

快速进化神经网络架构搜索

摘要

本研究复现了 EG-NAS 算法，一种结合梯度下降和进化策略的神经架构搜索方法，旨在平衡搜索成本与性能。在 CIFAR-10 数据集上，我通过调整复合适应度函数中的 \mathcal{L}_1 和 \mathcal{L}_2 权重系数，我们发现当 \mathcal{L}_1 权重为 1， \mathcal{L}_2 权重为 0.4 时，模型性能最佳。此外，种群大小的实验显示，种群大小为 50 时验证准确率提升最快，搜索时间增幅最小。我们专注于 CIFAR-10 和 CIFAR-100 数据集，验证了 EG-NAS 在不同环境下的稳健性。实验结果表明，复现模型在 CIFAR-10 上达到了 2.73% 的测试误差率，接近原文的 2.53%，且搜索成本仅为 0.08 个 GPU 天，展现了 EG-NAS 的高效性。总体而言，EG-NAS 不仅保持了高性能，还实现了低成本的架构搜索，为自动化神经网络设计提供了有效的工具。

关键词：神经架构搜索；进化策略；复合适应度函数；高效搜索；

1 研究背景

在人工智能和机器学习的快速发展背景下，神经网络架构的设计已成为深度学习领域中一个至关重要的环节。在众多应用场景中，如图像识别、自然语言处理和强化学习等，一个优秀的神经网络架构是实现高性能模型的关键。然而，传统的人工设计方法存在明显的局限性：它不仅耗费大量时间和精力，而且受限于设计者的经验和直觉，往往难以找到最优的网络结构。此外，随着网络规模的增大和复杂性的提高，人工设计变得越来越不切实际。

为了克服这些挑战，神经架构搜索 (Neural Architecture Search, NAS) 技术应运而生，它利用算法自动化地搜索最优的网络架构。NAS 的出现极大地减轻了研究人员的负担，使他们能够将更多的精力投入到其他更高层次的任务中，同时也推动了深度学习模型的创新和发展。NAS 通过系统地探索可能的架构空间，寻找在特定任务上表现最佳的网络结构，从而提高了模型设计的效率和效果。尽管 NAS 展现出巨大的潜力，但在实际应用中，它仍然面临两大主要挑战：高昂的计算成本和耗时的搜索过程。这些挑战限制了 NAS 技术的广泛应用，尤其是在资源有限的研究环境和实际项目中。

本文复现了一种名为 EG-NAS [1](Evolutionary Gradient-Based Neural Architecture Search) 的算法，该算法由 Zicheng Cai 等人提出，旨在通过结合梯度下降和进化策略来有效平衡 NAS 中的搜索成本和性能。本研究主要目的在于验证 EG-NAS 算法在标准数据集上的性能，并分析其在搜索成本和模型性能之间如何实现更优的平衡。通过对 CIFAR-10 和 CIFAR-100 数据集的实验复现，我们旨在评估 EG-NAS 算法的有效性，并探讨其在自动化神经网络架构设计中的潜力。

2 国内外研究现状

为了降低搜索神经网络的成本，研究者们提出了多种基于不同原理的 NAS 方法，涵盖了基于进化策略、强化学习以及梯度下降的多种技术。这些方法在减少搜索时间方面取得了一定的进展，但在搜索成本和模型性能之间的平衡上尚未达到理想的效果。在这一领域，一些具有代表性的算法包括基于进化策略的 AmoebaNet-B、基于强化学习的 NASNet-A、以及基于梯度的方法 DARTS 及其变种 PC-DARTS 陆续被提出。

AmoebaNet-B [7] 作为一种基于进化策略的 NAS 方法，在 CIFAR-10 数据集上取得了 97.45% 的高准确率，但其高昂的搜索成本，需要 3150 GPU 天，限制了其在资源有限环境下的应用。NASNet-A [10] 作为一种基于强化学习的方法，虽然减少了搜索时间至 1800 GPU 天，但仍然面临着效率挑战。这些方法虽然在性能上取得了突破，但在计算资源和搜索时间上的高消耗促使研究者寻求更高效的解决方案。DARTS [4] 作为一种开创性的基于梯度的方法，通过连续松弛技术将离散的操作转化为连续可微分的权重，使得架构搜索过程可以通过梯度下降高效进行，显著减少了搜索时间至 0.4 GPU 天。PC-DARTS [9] 作为 DARTS 的改进版本，通过在搜索阶段随机选择部分通道进行计算，进一步减少了内存开销，将搜索时间缩短至 0.1 GPU 天，同时保持了模型性能。

除了上述方法，SNAS [8] 和 ENAS [6] 通过参数共享机制，减少了搜索过程中的计算量，允许在单一网络上评估多种架构。这种方法不仅降低了搜索成本，还提高了模型的泛化能力。EPCNAS [3] 结合了进化策略和梯度下降，通过同时利用这两种策略的优势，提高了搜索效率。NSGANetV1-A2 [5] 利用多目标优化的进化算法，同时考虑了架构的性能和多样性，为 NAS 领域带来了新的视角。P-DARTS [2] 通过引入路径 dropout 机制，增强了 DARTS 的正则化能力，进一步提高了搜索效率和模型性能。这种方法通过随机丢弃网络中的某些路径，防止了过拟合，提高了模型的鲁棒性。

EG-NAS 的提出代表了 NAS 领域中的一个创新尝试，它不仅继承了前述方法的优点，还引入了新的策略以优化搜索过程。EG-NAS 通过结合梯度下降的高效性与进化策略的全局搜索能力，探索了一条在搜索成本和模型性能间取得平衡的新路径。与单纯依赖梯度下降的方法相比，EG-NAS 通过改进的进化策略，能够更全面地探索架构空间，从而增加了找到高性能架构的可能性。同时，EG-NAS 对适应度函数的重新设计，考虑了个体间的相似性，进一步提升了搜索方向的多样性。这种综合考虑性能和多样性的方法，为自动化神经网络架构设计提供了新的视角，也为未来 NAS 技术的发展指明了新的方向。

3 数据集介绍

CIFAR-10 和 CIFAR-100 数据集是深度学习和计算机视觉领域中广泛使用的基准数据集，它们为研究人员提供了标准化的测试平台，用以评估和发展图像识别算法。

CIFAR-10 数据集包括 60,000 张 32x32 像素的彩色图像，这些图像被均匀分配到 10 个不同的类别中，每个类别包含 6,000 张图像。这些类别包括飞机、汽车、鸟类、猫、鹿、狗、青蛙、马、船和卡车。部分数据集如图 1 所示。CIFAR-10 数据集的图像尺寸虽然较小，但类别的多样性和背景的复杂性使其成为一个具有挑战性的测试平台。由于其图像尺寸较小，CIFAR-10 也常被用于测试轻量级网络架构的性能，以及研究如何在有限的数据量下实现有效的特征学

习。

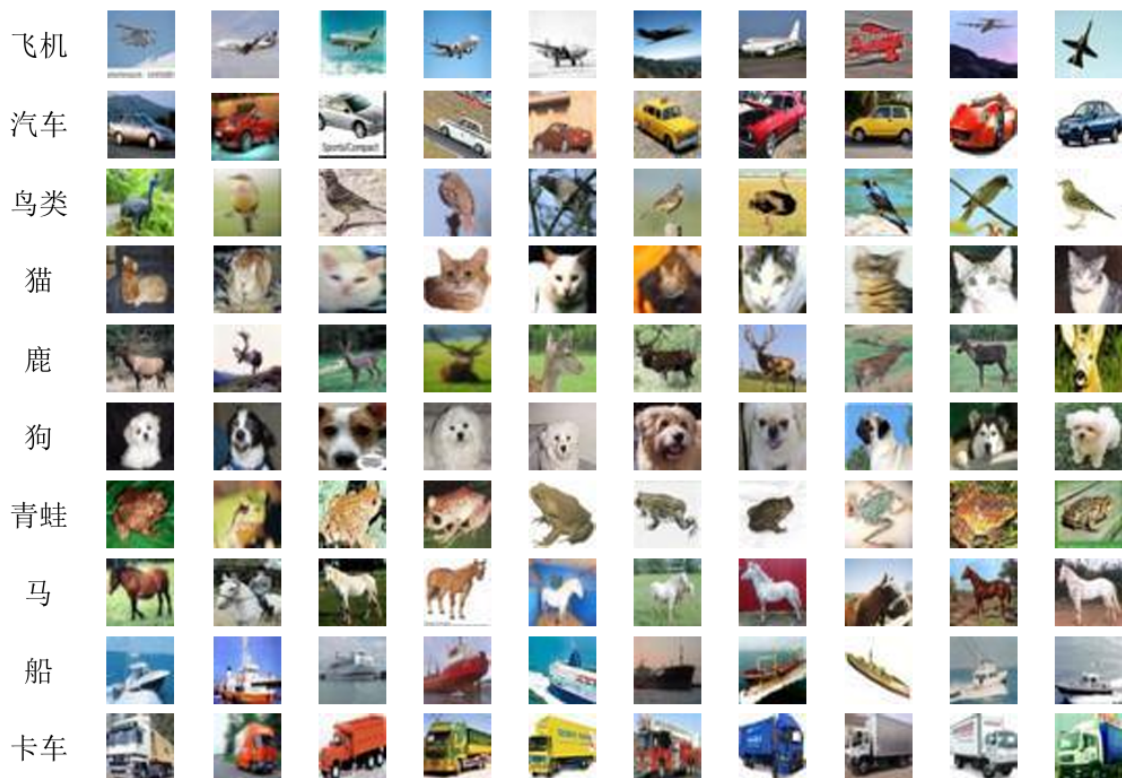


图 1. cifar10 数据集的部分图像

CIFAR-100 数据集在 CIFAR-10 的基础上增加了类别的数量，提供了 100 个类别，总共 60,000 张 32x32 像素的彩色图像。与 CIFAR-10 相比，CIFAR-100 的每个类别图像数量较少，仅 600 张，分为 50,000 张训练图像和 10,000 张测试图像。CIFAR-100 的设计旨在提供一个更具挑战性的视觉识别任务，因为其类别更多、每个类别的图像数量更少，这要求算法能够在类别间差异细微的情况下进行有效的识别。CIFAR-100 的复杂性和多样性使其成为评估模型泛化能力和鲁棒性的一个非常重要的数据集。

这两个数据集因其规模适中、多样性和挑战性而被广泛用于各种深度学习模型的基准测试。它们不仅用于验证模型的准确性，还用于研究模型的泛化能力、鲁棒性以及对不同类别和背景的适应性。CIFAR-10 和 CIFAR-100 数据集在图像识别领域的重要性不仅体现在它们的应用广泛，还体现在它们对推动计算机视觉技术进步的贡献上。通过在这些数据集上进行实验，研究人员可以更好地理解模型的行为，并据此改进模型的设计和训练策略。随着深度学习技术的不断进步，这些数据集将继续作为评估新算法性能的重要资源。

4 模型方法设计

4.1 可微分架构搜索 (Differentiable Architecture Search)

论文中提出的 EG-NAS 算法采用了可微分架构搜索 (DARTS) 作为基础框架。这种方法将网络分解为多个细胞结构，包括普通细胞和衰减细胞。普通细胞主要用于保持特征图的维度，而衰减细胞用于降低特征图的维度，即进行下采样。每个细胞包含多个节点，节点之间

的连接可以执行多种操作，如卷积、池化等。每条边上的各种操作的权重通过 softmax 激活函数转换为概率，表示选择每种操作的可能性。Softmax 混合计算公式如公式1所示。

$$\bar{o}^{(i,j)}(x^{(i)}) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x^i) \quad (1)$$

其中 i, j 代表节点的序号， $i < j$ ，所有候选运算都存储在 \mathcal{O} 中， $\alpha_o^{(i,j)}$ 表示运算 $o^{(i,j)}$ 在超网构造中的混合权重。经过训练后，概率最大的操作被选为该边的操作。这种方法允许通过梯度下降来优化网络架构，从而实现自动化的架构搜索。

$$\begin{aligned} \min_{\alpha} F(\omega^*(\alpha), \alpha) &= \mathcal{L}_{val}(\omega^*(\alpha), \alpha) \\ \text{s.t. } \omega^*(\alpha) &= \operatorname{argmin}_{\omega} \mathcal{L}_{train}(\omega, \alpha) \end{aligned} \quad (2)$$

算法的目标是解决公式2中的这个双层优化问题，即在给定的架构参数 α 下，找到最优的网络权重 ω ，使得网络在训练集上的性能最优；同时，找到最优的架构参数 α ，使得在验证集上的性能最优。

4.2 适应度函数 (Compound Fitness Function)

EG-NAS 还重新设计了一个复合适应度函数，该函数不仅考虑了模型的性能，如准确率，还考虑了个体之间的相似性。这种设计旨在增强搜索方向的多样性，避免搜索过程过早地收敛到局部最优解。适应度函数由两部分组成：交叉熵损失函数 (\mathcal{L}_1) 和余弦相似度函数 (\mathcal{L}_2)。交叉熵损失函数主要关注模型的准确率，而余弦相似度函数则是衡量当前架构与已有最佳架构之间的差异。通过调整 \mathcal{L}_1 和 \mathcal{L}_2 的权重，EG-NAS 能够在保持高性能的同时，鼓励架构的多样性。 \mathcal{L}_1 、 \mathcal{L}_2 公式及适应度函数如公式3所示：

$$\begin{aligned} \mathcal{L}_1(y_i, \hat{y}_i) &= - \sum_{i=1}^C y_i \log(\hat{y}_i) \\ \mathcal{L}_2(\alpha_t, z_i^{t+1}) &= \frac{1}{2} \left[\frac{t z_i^{t+1}}{\|t\| \|z_i^{t+1}\|} + 1 \right] \\ f(\alpha_t, z_i^{t+1}) &= \begin{cases} \zeta \mathcal{L}_1 - \eta \mathcal{L}_2 & \text{if } ACC_t > ACC(z_i^{t+1}) \\ \zeta \mathcal{L}_1 + \eta \mathcal{L}_2 & \text{if } ACC_t \leq ACC(z_i^{t+1}) \end{cases} \end{aligned} \quad (3)$$

其中 C 是类的数量， y_i 和 \hat{y}_i 分别为真实标签和架构 z_i^{t+1} 的预测标签。算法里对 \mathcal{L}_2 函数做了一个简单的调整，使函数 $\mathcal{L}_2 \in [0, 1]$ 。当 \mathcal{L}_2 结果趋于 1 时，表示两种架构相似。 ζ 为 \mathcal{L}_1 函数权重， η 为 \mathcal{L}_2 函数权重。

如果新架构的性能优于原始架构，则增加多样性损失的影响，鼓励新个体朝着增加架构多样性的方向发展。这样，即使新架构性能好，如果它与原始架构过于相似，也会受到惩罚。如果新架构的性能不如原始架构，则减少多样性损失的影响，引导新个体走向更有希望的表现。允许新架构在保持或提高性能的同时，模仿原始架构。

4.3 算法的主要流程

算法开始时，对网络的每一条边构建混合操作，通过 softmax 函数将操作权重转换为选择概率。在每个 epoch 的开始，算法通过梯度下降方法更新网络权重 ω ，以优化当前架构下的性能。然后对原始架构进行采样，并对原始架构添加高斯噪音，生成新的架构。在每次采样中又对这个采样生成的架构再次添加高斯噪音，以增加种群的多样性，并得到种群数数量的新架构。接下来，算法使用上面的复合适应度函数评估每个个体的性能，该函数不仅衡量模型的准确率，还考虑了个体之间的相似性，以促进搜索方向的多样性。在评估之后，算法选择表现好的一半架构用来更新候选架构集合，将每次采样得到的最优架构进行比较，并用其中最优的架构来更新参数 α 。每次迭代都会生成新的种群，评估，选择，并更新架构参数 α 。经过多次迭代后，算法将收敛到一个优化的架构参数集合 α ，这个集合定义了网络中的操作和连接，从而得到一个性能优异的神经网络架构。

5 实验结果分析

5.1 验证 \mathcal{L}_1 和 \mathcal{L}_2 在进化策略中的影响

在 CIFAR-10 数据集的搜索阶段，我们通过参数实验探索了 \mathcal{L}_1 （性能损失）和 \mathcal{L}_2 （多样性损失）在进化策略中的作用。

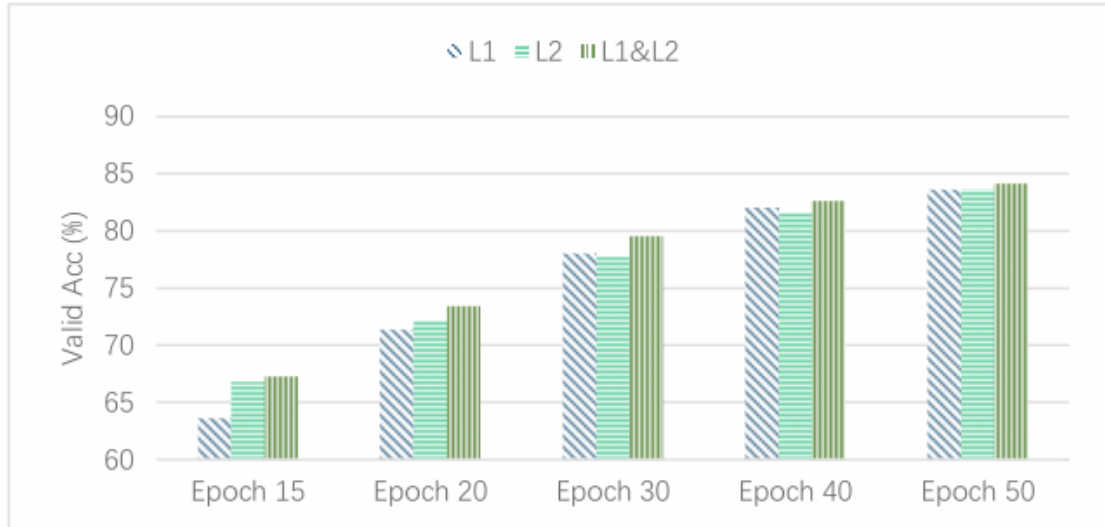


图 2. 在 CIFAR-10 搜索阶段， \mathcal{L}_1 和 \mathcal{L}_2 在进化策略中对 EG-NAS 的影响

从图2可以看出，结合 \mathcal{L}_1 和 \mathcal{L}_2 的适应度函数进行评估与单独使用 \mathcal{L}_1 或 \mathcal{L}_2 的适应度函数相比，能够有效提升模型的验证准确率。具体来说， \mathcal{L}_1 负责引导模型向高性能方向发展，而 \mathcal{L}_2 则鼓励架构的多样性，避免搜索过程过早收敛至局部最优解。这种结合 \mathcal{L}_1 和 \mathcal{L}_2 的方法不仅提升了模型的准确率，也增加了搜索过程中的探索性，从而有助于发现更优的网络架构。

5.2 复合适应度函数不同权重的 \mathcal{L}_1 、 \mathcal{L}_2 对性能影响

在 CIFAR-10 数据集的搜索阶段，我们进一步验证了复合适应度函数中不同权重的 \mathcal{L}_1 、 \mathcal{L}_2 对性能的影响。通过调整这些系数的值，我们能够控制适应度函数中性能损失和多样性损失的相对重要性。如表 1 所示，我们的实验结果与原文中的结果一致，证实了当 \mathcal{L}_1 权重 (ζ) 为 1.0， \mathcal{L}_2 权重 (η) 为 0.4 时，模型能够达到最佳的性能。这一结果表明，适当的权重设置可以显著影响搜索效率和最终模型的性能，进一步证实了复合适应度函数在平衡探索和利用方面的重要性。

表 1. 在 CIFAR-10 上，验证复合适应度函数的不同系数 ζ 和 η 值对任务的影响

ζ	$\eta = 1.0$		$\eta = 0.8$	
	valid acc (%)	params (M)	valid acc (%)	params (M)
1.0	84.036	3.00	83.984	2.36
0.8	83.980	2.39	83.668	2.74
0.4	83.768	3.17	83.852	3.51
0.2	83.768	3.51	83.880	2.48
ζ	$\eta = 0.4$		$\eta = 0.2$	
	valid acc (%)	params (M)	valid acc (%)	params (M)
1.0	84.088	2.58	84.028	3.21
0.8	83.724	1.83	83.356	2.28
0.4	83.632	2.77	83.184	3.46
0.2	83.684	2.77	83.096	3.40

5.3 种群大小对算法性能的影响

我们还验证了种群大小对 EG-NAS 算法性能的影响。种群大小是指在进化策略中生成的候选解的数量，它直接影响搜索的广度和深度。



图 3. 在 CIFAR-10 搜索阶段，验证种群大小对 EG-NAS 的影响

如图3所示，通过在 CIFAR-10 数据集上进行实验，我们发现当种群大小从 30 增加到 50 时，验证准确率提升最为显著，且搜索时间的增幅不大。这表明种群大小为 50 时，算法能够在保持较高搜索效率的同时，有效地探索更广泛的架构空间，从而提高了模型的性能。这一发现对于确定合适的种群大小以平衡搜索效率和性能具有重要意义。

5.4 训练评估搜索到的最优架构

在 CIFAR-10 数据集上，我们通过多次随机种子的实验，最终确定了最优架构。通过二十多次搜索后，找到的最优架构的验证准确率能够达到 84% 左右。如表2所示，为了进一步验证这一架构的稳健性，我们在不同的种子设置下对找到的最优架构进行了四次独立的训练和评估，最终得到的平均错误率为 2.73%。虽然这一结果略高于原文报告的 2.53%，但是与原文的差异不大，考虑到实验环境和条件的差异，这一结果仍然显示出 EG-NAS 算法在寻找高性能架构方面的潜力。

表 2. 在 CIFAR10 和 CIFAR100 上，在 4 个不同的种子上对架构进行重复训练评估

Architecture	Test Error(%)		Params(M)	Search Cost(GPU-Days)
	CIFAR10	CIFAR100		
1	2.77	18.05	2.58 / 2.39	0.08
2	2.68	17.79		
3	2.84	17.96		
4	2.65	17.84		
Avg	2.73	17.84		

对于更具挑战性的 CIFAR-100 数据集，我们所找到的最优架构在训练后得到的错误率 17.91% 比原文中的 16.22% 要高。这一结果可能暗示着在更复杂的数据集上，EG-NAS 算法需要更精细的调整或更广泛的搜索以发现更优的架构。由于搜索和训练存在随机性，我认为通过进一步对最优架构的搜索和训练，有望找到性能更接近原文报告的架构，得到更好的实验结果。尽管 CIFAR-100 的搜索空间更大，挑战更多，但 EG-NAS 算法仍具有发掘高性能架构的潜力。

6 结论

在本次对 EG-NAS 算法的复现实验中，我们在 CIFAR-10 和 CIFAR-100 数据集上对 EG-NAS 算法的性能进行了验证。虽然与原文报告的性能存在一些差异，但这些差异在可接受的范围内。EG-NAS 算法展现出了显著的搜索效率，平均只需 0.08 个 GPU 天，与原文报告的 0.1 个 GPU 天相一致，这进一步证实了 EG-NAS 在实际应用中的高效性。这一效率的提升对于资源有限的研究环境尤为重要，因为它极大地减少了算法运行所需的计算资源。

更重要的是，尽管 NAS 的搜索过程在初期可能需要较大的计算资源，但一旦确定了优秀的架构，EG-NAS 算法能够在后续的项目中大幅节省模型调整和超参数优化的时间与资源。这一点在实际应用中尤为重要，因为在实际的项目开发中，时间往往比计算资源更为宝贵。通

过 EG-NAS 算法, 研究人员和工程师可以快速地找到高性能的网络架构, 并将其应用于各种任务, 而无需花费大量时间在手动调整和优化上。

此次实验不仅验证了 EG-NAS 算法的有效性, 还展示了其在实际应用中的潜力。EG-NAS 算法的高效性和灵活性使其成为一个有吸引力的选择, 特别是在需要快速迭代和优化模型架构的场景中。随着深度学习技术的不断发展, 我们期待 EG-NAS 算法能够在未来的自动化神经网络设计中发挥更大的作用。

参考文献

- [1] Zicheng Cai, Lei Chen, Peng Liu, Tongtao Ling, and Yutao Lai. Eg-nas: Neural architecture search with fast evolutionary exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11159–11167, 2024.
- [2] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1294–1303, 2019.
- [3] Junhao Huang, Bing Xue, Yanan Sun, Mengjie Zhang, and Gary G Yen. Particle swarm optimization for compact neural architecture search for image classification. *IEEE Transactions on Evolutionary Computation*, 27(5):1298–1312, 2022.
- [4] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.
- [5] Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and Wolfgang Banzhaf. Nsga-net: neural architecture search using multi-objective genetic algorithm. In *Proceedings of the genetic and evolutionary computation conference*, pages 419–427, 2019.
- [6] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR, 2018.
- [7] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.
- [8] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*, 2018.
- [9] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient architecture search. In *International Conference on Learning Representations*, 2020.

- [10] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.