

宏观层面的免疫启发方法 神经集成搜索

吕克·弗拉雄
赫瑞瓦特大学数学与计算机科学
学院英国爱丁堡
luc.frachon@hw.ac.uk

庞伟
数学与计算机科学学院
赫瑞瓦特大学
英国爱丁堡
奥西德 ID 0000-0002-1761-6659

George M. Coghill自然与计
算科学学院 阿伯丁大学英
国阿伯丁 ORCID iD
0000-0002-2047-8277

摘要 近年来,人们对应用于深度神经网络架构自动设计的进化计算产生了新的兴趣,即神经架构搜索 (NAS)。NAS 中进化方法的优点包括概念简单以及在搜索空间定义和/或优化目标方面的灵活性。

然而,遵循进化计算范式的人工免疫系统 (AIS) 在 NAS 中的探索较少。在这项研究中,我们的目标是利用其内在和卓越的能力来平衡性能和种群多样性,以开发一种基于克隆选择算法 [1] 的新型神经集成搜索方法。为了更通用,我们专注于设计宏观架构而不是架构组件。

在流行的计算机视觉基准上进行的实验表明,尽管增强和后处理最少,但我们的方法达到了具有竞争力的准确性和效率。我们展示了 AIS 带来了切实的好处,包括保持解决方案的多样性、语义上直接的实现和高效率。此外,该 AIS 可以表现出“二次响应”:当出现相关但更困难的任务时,集成将在对架构或训练协议进行零修改的情况下胜任执行。

索引词 深度神经网络、人工免疫系统
项,神经集成搜索

一、引言

深度学习 (DL) 的成就很大程度上归功于专家团队精心开发的巧妙架构。神经网络配置的数量随着网络的深度呈指数增长。为了使问题易于处理,进行了简化假设,例如重复的操作块 [2]、[3]。即便如此,对于大多数研究团队来说,唯一可行的选择是迁移学习;然而,在与目标任务“相似”的数据上训练的模型并不总是可用的。因此,神经架构搜索 (NAS) 的吸引力越来越大,即自动设计 DL 架构的方法。在本节中,我们首先简要概述 NAS 中的不同方法,然后介绍人工免疫系统 (AIS) 的一些优势。最后,我们讨论微观和宏观搜索空间的特征。

A. NAS 方法

Zoph等人的第一项专注于自动化深度神经网络架构设计的工作是 [4],他们训练了强化学习 (RL) 代理来选择和配置层。他们取得了与一些最好的手工制作模型相当的结果,尽管有很多后处理。然而,这些结果需要超过 20,000 个 GPU 日,这使得该方法在很大程度上不切实际。大约在同一时间,Real等人。[5] 通过使用进化算法 (EA) 在 3,000 个 GPU 日内取得了类似的结果。

近年来,NAS 方法激增。除了 RL 和 EA,其他范例也被用来指导搜索,包括贝叶斯优化 [6]、[7]、一次性架构搜索 [8]–[10]。虽然高效,但后者使用限制性搜索空间。它的有效性也受到了审查 [11]–[13],建议性能更多地来自搜索空间工程或精心设计的培训程序,而不是搜索本身。此外,最先进的 (SOTA) 方法总是采用大量增强技术、后处理或训练技巧来改善结果,这在很大程度上取决于人类的专业知识 [13]。这与 NAS 的观点有些不一致,并且混淆了方法之间的比较。

进化方法由于其灵活性、易于实施和效率而在 NAS 中很受欢迎。示例包括遗传算法 (GA) [5]、[14]、[15]、遗传编程 [16] 和各种群体智能算法 [17]。然而,到目前为止,AIS 几乎没有被探索过。

一个新兴趋势是神经集成搜索 (NES),即使用 NAS 来构建神经网络的集成 [18]、[19]。在后一项工作中 (约会本文的预印本),Zaidi等人。直接优化集成性能并展示神经集成如何帮助处理测试集损坏。我们的重点不同:我们想了解 AIS 是否能在宏观级 NES 中带来显着优势,以及这种方法是否可以替代传统 NAS。

B. 人工免疫系统的好处

AIS 算法是在 1990 年代和 2000 年代初期针对模式识别和优化问题开发的 [1]。

他们以人口为基础,相比之下他们的主要优势

对许多 EA 来说,它们具有出色的内在平衡性能与种群多样性的能力,以及在多模态问题空间中进行搜索的能力 [1],[20]、[21]。

这些特性使 AIS 成为 NES 的自然候选者。事实上,多样化的人群使我们能够集成模型以获得经济的性能增益,因为模型已经可用。之前已经探索了 AIS 在浅层神经网络中的超参数 (HP) 优化的使用,并且 [21] 表明 AIS 比其他基于群体的算法能够实现更多样化,因此更有效的集成。这促使我们研究将 AIS 应用于深度神经集合的搜索。

C. 宏观与微观搜索空间NAS 的一个关键困难

是平衡搜索空间的大小与其表达能力和创新潜力。

自 [22] 以来,大多数 NAS 工作采用微搜索方法:他们不搜索完整的“宏观架构”,而是只设计最多两个不同的单元和少量组件。候选架构是手工设计的骨架中的一系列重复单元。微架构搜索极大地缩小了搜索空间。它还有助于转移到更复杂的任务,只需增加通道和细胞数量即可。

此类架构复制了常见手工制作模型 (例如 ResNet [2])的重复模式。虽然基于单元格的搜索可能是高效的并产生 SOTA 结果,但它有几个缺点:1)宏架构缺乏创新; 2) 同一个被发现的小区必须在网络中不同点的各种环境中工作; 3)对可能并不总是可用的主题专业知识的要求,例如在手动修复网络深度时; 4) 至关重要,参考文献 [13] 表明人为设计的元素在性能方面比自动搜索发挥更大的作用。在 [23] 中,Hu等人。表明微观搜索和宏观搜索之间的性能差异主要来自它们的起点:微观搜索方法从更复杂的候选对象开始,这些候选对象主要是手工制作的,并且这种优势贯穿于整个搜索过程。相比之下,目前的工作使用分支宏架构空间,其初始架构不是手工设计的。此外,它可以生成任意深度的模型,这在先前知识稀缺时很有用。

D. 研究问题而不是在基准上追

逐 SOTA 性能数据集,我们旨在回答以下问题:

·进化的宏观搜索方法能否产生具有竞争力的结果,即不求助于复杂的预处理和后处理工程? · AIS 的使用是否在基于人口的既定基线的集成方面提供了显着优势?

·这样的集成是否能够转移到更难的任务?

在本文的其余部分,我们首先在第 II 节中介绍 ImmuNES 框架。在第三节中,我们详细介绍了搜索算法及其运算符。然后我们呈现

第四节的实验结果。最后,在第五节中,我们讨论了我们的结果并建议了未来的工作。

二.免疫框架

在本节中,我们首先描述 ImmunES 中包含的关键思想以及它们如何相互依赖。然后我们解释架构是如何编码的。最后,我们展示了搜索空间。

A. 关键概念 1) 人工免

疫系统: AIS 是一系列基于群体的算法。一个典型的例子是克隆选择算法 (CLONALG) [1],它与 GA 有许多相似之处,但具有特定的免疫算子。第一个关键区别是克隆的突变大小源自父代的性能 (亲和力),而不是均匀采样。这使得搜索可以在优秀个体周围进行局部搜索,而在穷人周围进行更广泛的搜索,从而平衡探索和利用。其次,GA 依靠交叉操作来改进探索,而 CLONALG 只是简单地添加一些随机生成的个体。在宏 NAS 中,这似乎比合并来自两个网络的部分更合适,因为层在搜索过程中相互适应。

2)神经网络集成:神经网络集成 (NNE)已经被研究了很长时间,包括在卷积神经网络 (CNN)[24]的背景下。它们的性能基本上取决于两个因素:模型的质量和它们的错误多样性。后者可以作为一个明确的优化目标,例如 [25]。另一种方法是使用本质上能够保持多样性的搜索算法,例如 AIS。特别是,[21] 表明,仅针对准确性进行优化的 AIS 在准确性和多样性方面都优于多任务优化。

3) Evolutionary NAS Features of Evolutionary NAS:基于EA的NAS 通常有两个重要的特征:

·性能回报递减:搜索大部分时间都处于平稳阶段,性能增益变得越来越昂贵; · 每一代都有大量候选人评估。特别是,最后一代通常包含数十个个体,它们都包含通过搜索学到的有用信息。

4) 把所有东西放在一起:基于上面的观察,我们不是花费许多 GPU 天来争取小的性能提升并丢弃最终种群中除了一个成员之外的所有成员,而是在性能开始趋于平稳时停止搜索,并利用有能力的人以及我们的 AIS 发现的不同人群来构建一个整体。这减少了因提前停止而导致的性能差距。这种权衡,即最终训练的一次性额外成本和较慢的推理时间,在许多应用中是可以接受的,例如,在预测不是实时进行的情况下。

B. 架构的表示我们将架构表示为有向无环图,其中节点对应于层/块,边对应于张量,

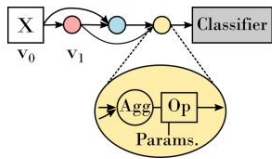


图 1. 具有三个隐藏层的架构的图形表示。

表一 简单搜索空间 (HP :超参数)			
手术 类型	HP 1 {价值观}	HP 2 {价值观}	HP 3 {价值观}
转换	内核大小 {1,3,5,7}	批量归一化 {是/否}	ReLU {是/否}
DSepConv	内核大小 {1,3,5,7}	批量归一化 {是/否}	ReLU {是/否}
水池	类型 {最大值,平均值}	内核大小 {3,5}	Ch.乘数 {1, 1 3, 1 2, 3, 2}
身份	-	-	-

见图1:节点v0是输入节点,接收数据样本。每个后续节点v_i至少有一个传入边,它来自节点v_{i-1}。除了v₁之外的每个隐藏层都可以选择从任何节点v_{i-k}中获得第二个传入边, k > 1,从而创建跳过连接 [2]。输入节点是唯一允许出度大于2的节点,这样所有的skip connections总能找到一个源节点。

每个节点都有一个聚合和一个操作。Aggs用于组合来自较早层的张量。当节点的入度为 1 时,它们为None,但当入度为 2 时,它们可以为Add或Concatenate。Ops对其输入应用数学变换 (例如 ResNet 块) 。根据表 I 和 II,每个操作可以有多个 HP。

C. 搜索空间

尽管在 NAS 搜索空间设计中人为偏差是不可避免的 [13],但我们希望发现具有创新潜力的宏架构,而不是手动强制执行 (参见第 IC 节)。我们定义了两个搜索空间:我们更简单的实验中使用的 “简单”空间,以及全面实验中使用的 “块”搜索空间。

1) 简单搜索空间:我们的方法最初是使用这个搜索空间和 Fashion-MNIST 数据集 [27] 开发的。这个任务是轻量级的,但具有足够的挑战性以产生性能差异。

我们定义了这个搜索空间以允许在开发过程中进行快速迭代 (见表 I)。所选操作是传统 CNN 的典型操作。Identity Op允许算法删除之前添加的层。通道。multiplier是应用于通道计数的乘法因子。输入层始终是没有批量归一化或激活函数的逐点卷积。它的作用是将通道数从 1 增加到 64,以快速为网络提供更多容量。分类器或网络头始终是全局序列

表二
块搜索空间 (HP :超参数)

操作类型	HP 1 {可能的值}	HP 2 {可能的值}
Resnet 块 [2]	内核大小 {3,5}	下采样{是/否}
Resnet 瓶颈 方块 [2]	内核大小 {3,5}	下采样{是/否}
密集网块 [3]	生长因子 {12, 24, 36}	过渡层{是/否}
密集网瓶颈 方块 [3]	生长因子 {12, 24, 36}	过渡层{是/否}
盗梦空间-Resnet A座 [26]	内核大小 {3, 5}	瓶颈因素 {0.1, 0.4, 0.75}
盗梦空间-Resnet B座 [26]	内核大小 {3, 5}	瓶颈因素 {0.1, 0.4, 0.75}
水池	类型 {最大值,平均值}	内核大小 {3,5}
身份	-	-

级联池化、批量归一化、 dropout (pdrop = 0.2) 和全连接层。

2)块搜索空间:我们的主要实验是在 CIFAR-10 [28] 上运行的。为此任务开发的体系结构传统上涉及相同块的重复 [2]、[3]、[26]。

几乎所有最近的 NAS 作品都复制了这种做法,并且只自动设计这些块 (或 “单元”) ,而不是整体架构。这些宏架构的关键特征 (深度、宽度等)是预定义的,这是主要的人为决定 (参见第 IC 节)。

为了为 CIFAR-10 生成具有足够容量的宏架构,我们还逐块构建它们,而不是一次一个操作,但没有强制重复相同的单元格。为了在避免搜索空间优化的同时做到这一点,我们创建了一个高级模块菜单,这些模块是从流行的深度学习文献中先验选择的。这与 [16] 类似,但配置更多。与所有微观和一些宏观搜索方法 [13] 不同,我们不预先定义块的数量、它们的位置、跳过连接的位置、网络的最大深度 1或激活图的维度。与微型 NAS 产生的架构不同,跳过连接可以发生在宏观层面。

这种自由为有趣的架构的出现提供了机会。同时,我们通过采用渐进式搜索 (参见第 III-C 节)使搜索变得易于处理。

在表 II 中, Downsample表示块是否应减小空间大小并增加通道数,增长 2 倍。增长因子是张量在通过块时获得的通道数。过渡层指示是否附加压缩层以使通道数减半。瓶颈因子是应用于块中间分支的通道数的压缩因子。我们

1虽然我们定义了理论上的最大世代数,但它是由于提前停止,在实践中从未达到。

AIS 的神经集成搜索

```
要求: N0:初始种群大小, Lini:初始层数, p:突变因子, nc:每个父代的克隆数, ni: # 随机
插入, na:每个父代的增强网络, π1:外循环耐心, τ1:外循环阈值, π2:内循环耐心, τ2:内
循环阈值pop ← MakeRandomArchitectures(N0) pop ← MakeAugmentedCopies(pop,
na) {每个父级的 na 个副本以增加初始光束宽度}

Evaluate(pop)
counter1 ← 1
while counter1 ≤ π1 do
  counter2 ← 1 while
    counter2 ≤ π2 do
      clones ← Clone(pop, nc) {每个父代的 nc 副本}
      clones ← Mutate(clones, p)
      Evaluate(clones)
      pop ← SelectNBest(pop ∪ clones, N0) avg
      affinity ← ComputeAverageAffinity(pop) pop 愿
      RandomArchitectures(Ni)
      Evaluate(pop{N0,...,N0+ni}) 如果平
      均亲和力改善小于τ2那么 counter2 ← counter2 + 1 结束
      如果

    结束时
    pop ← MakeAugmentedCopies(pop, na)
    Evaluate(pop{N+1,...,N(1+na)}) {N = N0 + ni}
    avg affinity ← ComputeAverageAffinity(pop)
    如果 avg affinity 改善小于τ1那么 counter1 ←
    counter1 + 1 结束如果

返回弹出时结
束
```

图 2. AIS 搜索程序的伪代码

有关这些模块的详细信息,请读者参阅相应的论文。

输入层和分类层与之前相同,但具有 32 个初始通道以减少架构的内存占用并允
许更深层次的模型。

三、 AIS的神经集成搜索

在本节中,我们将描述我们的方法及其关键组成部分。搜索算法的伪代码如图 III
所示。我们可以互换使用“层”和“块”。

一、概述

搜索由 AIS 进行,从N0个具有随机层的小型网络开始。为了启动这个过程,我们
首先制作这些候选者的多个副本并扩充它们,即为每个候选者附加一个随机层(参
见第 III-C 节)。对父母和后代进行培训,然后在验证集上进行评估(参见第 III-D
节)。他们的验证准确性代表了他们与任务的亲和力。

随后,在每一代,每个网络都会产生nc个克隆。这些克隆对其连接、 Aggs和Ops
进行突变。对生成的架构进行评估,并保留来自父代和克隆池的最佳N0以形成下一
代。然后计算人口的平均亲和力。如果它在π2代的耐心时间内没有提高超过阈值τ2,
则种群将再次进行扩充(随机附加块)。这种渐进式搜索机制允许

AIS 为每个任务生成最小网络。在每一代,一个或两个随机架构(深度等于种群的当
前平均深度)也被插入到种群中以增加探索。所有层都可以随时发生突变,这可以防止
种群被锁定在先前层定义的搜索空间的子区域中。

然后整个过程从克隆和突变步骤重新开始。当π1个连续增强阶段没有产生超过
τ1 的平均亲和力改进时,搜索终止。为简单起见,我们设置π1 = π2, τ1 = τ2。通过调整
这些值,可以使搜索在学习曲线中更早或更晚停止。在我们的主要实验中,我们在两代
内使用了 0.3% 的阈值,对应于学习曲线的一个相对陡峭的部分,我们还观察到调整
这个值(在合理范围内)对性能并不重要。

B. 克隆、变异和权重继承

在 AIS 中,突变的幅度取决于克隆父代的亲和力。在我们的例子中,因为较早的层
已经有多次变异机会,所以我们将线性较大的变异方差分配给较新的层,即更靠近网
络的头部。在实践中,给定一个具有 − ρ fp L 层的克隆,突变率 α 计算为 α = e,其中fp
是父体系结构的亲和力,ρ 是元参数。然后对于第l层,随机采样突变μ的强度如下:

$$\mu \sim N(0, \sigma^2) \text{ 其中 } \sigma = \alpha^{\frac{l+1}{\text{大小}}} \tag{1}$$

与 [5] 类似,我们为每个 HP 存储一个 [0, 1] 范围内的连续值,并在将其解码为架
构特征时将其动态离散化。这允许甚至很小的 μ 值随着时间的推移累加并触发架构
突变。

离散化只涉及将间隔分成与离散 HP 的可能值一样多的 bin。

对于每个克隆,我们首先尝试改变连接,然后是每个节点Agg,然后是它们的Op类
型,最后是它们的Op HP。所有这些扰动都遵循突变强度方程 (1),并且只有在前一步
没有导致离散值发生变化的情况下才会执行每一步。这确保在每一轮突变中只进行
微小的改变,从而保持父母和克隆人表现之间的一致性。我们在简单搜索空间中运行
了一个验证实验,显示了父代与其克隆体的平均亲和力之间的 Spearman 等级相关
值,范围在深度 3 的 0.53 和深度 9 的 0.70 之间。

所有克隆都在不变的层中继承其父母的权重。
在其他层中,我们应用 He 初始化 [29]。

C. 架构增强

我们通过渐进式搜索来发展架构:我们定期克隆所有网络,并在网络头部之前的
每个克隆中插入一个随机层,并使用随机(可选)跳过连接(见图 3)。原来的网络

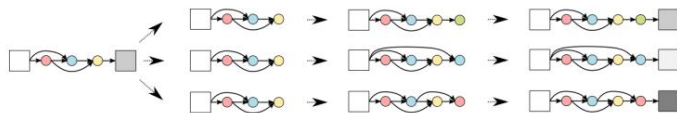


图 3. 渐进式搜索的架构增强策略。

留在人口中,以便可以忽略无用的增加。与突变一样,克隆体在不变层中继承其父代学习的权重,并且可以进一步训练这些权重。这种方法类似于带微调的迁移学习,提高了新网络训练过程的时间和数据效率。在第二个验证实验中,我们发现增强前和增强后亲和力之间的等级相关值为 0.66,其中浅层网络受益最大,并且没有灾难性的性能损失。

D. 搜索期间的评估

由于每个体系结构的评估是演化 NAS 的主要瓶颈,我们通过部分评估来提高效率。每个候选人仅接受 20% 的训练集训练,并采用积极的提前停止策略:如果验证准确度在 2 个时期内没有提高超过 0.3%,则训练停止。在较小规模的设置中,发现这些值在不到 0.5 GPU 天的时间内产生了良好的性能(参见第 IV-A 节)。每次验证准确性提高时,我们都会存储网络的权重,以便在开始提前停止时(或达到允许的最大 epoch 数),这些预训练的权重可以用作未来克隆的起点。最佳验证准确性记录为亲和力分数。第三个验证实验发现部分和完整评估结果之间存在高度相关性,在深度 3 时为 0.91,在深度 9 时为 0.65。后者的相关值可能会通过在部分评估中训练大型网络更长时间来增加。

在搜索过程中,通过计算每一代的平均种群亲和力来监控进度。我们尝试了更复杂的方法来明确实施多样性,如 [25] 中所述,以及一种过滤掉类似架构的机制,如 Opt-AINet [20] 中的一种免疫网络方法。然而,事实证明 CLONALG 保持多样性的内在能力同样有效,因此我们保留了更简单的方法。

E. 神经集成构建

一旦搜索终止,我们加载看到的最好的一代,因此有 N_0 个部分训练的架构。关于有多少人应该成为整体的一部分的决定涉及准确性和最终训练时间之间的权衡。为 CIFAR-10 开发的架构往往训练起来相当昂贵(每个需要几个 GPU 小时),并且随着更多网络的添加,准确性会随着收益递减而提高。本文给出的结果使用了 5 个最佳架构(按亲和力排名),以便在最后的再训练阶段花费大约 1 GPU 天。²我们完全训练

²我们注意到很少有论文报告最终训练所花费的时间,有时 [10] 包括非常昂贵的技术,例如 AutoAugment [30]。

与部分训练相比,它们没有改变正则化和增强协议,只是我们不再启用提前停止(详见表 IV)。

NNE 的类别预测是通过加权软多数投票获得的。给定一个数据样本 X_i 和一个包含 N_c 个神经网络 g_j 的集合,其亲和力得分为 f_j ($j \in \{1, \dots, N_c\}$),我们得到一个 k 维概率向量 $g_j(X_i)$ 的集合,其中 k 是我们分类任务中的类数。然后我们计算由相应架构的归一化亲和力和加权的这些向量的总和 $G(X_i)$ 。最后,NNE 的预测由 $\arg\max G(X_i)$ 给出。

四、实验和结果

在本节中,我们首先讨论元参数的选择。然后将我们的主要实验结果与现有的 NAS 方法进行比较,两者都使用更简单的任务(简单搜索空间)和更复杂的任务。我们还确认我们的方法明显优于随机搜索,这解决了对许多 NAS 方法 [11]-[13] 的主要批评之一。

与 GA 的比较展示了 AIS 的优势。

最后,我们证明尽管没有使用重复的单元格,我们的方法允许任务转移。

A. 元参数选择

杨等。[13] 指出许多论文忽略了解释如何选择元参数,即使它们对最终性能有很大贡献。我们在下面描述我们的元参数选择过程,并在表 III 中详细说明了我们的实验中使用的值: 1) 搜索参数:使用更简单的任务设置(Fashion-MNIST)选择元参数,以允许更快的迭代。种群大小设置为 12,该值接近原始 CLONALG 论文 [1] 中使用的 10,并且具有大量除数(在我们的实施中很有用)。选择突变和增强克隆的数量以实现大约 0.5 GPU 天的搜索时间。随机插入率和突变因子遵循 [1] 中使用的值(初始种群大小的 10%, $p = 5$)。我们观察到对提前停止耐心 π 和阈值 τ 的低敏感性:将 τ 减半仅将最终精度改变了 0.1%。然后,我们在转向 CIFAR-10 时应用了以下经验法则更改:更多增强副本以探索块搜索空间的更大基数;较低的阈值 τ 以解释较慢的进展;扰动操作类型时将变异方差减半,以反映变异区间 $[0, 1]$ (更多操作)中更高的解密度。

2) 部分和最终评估参数:我们将数据增强限制在 NAS 中最常见的技术:pad-crop、水平翻转、Cutout [31]。正则化仅限于分类器层中的权重衰减和丢失。我们的最终训练协议类似于部分评估,不同之处在于我们将初始学习率降低了 3 倍,增加了训练时间,使用了完整的训练集,并添加了热重启,这在余弦学习率退火 [32] 中很常见。[13] 表明添加了 DropPath [33]、辅助塔 [34] 和

表三
元参数设置

搜索空间：	简单的	堵塞
搜索		
初始种群大小 N	12	12
初始随机层数	3	6
每个父母的突变/增强克隆	3 / 3	3 / 5
每代随机插入		
突变因子 p	1	1
耐心 π / 阈值 τ	5.0 2 / 0.0075	5.0 2 / 0.005
部分评估		
训练集大小（20% 的数据）	12000	10000
验证集大小	12000	10000
批量大小	128	64
数据扩充	旱作	Pad-crop, Flip, Cutout
Adam 初始学习率 Learning rate	0.1	0.05
annealing Weight decay 早停耐	余弦	余弦
心/阈值 Max.纪元	10—5	10—5
	2 / 0.005	2 / 0.003 30
	15	
期末培训		
整体规模	12	
训练集大小	60000	5 50000
Adam 初始学习率	0.1 3	0.05 3
在纪元后重启	25	30, 90
历元数	50	210

“ 未提及的元参数与部分评估相同。

表IV
FASHION-MNIST 的实验结果

方法	举报了	测试精度 % GPU-days	
1 个XNAS [10]	最好	96.36	0.3
深群 [17]	最好	93.56	1.2
NES-RE 恩斯。 30,无偏移 [19] 3,†平均值± CI Auto-Keras [35]	最好	93.0±0.1	不适
纳什 [36]	最好	92.56	用 0.5
梯度演化 [37]	最佳（医学）	91.36 (90.58)	不适用
免疫力（我们的） †	最佳 (μ ± σ)	94.61 (94.37±0.15)	0.4

“ 微搜索,从 CIFAR-10 上的搜索转移的架构,强正则化和扩充如 [35] 中实现

进化方法 †带有集成的进化方法。 “ 没有增强。

即使是 AutoAugment [30] 也可以在 ResNet-50 [2] 上将 top-1 精度提高近 2%,这超过了大多数 NAS 算法对随机搜索的贡献。由于我们不追求 SOTA top-1 性能,因此我们选择不应用这些技术来更准确地表示我们方法的贡献。

B. 神经集成搜索实验

我们在 Fashion-MNIST 上更简单的搜索空间和全面设置（块搜索空间，CIFAR-10)中展示实验结果。所有实验均使用 PyTorch 框架在 NVidia RTX 2080Ti GPU 上运行。

1) 结果 小规模设置:简单搜索空间和 Fashion-MNIST 任务的组合旨在作为开发和概念验证设置。然而,我们获得了有趣的结果,这些结果显示了我们的算法的搜索空间不可知性。

表 IV 报告了五次运行的结果,并与代表性作品进行了比较（注意数量

表五
CIFAR-10 的实验结果,主要范式。

方法	举报了	测试精度 % GPU-days	
手工制作			
ResNet-110 [2]	最佳（平均值±标准差)93.57 (93.39±0.16)	-	
微型非进化 XNAS-Large [10]			
p,r,h,o NASNet-A 28M [22] p,r	最好	98.40	0.3
BANANAS [7]	最好	97.60	1800
H	最好	97.36	12
NASNet-A 3.3M [22] p,r DARTS	最佳	97.35	1800
二阶 [9] p,r,h,o ENAS Micro +	μ ± σ	97.17±0.06	4
Cutout h,o	最好	97.11	0.5
微型、进化的 NSGANetV1-A4 [38]	r 最佳	97.98	27
AmoebaNet-B 34M [15] p,r	μ ± σ	97.87±0.04	4500（热塑性聚氨酯膜）
AmoebaNet-B 2.8M [15] p,r柠檬水	μ ± σ	97.45±0.05	4500 (TPU)
II [39]	最好	96.60	56 56
DPP-Net [40] p,h NES-	最好	95.64	
RE 合奏 30,无偏移 [19]	n 平均值±CI	90.8±0.1	北美
宏观的、非进化的			
NAS 深度 39,额外过滤器 [4] p	最好	96.35	22400
ENAS 宏 [8]	最好	95.77	0.3 2
NASH 快照集成 [36]	意思	95.30	1
NASH 单一模型 [36]	意思	94.80	
NAS 深度 15 [4] p	最好	94.50	22400
宏观、进化的 CNN-GA Cutout [41]	最好	96.78	35
RandGrow No-DropPath [23] r,h柠檬水 I	意思	96.62	6
[39]	意思	96.50	56
大型 Evo.合奏 [5]	最好	95.60	3000
CGP-CNN [16]	H 最佳 (μ ± σ)	94.99 (93.90±0.89)	30
大型 Evo. [5]	最佳 (μ ± σ)	94.60 (94.10±0.40)	3000
免疫力（我们的）	最佳 (μ ± σ)	95.62 (94.97±0.50)	14

p 重要的后处理

r 复杂的手工设计初始架构

r 强增强/正则化

o 具有权重共享的一次性搜索

n 无增强

报告 Fashion-MNIST 结果的论文相当有限）。

XNAS [10] 优于我们的方法,但其架构是基于 CIFAR-10 的基于单元的搜索的结果,其最终训练过程涉及辅助塔和 AutoAugment [30],其时间要求可能相当可观。在其他方法中,就可以比较涉及不同预处理和训练协议的结果而言，ImmuNES 在搜索时间相似的情况下具有很强的竞争力。与最佳单个模型相比,集成的增益约为 0.8%pt。

所有 12 个模型的最终再训练大约需要 9 个 GPU 小时。

2) 结果 全面设置:我们使用块搜索空间在 CIFAR-10 任务上完成了三轮 ImmunES。表 V 将我们的结果与 NAS 中有影响力和多样化的作品进行了比较。训练程序、增强和后处理方法以及搜索空间定义的多样性使得直接比较变得困难。为了尽量清晰,我们将工作分为两个轴:微观/宏观搜索和进化/其他算法。对于每种方法,我们还描述了手动后处理的数量（例如元参数搜索、折痕中的深度或宽度）、增强/正则化（例如下降路径、辅助塔、AutoAugment)以及起始架构的复杂性。似乎出现了两种趋势:首先,准确性与人工干预的数量和所涉及的专业知识密切相关。微搜索通常优于宏搜索,这可能是因为它们具有大型手工设计的宏架构 [23]。其次,一次性 NAS（XNAS、ENAS、DARTS)比进化 NAS 更快（参见第 IA 节中的关注点）。

我们的方法使用简单的搜索空间定义、无后处理、有限正则化和随机 ini

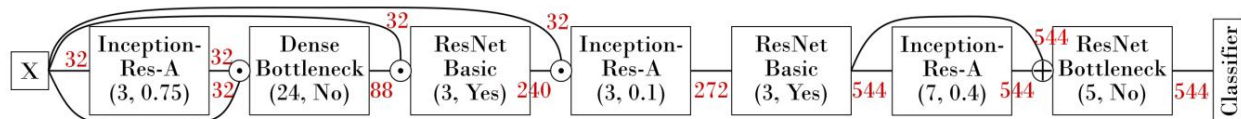


图 4. CIFAR-10 上的最佳个体架构 (94.5%pt 测试精度), 每个节点具有 (HP 1, HP 2) 值。连接, 外面的框代表通道数。

: 添加。数字

临时架构。我们相信这些选择对于清晰度和可访问性很重要,即使它们牺牲了一些性能。尽管如此,我们实现了合理的性能/效率平衡,特别是在进化宏观搜索方法中。值得注意的是,我们的 GPU 天数比 Large Scale Evolution [5] 少两个数量级。

CNN-GA 在没有复杂的手动技巧的情况下实现了略高的精度,但计算成本是其两倍。

最终的再训练 (5 种架构) 大约需要 1 个 GPU 日,每个样本的推理时间为 2 毫秒。集成始终如一地带来 1.1-1.5%pt 的分类精度增益,根据对学习曲线的观察,这只有至少再增加 8-10 个 GPU 天才有可能实现。

图4显示了ImmuNES找到的最佳架构,其单项测试准确率为94.50%。这种架构明显比流行的人工设计模型浅。

有趣的是,它在早期层的块之间使用跳跃连接和串联来快速增加容量,然后进入更线性的结构。

C. 与基于随机搜索的 NES 的比较最近的论文 [11]-[13]

表明,在精心设计的搜索空间上的随机搜索 (RS) 可能与一些流行的 NAS 方法类似。与 [19] 一样,我们在较小规模的环境中将 ImmuNES 与 RS NES 进行对比,生成随机网络架构,然后像在 ImmuNES 中一样训练和集成它们。我们以与 ImmuNES 相同的评估预算运行 RS 五次。我们密切关注层数的采样。ImmuNES 从 3 层开始,其最佳模型有 8 层。为了对这两种方法公平,范围不应太宽 (这会惩罚 RS),RS 也不应直接受益于 ImmuNES 发现的最佳模型深度。因此,我们让 RS 从范围 {3...12}。RS 仅达到 $93.37 \pm 0.35\%$ 的集成测试精度 (最佳:93.79%),明显低于我们的 AIS (单尾 $p < 0.001$)。它最好的模型有 10 层,而 ImmuNES 有 8 层。

D. 与遗传算法的比较

在 [5] 中,Real等人。实现没有交叉的 GA。与 CLONALG 不同,它从均匀分布中采样突变强度,并通过成对锦标赛选择来选择候选者。它们包括 GA 发现的一组网络的结果,在 CIFAR-10 上的准确度提高了 1%。

我们采用我们的 ImmuNES 框架,并用这个 GA 简单地替换 AIS 搜索算法。我们在较小规模的环境中以相同的评估预算运行了五次。最终集成的平均测试精度为 $93.95 \pm 0.4\%$,最佳

结果为 94.41%,显著低于 ImmuNES (单尾 $p = 0.039$)。虽然最佳单个模型的平均准确度非常接近 (ImmuNES 为 93.39% vs. GA 为 93.31%),但 ImmuNES 中集成的增益更高 (0.8%pt vs. 0.6%pt),表明 AIS 更好地促进多样性。

E. 从 CIFAR-10 到 CINIC-10 的可迁移性

Micro-NAS 解决方案可以通过手动增加单元数量或其通道数来转移到更艰巨的任务。

然而,在 macro-NAS 中,手动添加层不是一种选择,当通道数由算法确定时 (如 [5],[16] 中),手动增加它也没有意义。因此,很少评估此类方法的可迁移性。

然而,我们假设 ImmuNES 解决方案的多重性质将帮助他们在不进行任何修改的情况下转移到更艰巨的任务。为了进一步推动生物学类比,这种能力可以被视为抗体群体对相关但不同的抗原的二次反应。

为了直接重用这些架构,我们选择了 CINIC-10 [42],它是 CIFAR-103 的直接替代品,但难度要大得多 (参见引用的论文)。我们采用最佳集成,并使用完全相同的训练程序在 CINIC-10 上对其进行训练,包括 epoch 的数量。我们获得了 88.72% 的测试精度,并从集成中获得了 1.36%pt 的增益。请注意,更艰巨的任务将证明集成中有更多的增强、训练或网络。就目前而言,该结果排名在 VGG-16 [43] (87.77%) 和 ResNet-18 [2] (90.27%) 这两个流行的手工制作模型之间。重要的是,这两个任务的架构排名完全不同。因此,重复使用在 CIFAR-10 上找到的最佳模型会导致更糟糕的结果:“抗体”的数量提高了解决方案的可迁移性。

五. 结论

通过 ImmuNES,我们证明了 AIS 在提供相关效率改进的情况下能够在宏观架构级别胜任执行 NES。它能够在两个截然不同的搜索空间上获得有竞争力的结果,而无需复杂的预处理和后处理工程或培训技术。我们还表明,在集成方面,AIS 比两个基线带来了显著的好处。最后,与大多数宏架构 NAS 方法不同,ImmuNES 解决方案的多重性使它们能够自然地转移到更艰巨的任务。尽管我们的实验没有达到 CIFAR-10 的 SOTA,但我们相信这种方法是

“如果需要不同数量的类,我们只需要修改每个分类器块中的全连接层。

有望成为微架构搜索方法的未来替代方案,并且没有许多缺点。未来的研究方向包括通过实施性能预测来指导搜索来进一步提高我们方法的效率,并探索更先进的 AIS 算法。

参考

[1] LN de Castro 和 FJ Von Zuben,“使用克隆选择原理进行学习和优化”, IEEE 进化计算汇刊,卷。6,没有。3,第 239–251 页,2002 年。

[2] K. He,X. Zhang,S. Ren 和 J. Sun,“用于图像识别的深度残差学习”,载于 IEEE 计算机视觉和模式识别会议论文集, 2016 年,第 770–778 页。

[3] G. Huang,Z. Liu,LVD Maaten 和 KQ Weinberger,“密集连接的卷积网络”, 2017 年 IEEE 计算机视觉和模式识别会议 (CVPR), 2017 年,第 2261–2269 页。

[4] B. Zoph 和 QV Le,“使用强化学习进行神经架构搜索”,国际学习表征会议, 2017 年。

[5] E. Real,S. Moore,A. Selle,S. Saxena,YL Suematsu,J. Tan,QV Le 和 A. Kurakin,“图像分类器的大规模演化”,载于第 34 届国际机器学习会议论文集,卷。70. JMLR.org,2017 年,第 2902–2911 页。

[6] J. Wang,J. Xu 和 X. Wang,“深度学习中超参数优化的超带和贝叶斯优化组合”, arXiv 预印本 arXiv:1801.01596,2018年。

[7] C. White,W. Neiswanger 和 Y. Savani,“香蕉:用于神经架构搜索的神经架构贝叶斯优化”, arXiv 预印本 arXiv:1910.11858,2019。

[8] H. Pham,M. Guan,B. Zoph,Q. Le 和 J. Dean,“通过参数共享进行高效神经架构搜索”,机器学习国际会议, 2018 年,第 4092–4101 页。

[9] H. Liu,K. Simonyan 和 Y. Yang,“DARTS:可微架构搜索”, arXiv 预印本 arXiv:1806.09055,2018年。

[10] N. Nayman,A. Noy,T. Ridnik,I. Friedman,R. Jin 和 L. Zelnik,“XNAS:神经架构搜索与专家建议”,神经信息处理系统进展, 2019 年,pp. 1977–1987。

[11] L. Li 和 A. Talwalkar,“神经网络的随机搜索和再现性架构搜索”, arXiv 预印本 arXiv:1902.07638,2019年。

[12] C. Sciuto,K. Yu,M. Jaggi,C. Musat 和 M. Salzmann,“评估神经架构搜索的搜索阶段”, arXiv 预印本 arXiv:1902.08142,2019。

[13] A. Yang,PM Esperanc,a 和 FM Carlucci,“NAS 评估非常困难”, arXiv 预印本 arXiv:1912.12522,2019年。

[14] L. Xie 和 A. Yuille,“Genetic CNN”, arXiv 预印本 arXiv:1703.01513, 2017。

[15] E. Real,A. Aggarwal,Y. Huang 和 QV Le,“图像分类器架构搜索的正则化演化”,载于 AAAI 人工智能会议论文集,卷。33, 2019, 第 4780–4789 页。

[16] M. Suganuma,M. Kobayashi,S. Shirakawa 和 T. Nagao,“使用笛卡尔遗传程序的深度卷积神经网络的进化”,进化计算,卷。28,没有。1,第 141–163 页,2020 年。

[17] E. Byla 和 W. Pang,“DeepSwarm:使用群体智能优化卷积神经网络”, arXiv 预印本 arXiv:1905.07350,2019年。

[18] V. Macko,C. Weill,H. Mazzawi 和 J. Gonzalvo,“通过集成学习改进神经架构搜索图像分类器”, arXiv 预印本 arXiv:1903.06236,2019年。

[19] S. Zaidi,A. Zela,T. Elsken,C. Holmes,F. Hutter 和 YW Teh,“神经集成搜索性能和校准预测”, arXiv 预印本 arXiv:2006.08573,2020年。

[20] LN de Castro 和 J. Timmis,“用于多模式功能优化的人工免疫网络”,进化计算,2002 年,CEC 02。2002 年国会论文集,卷。1, 2002, 第 699–704 页。

[21] R. Pasti,LN de Castro,GP Coelho 和 FJV Zuben,“神经网络集成:组件多样性的免疫启发方法”,自然计算,卷。9,没有。3,第 625–653 页,2010 年 9 月 1 日,iD: Pasti2010。[在线的]。可用 <https://doi.org/10.1007/s11047-009-9124-1>

[22] B. Zoph,V. Vasudevan,J. Shlens 和 QV Le,“学习可扩展图像识别的可迁移架构”, IEEE 计算机视觉和模式识别会议论文集, 2018 年,第 8697–8710 页。

[23] H. Hu,J. Langford,R. Caruana,E. Horvitz 和 D. Dey,“重新审视宏观神经架构搜索”, 2018 年NeurIPS 元学习第二届研讨会。

[24] DC Ciresan,U. Meier,LM Gambardella 和 J. Schmidhuber,“用于手写字符分类的卷积神经网络委员会”,国际文档分析和识别会议 (ICDAR),2011 年, 2011 年,第 1135–1139 页。

[25] E. Bochinski,T. Senst 和 T. Sikora,“基于进化算法的卷积神经网络委员会的超参数优化”,第 24 届 IEEE 图像处理国际会议论文集, 2018 年,第 3924–3928。

[26] C. Szegedy,S. Ioffe,V. Vanhoucke 和 AA Alemi,“Inception-v4,Inception-ResNet 和残差连接对学习的影响”。在AAAI 中,卷。4, 2017, p. 12。

[27] H. Xiao,K. Rasul 和 R. Vollgraf,“Fashion-MNIST:用于基准机器学习算法的新型图像数据集”, arXiv 预印本 arXiv:1708.07747,2017年。

[28] A. Krizhevsky 和 G. Hinton,“从微小图像中学习多层特征”,技术报告,多伦多大学, 卷。1,没有。4, 2009。

[29] K. He,X. Zhang,S. Ren 和 J. Sun,“深入研究整流器:在 ImageNet 分类上超越人类水平的性能”, IEEE 计算机视觉国际会议论文集, 2015 年,第. 1026–1034。

[30] ED Cubuk,B. Zoph,D. Mane,V. Vasudevan 和 QV Le,“Au toaugment:从数据中学习增强策略”, arXiv 预印本 arXiv:1805.09501,2018年。

[31] T. DeVries 和 GW Taylor,“使用 Cutout 改进卷积神经网络的正则化”, arXiv 预印本 arXiv:1708.04552,2017年。

[32] I. Loshchilov 和 F. Hutter,“SGDR:带热重启的随机梯度下降”, arXiv 预印本 arXiv:1608.03983,2016年。

[33] G. Larsson,M. Maire 和 G. Shakhnarovich,“Fractalnet:无残差的超深度神经网络”, arXiv 预印本 arXiv:1605.07648,2016年。

[34] C. Szegedy,W. Liu,Y. Jia,P. Sermanet,S. Reed,D. Anguelov,D. Erhan,V. Vanhoucke 和 A. Rabinovich,“深入了解卷积”,发表于IEEE 计算机视觉和模式识别会议, 2015 年,第 1–9 页。

[35] H. Jin,Q. Song 和 X. Hu,“Auto-Keras:一种高效的神经架构搜索系统”,第 25 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。

ACM,2019 年,页数 1946–1956。

[36] T. Elsken,JH Metzen 和 F. Hutter,“简单高效的卷积神经网络架构搜索”, arXiv 预印本 arXiv:1711.04528,2017年。

[37] N. Mitschke, M. Heizmann, K.-H. Noffz 和 R. Wittmann,“基于梯度的进化以优化卷积神经网络的结构”,第 25 届 IEEE 图像处理国际会议 (ICIP), 2018 年,第 3438–3442 页。

[38] Z. Lu,J. Whalen,Y. Dhebar,K. Deb,E. Goodman,W. Banzhaf 和 VN Boddeti,“用于图像分类的深度卷积神经网络的多目标进化设计”, IEEE 汇刊关于进化计算,第 1–1 页,2020 年。

[39] T. Elsken,JH Metzen 和 F. Hutter,“通过拉马克进化进行高效多目标神经架构搜索”, arXiv 预印本 arXiv:1804.09081,2018年。

[40] J.-D.董 A.-C. Cheng, D.-C. Juan,W. Wei 和 M. Sun,“DPP-Net:帕累托最优神经架构的设备感知渐进式搜索”,欧洲计算机视觉会议 (ECCV) 会议记录, 2018 年 9 月。

[41] Y. Sun,B. Xue,M. Zhang,GG Yen 和 J. Lv,“使用图像分类的遗传算法自动设计 CNN 架构”, IEEE 控制论汇刊,卷。50,没有。9,第 3840–3854 页,2020 年。

[42] LN Darlow,EJ Crowley,A. Antoniou 和 AJ Storkey,“CINIC-10 不是 ImageNet 或 CIFAR-10”, arXiv 预印本 arXiv:1810.03505,2018年。

[43] K. Simonyan 和 A. Zisserman,“用于大规模图像识别的超深卷积网络”, arXiv 预印本 arXiv:1409.1556,2014年。