汽车:高效神经架构搜索的持续进化

赵辉杨 1，2，云和王 2，星昊辰 2，博信世 3，4，
朝旭 1，春景旭 2，齐天 2，长旭* 5

[1] 机械知觉重点实验室。北京大学机器智能学院。

[2] 华为技术有限公司诺亚方舟实验室。[3] NELVT，部门。北京大学计算机科学系。[4] 程鹏实验室。

[5] 悉尼大学工程学院计算机科学学院。

{zhaohuiyang,shiboxin}@pku.edu.cn；xuchao@cis.pku.edu.cn。

{yunhe.wang,xinghao.chen,tian.qi1,xuchunjing}@huawei.com；c.xu@sydney.edu.au。

DOI 10 . 1109/cvpr 42600 . 2020 . 00190

摘要

由于效率的原因，现有的大多数神经网络结构搜索算法中的搜索技术主要由可微方法控制。相比之下，我们开发了一种高效的连续进化方法来搜索神经网络。在最新一代中，在一个超网内共享参数的群体中的体系结构将在具有几个时期的训练数据集中进行调整。下一代进化中的搜索将直接继承超网和种群，加速最优网络生成。非支配排序策略进一步应用于仅保留帕累托前沿的结果，以准确更新超网。只需 0.4 GPU 天的连续搜索，就会产生几个不同模型大小和性能的神经网络。因此，我们的框架提供了一系列网络，其参数数量在移动设置下从 3.7M 到 5.1M 不等。这些网络超过了在基准 ImageNet 数据集上由最先进的方法产生的网络。

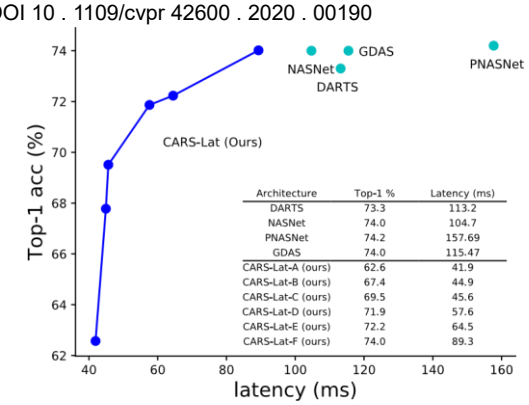


图 1。CARS-Lat 模型在 CIFAR-10 数据集上搜索。搜索阶段考虑验证性能和设备感知变量，即，。华为 P30 Pro 上的移动设备延迟。最高精度是在 ILSVRC2012 数据集上的性能。

1。 正式介绍

卷积神经网络在大范围的计算机视觉任务中取得了很大的进展，如识别[22，19，18]，检测[12]和分割[21]。过度参数化的深度神经网络可以产生令人印象深刻的性能，但同时会消耗巨大的计算资源。高效块设计[57，56，55]，张量分解[27，52，53，45，46]，剪枝[28，29，13]，蒸馏[4，51]和量化[25]是使网络高效的流行技术。设计新颖的网络体系结构在很大程度上依赖于人类专家的知识和经验，并且可能需要许多三重

在获得有意义的结果之前进行 als[22]。为了加速这一过程并使其自动化，已经提出了网络体系结构搜索(NAS) [44，14，36，20，16]，所学习的体系结构在各种任务上已经超过了那些人工设计的体系结构。然而，这些搜索方法通常需要大量的计算资源来搜索性能可接受的体系结构。

用于搜索神经架构的技术主要分为三组，即。基于进化算法、基于强化学习和基于梯度的方法。基于环境分析的工作，[38，37，48，42，40，41]初始化一组模型并进化为更好的架构，这很耗时，例如，。Real et al . 搜索需要 3150 个 GPU 天[38]。基于 RL 的工作，[60，58]使用控制器来预测一系列操作，并训练不同的架构来获得回报。给出一个
体系结构中，这些方法必须针对大量时期对其进行训练，然后评估其性能以指导进化或优化控制器，这使得搜索阶段效率较低。基于梯度的方法(例如，。DARTS [32])首先训练一个 SuperNet，并在搜索时引入对连接的注意机制，然后在搜索后移除弱连接。这个

*对应作者。

2575-7075/20/$31.00 2020 IEEE

阶段是通过梯度下降优化进行的，非常有效。然而，搜索到的架构缺乏多样性。

虽然[37]中的一些实验表明，进化算法发现了比基于反向传播的方法更好的神经结构，但由于每个个体的评估过程，即。独立评估进化算法中的神经网络。此外，在搜索空间中可能有一些性能极其糟糕的体系结构。如果我们直接遵循ENAS [35]提出的权重分配方法，则必须训练超网来补偿那些较差的搜索空间。有必要对现有的进化算法进行改造，以实现高效而精确的神经结构搜索。

在本文中，我们提出了一个有效的基于进化算法的神经结构搜索框架。开发了一个持续进化策略，以最大限度地利用我们在上一代进化中学习到的知识。具体来说，首先用相当多的单元和块初始化超级网络。进化算法中代表从超网中导出的体系结构的个体将通过几个基准操作(即，。交叉和突变)。采用非支配排序策略选择几个模型大小和精度不同的优秀架构，并更新 SuperNet 中对应的单元进行后续优化。下一代中的进化过程基于更新的超网络和通过非支配排序获得的多目标解集而持续执行。此外，我们建议开发一种保护机制来避免小模型陷阱问题。所提出的连续进化体系结构搜索(CARS)可以高效地提供帕累托前沿的一系列模型。在基准数据集上验证了该方法相对于现有方法的优越性。

2。 相关作品

2.1。 网络架构搜索

基于梯度的网络体系结构搜索方法包含两个步骤:网络参数优化和体系结构优化。网络参数优化步骤优化标准层中的参数(即，。卷积、批量归一化、完全连接层)。架构优化步骤学习精确网络架构的模式。

参数优化步骤可以分为两类，独立优化和共享优化。独立优化分别学习每个网络，即，。阿米巴网[37]评估数千个模型需要数千个 GPU 天。为了加速训练，[11，10]通过网络态射初始化参数。Oneshot 方法[1，17]通过在一个 SuperNet 内共享不同体系结构的所有参数而更进一步。不需要训练成千上万个不同的体系结构，只需要优化一个超级网络。

架构优化步骤包括基于 RL、基于 EA 和基于梯度的方法。基于 RL 的方法[60，61，35]使用递归网络作为网络架构控制器，并且所生成的架构的性能被用作训练控制器的奖励。控制器在训练过程中收敛，最终输出性能优越的架构。基于进化算法的方法[48，37]借助进化算法搜索体系结构。每个个体的验证准确性被用作进化下一代的适应性。基于梯度的方法[32，49，47]将网络架构视为一组可学习的参数，并通过标准反向传播算法优化参数。

2.2。 多目标网络架构搜索

考虑多个互补目标，即。精度、参数数量、浮点运算(FLOPs)、能量和延迟，在所有目标上，没有一个单一的架构能够超越所有其他架构。因此，帕累托前沿的架构是理想的。已经提出了许多不同的工作来处理多目标网络体系结构搜索。NEMO [26]和 MNASNet [44]瞄准速度和精度。DPPNet 和柠檬水[8，11]考虑设备相关和设备无关的目标。MONAS [23]瞄准精度和能量。NSGANet [33]考虑了 FLOPs 和精度。

这些方法对于单独优化的模型效率较低。相比之下，我们的架构优化和参数优化步骤是交替进行的。此外，不同架构的参数是共享的，这使得搜索阶段更加高效。

3。 接近

在这一节中，我们开发了一种新的连续进化方法来搜索神经结构，即
汽车。汽车搜索阶段包括两个过程，即，。参数优化和架构优化。

我们使用遗传算法进行架构进化，因为遗传算法维护一组覆盖广阔空间的性能良好的架构。我们维护一套架构(也称为。联系)C = {C1，...,。*CP}，其中 P 为种群规模。*在体系结构优化步骤中，群体中的体系结构根据建议的 pNSGA-III 方法逐步更新。为了提高搜索阶段的效率，我们维护了一个超级网络，它为不同的体系结构共享参数。参数共享策略显著降低了分别训练这些不同架构的计算复杂度。

3.1。 汽车超级网络

从超网络 N 中采样不同的网络，并且每个网络可以由一组全精度参数 Wi 和一组二进制连接参数(即。{0，1}) Ci。连接配置项中的 0 元素表示网络不包含此连接来转换数据流，1 元素连接表示网络使用此连接。从这个角度来看，每个网络 Ni 可以表示为(Wi，Ci)对。

全精度参数 W 由一组网络共享。如果这些网络架构是固定的，参数可以通过反向传播进行优化。最优 W 适合于所有的网络 Ni，以获得更高的识别性能。在参数收敛后，我们可以通过遗传算法交替优化二进制连接。这两个步骤构成了我们提出的方法的主要优化。我们将在下面介绍这两个优化步骤。

3.2。 参数最优化

参数 W 是网络中所有参数的集合。第一个人的参数是$W_i = W \odot C_i, \quad i \in \{1, \ldots, P\}$
是一种掩码操作，它只保留对应于连接 Ci 中 1 个元素的位置的完整图形的参数。把 X 表示为输入数据，这个网络的预测是 Ni(X)，其中 Ni 是第 I 个架构。预测损失可以表示为 Li＝H(Ni(X)，Y)，其中 H 为判据，Y 为目标。Wi 的梯度可以计算为

$$dW_i = \frac{\partial \mathrm{Li}}{\partial W_i} = \frac{\partial \mathrm{L}}{\partial W} \odot C_i \quad \text{。(1)}$$

参数 W 应该适合所有个体，因此所有网络的梯度被累加以计算参数 W 的梯度

$$dW = \frac{1}{P}\sum_{i=1}^{P} dW_i = \frac{1}{P}\sum_{i=1}^{P} \frac{\partial L_i}{\partial W} \odot C_i \quad \text{. (2)}$$

任何层都只能由在转发过程中使用该层的网络进行优化。通过收集种群中个体的梯度，通过 SGD 算法更新参数 W。

由于我们在 SuperNet 中维护了一大组具有共享权重的架构，我们借用了随机梯度下降的思想，并使用小批量架构来更新参数。累积所有网络的梯度将花费大量时间进行一步梯度下降，因此我们使用小批量架构来更新共享权重。我们使用 B 不同的架构，其中 B＜P，架构的索引为{n1，...,。*nB}更新参数*。方程 2 的有效参数更新详见方程 3

$$dW \approx \frac{1}{B}\sum_{j=1}^{B} \frac{\partial L_{n_j}}{\partial W_{n_j}} \quad \text{. (3)}$$

因此，小批量结构上的梯度被视为所有 P 个不同个体的平均梯度的无偏近似。每次更新的时间成本可以大大降低，适当的小批量可以在效率和准确性之间达到平衡。

3.3。 架构优化

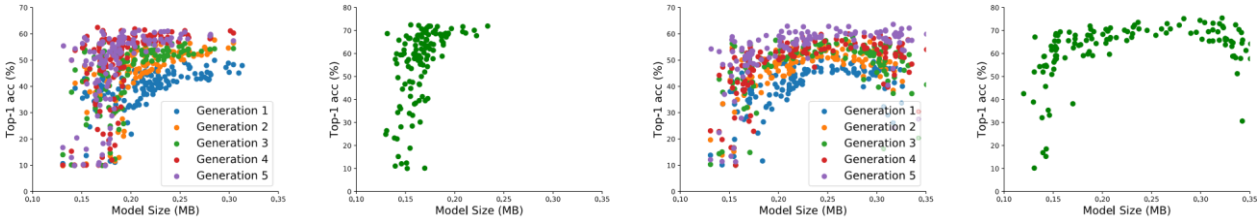
对于体系结构优化过程，我们使用进化算法和非支配排序策略。非支配排序策略已在 NSGA 三世[7]中引入。表示为{NM1，...,。我们希望最小化不同的测量。然后 P}作为 P 个不同的网络和{F1，...,。FM}

测量，例如参数数量、浮点运算、延迟、能量和精度，可能会有一些冲突，这增加了发现最小化所有这些度量的最佳解决方案的难度。

在实践中，如果满足两个条件，Ni 支配 Nj:(1)对于任何测量，Ni 的性能不差于 Nj。(2)在至少一次测量中，模型 Ni 的表现优于模型 Nj。形式上，支配的定义可以概括如下。

FM}我们希望最小化。如果

„M} (4) $F_k(N_i) < F_k(N_j), \exists k \in \{1,...M\}$ *(4)* $Fk(Ni) <$ $Fk(Nj)$，$\exists k \sum\{1，...,M\}$，



(a) NSGA-III (1-5 generation) (b) NSGA-III (20 generation) (c) pNSGA-III (1-5 generation) (d) pNSGA-III (20 generation)

图 2。不同进化策略的比较。超级网络在训练集上训练，并在验证集上评估。图 2(a)显示了使用 NSGA 三号的3代进化。NSGA 三号的进化受到小模型陷阱的影响，导致分布偏向于更小的模型，图 2(b)显示了 NSGA 三号在 2 代之后的进化分布，其中维护的架构都是小模型。图 2(c)显示了使用建议的 pNSGA-III 的进化世代。pNSGA-III 的演变为更大的模型提供了保护，图 2(d)显示了 pNSGA-III 在 20 代之后的演变分布，其中维护的架构在模型大小维度上覆盖了很大的范围。

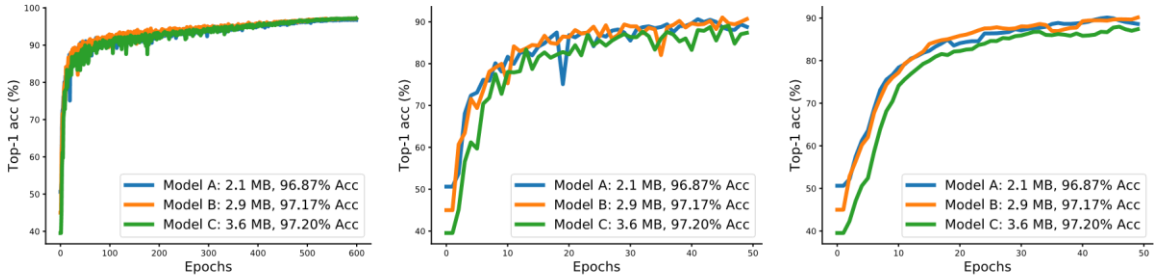
新参数 W。

据说倪主宰了$\mathcal{N}_j, i.e., \mathcal{N}_i \preceq \mathcal{N}_j$。

根据上述定义，如果 Ni 支配 Nj，则 Nj 可以在演进过程中被 Ni 替换，因为 Ni 在至少一个度量方面表现更好，而在其他度量方面不差。通过利用这种方法，我们可以从当代人群中选择一系列优秀的神经架构。然后，这些网络可以用于更新超网中的相应参数。

尽管上述非支配排序策略使用 NSGA-iii方法[7]来选择一些更好的模型

更新参数，在搜索过程中存在一个小的模型陷阱现象。具体来说，由于超网中的参数仍然需要优化，所以当前一代中每个单独架构的精度可能并不总是代表其最终可以实现的性能，如 NASBench101 [54]



(a) Acc curve (epochs=500, window=1) (b) Acc curve (epochs=50, window=1) (c) Acc curve (epochs=50, window=5)

图 3。三种不同模型尺寸的精度曲线。左图为 600 个时期的训练精度曲线，中图为前 50 个时期的精度曲线，右图为窗口平滑后的曲线

中所讨论的。因此，一些参数较少但测试精度较高的较小模型往往会主导那些精度较低但有可能达到较高精度的较大模型，如图 3 所示。

因此，我们建议改进传统的 NSGA 三号，以保护这些较大的型号，即第三代核动力源。更具体地说，pNSGA-III 算法考虑了精度的增加速度。我们以验证精度和参数数量为例。对于 NSGA-iii方法，非支配排序算法考虑两个不同的目标，并根据排序后的帕累托阶段选择个体。对于所提出的 pNSGA-III，除了考虑参数数量和精度之外，我们还进行了一个非支配排序算法，该算法考虑了精度和参数数量的增加速度。然后将两个不同的帕累托阶段合并。假设 P 是人口规模，经过两个帕累托阶段 R1...n1，Q1...n2，我们从第一个帕累托前沿开始逐渐合并两个帕累托阶段，合并第 I 个前沿后的并集 Ui 为 Ui =(∪R1)∪∪(Ri∪齐)。我们从 Umax(n1，n2)中保留第一批 P 个体。这样，性能增长速度较慢的大型网络可以保留在群体中。

在图 2 中，显示了使用 NSGA-iii和三七总皂苷-iii的人群。如果我们使用 NSGA 三号来更新架构，就会遇到小模型陷阱问题。很明显，pNSGA-III 可以在进化过程中保护大型模型，并提供广泛的模型。更详细的讨论将在下一节中介绍。3.4.汽车的持续进化

总之，通过使用建议的 CARS 管道搜索最佳架构有两个步骤，1)架构优化 2)参数优化。此外，首先还引入了参数预热来更新参数。

参数预热。由于我们的超网的共享权值是随机初始化的，如果群中的体系结构也是随机初始化的，那么与其他操作相比，所有体系结构中最常用的操作将被训练更多次。因此，通过遵循一次性网络连接存储方法[1，17，6，47]，我们使用统一的采样策略来初始化超网中的参数。这样，超网以相同的可能性训练每个可能的操作。例如，在 DARTS [32]管道中，每个节点有八种不同的操作，包括卷积、池化、身份映射和无连接。每个操作都将以$\frac{1}{8}$概率进行采样。

架构优化。在初始化超网的参数后，我们首先随机抽样 P 个不同的结构，其中 P 是一个超参数，表示种群中保持的个体数量。在…期间

5 码。

在体系结构演化步骤中，我们首先生成 t × P 子代，其中 t 是控制扩展比的超参数。然后我们用 pNSGA-III 对架构进行排序，从(t +1) × P 个个体中选择 P 个个体。选择的 P 架构形成下一代。

参数优化。给定一组架构，我们根据等式 3 使用所提出的小批量架构更新方案进行参数优化。

算法 1 总结了所提出的用于搜索神经体系结构的连续进化算法的详细过程。

3.5。 搜索时间分析

在 CARS 的搜索阶段，训练集用于更新网络参数，验证集用于更新架构。假设一个架构在训练集上的平均训练时间是 Ttr，在验证集上的推理时间是 Tval。第一个预热阶段以 Ewarm 为纪元，在这个阶段需要 Twarm = Ewarm × Ttr 来初始化 SuperNet N 中的参数。

假设架构总共为 Eevo 代进化。并且每一代都包含参数优化和架构优化步骤。参数优化步骤在各代之间的训练集上为 Eparam 时期训练 SuperNet，因此一个进化代中参数优化的时间成本为 Tparam = Eparam × Ttr × B，B 为小批量。对于架构优化步骤，可以并行推断所有个体，因此该步骤中的时间成本可以计算为 Tarch = Tval。因此，Eevo 进化世代的总时间成本为 Tevo = Eevo × (Tparam + Tarch)。在 CARS 中所有的搜索时间成本是，

$$Ttotal = Twarm + Tevo。$$

=Ewarm × Ttr+ (5)

Eevo × (Eparam × Ttr × B + Tval)

用于高效神经架构搜索的算法 1 连续进化 FM}、参数优化时代 Eparam、准则 h。

1:为电子时代热身。

Eevo do

„E_{param} do 4: for Mini-batch data X, target Y in loader do

5: Random sample B indices n_1,...*Eparam do 4:对于小批量数据 X，加载器 do 5 中的目标 Y:随机样本 B 索引 n1，…,nB。*

6: 选择 这 相应的 B 联系 *CnB 根据指数。*

NnB。

8: 正向 B 采样网络。

9: 计算$L = \frac{1}{B}\sum_{i=1}^{B}\mathcal{H}\left(\mathcal{N}_{n_i}(X), Y\right)$,损失

10: 根据等式 3 计算梯度。

11: 更新网络参数 w。

12: 结束于

13: 结束于

14: 使用 pNSGA-III 更新$\{C_1^{(e)}, \ldots, C_P^{(e)}\}$。

15:结束

输出: $C = \{C_1^{(E_{evo})}, \ldots, C_P^{(E_{evo})}\}$建筑。

4。 实验

在这一节中，我们首先介绍超网，以及我们实验中的实验细节。然后，我们检查了小模型陷阱现象，并比较了 NSGA-三与我们提出的 pNSGA-三。我们在 CIFAR-10 数据集上搜索了两次，分别考虑了设备无关性和设备感知性目标。所有搜索到的体系结构都在 CIFAR-10 和 ILSVRC2012 数据集上进行评估。这两个数据集是识别任务的基准。

Table 1. Comparison with state-of-the-art image classifiers on CIFAR-10 dataset. The multi-objectives used for architecture optimizatio are performance and model size. We follow DARTS and use the cutout strategy for training.

Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	Search Method
DenseNet-BC [24]	3.46	25.6	-	manual
PNAS [30]	3.41	3.2	225	SMBO
ENAS + cutout [35]	2.91	4.2	4	RL
NASNet-A + cutout [61]	2.65	3.3	2000	RL
AmoebaNet-A + cutout [37]	3.12	3.1	3150	evolution
Hierarchical evolution [31]	3.75	15.7	300	evolution
SNAS (mild) + cutout [49]	2.98	2.9	1.5	gradient
SNAS (moderate) + cutout [49]	2.85	2.8	1.5	gradient
SNAS (aggressive) + cutout [49]	3.10	2.3	1.5	gradient
DARTS (first) + cutout [32]	3.00	3.3	1.5	gradient
DARTS (second) + cutout [32]	2.76	3.3	4	gradient
Random Search [32]	3.29	3.2	4	random
RENA [59]	3.87	3.4	-	RL
NSGANet [33]	3.85	3.3	8	evolution
LEMONADE [11]	3.05	4.7	80	evolution
CARS-A	3.00	2.4	0.4	evolution
CARS-B	2.87	2.7	0.4	evolution
CARS-C	2.84	2.8	0.4	evolution
CARS-D	2.95	2.9	0.4	evolution
CARS-E	2.86	3.0	0.4	evolution
CARS-F	2.79	3.1	0.4	evolution
CARS-G	2.74	3.2	0.4	evolution
CARS-H	2.66	3.3	0.4	evolution
CARS-I	2.62	3.6	0.4	evolution

4.1。 实验设置

超网主干。为了说明我们的方法的有效性，我们在一个与 DARTS [32]相同的流行搜索空间上评估了我们的 CARS。DARTS 是一个可区分的 NAS 系统，搜索还原和正常细胞。普通单元用于输入要素和输出要素具有相同空间大小的图层。缩减单元用于对输入要素地图进行下采样的图层。在搜索这两种小区后，通过堆叠一组搜索到的小区来构建网络。搜索空间包含八种不同的操作，包括四种卷积、两种池、跳过连接和无连接。

进化细节。在 DARTS 搜索空间中，一个单元中的每个中间节点都与前面的两个节点相连。在相应的节点上进行交叉和变异。交叉率和变异率都设置为 0.25，我们以 0.5 的概率随机生成新的架构.对于交叉操作，每个节点交叉其连接的比率为 0.5，对于变异操作，每个节点随机重新分配的比率为 0.5。

4.2。 在 CIFAR-10 上的实验

我们在 CIFAR-10 上的实验包括小模型陷阱现象的演示、NSGA-iii和 PNSGA-iii的比较、设备不可知和设备感知搜索。评估在 CIFAR10 数据集和大型 ILSVRC2012 数据集上进行。

小模型陷阱。在图 3 中，显示了三个模型的精度曲线。参数数量分别为 2.1M、2.9M 和 3.6M。经过 600 个时期的训练，CIFAR-10 数据集的准确率与模型大小呈正相关，分别为 96.87%、97.17%和 97.20%。我们观察了前 50 个时期的精度曲线，总结了导致小模型陷阱现象的两个主要原因。两个原因是(1)小模型自然收敛更快，(2)训练时精度波动。对于最大的型号 C，其精度始终低于前 50 个时代的型号 A 和型号 B。因此，如果使用 NSGA-三号算法，C 型将被淘汰，这是我们提出的第一个动机。这是因为更大的模型更复杂，因此更难优化。对于型号 B 和型号 A，精度曲线相似(图 3(c))。然

而，由于训练过程中的精度波动(图 3(b))，如果模型 a 的精度在一个时期内高于模型 B，则模型 B 会因非支配排序策略而被淘汰，这是我们提出的第二个原因。这两个原因都可能导致架构更新过程中大型模型的淘汰。因此，提出的 pNSGA-iii是解决小模型陷阱问题所必需的。

NSGA 三世 vs。pNSGA-III。在架构优化步骤中，我们使用 CARS 搜索具有不同 NSGA 方法的架构。多目标是参数数量和模型大小。我们设想在人群中保持的建筑的分布趋势。如图 2 所示，使用 NSGA 三号更新架构会遇到小模型陷阱问题，而大模型在架构优化步骤中会被淘汰。相比之下，通过使用 pNSGA-III 更新架构可以保护更大的模型。较大的模型有可能在以后的时期提高它们的精度，但在开始时比小模型收敛得慢。如果搜索目标是寻找具有各种计算资源的模型，那么在体系结构优化阶段，在群体中维护较大的模型而不是丢弃它们是至关重要的。

在 CIFAR-10 上搜索.我们将 CIFAR-10 列车组分为两部分，即。25，000 个镜像用于更新网络参数，25，000 个镜像用于更新架构。分裂策略与飞镖[32]和 SNAS [49]相同。我们总共搜索 500 个纪元，参数预热阶段持续前 10%纪元(50)。之后，我们初始化种群，种群维护 128 个不同的架构，并使用建议的 pNSGAIII 逐步演化它们。我们用 pNSGA-III 在网络参数更新十个纪元后更新架构。

在 CIFAR-10 上评估。在完成 CARS 搜索阶段后，群体中维护了 N = 128 个架构。我们评估了一些与以前的作品[32，49]具有相似模型大小的架构，以供比较。我们在 CIFAR-10 数据集上重新训练搜索到的体系结构。所有训练参数与 DARTS [32]相同。

我们将搜索到的体系结构与表 1 中的现有技术进行了比较。所有搜索到的架构都可以在补充资料中找到。我们搜索的体系结构在 CIFAR-10 数据集上的参数数量从 2.4M 到 3.6M 不等，这些体系结构的性能与最先进的水平相当。同时，如果我们通过使用 NSGA-三号方法而不是 PNSGA-三号方法来进化体系结构，我们只能搜索一组具有大约 2.4M 参数的体系结构，而没有更大的模型，并且模型的性能相对较差。

与以前的方法如飞镖和 SNAS 相比，我们的方法能够在更大的搜索空间范围内搜索体系结构。CARS-G 实现了与 DARTS(二阶)相当的精度，在较小的模型尺寸下，误差率约为 2.75%。使用与 DARTS 相同的 3.3M 参数(秒-

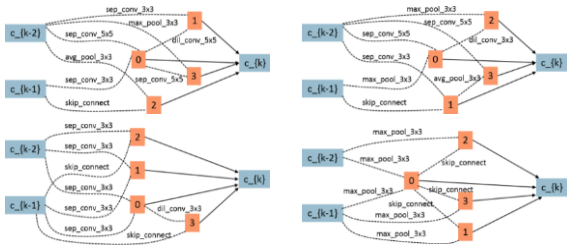


图 4。CARS-H 和飞镖。上面是 CARS-H 的法线和归约块，下面是 DARTS(二阶)中的法线和归约块。

订单)，我们的 CARS-H 实现了较低的测试误差，2.66%对。2.76%. 对于小型车型，我们搜索的 CARS-A/C/D 也取得了与 SNAS 相当的结果。此外，我们的大模型 CARS-I 在参数稍多的情况下实现了 2.62%的较低错误率。从 CARS-A 到 CARS-J 的总体趋势是误差率在增加模型尺寸的同时逐渐降低。这些模型都是帕累托解。与其他多目标方法相比，如 RENA [59]，神经网络[33]和柠檬水[11]，我们搜索的架构也显示出优于这些方法的性能。

已搜索单元格的比较。为了对所提出的方法有一个清晰的理解，我们在图 4 中进一步可视化了由 CARS 和 DARTS 搜索的正常和简化单元.汽车和飞镖(二阶)具有相似数量的参数(3.3M)，但是 CARS-H 比 DARTS(二阶)具有更高的精度。在图 4 中可以发现，CARS-H 约简块中有更多的参数用于保存更多有用的信息，并且 CARS-H 的正常块的大小小于 DARTS(二阶)的大小以避免不必要的计算。CARS 维护着一个覆盖大范围搜索空间的群体。

4.3。 对 ILSVRC2012 进行评估

我们通过在 ILSVRC2012 数据集上训练来评估搜索到的架构的可移植性。我们用 8 款英伟达特斯拉 V100 来训练车型，批量 640.我们总共训练了 250 个纪元。用线性衰减调度器的学习率是 0.5，我们预热前五个时期的学习率。动量为 0.9，重量衰减为 3e-5.标签平滑也以 0.1 的平滑比使用。

表 2 中的结果显示了我们搜索的架构的可移植性。我们的模型涵盖了广泛的参数。型号尺寸从 3.7M 到 5.1M 不等，浮点型从 430 到 590 微操作数不等。对于不同的部署环境，我们可以轻松选择满足计算资源的体系结构。本实验考虑了设备不可知的变量、模型大小和性能，因此 CARS A-I 的延迟不是严格的

Table 2. An overall comparison on ILSVRC2012 dataset. The CARS models are the architectures searched on the CIFAR-10 dataset.

Architecture	Top-1 Acc (%)	Top-5 Acc (%)	Params (M)	+× (M)	Search Cost (GPU days)	Search Method
ResNet50 [22]	75.3	92.2	25.6	4100	-	manual
MorphNet [15]	75.2	-	15.5	3880	-	manual
InceptionV1 [43]	69.8	90.1	6.6	1448	-	manual
MobileNetV2 (1×) [39]	72.0	90.4	3.4	300	-	manual
ShuffleNetV2 (2×) [34]	74.9	90.1	7.4	591	-	manual
PNAS [30]	74.2	91.9	5.1	588	224	SMBO
AutoSlim [55]	75.4	-	8.3	532	-	greedy
SNAS (mild) [49]	72.7	90.8	4.3	522	1.5	gradient
DARTS [32]	73.3	91.3	4.7	574	4	gradient
PDARTS [5]	75.6	92.6	4.9	557	0.3	gradient
PARSEC [3]	74.0	91.6	5.6	548	1	gradient
ProxylessNAS (GPU) [2]	75.1	92.5	7.1	465	8.3	gradient
FBNet-C [47]	74.9	-	5.5	375	20	gradient
RCNet [50]	72.2	91.0	3.4	294	8	gradient
GDAS [9]	74.0	91.5	5.3	581	0.8	gradient
NASNet-A [61]	74.0	91.6	5.3	564	2000	RL
MNASNet-A1 [44]	75.2	92.5	3.9	312	-	RL
AmoebaNet-A [37]	74.5	92.0	5.1	555	3150	evolution
CARS-A	72.8	90.8	3.7	430	0.4	evolution
CARS-B	73.1	91.3	4.0	463	0.4	evolution
CARS-C	73.3	91.4	4.2	480	0.4	evolution
CARS-D	73.3	91.5	4.3	496	0.4	evolution
CARS-E	73.7	91.6	4.4	510	0.4	evolution
CARS-F	74.1	91.8	4.5	530	0.4	evolution
CARS-G	74.2	91.9	4.7	537	0.4	evolution
CARS-H	74.7	92.2	4.8	559	0.4	evolution
CARS-I	75.2	92.5	5.1	591	0.4	evolution

与最终表现呈正相关。华为 P30 Pro 的延迟分别为 82.9、83.3、83.0、90.0、93.8、92.2、98.1、97.2 和 100.6 毫秒。

CARS-I 以 1%的最高精度超越了 PNAS，具有相同数量的参数和近似的浮点运算。CARS-G 在相同数量的参数下显示出比 DARTS 高 0.9%的前 1 名精度的优异结果。此外，CARS-D 在相同的参数数量下超过 SNAS(轻度)0.6%的前一级精度。对于 NASNet 和 AmoebaNet 的不同模型，我们的方法也有各种模型，使用相同数量的参数可以获得更高的精度。通过使用建议的 pNSGA-III，像 CARS-I 这样的大型架构可以在架构优化阶段得到保护。由于高效的参数共享策略，我们可以在一次搜索中搜索到一组优秀的可转移架构。

对于考虑设备感知变量的实验，即。运行时延迟和性能，我们在 ILSVRC2012 数据集上评估了搜索到的架构。结果如图 1 所示.搜索到的架构涵盖了 40 毫秒到 90 毫秒的实际运行时延迟，并且超过了对应的架构。

5。 结论

基于进化算法的网络连接存储方法能够找到高性能的模型，但搜索时间非常长，因为每个候选网络都是单独训练的。为了提高效率，我们提出了一种连续进化的体系结构搜索方法，即 CARS。在进化过程中，CARS 最大限度地利用最新一代进化中学习到的知识，如架构、参数等。超级网络由相当多的单元和块构成。个体是通过进化算法中的基准操作生成的。非支配排序策略被用来选择用于更新超网的体系结构。在基准数据集上的实验表明，所提出的 CARS 能够有效地提

供多种帕累托前沿体系结构。搜索到的模型在模型大小/延迟和准确性方面优于现有技术。

本工作得到了国家自然科学基金项目的资助。61876007，61872012，澳大利亚研究委员会下属项目 DE180101438，北京人工智能研究院(BAAI)。

引用

[1] Gabriel Bender，Pieter-Jan Kindermans，Barret Zoph，Vijay Vasudevan 和 Quoc V . 乐。理解和简化一次性架构搜索。*ICML，2018 年。*

[2] 、、朱、宋涵。Proxylessnas:直接在目标任务和硬件上进行神经架构搜索。*ICLR，2019 年。*

[3] 弗朗切斯科·保罗·卡萨勒、乔纳森·戈登和尼科洛·弗西。概率神经架构搜索。*arXiv，2019 年。*

[4] 陈汉庭、、常旭、杨、刘传坚、石博新、、、齐天。学生网络的无数据学习。*ICCV，2019 年。*

[5] 陈新、谢凌熙、、齐天。渐进式飞镖:弥合网络连接存储在野外的优化差距。*arXiv，2019 年。*

[6] 楚湘湘、、徐瑞军、。费尔纳斯:重新思考权重分担神经架构搜索的评价公平性。*arXiv，2019 年。*

[7] Kalyanmoy Deb 和 Himanshu Jain。一种使用基于参考点的非支配排序方法的进化多目标优化算法，第一部分:解决带盒约束的问题。*TEC，2014。*

[8] 董劲东、安杰成、大程娟、魏伟和孙敏。设备感知渐进搜索帕累托最优神经架构。*ECCV，2018 年。*

[9] 董宣仪和杨熠。在四个图形处理器小时内寻找强健的神经架构。*CVPR，2019 年。*

[10] 托马斯·艾尔斯肯、简·亨德里克·梅岑和弗兰克·哈特。简单有效的卷积神经网络结构搜索。*ICLR，2018 年。*

[11] 托马斯·艾尔斯肯、简·亨德里克·梅岑和弗兰克·哈特。通过拉马克进化实现高效的多目标神经架构搜索。*ICLR，2019 年。*

[12] 罗斯·吉尔希克。美国有线电视新闻网。*ICCV，2015 年。*

[13] 艾登 n。戈麦斯、伊万·张、凯文·斯韦斯基、亚林·加尔和杰弗里·埃。韩丁。使用目标辍学学习稀疏网络。*arXiv，2019 年。*

[14] 龚新宇、常、、王。自动增长:生成性对抗网络的神经结构搜索。*ICCV，2019 年。*

[15] 阿里尔·戈登、埃拉·埃班、奥菲尔·纳库姆、陈博、杨天柱和爱德华·蔡。Morphnet:快速简单的深度网络资源受限结构学习。*arXiv，2017 年。*

[16] 郭建元、、、、、杨、、陈星浩、常旭。命中检测器:用于对象检测的分层三位一体架构搜索。*CVPR，2020 年。*

[17] 郭子超、、好远木、文恒、刘泽春、卫、。均匀采样的单通道单次神经网络结构搜索。*arXiv，2019 年。*

[18] 韩凯、王运合、韩曙、刘传坚、徐春净和常旭。行人属性识别的属性感知池。*IJCAI，2019。*

[19] 、王运合、齐天、郭建元、、常旭。Ghostnet:廉价运营带来更多功能。*arXiv，2019 年。*

[20] 何朝阳，叶海山，，。Milenas:通过混合层次重构进行有效的神经架构搜索。*CVPR，2020 年。*

[21] 明凯·何，乔治娅·格基奥沙里，彼得·多拉尔和罗斯·吉尔希克。面具 r-cnn。*ICCV，2017 年。*

[22] 、何、、、任、。用于图像识别的深度残差学习。*CVPR，2016 年。*

[23] 徐志鸿、舒、大、潘家瑜、俞、魏巍、张世杰。Monas:使用强化学习的多目标神经架构搜索。*arXiv，2018 年。*

[24] 黄高、刘庄、劳伦斯·范德马滕和基利安·Q·温伯格。密集连接的卷积网络。*CVPR，2017 年。*

[25] Itay Hubara，Matthieu Courbariaux，Daniel Soudry，Ran ElYaniv，Yoshua Bengio。二值化神经网络。*NIPS，2016 年。*

[26] 金烨勋、巴尔加瓦·雷迪、索金云和昌原·徐。Nemo:深度神经网络多目标优化的神经进化，速度和精度。*国际劳动和社会福利理事会，2017 年。*

[27] 邵会·林、荣融、陈超、陶大成、罗杰波。通过低秩分解和知识转移的整体 cnn 压缩。*T-PAMI，2019。*

[28] 林、嵇蓉蓉、李、吴永健、黄飞跃、张。通过全局和动态滤波器修剪加速卷积网络。*IJCAI，2018。*

[29] 林，荣嵇，颜，张，曹，叶企祥，黄飞跃，大卫·多德曼。通过生成性对抗学习实现最优结构化 cnn 剪枝。*CVPR，2019 年。*

[30] 刘晨曦、巴瑞特·佐夫、马克西姆·诺依曼、黄邦贤·施伦斯、魏华、李、、、黄宗智和。渐进式神经架构搜索。*ECCV，2018 年。*

[31] 刘韩啸、卡伦·西蒙扬、奥瑞尔·温雅斯、克丽珊莎·费尔南多和科莱·卡武科格鲁。高效架构搜索的分层表示。*ICLR，2018 年。*

[32] 刘韩啸、卡伦·西蒙扬和杨益明。飞镖:差异化架构搜索。*ICLR，2019 年。*

[33] 鲁治超、伊恩·维伦、毗湿奴·博德蒂、亚希什·德巴尔、卡尔扬莫·德布、埃里克·古德曼和沃尔夫冈·班扎夫。Nsga-net:一种用于神经架构搜索的多目标遗传算法。*GECCO，2019。*

[34] 马宁宁、张翔宇、郑海涛和孙健。Shufflenet v2:高效 cnn 架构设计的实用指南。*ECCV，2018 年。*

[35] Hieu Pham，Melody Y Guan，Barret Zoph，Quoc V Le，and

杰夫·迪恩。通过参数共享实现高效的神经架构搜索。*ICML，2018 年。*

[36] 芮捷全、董宣仪、、、朱、。自动识别:寻找一个能识别零件的 convnet 来重新识别人。*ICCV，2019 年。*

[37] 伊斯特班·雷亚尔、阿洛克·阿加尔瓦尔、黄雁萍和 Quoc V Le。用于图像分类器结构搜索的正则化进化。*AAAI，2019 年。*

[38] 伊斯特班·雷亚尔、雪莉·摩尔、安德鲁·塞尔、绍拉布·萨塞纳、尤塔卡·莱昂·苏马苏、谭洁、魁克·维·勒和阿列克谢·库拉金。图像分类器的大规模进化。*ICML，2017 年。*

[39] 马克·桑德勒、、朱梦龙、安德烈·兹莫吉诺夫和陈良杰。Mobilenetv2:反转残差和线性瓶颈。*CVPR，2018 年。*

[40] 明珠沈，，，。寻找精确的二元神经结构。*ICCVW，2019。*

[41] 、王运合、徐佳、韩凯、陈汉庭、、齐天、常旭。不成对图像翻译的协同进化压缩。*ICCV，2019 年。*

[42] 宋德华、常旭、徐佳、陈依依、徐春净、王运合。图像超分辨率的有效剩余密集块搜索。*AAAI，2020 年。*

[43] 克里斯蒂安·塞格迪、、·贾、皮埃尔·塞马奈、斯科特·里德、德拉戈米尔·安古洛夫、杜米特鲁·埃汉、文森特·范豪克和安德鲁·拉比诺维奇。盘旋着越陷越深。*CVPR，2015 年。*

[44] 谭明星、、彭若明、维贾伊·瓦苏德万、Quoc V·乐。Mnasnet:面向移动的平台感知神经架构搜索。*CVPR，2018 年。*

[45] 王运合、常旭、徐春净、许超和
大成道。高效卷积神经网络的学习通用滤波器。*NIPS，2018 年。*

[46] 王运合、常旭、尤山、陶大成、许超。Cnnpack:在频域中打包卷积神经网络。*NIPS，2016 年。*

[47] 吴，戴，张培昭，王，，吴，田远东，彼得·瓦依达，·贾，库尔特·库特泽。Fbnet:通过可微分神经结构搜索的硬件感知的高效 convnet 设计。*CVPR，2019 年。*

[48] 谢凌熙和艾伦·尤尔。遗传 cnn。*ICCV，2017 年。*

[49] 谢思睿、、郑、、。随机神经架构搜索。*ICLR，2019 年。*

[50] 云阳熊，罗纳克·梅塔，维卡斯·辛格。资源受限的神经网络体系结构搜索。*ICCV，2019 年。*

[51] 徐艺兴、、陈汉庭、、、陶大成、常旭。正-云上无标签压缩。*神经科，2019 年。*

[52] 杨映真、于佳卉、内博伊沙·乔吉奇、军欢和托马斯·s。黄。Fsnet:通过过滤器总结压缩深度卷积神经网络。*ICLR，2020 年。*

[53] 杨、、刘传坚、陈汉庭、、石伯鑫、、常旭。乐高网络:具有乐高滤波器的高效卷积神经网络。*ICML，2019 年。*

[54] 克里斯·英、艾伦·克莱因、伊斯特班·雷亚尔、埃里克·克里斯蒂安森、凯文·墨菲和弗兰克·赫特。Nas-bench-101:走向可再现的神经架构搜索。*ICML，2019 年。*

[55] 于佳卉和托马斯·黄。Autoslim:走向一次性架构搜索频道号。*arXiv，2019 年。*

[56] 于佳卉和托马斯。黄。可普遍缩小的网络和改进的训练技术。*ICCV，2019 年。*

[57] 、林杰·杨、、杨建超和托马斯·黄。纤细的神经网络。*ICLR，2019 年。*

[58] 赵衷、阎俊杰、吴伟、京少和刘成林。实用的分块神经网络体系结构生成。*CVPR，2018 年。*

[59] 周燕琪、易卜拉希米、塞尔詹·阿尔克、于浩南、刘海荣和葛瑞格·迪亚莫斯。资源节约型神经建筑师。*arXiv，2018 年。*

[60] Barret Zoph 和 Quoc V Le。基于强化学习的神经结构搜索。*ICLR，2017 年。*

[61] 巴雷特·佐夫、维贾伊·瓦苏德万、黄邦贤·史伦斯和魁克·维勒。学习用于可扩展图像识别的可转移架构。*CVPR，2018 年。*