, Canada, 2018, 31: 7827-7838

- [67] Pytorch Document of Margin Ranking Loss (Website). <https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html>



JIANG Peng-Cheng, Ph. D. candidate. His main research interests include evolutionary neural architecture search and surrogate model.

XUE Yu, Ph. D. , professor. His main research interests include evolutionary computing, neural architecture search, artificial intelligence and feature selection.

Background

Neural architecture search (NAS) is an automated approach to designing neural networks, using which the most appropriate architecture can be found easily and quickly. Evolutionary neural architecture search (ENAS) is a general term for methods that use evolutionary computation to solve this problem. Despite the advantages of flexibility and automated design, ENAS is computationally expensive because the networks need to be trained for each fitness evaluation. Therefore, the study of efficient methods for searching evolutionary neural architectures is a current research hotspot. Nowadays, many researchers are devoted to studying surrogate models applicable to this problem thus reducing the huge consumption of resources and incorporating surrogate models into the process of evolutionary computation. In these studies, regression models are widely used to predict the performance of neural networks to reduce the pressure of evaluation. Some other researchers have tried to indirectly obtain the ranking between neural network performances using methods such as learning the binary comparative relationships. However, current methods generally cannot adapt well to performance ranking.

In this paper, we propose a score predictor that focuses more on the ranking information and integrate this predictor into the evaluation process of evolutionary algorithms. At first, small populations are used for real evaluation. When the predictor has been trained, it replaces the real evaluation process at a later stage thereby reducing the search time by a significant amount. The main contribution of this ranking-based score predictor is the ability of the score prediction to keep the ranking of its predicted values as consistent as

possible with the ranking of the actual performance. Unlike algorithms in previous research, this novel alternative adaptation does not approximate the accuracy rate, but rather aims to preserve its ranking relationship over the candidate architecture in general, which is capable of being used by the evolutionary algorithmic framework. The experiments in this paper were conducted on the EvoXBench platform and were able to achieve good results on all benchmarks. Compared to other methods, the proposed method is able to search for the optimal architecture on NASBench-101 space. The accuracy rates on the three datasets in NASBench-201 also achieve an improvement over the other optimal methods. In experiments utilizing real datasets on ImageNet, our method demonstrated an enhancement in classification accuracy. Moreover, with the same quantity of data, the ranking results generated by the proposed rank score prediction model exhibited an improvement in Kendall's Tau coefficient when compared to other optimal approaches. We further validated the effectiveness of One-hot encoding and the proposed rank loss within the surrogate model, demonstrating the efficacy of these two components for the overall algorithm.

This project belongs to the National Natural Science Foundation of China (NSFC) "Research on Multi-objective and Multi-task Evolutionary Deep Neural Networks Based on Efficient Surrogate Models". This project is dedicated to solving the current problem that NAS search is inefficient and cannot meet the needs of practical task scenarios. The results of this paper advance the study of surrogate models in this project and provide a partial foundation for subsequent research.