

一种进化深度的模因算法

图像中的卷积神经网络分类

1st Junwei Dong
重庆大学计算机学院 中国重
庆 JWDong@cqu.edu.cn

2nd Liangjie Zhang
重庆大学计算机学院 中国重
庆 LJZhang@cqu.edu.cn

第三届重庆大学
侯博宇计算机学院 中国重庆
byhou@cqu.edu.cn

第四届梁峰重庆
大学计算机学院 中国重庆
liangf@cqu.edu.cn

摘要 由于进化算法 (EA) 对问题制定具有鲁棒性且易于使用,因此近年来人们对设计用于自动神经架构搜索的 EA 的兴趣与日俱增。特别是,EvoCNN 是最近提出的一种进化算法,用于自动配置用于图像分类的深度卷积神经网络 (CNN)。

在广泛使用的图像分类任务上,其有效性已针对 22 种现有的 CNN 配置算法得到证实。然而,尽管这种方法取得了成功,但我们注意到这种方法存在一些局限性。

例如,只考虑链式结构网络进行进化。此外,决策变量很多,计算量大。在本文中,我们通过提出一种模因算法 (MA) 来研究进化神经结构搜索,以解决上述问题。特别地,首先,除了进化链式结构网络外,局部搜索是为多分支网络搜索而设计的。接下来,为了减少优化的网络参数,我们只关注卷积层的架构搜索。此外,根据文献中最近的假设,网络评估仅基于我们提出的 MA 中的早期训练过程进行。为了确认所提出方法的有效性,在常用的图像分类基准上针对 NAS 的 EvoCNN 进行了全面的实证研究。

索引词 模因算法,神经结构搜索,图像分类,卷积神经网络。

一、引言

近年来,深度卷积神经网络 (CNN) 在许多图像分析任务中取得了相当大的成功,例如图像分类 [1]-[5]、目标检测 [6]-[8] 和语义分割 [9]-[11] 等。CNN 可以通过交替的卷积层和池化层从特定基准学习低/中/高级特征。

与具有相似大小层的传统前馈神经网络相比,CNN 的连接和参数要少得多 [1]。在文献中,随着 CNN 的发展,已经提出了许多成功的架构,例如 VGGNet [2]、GoogleNet [3]、ResNet [4] 和

DenseNet [5] 等。此外,这些设计巧妙的网络进一步减少了大量的连接和参数,并在许多现实世界的应用中取得了卓越的学习性能 [1]、[6]、[11]。然而,尽管这些不同的 CNN 架构取得了巨大的成功,但其中大部分是由人类专家手动开发的,这是一个耗时且容易出错的过程。由于不同的应用程序或任务可能需要独特的 CNN 架构,因此需要开发可以自动化 CNN 架构工程过程的智能方法。

在文献中,为了自动化 CNN 的架构配置,人们对神经架构搜索 (NAS) 以及许多优化和学习方法越来越感兴趣,包括贝叶斯优化 (BO)、强化学习 (RL) 和梯度-基于方法等,已经提出来解决这个问题。例如,Bergstra 等人。[12] 提出了一种支持自动超参数优化的元建模方法,在 [13] 中,Mendoza 等人。开发了一个全自动调整的神经网络,赢得了 2015 年官方 AutoML 人类专家赛道与人类专家的竞争。此外,Kandasamy 等人。[14] 提出了一种基于高斯过程的 NAS BO 框架,它可以有效地为 MLP 和 CNN 找到有竞争力的架构。此外,Zoph 和 Le。[15] 将强化学习纳入循环网络,以最大限度地提高生成架构的预期准确性。贝克等。[16] 训练了一个代理,使用带有贪婪探索策略的 Q 学习和经验重播来发现具有改进性能的设计。蔡等。[17] 还将 NAS 作为一个具有强化学习的顺序决策过程来处理,并使用元控制器通过重用这些架构的权重来探索架构空间。然而,强化学习方法的成功主要基于庞大的计算资源。最近,刘等人。[18] 提出了 DARTS,它可以通过以可区分的方式制定 NAS 来大大减少计算资源的消耗,从而允许梯度下降以有效搜索 CNN 架构。此外,Hundt 等人。[19] 提出了 sharpDARTS 进一步加速搜索

通过将 SharpSepConv 块和 Consine Power Annealing 学习率计划纳入 DARTS 来处理 NAS。

除了上面讨论的优化方法外,进化算法 (EA) 在探索 NAS 的搜索空间方面也起着重要作用,因为与强化学习和梯度下降等现有算法相比,它不需要大量的问题领域知识。基于方法。将 EA 用于 NAS 的早期方法是拓扑的神经进化 [20],它只进化了简化的神经网络拓扑和权重。随着硬件的改进,EA 现在可以为 NAS 制作复杂的架构。例如,Esteban 等人。 [21] 在没有精心设计的交叉算子的情况下使用遗传算法 (GA) 来搜索具有 250 个 GPU 的高级架构,以对抗手动设计的网络。为了设计更多可用的搜索空间,Xie 等人。 [22] 提出了一种新的编码方法,以固定长度的二进制字符串表示每个架构,并使用遗传算法 (GA) 找到高质量的解决方案。此外,刘等人。 [23] 在遗传算法中铸造了一种新的层次表示,获得了最先进的网络架构结果。

最近,Esteban 等人。 [24] 设计了一个具有年龄属性的锦标赛选择进化算法,以使每一代中的年轻个体受益,与专家设计的最先进模型相比,该算法获得了极具竞争力的性能。尽管这些方法获得了令人鼓舞的性能,但它们需要大量的计算资源。因此,孙等人。 [25] 提出了一种新的基于 GA 的进化深度卷积神经网络算法 (EvoCNN),可以大大降低计算资源的成本。 EvoCNN 采用链式结构神经网络的搜索空间,如图 1 所示 (见图 1 左图),链式结构神经网络架构 A 由一系列 n 层组成,其中 i 第 Li 层仅接收来自 Li-1 层的输入, $A = L_n \circ \dots \circ L_1 \circ L_0$ [26]。此外,EvoCNN 在 Fashion benchmark、MBI benchmark、MRD benchmark 和 MRDBI benchmark 等一系列基准数据集上取得了显著的成功。然而,值得注意的是,EvoCNN 存在几个缺点:1) EvoCNN 中发现的架构都是链式结构网络,这可能会导致 [4] 中提到的退化问题; 2) 最终解中可能存在一些全连接层,由于参数量大,容易出现过拟合 [27]; 3) 性能估计仍然是计算昂贵的。

为了解决上述问题,在本文中,我们提出了一种基于模因算法 (MA) 的神经网络 MA Net,它将局部搜索集成到 EvoCNN 的全局搜索中。特别是,为了解决第一个问题,MA-Net 采用如图 1 所示的精心设计的局部搜索策略 (见图 1 中的右图) 来探索多分支网络的空间。此外,为了减少参数过多引起的过拟合的影响,MA-Net 将全连接层替换为卷积层,大幅减少了参数数量。

网络中的参数最后,为了减少时间消耗,MA-Net 基于 [28] 中提出的评估假设,仅在早期训练过程中比较网络。为了确认我们提出的 NAS 方法的有效性,使用 EvoCNN 中考虑的数据集进行了全面的实证研究,例如 Fashion benchmark、MBI benchmark、MRD benchmark、MRDBI benchmark。

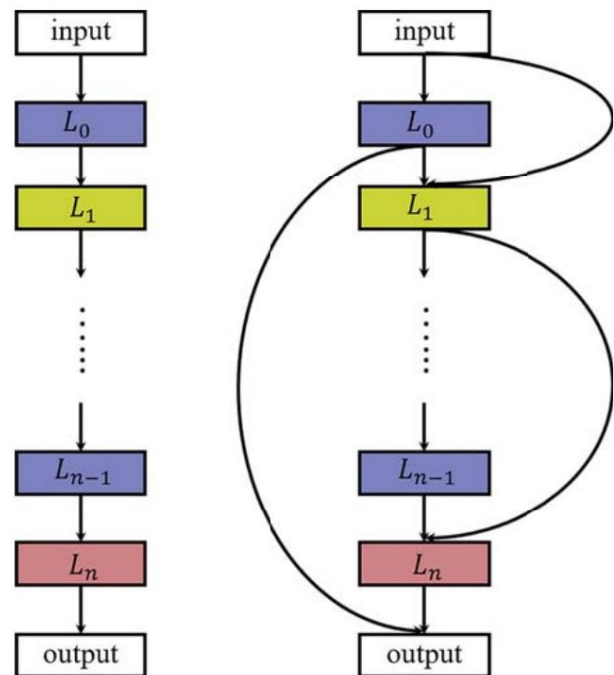


图 1. 两个不同的搜索空间。图中的一个节点代表一个操作,可以是卷积层或池化层,不同的颜色表示不同的设置或类型。每个连接都显示了两个节点之间的信息流。左图:EvoCNN 中使用的链式结构空间。

右图:我们提出的算法中使用的多分支空间。

本文的其余部分安排如下。第二节简要介绍了 NAS,并对相关工作进行了回顾。接下来,所提出的算法在第三节中介绍。然后,实验研究和获得的结果显示在第四节中。最后,第五节给出了这项工作的结论。

二. 神经结构搜索

NAS 是自动化架构工程的过程,可以看成是 Automated Machine Learning (AutoML) [26]。在 NAS 中,一般存在三个核心问题:搜索空间、搜索策略和性能评估策略。

(1) 搜索空间包含 NAS 方法的可行解集,在图像分类中可分为全局搜索空间和基于单元的搜索空间 [26]。(2) 搜索策略详细说明如何探索定义的搜索空间,快速找到性能良好的架构。现有的搜索策略主要有强化学习 (RL) [15]、[16]、[29]、[30]、进化算法 (EA) [21]、[22]、[24]、[25]、[31]、[32]、gradient-based methods [18]、[19]、multi objective algorithm (MO) [33]–[35], etc. (3) 性能

估计策略是指估计所找到架构的性能的过程。通常,估计架构的最简单方法是训练模型,直到它收敛于训练数据。然而,这个过程需要大量的计算成本。已经提出了许多方法来简化这个过程,包括代理度量[30]、[36]、[37]、网络态射[38]、[39]、一次性[40]–[43]、代理模型[14]、[44]、[45]等。

在本文中,我们提出了 MA-Net,这是一种基于模因算法 [46]、[47] 和 EvoCNN 的 NAS 方法。具体来说,我们主要关注搜索策略,设计局部搜索策略来探索多分支网络的新设计空间。此外,我们引入了一种基于[28]的更快的性能估计技术,以加速所提出算法的收敛。

三、拟议的算法

本节详细描述了我们的 MA-Net。首先,我们将给出所提出算法的概要。接下来,我们针对第 2 节中讨论的三个方面,即搜索空间、搜索策略和性能估计策略,详细介绍所提出的方法。

A. 拟议的 MA-Net 的工作流程

算法 1:拟议的 MA-Net 框架	
1	$P0 \leftarrow$ 用新设计的种群初始化策略初始化种群;
2	用提出的评估技术评估 $P0$ 中个体的适应度; 3 $t \leftarrow 0$; 4 不满足终止条件时做
5	$S \leftarrow$ 选择具有松弛二元锦标赛选择的父解决方案;
6	$Q_t \leftarrow$ 使用来自 S 的交叉和变异算子生成后代;
7	在 Q_t 中评估个体的适应度;
8	使用精心设计的本地搜索策略来探索 Q_t 中每个个体的多分支搜索空间,然后评估额外生成的体系结构并更新每个个体;
9	$P_{t+1} \leftarrow$ 从 $P_t \cup Q_t$ 环境选择; $t \leftarrow t + 1$; 11 结束
10	
12	从 P_t 中选出最好的个体解码到对应的卷积神经网络中。

藻类。图 1 显示了所提出的 MA-Net 的工作流程。如图所示,首先,种群 $P0$ (第 1 行)由一个新设计的策略初始化,这将在 3.2 节中讨论。随后,使用将在 3.4 节中讨论的建议评估策略 (第 2 行)评估初始化种群。接下来,该算法通过交叉和变异算子生成后代,随机选择两个

父解决方案并使用精心设计的本地搜索策略更新这些后代 (第 5-8 行)。值得注意的是,这里的交叉和变异算子与 EvoCNN 中使用的不同。由于本研究中的外连接,为了改变额外的连接,我们对外连接代码中的每一位采用统一的交叉和变异算子。此外,在局部搜索之后,将额外生成的体系结构与初始个体进行比较,如果额外体系结构的性能更好,则将替换初始体系结构。进化过程中所有生成的个体都通过新设计的性能评估策略进行评估,可以大大减少时间消耗。最后,一旦满足终止标准,就会从最后一代中选出最好的个体,然后将其解码到相应的卷积神经网络中 (第 12 行)。

B. 搜索空间搜索空间

包含一组在 NAS 方法中找到的解决方案。在 EvoCNN 中,搜索空间中有三个不同的构建块,即卷积层、池化层和全连接层。然而,由于完全连接层, EvoCNN 中发现的体系结构复杂且具有大量权重。特别地,如 [27] 所示,完全连接的层容易过度拟合,从而阻碍了神经网络的泛化能力。

更重要的是,全连接层中的权重数量远大于卷积层中的权重数量。

因此,所提出的算法采用全局平均池化层来代替全连接层,全局池化层具有以下优点: (1)该层没有权重,这意味着该层可以节省大量计算。(2) 全局池化层可以加强特征图和类别之间的对应关系。由于这种变化,种群初始化策略与 EvoCNN 中的种群初始化策略有很大不同,这在 Alg.2 中有所说明。 2.

从 Alg 可以观察到。 2、首先初始化一些辅助变量 (第 1-4 行)。请注意, $P0$ 是一个空数组,用于存储生成的卷积层和池化层 (第 1 行)。 nc 和 np 分别是随机生成的卷积层和池化层数 (第 2-3 行)。此外, $A0$ 表示一个序列,描述了卷积层和池化层如何堆叠。

随后,为了生成完整的体系结构序列,它随机将 np 个 0 插入到 nc 中。然后,根据修改后的序列 $A0$,以下循环通过一系列卷积层和池化层的有序堆叠生成染色体 (第 7-15 行)。最后,我们在染色体顶部添加一个全局平均池化层以生成完整的架构 (第 16-17 行)。通过这种方式,它可以生成个体的种群。

C. 搜索策略

搜索策略用于探索搜索空间。所提出的算法中采用的搜索策略是

算法 2:种群初始化输入、种群大小 N,卷积层和池化	
层的最大数量Ncp。	
输出:初始化种群P0。	
1	P0 ← ∅; 2
nc ← 随机生成一个介于[1, Ncp]之间的整数; 3 np ← 随机生成一个介于[1, min{nc - 1, N - nc}]之间的整数; 4 A0 ← 生成一个长度为nc的数组,这个数组中的所有元素都设置为1; 5从A0 中随机选择np个节点,除了第一个节点。然后在A0中所有选中的节点前插入数字0; 6我←0; 7当我<N做	
8	if A0[i] == 1 then l ← 使
9	用随机设置初始化一个卷积层;
10	别的
11	l ← 使用随机设置初始化一个池化层;
12	结束
13	P0 ← P0 ∪ l;我
14	← 我 + 1; 15结
束	
16 part2 ← 初始化全局平均池化层; 17 P0 ← P0 ∪ 第二部分;	
18返回P0。	

EvoCNN 中的扩展。该搜索策略的主要过程已在算法中显示。1. 特别是,由于所提出的方法采用新设计的局部搜索策略来探索多分支搜索空间,如图 1 所述,因此本节主要讨论局部搜索策略的细节。

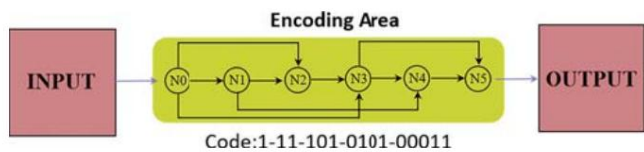


图 2. 编码序列。两端的节点是默认的输入输出,中间部分是编码区。每个圆节点代表一个卷积层并被编号,编号较高的节点可以从编号较低的节点接收特征图。每个节点都用一个短序列来表示它的关系,这个序列的长度与分配给节点的编号相同,这意味着第一个节点不会有序列,因为它的编号是“0”,它可以只被连接到输入。最后,所有这些短序列都用“-”连接起来,这个长序列就是外连接表的最终表示。

特别地,为了表示多分支搜索空间,我们采用[22]中提出的方法来表示网络架构的外部连接。如图2所示,为简单起见,输入和输出分别是两端的两个默认节点。每个节点代表一个

卷积层在编码区被分配一个编号。此外,编号较低的节点无法从编号较高的节点接收信息以避免循环。例如,如果一个节点的编号是i,则该节点将有i位来表示它与编号小于i的其他节点的对应关系。如图2所示,N3的节点数为3,即N3有3个比特,N3与N0、N1、N2之间可能存在三个连接。可以看出,N3节点的序号是“101”,说明N3可以接收来自N0和N2的信息。请注意,更改网络架构的外部连接并不困难。例如,翻转选定的位。因此,利用这种表示,可以通过交换这种表示中的节点序列来进行交叉,而变异则可以通过以一定的概率翻转染色体的每一位来进行。此外,局部搜索还被设计用于对神经网络进行局部开发,这在算法中进行了总结。 3.

算法 3:本地搜索	
输入:个体I0,数字 N。	
输出:一组个体Aind。 1如果I0没有外部连接序列则	
为I0生成初始连接序列; 3 end 4 Ac ← 生成所有可能的连接序列	
2	
I0;	
5 C0 ← I0 的连接顺序;	
6从Ac中去掉与C0相同的序列;	
7 Aind ← ∅; 8	
length ← Ac的长度;	
9 N ← min{N, 长度};	
10从中随机选择N个不重复的序列	
Ac添加到Aind;	
11归来_	

正如算法中所描述的那样。 3.局部搜索策略为输入个体I0初始化连接序列,如果I0没有连接序列(第2-4行)。请注意,初始连接序列仅表示I0 的链架构。然后,为I0生成所有可能的连接序列(第 5 行)。例如,假设单个I0有 i 个带有卷积层的节点,并且每个第 i 个节点拥有最多 i 个额外的连接,这意味着I0有 at 个额外的连接。因此,总共有(1+i) × i

$$2\binom{(1+i) \times i}{2}$$
 I0的不同连接选择,所有这些选择都存储在数组Ac 中。为了避免从Ac 中选择与C0相同的序列,有必要从Ac中删除C0 (第 7 行)。最后,为了探索多分支空间,从Ac中选择 N 个非重复序列。所有选择的序列都独立地合并到I0中,并使用新设计的评估策略进行评估,如下一节所述。然后用最好的一个来表示I0的外接。在局部搜索策略之后,种群中的所有个体都可以拥有外部联系

序列。

D. 绩效评估策略

性能评估策略用于评估 NAS 方法的解决方案。EvoCNN 中使用的方法是在特定基准上训练模型,直到它收敛。

但是,这种方式会消耗大量的计算资源。为了节省大量计算,我们根据 [28] 中提出的假设,将训练时间减少到收敛前的几个时期。根据评估假设,我们可以在训练过程的早期阶段比较所提出算法中发现的网络的性能。因此,我们可以简单地在几个甚至一个 epoch 中训练这些解决方案,以估计它们在不同数据集上的性能。

四、实验

为了验证所提出的 MA-Net 的性能,使用 EvoCNN 中也使用的图像分类基准数据集进行了全面的实证研究。特别是,在本节中,我们首先介绍数据集和实验设置。接下来我们展示我们的实验结果并讨论不同数据集上的结果。

A. 数据集和实验设置

在我们的实验中,EvoCNN 使用了五个具有代表性且困难的数据集,即 Fashion [48],MNIST with Background Images (MBI) [49],Random Background (MRB) [49],旋转数字 (MRD) [49] 和 RD 加背景图像 (MRDBI) [49] 基准,被考虑用于调查。请注意,由于矩形图像 (RI) [49] 和 MNIST Basic [49] 基准是简单的数据集,并且大多数卷积网络方法可以始终如一地获得 100% 的准确度,我们在本研究中不使用这两个数据集。

这五个基准数据集可以分为两个类别和图像大小均为28×28,单通道。第一类是时尚基准,它是 10 类文章图像的数据集,如图 3 所示 (参见图 3 中的上图)。它包含一个包含 60, 000 个示例的训练集和一个包含 10, 000 个示例的测试集。

另一个类别包含 MNIST 数字的变体。由于基本的MNIST benchmark比较简单,在原来的MNIST benchmark基础上增加了一些不同的变化,分别是MBI,MRB,MRD,MRDBI benchmarks,如图3所示 (见图3下图)。此外,每个基准在训练集中只有 12, 000 张图像,而在测试集中有 50, 000 张图像,这意味着这些分类任务变得比原始 MNIST 基准更加不平衡和困难。

为了确保与 EvoCNN 的公平比较,大部分参数设置都是根据 EvoCNN 配置的,如表 I 所示。为了节省计算成本,种群大小和最大世代数设置为 EvoCNN 的一半。如果最后 8 代没有任何改进,进化将终止。



(a) 来自具有 10 个类的时尚基准的示例。从左到右,这些图像是 T 恤、裤子、套头衫、连衣裙、外套、凉鞋、衬衫、运动鞋、包、踝靴。



(b) 来自具有 10 个类的 MNIST 基准变体的示例。从左到右,每两张图片来自同一个基准数据集,分别是MBR,MBI,MRD,MRDBI。

图 3. 来自两种基准的示例。一个是 Fashion benchmark,另一个是 MNIST benchmark 的变体。

此外,卷积层的最大长度为 10,因为全连接层被卷积层取代。通常,如算法中所述。2,池化层的最大长度应小于卷积层的长度,因此本研究中池化层的最大长度设置为9。此外,用于本地搜索的最大次数为 2。个人在 Fashion 基准测试和其他基准测试中的训练次数分别为 1 次和 4 次。

表一
参数设置

参数	设置50
种群规模和最大世代数	20 10
SBX的分布指数和多项式变异	9 0.9
卷积层的最大长度	0.1 2
池化层的最大长度	1 相
交叉概率	同
突变概率	
最大本地搜索次数	
卷积层的进步	
池化层中的填充	

我们通过 Tensorflow [50] 实现了所提出的算法。为了加速训练过程并避免过度拟合问题,将 BatchNorm 应用于每个卷积层之后的特征图。此外,每个实验都在 RTX2080ti 上执行,我们的方法中没有使用数据增强方法。最后,我们在 Fashion benchmark 上独立运行我们的方法 5 次,并在 MNIST benchmark 数据集的变体上随机运行 1 次。

B. 结果与讨论

与 EvoCNN 在 Fashion 基准数据集和 MNIST 基准的变体上进行比较的所有实验结果分别总结在表 II 和表 III 中。如表 II 所示,最后两行显示了所提出的 MA-Net 获得的最佳和平均分类错误,其他行显示了 EvoCNN 的相应结果。此外,为了在建议的 MA-Net 和 EvoCNN 之间提供更多比较,在最后一列中给出了 Fashion 基准数据集的参数数量和计算时间的附加信息。在

表三,最后一行只展示了MA Net在MRD、MRB、MBI、MRDBI benchmarks上的随机分类误差,第一行是EvoCNN在相应 benchmarks上得到的最好结果。最后,所有结果都是通过这两种算法在没有任何数据增强预处理策略的情况下获得的。

从表 II 可以看出,通过比较平均性能,所提出的方法 MA-Net 优于 EvoCNN。MA-Net 的分类错误率为 6.85%,而 EvoCNN 为 7.28%。此外,MA-Net 的参数数量和使用时间仅为 78.6 万个和 1.54 个 GPU 天,大大低于 EvoCNN 的 652 万个权重和 4 个 GPU 天。尽管 EvoCNN 实现的最佳分类错误率为 5.47,略小于 MA-Net 中获得的 6.06,但 MA-Net 仅使用 124 万个参数,仅需 1.03 GPU 天即可获得具有竞争力的架构。请注意,尽管 RTX2080ti 和 GTX1080 之间存在差异,但时间差距仍然很大。因此,同时考虑最佳和平均性能,很明显 MA-Net 在 Fashion 基准数据集上的性能更加稳定和高效。

表二

拟议方法的分类错误

EVOCNN谈时尚

分类器	错误#Parameter #Time	
EvoCNN (最佳)	5.47	6.68M 4d
EvoCNN (平均)	7.28	MA-Net (最佳)
Net (最佳)	6.06	MA-Net (平均)
均)	6.85	0.786M 1.54d

此外,如表 III 所示,MA-Net 的随机性能优于 EvoCNN 在所有测试基准 (MRD、MRB、MBI 和 MRDBI 数据集)上的最佳性能。特别是,在 MRD 和 MRDBI 基准上,MA-Net 的分类误差分别为 3.56% 和 15.92%,小于 EvoCNN 获得的 4.53% 和 35.03%。因此,MA-Net 在 MNIST 的变体上也有更好的性能。

表三

拟议方法的分类错误

MRD、MRB、MBI、MRDBI 上的EVOCNN

分类器	MRD	MRB	MBI	MRDBI
EvoCNN (最佳)	5.22	2.80	4.53	35.03
MA-Net(随机)	3.33	2.48	3.56	15.92

五. 结论

在本文中,我们提出了 MA-Net,这是一种基于 EvoCNN 和模因算法的新 NAS 方法。实验研究证实,与 EvoCNN 相比,该算法可以用更少的参数和更少的计算资源找到更好的架构。未来,我们希望探索更合适的交叉和变异算子来加速自动化过程。此外,

还想将所提出的方法应用于现实世界的深度学习应用程序。

六.致谢

这项工作得到了国家自然科学基金 (NSFC) 的部分支持,资助号为 61876025,重庆市海归创业创新支持计划资助号为 cx2018044 和 cx2019020。

参考

[1] A. Krizhevsky,J. Sutskever 和 GE Hinton,“Imagenet classification with deep convolutional neural networks”,*神经信息处理系统进展*,2012 年,第 1097-1105 页。

[2] 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。

[3] C. Szegedy,W. Liu,Y. Jia,P. Sermanet,S. Reed,D. Anguelov,D. Erhan,V. Vanhoucke 和 A. Rabinovich,“深入卷积”,在 *Proceedings of IEEE 计算机视觉和模式识别会议*,2015 年,第 1-9 页。

[4] 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。

[5] G. Huang,Z. Liu,L. Van Der Maaten 和 KQ Weinberger,“密集连接的卷积网络”,*IEEE 计算机视觉和模式识别会议论文集*,2017 年,第 4700-4708 页。

[6] S. Ren,K. He,R. Girshick 和 J. Sun,“Faster r-cnn:使用区域建议网络实现实时目标检测”,*神经信息处理系统进展*,2015 年,第91-99页。

[7] W. Liu,D. Anguelov,D. Erhan,C. Szegedy,S. Reed,C.-Y. 傅和AC Berg,“Ssd: Single shot multibox detector”,*欧洲计算机视觉会议*,斯普林格,2016 年,第 21-37 页。

[8] J. Redmon 和 A. Farhadi,“Yolov3:增量改进”,*arXiv 预印本 arXiv:1804.02767*,2018。

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and AL Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” *arXiv preprint arXiv:1412.7062*, 2014.

[10] O. Ronneberger,P. Fischer 和 T. Brox, “U-net:用于生物医学图像分割的卷积网络”,*国际医学图像计算和计算机辅助干预会议*,斯普林格,2015 年,第 234-241 页。

[11] V. Badrinarayanan,A. Kendall 和 R. Cipolla, “Segnet:一种用于图像分割的深度卷积编码器-解码器架构”,*IEEE 交易模式分析和机器智能*,卷. 39,没有. 12,第 2481-2495 页,2017 年。

[12] J. Bergstra,D. Yamins 和 DD Cox, “建立模型搜索科学:视觉架构数百个维度的超参数优化”,2013 年。

[13] H. Mendoza,A. Klein,M. Feurer,JT Springenberg 和 F. Hutter, “To wards automatically-tuned neural networks”,*自动机器学习研讨会*,2016 年,第 58-65 页。

[14] K. Kandasamy,W. Neiswanger,J. Schneider,B. Póczos 和 EP Xing, “使用贝叶斯优化和最优传输进行神经架构搜索”,*神经信息处理系统进展*,2018 年,2016-2025 页。

[15] B. Zoph 和 QV Le, “使用强化学习进行神经架构搜索”,*arXiv 预印本 arXiv:1611.01578*,2016 年。

[16] B. Baker,O. Gupta,N. Naik 和 R. Raskar, “使用强化学习设计神经网络架构”,*arXiv 预印本 arXiv:1611.02167*,2016 年。

[17] H. Cai,T. Chen,W. Zhang,Y. Yu 和 J. Wang, “通过网络转换进行高效架构搜索”,第 30 届 AAAI 人工智能会议,2018 年。

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

[19] A. Hundt,V. Jain 和 GD Hager, “sharpdarts:更快更准确的可微架构搜索”,*arXiv 预印本 arXiv:1903.09900*,2019 年。

[20] KO Stanley 和 R. Miikkulainen, “通过增强拓扑进化神经网络”,*进化计算*,卷. 10,没有. 2,第 99-127 页,2002 年。

- [21] 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”,第 34 届国际机器学习会议论文集第 70 卷,JMLR.组织,2017 年,第 2902–2911 页。
- [22] L. Xie 和 A. Yuille,“Genetic cnn”,载于 IEEE 计算机视觉国际会议论文集,2017 年,第 1379–1388 页。
- [23] H. Liu,K. Simonyan,O. Vinyals,C. Fernando 和 K. Kavukcuoglu,“高效架构搜索的层次表示”,arXiv 预印本 arXiv:1711.00436,2017 年。
- [24] E. Real,A. Aggarwal,Y. Huang 和 QV Le,“图像分类器架构搜索的正则化演化”,aaai 人工智能会议论文集,卷. 33, 2019, 第 4780–4789 页。
- [25] Y. Sun,B. Xue,M. Zhang 和 GG Yen,“用于图像分类的进化深度卷积神经网络”,IEEE Transactions on Evolutionary Computation,2019 年。
- [26] M. Wistuba,A. Rawat 和 T. Pedapati,“神经网络搜索调查”,CoRR,卷. abs/1905.01392,2019 年。[在线]。
可用:hthttp://arxiv.org/abs/1905.01392
- [27] M. Lin,Q. Chen 和 S. Yan,“网络中的网络”,arXiv 预印本 arXiv:1312.4400, 2013.
- [28] X. Zheng,R. Ji,L. Tang,B. Zhang,J. Liu 和 Q. Tian,“**用于有效神经架构搜索的多项分布学习**”,CoRR,卷. abs/1905.07529,2019 年。[在线]。可用:hthttp://arxiv.org/abs/1905.07529
- [29] Z. Zhong,J. Yan,W. Wu,J. Shao 和 C.-L. Liu,“实用的块式神经网络架构生成”,载于 IEEE 计算机视觉和模式识别会议论文集,2018 年,第 2423–2432 页。
- [30] 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。
- [31] T. 埃尔斯肯,J.-H. 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。
- [32] E. Real,A. Aggarwal,Y. Huang 和 Q. Le,“图像分类器架构搜索的老化演化”,AAAI 人工智能会议,2019 年。
- [33] T. Elsken,JH Metzen 和 F. Hutter,“通过拉马克进化进行高效的多目标神经架构搜索”,arXiv 预印本 arXiv:1804.09081,2018 年。
- [34] J.-D.董,A.-C. Cheng, D.-C. Juan,W. Wei 和 M. Sun,“Dpp net:设备感知渐进搜索帕累托最优神经架构”,欧洲计算机视觉会议 (ECCV) 会议记录,2018 年,第 517–531 页。
- [35] Z. Lu,J. Whalen,V. Boddeti,Y. Dhebar,K. Deb,E. Goodman 和 W. Banzhaf,“Nsga-net:一种用于神经架构搜索的多目标遗传算法”,arXiv预印本 arXiv:1810.03522, 2018。
- [36] A. Klein,S. Falkner,S. Bartels,P. Hennig 和 F. Hutter,“大型数据集上机器学习超参数的快速贝叶斯优化”,arXiv 预印本 arXiv:1605.07079,2016 年。
- [37] P. Chrabaszcz,I. Loshchilov 和 F. Hutter,“作为 cifar 数据集替代品的 imagenet 下采样变体”,arXiv 预印本 arXiv:1707.08819,2017 年。
- [38] T. Wei,C. Wang,Y. Rui 和 CW Chen,“网络态射”,国际机器学习会议,2016 年,第 564–572 页。
- [39] H. Cai,J. Yang,W. Zhang,S. Han 和 Y. Yu,“高效架构搜索的路径级网络转换”,arXiv 预印本 arXiv:1806.02639,2018 年。
- [40] S. Saxena 和 J. Verbeek,“卷积神经网络”,神经信息处理系统进展,2016 年,第 4053–4061 页。
- [41] A. Brock,T. Lim,JM Ritchie 和 N. Weston,“Smash :通过超网络进行一次模型架构搜索”,arXiv 预印本 arXiv:1708.05344,2017 年。
- [42] H. Pham,MY Guan,B. Zoph,QV Le 和 J. Dean,“通过参数共享进行高效神经架构搜索”,arXiv 预印本 arXiv:1802.03268,2018 年。
- [43] G. Bender,“理解和简化一次性架构搜索”,2019 年。
- [44] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei,A. Yuille,J. Huang 和 K. Murphy,“渐进式神经架构搜索”,欧洲计算机视觉会议 (ECCV) 会议记录,2018 年,第 19–34 页。
- [45] R. Luo,F. Tian,T. Qin,E. Chen 和 T.-Y. Liu,“神经结构优化”,神经信息处理系统进展,2018 年,第 7816–7827 页。
- [46] EK Burke,JP Newall 和 RF Weare,“大学考试时间表的模因算法”,在关于自动时间表的实践和理论的国际会议上。斯普林格,1995 年,第 241–250 页。
- [47] Y.-S. Ong,MH Lim 和 X. Chen,“模因计算 过去、现在和未来 [研究前沿]”,IEEE 计算智能杂志,卷. 5,没有. 2,第 24–31 页,2010 年。
- [48] H. Xiao,K. Rasul 和 R. Vollgraf,“Fashion-mnist:一种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。
- [49] H. Larochelle,D. Erhan,A. Courville,J. Bergstra 和 Y. Bengio,第 24 届国际机器学习会议论文集 “深度架构对具有多种变异因素问题的实证评估”.美国计算机学会,2007 年,第 473–480 页。
- [50] M. Abadi,A. Agarwal,P. Barham,E. Brevdo,Z. Chen,C. Citro,GS Corrado,A. Davis,J. Dean,M. Devin 等人,“Tensorflow :异构分布式系统上的大规模机器学习”,arXiv 预印本 arXiv:1603.04467,2016 年。