

神经结构搜索和多目标进化算法

异常检测

恰克里特里提昆

可再生能源与智能电网技术学院（SGtech）

泰国彭世洛那黎宣大学

chakkritt60@nu.ac.th

刘叶萌 大理理工

大学软件学院 中国大

连 liuziweiww@outlook.com

林旭

干

南澳大利亚大学澳大利亚阿德莱德

xuyly032@mymail.unisa.edu.au

伊万·李

STEM

南澳大利亚大学 澳大利亚阿德莱

德 Ivan.Lee@unisa.edu.au

摘要 处理速度和内存占用是物联网设备和嵌入式系统等资源受限设备上应用程序处理的重要因素。深度学习一直在不断发展,因此它可以用于资源受限的设备,但使用它仍然存在一些限制,因为这些设备不是为处理复杂任务而设计的。此外,卷积神经网络 (CNN) 模型的复杂性是在这些设备上实施的障碍。在本文中,我们开发了神经架构搜索 (NAS),它使用多目标遗传算法和萤火虫算法来创建一个不太复杂且稳健的 CNN 模型,重点是搜索具有更快处理时间和最小存储大小的模型。应用五个图像数据集来检查所提出技术的性能,包括两个用于表面的裂缝检测数据集或用于工业应用的已建成基础设施检查。

实验结果表明,所提出的技术在可比较的分类精度下持续降低了参数计数和处理时间。

关键词 深度学习,神经结构搜索,多目标、遗传算法和萤火虫算法

一、引言

人工智能在过去几年的快速发展催生了许多有意义和有价值的研究。异常检测是一个典型的应用,已经应用于医学影像[1]、医疗保健[2]、图像处理[3]、网络入侵检测[4]、欺诈检测、视频监控[5]等多个领域。 [6]。

一般来说,异常检测旨在区分异常样本和正常样本。看似普通的二分法问题,其实异常检测有一个固有的属性,就是样本极度不平衡。在某些极端情况下,没有异常

样本存在于训练数据集中[7]。因此,主流的研究方向可以分为两类。

第一类是基于正常的例子,甚至只是正常的例子。在这些情况下,异常检测问题首先设置构造正常样本分布的模型,然后通过这些模型来实现异常样本检测[8][9]。对于网络分析,可以使用基于图形的方法 [99]。对于基于图像的分析,生成对抗网络 (GAN) 可以很好地处理异常检测问题的不平衡样本 [10][11]。

其中,GANomaly[8]的设计是近几年最具代表性和影响力的基于GAN的异常检测。作者认为,异常检测中异常样本通常很少,因此很难提供可以直接区分异常样本的模型。由于检测异常样本并不可靠,本文采用相反的方法。它只训练正常样本,如果测试集中的样本看起来不像正常样本,则将其区分为异常样本。不同于原始的 GAN [12],GANomaly 使用变异的 GAN 和自动编码器网络。它将生成器网络改进为编码器解码器编码器,在很大程度上优化了网络,例如梯度下降。自动编码器是异常检测中的经典方法。它的解决方案是使用尽可能多的正常样本来训练自编码模型。结果,它可以很好地重建正常样本。然而,由于训练数据集中只有少数异常样本,甚至更多,训练集中没有任何异常样本,因此无法很好地重建异常样本。因此,通过输入图像的重建误差,不难区分正常样本和异常样本。

然而,这种方法非常容易受到噪音的影响。所以如果我们

需要一个可用的异常检测模型,它需要在编码器上添加各种约束。

Auto-GAN [13] 是在 GAN 中使用网络架构搜索 (NAS) [14] [15] 的首次尝试。网络架构搜索在图像分类和其他任务中取得了一些成功。然而,大多数现有的 NAS 方法都表现出很高的计算复杂性,这不仅需要昂贵的硬件,而且需要大量的处理时间。Sirui 等人。在 2019 年 [16] 讨论了这个问题,并假设在 NAS 的目标函数的逼近中引入过多的偏差是 NAS 需要两阶段优化参数的主要原因。因此,他们提供了一种解决方案,通过更严格的近似来确保一级优化。但是这种方法不适用于大规模的网络搜索任务。因此,为了解决这个问题,他们提出了 DSNAS [17] 方法,该方法具有可微分的搜索架构,可以同时优化结构参数和网络参数。

与第一类相反,第二类考虑具有足够异常 (缺陷)样本进行训练的场景。

在这些情况下,异常检测也可以定义为分类任务。有时,如果检测区域太大而检测位置相对较小,我们可以将区域细分成小区域,通过滑动窗口进行分类,覆盖每个子区域。

然而,这两种方法在参数选择方面都面临着相同的限制。众所周知,在定义一组适当的参数以在实验中产生良好的鲁棒性方面存在许多挑战。这些参数包括 (但不限于)内核大小、过滤器的滚动位置、输出大小、零填充等。随着神经网络设计的越来越复杂,参数和超参数的数量也大大增加。虽然神经网络的结构可以手动设计 [18] [19] 并且网络中参数和超参数的选择是基于人工经验,但它往往不一定是最佳选择并且总是很耗时,因为我们需要反复尝试。

神经架构搜索 (NAS) [21] 旨在为特定任务找到最合适的参数集和合适的网络结构。这种方法不需要遵循以前设计的结构,也不依赖于人类的专业知识。因此,它极大地优化了神经网络的架构,使设计最稳健的模型成为可能。NAS 的另一个改进是它的架构足够灵活,因此它适用于大小数据集。搜索空间是 NASNet 中的核心元素之一,它定义了一组可被搜索的神经网络结构 (单元)。图 1 显示了 NASNet 的一个例子。与普通架构 (左) 相比, NASNet 不仅有一种输入 (右)。

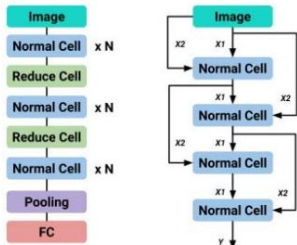


图 1. NASNet 搜索空间。

Barret [15] 将每层过滤器的数量和大小变成了一个优化问题,之前需要手动确定。网络可以自行选择这些参数的最佳组合。然后作者优化了算法[22],他们只在一些特殊的单元、正常单元和降采样使用的缩减单元中搜索,而不是在整个网络中搜索,并获得了更高的速度和更好的性能。然而,这种方法的局限性在于其计算量大,耗费大量时间。尽管作者不断改进算法 [23] [24],但它仍然需要数百个 GPU-days。

因此,一些可以加快这一进程的解决方案被提出,例如,wight sharing [25] 和 differentiable equation [26],这大大降低了硬件和时间的要求。

然而,大多数现有的 NAS 方法不仅需要昂贵的硬件,而且需要大量的时间。例如,使用基于 CIFAR-10 数据集 [22] 的强化学习 (RL) 完成任务最多需要 2,000 个 GPU 天。

而 AmoebaNet [27] 需要 3,150 个 GPU 天才能使用进化算法并基于相同的数据集找到最佳模型。

此外,这些算法无法在小型设备上运行,因为它们需要过多的处理时间。因此,有必要简化NAS网络的架构,加快研究速度。这个问题已经有很多杰出的工作,包括 SMBO 方法 (基于序列模型的优化)[24]、所需结构定义方法 [28]、权重共享 [25] 和使用 DARTS 模型 [26]。Sirui 等人。在 2019 年 [16] 讨论了这个问题,并假设在 NAS 的目标函数的逼近中引入过多的偏差是 NAS 需要两阶段优化参数的主要原因。

因此,他们提供了一种解决方案,通过更严格的近似来确保一级优化。但是这种方法不适用于大规模的网络搜索任务。因此,为了解决这个问题,他们提出了 DSNAS [17] 方法,该方法具有可微分的搜索架构,可以同时优化结构参数和网络参数。

在本文中,提出了一种多目标遗传算法网络模型 (MOGA-Net)以减少搜索时间并找到最适合给定数据集的最佳神经网络结构。我们的方法遵循一个典型的过程:首先,分析和识别目标数据集的结构和内容。该模型的体系结构是在训练集的过程中设计的,同时它的有效性是基于测试数据集来衡量的。上述分析、开发和测量的过程将不断迭代,直到为目标数据集选择了最佳模型。该模型的最佳架构发挥了高效率,需要小的存储空间,使其可以在处理能力不足的设备上运行。

本文的贡献包括通过将多目标遗传算法和萤火虫算法分别应用于 NAS 网络,设计了 MOGA Net 和 FA-Net。使用裂缝图像数据集检查所提出的技术,并在其他分类数据集上进行进一步的实验以检查跨区域的鲁棒性

不同的应用。
本文组织如下。第二部分描述了本文的方法。实验和结果是

第 III 节介绍,而第 IV 节介绍本文的结论性评论。

二.方法

A. 多目标遗传算法

遗传算法 (GA)[29]起源于生物系统的计算机模拟。它是借鉴了达尔文的进化论和孟德尔的遗传学说,模仿自然界生物进化机制发展起来的一种随机全局搜索和优化的方法。遗传算法是一种使用并行搜索技术寻找全局最优解的有效方法,它可以在搜索过程中自动获取和积累关于搜索空间的知识。搜索过程自适应控制,最终得到最优解。

遗传算法从初始种群演化而来,初始种群是研究问题的潜在解集。在这里,种群由一定数量的基因编码的个体组成。每个个体都是具有特征染色体的实体。染色体是主要的

遗传物质的载体,即多个基因的集合体。它的内在表现 (即基因型)是一定的基因组合,决定了一些个体特征,我们称之为表型。例如,头发的颜色是由控制这种特性表现的染色体中的某种基因组合决定的。因此,有必要实现表型到基因型的映射,可以定义为二进制编码。一般来说,遗传算法要解决的问题会映射为一个数学问题,其可行解称为染色体。这个可行解一般由多个元素组成,每个元素定义为这条染色体上的一个基因。

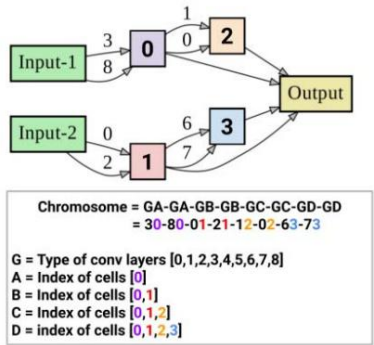


图 2. 多目标遗传算法和萤火虫算法的染色体结构。

在本文中,基因型被定义为模型的架构。模型的完整结构由从 NAS 的搜索空间中搜索的单元组成。每个单元格由多个层合并,例如最大池化层、平均池化层和卷积层。这些细胞通过相互关联的方式形成模型基因型。这是图 2 中所示的示例。在该模型中,基因个数定义为偶数,必须满足GA-GA、GB-GB、GC-GC、GD-GD等8种类型。每个基因由一个操作 (G) 和一个索引 (A、B、C 和 D)组成。本例中的操作集定义了9种不同的操作,包括 max pooling,average

池化和卷积,用[0,1,2,3,4,5,6,7,8]表示。操作的位置由索引 (A、B、C 和 D)表示。每个操作都需要连接到一个索引,可以指定为 A=[0]、B=[0,1]、C=[0,1,2] 和 D=[0,1,2,3]。这种设计允许通过 NAS 形成各种网络,包括具有简单或复杂结构的网络 [20]。

有必要使用适应度函数来衡量每个染色体的解决方案的质量。当初始种群被定义时,随着代际变化演化出越来越好的近似解,应该遵循适者生存的原则。

也就是说,在每一代进化过程中,个体都需要根据问题域的适应度来进行选择或淘汰。同时参照自然遗传学的遗传操作进行交叉变异。在本文中,将基于准确性搜索模型的体系结构,因此将适应度函数定义为Eq. (1),它使用从测试数据库中获得的确度。

最小化 $f(x) = (x_1, x_2, x_3, \dots, x_n)$ (1)

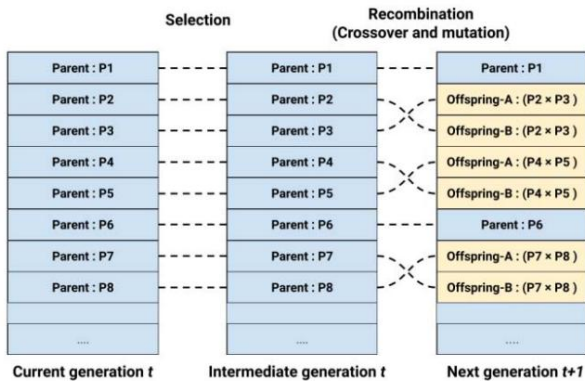
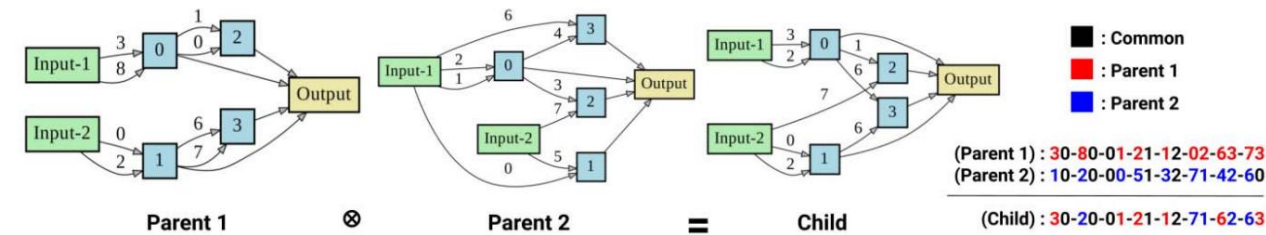
其中n是个体的数量。在这个过程中,每一代人都会比上一代更适应环境。因此,最后一代种群中的最优个体将被认为是问题的近似最优解。近年来,遗传算法逐渐扩展到解决多目标优化问题。在本文中,训练误差、计算成本和参数大小是我们在寻找神经网络模型架构时应该考虑的三个核心指标。因此,这是一个典型的多目标优化问题。我们将目标函数定义为方程式。(2) 其中f为适应度值,整数表示目标个数, X为个体集合。方程式的目标。(2)是所有个体的适应度必须尽可能小,因此该算法被称为多目标遗传算法 (MOGA) 。

最小化 $f(x) = (x_1, x_2, x_3, \dots, x_n)$ (2)

交叉和变异是进化过程中产生新解集的主要途径。在这里,交叉需要上一代的两条染色体,新的染色体是通过切割和拼接前一代的染色体产生的。交叉可以采用多种方式进行,例如多点交叉、算术交叉和均匀交叉 (如图3所示)。多点交叉是指在染色体上随机设置交叉点,并在交叉点交换两个成对个体的部分染色体。算术交叉定义为两条染色体的线性组合,由此产生两个新的个体。一般来说,算术交叉的运算对象是用浮点数编码表示的个体。均匀交叉以相同的概率交换两条成对染色体上的每一个基因,由此形成两个新的个体。

突变是指用其他等位基因替换染色体中某个位置的基因值,然后形成新的染色体。突变突破局限

的搜索,因此更容易找到算法的全局最优解。



B. 萤火虫算法

在处理复杂的优化问题时,群体智能算法 (SIA)往往表现出高效的性能。 SIA是为模仿自然界中动物的群体行为而提出的一种算法。这些动物具有简单的个体行为,但可以通过群体间的合作完成复杂的工作。萤火虫算法 (FA) 是 SIA [30] 的一个代表性例子,它的灵感来自于萤火虫在自然界中通过发光来吸引配偶的行为。在自然界中,萤火虫通过发光来吸引配偶,较弱的萤火虫会被较强的萤火虫吸引并靠近它们。

基于这种行为,萤火虫由优化问题定义域中的点表示,萤火虫的发光强度由点的适应度值表示。萤火虫通过向比它们发光的萤火虫移动来不断改变它们的位置。萤火虫移动的距离是由两只萤火虫之间的吸引力决定的,而吸引力的强弱是由两只萤火虫之间的距离决定的。在运动过程中,如果萤火虫位置发生变化,相应的适应度值也会发生变化。因此,需要在每次运动后重新计算萤火虫的适应度值,以评估萤火虫位置的优劣。萤火虫算法就是重复上述动作,完成对优化问题解空间的搜索。另外,在萤火虫算法中,萤火虫之间没有性别划分,即种群中任意两只萤火虫之间存在吸引和吸引关系,这种吸引和吸引关系由适应度的大小决定两只萤火虫。

FA 的简单性和高效性使其在解决许多工程和科学问题方面广受欢迎,例如

资源分配[31]、医疗诊断[32]、方程求解[33]。在标准 FA 之上引入了各种扩展,例如 Lv 等人的工作使用精英学习策略为萤火虫提供更好的学习对象 [34]。

为了说明FA算法的流程,假设优化问题的维数为 ,萤火虫的位置用 (j_1, j_2, \dots, j_n) 表示,萤火虫构成萤火虫种群。

定义一:萤火虫的亮度强度:

$$I_j = \text{brightness of firefly } j$$

在实践中,亮度强度由萤火虫的适应度值决定,即

定义二:萤火虫的吸引力:

$$A_{ij} = \frac{I_j - I_i}{r_{ij}^{\alpha}}$$

其中 A_{ij} 是 $j=0$ 时的吸引力。 α 为光吸收系数,一般取常数 1。 r_{ij} 为萤火虫个体之间的欧氏距离,由下式确定。

$$r_{ij} = \sqrt{\sum_{k=1}^n (j_k - i_k)^2}$$

定义3:萤火虫向萤火虫的位置更新:

$$j_{k+1} = j_k + \lambda \cdot A_{ij} \cdot (j_k - i_k) + \lambda \cdot \text{rand}() \cdot (j_k - i_k)$$

其中, i , j 是萤火虫的位置,在维度上, λ 是步长因子,是[0,1]上的常数。 $\text{rand}()$ 是在[-0.5,0.5]上服从均匀分布的随机数。

C. 遗传算法和萤火虫算法的搜索空间

本节描述了将进化算法的概念应用于 NAS 网络的机制。通过应用多目标遗传算法或萤火虫算法来选择一组结构模型,以形成可提高性能的网络结构。

构成cell的层的类型已经基于NASNet的方法进行了修改,超出了参数的定义。此外,我们补充两个倒置残差

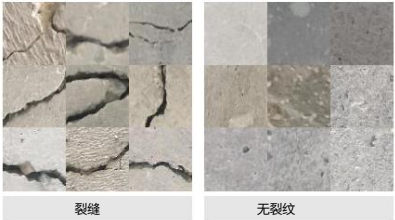
3×3 和 5×5 的层到搜索空间。因为 MNASNet 提到这种方法可以有效降低操作的复杂度,同时保持模型的精度,就像深度可分离卷积 [35] 一样。搜索空间由多种格式组成,可以从中找到最佳的架构模型。

本文使用的结构模型包括:

内核3x3	类型
3x3 3x3 5x5	max pooling
3x3 5x5 3x3	average pooling
5x5	depthwise-separable conv
	depthwise-separable conv
	dilated convolution 扩张卷
	积 inverted residuals conv
	inverted residuals conv
	skip connection
-	

根据多目标遗传算法和萤火虫算法,需要对每一代种群进行评价,确定目标数据集上的适应度值。因此,该任务的 GPU 时间取决于种群和世代的总数。从这个角度来看,我们定义了每个型号的 NAS 的“人口”。创建随机模型(种群),搜索空间将考虑以下几点:错误率、参数数量和计算成本,包括每秒浮点运算(FLOPs)。这些种群数量已经迭代了几代,所有的适应度值都被标记出来,以便以后进行比较。在初级阶段,我们为了实际操作的目的指定处理代数。我们设计了所有种群的世代数后,可以比较每一代的适应度值,找出最好的。

根据NASNet搜索空间的核心思想,模型的架构由上述单元按照多目标遗传算法和萤火虫算法的方法设计,基于最佳种群搜索最优模型。该系统遵循以下进化策略:(1)在一代中,交叉不同种群之间的数据,选择精度最高的模型。(2)对错误率最低的种群进行变异,将其作为下一代种群。(3)将下一代种群作为错误率最低的当前代种群。(4)交叉、选择和变异当前代种群作为步骤(1)、(2)



(a) 混凝土裂缝数据集



(b) SDNET2018数据集

图 6. 裂纹和非裂纹图像样本。

(3) 直到上一代结束。整体工作流程总结在图 5 中:

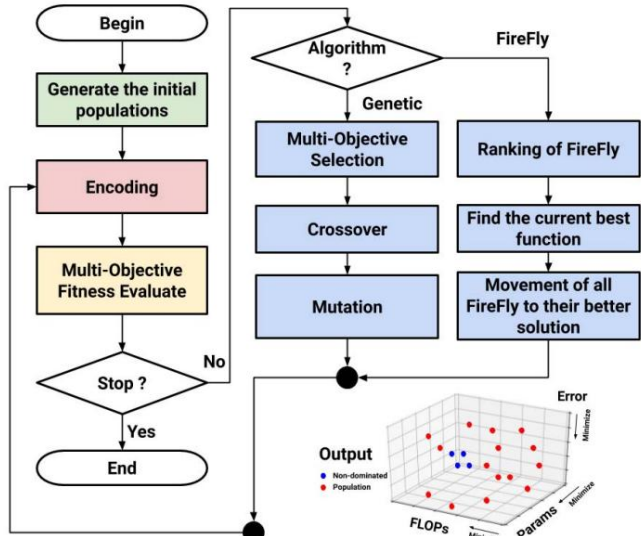


图 5. 多目标遗传算法和萤火虫算法的搜索空间。

三、实验和结果

实验分两部分进行:首先,使用在 Intel(R) 上运行的多目标遗传算法和萤火虫算法确定模型的结构

Xeon(R) W-3235 CPU @ 3.30GHz 12 核 CPU、192 GB RAM 和 NVIDIA RTX 2080Ti GPU,在 Ubuntu 18.04.3 操作系统上。第二部分是使用五个选定数据集评估 MOGA-Net 和 FA-Net: Fashion-MNIST、CINIC-10、STL-10、Concrete Crack 和 SDNET2018。PyTorch 深度学习库用于实现所提出的算法。

A. 数据集

本文使用的裂缝数据集是混凝土裂缝和 SDNET2018,如图 6 所示。这些数据集集中的图像可以分为两类:裂缝和非裂缝。在混凝土裂缝中,有 40,000 张图像,每个类别有 20,000 张图像。SDNET2018 数据集包含 56,092 张图像,其中裂缝图像 8,484 张,非裂缝图像 47,608 张。进一步,将这些图像按各 50% 的等比例分成训练数据和测试数据。然后,将图像调整为 112×112 像素以适应 CNN 模型。

B. SDNET2018架构搜索

我们定义了 10 代,每代 20 个种群,给定 200 个种群,使用多目标遗传算法和萤火虫算法找到模型结构。第一代种群是随机分布的,第2-10代种群是通过遗传算法或萤火虫位置的移动进化而来的。

搜索过程的超参数定义为:细胞总数(正常细胞和减少细胞)为 4 层,8 个初始通道,通过在 SDNET2018 数据集上从头开始训练网络 1 个时期,批量大小为128,权重衰减等于 0.0003,动量等于 0.9 的 SGD 优化器,初始学习率为 0.05,使用余弦规则的调度器,长度为 16 的 cutout 正则化,drop-path of probability 为 0.2。

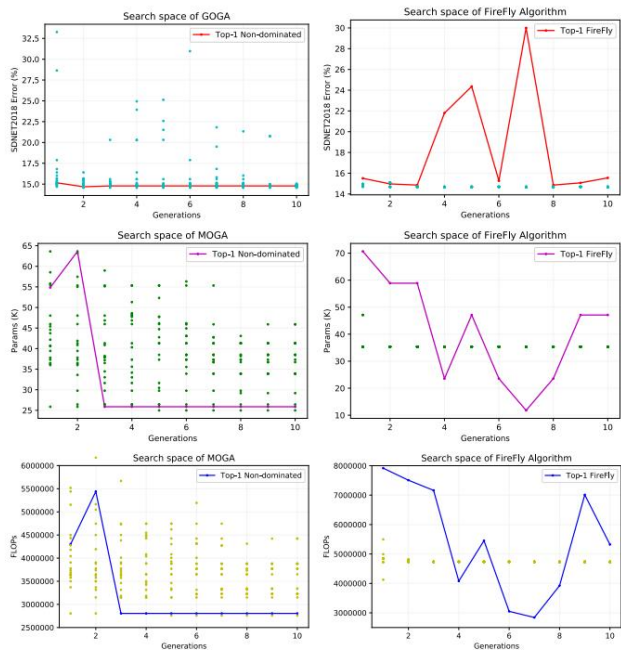
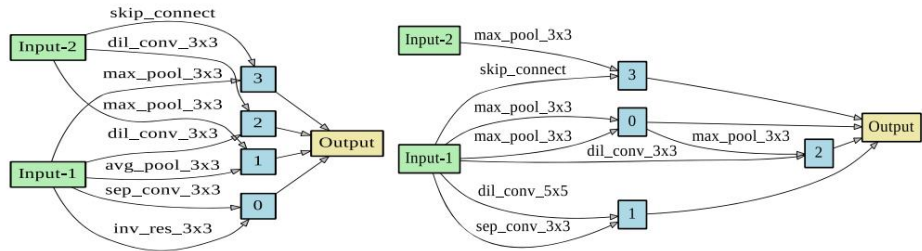


图 7. MOGA-Net 和 FA-Net 的搜索空间



(a) MOGA-Net 正常和减少单元格。

根据对 SDNET2018 数据集的模型搜索,MOGA 在所有性能指标(包括错误率、参数数量和 FLOPs)中找到了一个有效的模型。从图 7 可以看出,MOGA 可以在每一代中找到最有效的模型。然而,萤火虫算法的功能与 MOGA 不同,因为它会将每条染色体调整为每一代中性能最好的染色体。据观察,这两种算法具有相似的性能。

相比之下,MOGA种群的染色体是多种多样的,因为新兴种群使用交叉,有些是通过突变形成的,从而导致一条染色体的后代多样化。此外,我们还使用了 Firefly 算法群体的多目标排名来为每个性能指标找到合适的群体。如图 7 所示,每一代最合适的种群将被定义为 Firefly 算法的 Top-1 Firefly 和 MOGA 的 Top-1 Non-dominated。

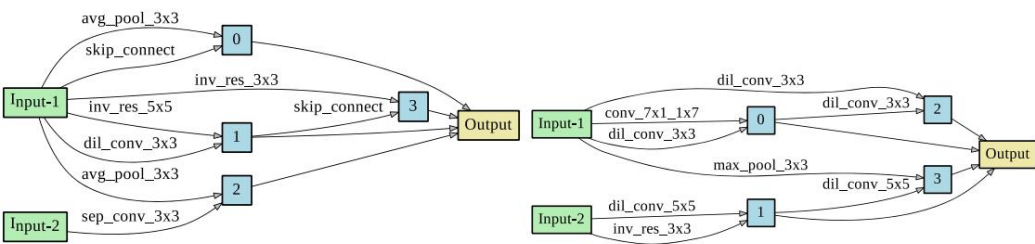
每个性能指标都具有良好性能的染色体是从上一代中选择的,其中从 MOGA 中获得的最佳染色体称为 MOGA-Net,而从 Firefly 算法中获得的最佳染色体称为 FA-Net,如图 8 所示。

C. 混凝土裂缝评价和SDNET2018

与模型搜索中定义的相同的超参数,除了从头开始训练网络是 100 个时期外,用于评估模型以确定正常和减少 MOGA-Net 和 FA-Net 的单元格。

表 2 展示了 MOGA-Net 和 FA Net 与其他最先进模型的比较。在考虑错误性能时,PC-DARTS 的错误最小。

但是,PC-DARTS 的参数比 MOGA-Net 参数多 24.8K,比 FA-Net 参数多 2.91K。MOGA-Net 的计算时间比 PC-DARTs 少 2.45 MMac,而 PC-DARTs 在计算时间方面优于 FA-Net,相差 0.25 MMac。



(b) FA-Net 正常和减少单元格。
图 8. MOGA-Net 和 FA-Net。

表2.混凝土裂缝的结果

模型	具体的 错误 (%)	SDNET 错误 (%)	参数 (K)	失败者 (MMac)
NASNet [22]	0.11	8.14	56.49	8.12
变形虫网 [23]	0.15	8.13	62.59	8.68
DARTS_V2 [26]	0.11	8.57	28.23	5.16
电脑飞镖 [36]	0.07	8.12	52.43	7.33
FA网络	0.16	8.89	49.52	7.58
MOGA网	0.22	9.68	27.63	4.84

表4. CINIC-10 的结果

模型	CINIC-10 错误 (%)	参数 (K)	失败者 (MMac)
NASNet [22]	11.22	3.83	624.23
变形虫网 [23]	11.04	3.14	506.29
DARTS_V2 [26]	10.95	3.34	547.47
电脑飞镖 [36]	10.94	3.63	576.06
FA网络	11.73	2.81	461.23
MOGA网	11月9日	2.53	413.16

D. 学习架构的可迁移性

MOGA-Net 和 FA-Net 已经用 Fashion-MNIST、CINIC-10 和 STL-10 数据集进行了测试,以确定模型的可靠性能,并将它们与其他最先进的模型进行比较。这些数据集的图像在图像大小上有所不同。例如,尺寸为 28×28 像素的 Fashion-MNIST,尺寸为 32×32 像素的 CINIC-10,以及尺寸为 96×96 像素的 STL-10。

改变了训练超参数,以评估
带有这些数据集的 MOGA-Net 和 FA-Net

所有单元格 (正常和减少单元格)的所有单元格 (正常单元格和减少单元格)设置为 20 层和 36 个初始通道,通过在 100 个时期从头开始训练网络,批量大小为 128,SGD 优化器,权重衰减为 0.0003,动量为 0.9,初始学习率为0.025,余弦规则的调度器,长度等于16的 Cutout正则化,概率下降路径为0.2,权重辅助塔为0.4。

表3. FASHION-MNIST 的结果

模型	时尚-MNIST 错误 (%)	参数 (K)	失败者 (MMac)
NASNet [22]	4.85	3.82	476.41
变形虫网 [23]	4.86	3.14	386.11
DARTS_V2 [26]	5.07	3.34	417.64
电脑飞镖 [36]	4.99	3.63	439.52
FA网络	5.01	2.80	351.60
MOGA网	4.85	2.52	314.80

表5. STL-10 的结果

模型	STL-10 错误 (%)	参数 (K)	失败者 (MMac)
NASNet [22]	16.01	3.83	5618.05
变形虫网 [23]	14.16	3.14	4556.62
DARTS_V2 [26]	13.27	3.34	4927.21
电脑飞镖 [36]	12.99	3.63	5184.50
FA网络	13.42	2.81	4151.02
MOGA网	15.37	2.53	3718.39

我们发现,当使用 Fashion MNIST 数据集测试 MOGA-Net 时,MOGA-Net 的误差值最小,与 NASNet 相等,而 MOGA-Net 模型的参数数量和计算成本最低,如表 2 所示。然而,PC-DARTS 分别实现了 CINIC-10 和 STL-10 数据集的最低误差值,如表 3 和表中所示。当我们考虑表 4 中的 FA-Net 时,这是令人震惊的,它在其他数据集中具有很高的错误值,但对于 STL-10 数据集,错误小于 MOGA-Net,与 DARTSv2 相同。FA-Net 的参数较少,计算成本低于 DARTSv2 模型。

四. 结论

本文提出了使用多目标遗传算法和萤火虫算法的神经结构搜索,重点是寻找具有小参数和较少计算成本的 CNN 模型。根据在 Fashion-MNIST、CINIC-10、STL-10、Concrete Crack 和 SDNET2018 数据集上的实验结果,我们发现 MOGA-Net 实现了

与其他最先进的模型（如 NASNet、AmoebaNet、DARTSv2 和 PC-DARTS）相比,参数数量最少且计算成本更低。同时,我们看到了萤火虫算法修改模型染色体的能力,使其看起来尽可能接近性能最佳的模型。

研究表明,所提出的算法适合部署在资源有限的设备上,例如物联网设备和嵌入式系统。

参考

[1] T. Fernando 等人,“用于医学异常检测的神经记忆可塑性”,神经网络,卷。127,第 67–81 页,2020 年 7 月。

[2] KI Withanage,I. Lee,R. Brinkworth,S. Mackintosh 和 D. Thewlis,“使用 RGB-D 相机进行跌倒恢复子活动识别”,IEEE 工业信息学汇刊,卷。12,没有。6,第 2312–2320 页,2016 年。

[3] B. Tu,X. Yang,N. Li,C. Zhou 和 D. He,“通过密度峰聚类进行高光谱异常检测”,模式识别快报,卷。129,第 144–149 页,2020 年 1 月。

[4] S. Zavrak 和 M. skefiyeli,“使用变分自动编码器从网络流特征中进行基于异常的入侵检测”,IEEE Access,卷。8,第 108346–108358 页,2020 年。

[5] K. Doshi 和 Y. Yilmaz,“监控视频异常检测的持续学习”,IEEE/CVF 计算机视觉和模式识别研讨会论文集 2020,第 254–255 页。

[6] SC Wong,V. Stamatescu,A. Gatt,D. Kearney,I. Lee 和 MD McDonnell,“跟踪一切:限制在线多对象识别中的先验知识”,IEEE 图像处理交易,卷。26,没有。10,第 4669–4683 页,2017 年。

[7] V. Chandola,A. Banerjee 和 V. Kumar,“异常检测:A 调查”,ACM 计算调查 (CSUR),卷。41,没有。3,页。2009 年 7 月 1–58 日。

[8] S. Akcay,A. Abarghouei 和 T. Breckon,“GANomaly:通过对抗训练进行半监督异常检测”,亚洲计算机视觉会议,Cham,2018 年,第 622–637 页。

[9] S. Yu, F. Xia, Y. Sun, T. Tang, X. Yan, and I. Lee.检测异常值具有基于查询的人工生成搜索条件的模式,IEEE 计算社会系统交易,卷。8,没有。第 1 页 134–147,2021 年 2 月。

[10] F. Carrara,G. Amato,L. Brombin,F. Falchi 和 C. Gennaro,“结合 GAN 和自动编码器进行高效异常检测”,2020 年第 25 届模式识别国际会议 (ICPR),第 3939 页–3946。

[11] D. Smolyak,K. Gray,S. Badirli 和 G. Mohler,“Coupled IGMM GANs with Applications to Anomaly Detection in Human Mobility Data”,ACM Trans.关于空间算法和系统 (TSAS),卷。6,没有。4,第 1–14 页,2020 年 6 月。

[12] F. Di Mattia,P. Galeone,M. De Simoni 和 E. Ghelfi,“关于 GANs for Anomaly Detection”,arXiv 预印本 arXiv:1906.11632, 2019。

[13] X. Gong,S. Chang,Y. Jiang 和 Z. Wang,“AutoGAN:生成对抗网络的神经架构搜索”,IEEE/CVF 计算机视觉国际会议论文集,2019 年。

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

[15] B. Zoph 和 Q. 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] S. Xie et al., “SNAS: Stochastic Neural Architecture Search”,arXiv 预印本 arXiv:1812.09926, 2018。

[17] S. Hu 等人,“DSNAS:直接神经架构搜索,无需 Parameter Retraining”,2020 年 IEEE/CVF 计算机视觉和模式识别会议论文集,第 12084–12092 页。

[18] C. Terrmritthikun,Y. Jamtsho 和 P. Muneesawang,“改进的使用快照集合和剪切技术的组合进行图像识别的剩余网络模型”,多媒体工具和应用程序,卷。79,没有。1,第 1475–1495 页,2020 年 1 月。

[19] C. Terrmritthikun,Y. Jamtsho 和 P. Muneesawang,“使用深度学习的 NUF-Net 模型进行设备上面部验证”,人工智能的工程应用,卷。85,第 579–589 页,2019 年 10 月。

[20] C. Terrmritthikun,Y. Jamtsho,J. leamsaard,P. Muneesawang 和 I. Lee,“EEEE-Net:早期退出进化神经架构搜索”,人工智能的工程应用,卷。第 104 页104397,2021 年。

[21] T. Elsken,J. Metzen 和 F. Hutter,“神经架构搜索:A 调查”,机器学习研究杂志,卷。20,没有。1,页 1997–2017,2019。

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

[23] E. Real,A. Aggarwal,Y. Huang 和 Q. 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,没有。2019 年 7 月 1 日。

[24] C. Liu 等人,“Progressive Neural Architecture Search”,2018 年欧洲计算机视觉会议 (ECCV) 会议记录,第 19–34 页。

[25] H. Pham 等人,“通过参数共享进行高效神经架构搜索”,2018 年国际机器学习会议 (PMLR),第 4095–4104 页。

[26] H. Liu,K. Simonyan 和 Y. Yang,“DARTS:可区分架构搜索”,arXiv 预印本 arXiv:1806.09055,2018 年。

[27] A. Krizhevsky,I. Sutskever 和 G. Hinton,“使用深度卷积神经网络进行 ImageNet 分类”,神经信息处理系统进展 2012,卷。25,第 1097–1105 页。

[28] 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。

[29] A. Gómez,Y. Achaerandio 和 P. Isasi,“进化卷积神经网络:手写识别的应用”,神经计算,卷。283,第 38–52 页,2018 年。

[30] X. Yang,“受自然启发的元启发式算法”,Luniver Press, 2010。

[31] H. Wang 等,“一种新的需求动态萤火虫算法水资源估算”,信息科学,卷。438 页 95–106,2018 年 4 月。

[32] N. Long,P. Meesad 和 H. Unger,“一种基于萤火虫的高精度心脏病预测算法”,专家系统与应用,卷。42,没有。21,第 8221–8231 页,2015 年 11 月。

[33] M. Ariyaratne,T. Fernando 和 S. Weerakoon,“使用改进的萤火虫算法 (MODFA) 求解非线性方程组”,群体和进化计算,卷。48,第 72–92 页,2019 年 8 月。

[34] 升。Li 等人,“基于多目标萤火虫算法补偿因素和精英学习”,下一代计算机系统,卷。91,第 37–47 页,2019 年。

[35] M. Tan 等人,“MNASNet:移动设备的平台感知神经架构搜索”,IEEE/CVF 计算机视觉和模式识别会议论文集 2019,第 2820–2828 页。

[36] Y. Xu 等人,“PC-DARTS:内存的部分通道连接高效架构搜索”,arXiv 预印本 arXiv :1907.05737,2020 年 4 月。

[37] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, H. Liu.图学习:一项调查,IEEE 人工智能交易,卷。2,没有。2,页。2021 年 109–127 号。