





# 残差神经的进化优化 调制分类的网络架构

埃尔玛佩伦达 , 斯里拉吉·拉詹德兰,  Gerome Bovet、Sofie Pollin 和 , 高级会员,IEEE,  
Mariya Zheleva , 会员,IEEE

**摘要** 自动调制分类在当前和未来的无线通信系统中受到了极大的关注。深度学习成为调制分类的强大工具,因为它允许联合鉴别原生特征学习和信号分类。然而,用于调制分类的神经网络架构的优化是一个手动且耗时的过程,需要深厚的领域知识和大量的努力。

大多数最先进的解决方案主要关注分类准确性,而忽略了网络复杂性的优化。本文提出了一种新颖的**双目标模因算法 BO-NSMA**,以搜索用于调制分类的最佳深度神经网络架构,以最大限度地提高分类精度并最大限度地降低网络复杂性。实验表明,BO-NSMA 的初始种群为 6 个个体,只有 10 代,它发现了一种深度神经网络架构,其性能优于所有人工设计的架构。此外,BO-NSMA 发现了第一个低复杂度的卷积神经网络架构,其性能比昂贵的循环神经网络架构略好,使网络复杂度降低了 2.9 倍,性能提高了 1.43%。与网络架构搜索的对应物相比,BO-NSMA 找到了最佳架构,准确率提高了 18.73%,网络复杂度降低了 82 倍。使用 Wilcoxon 符号秩检验验证结果。

**索引词** 调制分类、深度学习、网络架构搜索、多目标遗传算法。

## 一、引言

调制分类 (AMC) 是信号检测和解调之间的中间步骤,是设计未来与关键应用进行无线通信的智能收发器的组成部分

手稿于 2021 年 5 月 2 日收到;修订于 2021 年 8 月 5 日和 2021 年 10 月 13 日; 2021 年 12 月 16 日接受。出版日期 2021 年 12 月 23 日;当前版本的日期为 2022 年 6 月 9 日。这项工作得到了瑞士联邦国防采购办公室 (armasuisse) (项目代码 Aramis CYD-C-2020003)和北约和平与安全科学计划 G5461 资助的部分支持。协调本文审阅并批准发表的副主编是 Z. Li。(通讯作者:Erma Perenda。)

Erma Perenda 和 Sofie Pollin 在比利时鲁汶大学 ESAT 的 WaveCore 工作 (电子邮件:erma.perenda@esat.kuleuven.be, sofie.pollin@esat.kuleuven.be)。

Sreeraj Rajendran 在比利时 Sirris 的 EluciDATA 实验室工作 (电子邮件:sreeraj.rajendran@sirris.be)。  
Gerome Bovet 就职于瑞士国防部 Armasuisse 科技公司,3602 图恩,瑞士 (电子邮件:gerome.bovet@armasuisse.ch)。

Mariya Zheleva 就职于纽约州立大学奥尔巴尼分校计算机科学系,纽约州奥尔巴尼 12222 (电子邮件:mzheleva@albany.edu)。  
数字对象标识符 10.1109/TCCN.2021.3137519

2332-7731 c 2021 IEEE。允许个人使用,但再版/再分发需要 IEEE 许可。  
有关详细信息,请参阅 <https://www.ieee.org/publications/rights/index.html>。

在动态频谱接入 (DSA) 和资源分配方面。此外,它还是许多其他频谱传感应用的关键推动因素,例如信号监测、入侵者检测、干扰机识别以及众多监管和国防应用。

自动调制分类 (AMC) 的研究已经进行了 40 多年,使用了三个主要的方法论主题:基于似然 (LB)、基于特征 (FB) 和深度学习 (DL)。LB 方法将 AMC 表述为一个多重复合假设问题,其中假设的数量等于目标调制的数量 [1]。虽然在贝叶斯意义上是最优的,但 LB 方法需要有关所有信号和通道参数的先验知识,并且计算成本高 [2]。另一方面,FB 方法通过利用预先设计的判别特征以比 LB 方法 [3]、[4] 更低的计算成本执行分类。FB 方法的性能在很大程度上取决于所提取特征的判别性和噪声鲁棒性,并且预先设计正确的特征非常具有挑战性 [5]、[6]。与 FB 相比,DL 自动从原始同相/正交 (I/Q) 数据中学习无线电特征,并且优于 FB 方法 [7]–[11]。

由于能够从原始 I/Q 数据中共同学习判别特征并基于这些特征进行信号分类,DL 已被广泛用于 AMC。

可以区分两种基于 DL 的方法:递归神经网络 (RNN) [7] 和卷积神经网络 (CNN) [8]–[10]。由于 RNN 较高的计算成本和内存要求,CNN 已成为分类任务的首选。更深的 CNN 架构存在梯度消失问题,这使得它们不适合复杂的分类任务 [11],因为网络性能会随着深度的增加而降低。最近,受 RNN 的启发,提出了新的基于 CNN 的模型,例如残差神经网络 (ResNet) [12] 和深度神经网络的聚合残差变换 (ResNeXt) [13]。

这些模型优于最先进的 (SoA) CNN 模型,如 [8] 中基于 ResNet 的 AMC 模型和 [11] 中基于 ResNeXt 的 AMC 模型所示。ResNet 和 ResNeXt 是模块化架构,其中堆叠了预先设计的块。已经提出了 ResNet 块 [12] 和 ResNeXt 块 [13] 的几种设计。尽管在 AMC 中使用深度神经网络 (DNN) 取得了巨大成功,但设计高效且准确的 DNN 架构通常是一个手动、耗时的过程,需要深厚的领域知识。此外,许多 AMC 应用程序实时运行

并需要快速推理。DNN 架构复杂度越低,推理速度就越快。以下挑战使这一过程更加困难。

巨大的搜索空间:即使对于简单的 CNN 架构,搜索空间也非常大,因为自由度包括层数、每层过滤器的数量和过滤器大小。例如,让我们考虑一个简单的 ResNet-18 网络,它有 18 个层(其中 16 个卷积层(Conv)和 2 个池化层)。ResNet-18 的典型架构优化任务会考虑 16 层,具有 8、16、32、64 或 128 个过滤器,过滤器大小为 1 或 3。这会创建  $(5 \times 2)^{16} = 1016$  的大搜索空间可能的架构。随机搜索此类空间可能需要几天或几周的时间。最近,基于强化学习(RL) [14] 和遗传算法(GA) [15]-[17] 的两种启发式方法已在计算机视觉中广泛采用,以自动化网络架构搜索(NAS)。前者使用 RL 来指导搜索,而后者是反映自然选择的基于种群的元启发式 [18]。基于 RL 的方法 [14] 的计算成本过高,并且不容易应用于多目标优化问题(MOOP) [19]。

通常,在文献中,多目标 RL 存在两种不同的方法:单策略和多策略。单策略方法 [20] 通过特定的标量化方法将 MOOP 转换为单目标优化问题(SOOP),表示目标的偏好。具体的权重代表了偏好,并且在每次运行中都不相同。这种方法在计算和模型表示方面都是冗余的。最常用的标量化方法是加权目标的线性组合。在 [21] 中表明,任何基于目标线性组合的系统都无法为呈现非凸区域的问题产生良好的 Pareto 前沿近似值。许多现实世界的 MOOP 问题都具有这种性质,而标量化方法是解决此类问题的低效工具。相比之下,多策略方法 [22] 必须找到多种策略来满足目标之间的权衡。它可以同时(在单次运行中)或迭代地(每次运行一个策略)完成。多策略 RL 方法仍然具有很高的计算成本,即使在其扩展和优化版本 [22] 中也是如此。由于这些方法明确维护多个策略,因此它们很难扩展到目标之间的高维偏好空间。相比之下,GA 对于 MOOP [18] 非常有效,因为它们基于群体的特性可以在单次运行中获得一组解决方案。NAS 很少被考虑用于基于 DL 的 AMC [23]。

以数据集为中心的解决方案:大多数现有的人工 AMC DNN 架构都针对单组调制进行了优化 [7]、[9]。添加新的调制格式和/或更改输入功能很可能会降低 DNN 性能 [11]。因此,新的目标类触发了架构的重新优化。为了使这种重新优化易于处理,需要一个灵活的搜索空间和 GA 编码方案,以使其对输入特征变化具有鲁棒性。

大多数用于 NAS 的 GA 都忽略了这一点,并且在输入特征空间时需要新的搜索空间和编码方案

变化。[16] 提出了预先设计的块,这些块在输入特征更改时失败,如后文所示。参考文献 [23] 提出了一种在复杂特征空间上失败的浅层 CNN 架构,如 [11] 所示。

在忽略网络复杂性的同时最大化分类精度:所有人工设计的 AMC DNN 架构都专注于最大化分类精度,同时忽略了降低网络复杂性。

然而,这种人工设计的 DNN 架构可能不会在连接和超参数值方面得到优化。如前所述,GA 提供了一些解决 MOOP 的技术,但应用于 AMC NAS 的 GA 仍然是单目标驱动的 [23]。

除了 RL 和 GA,还有一些替代方法可以优化 DNN 架构。在 [24] 中,预先选择了人工制作的 DNN。然后应用基于贪心准则的剪枝来减少通过剪枝每层不重要的特征实现的训练参数的数量。该方法的性能主要取决于人工设计的初始架构。[25] 中考虑了 NAS 的知识蒸馏,其中 DNN 架构压缩是通过将环知识从训练有素的教师网络转移到更小更快的学生模型来完成的。该方法的计算成本明显低于 GA,并且由于不探索整个搜索空间,因此可能会得出次优解。

知识蒸馏可以与 RL 相结合,以加快基于 RL 的 NAS 的收敛,如 [26] 所示。

对于 AMC 应用程序,拥有具有高分类精度同时保持低复杂度的 DNN 架构非常重要。搜索此类架构的时间消耗并不重要,这使我们能够应用 GA 方法,这可能会找到接近全局最优的架构。

本文提出了一种称为 BO-NSMA (使用模因算法的双目标网络搜索)的新型 AMC 算法,以优化分类精度和网络复杂度。模因算法是指具有局部搜索 (LS) [18] 的 GA 的扩展。LS 算子的主要目标是将候选解决方案推向搜索空间中更有希望的区域,在这些区域很有可能找到最佳解决方案。因此,精心设计的 LS 算子可以加快 GA 的收敛速度,如本工作所示。网络复杂度的优化同时考虑了可变长度网络架构的连接和超参数。本文的主要贡献总结如下: ·对基于 GA 的多目标 NAS 用于 AMC 的首次研究。本文确定了所提出的 BO-NSMA 的关键组成部分,例如适应度共享 (FS)、局部搜索 (LS) 以及突变和交叉率的自适应性,使其能够找到接近帕累托最优的多样化种群正面。请注意,帕累托最优前沿是一组非支配个体,称为帕累托最优解。存在(可能无限)个 Pareto 最优解。·搜索空间和编码对 GA 收敛速度和 AMC 性能的影响得到了彻底的探索。

结果表明,BO-NSMA 可以找到一个非常接近 Pareto 最优前沿的多样化群体,该群体只有 6 个个体。此外,结果表明,BO NSMA 的性能优于所有人工设计的基于 DNN 的 AMC,最高可达 cc。精度提高 2%,最高可达 cc。网络复杂度降低 5 倍。

创建新的终止标准以有效地平衡搜索持续时间和获得的性能之间的权衡。

在本文的其余部分安排如下。第二部分介绍了用于 NAS 的最先进 (SoA) GA 方法的概述。问题陈述在第 III 节中介绍。第四节解释了为 AMC 的 NAS 提出的 GA 方法的结构。第五节介绍了所提出方法的深度性能分析及其与人工 AMC 方法的比较。第六节讨论了 BO NSMA 的时间复杂度、局限性和潜在应用。第七节简要介绍了结论。

二.相关工作

在 [27] 中可以找到 NAS 研究领域现有工作的详细概述。由于在上一节中为 NAS 选择 GA 方法是合理的,下面给出了基于 GA 的 NAS 研究的详细概述。

GA 已成功用于 NAS 的图像处理 [15]、[16]、[28]。通过将网络架构编码为染色体或个体,GA 方法努力优化 DNN 架构的权重和/或 DNN 架构的连接和超参数。用于 NAS 的 GA 文献可以分为两个主要流派。

A. 协同组合

GA 通过使用单个或多个 GA 优化连接和超参数以及 DNN 架构的权重。然而,所有提出的 GA 都假设一个固定长度的网络,该网络具有一个隐藏层的简单架构,例如前馈神经网络 (FFNN) [29]、[30]。

参考文献 [29] 寻求仅优化权重,而 [30] 寻求优化神经元数量及其权重。复杂的网络架构增加了单个表示的复杂性,并导致计算成本高昂的最佳权重搜索。另一方面,反向传播算法已成为权重优化的有效方法 [31]。此外,智能权重初始化可以略微提高反向传播性能,如 [15] 所示,其中权重初始化值作为额外的超参数编码到个体中,在几代中得到优化。

B. 支持组合

在这一工作流程中,GA 用于优化 DNN 架构的连接和超参数,同时使用反向传播等其他算法优化权重 [31]。 [32] 中提出了 FFNN 优化,[15]–[17]、[33] 中提出了 CNN 优化,[33]、[34] 中提出了 RNN 优化。最常考虑的超参数是隐藏层的数量、学习率、类型

优化器、过滤器数量、层的位置和激活函数。这个流中有几种不同的研究方法。它们的区别在于它们是否关注网络深度、优化问题的制定或 DNN 架构的构建方式。首先,考虑网络深度,有两类:固定[17]和可变[15]、[16]、[32]–[34]网络深度方法。

前者在网络深度设置为高于最佳值的情况下可能会浪费计算能力,而后者则试图找到最佳网络深度,从而提供计算效率更高的候选解决方案。

其次,考虑到优化问题的制定,有两类:单目标[15]、[17]、[33]、[34]和双目标[16]、[32]优化方法。前者仅最小化分类误差或均方误差,而后者最小化分类误差和网络复杂性。使用[32]中的标量化方法或[16]中的帕累托支配 (PD)方法将双目标优化问题转化为单目标。标量化方法对目标的权重非常敏感。它可能需要大量的迭代才能收敛到 Pareto 最优前沿的一小部分。第三,考虑到 DNN 架构的构建方式,有两类:层堆叠 [15]、[17]、[32]–[34] 和块堆叠 [16] 方法。对于复杂的分类问题,层堆叠方法不是首选。它需要一个更深的网络,该网络容易受到梯度消失问题的影响,梯度变得非常小,从而阻碍了网络训练 [12]。另一方面,精心设计具有恒等捷径的块可以使网络对梯度消失问题具有鲁棒性。这些身份捷径允许梯度信息通过层,即使在更深的网络中,也使得训练独立于网络深度。

除了添加身份快捷方式外,块的设计对 DNN 的性能有很大影响。例如,在 [16] 中,提出了一些预先设计的块,这些块具有相同的人工设计连接设置。在如此有限的搜索空间上运行的 NAS 可能无法找到最佳的 DNN 架构。

在调制分类的背景下,GA 方法已被用来提取和优化分类特征 [35]–[38] 或优化 DNN 架构 [23]。

然而,[23] 没有考虑 AMC 的 DNN 架构的连接和超参数的联合优化。此外,仅考虑了简单的 CNN 网络架构 [23]。本文旨在通过一种新的模因算法联合优化 AMC 的 DNN 架构的连接和超参数,该算法解决了上述 GA 方法在图像处理中的缺点。

三、问题定义

本节给出了 NAS 的数学表示对于AMC。还解释了 AMC 的输入。

A. 信号模型在为 AMC

定义 NAS 之前,让我们介绍调制信号作为基于 DNN 的调制分类器的输入。假设一个有源发射机发射一个信号,

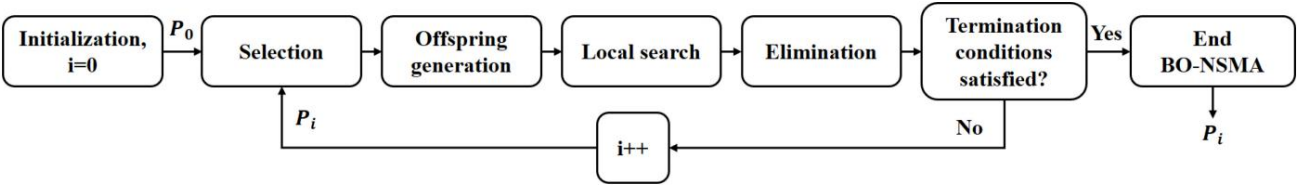


图 1. BO-NSMA 概述。

$s(t)$ ,在具有脉冲响应 $h_c$  的动态无线衰落信道上。假设接收器有一根天线,失真和噪声损坏的接收信号 $r(t)$ 为

$$r(t)=e^{j(\phi_0-2\pi\Delta ft)}s(t-\Delta t)h_c(t)+v(t),\tag{1}$$

其中  $\Delta t$  是定时偏移, $\Delta f$  是频率偏移,  $\phi_0$ 是相位偏移,  $v(t)$ 是加性高斯白噪声(AWGN),均值为 0,方差为  $2\sigma^2$ 假设接收器工作在同一中心频率作为发射机。接收到的信号 $r(t)$ 在时域中进行采样,  $N$ 个原始 I/Q 样本作为其输入。分类器接收到的 $N$ 个原始 I/Q 样本被称为一个实例,表示为一个 $2\times N$  维度的矩阵,其中第一行包含 $I$ 值,第二行包含相应的 $Q$ 值。例如,二进制相移键控 (BPSK) 调制信号的大小 $N=4$  的实例如下所示

$$[\text{我}/\text{问}]_{2\times 4}=\begin{matrix}&0.027&0.053&0.119&0.144&-0.014&0.003\\&0.005&0.035\end{matrix}.\tag{2}$$

AMC 分类器的任务是从已知 $N_{\text{mod}}$ 候选调制池中正确选择调制格式。

B. 调制分类网络架构搜索问题定义

AMC 的网络架构搜索 (NAS) 可以被视为一个双目标优化问题,其中分类精度应该最大化,同时,在计算成本和内存需求方面的网络复杂性应该最小化。当这项工作寻找最佳的 CNN 架构时,网络复杂度可以粗略地近似为模型的可训练参数总数。下面给出了所考虑问题的数学表示。

设 $Z\subset 2\times N$ 是上述原始 I/Q 样本的特征空间, $Y=\{1,...,c\}$  是标签空间,其中 $c$ 是考虑的调制类别的数量。因此,训练数据集可以定义为  $D=\{(z_i,y_i)|z_i\in Z,y_i\in Y\}$ 。分类器被定义为将输入特征空间映射到标签空间的函数, $f:Z\rightarrow c$ 。

我= 1,

AMC分类器采用带有交叉熵损失的Softmax输出层进行分类。因此,捕获 DNN 学习到的特征的判别性质的分类风险为

$$RL(f)=ED[\underset{n}{L}(\underset{c}{f}(z;\theta),y_z)]=\sum_{n=1}^N\sum_{c=1}^cy_{ij}\log f_j(z_i;\theta),\tag{3}$$

其中  $\theta$  是分类器的参数集, $L$  是交叉熵损失,  $y_{ij}$ 是实例 $z_i$ 的标签 (表示为one-hot 编码标签的第  $j$  个元素),  $f_j$ 表示分类器函数的第 $j$  个元素 $F$ 。分类风险越低,分类准确率 $p_c$ 就越高。因此,DNN 架构的联合连接和超参数优化可以表示为

$$\begin{matrix}\text{最小化 } F(z)=(RL(f(z)),\#\theta) \text{ subject to } z\in \\ Z, f(z)\in A, \#\theta>0,\text{其中 } A \text{ 是架构搜索空间},\#\theta \text{ 是}\end{matrix}\tag{4}$$

所有分类器参数  $\theta$  中的可训练参数。给定  $A$ ,目标是为具有最少数量可训练参数 $\#\theta$  的分类器找到最佳架构 $f(z)$ ,以便在训练这些参数后架构可以实现最小分类风险 $RL$ 。

四、方法

本节介绍所提出的 BO-NSMA (双目标网络搜索模拟因算法),其流程图如图 1 所示。BO-NSMA 考虑块级设计并利用 Pareto Dominance (PD) with Fitness Sharing (FS) 策略来求解双目标优化问题。同时,与 [16] 相比,它允许具有不同连接设置的块,使用扩展的超参数搜索空间。由于这项工作考虑了复杂的网络架构,因此采用反向传播来使用默认的 Xavier 权重初始值设定器 [31] 进行权重优化。第 II 节中提到的所有方法都使用绝对迭代次数来终止其 GA。这个终止标准是低效的并且可能导致不必要的计算。因此,这项工作采用平均 Hausdorff 距离来避免它。此外,自适应突变和交叉率增强了探索和利用。下面将解释拟议的 BO-NSMA 的组成部分。

A. 搜索空间和编码

受 ResNet [12] 和 ResNeXt [13] 架构的启发,基于 CNN 的网络架构被设计为块数的串行融合,后跟一个全局池化层和几个密集层。每个块被定义为 $w$ 个分支与 $d$ 个Conv 层的并行融合,其输出首先连接,然后与Identity分支合并,如图 2 所示。以 ppool 的概率,每个块后面跟着一个池化层。考虑了合并函数的不同变体,包括乘法、加法和连接。

Multiply和Add是逐元素操作,而



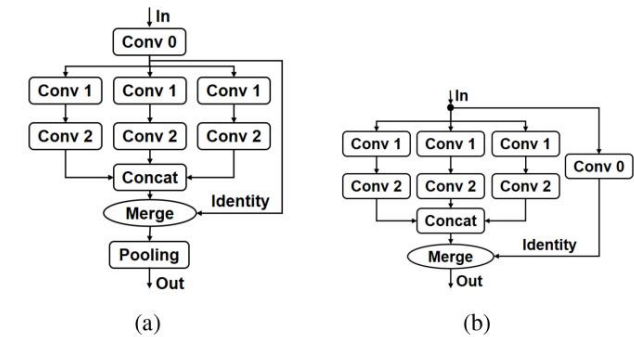


图 2. 块结构示例,  $w = 3$ ,  $d = 2$  且 (a) Dim.在 Conv 和 Identity分支之前进行缩减,然后是池化层; (b) 省略减少身份分支,并且没有池化层。

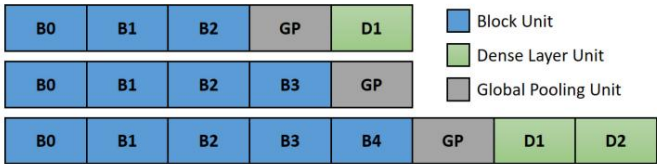


图 3. 个体基因型示例。

Concat沿列轴 (轴 = 1)完成。传统上,CNN 将底层信号的空间属性捕获为分类特征。然而,这些空间属性本质上对噪声条件很敏感,可能会导致性能显著下降 [11]。这项工作建议通过扩展合并功能的搜索空间来解决这个问题,这可能会启用捕获类似累积量信号属性的新功能的提取。为了使用网络深度强制特征空间的降维,在 Conv 分支和 Identity分支 (图 2 (a))之前或在 Identity分支 (图 2 (b))中添加一个 Conv 层。网络中的第一个块的宽度等于 1,深度等于 1,并且没有 Identity分支,而对于其他块,宽度和深度从以下范围内随机选择: $w \in [1, w_{max}]$ ;  $d \in [1, d_{max}]$ 。

由于密集层需要更多数量的可训练参数,因此添加它们的概率为  $p_{dense}$ 。GA 寻求最佳块数、密集层数以及每个块的最佳深度和宽度。此外,GA 为每个架构层寻找最佳超参数值。表 I 给出了超参数的搜索空间。

一个人的基因型以几个块的列表形式给出,一个全局池化层,以及几个或没有密集层。个体的长度  $L$  等于块数、密集层数和一个全局池化层的总和。图 3 展示了个人基因类型的几个例子。个体的表型是建立在其基因型之上的 DNN 架构。

B.种群初始化

表 I 中给出的搜索空间可能会导致非常复杂的网络架构。已经为 AMC [7]、[8]、[11] 提出了许多人工设计的 DNN 架构,并且可以帮助更智能地设计种群初始化。因此,称为可训练参数的最大允许数量的控制参数  $\theta_{max}$ ,

表一  
编码到个体中的超参数

Unit	Hyperparameters	Search space
Network	No. of blocks	[1, 10]
	No. of dense layers	[0, 4]
Block	width	[1, 32]
	depth	[1, 4]
	Merge function	{Add, Multiply, Concat}
	Dim. reduction	{Before, After}
	Pooling	{Yes, No}
Conv	Filters	{4, 8, 16, 32, 64, 128}
	Kernel size	{1, 3, 5, 7}
	Activation	{relu, selu, tanh, linear}
Pooling	Type	{Max, Average}
Global pooling	Kernel size	{2}
		{Average, Flatten}
Dense	Units	[32, 256]
	Activation	{relu, selu, tanh, linear}

具有由人工 SoA 确定的值,被引入。即使网络复杂性有限,仍有许多架构需要探索。种群初始化过程在算法 1 中进行了解释。个体的初始化包括添加块单元 (第 11-17 行)、添加全局池化单元 (第 18-19 行)和添加密集层 (第 20-22 行)。如果个体的可训练参数数量大于  $\theta_{max}$ ,它将被丢弃并重新初始化,直到生成的候选者满足目标可训练参数数量或试验次数  $l$  未达到最大值。虽然这可能会延长初始化时间,但它会导致替代复杂网络体系结构引起的总体时间成本低得多。

C. 健身评估

适应度评估分三个步骤进行:(1) 对可训练参数进行计数,(2) 通过预定义数量的 epochs  $N_{epochs}$ 训练解码个体,以及 (3) 在验证数据集上评估训练模型。

可训练参数的数量和验证分类的准确性是在后代生成和消除过程中使用的目标。Adam 优化器 [39] 在这项工作中采用了学习率为 0.001 的方法。这种学习率是较低速率下的缓慢收敛和较高速率下的不准确结果之间的合理权衡。

请注意,Adam 优化器提供了梯度归一化和动量,这使得学习率仅对更新学习率之前的初始学习很重要。

我们选择的学习率是基于先前工作 [39] 的建议,其中 0.001 的学习率适用于大多数问题。

D. 后代

GA 反映了自然选择的过程,其中选择最适合的个体进行交配以产生下一代的后代。GA 中的自然选择是通过选择、交叉和变异算子执行的 [18]。

1)选择:确定性的 k-tournament [18] 用于选择个体进行交配而无需更换,其中  $k$  个个体被随机评估,并且

算法 1:种群初始化输入:表中给出的输入参数。l输出:初始化	
1	种群P0 1 Po ← ∅ 2 for i ← 1 to λ do j ← 0 while True
2	do j ← j + 1 Individual ← Null nb ← Uniformly generate
3	an integer between [1, NB ] nd = 0 r ← Uniformly生成一个
4	介于[0, 1]之间的数if r ≤ pdense then nd ← 统一生成一个介
5	于[0, ND ]之间的整数_list ← [] block0 ← 随机初始化
6	一个block单元, d = 1, w = 1, 没有Identity分支, 和
7	ppool_list ← _list ∪ block0 for j ← 1 to nb
8	do
9	
10	
11	
12	
13	
14	
15	
16	←
17	统一生成一个介于 [1, wmax ] d ← 之间的整数
18	
19	统一生成一个介于[1, dmax ]之间的整数block ← 随机初始化一个
20	block单元d , w , ppool_list ← _list ∪ block
21	
22	gp ← 随机初始化一个全局池化单元_list ← _list ∪ gp for j
23	← 1 to nd do dl ← 随机初始化一个密集层单元_list ← _list
24	∪ dl Individual.units ← _list Individual.accuracy ← 0.0
25	Individual.complexity ← count_#0 ()如果
26	Individual.complexity < #0max or j == l then P0 = P0
27	∪ Individual break
28	
29	
30	
31	返回P0

最好的一个被选中。确定性 k-tournament 选择不需要任何总体知识,也不需要像轮盘赌方法等其他选择方法那样的可量化质量度量。相反,它使用排序关系来比较和排名任何两个人。此外,选择压力很容易通过改变锦标赛规模k [40] 来控制。由于考虑的问题有两个目标,因此使用众所周知的帕累托优势 (PD) 概念来确定哪个个体更好。如果满足以下条件之一,则称一个人支配另一个人: (1)它具有更高的分类精度,以及更少或相等数量的可训练参数,或 (2)它具有更高或相等的分类准确性,以及较少数量的可训练参数。对于每个个体,支配它的个体都被计算在内。受其支配的个体数量越少的个体被视为越好。

2)交叉:交叉算子类似于自然繁殖,通常以1的比率进行。但是,这样的比率可能会导致基因差的父母交配,从而导致后代表现不佳。

生存选择很可能会消除不良后代,从而减缓或完全停止 GA 进程。因此,重要的是产生好的后代以加速解决方案搜索并增加

算法 2: BO-NSMA 中的突变	
1	输入:个体x,表中给出的输入参数。l输出:变异个体 x^ 1 r ←
2	Uniformlygenerateanumberbetween[0, 1] 2 if r > px then
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	

后代的存活率。为此,开发了受 Q 学习 [41] 启发的自适应交叉率。具体来说,个体是否适合交配的轨迹是通过编码有关其交叉率的信息来保持的,初始化种群P0中的每个个体的交叉率为 1。交叉算子将两个父母作为输入,parent1, parent2,返回两个后代,offspring1, offspring2。在应用交叉操作之前,后代是所选父代的纯副本,即offspring1继承了parent1 的所有属性,而offspring2继承了parent2 的所有属性。应用交叉算子后,后代交叉率更新如下:poffspringi cr = γ pparenti cr + (1 - γ) poffspringi ; i = 1, 2, (5) 其中 γ 是学习率,范围从 0 到 1, pparenti cr表示 parenti 的交叉率,而poffspringi c 是 offspringi的验证分类准确率。γ的值在实践中大多设置为0.3,对于更高的值,Q-learning变得不稳定[41]。直观上,最大化目标分数的奖励越高,交叉率就越高。为了避免种群中的重复,对于没有交叉算子 (其父母的副本)生成的后代,应用比率为 1 的变异算子。

c

表示

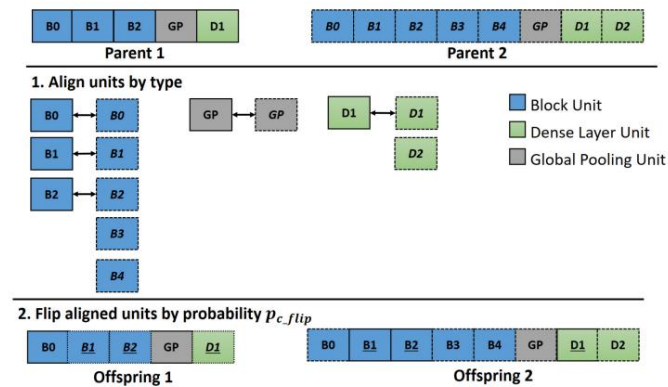


图 4. 交叉示例。

3) 变异:由于对齐单元的均匀交叉不影响后代的长度,因此每个后代都继承父代的长度。为了允许后代长度可变性和超参数的值发生变化,引入了以下变异运算符:添加新的、复制的、删除的和重置的。概率 $p_{m\_unit}$ 与个体的长度 $L$ 成反比,一个统一选择的变异算子将应用于个体的每个单元。Add new运算符添加一个新单元,其中包含随机选择的超参数值。Duplicate运算符添加一个与当前单元具有相同结构和超参数设置的单元。Delete运算符删除单元。

Reset运算符使单元保持相同的结构,但随机更改其超参数的设置。算法 2 中解释了变异过程。类似地,每个个体都将有关变异率 $p_{mr}$ 的信息编码为交叉率,该信息会在几代人之间进行调整。初始化种群在一定范围内随机选择突变率。高突变率会在个体长度上引入更多探索。使用对数正态变换 [42] 更新突变率,如下所示:

$$p_{mr}^{new} = 1 + \frac{1 - \text{波尔兹曼}}{\text{先生}} \exp - \tau N(0,1) - 1, \quad (6)$$

其中  $\tau$  是自适应速度控制参数 (设置为 0.22),  $N(0, 1)$  是具有零均值和单位方差的正态变量。突变率的对数正态变换使它们保持在 0 和 1 之间,并且已被证明是突变率自适应的有效技术 [42]。

E. 本地搜索

在应用淘汰策略之前,将局部搜索 (LS) 应用于最佳和最差个体 (后代 + 父母)。最好的个体不受任何其他个体支配,属于帕累托最优前沿。相反,最差的个体拥有最多的个体来支配它们。LS 探索最差个体的社区,以找到可能有机会生存的更健康的邻居。LS 包括将突变应用于两次运行的选定个体。仅当满足以下条件之一时,选定的未变异个体才被其变异版本替换: (1) 变异个体

算法 3: BO-NSMA输入:表中给出的输出:种群1 Po ← 使用算法 1 2 i ← 1 3 初始化种群while True do Pi ← Pi-1 for j ← 1 to μ/2 do parent1, parent2 ← run k-tournament selection without replacement r ← uniformly generate a [0, 1] 之间的数字crossover\_done ← False

```
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

返回Pi

具有更高的分类准确率,其可训练参数数量增加不超过5%; (2)变异个体的可训练参数较少,分类精度下降不超过0.5%。

F. 淘汰策略交叉和变异算子

产生 $\mu$  后代。淘汰策略[18]选择 $\lambda + \mu$ 策略,合并父母和后代,为下一代选择 $\lambda$ 最好的个体。最佳个体选择是根据修改后的分类精度,通过使用适合度共享 (FS)进行多样性提升。FS 是最成功和广泛使用的多样性促进方法。虽然 [16] 使用拥挤方法来促进多样性,但这项工作出于以下原因选择了 FS。

尽管 FS 的计算成本高于拥挤,但 FS 倾向于鼓励在空间的未探索区域进行搜索,并有利于形成稳定的子种群 [43]。

相反,由于替换错误,拥挤在某些情况下难以保持稳定的子种群。

多样性促进方法的完整概述可以在 [44] 中找到。修改后的分类精度为

给出如下:

$$p_c = \text{个人电脑} \times \frac{1 - \frac{d(x, r)}{\sigma}}{\sum_{r \in N_i(\sigma(x))} 1}, \quad (7)$$

其中  $\sigma$  表示相异性阈值,  $d(x, r)$  是个体  $x$  和个体  $r$  之间的距离,  $\alpha$  是调节共享形状的常数参数  $\sigma(x)$  表示个体函数的  $\sigma$  邻域, 和  $N$

$\sigma(x) = \{r \in \text{ual } x \mid d(x, r) \leq \sigma\}$ 。 当前种群  $P$  中的

距离  $d(x, r)$  计算为个体归一化复杂度与分类准确度之间的欧氏距离, 如下所示:

$$d(x, r) = \sqrt{\frac{(\# \theta_x - \# \theta_r)^2}{\# \theta_{\max}^2} + (p_c, x - p_c, r)^2}, \quad (8)$$

其中  $\# \theta_{\max}$  是可训练参数的最大允许数量。由于最大距离可达  $\sqrt{2}$ , 故  $\sigma$  设为 0.2。形状参数  $\alpha$  设置为 2, 确保  $\sigma$  邻域内的高多样性压力。

G. 终止策略未学习的终止标准

是 GA 的主要缺点之一。可能需要无数次迭代才能在大型搜索空间中获得最佳全局解决方案。

由于 GA 的每次迭代都非常昂贵, 因此有必要有适当的终止标准, 这表明进一步计算变得不必要的时间点, 因为它们没有获得实质性的性能改进。采用以下终止条件 (算法 3 中的第 27-34 行): (1) 迭代次数大于或等于固定数,  $J$  先验决定; (2) 达到可接受的解决方案 如果算法生成的个体的分类精度高于预定义值  $p^c$  且可训练参数低于预定义数量  $\# \theta$ , 则 GA 将终止; (3) 当最后  $H$  次迭代种群没有任何改善, 即最后  $H$  次迭代中各代之间的差异小于某个收敛阈值时, 。

每一代都表示为一组 Pareto 点, 其大小等于种群大小  $\lambda$ 。帕累托点表示一个个体  $x$ , 由其属性的有序对  $(1 - p_c, x, \# \theta_x)$  表示。为了测量两个 Pareto 集之间的相似性, 使用了众所周知的平均 Hausdorff 距离, 这是在多个研究领域广泛使用的测量不同对象之间距离的工具 [45]。BO-NSMA 中两个 Pareto 集之间的平均 Hausdorff 距离,  $U = \{u_1, u_2, \dots, u_\lambda\} \subset R^2$  和  $V = \{v_1, v_2, \dots, v_\lambda\} \subset R^2$

定义如下:

$$\Delta p(U, V) = \text{最大值} \frac{\lambda}{\lambda_i = 1} \frac{1}{\text{距离}(\text{用户界面}, V)^p}, \quad (9)$$

$$\frac{\lambda}{\lambda_i = 1} \frac{1}{\text{距离}(\text{六}, \text{你})^p} \frac{1}{p} \quad (10)$$

其中  $\text{dist}(u_i, V)$  是从  $u_i$  到集合  $V$  的最小欧氏距离,  $\text{dist}(v_i, U)$  是从  $v_i$  到集合  $U$  的最小欧氏距离,  $p$  是异常值惩罚的控制因子。

$p$  的值越高, 异常值受到的惩罚越多。由于  $\Delta p$  用作终止条件, 因此  $p$  设置为 1。

五. 绩效评价

本节解释用于模拟的实验设置, 包括选定的基线、数据集和实现细节。给出并分析了所获得的结果。

A. 实验设置 1) 基线: 这项工

作采用人类专家手动设计的文献中的四个基线: LSTM [7], ResNeXt [11], ResNet [8] 和 1D-CNN [8]。此外, 这项工作图像处理中采用了 GA NAS 的两个最新基线: NSGA-Net [16] 和 EvoCNN [15]。前者是针对块级 NAS 采用 PD 的双目标优化, 而后者是针对层堆叠 NAS 的单目标优化。NSGA-Net 基于一些预先设计的具有固定超参数的块来搜索网络架构。相比之下, EvoCNN 试图优化 DNN 架构中每一层的超参数, 包括权重初始化值。当这些基线应用于 2D 图像处理问题时, 架构中的每个 2D 层都被相应的 1D 层替换, 同时保持所有其他超参数相同。

2) 数据集: 使用了两个调制集: (1) 一个 Baseline 集, 包含  $N_{\text{mod}} = 11$  种低阶调制格式: BPSK、QPSK、8-PSK、16/64-QAM、PAM4、GFSK、CPFSK、BFM、DSB-AM 和 SSB-AM; (2) 一个扩展集, 包含简单调制和 9 个附加调制: OQPSK、32/128/256-QAM、16/32/64/128/256-APSK ( $N_{\text{mod}} = 20$ )。

I/Q 样本是在信噪比 (SNR) 增加 (−6 dB 至 18 dB) 时生成的。只要有足够的标记数据可供使用, 不同信道模型 (AWGN、Rayleigh、Rician) 的 DNN 模型的性能遵循相同的性能趋势, 如 [11] 所示。在这项工作中, 重点是 BO NSMA 的性能评估, 其中信道建模没有发挥重要作用。因此, 信道被建模为简单的 AWGN。对于 SNR 和调制类型的每个组合, 分别为基线集和扩展集生成 1000 个实例, 大小分别为  $N = 128$  和  $N = 1024$ 。种子用于生成随机互斥实例索引, 然后用于将数据分为三个子集: 分别以 80:10:10 的比例进行训练、验证和测试。

3) 实现细节: BO-NSMA 在 Python [46] 中实现。每个评估的 AMC 方法都是使用 TensorFlow [47] 实现的。健身评估训练在  $N_{\text{epochs}} = 10$  个时期和 256 的批量大小上进行。模型在具有八个 Nvidia RTX 2080Ti 卡的 GPU 服务器上训练和评估。BO-NSMA 输入参数的默认值 (见表 II) 在所有实验中保持不变。人口规模设置为较低的值,  $\lambda = 6$ , 因为



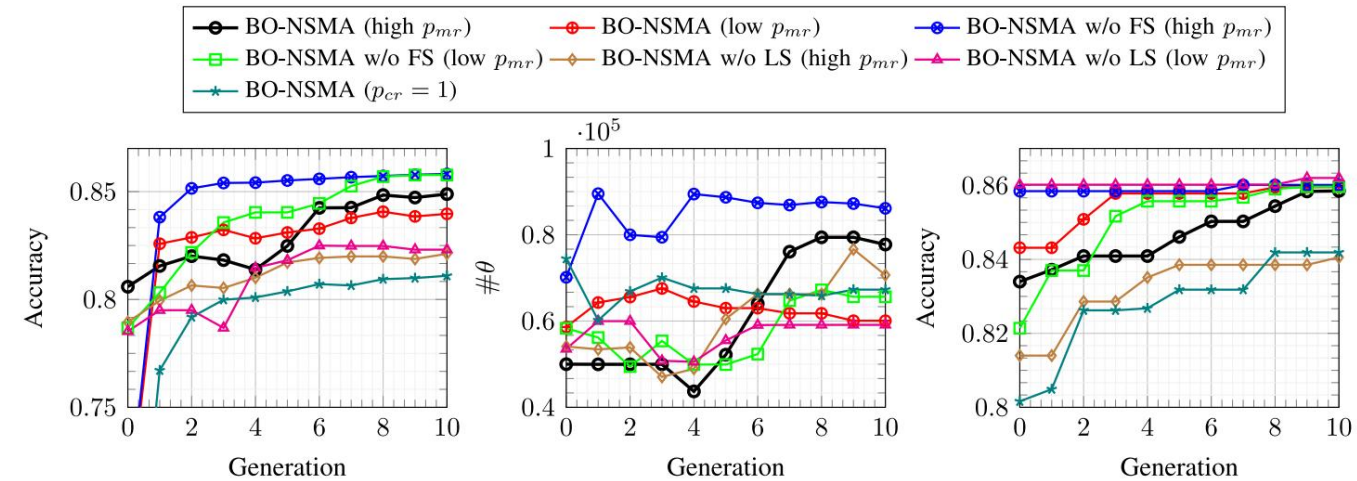


图 5. BO-NSMA 组件及其影响:平均精度 (左上) ;可训练参数的平均数量 (右上) ;几代人的最大准确度 (底部) 。

表二  
BO-NSMA输入参数

Name	Notation	Default Value
Population size	$\lambda$	6
Offspring size	$\mu$	6
Max. no. of blocks	$N_B$	8
Max. no. of Dense layers	$N_D$	4
Max. block width	$w_{max}$	32
Max. block depth	$d_{max}$	4
Probability of adding a Dense layer	$p_{dense}$	0.4
Probability of adding a Pooling layer	$p_{pool}$	0.5
Max. allowed no. of trainable parameters	$\# \theta_{max}$	100,000
Absolute no. of iter.	$\mathcal{I}$	10
No. of initialization trials	$\mathcal{I}$	100
No. of iter. for convergence check	$\mathcal{H}$	3
Convergence threshold	$\epsilon$	$10^{-4}$
No. of epochs	$N_{epochs}$	10
Probability to flip units in crossover	$p_{c\_flip}$	0.6
Optimal classification accuracy	$\hat{p}_c$	0.9
Optimal no. of trainable parameters	$\# \theta$	10,000
Tournament Selection Parameter	$k$	2
Max. length of individual	$L_{max}$	20

每个人的评估都需要大量计算。确定性锦标赛选择的k参数设置为 2,这为交叉算子选择每个个体的机会很大。所有呈现的分类精度均在  $[-6, 18]$  dB 的整个 SNR 范围内取平均值。

B. 结果

1) BO-NSMA 的性能评估:除了 GA 对任何 GA 共有的几代性能外,还有两个定性指标可以用来评估某个多目标 GA 对于给定问题的好坏程度:(1) population accuracy,即确定种群与 Pareto 最优前沿的相似程度,以及 (2) 种群多样性,即评估个体在

人口。请注意,帕累托最优前沿表示一组非支配个体。牢记这一点,下面证明了为什么 BO-NSMA 使用基线数据集按照第 IV 节中的描述进行设计。由于扩展数据集的评估遵循与基线数据集相同的性能趋势,因此下面给出的结论对扩展数据集仍然有效。BO-NSMA 表示所提出的具有应用 FS、LS、高突变率、 $p_{mr} \in [0.5, 1]$  和自适应交叉率的 GA。为了评估它们对几代人所提到的定性指标的影响,进行了七个实验:(1)BO-NSMA,(2)具有低突变率 $p_{mr} \in [0.05, 0.2]$  的 BO-NSMA,(3)没有突变率的 BO-NSMA FS 在淘汰中保留  $\lambda$  个个体具有最高的分类准确率,(4) BO-NSMA 具有低突变率 $p_{mr} \in [0.05, 0.2]$  并且没有 FS 在淘汰中并保留  $\lambda$  个个体具有最高的分类精度,(5) 没有 LS 的 BO-NSMA,(6) 具有低突变率 $p_{mr} \in [0.05, 0.2]$  且没有 LS 的 BO-NSMA,(7) 恒定交叉率等于 1 的 BO-NSMA。

图 5 展示了 BO-NSMA 的性能,显示了整个种群的平均分类精度 (左上)、整个种群的平均可训练参数数 (右上)以及最佳分类器的最大分类精度人口中的个人 (底部)。请注意,分类准确度值是在  $[-6, 18]$  dB 的整个 SNR 范围内的平均值。图 6 (右)显示了第 10 代的 Pareto 前沿近似。帕累托最优前沿如图 6 (右)中的实线所示,性能最佳的方法应该紧随其后。图 6 (左)显示了两代种群之间的 Pareto 集差异,这对于 GA 收敛率监测很重要。a) BO-NSMA performance over generations:由于没有 FS 的 BO NSMA 只关注最大化分类精度,它在第十代达到了最好的平均和最大精度 (见图 5 (左上和下))。

高突变率引入了高水平的搜索探索,对 GA 性能具有不可预测的影响。

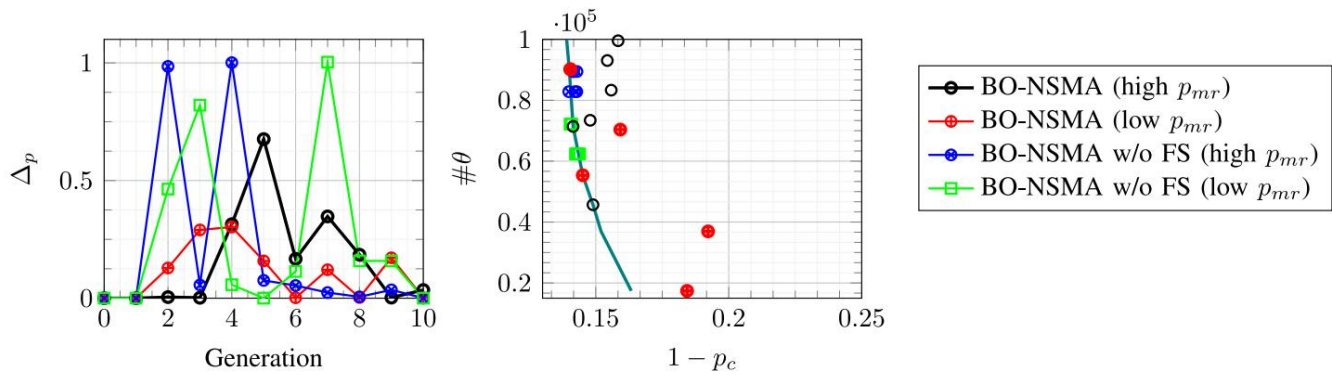


图 6. 两个后续世代的平均 Hausdorff 距离 (左)。第 10 代的 Pareto 前沿近似 (右)。

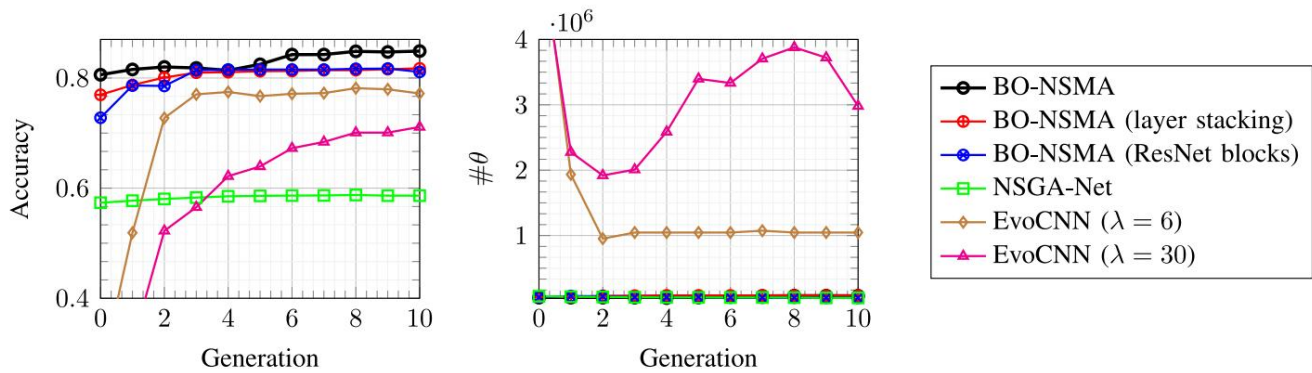


图 7. 搜索空间和编码对几代人的平均准确度 (左上) 和可训练参数的平均数量 (右上) 的影响。

对于有和没有 FS 的 BO-NSMA, LS 结合高突变率总能提高准确性。然而, 没有 LS 的 BO-NSMA 对于低突变率效果更好。

因此, 更多的搜索探索会增加可训练参数的平均数量。每个具有低突变率的 BO-NSMA 收敛到几乎相同数量的可训练参数, 大约 60k。相比之下, 具有高突变率的 BO-NSMA 案例会产生大约 8 万个可训练参数 (见图 5 (右上))。此外, 图 5 证明 LS 加快了 GA 的收敛速度。具有恒定交叉率的 BO-NSMA,  $p_{cr} = 1$  可能与贫穷的父母交配, 导致更差的表现。与具有自适应交叉率的 BO-NSMA 相比, 第十代的平均准确率降低了 5%, 最大准确率降低了 2%。

b) 种群准确性和多样性: 图 6 显示具有 FS 的 BO-NSMA 为两种突变率提供了多样化的种群。另一方面, 没有 FS 且突变率高的 BO-NSMA 在第六代后收敛到一个最优 Pareto 点, 而没有 FS 且突变率低的 BO-NSMA 收敛缓慢, 在第十代时有两个最优 Pareto 点可以还是会被注意到。

具有高突变率的 BO-NSMA 找到六个不同的个体和两个最佳 Pareto 点, 而具有低突变率的 BO-NSMA 找到五个不同的个体和三个最佳 Pareto 点 (图 6 (右))。与低突变率的 BO-NSMA 相比, 高突变率的 BO-NSMA 发现的个体更接近 Pareto 最优前沿。

因此, 可以说具有高突变率的 BO-NSMA 在种群准确性和多样性方面找到了最佳种群。

综上所述, 没有 FS 的 BO-NSMA 实现了最佳平均和最大准确度, 但代价是多样性损失。相比之下, 具有 FS 的 BO-NSMA 缓慢收敛到最优 Pareto 点, 但它提供了多样化的种群。高突变率使群体更准确, 更接近帕累托最优前沿。

2) 搜索空间和编码对性能的影响: 任何 GA 找到的解决方案在很大程度上取决于给定的搜索空间和个体的表示。为了评估所提出的搜索空间有多好, 我们探索了它与著名的 ResNet 块 [12] 和具有层堆叠的简单网络相比的优势, 同时保留了 BO-NSMA 的所有组件 (FS、LS、后代) 相同。

ResNet 模块的设计有  $d = 3$ 、 $w = 1$  和 Add as merge function [12]。因此, 为了证明编码的影响, 进行了五个实验: (1) 使用建议编码的 BO-NSMA, (2) 使用层堆叠的 BO-NSMA, (3) 使用 ResNet 块堆叠的 BO-NSMA, (4) NSGA-Net [16] 及其预先设计的块, 以及 (5) EvoCNN [15]。每个实验都使用基线调制集。EvoCNN 堆叠了 Conv、pooling 和 dense 层 (最大层数设置为 15)。

图 7 显示, 与具有层堆叠和 ResNet 块堆叠的 BO-NSMA 相比, 具有所提出的块设计的 BO-NSMA 在第六代之后的平均分类精度提高了 4%, 最大实现分类精度提高了 4%。此外, 带有 ResNet 块堆叠的 BO-NSMA 找到了可训练参数平均数量最少的群体。NSGA-Net 实现了非常差的 AMC 性能, 其中每个找到的架构都是

表三  
BO-NSMA TOP-1准确性和相应的# $\theta$  VS基线

Model	Baseline dataset		Extended dataset	
	Acc.(%)	# $\theta$	Acc. (%)	# $\theta$
LSTM [7]	86.17	200,075	46.49	201,236
ResNet [8]	85.58	255,115	79.71	313,620
ResNeXt [11]	85.72	85,051	80.40	86,212
1D-CNN [8]	82.01	100,811	79.54	142,932
<b>BO-NSMA</b>	<b>87.60</b>	<b>68,851</b>	<b>83.20</b>	<b>76,372</b>
EvoCNN [15]	80.30	450,157	64.47	6,261,901
BO-NSMA (layers)	83.73	83,115	82.12	72,708
BO-NSMA (ResNet blocks)	85.07	58,591	82.09	45,284

过度拟合。尽管 NSGA-Net 采用 PD,但其提出的具有预先设计的块的架构搜索空间和固定的超参数值阻止 GA 根据所考虑的问题优化此类架构。相比之下,EvoCNN 为超参数值提供了更多的自由度,导致与 NSGA-Net 和 BO NSMA 相比复杂的网络架构。此外,人口规模  $\lambda = 6$  的 EvoCNN 在第二代之后过早地收敛到一个非最佳 Pareto 点。超参数值的高自由度和不受控制的种群初始化需要更高的种群大小以避免过早收敛。因此,EvoCNN 在  $\lambda = 30$  时运行。图 7 显示,人口规模越大的 EvoCNN 收敛时间越长,而人口规模越大,它找到至少一个帕累托最优点的机会就越大。

3) 与基线的比较:最后,将 BO-NSMA 找到的最佳个体的性能与两组调制的选定基线进行比较。所有模型都经过 80 个时期的训练,并在测试数据集上进行评估。表 III 分别显示了第 V-A2 节中提到的基线和扩展数据集的所有 SNR 的平均分类准确度,而图 8 显示了基线数据集的 SNR 的准确度。

LSTM [7] 是在基线数据集上评估的 AMC 性能最佳的 SoA 架构,其平均准确率达到 86%,可训练参数超过 200k。BO NSMA 找到了第一个基因优化的架构,它实现了略高的平均精度(提高了 1.43%),同时将可训练参数的数量减少到 69k(可训练参数的数量减少了 2.90 倍)。具有 [7] 中给出的默认训练参数的 LSTM 无法针对扩展数据集收敛。就实现的性能而言,接下来是 ResNet 和 ResNeXt 架构。对于基线数据集,BO-NSMA 分别比 ResNet 和 ResNeXt 获得了 2.02% 和 1.88% 的准确率增益,同时与 ResNet 相比,可训练参数的数量保持在三倍以上。对于扩展数据集,BO NSMA 分别比 ResNet 和 ResNeXt 实现了 3.49% 和 2.80% 的精度增益,同时与 ResNet 相比,可训练参数的数量保持在 4.1 倍以上。尽管

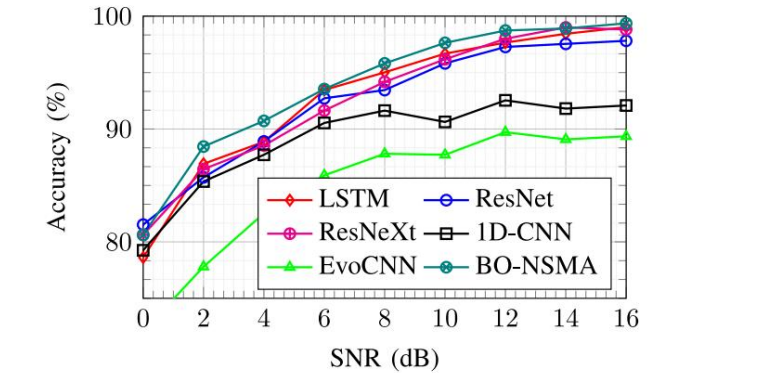


图 8. SNR (基线数据集)的分类精度。

探索 BO-NSMA 在基线数据集的 SNR (图 8)上的性能,请注意,它在 SNR 时的分类精度提高了 3%,这说明与其他基线相比,它对噪声具有鲁棒性。BO-NSMA 还可以通过层堆叠成功优化 CNN,与基线数据集的 1D-CNN [8] 和扩展数据集,分别。类似地,BO-NSMA 发现了一个更好的 ResNet 模块架构,其分类准确率比基线数据集和扩展数据集的 ResNet 分别低 4.0 倍和 6.9 倍,分类准确率低 0.5% 和高 2.4%。具有层堆叠的 BO-NSMA 实现了在 10 代和人口规模为 6 的情况下找到解决方案,与基线数据集和扩展的对应 EvoCNN 相比,可训练参数数量分别减少了 5.41 倍和 78.7 倍,准确率分别提高了 3.43% 和 18.2% 数据集,分别。

4) 统计显着性测试:表 III 显示了通过多次运行找到的最佳个体的 BO NSMA 和 EvoCNN 性能。然而,遗传算法具有随机性和非确定性;因此,在同一个优化问题上运行一个 GA 两次通常会产生不同的结果。然后应用统计测试来确定是否有足够的经验证据来声明两种算法之间的差异。BO-NSMA 与它的 GA 对应物:EvoCNN 和 NSGA-Net 进行了比较。由于 NSGA-Net 搜索空间有限,不适用于 AMC,不与 BO-NSMA 进行比较。BO-NSMA 与两个数据集的 EvoCNN 进行了比较。使用显着性水平  $\alpha = 0.05$  的 Wilcoxon 符号秩检验 [48] 来量化比较的显着性。

Wilcoxon 符号秩检验是一种非参数统计检验,用于比较两组观察值的差异来自零中值分布。

在 [49] 中,表明参数测试不适合对连续优化问题的进化算法进行统计分析,而非参数测试(例如 Wilcoxon 符号秩检验)是此类分析的好工具。

零假设  $H_0$  定义为“BO-NSMA 和 EvoCNN 之间没有区别”。在本文中,使用最佳发现个体的分类精度进行比较。

表IV  
BO-NSMA和EvoCNN统计比较

Metric	Baseline dataset		Extended dataset	
	BO-NSMA	EvoCNN	BO-NSMA	EvoCNN
Max. Acc. (%)	87.6	80.3	83.7	64.5
Min Acc. (%)	85.2	67.6	80.2	45.3
Mean. Acc. (%)	86.5	76.7	82.2	57.7
Max # $\theta$	99,643	1,679,792	98,020	50,803,314
Min # $\theta$	21,803	193,874	51,252	814,226
Mean # $\theta$	68,405	611,113	81,179	9,517,301
Mean Search Time (h)	5.65	0.57	32.2	1.85

表五  
WILCOXON符号秩检验获得的P值

Model	p-value	
	Baseline dataset	Extended dataset
LSTM [7]	0.004	0.005
ResNet [8]	$2.47 \cdot 10^{-6}$	0.005
ResNeXt [11]	$3.69 \cdot 10^{-6}$	0.0068
1D-CNN [8]	$1.64 \cdot 10^{-6}$	0.005
EvoCNN [15]	$1.73 \cdot 10^{-6}$	0.005

表 IV 给出了 BO NSMA 和 EvoCNN 的最佳发现个体的统计分析。对于基线数据集,BO-NSMA 找到的最佳个体的分类精度介于 85.2% 和 87.6% 之间,而 EvoCNN 找到的最佳个体的分类精度在 [67.6, 80.3]% 的更宽范围内。BO-NSMA 发现的最佳个体的分类精度平均值对于基线数据集低 9.8%,对于扩展数据集低 24.5%。BO-NSMA 发现的最佳个体的可训练参数数量的平均值对于基线数据集低 8.9 倍,对于扩展数据集低 117.23 倍。表 IV 中给出的统计分析清楚地表明,BO-NSMA 在两个数据集的分类精度和网络复杂性方面都优于 EvoCNN。

运行 Wilcoxon 符号秩检验以检查这些结果是否显著。Wilcoxon 符号秩检验也用于人工基线。人为设计的基线是确定性的,因此表 III 中显示的结果可以用作多次运行的常数。获得的每个 BO-NSMA 对应物的 p 值低于显著性水平 ( $\alpha = 0.05$ ) (见表 V)。因此,在高度自信的情况下,可以说 BO-NSMA 优于 EvoCNN 和人工基线。此外,结果具有统计学意义。

六。讨论与研究方向

在本节中,利用实验结果的发现来讨论 BO-NSMA 的时间复杂度和局限性,这可以为 BO-NSMA 的应用提供有价值的见解。

建议的 BO-NSMA 方法。最后,概述了未来可能的研究方向。

A. BO-NSMA 时间复杂度给定算法 3

中的步骤,每个算法的时间复杂度 BO-NSMA 的组成部分如下。  
1)种群初始化:设  $T(x)$  和  $E(x)$  为计算可训练参数个数的时间和评估个体  $x$  分类精度的时间。

种群初始化的最小时间复杂度为  $O(\lambda \times (E(x) + T(x)))$ ,最大时间复杂度为  $O(\lambda \times E(x) + I \times \lambda \times T(x))$ ,其中  $I$  是初始化步骤中允许的最大试验次数。由于  $E(x) \gg T(x)$ ,它是主导项;因此,初始化步骤的时间复杂度为  $O(\lambda \times E(x))$ 。

2)选择:选择步骤是通过二进制巡回命名和PD完成的。需要通过整个种群的两个迭代循环来计算种群中每个个体的优势个体数量。因此,选择步骤的时间复杂度为  $O(\lambda^2)$ 。

3) 后代生成:设  $C(x_1, x_2)$  和  $M(x)$  为两个个体  $x_1, x_2$  进行交叉的时间,个体  $x$  进行变异的时间。

后代生成步骤的时间复杂度为  $\lambda/2 \times (\sum_{i=1}^{px2i-1} cr + \sum_{i=1}^{px2i} cr)/2 \times C(x_2i-1, x_2i) + \sum_{i=1}^{pxi} M(x_i)$ 。通过只关注主导项,后代生成步骤的时间复杂度为  $O(\lambda)$ 。

4)局部搜索:局部搜索适用于父母和后代。LS 的时间复杂度为  $\sum_{i=1}^{pxi} (2E(x_i))$ ,其中  $\lambda + \mu_i = 1$   $l_s$  表示个体  $x_i$  满足 LS 条件的概率。LS 适用于最好和最差的个体,并且很可能对整个种群都适用。因此,最坏情况下 LS 步骤的时间复杂度为  $O(2 \times (\lambda + \mu)) = O(4\lambda)$ 。

5)淘汰:淘汰步骤采用FS。令  $D(x_1, x_2)$  为计算两个个体  $x_1, x_2$  之间距离的时间。(8).计算整个群体  $\lambda + \mu$  的距离所需的时间具有  $O((\lambda + \mu)2 \times D(x_1, x_2))$  的时间复杂度。



使用等式为下一代选择  $\lambda$  个个体。(7)的时间复杂度为  $O(\lambda(\lambda+\mu))$ 。

通过总结所有五个步骤,关注主导项并删除任何常数,J代的总体BO-NSMA时间复杂度为  $O(J(6\lambda+7\lambda^2)) \approx O(J\lambda^2)$ 。EvoCNN的时间复杂度为  $O(J\lambda)$ ,而NSGA-Net的时间复杂度为  $O(J\lambda \log \lambda)$  [50]。为基线数据集运行的BO-NSMA和EvoCNN的平均时间复杂度(以小时为单位)分别为5.65小时和0.57小时。而针对扩展数据集运行的BO-NSMA和EvoCNN的平均时间复杂度(以小时为单位)分别为32.2小时和1.85小时(见表IV)。扩展数据集需要更长的时间来获得种群的适应度。

接下来评估EvoCNN是否可以在与BO-NSMA相同的时间窗口内找到更好的个体。因此,又进行了一项实验,其中EvoCNN有6小时的时间为基线数据集搜索最佳个体。对于该实验, $\lambda$ 设置为36,J设置为20。在6h中,EvoCNN找到了最佳个体,分类准确率为81.89%,可训练参数数量为133,307。EvoCNN略微提高了分类精度,并将网络复杂度降低了其最佳发现个体的4.5倍,以实现更长的搜索周期。综上所述,对于相同的时间复杂度,BO-NSMA在分类精度和网络复杂度方面均优于EvoCNN。BO-NSMA的best-found个体分类准确率提高4.6%,网络复杂度降低50%。

B. BO-NSMA的局限性和应用基于上述分析,确定了BO-NSMA的以下局限性及其潜在应用的见解。

1) 高时间复杂度:表明BO-NSMA找到了一种DNN架构,对于基线数据集和扩展数据集,分别在5.65小时和32.2小时内优于每个人工设计的DNN架构。手动搜索最佳DNN架构可能需要几天或几个月的时间。即使在与BO-NSMA相同的搜索时间窗口内,EvoCNN也无法找到更好的个体。对于计算资源不重要的离线搜索,BO-NSMA是一种有效的NAS工具。对于运行在计算能力较低的边缘设备上的NAS,没有一种GA会是最佳选择。

2)控制参数设置:虽然BO-NSMA具有自适应交叉和变异率,但一些控制参数是手动设置的。FS的控制参数是根据目标最优解的先验知识设置的(分类精度接近100%,可训练参数的数量低于允许的最大可训练参数数量)。可训练参数的最大数量是根据感兴趣任务的专业知识设置的。探索了最著名的用于AMC的人工DNN架构,并将其可训练参数计数用作BO-NSMA的范围。如果BO-NSMA应用于非分类任务,FS的控制参数可能不是最佳的。允许的最大值

可训练参数的数量将需要有所考虑任务的一些先验知识。因此,要应用于非分类任务,必须重新访问那些控制参数值。

C. 未来的研究方向

在本文中,BO-NSMA的性能评估仅针对AMC进行。探索BO-NSMA对其他分类任务(例如计算机视觉中的图像分类)的性能会很有趣。应评估BO-NSMA对其他任务的手动控制参数的敏感性。未来的研究还可以考虑自动学习这些参数,因为它已经完成了交叉和突变率。LS和FS的计算成本很高。

拥有明智的政策来在几代人之间打开/关闭它们将导致BO-NSMA的时间复杂度显著降低。这些政策应该在时间复杂性和收敛性之间取得良好的权衡。尽管可训练参数的数量是基于CNN架构的网络复杂性的一个有价值的指标,但未来的研究将受益于更精细的网络复杂性指标,例如推理时间。

七。结论

尽管DNN为AMC取得了显著成果,但由于巨大的搜索空间,手动优化其架构具有挑战性。此外,当输入特征发生变化时,给定的优化架构通常无法正确迁移,从而引发重复且乏味的优化。

使用遗传算法(GA)的自动网络架构搜索(NAS)在计算机视觉领域受到了相当大的关注。然而,将这些方法平滑地转移到时间序列问题(例如调制识别)会导致性能欠佳,如NSGA-Net所示。

因此,本文提出了BO-NSMA,一种用于基于DNN的调制识别应用程序的联合架构和网络复杂性优化的新型双目标模因算法。将局部搜索(LS)添加到GA中以加快收敛速度并提高性能。

经过广泛的实验,表明BO-NSMA找到了一个非常接近由性能和复杂性定义的帕累托最优前沿的多样化群体。BO-NSMA发现的架构优于所有人工制作的DNN。此外,证明BO-NSMA不存在人口规模小的早熟收敛问题,与其对应的EvoCNN一样。

使用Wilcoxon符号秩检验证明了所得结果的显著性。

参考

[1] F. Hameed, O. A. Dobre 和 D. C. Popescu, “基于可能性的调制分类方法”, IEEE Trans. 无线通信, 卷。8, 没有。12, 第5884–5892页, 2009年12月。

[2] J. L. Xu, W. Su 和 M. Zhou, “自动调制分类的似然比方法”, IEEE Trans. 系统, 人, 赛伯恩。C, 申请。教师, 卷。41, 没有。4, 第455–469页, 2011年7月。

[3] S. Majhi,R. Gupta,W. Xiang and S. Glisic, “线性调制信号的分层假设和基于特征的盲调制分类”, IEEE Trans.呃.技术,卷. 66,没有. 12,第 11057–11069 页,2017 年 12 月。

[4] F. Yang,B. Hao,L. Yang and Q. Han, “一种基于高阶累积量和 SVM 的高精度信号识别方法”, Proc. 5 诠释.会议.系统.通知. (ICSAI), 2018 年,第 455–459 页。

[5] W. Xiong,P. Bogdanov and M. Zheleva, “基于局部顺序 IQ 特征的稳健且高效的调制识别”, Proc. IEEE INFOCOM 会议.电脑.共同体, 2019 年,第 1612–1620 页。

[6] X. Zhang,J. Sun and X. Zhang, “基于新颖特征提取算法的自动调制分类”, IEEE Access,卷. 8,第 16362–16371 页,2020 年。

[7] S. Rajendran,W. Meert,D. Giustiniano,V. Lenders and S. Pollin, “使用分布式低成本频谱传感器进行无线信号分类的深度学习方法”, IEEE Trans.科恩.公社.网络,卷. 4,没有. 3,第 433–445 页,2018 年 9 月。

[8] T. O Shea,T. Roy and TC Clancy, “基于无线深度学习的无线电信号分类” IEEE J. Sel.主题信号处理,卷. 12,没有. 1,第 168–179 页,2018 年 2 月。

[9] TJ O Shea and J. Corgan, “卷积无线电调制识别网络”, Proc.诠释.会议.工程.申请神经网络, 2016 年,第 213–226 页。

[10] Y. Wang,M. Liu,J. Yang and G. Gui, “用于认知无线电中自动调制识别的数据驱动深度学习”, IEEE Trans.呃.技术,卷. 68,没有. 4,第 4074–4077 页,2019 年 4 月。

[11] E. Perenda.S. Rajendran,G. Bovet,S. Pollin and M. Zheleva, “学习未知:提高未知场景中的调制分类性能”, Proc. IEEE INFOCOM 会议.电脑.共同体, 2021 年,第 1–10 页。

[12] K. He,X. Zhang,S. Ren and J. Sun, “用于图像识别的深度残差学习”, Proc. IEEE 会议.电脑.可见.模式识别. (CVPR), 2016 年,第 770–778 页。

[13] S. Xie,R. Girshick,P. Dollár,Z. Tu and K. He, “深度神经网络的聚合残差变换”,载于 Proc. IEEE 会议.电脑.可见.模式识别. (CVPR), 2017 年,第 5987–5995 页。

[14] Z. Zhong,J. Yan,W. Wu,J. Shao and C. Liu, “实用的块式神经网络架构生成”, Proc. IEEE/CVF 会议.电脑.可见.模式识别, 2018 年,第 2423–2432 页。

[15] Y. Sun,B. Xue,M. Zhang and GG Yen, “用于图像分类的进化深度卷积神经网络”, IEEE Trans.进化.计算机,卷. 24,没有. 2,第 394–407 页,2020 年 4 月。

[16] Z. Lu等人, “用于图像分类的深度卷积神经网络的多目标进化设计”, IEEE Trans.进化.计算机,卷. 25,没有. 2,第 277–291 页,2021 年 4 月。

[17] A. Shrestha and A. Mahmood, “使用增强型遗传算法优化深度神经网络架构”, Proc. 第 18 届 IEEE Int.会议.马赫.学习.申请(ICMLA), 2019 年,第 1365–1370 页。

[18] AE Eiben and JE Smith,进化计算简介, 第二版.德国海德堡:施普林格出版社,2015 年。

[19] B. Wu等人, “FBNet:通过微分神经架构搜索进行硬件感知的高效卷积网络设计”, Proc. IEEE 会议.电脑.可见.模式识别, 2019 年,第 10726–10734 页。

[20] P. Vamplew,R. Dazeley and C. Foale, “多目标强化学习的 Softmax 探索策略”,神经计算,卷. 263,第 74–86 页,2017 年 11 月。

[21] P. Vamplew,J. Yearwood,R. Dazeley and A. Berry, “关于帕累托前沿多目标强化学习标化的局限性”,人工智能进展, W. Wobcke and M.张,主编.新西兰奥克兰:Springer, 2008 年,第 372–378 页。

[22] THF de Oliveira,LP de Souza Medeiros.A. Neto and JD Melo, “Q 管理:一种用于多目标强化学习的新算法”, Expert Syst.申请,卷. 168,2021 年 4 月,艺术.不. 114228。

[23] S. Wei,S. Zou,F. Liao,W. Lang and W. Wu, “使用神经架构搜索进行自动调制识别”, Proc.诠释.会议.高性能.大数据智能系统. (HPBD IS), 2019 年,第 151–156 页。

[24] P. Molchanov,S. Tyree,T. Karras,T. Aila and J. Kautz, “修剪卷积神经网络以实现资源高效迁移学习”, 2016 年, arxiv:1611.06440。

[25] C. Li等人, “基于知识蒸馏的块式监督神经架构搜索”, Proc. IEEE/CVF 会议.电脑.可见.模式识别,美国华盛顿州西雅图,2020 年 6 月,1986–1995 页。[在线的]。可用:https://doi.org/10.1109/CVPR42600.2020.00206 [26] V. Nekrasov,H. Chen.C. Shen and I. Reid, “通过辅助单元对紧凑语义分割模型进行快速神经架构搜索”,在过程中. IEEE/CVF 会议.电脑.可见.模式识别. (CVPR),第 9118–9127 页,2019 年。

[27] T. Elskén,JH Metzen and F. Hutter, “神经架构搜索:一项调查”, J. Mach.学习.水库,卷. 20,没有. 1,第 1–21 页,2019 年 1 月。

[28] A. Bakhshi,N. Noman.Z. Chen,M. Zamani and S. Chalup, “使用遗传算法快速自动优化 CNN 体系结构以进行图像分类”, Proc. IEEE 国会进化.电脑. (CEC), 2019 年,第 1283–1290 页。

[29] WM Jenkins, “通过突变进行神经网络权重训练”, Comput. 结构,卷. 84号31–32,第 2107–2112 页,2006 年。

[30] A. Nadi,SS Tayarani-Bathaie and R. Safabakhsh, “使用基于突变的遗传算法进行神经网络架构和权重的演化”, Proc.第 14 国际CSI 计算机.会议, 2009 年,第 536–540 页。

[31] X. Glorot and Y. Bengio, “了解训练深度前馈神经网络的难度”, Proc.第 13 国际会议.神器.智能.统计,卷. 9,2010 年 5 月,第 249–256 页。[在线的]。可用:http://proceedings.mlr.press/v9/glorot10a.html

[32] MAJ Idrissi,H. Ramchoun.Y. Ghanou and M. Ettaouil, “神经网络架构优化的遗传算法”, Proc.第三诠释.会议.后勤人员.歌剧.管理. (GOL), 2016 年,第 1–4 页。

[33] R. Akut and S. Kulkarni, “NeuroEvolution:使用遗传算法优化深度学习模型的设计。”在过程中. IEEE 诠释.会议.电工.电脑.公社.技术. (ICECCT), 2019 年,第 1–6 页。

[34] RA Viswambaran.G. Chen.B. Xue and M. Nekooei, “使用新的可变量度遗传算法进化深度循环神经网络”, Proc. IEEE 国会进化.电脑. (CEC), 2020 年,第 1–8 页。

[35] N. Ahmadi and R. Berangi, “使用遗传算法和层次聚类从其星座中对 QAM 和 PSK 进行调制分类”, Proc.第三诠释.会议.信息.公社.技术.理论应用, 2008 年,第 1–5 页。

[36] MW Aslam,Z. Zhu and AK Nandi, “结合遗传编程和 KNN 进行自动调制分类”, IEEE Trans.无线通信,卷. 11,没有. 8,第 2742–2750 页,2012 年 8 月。

[37] S. Huang,Y. Jiang,X. Qin,Y. Gao,Z. Feng and P. Zhang, “使用具有结构风险最小化原理的多基因遗传编程对重叠源进行自动调制分类”, IEEE Access ,卷. 6,第 48827–48839 页,2018 年。

[38] R. Dai,Y. Gao,S. Huang,F. Ning and Z. Feng, “基于多目标遗传编程的自动调制分类”, Proc. IEEE 无线通信.网络.会议. (WCNC), 2019 年,第 1–6 页。

[39] DP Kingma and JL Ba, “Adam:一种随机优化方法”, Proc.诠释.会议.学习.代表. (ICLR),第一卷. 1, 2015, 第 1–15 页。

[40] JH Holland, “遗传算法和试验的最优分配”, SIAM J. Comput.,第一卷. 2,没有. 2,第 88–105 页,1973 年。

[41] RS Sutton and AG Barto,强化学习简介,第 1 版.美国马萨诸塞州剑桥市:麻省理工学院出版社,1998 年。

[42] T. Bäck and M. Schütz, “规范遗传算法中的智能突变率控制”,智能系统基础 (计算机科学讲义)。德国柏林:1996 年,第 158–167 页。

[43] **B. Sareni and L. Krahenbuhl, “重新审视适应度共享和壁龛方法”, IEEE Trans.进化.计算机,卷. 2,没有. 3,第 97–106 页,1998 年 9 月。**

[44] SW Mahfoud, “遗传算法的生态位选择方法”,博士.学位论文,哲学博士.科学,大学.伊利诺伊厄巴纳香槟,香槟,伊利诺伊州,美国,1996 年。

[45] O. Schutze,X. Esquivel,A. Lara and CAC Coello, “使用平均 hausdorff 距离作为进化多目标优化中的性能度量”, IEEE Trans.进化.计算机,卷. 16,没有. 4,第 504–522 页,2012 年 8 月。

[46] G. Van Rossum and FL Drake, Python 教程.荷兰阿姆斯特丹:Centrum Voor Wiskunde Informatica,1995 年。

[47] M. Abadi and A. Agarwal. TensorFlow:异构系统上的大规模机器学习. 2015. [在线]。可用:https://www.tensorflow.org/ [48] F. Wilcoxon, “通过排名方法进行的个体比较”, 生物识别.公牛,卷. 1,没有. 6,第 80–83 页,1945 年。[在线的]。可用:http://www.jstor.org/stable/3001968

[49] S. García,D. Molina,M. Lozano and F. Herrera, “使用非参数检验分析进化算法行为的研究:CEC-2005 特别会议的案例研究关于实际参数优化,” J. Heurist,卷. 15,没有. 6,第 617–644 页,2009 年。

[50] MT Jensen, “降低多目标 EA 的运行时复杂性:NSGA-II 和其他算法”, IEEE Trans.进化.计算机,卷. 7,没有. 5,第 503–515 页,2003 年 10 月。



Erma Perenda于 2013 年获得萨拉热窝大学电气工程学院电信硕士学位。她目前正在攻读博士学位。比利时鲁汶大学电气工程系学士学位。在加入 KU Leuven 之前,她曾在 Nokia Shanghai Bell 的 Nokia Digital Home 产品组担任高级 Wi-Fi 软件工程师。她的研究兴趣包括频谱测量和管理、无线网络优化和架构设计。



Sofie Pollin (高级会员,IEEE)是 KU Leuven 的教授,专注于无线通信系统。在此之前,她曾在 imec 和加州大学伯克利分校工作。她与 KU Leuven 一起为分布式 Massive MIMO 构建了一个 5G 测试台,参与了频谱感知和开放数据计划,目前正在多个大型欧盟项目中引领 6G 测试。她的研究围绕无线网络展开,这些网络需要更加密集、异构、电池供电和频谱受限的网络。



Sreeraj Rajendran获得博士学位。KU Leuven 电气工程学位。他是 Sirris EluciDATA 实验室的高级数据科学家。他参与了支持基于机器学习的解决方案的工业研发项目,以满足比利时工业的需求。在加入 Sirris 之前,他是 KU Leuven 的博士后研究员。作为电信部门的数字信号处理工程师,他还拥有几年的行业经验。



Gerome Bovet获得博士学位。2015 年获得法国巴黎高科电信大学网络和计算机系统学士学位,2021 年获得瑞士弗里堡大学行政工商管理硕士学位。他是瑞士国防部数据科学负责人,领导一项研究团队和大约 30 个项目的投资组合。

他的工作重点是应用于网络防御用例的机器和深度学习方法,重点是异常检测、对抗和协作学习。



Mariya Zheleva (IEEE 会员)获得工程学士学位。和工程硕士索非亚技术大学的电信学位,以及硕士和博士学位。加州大学圣巴巴拉分校计算机科学学位。她是纽约州立大学奥尔巴尼分校计算机科学系的助理教授。

她从事小型本地蜂窝网络、数据驱动的动态频谱访问、频谱管理和传感以及网络性能和表征方面的工作。她是奥尔巴尼大学 UbiNet 实验室的创始人和主任。她的研究由美国国家科学基金会和微软资助。她的研究是在无线网络和信息通信技术的交叉领域进行发展。