FBNetV3:使用预测器预训练的联合架构-食谱搜索

Xiaolian Dai1 , Alvin Wan2 *, Peizhao Zhang1 , Bichen Wu1, Zijian He1 , Zhen Wei3,
Kan Chen1, Yuandong Tian1 , Matthew Yu1, Peter Vajda1, and Joseph E. Gonzalez2
1Facebook Inc., 2UC Berkeley, 3UNC Chapel Hill

{xiaoliangdai,stzpz,wbc,zijian,kanchen18,yuandong,mattcyu,vajdap}@fb.com
{alvinwan,jegonzal}@berkeley.edu, zhenni@cs.unc.edu

抽象的

神经架构搜索 (NAS) 产生了最先进的神经网络,其性能优于其最佳手动设计的对应网络。然而,以前的 NAS 方法在一组训练超参数（即训练配方)下搜索架构,忽略了高级架构-配方组合。为了解决这个问题,我们提出了神经架构-配方搜索 (NARS) 来同时搜索 (a) 架构和 (b) 它们相应的训练配方。 NARS 利用一个准确度预测器对架构和训练方法进行联合评分,指导样本选择和排名。此外,为了补偿扩大的搜索空间,我们利用“免费”架构统计信息（例如,FLOP 计数)来预训练预测器,显着提高其样本效率和预测可靠性。通过约束迭代优化训练预测器后,我们在短短 CPU 分钟内运行快速进化搜索,为各种资源约束生成架构配方对,称为 FBNetV3。 FBNetV3 构成了一系列最先进的紧凑型神经网络,其性能优于自动和手动设计的竞争对手。例如,FB NetV3 在 ImageNet 上与 EfficientNet 和 ResNeSt 精度相匹配， FLOPs 分别减少了 2.0和 7.1 。此外,FBNetV3 为下游目标检测任务带来了显着的性能提升,提高了 mAP,尽管与基于 EfficientNet 的同等任务相比， FLOP 减少了 18%,参数减少了 34% 。

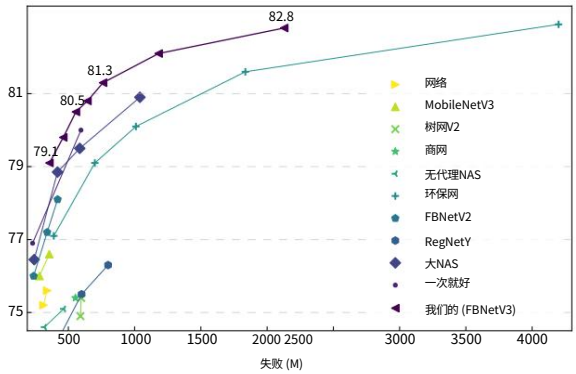

图 1： FBNetV3与其他高效卷积神经网络的 ImageNet 精度与模型 FLOPs 比较。 FB NetV3 以 557M (2.1G) 达到 80.8% (82.8%) top-1 准确率

一、简介

设计高效的计算机视觉模型是一个具有挑战性但很重要的问题:从自动驾驶汽车到增强现实的无数应用都需要紧凑的模型,这些模型必须高度准确 即使在约束条件下也是如此

FLOPs,为准确率-效率权衡设定了新的 SOTA。

关于功率、计算、内存和延迟。可能的约束和架构组合的数量非常大,使得手动设计几乎不可能。

作为回应,最近的工作采用神经架构搜索 (NAS) 来设计最先进的高效深度神经网络。一类 NAS 是可微分神经结构搜索 (DNAS)。这些寻路算法效率很高,通常可以在训练一个网络所需的时间内完成搜索。然而, DNAS 无法搜索对模型性能至关重要的非架构超参数。此外,基于超网的 NAS 方法搜索空间有限,因为整个超图必须适合内存以避免缓慢收敛[5] 或分页。其他方法包括强化学习 (RL) [45]和进化算法 (ENAS) [41]。

然而,这些方法都有几个缺点:

1.忽略训练超参数： NAS,顾名思义,只搜索架构而不是相关的训练超参数（即“训练配方”）。

这忽略了一个事实,即不同的训练方法可能

*平等贡献

978-1-6654-4509-2/21/31.00 美元 ©2021 IEEE
DOI 10.1109/CVPR46437.2021.01601

模型 \ 训练	食谱 1	食谱 2
ResNet18（1.4 倍宽度）	70.8%	73.3%
ResNet18（2 倍深度）	70.7%	73.8%

表 1:不同的训练方法可以改变架构的排名。 ResNet18 1.4x 宽度和 2x 深度分别指的是具有 1.4 宽度和 2.0 深度比例因子的 ResNet18。培训配方的详细信息可以在附录A.1 中找到。

彻底改变架构的成败,甚至改变架构排名（表1）。

2.仅支持一次性使用:许多传统的 NAS 方法为一组特定的资源约束生成一个模型。这意味着部署到一系列产品,每个产品都有不同的资源限制,需要为每个资源设置重新运行一次 NAS。

或者,模型设计者可以搜索一个模型并使用手动试探法对其进行次优缩放,以适应新的资源限制。

3.过大的搜索空间进行搜索:天真地在搜索空间中包含训练方法要么是不可能的（ DNAS ,基于超网的 NAS），要么是非常昂贵的,因为仅架构的准确性预测器在计算上已经很昂贵了（RL,ENAS） .

为了克服这些挑战,我们提出了神经架构-食谱搜索 (NARS) 来解决上述限制。我们的见解分为三个方面:(1) 为了支持NAS 结果在多个资源约束下的重用,我们训练了一个准确度预测器,然后使用该预测器在 CPU 分钟内为新的资源约束找到架构配方对。

(2) 为了避免仅架构或仅配方搜索的陷阱,该预测器同时对训练配方和架构进行评分。 (3) 为了避免预测器训练时间的过度增长,我们在代理数据集上预训练预测器以从架构表示中预测架构统计信息（例如,FLOPs、#Parameters ）。在依次执行预测器预训练、约束迭代优化和基于预测器的进化搜索之后， NARS 生成了可泛化的训练方法和紧凑模型,这些模型在 ImageNet 上达到了最先进的性能,优于所有现有的手动设计或自动搜索的模型神经网络。我们在下面总结了我们的贡献:

1.神经架构-配方搜索:我们提出了一个预测器,它对训练配方和架构进行联合评分,这是第一次联合搜索,根据我们的知识规模对训练配方和架构进行联合搜索。

2.预测器预训练:为了在这个更大的空间中进行有效搜索,我们进一步提出了一个预训练

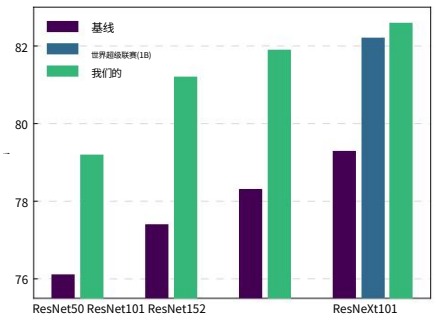

图 2:使用搜索到的训练方法提高现有架构的准确性。 WSL指的是使用1B附加图像的弱监督学习模型[33]。

技术,显着提高了精度预测器的采样效率。

3.多用途预测器:我们的预测器可用于快速进化搜索,以在短短 CPU 分钟内快速生成各种资源预算的模型。

4.对于搜索到的 FBNetV3 模型,每个 FLOP 的ImageNet 精度是最先进的。例如,我们的 FB NetV3 与 EfficientNet 精度相匹配，FLOPs 低至49.3%,如图1 所示。

5.可推广的训练配方： NARS 的仅配方搜索在各种神经网络中实现了显着的精度提升,如图 2 所示。我们的 ResNeXt101-32x8d 达到了 82.6% 的 top-1 精度;这甚至优于其在 1B 额外图像上训练的弱监督对手[33]。

二、相关工作

紧凑型神经网络的工作始于手动设计,可分为架构修改和非架构修改。

手动架构设计:大多数早期工作压缩现有架构。一种方法是修剪[12、7、60、4] ,其中根据某些启发式方法删除层或通道。然而,修剪要么只考虑一种架构[13] ,要么只能顺序搜索越来越小的架构[58]。这限制了搜索空间。

其他工作使用成本友好的新操作从头开始设计新架构。这包括卷积变体,例如MobileNet 中的深度卷积； MobileNetV2 中的反转残差块； MobileNetV3 [18、42、17] 中的 hswish 等激活;以及 shift [52] 和 shuffle [32] 等操作。尽管其中许多仍在最先进的神经网络中使用，但手动设计的架构已被自动搜索的对应架构所取代。

非架构修改:许多网络压缩技术包括低位量化[12]

少至两个[65]甚至一位[21]。其他工作不均匀地对输入进行下采样[53、57、34]以降低计算成本。这些方法可以与架构改进相结合,以粗略地减少延迟。其他非架构修改涉及超参数调优,包括前深度学习时代的调优库[2]。几个深度学习特定的调整库也被广泛使用[26]。一种较新的方法类别会自动搜索数据增强策略的最佳组合。这些方法使用策略搜索[6]、基于人口的训练[16]、基于贝叶斯的增强[47]或贝叶斯优化[22]。

自动架构搜索： NAS 自动化神经网络设计以获得最先进的性能。NAS 的几种最常见技术包括强化学习[66、45] 、进化算法[41、40、59]和 DNAS [30、51、48、11、56]。 DNAS 使用很少的计算资源进行快速训练,但由于内存限制而受到搜索空间大小的限制。有几项工作试图通过一次只训练子集[5]或引入近似值[48] 来解决这个问题。然而,它的灵活性仍然不及竞争对手的强化学习方法和进化算法。反过来,这些先前的工作仅搜索模型架构[29、50、49、43、4] 或对小规模数据集（例如CIFAR ） [1、62] 执行神经架构-配方搜索。相比之下,我们的 NARS在 ImageNet 上联合搜索架构和训练方法。为了补偿更大的搜索空间,我们 (a) 引入了预测器预训练技术以提高预测器的收敛速度,并且 (b) 采用基于预测器的进化搜索来在 CPU 分钟内设计架构配方对,适用于任何资源约束设置在进化搜索之前显着优于预测器排名最高的候选日期。我们还注意到先前的工作在一次搜索后生成了一组成本可以忽略不计或没有成本的模型[11、59、31]。

算法 1:三阶段约束感知神经架构-食谱搜索输入： :设计的搜索空间; n:约束迭代优化中候选池的大小； m:每次迭代中要训练的 DNN 候选者 (X) 的数量； T:约束迭代优化的批次数;第 1 阶段:预训练预测器生成一个包含 n 个样本的池 ⇒ 从搜索空间 进行 QMC 采样;使用架构统计预训练精度预测器 u;第 2 阶段:训练预测器（约束迭代优化）:将 D0初始化为 ;;对于t = 1, 2, ..., T do根据预测分数 u(x) 找到一批最有希望的 DNN 候选者 X ;通过并行训练评估所有x 2 X;如果t = 1:确定提前停止标准;更新数据集： Dt = Dt1 [{(x1, acc(x1)), (x2, acc(x2)), ...};在 Dt上重新训练准确度预测器 u ;第三阶段结束:使用预测器（Predictor-Based Evolutionary Search）

初始化D?在DT中有 p 个表现最好的样本和 q 个随机生成的样本
与 u 预测的分数配对； = 0;设置 = 106;初始化s?在D 中获得最高分?
0) > x 2 D ? do生成一组受资源约束的孩子C 设置s自适应进化算法
[8];
结束
增强D? C与u预测的分数配对;从增广集中选出前K个候选更新
D?;将之前的最佳排名分数更新为s? = s?;更新当前最佳排名分数s?
通过D?中的最佳预测分数。
0
end
Result: D?,即所有前K个best samples及其预测值
分数。

三、方法

我们的目标是找到最准确的架构和训练配方组合,以避免像以前的方法那样忽视架构配方对。然而,搜索空间通常组合很大,不可能进行详尽的评估。为了解决这个问题,我们训练了一个接受架构和训练配方表示的准确性预测器（第3.1 节）。为此,我们采用三阶段流水线（算法1）:(1) 使用架构统计预训练预测器,显着提高其准确性和样本效率（第3.2 节）。 (2) 使用约束迭代优化训练预测器（第3.3 节）。 (3) 对于每组资源约束,仅在 CPU 分钟内运行基于预测器的进化搜索以生成高精度架构-配方对（第3.4 节）。

3.1.预测器

我们的预测器旨在预测给定架构和训练方法表示的准确性。使用单热分类变量（例如,用于块类型)和最小-最大归一化连续值（例如,用于通道计数)对体系结构和训练方法进行编码。请参阅表2中的完整搜索空间。

预测器架构是一个多层感知器（图3） ,由几个完全连接的层和两个头组成：(1)辅助“代理”头,用于对编码器进行预训练,预测架构统计信息（例如,FLOPs 和#参数)来自架构表示； (2)精度头,在约束迭代优化（第3.3 节)中进行微调,从架构和训练方法的联合表示中预测精度。

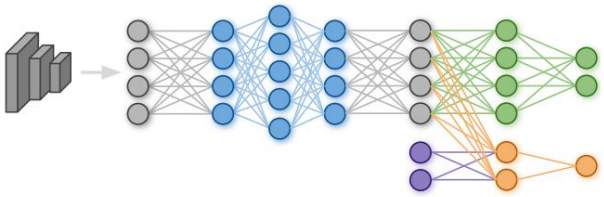

图 3:预测架构统计数据的预训练（上）。训练以预测架构-配方对的准确性（底部）

3.2.第一阶段:预测器预训练

训练准确度预测器的计算成本可能很高,因为每个训练标签表面上都是在特定训练配方下经过全面训练的架构。为了缓解这种情况，我们的见解是首先对代理任务进行预训练。预训练步骤可以帮助预测器形成输入的良好内部表示,从而减少所需的准确性架构配方样本的数量。这可以显着降低所需的搜索成本。

为了构建用于预训练的代理任务,我们可以使用架构标签的“免费”来源:即架构统计信息,如 FLOP 和参数数量。在这个预训练步骤之后,我们转移预训练的嵌入层来初始化准确度预测器（图3）。这导致最终预测器的样本效率和预测可靠性显着提高。例如,为了达到相同的预测均方误差 (MSE),经过预训练的预测器只需要比没有预训练的预测器少5的样本,如图4(e) 所示。结果,预测器预训练大大降低了整体搜索成本。

3.3.第 2 阶段:训练预测器

在这一步中,我们训练预测器并生成一组有希望的候选者。如前所述,我们的目标是在给定的资源限制下找到最准确的架构和训练方法组合。因此,我们将架构搜索模拟为一个约束优化问题：

$$\max_{(A,h)2} \ \text{acc}(A, h), \text{ st } g_i(A) \ 6 \ C_i, \ i = 1, \ldots, \tag{1}$$

其中A、 h 和分别指神经网络架构、训练方法和设计的搜索空间。 acc 将架构和训练方法映射到准确度。 gi(A)是指计算成本、存储成本、运行时延迟等资源约束的公式和计数。

约束迭代优化:我们首先使用准蒙特卡洛 (QMC) [37]采样从搜索空间生成架构-配方对的样本池。然后，

我们迭代地训练预测器:我们（a)通过根据预测的准确性选择一个有利候选的子集来缩小候选空间，(b)使用早期停止启发式训练和评估 can didates,以及（c)微调具有 Huber 损失的预测变量。这种候选空间的迭代缩小避免了不必要的评估并提高了探索效率。

·提前停止培训候选人。我们引入了一种提前停止机制来减少评估候选人的计算成本。具体来说,我们（a)在约束迭代优化的第一次迭代之后通过早期停止和最终准确度对样本进行排名，(b)计算排名相关性,以及（c)找到相关性超过特定阈值的时期e （例如，0.92),如图5 所示。

对于所有剩余的候选者,我们仅训练(A, h) e个时期来近似acc(A, h)。这使我们能够使用更少的训练迭代来评估每个查询样本。

·使用Huber 损失训练预测器。在获得预训练的架构嵌入后,我们首先训练预测器 50 个 epochs,嵌入层冻结。然后,我们以降低的学习率训练整个模型另外 50 个时期。我们采用 Huber 损失来训练准确度预测器,即L = 0.5(yy^)2 if |y y^| < 1其他|y y^| 0.5,其中y 和y ^ 分别是预测和真实标签。这可以防止模型被异常值支配,这表明可能会混淆预测器[50]。

3.4.第 3 阶段:使用预测器

所提出方法的第三阶段是基于自适应遗传算法的迭代过程[44]。第二阶段表现最好的架构-配方对作为第一代候选的一部分被继承。在每次迭代中,我们向候选者引入突变并生成一组受给定约束约束的孩子C 。

我们使用预训练的准确度预测器u评估每个孩子的分数,并为下一代选择得分最高的前K个孩子。我们在每次迭代后计算最高分的增益,并在改进饱和时终止循环。最后,基于预测器的进化搜索产生高精度的神经网络架构和训练方法。

请注意,使用精度预测器,搜索网络以适应不同的使用场景只会产生可忽略不计的成本。这是因为准确度预测器可以在不同的资源约束下大量重复使用,而基于预测器的进化搜索只需 CPU 分钟。

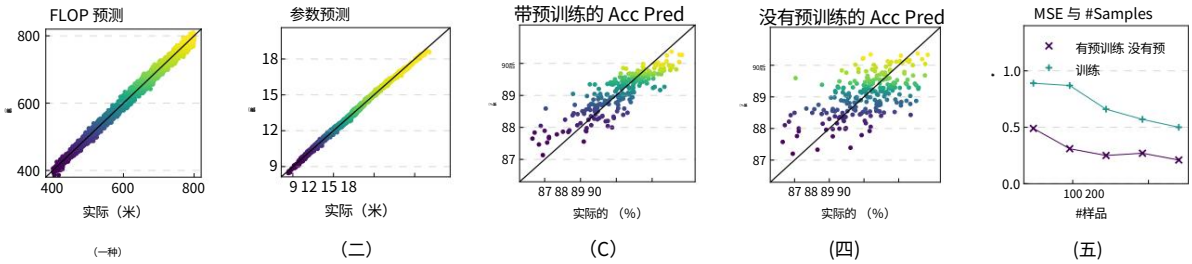

图 4： (a) 和 (b):预测器在代理指标上的表现,(c) 和 (d):预测器在使用和不使用预训练时的准确性性能,(e):预测器的 MSE 与使用和不使用预训练的样本数.

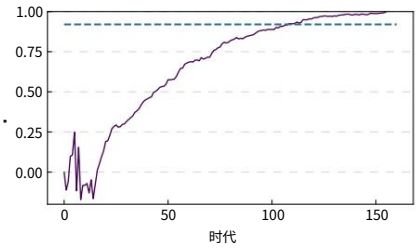

图 5:排名相关性与时代。相关阈值（青色)为 0.92。

3.5.预测搜索空间

我们的搜索空间包括训练方法和架构配置。训练配方的搜索空间具有优化器类型、初始学习率、权重衰减、混合率[63]、退出率、随机深度下降率[20]以及是否使用模型指数移动平均线 (EMA) [23] 。我们的架构配置搜索空间基于倒置残差块[42] ，包括输入分辨率、内核大小、扩展、每层通道数和深度,详见表2。

在 recipe-only 实验中,我们只在固定架构上调整训练 recipe。然而,对于联合搜索,我们在表2的搜索空间内同时搜索训练方法和架构。总体而言,该空间包含1017个候选架构和107 个可能的训练方法。在如此广阔的搜索空间中探索最佳网络架构及其相应的训练方法并非易事。

4. 实验

在本节中,我们首先在缩小的搜索空间中验证我们的搜索方法,以发现给定网络的训练方法。然后,我们评估我们的搜索方法以联合搜索架构和训练方法。我们使用PyTorch [38] ,并在 ImageNet 2012 分类数据集 [9] 上进行搜索。在搜索过程中,我们从整个数据集中随机抽取 200 个类以减少训练时间。然后,我们从 200 类训练集中随机保留 10K 个图像作为验证集。

4.1.仅搜索食谱

为了确定即使是现代 NAS 生成的架构的性能也可以通过更好的训练配方进一步提高,我们优化了固定架构的训练配方。我们采用 FBNetV2-L3 [48] （附录A.2)作为我们的基础架构,这是一个 DNAS 搜索架构,使用[48]中使用的原始训练方法达到 79.1% top-1 准确率。我们在约束迭代优化中设置样本池大小n = 20K,批量大小m = 48和迭代T = 4 。我们在搜索过程中以每个时期0.963 的学习率衰减因子训练 150 个时期的样本候选者,并以3较慢的学习率衰减（即每个时期 0.9875)训练最终模型。我们在图 6 中展示了每一轮的样本分布以及我们实验中的最终搜索结果,其中第一轮样本是随机生成的。搜索到的训练方法（附录A.3)将我们的基础架构的准确性提高了0.8%。

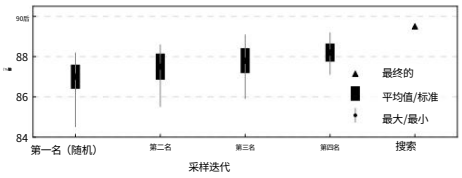

图 6:采样和搜索过程的图示。

我们将 NARS 搜索的训练方法扩展到其他常用的神经网络,以进一步验证其通用性。尽管 NARS 搜索的训练方法是为 FBNetV2-L3 量身定制的,但它的泛化能力出奇地好,如表3所示。NARS 搜索的训练方法在 ImageNet 上使准确率提高了 5.7%。

事实上,ResNet50 比基准 ResNet152 高出 0.9%。 ResNeXt101-32x8d 甚至超过了弱监督学习模型,该模型使用 10 亿张弱标记图像进行训练，达到了 82.2% 的 top-1 准确率。

值得注意的是,通过为每个神经网络搜索特定的训练方法可以获得更好的性能,但这会增加搜索成本。

堵塞	k		c	n		硒	行为。
转换		-	(16, 24, 2)			-	许愿
MBConv	[3, 5]		(16, 24, 2)	(1, 4)		否	许愿
MBConv	[3, 5]	(4, 7) / (2, 5) (4,	(20, 32, 4)	(4, 7)		否	许愿
MBConv	[3, 5]	7) / (2, 5) (4,	(24, 48, 4)	(4, 7)		是	许愿
MBConv	[3, 5]	7)1 / (2, 5)2 (4,	(56, 84, 4)	(4, 8)		否	许愿
MBConv	[3, 5]	7)1 / (2, 5)2 (4 , 7)	(96, 144, 4)	(6, 10)		是	许愿
MBConv	[3, 5]		(180, 224, 4)	(5, 9)		是	许愿
MBConv	[3, 5]		(180, 224, 4)			是	许愿
MB池	[3, 5]		1984		-	-	许愿
FC	-	-	1000		-	-	-

资源	lr(103) (20,	优化	续马	p(102) (1,	d(101)	米 (101) (0,	wd(106) (7,
(224, 272, 8)	30)	[RMSProp, SGD]	[真假]	31)	(10, 31)	41)	21)

表 2:我们实验中的网络架构配置和搜索空间。 MBConv、MBPool、k、e、c、n、s、se 和 act。分别参考倒置残差块[42]、有效最后阶段[17]、内核大小、扩展、#Channel、#Layers、步幅、挤压和激发以及激活函数。 res, lr, optim, ema, p, d, m, wd分别指resolution, initial learning rate, optimizer type, EMA, dropout ratio, stochastic depth drop probability, mixup ratio, weight decay。斜线左侧的扩展用于阶段的第一个块,而右侧的扩展用于其余部分。括号内三个值的元组分别代表最低值、最高值和步长;二值元组表示步长为 1,括号中的元组表示搜索时所有可用的选择。请注意,如果优化选择 SGD ,lr 将乘以 4 。具有相同上标的架构参数在搜索过程中共享相同的值。

模型	Top-1 准确率 (%)		
	原版的	仅限食谱	
FBNetV2-L3 [48]	79.1	79.9	+0.8
亚历克斯网[25]	56.6	62.3	+5.7
ResNet34 [15]	73.3	76.3	+3.0
ResNet50 [15]	76.1	79.2	+3.1
ResNet101 [15]	77.4	81.2	+3.8
ResNet152 [15]	78.3	81.9	+3.6
DenseNet201 [19]	77.2	80.2	+3.0
ResNeXt101 [55]	79.3	82.6	+3.3

表 3:在现有神经网络上搜索到的训练方法提高了准确性。上面,ResNeXt101 指的是32x8d 变体。

4.2.神经结构-食谱搜索 (NARS)

搜索设置接下来,我们对架构和训练方法进行联合搜索,以发现紧凑的神经网络。请注意,根据我们在第二节中的观察。 4.1,我们缩小搜索空间以始终使用 EMA。大多数设置与 recipe-only search 中的相同,而我们增加了优化迭代T = 5并将样本池的 FLOPs 约束从 400M 设置为 800M。

我们使用包含 20K 个样本的80% 的样本池对架构嵌入层进行预训练,并在图4 中绘制其余 20% 的验证。在基于预测的进化搜索中,我们设置了四种不同的 FLOPs 约束： 450M、550M , 650M, 和 750M 并发现具有相同精度预测器的四个模型（即 FBNetV3-B/C/D/E）。我们进一步缩小和扩大最小值和最大模型并生成 FBNetV3-A 和 FBNetV3- F/G 以分别适应更多的使用场景,并使用[46]中提出的复合缩放。

训练设置对于模型训练,我们使用基于蒸馏的两步训练过程:(1) 我们首先使用带有地面真值标签的搜索配方训练最大的模型（即 FBNetV3-G）。 (2) 然后,我们对所有模型（包括 FBNetV3-G 本身)进行蒸馏训练,这是[4][61]中采用的典型训练技术。与[4][61]中的原地蒸馏方法不同,这里的教师模型是从步骤（1)导出的ImageNet预训练FBNetV3-G 。训练损失是两个部分的总和:按比例缩放 0.8 的蒸馏损失和按比例缩放 0.2 的交叉熵损失。

在训练期间,我们在8 个节点和每个节点 8 个 GPU的分布式训练中使用同步批量归一化。我们训练模型 400 个 epoch,在 5 个 epoch 预热后,每个 epoch 的学习率衰减因子为 0.9875。我们分别使用 FBNetV3-B 和 FBNetV3-E的搜索训练方法训练缩放模型 FBNetV3-A 和 FBNetV3-F/G ,仅将 FBNetV3-F/G 的随机深度下降比增加到 0.2。更多培训细节可以在附录A.5 中找到。

搜索模型我们将搜索模型与图 1 中的其他相关 NAS 基线和手工制作的紧凑型神经网络进行比较,并在表 4 中列出详细的性能指标比较,其中我们按模型的 top-1 精度对模型进行分组。在所有现有的高效模型中,如 EfficientNet [46]、 MobileNetV3 [17]、 ResNeSt [64]和 FBNetV2 [48],我们搜索的模型在准确性-效率权衡方面取得了实质性的改进。例如,在低计算成本制度下,

模型	搜索方法	搜索空间	搜索成本 (GPU/TPU 时数)	失败者	准确性 (%， 前5)	准确性 (%, Top-1)
FBNet [51]	梯度 RL/	拱	0.2K	375M	-	74.9
无代理NAS [5]	梯度预测器弹	拱	0.2K	465M	-	75.1
商网[8]	出。 param.⇒	拱	28K	553M	-	75.4
RegNetY [39]	RL/NetAdapt	拱	11K	600M	-	75.5
MobileNetV3-1.25x [17]	RL/scaling	拱	>91K	356M	-	76.6
高效网络 B0 [46]	gradient	拱	>91K	390M	93.3	77.3
原子NAS [35]	gradient NARS	拱	0.8K	363M	-	77.6
FBNetV2-L2 [48]		拱	0.6K	423M	-	78.1
FBNetV3-A		拱门/食谱	10.7K	357M	94.5	79.1
ResNet152 [15]	手动的	-	-	11G	93.8	78.3
高效网络 B2 [46]	强化学习/缩放	拱	>91K	1.0G	94.9	80.3
ResNeXt101-32x8d [55]	手动的	-	-	7.8G	94.5	79.3
一劳永逸[4]	坡度	-	-	595M	-	80.0
FBNetV3-C	纳斯	拱门/食谱	10.7K	557M	95.1	80.5
BigNASModel-XL [61]	渐变手册	拱	2.3K	1.0G	-	80.9
ResNeSt-50 [64]		-	-	5.4G	-	81.1
FBNetV3-E	纳斯	拱门/食谱	10.7K	762M	95.5	81.3
高效网络 B3 [46]	强化学习/缩放	拱	>91K	1.8G	95.7	81.7
ResNeSt-101 [64]	手动的	-	-	10.2G	-	82.3
高效网络 B4 [46]	强化学习/缩放	拱	>91K	4.2G	96.4	82.9
FBNetV3-G	纳斯	拱门/食谱	10.7K	2.1G	96.3	82.8

表 4:不同紧凑型神经网络的比较。对于基线,我们引用了原始论文中 ImageNet 的统计数据。我们的结果以粗体显示。 *:人口参数化。有关训练技巧和其他 EfficientNet 比较的讨论,请参阅A.6 。

FBNetV3-A 仅用 357M FLOPs 就达到了 79.1% 的 top-1 精度（比具有类似 FLOPs 的 MobileNetV3-1.25x [17]精度高 2.5% ）。在高精度状态下,与 ResNeSt-50 [64]相比,FBNetV3-E的精度提高了 0.2，FLOPs减少了 7 以上,而 FBNetV3-G 达到了与 EfficientNetB4 [46]相同的精度水平， FLOPs减少了2 。请注意,我们通过使用更大的教师模型进行蒸馏进一步提高了 FBNetV3 的准确性,如附录A.7 所示。

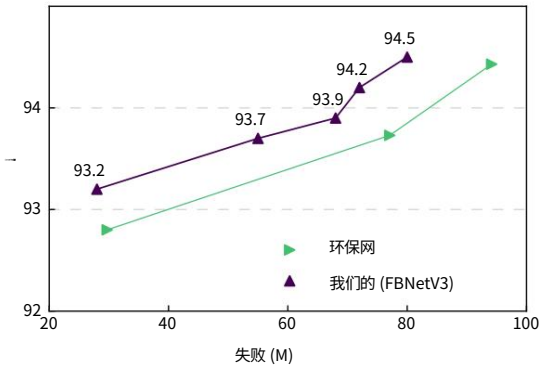

图 7： CIFAR-10数据集上的准确度与 FLOP 比较。

4.3.搜索模型的可转移性

CIFAR-10 上的分类我们进一步在 CIFAR-10 数据集上扩展搜索到的 FBNetV3,该数据集具有来自 10 个类别[24]的 60K 图像,以验证其可迁移性。请注意,与[46]将基本输入分辨率放大到224 224 不同,我们将原始基本输入分辨率保持为32 32,并根据缩放比例放大更大模型的输入分辨率。我们还将第二个步幅为 2 的块替换为步幅为 1 的块以适应低分辨率输入。为简单起见,我们不包括蒸馏。我们在图7中比较了不同模型的性能。同样,我们搜索到的模型明显优于EfficientNet 基线。

骨干	#Params (M)	FLOPs (G)	mAP
高效网B0	8.0	3.6	30.2
FBNetV3-A	5.3	2.9	30.5
效率网B1	13.3	5.6	32.2
FBNetV3-E	10.6	5.3	33.0

表 5： COCO 上具有不同主干的 Faster RCNN 的目标检测结果。

COCO 上的检测为了进一步验证搜索到的模型在不同任务上的可迁移性,我们使用 FB

NetV3 作为具有 conv4 (C4) 骨干的 Faster R-CNN 骨干特征提取器的替代品,并与 COCO 检测数据集上的其他模型进行比较。我们采用[54]中的大部分训练设置,进行3训练迭代,同时使用同步批量归一化,将学习率初始化为 0.16,打开 EMA,将非最大抑制（NMS)减少到 75,并更改为热身后学习率计划为余弦。请注意,我们仅传输搜索到的架构并对所有模型使用相同的训练协议。

我们在表 5 中显示了详细的 COCO 检测结果。与 EfficientNet 骨干网相比,我们的 FBNetV3 在具有相似或更高 mAP 的情况下,将 FLOP 和参数数量分别减少了 18.3% 和34.1%。

5. 消融研究与讨论

在本节中,我们将重新审视从联合搜索中获得的性能改进、基于预测器的进化搜索的重要性,以及几种训练技术的影响和普遍性。

架构和培训配方配对。我们的方法为不同的模型产生不同的训练方法。例如,我们观察到较小的模型倾向于较少的正则化（例如,较小的随机深度下降比和混合比）。为了说明神经架构-配方搜索的重要性,我们交换了搜索 FBNetV3-B 和 FBNetV3-E 的训练配方,观察到这两个模型的准确率显着下降,如表6 所示。

这凸显了正确架构-配方配对的重要性,强调了传统 NAS 的失败:忽略训练配方,仅搜索网络架构无法获得最佳性能。

	FBNetV3-B 训练食谱	FBNetV3-E 训练食谱
FBNetV3-B 架构	79.8%	78.5%
FBNetV3-E 架构	80.8%	81.3%

表 6:具有交换训练方法的搜索模型的准确度比较。

基于预测器的进化搜索改进。
基于预测器的进化搜索在约束迭代优化之上产生了实质性的改进。为了证明这一点,我们将在相同的 FLOPs 约束下从第二搜索阶段派生的最佳表现候选者与最终搜索的 FBNetV3 进行比较（表7）。如果丢弃第三阶段,我们观察到准确度下降高达 0.8% 。因此,第三个搜索阶段虽然只需要微不足道的成本（即几个 CPU 分钟）,但对最终模型的性能同样至关重要。

模型	进化搜索	FLOPs	准确性
FBNetV3-B	是	4.61亿	79.8%
FBNetV3-B	否	448M	79.0%
FBNetV3-E	是	762M	81.3%
FBNetV3-E	否	7.46亿	80.7%

表 7:基于预测的进化搜索搜索的性能改进。 *:模型源自约束迭代优化。

蒸馏和模型平均的影响我们在表8中展示了具有不同训练配置的 FBNetV3-G 模型性能,其中基线指的是没有 EMA 或蒸馏的普通训练。 EMA 带来了更高的准确性,尤其是在训练的中期。我们假设 EMA 本质上是一种强大的“集成”机制,从而提高了单模型的准确性。我们还观察到蒸馏带来了显着的性能提升。这与[4, 61]中的观察结果一致。请注意,由于教师是预训练的 FBNetV3-G,因此 FBNetV3-G 是自蒸馏的。 EMA 和蒸馏的结合将模型的top-1 准确率从 80.9% 提高到 82.8%。

训练 / 模型	基线	EMA	Dist	Dist +EMA
FBNetV3-G	80.9%	82.3%	82.2%	82.8%

表 8： EMA 和蒸馏的性能改进。 *:基于蒸馏的训练

六,结论

顾名思义,以前的神经架构搜索方法仅使用一组固定的训练超参数（即“训练配方”）搜索架构。因此,以前的方法忽略了更高精度的架构配方组合。然而,我们的 NARS 没有,它是第一个同时联合搜索架构和训练方法的算法,用于像 ImageNet 这样的大型数据集。至关重要的是,NARS 的预测器在“免费”架构统计数据（即 FLOP 和#Parameters)上进行预训练,以显着提高预测器的样本效率。在训练和使用预测器之后,生成的 FBNetV3 架构-配方对在 ImageNet 分类上达到了最先进的 per-FLOP 精度。

1致谢： Alvin Wan 得到美国国家科学基金会研究生研究奖学金的支持,资助号为 DGE 1752814。
除了 NSF CISE Expeditions Award CCF-1730628 之外,加州大学伯克利分校的研究还得到了阿里巴巴、亚马逊网络服务、蚂蚁金服、CapitalOne、爱立信、Facebook、Futurewei、谷歌、英特尔、微软、Nvidia、Scotiabank、Splunk 和 VMware的捐赠支持.

参考

[1] Bowen Baker、Otkrist Gupta、Ramesh Raskar 和 Nikhil Naik。使用性能预测加速神经结构搜索。 arXiv 预印本 arXiv:1705.10823, 2017. 3

[2]詹姆斯·伯格斯特拉、丹尼尔·亚明斯和大卫·丹尼尔·考克斯。使模型搜索成为一门科学:视觉架构的数百个维度的超参数优化。 2013.3 _

[3]安德鲁·布罗克、杰夫·多纳休和凯伦·西蒙尼安。用于高保真自然图像合成的大规模 gan 训练。
ICLR, 2019. 12 [4]
蔡寒, 干闯, 王天哲, 张哲凯,韩松.一劳永逸:训练一个网络并对其进行专门化以实现高效部署。 ICLR, 2020. 2, 3, 6, 7, 8 [5]蔡寒,朱力耕,韩松。
Proxylessnas:在目标任务和硬件上直接进行神经架构搜索。 ICLR, 2019. 1 , 3, 7

[6] Ekin D Cubuk、Barret Zoph、Dandelion Mane、Vijay Vasude van 和 Quoc V Le。自动增强:从数据中学习增强策略。 In CVPR, 2019. 3 [7]
Xiaolian Dai, Hongxu Yin, and Niraj K Jha. NeST:一种基于增长和修剪范式的神经网络综合工具。

IEEE Trans on Computers, 2019. 2 [8]戴
晓亮,张培钊,吴碧晨,尹鸿旭,孙飞,王洋汉,Marat Dukhan,胡云清,吴一鸣,贾扬清,等。 Chamnet:通过平台感知模型适应实现高效网络设计。在 CVPR, 2019. 3, 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei.
ImageNet:一个大规模的分层图像数据库。
在 CVPR,2009. 5
[10] Piotr Dollár、Mannat Singh 和 Ross Girshick。快速准确的模型缩放。 arXiv 预印本 arXiv:2103.06877, 2021. 12

[11]郭子超, 张翔宇, 穆浩源, 文恒,刘泽春, 魏一臣, 孙健.具有均匀采样的单路径一次性神经结构搜索。 ECCV, 2020. 3

[12]宋涵,毛惠子,William J Dally。深度压缩:通过修剪、训练量化和霍夫曼编码压缩深度神经网络。 ICLR, 2016. 2 [13] Song Han, Jeff Pool, John Tran, and William Dally.学习有效神经网络的权重和连接。在NeurIPS, 2015 年。2

[14]何开明、范浩琪、吴宇新、谢赛宁和罗斯·吉尔希克。无监督视觉表示学习的动量对比。 In CVPR, 2020. 12 [15]何开明,张翔宇,任少卿,孙健。

用于图像识别的深度残差学习。在 CVPR, 2016. 6, 7

[16] Daniel Ho、Eric Liang、Ion Stoica、Pieter Abbeel 和 Xi Chen。基于人口的增强:增强策略时间表的有效学习。 ICML, 2019. 3 [17] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al.正在搜索 mo bilenetv3。国际计算机视觉大会, 2019. 2, 6, 7

[18] Andrew G Howard、Menglong Zhu、Bo Chen、Dmitry Kalenichenko、Weijun Wang、Tobias Weyand、Marco An dreetto 和 Hartwig Adam。MobileNets:用于移动视觉应用的高效卷积神经网络。 arXiv预印本 arXiv:1704.04861, 2017. 2 [19]高煌、刘庄、Laurens Van Der Maaten 和 Kil ian Q Weinberger。密集连接的卷积网络。

在 CVPR,2017. 6
[20]高煌、孙宇、刘庄、丹尼尔·塞德拉和基利安·Q·温伯格。具有随机深度的深度网络。在 ECCV, 2016. 5

[21] Itay Hubara、Matthieu Courbariaux、Daniel Soudry、Ran El Yaniv 和 Yoshua Bengio。二值化神经网络。在NeurIPS,2016 年。3

[22] Kirthevasan Kandasamy、Willie Neiswanger、Jeff Schneider、Barnabas Poczos 和 Eric P Xing。具有贝叶斯优化和最优传输的神经结构搜索。在 NeurIPS， 2018年。3

[23] Diederik P Kingma 和 Jimmy Ba。 Adam:一种随机优化方法。 ICLR, 2015. 5 [24] Alex Krizhevsky, Geoffrey Hinton, et al.从微小图像中学习多层特征。 2009. 7 [25] Alex Krizhevsky、Ilya Sutskever 和 Geoffrey E Hinton。使用深度卷积神经网络进行图像网络分类。

In NeurIPS, 2012. 6
[26] Richard Liaw、Eric Liang、Robert Nishihara、Philipp Moritz、 Joseph E Gonzalez 和 Ion Stoica。 Tune:分布式模型选择和训练的研究平台。 ICML AutoML Workshop, 2018. 3

[27] Tsung-Yi Lin、Priya Goyal、Ross Girshick、Kaiming He 和Piotr Dollár。密集物体检测的焦点损失。在 ICCV, 2017. 12

[28] Tsung-Yi Lin、Michael Maire、Serge Belongie、James Hays、 Pietro Perona、Deva Ramanan、Piotr Dollár 和 C Lawrence Zitnick。Microsoft coco:上下文中的常见对象。在ECCV, 2014. 12

[29] Chenxi Liu, Barret Zoph, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy.
渐进式神经架构搜索。 ECCV, 2018. 3 [30]刘寒晓,Karen Simonyan,杨一鸣。 Darts:可区分的架构搜索。 ICLR, 2019. 3

[31] Zhichao Lu、Kalyanmoy Deb、Erik Goodman、Wolfgang Banzhaf 和 Vishnu Naresh Boddeti。 Nsganetv2:进化多目标代理辅助神经结构搜索。在欧洲计算机视觉会议上,第35-51 页。施普林格,2020 年。3

[32]马宁宁,张翔宇,郑海涛,孙建。
ShuffleNet V2:高效 CNN 架构设计实用指南。 ECCV, 2018. 2

[33] Dhruv Mahajan、Ross Girshick、Vignesh Ramanathan、Kaim ing He、Manohar Paluri、Yixuan Li、Ashwin Bharambe 和Laurens van der Maaten。探索弱监督预训练的局限性。 In ECCV, 2018. 2 [34] Dmitrii Marin、Zijian He、Peter Vajda、Priyam Chatterjee、 Sam Tsai、Fei Yang 和 Yuri Boykov。高效分割:学习语义边界附近的下采样。在 ICCV, 2019. 3

[35]梅洁茹, 李英伟, 连晓晨, 金晓杰, 杨林杰, Alan Yuille, 杨建超. Atomnas:细粒度端到端神经架构搜索。 ICLR,2020. 7 [36] Luke Metz、Niru Maheswaranathan、Ruoxi Sun、C Daniel Freeman、Ben Poole 和 Jascha Sohl-Dickstein。使用一千个优化任务来学习超参数搜索策略。 arXiv 预印本 arXiv:2002.11887, 2020.11 [37] Harald Niederreiter。随机数生成与准

蒙特卡洛方法。暹罗, 1992. 4

[38] Adam Paszke、Sam Gross、Soumith Chintala、Gregory Chanan、Edward Yang、Zachary DeVito、Zeming Lin、Al ban Desmaison、Luca Antiga 和 Adam Lerer。 PyTorch 中的自动微分。在关于 Autodiff 的 NeurIPS 研讨会上， 2017 年。5

[39] Ilija Radosavovic、Raj Prateek Kosaraju、Ross Girshick、Kaim ing He 和 Piotr Dollár。设计网络设计空间。
CVPR, 2020. 7

[40] Esteban Real、Alok Aggarwal、Yanping Huang 和 Quoc V Le。图像分类器架构搜索的正则化演化。在 aaai 人工智能会议论文集，第 33 卷，第 4780-4789 页，2019年。3

[41] Esteban Real、Sherry Moore、Andrew Selle、Saurabh Saxena、 Yutaka Leon Suematsu、Jie Tan、Quoc V Le 和 Alexey Kurakin。图像分类器的大规模进化。在 JMLR, 2017. 1, 3

[42]马克·桑德勒、安德鲁·霍华德、朱梦龙、Andrey Zh moginov 和陈良杰。反向残差和线性瓶颈：用于分类、检测和分割的移动网络。 CVPR, 2018. 2, 5, 6

[43]韩石, 皮仁杰, 许航, 李振国, James T Kwok, and Tong Zhang.具有可学习预测器的高效基于样本的神经架构搜索。 arXiv，arXiv–1911 页， 2019。3

[44] Mandavilli Srinivas 和 Lalit M Patnaik。遗传算法中交叉和变异的自适应概率。 IEEE跨。 Systems, Man, and Cybernetics, 1994. 4 [45] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, and Quoc V Le. MnasNet：移动平台感知神经架构搜索。 CVPR, 2019. 1, 3 [46] Mingxing Tan 和 Quoc V Le。 Efficientnet：重新思考卷积神经网络的模型缩放。 ICML，2019 . 6、7 [47] Toan Tran、Trung Pham、Gustavo Carneiro、Lyle Palmer 和Ian Reid。用于学习深度模型的贝叶斯数据增强方法。在 NeurIPS，2017 年。3

[48] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuan dong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2：可区分的神经架构搜索空间和通道维度。 CVPR, 2020. 3, 5, 6, 7

[49] Linnan Wang, Yiyang Zhao, Yuu Jinnai, Yuuandong Tian, and Rodrigo Fonseca.使用深度神经网络和蒙特卡洛树搜索的神经架构搜索。在 AAAI 中，2020. 3

[50] Wei Wen、Hanxiao Liu、Hai Li、Yiran Chen、Gabriel Ben der 和 Pieter-Jan Kindermans。用于神经结构搜索的神经预测器。 ECCV, 2020. 3, 4

[51] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet：通过可微神经架构搜索进行硬件感知的高效卷积网络设计。在CVPR, 2019. 3, 7

[52] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonza lez, and Kurt Keutzer. Shift：零 FLOP、零参数替代空间卷积。 CVPR, 2018. 2 [53] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer。

Squeezeseg：具有循环 crf 的卷积神经网络，用于从 3d 激光雷达点云进行实时道路对象分割。
在 ICRA，2018年。3

[54] Yuxin Wu、Alexander Kirillov、Francisco Massa、Wan-Yen Lo 和 Ross Girshick。探测器2。 https://github. com/facebookresearch/detectron2, 2019. 8 , 12 [55]谢赛宁、Ross Girshick、Piotr Dollár、涂卓文和何开明。深度神经网络的聚合残差变换。在 CVPR, 2017. 6, 7

[56]谢斯瑞, 郑和辉, 刘春晓, 林梁. SNAS：随机神经结构搜索。 ICLR, 2019. 3 [57] 徐辰峰, 吴碧晨, 王子宁, 詹伟, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3：用于高效点云分割的空间自适应卷积。 ECCV, 2020. 3 [58] Tien-Ju Yang、Andrew Howard、Bo Chen、Xiao Zhang、Alec Go、Mark Sandler、Vivienne Sze 和 Hartwig Adam。 Net tAdapt ：用于移动应用程序的平台感知神经网络自适应。在 ECCV, 2018. 2

[59]杨朝晖, 王运河, 陈兴浩, 施博新,徐超, 徐春景, 齐田, 徐畅.汽车：高效神经架构搜索的持续进化。进行中

IEEE/CVF 计算机视觉和模式识别会议的 ings，第 1829–1838 页， 2020。3 [60] Hongxu Yin、Pavlo Molchanov、Zhizhong Li、Jose M Alvarez、Arun Mallya、Derek Hoiem、Niraj K Jha 和 Jan考茨。

梦想提炼：深度版无数据知识传递。 CVPR, 2020. 2

[61] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas：使用大型单阶段模型扩展神经架构搜索。
ECCV, 2020. 6, 7, 8 [62]
Arber Zela、Aaron Klein、Stefan Falkner 和 Frank Hut ter。迈向自动化深度学习：高效的联合神经架构和超参数搜索。 arXiv 预印本arXiv:1807.06906, 2018.3 [63] Hongyi Zhang、Moustapha Cisse、Yann N Dauphin 和 David Lopez-Paz。混合：超越经验风险最小化。

ICLR, 2018. 5

[64]张航，吴崇若，张中岳，朱一，张志，林海滨，孙悦，何彤，Jonas Mueller，R Manmatha，等。 Resnest：分裂注意网络。 arXiv preprint arXiv:2004.08955, 2020. 6, 7 [65] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally。

训练有素的三元量化。 ICLR，2017. 3 [66] Barret Zoph 和 Quoc V Le。具有强化学习的神经结构搜索。国际语言文字学会, 2017. 3