

稀疏 CNN 架构搜索 (SCAS)

Yeshwanth V *, Ankur Deshwal *, Sundeep Krishnadasan *, Seungwon Lee †, 俊昊宋 †

三星研发研究所印度-班加罗尔; †韩国三星高级技
术研究院 (SAIT) {yesh.v, a.deshwal, sundeep.k, seungw.lee,
joonho71.song}@samsung.com

抽象的

深度神经网络的出现彻底改变了计算机视觉。然而,设计这种具有高精度和低计算要求的模型是一项艰巨的任务,需要大量的人类专业知识。神经架构搜索的最新进展使用深度强化学习、进化方法、梯度下降、超网络等各种方法来自动生成高精度的神经网络。然而,这种生成的模型的大尺寸限制了它们的实际使用。最近关于彩票假说的发现表明存在稀疏子网络(中奖彩票),其精度可以与原始密集网络相媲美。在本文中,我们**提出了一种利用深度卷积神经网络(CNN)固有的冗余来指导生成稀疏 CNN 模型(以找到具有中奖彩票的架构)而不会显着降低准确性的方法**。我们在 CIFAR-10、CIFAR-100 和 MNIST 数据集上使用不同的 NAS 方法评估我们提出的方法。我们的结果表明,模型大小减少了2 倍到12 倍,MAC 操作数量减少了2 倍到19 倍,精度下降不到 1%。

已付费研究生成模型的大小。许多令人印象深刻的结果是以模型为代价获得的,这些模型比手工制作的版本大一个数量级。

为了促进神经网络在移动和嵌入式设备中的快速部署,自动化设计紧凑且在推理时高效执行的神经网络架构非常重要。

神经架构搜索。MetaQNN [3]是 NAS 领域的早期工作之一。它使用具有贪婪探索策略的 Q 学习和经验回放来搜索神经网络。[4]的作者提出使用循环神经网络(RNN)的策略梯度方法,称为控制器,对策略进行编码。它对模型(称为子模型)进行采样,并在训练后使用它们的验证准确性来缩放梯度以改进策略。

上面介绍的工作遭受了极高的计算和内存资源要求,例如[4]使用 450 个 GPU 3-4 天(即 32,400-43,200 GPU 小时)以达到报告的准确度水平。

为了应对高资源需求的问题,ENAS [5]在**基于策略梯度的 RL 设置中使用控制器 RNN 采样的候选模型的参数共享**。ENAS 将网络架构搜索的资源需求降低了3 个数量级。之前针对 NAS 提出了 RL 的替代方法。[6]中提出的工作应用 HyperNet [7]来训练辅助模型,该辅助模型动态生成具有可变架构的主模型的权重。与基于 RL 的技术相比,在这种情况下搜索空间是有限的。在[8]中,作者使用连续松弛表示卷积块之间的连接。然后它使用梯度下降进行架构搜索。最近的工作包括 MnasNet [9]、RENA [10]使用多种方法来实现具有计算约束的最先进的体系结构。

索引词 计算机视觉、AutoML、神经架构搜索、稀疏

一、简介

与从原始输入数据中提取手工制作的特征相比,卷积神经网络(CNN)促进了自动表示学习[1]。然而,设计最先进的神经网络架构需要大量的人类专业知识。一组称为神经架构搜索(NAS)的技术的最新进展旨在为给定数据集和成本函数[2]自动设计复杂的神经网络。

NAS 是一个快速发展的领域。不同的优化技术,**例如强化学习、梯度下降、超网络、进化方法等**,已被用于自动生成在视觉和文本数据集上具有接近最先进结果的模型。然而很少有人关注

N2N Learning [11]采用替代方法使用网络蒸馏从头开始生成神经架构。[12]中介绍的工作使用 RL 来预测输入网络每一层中可能的细粒度稀疏度。

尽管方法[11]和[12]侧重于优化计算和能量需求,但这些技术错过了从头开始生成模型的机会。这将它们的适用性限制在最先进的情况下

学习模型已经存在。

Sparse [13]提出了一种用于生成具有稀疏性的神经网络的多目标 BO 方法。如本文所示,我们的方法可以应用于多种优化技术。

修剪。修剪是一种流行的方法,用于去除冗余以及提高深度神经网络的泛化能力。在 CNN 出现在嵌入式设备的现代应用之后,人们越来越关注优化CNN 模型以适应有限资源环境。修剪冗余连接是神经网络模型压缩和加速中的常用方法之一。 [14]和[15]中提出的工作使用基于迭代阈值的修剪和再训练来学习权重并同时修剪不重要的权重。 [16]的作者通过分别在过滤器级别和组级别进行修剪来寻求一种更加结构化的修剪方式。

彩票假设。 [17]提供了一个关于大型密集网络如何具有稀疏子网络的新视角,如果发现这些网络可以有效地训练以达到与大型密集模型在相同数量的训练迭代中达到的相同精度。修剪方法用于发现这种稀疏子网。此外,彩票假设被扩展为一个猜想,即密集网络具有更多可能的子网络,从中可以发现中奖彩票,因此可以比训练稀疏网络更容易训练。

贡献。在本文中,我们探索了生成高性能稀疏神经网络的范围,即通过引入约束以支持稀疏架构来使用多种 NAS 技术找到具有中奖彩票的架构。

我们使用[5]、[6] 和 [8] 中介绍的针对 NAS 的不同方法来评估我们的方法。

我们的贡献可以总结如下:

- 我们提出了一种新技术来指导 NAS 方法生成具有高参数稀疏性的模型 (以找到具有中奖彩票的架构) ,同时保持与最先进模型相当的准确性。
- 我们研究了流行的 NAS 技术在 CIFAR-10、CIFAR-100 和 MNIST 数据集上的相对有效性。
- 我们通过使用基于正交方法的NAS 技术进行广泛实验来评估我们方法的稳健性。即 ENAS (使用深度强化学习) 、DARTS (使用梯度下降)和 SMASH (使用 HyperNetworks) 。

参数稀疏性导致内存需求的成比例改善以及可以利用这种稀疏性的 AI 加速器/硬件架构的性能。我们注意到,虽然由于 CNN在计算机视觉中的核心作用,我们将重点放在 CNN 上,但该技术足够通用,可以扩展到其他类型的深度神经网络作品。此外,由于我们的方法足够通用,可以一起工作

使用基于不同原理的 NAS 技术,它可以以最小的努力扩展到其他 NAS 方法。

2. 稀疏 CNN 架构搜索

在本节中,我们将讨论修改不同 NAS 技术以生成稀疏神经网络架构的方法。一般的想法是在被搜索的模型中引入稀疏性,从而将搜索过程引向稀疏但高性能的架构。我们方法的流程如图 1 所示。

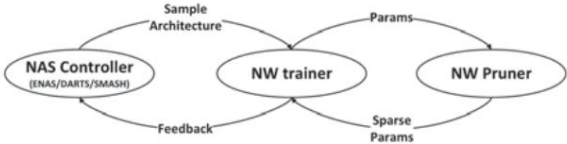


图1. SCAS 的关键组件。

NAS 控制器节点托管 NAS 方法。对于ENAS,这是一个 RNN 控制器。对于 DARTS,这是架构编码的权重。在 SMASH 的情况下,这是为子网络生成权重的 HyperNet。在控制器的每个训练步骤中,对子模型进行采样并将其提供给训练器节点。当训练器训练子模型时,它与修剪器交错,修剪器通过将权重修剪到预设水平来引入稀疏性。向 NAS 控制器提供反馈, NAS 控制器据此学习生成更好的架构。在 ENAS 中,这是通过使用 REINFORCE 算法[18]更新 RNN 控制器来完成的,并以子模型的验证准确性作为奖励。

在 DARTS 的情况下,使用梯度下降更新编码架构的权重,以最大化子模型的验证精度。在 SMASH 的情况下,由于权重是由 HyperNet 生成的,因此它们没有在训练器中训练。修剪器将生成的权重修剪到指定的目标稀疏度,并将此修剪后的子模型的训练误差反向传播到 HyperNet。

在第 2.1、2.2 和 2.3 节中,我们分别介绍了应用于 ENAS [5]、DARTS [8] 和 SMASH [6] NAS 技术的方法的实现细节。最后,在第 2.4 节中,我们概述了在每种 NAS 技术中用作交错子例程的一般修剪方法。

2.1. 易纳斯

在本节中,我们将解释我们建议的更改,这些更改指示ENAS [5] 搜索稀疏网络。

ENAS 是基于强化学习的 NAS 方法。ENAS 由自动回归模式下的 LSTM 网络设置组成。每集生成一个 CNN 模型,对其进行评估并将达到的准确度级别用作奖励

更新 LSTM 网络以生成更好的模型。下面介绍了 ENAS 使用的两种构建搜索框架的方法（详情请参阅[5]）。我们用这两种方法展示结果。

(a) 宏搜索空间:在这种情况下,设计整个网络被视为 RL 任务, RL 代理的搜索空间包括生成网络层的配置。(b) 微搜索空间:在微搜索的情况下,目标是设计一个计算单元。ENAS 生成两种类型的单元,称为卷积单元和缩减单元。为了生成完整的网络,“N”个卷积单元后跟一个缩减单元。

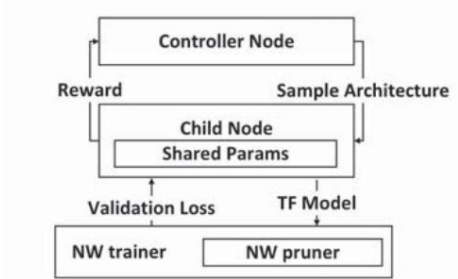


图2. 使用 ENAS 的 SCAS 架构

算法 1 ENAS 上的 SCAS

```
1:初始化控制器网络 $\theta$ 和子网络  $\omega$ 的共享参数
2: loop
3:   Fix controller's policy  $\pi(\theta)$  and sample m
4:   networks Create computation graph using shared
5:   parameters  $\omega$  for an epoch of training data do更新
6:     权重  $\omega$  （使用蒙特卡洛估计）
7:     将子模型修剪到目标稀疏度  $s$ 。修复修剪后的权重  $\omega(s)$ 
8:
9:
10:  更新 $\pi(\theta)$ 以最大化预期奖励 $E_m \pi(\theta)[R(m; \omega(s))]$  11:结束循环12:
    来自经过训练的控制器网络的样本架构13:从头开始训练它们以及稀疏
    性约束
```

在随后的文本中,我们描述了我们ENAS 提出的更改以生成稀疏模型（算法 1）。图 2 显示了我们的设置。和 ENAS 一样,controller node 以 LSTM 模型的形式封装了 policy。为了开始迭代,控制器节点对 m 个 CNN 网络架构进行采样,并将它们以文本形式传递给子节点。在 RL 术语中,这构成了 m rollouts。

然后子节点为采样的架构构建计算图。这

采样模型的共享参数由网络训练器训练。然后它根据目标修剪级别修剪所有网络。然后在数据集的子集上重新训练模型,以恢复由于修剪而导致的准确性下降。每个模型 m_i 的交叉熵损失 $L(m_i, \omega)$ 是在验证数据集上计算的,其中 ω 表示模型的共享参数。

上面计算的梯度作为信号提供给LSTM,以使用 REINFORCE [18]算法更新其参数 θ 。为了减少由于噪声梯度引起的变化,我们使用移动平均基线,这是稳定和加速 REINFORCE 训练的标准做法。算法 1 总结了该方法。

2.2.飞镖

DARTS [8]通过在连续空间中表示架构并使用梯度下降对架构及其权重进行联合优化来实现 NAS。

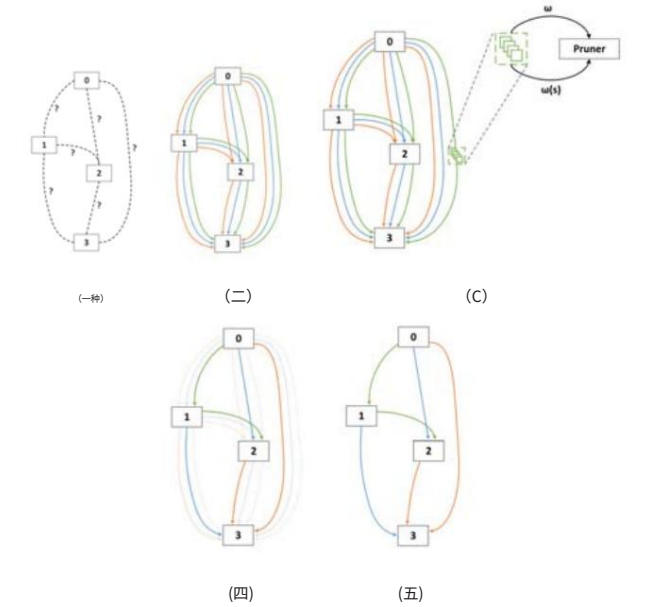


图3. 使用 DARTS 的 SCAS 架构:(a)边缘操作最初是未知的。(b)通过用混合操作替换每条边来不断放松搜索空间。(c) 剪枝子程序与网络权重优化方法交错。(d) ω 和 α 的联合优化。(五)

从学习的 α 参数中提取最终架构（最好以彩色显示）。

架构编码 α 和所有操作中的权重 ω 作为双层优化问题共同学习,其中 α 作为上层变量最小化验证损失 L_{val} , ω 作为下层变量最小化训练损失 L_{train} （参考方程（3）,（4）在[8]）。为了驱动DARTS 生成高性能的稀疏网络,我们在

算法 2 DARTS 上的 SCAS

```

1: 创建一个由  $\alpha(i, j)$  参数化的混合运算  $o^-(i, j)$ 
   对于每条边  $(i, j)$  2: 循环
   通过冻结  $\alpha$  来更新权重
3:  $\omega = \omega - \xi \nabla \omega L_{\text{train}}(\omega, \alpha)$ 

4: 将权重  $\omega$  修剪到给定的目标稀疏度  $s_o$ 。
5: 通过冻结稀疏更新架构参数  $\alpha$ 
   权重  $\omega(s)$  和降序  $\nabla \alpha L_{\text{val}}(\omega(s), \alpha)$  6: 结束循环 7: 用最可能的
   操作替换  $o^-(i, j)$ 。  $o^-(i, j) = \arg \max_{o \in O} \alpha(i, j, o)$  对于每条边
    $(i, j)$ 

```

在每个操作的权重中引入稀疏性。令 $\omega(s)$ 为稀疏权重, 其中 s 表示引入的稀疏程度, 这意味着权重 ω 的 s 百分比为零。最终的优化问题简化为:

$$\min_{\alpha} L_{\text{val}}(\omega(\alpha, s), \alpha) \quad (1)$$

$$s(\alpha, s) = \arg \min_{\omega} L_{\text{train}}(\omega(s), \alpha) \quad (2)$$

这种双层优化通过近似迭代过程求解, 其中 ω 和 α 通过分别在权重和架构空间中的梯度下降步骤之间交替进行优化。在搜索结束时, 采用最有可能的操作来构建最终架构。通过对权重引入稀疏性约束, 我们可以针对稀疏权重 $\omega(s)$ 优化架构参数 α , 从而实现稀疏但高性能的模型。将 SCAS 应用于 DARTS 优化过程的方法如图 3 所示, 并在算法 2 中进行了总结。

2.3. 粉碎

SMASH [6] 对 NAS 采取了不同的方法。它没有使用控制器来生成架构, 而是使用辅助 HyperNet, 它根据作为输入给出的架构为模型生成权重。

为了生成稀疏架构, 我们建议将 SMASH 方法与一个子程序交织在一起, 该子程序会修剪 HyperNet 生成的权重。因此, 对尽管有稀疏性约束但表现良好的网络执行了候选架构的相对验证准确性的评估。该方法在算法 3 中进行了概述。在我们提出的方法中, HyperNet 被训练以生成显示对修剪具有弹性的权重。因此, 在基于相对验证准确性进行比较时, 在稀疏约束下工作得更好的架构得分更高。

2.4. 修剪方法

我们使用逐渐修剪算法修剪最小幅度的权重以达到预设的网络稀疏度水平

算法 3 SMASH 上的 SCAS

```

要求:  $R_c$  - 所有候选架构的空间,  $s$  - 目标稀疏性
确保:  $c^*$  - 在给定的稀疏度水平下表现最好的模型
1: 初始化 HyperNet 权重  $H$  2: 循环

3: 对随机架构  $c$  和小批量  $x_i$  进行采样
4: 从 HyperNet 中获取架构权重  $\omega = H(c)$ 
5: 将权重修剪为给定的目标稀疏度
6: 得到训练误差  $E_t = f_c(\omega(s), x_i)$ , 并使用反向传播更新  $H$  7: 结束循环 8: 循环

9: 样本随机架构  $c$ 
10: 从 HyperNet  $\omega(s) = H(c)$  中获取架构权重
11: 将权重修剪为给定的目标稀疏度并获得稀疏权重  $\omega(s)$ 

12: 在验证集上评估错误  $E_v = f_c(\omega(s), x_v)$  13: 结束循环

14: 以最小的相对验证错误修复架构
   并与目标稀疏度  $s$  一起训练

```

在 [19] 的实施之后。对于要修剪的每一层, 维护一个与权重张量形状相同的二进制掩码变量。此掩码指定是否修剪相应的权重。这使得能够在反向传播期间有条件地阻塞梯度流。因此, 仅更新未修剪的权重。在 ENAS 的情况下, 我们遵循 [19] 中介绍的实现, 它是 Tensorflow 库的扩展。为了在 DARTS 和 SMASH 中引入修剪, 我们在 PyTorch 中使用自定义实现的修剪子例程。

这个想法是为每一层保持一个阈值 th_l 。大小小于该阈值的权重将被修剪。随着训练的进行, 这个阈值会根据设定的目标稀疏度和达到的稀疏度而增加或减少。这等效于 [19] 中描述的过程, 即根据权重的大小对每一层中的权重进行排序, 并在随后的迭代中逐渐移除一小部分权重, 直到达到目标稀疏度。

3. 实验和结果

我们将我们提出的方法应用于 CIFAR-10、CIFAR-100 和 MNIST 数据集。这些数据集在 NAS 文献中被广泛使用, 因此有助于对已发布的结果进行基准测试。我们根据 CIFAR-10 数据集上的实验解释了关键见解, 并总结了其他人的结果。

我们使用 [5]、[8] 和 [6] 的实现作为我们的基线。默认超参数、输入转换、数据增强和正则化技术与相应 au 的原始实现没有变化

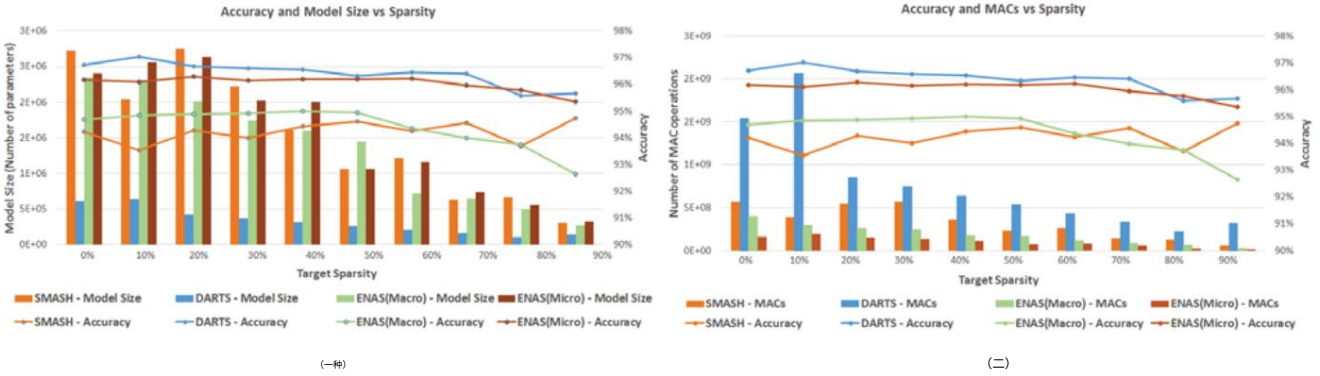


图4. (a) CIFAR-10 数据集上不同 NAS 模型的精度和模型大小与目标稀疏度的趋势。(b) CIFAR-10 数据集上不同 NAS 模型的精度和乘法和累加 (MAC) 运算与目标稀疏度的趋势（最佳彩色视图）。

表1. SCAS 使用不同的 NAS 技术在不同的数据集上实现了模型大小和 MAC 操作的减少,精度下降 <1%。基线列显示未应用稀疏约束时的模型大小（以百万个参数为单位)和 MAC操作数（以百万个为单位）。下一列显示了 SCAS 发现的最佳模型,精度下降不到1%。Redn 列显示实现的减少。

数据集 NAS 方法	型号尺寸		MAC操作	
	基线 <1% 下降 Redn	基线 <1% 下降 Redn	基线 <1% 下降 Redn	基线 <1% 下降 Redn
CIFAR10	ENAS-宏 2.35M 0.50M	ENAS-微 2.40M 0.33M	4.7x 395.35m	60.73m 6.5x 7.3x 162.84m
	DARTS 0.61M 0.16M	SMASH 2.72M 0.30M	20.41m 8.0x 3.9x	1542.26m 329.70m 4.7x 9.1x
	CIFAR100 ENAS-宏 237.63M 11.40M	ENAS-微 237.63M 11.40M	567.43m 57.90m 57.90m	9.8x 2.0x 314.94m
	0.36M 0.26M	MASH 0.36M 0.26M	109.91M 57.77M	1.9x 12.3x 759.66M 115.56M
MNIST	ENAS-微距 2.48M 0.21M		11.6x 221.46M 23.36M	9.5x 11.1x 229.26M
	ENAS-Micro 2.84M 0.26M	DARTS 0.62M	11.89M 6.4x 1620.15M	312.35M 5.2xM 451.46M
	0.10M	SMASH 2.58M 0.24M		
				19.3倍

雷神。对于这些实现中的每一个,我们应用前面提到的修剪方法,目标稀疏度的不同级别从 0% 到 90%,以 10% 的固定间隔。

图 4a 展示了随着稀疏性约束从 0%增加到 90%,通过不同 NAS 方法为 CIFAR-10 数据集找到的最佳模型的准确性和模型大小的趋势。这里的关键观察结果是准确率的下降不像模型大小的下降那么急剧,这意味着我们可以使用我们的方法自动设计高性能的稀疏神经网络架构。

在不同的 NAS 方法中,我们观察到 SCAS DARTS 提供了性能最好的模型,在评估方法中参数少得多。另一方面,如果我们检查 Multiply 和 Accumulate 的数量趋势

(MAC) 操作在图 5b 中,我们可以观察到 SCAS DARTS 是所有评估方法中最昂贵的,而 SCAS-ENAS (micro) 优于其他方法。这可以归因于 NAS 方法对搜索空间的选择,例如,在比较具有相同卷积参数大小的两个模型时,具有更大特征图的模型需要更多的计算,而卷积参数的数量相同。DARTS 中使用的搜索空间使用较少的缩减操作,因此尽管参数数量最少,但会产生更大的特征图和更多数量的 MAC 操作。

我们现在在 1 中总结了 SCAS 在不同数据集 (MNIST,CIFAR-10、CIFAR-100)上的有效性。基线指标对应于各自找到的最佳模型

具有默认超参数且无稀疏约束的 NAS 技术。接下来的列显示了最佳模型中的参数数量和 MAC 操作数量,该模型具有稀疏性约束,验证精度下降不到 1%。

与具有宏搜索空间设置的 SCAS ENAS相比,我们观察到模型大小和 MAC 操作分别减少了11.6 倍和9.5 倍。与具有微搜索空间的 SCAS-ENAS 相比,观察到11.1X和19.3X的减少。同样,与 SCAS-DARTS 相比,观察到 6.4X和5.2X的改进。

在 SCAS-SMASH 的情况下,观察到参数和 MAC减少了10.8 倍和14.4 倍。

我们在比较不同数据集上下文中的绝对模型大小时观察到,SCAS-DARTS 始终如一地实现了最小模型。此外,除了 CIFAR-10 的情况外,SCAS-DARTS 还产生了具有最小计算要求的模型。

4.结论

我们提出了一种简单有效的技术,用于指导不同的 NAS 方法生成相当稀疏的 CNN模型(具有中奖票的架构),准确率下降不到1%。我们比较了我们的方法以及不同的 NAS 方法在 CIFAR-10、CIFAR-100 和 MNIST 数据集上的有效性,并展示了我们的见解。

自动生成优化的 CNN 模型有助于将 CNN 模型广泛部署到嵌入式设备(包括智能手机、手表和物联网设备)中的各种应用。

5. 参考文献

- [1] Alex Krizhevsky,Ilya Sutskever 和 Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks”,神经信息处理系统进展 25,F. Pereira,CJC Burges,L. Bottou 和 KQ Weinberger,编辑,第 1097-1105 页。柯伦联合公司,2012 年。
- [2] Thomas Elsken,Jan Hendrik Metzen 和 Frank Hutter, “神经架构搜索:一项调查”,arXiv 预印本arXiv:1808.05377,2018 年。
- [3] Bowen Baker,Otkrist Gupta,Nikhil Naik 和 Ramesh Raskar, “使用强化学习设计神经网络架构”,CoRR,卷。 abs/1611.02167, 2016。
- [4] Barret Zoph,Vijay Vasudevan,Jonathon Shlens 和 Quoc V. Le, “学习可扩展图像识别的可迁移架构”,CoRR,卷。 abs/1707.07012, 2017。
- [5] Hieu Pham,Melody Y. Guan,Barret Zoph,Quoc V. Le和 Jeff Dean, “通过参数共享进行高效的神经架构搜索”,CoRR,卷。 abs/1802.03268, 2018。
- [6] Andrew Brock,Theodore Lim,James M. Ritchie 和 Nick Weston, “SMASH:通过超网络进行一次性模型架构搜索”,CoRR,卷。 abs/1708.05344, 2017。
- [7] David Ha,Andrew M. Dai 和 Quoc V. Le, “超网络有效”,CoRR,卷。 abs/1609.09106, 2016。
- [8] Hanxiao Liu,Karen Simonyan 和 Yiming Yang, “DARTS:可区分架构搜索”,CoRR,卷。 abs/1806.09055, 2018。
- [9] Mingxing Tan,Bo Chen,Ruoming Pang,Vijay Vasudevan 和 Quoc V. Le, “Mnasnet:移动平台感知神经架构搜索”,CoRR,卷。 abs/1807.11626, 2018。
- [10] Yanqi Zhou,Siavash Ebrahimi,Sercan Ömer Arik,Hao nan Yu, Hairong Liu 和 Greg Diamos, “资源高效的神经架构师”,CoRR,卷。 abs/1806.07912, 2018。
- [11] Anubhav Ashok,Nicholas Rhinehart,Fares Beainy 和Kris M. Kitani, “N2N 学习:通过策略梯度强化学习进行网络到网络压缩”,CoRR,卷。 abs/1709.06030, 2017。
- [12] Yihui He 和 Song Han, “ADC:使用强化学习进行自动深度压缩和加速”,CoRR,卷。 abs/1802.03494, 2018。
- [13] Igor Fedorov,Ryan P Adams,Matthew Mattina 和 Paul Whatmough, “稀疏:在资源受限的微控制器上搜索 CNN 的稀疏架构”,神经信息处理系统进展,2019 年,第 4978-4990 页。
- [14] Song Han,Jeff Pool,John Tran 和 William Dally, “学习有效神经网络的权重和连接”,神经信息处理系统进展 28,C. Cortes,ND Lawrence,DD Lee,M. Sugiyama 和 R. Garnett 编着,第 1135-1143 页。柯伦联合公司,2015 年。
- [15] Song Han,Huizi Mao 和 William J Dally, “深度压缩:通过修剪、训练量化和哈夫曼编码压缩深度神经网络”,arXiv预印本 arXiv:1510.00149, 2015 年。
- [16] Vadim Lebedev 和 Victor Lempitsky, “Fast convnets using group-wise brain damage”,载于IEEE 计算机视觉和模式识别会议论文集,2016 年,第 2554-2564 页。
- [17] Jonathan Frankle 和 Michael Carbin, “彩票假设:训练修剪神经网络”,CoRR,卷。 abs/1803.03635, 2018。
- [18] Richard S Sutton,David A McAllester,Satinder P Singh和 Yishay Mansour, “使用函数近似进行强化学习的策略梯度方法”,神经信息处理系统进展,2000 年,第 1057-1063 页。
- [19] Michael H. Zhu 和 Suyog Gupta, “修剪还是不修剪:探索修剪对模型压缩的功效”,2018 年。