引用格式:王淳,赵艳明,冯燕.基于ConvMixer架构的高效点云分类方法[J].中国传媒大学学报(自然科学版),2024,31(01):56-64.
文章编号:1673-4793(2024)01-0056-09

基于ConvMixer架构的高效点云分类方法

王淳,赵艳明*,冯燕
(中国传媒大学信息与通信工程学院,北京 100024)

摘要:近年来,视觉Transformer模型在点云分类等三维计算机视觉任务中显现出潜在的优越性,但其有效性来源仍然模糊不清。研究它们在视觉任务中的性能是完全归功于Transformer结构本身的优越性,还是至少部分得益于使用局部块作为输入表示,是非常必要的。受此启发,本文提出了一种简单但仍然有效的点云分类和分割模型PointConvMixer,用ConvMixer架构取代了Point-BERT中的标准Transformer。PointConvMixer在ModelNet40数据集上的整体分类准确率达到92.3%,在ShapeNet Parts数据集上进行点云部分分割时mIOUI和mIOUC分别为85.4%和83.9%,均优于基于Transformer的对比模型。此外,本文还进一步提出PPFConvMixer,其利用高效的局部特征描述符PPF增强了PointConvMixer,从而优化了点云分类性能。在查询半径为0.25m时,PPFConvMixer的总体分类准确率达到了93.8%。

关键词:三维点云分类;深度学习;ConvMixer;Point Pair Feature
中图分类号:TP391 **文献标识码**:A

An efficient point cloud classification method based on ConvMixer architecture

WANG Chun, ZHAO Yanming*, FENG Yan
(School of Information and Communication Engineering, Communication University of China, Beijing 100024, China)

Abstract: In recent years, Vision Transformers (ViTs) show potential superiority on 3D computer vision tasks, including point cloud classification, but the provenance of their effectiveness remains ambiguous. It is highly essential to investigate whether their performance in vision tasks is entirely due to the superiority of the structure itself, or at least partially benefits from the use of local patches as input representations. Motivated by this, in this paper PointConvMixer was proposed, a simple but still effective point cloud classification and segmentation model, replacing the standard Transformer in Point-BERT with the ConvMixer architecture. The overall classification accuracy of PointConvMixer on the ModelNet40 dataset reaches 92.3%, and the mIOUI and mIOUC for point cloud segmentation on the ShapeNet Parts dataset are 85.4% and 83.9% respectively, both of which outperform the compared Transformer-Based networks. In addition, PPFConvMixer was further introduced, which augmented PointConvMixer with an efficient local feature descriptor Point Pair Feature (PPF) to optimize the point cloud classification performance. Our method has shown promising results for point cloud analysis despite its simplicity.The overall classification accuracy of PPFConvMixer achieves 93.8% at a query radius of 0.25m.

Keywords: 3D point cloud classification; deep learning; ConvMixer; Point Pair Feature

基金项目:国家重点研发计划(2018YFB1404103)

作者简介(*为通讯作者):王淳(2000-),女,硕士研究生,主要从事点云处理、三维场景重建等研究。Email:chun1206@cuc.edu.cn;赵艳明(1973-),女,博士,副教授,主要从事空间投影校正、点云处理、三维场景重建等研究。Email:yanmingzhao@cuc.edu.cn

1 引言

近年来，二维传统视觉任务随着深度学习技术的飞速发展日益成熟。而随着三维扫描技术的发展和三维视觉算法的广泛应用，三维识别任务也在自动驾驶[1]、机器人[2]、增强现实[3]等领域越来越受到关注。点云分类任务作为目标识别、三维重建等任务的前提，是三维识别领域的一大研究热点。点云作为一种常用的三维数据，具有非常强的空间表达能力，能够在保留三维空间位置坐标的同时，附加上可选的其他信息，如颜色、法向量和反射强度信息等。然而，由于点云的稀疏、不规则和无序结构等特性，有效设计局部几何关系提取器和网络架构来完成对点云数据的特征学习仍然是一项具有挑战性的任务。

为了应对这一挑战，以往的点云分析方法可以大致分为两类。第一类是基于投影的方法[4-6]，有时也称为基于结构化点云的方法。此类方法将点云变换成规则化的、可以使用卷积神经网络直接处理的形式。按照将点云转换成规则化数据所采用策略的不同，可进一步分为基于多视图和基于体素的方法。尽管结构化表示法在一定程度上解决了不规则和无序的问题，但缺陷依然存在。基于多视图的方法不是真正的三维表示，而体素化严重影响内存和计算成本，且两者都可能丢失重要的几何信息。

第二类是基于原始点云的方法[7-9]，其直接对原始点云数据进行处理，最大限度地保留了点云信息的完整性，是现如今基于深度学习的点云处理的主要研究趋势。其中，斯坦福大学Charles等人[7]提出的PointNet开创性地将多层感知器（Multilayer Perceptron，MLP）[10]与全局聚合相结合，对每个点进行编码。其一系列后续研究[11-14]表明，高效的局部特征描述符可以大大提高点云分类的性能。

然而，先前研究的问题在于，虽然提供了高效的局部几何关系提取器，但复杂的网络设计阻碍了其应用效率。针对这一点，关于图像分类的文献[15]引起了本文作者的注意，它提出了ConvMixer这一极其简单的架构，证明了使用局部块（Patch）作为输入表示可能是实现卓越性能的关键。受此启发，本文通过其架构设计了一个简单但仍然有效的点云处理网络PointConvMixer。此外，本文还发现，点对特征（Point Pair Feature，PPF）[16]作为一种快速的局部特征编码方法，可以通过三维局部块的形式有效地探索局部几何信息，并在实验中得到了证实。

本文提出了一种新颖的点对特征卷积网络PPFConvMixer，用于基于三维点云的物体分类和部件分割。PPFConvMixer包含了三维Patch嵌入策略和改进的ConvMixer架构，并将PPF描述子纳为局部点云特征编码方法。首先，根据每个局部区域的参考点及其邻近点计算PPF。最终的局部几何图形将由一组增强的几何关系来表示：点、法向量和PPF。与PointNet++[8]类似，PPFs将中心点的成对特征聚合到其他点。不过，PPFConvMixer使用反对称四维描述符来表示一对定向三维点的表面，从而更好地描述了局部区域，且不会重复组合附近的Patch嵌入，这使得可以围绕Patch本身的有效性进行研究。然后，应用三维Patch嵌入处理来保持局部性，将小区域的点组合成单一的输入特征。在ConvMixer[15]的启发下，最终利用标准卷积构成的各向同性架构（Isotropic Architecture），分别实现了空间维度和通道维度的混合，同时在整个网络中保持了相同的大小和分辨率。本文在两个具有挑战性的基准上进行了广泛实验。该卷积网络设计在实现上非常简单，但能够在形状分类和部件分割任务中产生具有竞争力的准确性。

本文的主要贡献有三个方面：

（1）提出了PointConvMixer网络，该网络在ModelNet40和ShapeNet数据集上的物体分类和形状分割任务中获得了极高的准确率，证明了其有效性和通用性。

（2）创新地在点云分类任务中使用PPFs来有效地描述局部几何信息，采用PointConvMixer的优化网络PPFConvMixer的点云分类准确率高达93.8%。

（3）通过实验证明，和注意力机制及Transformer架构的开发相比，Token化的输入设计在点云学习中同样值得关注。对于点云分析来说，“Patches are all you need”这一结论仍然有效。

2 相关研究工作

2.1 基于多视图和基于体素的方法

基于多视图的方法和基于体素的方法是基于投影的方法的两个分支。由于点云的不规则性，早期的研究[17-20]将点云投影到多视图图像以对点云数据进行卷积。虽然基于视图的三维表示方法可以实现良好的性能，但它需要花费大量时间并需要更多内存进行渲染，这使得该方法无法应对实时应用。获取规则化点云数据的另一种直接方法是将点转换为空间体素，这可以归纳为基于体素的方法。对于体素模型，内存消耗的限制决定了输入3D网格的分辨率较低，导致点云结构信息丢失，随后的研究一直在努力克服这一

缺陷。例如,OctNet[11]和Kd-Net[12]分别利用八叉树结构和KD树结构来替换固定大小的体素网格,以减轻分析难度。但由于表示质量在很大程度上依赖于高分辨率网格,使用体素模型仍然效率不高。与上述两种方法不同,本文直接从原始点云中提取特征。

2.2 基于原始点云的深度学习方法

逐点(Point-Wise)网络直接处理原始点集以提取特征。该领域的先驱性网络PointNet[7]利用MLP对每个点进行单独编码,然后通过全局池化整合提取的点特征。然而,这种网络设计忽略了对于点云识别任务至关重要的局部细节。为此,PointNet++[8]通过局部特征聚合与多层次特征提取结构改进了PointNet。然而,由于其局部特征的聚合仅通过最大池化实现,PointNet++网络并未充分利用区域信息。为充分挖掘局部结构信息,DGCNN[13]设计了EdgeConv模块来生成边缘特征,将与同一指定局部区域内中心点及其邻近点的特征差值相连接,然后按MLP编码方法和最大池化聚合操作进行处理。为了整合区域信息,PointWeb[14]通过连接和探索区域内的所有点对,来穷举上下文信息。虽然其获得了更具代表性的区域特征,但需要更多的时间成本和计算资源。

最近的一些研究转为聚焦于点卷积核的设计。PointCNN[21]通过-Conv算子对输入点和特征进行置换和加权,将邻近点转换为规范顺序。PAConv[22]通过动态组合存储在权重库中的基本权重矩阵来构造卷积核,并且可以作为即插即用的卷积操作使用。

本文的研究重点之一是通过学习反对称四维描述符(包括点对之间的交角和距离参数)来捕捉点的局部空间布局。相对而言,本文网络占用的时间和计算资源更少,且能很好地保留点之间的关系。

2.3 各向同性架构

与呈金字塔形的主流CNN模型架构不同,各向同性架构(也称同质架构)由串联的重复块(Block)组成。这种新的架构范式受到视觉Transformer的启发,其特点是各个块的大小和形状相同,并在第一层使用Patch嵌入。在图像分类任务中,很多研究尝试对一或两个重复块进行各种新颖的操作以获得良好的性能,例如MLP-Mixer[23]、ResMLP[24]、gMLP[25]等。但这也带来了一个问题:它们的良好性能是通过应用新的操作实现的,还是通过使用Patch嵌入和由此产生的同质结构实现的?

一些学者还尝试采用注意力机制和Transformer架构来进行点云处理。PointASNL[26]提出了一种自注意力机制来更新局部点簇的特征,以应对点云处理中的噪声。Point Transformer[27]为点云设计了自注意力层,并使用它们构建用于点云识别任务的自注意力网络。Point Cloud Transformer(PCT)[28]创建了一个由增强的输入嵌入和简单Transformer组成的点云处理架构,以进行特征学习。尽管这些方法相当强大,但它们的有效性来源依然并不明确。

本文的研究重点之一为:明确这些基于Transformer的点云处理方法的有效性是源于Transfromer编码器Block的使用,还是源于Patch嵌入的使用和由此产生的同质结构。为了排除前者对网络性能带来的影响,并证明Patch嵌入和由此产生的同质结构的组合足够有效,本文使用的网络与PCT类似,直接对Patch进行操作,在所有层中保持等分辨率和大小的表示,并将信息的"通道混合"与"空间混合"分开。不同的是,本文只使用标准卷积来制定所提出的架构,并获得了更好的性能。

3 基于ConvMixer的点云分类方法

本节首先分析局部几何提取器的一般用法,并回顾PointNet++[8]、PointWeb[14]和RS-CNN[9]中的相关操作。然后,介绍用于编码局部Patch的PPF模块,以及用于点云分类中间特征处理的ConvMixer层。通过将用于图像的ConvMixer架构移植到点云处理中,获得底层模型PointConvMixer,该模型可用于形状分类或部件分割等不同任务。最终提出了PPFConvMixer网络并详细阐述其用于点云分类的处理链。

3.1 基础方法阐述

以往的研究重视局部特征聚合,因为利用局部特征描述器的目的通常是为了学习局部信息的隐含模式,以获得更好的点云学习结果。给定三维点云$X=\{x_i|i=1,\cdots,N\}\in\mathbb{R}^{3\times N}$,其中$N$表示输入点的数量。一般来说,第一步是选择参考点作为中心点,按照确定性规则形成局部区域。最远点采样和均匀采样是两种常用的方法。然后,通常选K近邻(K-Nearest Neighbor,KNN)算法作为分组算法来计算每个中心点的邻近点,因为其计算效率较高。

将一个输入点表示为$x_i\in\mathbb{R}^{3\times N}$,其邻近点表示为$x_{ij}$,其卷积层中的输入特征表示为$f_i\in\mathbb{R}^{c_{in}\times N}$、其输出

特征图为$g_i \in \mathbb{R}^{c_{out} \times N}$，其中$c_{in}$和$c_{out}$表示输入和输出的通道维度。局部特征聚合过程可表述为式(1)：

$$g_i = A(M(f_{ij}) | j = 1,\cdots,k) \tag{1}$$

其中，$A(\cdot)$表示聚合函数，$M(\cdot)$表示局部特征提取的映射函数，f_{ij}是代表第i采样点的第j个邻近点特征的关系编码函数。参数k是每个局部Patch包含的点数。对PointNet++[8]来说，$A(\cdot)$是最大池化操作，$M(\cdot)$是共享的MLP网络，f_{ij}实际上是第一层中受分组方法影响的x_{ij}。此外，它还堆叠了多个学习阶段来学习分层特征，并在每个阶段通过最远点采样对点进行重新采样。通过这种方式，该方法可以逐步扩大感受野。基于这种处理流程，PointWeb[14]提出了一个即插即用的自适应特征调整(AFA)模块，用于学习每个点对其他点的影响，并将点密集连接成局部点网。其关系编码函数替换为式(2)：

$$f_i' = f_i + \sum_{j=1, j \neq i}^{N} \mathrm{MLP}(f_j - f_i) \cdot (f_j - f_i) \tag{2}$$

RS-CNN[9]是另一种强调局部特征提取的网络，其通过关系学习提取出两点之间的关系表达式，然后利用关系表达式更新参考点的特征。与PointWeb不同，RS-CNN通过式(3)深入挖掘局部几何信息：

$$M(f_{ij}) = \mathrm{MLP}\left(\left[\left\| x_{ij} - x_i \right\|_2, x_{ij} - x_i, x_{ij}, x_i\right]\right) * f_{ij}, \quad \forall j \in \{1,\cdots,N\} \tag{3}$$

与PointWeb和RS-CNN类似，大量方法侧重设计精细的局部特征提取器，利用详细的局部几何信息，获得了令人满意的性能。尽管如此，一个问题依然存在：计算复杂度非常高。对于基于PointNet++架构的PointWeb，其分层网络由多个集合抽取层组成，这意味着采样、分组和局部特征聚合的过程需要多次执行。且PointWeb的关键在于利用所有点对之间的上下文信息，对它们进行混合计算既复杂又耗时。此外，重复组合附近的Patch嵌入会混淆Patch嵌入策略的效果和类似归纳偏置的效果。

为了聚焦于Patch的使用，同时丰富局部点特征并控制计算成本，本文尝试最大化使用输入特征来表示局部邻近区域。因此，选择PPFs作为局部Patch的编码方法。

3.2 PPFConvMixer网络架构

(1)点云Patch的划分

受视觉Transformer中patch嵌入策略的启发，Point-BERT[29]将点云转换为由局部点云簇组成的集合。为了将ConvMixer图像处理模型应用于点云数据，本文采用类似的预处理方法。具体地，在给定点云数据的整体集合后，首先使用最远点采样(FPS)方法选择g个局部点云簇的簇中心。接下来，以固定查询半径r为条件，在选定的g个局部中心点周围选择k个最近邻点，构成包含细节局部几何信息和结构的g个局部点云簇。然后，将近邻点的坐标都减去中心点坐标，通过局部区域归一化来排除点云真实坐标带来的影响。这样，就能在三维点云中获得与二维图像Patch对应概念的局部Patch。

(2)点云局部Patch的编码

如前所述，最终的局部几何将由一组增强的几何关系来表示：点的三维坐标、法向量、点对特征PPF。这三者构成的集合共同作为网络的输入。具体地，如图1所示，给定一个参考点为x_r，其表面法向量为n_r，局部区域中的k个相邻点为$x_i, i = 1,\cdots,k$。可以将局部几何特征表示为由参考点和KNN算法决定的一个局部Patch$\{x_r \cup \{x_i\}\}$，局部几何特征具体的计算公式如式(4)：

$$f_r = \{x_0,\cdots,x_k,n_0,\cdots,n_k,\psi_{r0},\cdots,\psi_{rk}\} \tag{4}$$

其中，ψ_{ri}表示三维点对间关系的非对称四维描述子，其具体计算方式为式(5)所描述：

$$\psi_{ri} = \left(\| d \|_2, \angle(n_r,d), \angle(n_i,d), \angle(n_r,n_i)\right) \tag{5}$$

其中，d代表点间的距离向量，$\|\cdot\|$代表欧式距离，$\angle$代表角度计算子。$\angle(n_r,n_i)$的计算如式(6)所示，注意$\angle(n_r,n_i)$的范围在$[0,\pi]$：

$$\angle(n_r,n_i) = \mathrm{atan}\,2\left(\| n_r \times n_i \|, n_r \cdot n_i\right) \tag{6}$$

本文将PPF描述子ψ_{ri}作为输入表示主要有两个原因：一方面，相对于PointWeb等重复融合周围的Patch嵌入层的网络而言，由于采用的是一次性配对，PPF对于输入特征的计算更加简单方便；另一方面，它的计算复杂度也更低。

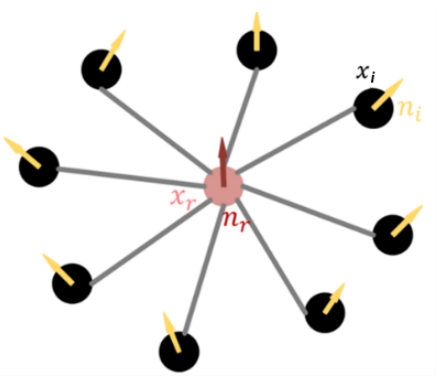

图1 局部点对特征PPF示意图

(3)PointConvMixer网络架构

用于图像分类任务的ConvMixer模型的操作可以简单概括如下:第一,首先设定Patch大小,对图像分Patch作为输入表示。将图像Patch输入Patch嵌入模块进行空间维度上的降维以及通道维度上的升维,然后经过一个激活函数和归一化层。Patch嵌入模块其实就是一个核大小和步长都等于设定的Patch大小的卷积。第二,将经过激活和归一化的Patch特征输入如图2所示结构组成的ConvMixer模块中。ConvMixer模块由一个深度卷积模块和一个逐点卷积模块组合而成。深度卷积即组数等于通道数的分组卷积,逐点卷积本质上是1×1的卷积,每个卷积后面是激活函数和批归一化,深度卷积结构的上方还包含一个残差连接。深度卷积用来混合空间维度信息,逐点卷积则用来混合通道维度上的信息,这使得特征在空间域和特征域不断混合。

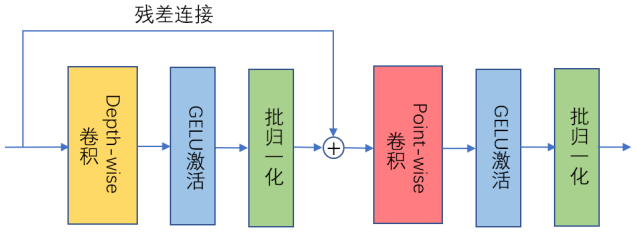

图2 ConMixer层的实现

ConvMixer层起到了分离空间和通道维度的混合的作用,使用的ConvMixer层数可通过参数*depth*来进行调控。但由于图像是二维数据而点云是三维数据,因此,在将用于图像分类任务的ConvMixer网络模型迁移到点云分类时,输入的数据需要进行不同的处理,同时需要对原本的网络结构进行进一步的优化以达到更好的效果。具体地说,点云需要选定性能较优的分块处理方法,对应的块嵌入层需要重新的设计来适应不同维度的数据输入,同时网络需要添加优化方法来优化得到的最终分类效果。

(4)PPFConMixer点云分类流程

在PointConvMixer的基础上,本文通过将局部点云Patch编码成PPF,得到了一个简单但非常高效的点云处理网络PPFConvMixer。该网络由输入数据编码、三维Patch嵌入层和多层改进的ConvMixer层,以及最后用于分类的MLP层组合而成。除去分类MLP层,PPFConvMixer可表述为式(7):

$$g = \Phi\left(A\left(M\left(f_r\right)|r = 1,\cdots,N\right)\right) \tag{7}$$

其中,$\Phi(\cdot)$表示由重复的全卷积模块组成的各向同性的网络,即多次重复堆叠的改进的ConvMixer层,每层实现流程如图3所示。重复次数是一个超参数d,即前文所述的深度参数*depth*。$A(\cdot)$和$M(\cdot)$的组合是Patch嵌入层的具体实现,$M(\cdot)$通过共享MLP来提取局部特征,$A(\cdot)$代表聚合操作,将每个局部区域的点聚集成单个的输入特征,实际操作当中采用最大值池化来实现。f_r表示采用PPF局部编码方式构建的局部Patch,N表示局部Patch的个数,其大小取决于最开始的采样点数。最终的PPFConMixer点云分类模型整体处理流程如图3所示。

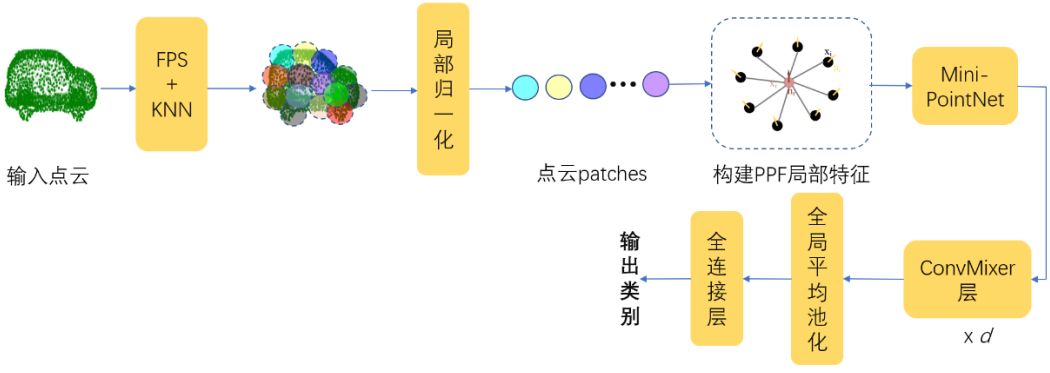

图3 PPFConvMixer点云分类架构流程图

4 实验结果及分析

本节将在多个基准上对本文提出的模型进行全面评估,并展示其在物体分类和形状部分分割任务中的实验结果。

4.1 形状分类

(1)数据集

首先在ModelNet40[30]数据集上对所提出的网络模型PPFConvMixer进行了评估。ModelNet40是最常用的

点云形状分类数据集，包括12311个CAD模型，这些点云模型被分为40个类别。实验中使用和Point2Sequence[31]一样的策略，将ModelNet40数据集分成9843个训练样本和2468个测试样本。在训练时，均匀采样1024个点作为输入。

（2）实现细节

数据预处理时，选择半径为0.25m的局部邻域，并在该邻域内均匀采样64个点。定义的邻域中可能出现点数不足64的Patch，对此随机重复一些点以确保Patch大小一致。使用AdamW优化器对模型进行75个epoch的训练，*batchsize*大小为16。采用预热（Warmup）与余弦退火（Cosine Annealing）相结合的策略控制学习率变化，warmup的*epoch*设置为10。最大学习率为0.0005，最小学习率为1e-6。权重衰减系数为0.001，ConvMixer深度*depth*设为4，特征维度为368。

据SimpleView[32]所述，在不使用任何集成方法的情况下比较模型性能更为准确，因此所有的实验中均不使用voting策略来优化预测。此外，为了增强鲁棒性，采用了两种数据增强策略：在xyz三个方向在[2/3，3/2]的范围内进行同比例的随机缩放；在[-0.2，0.2]范围内随机平移。

（3）实验结果

表1给出了PPFConvMixer网络在ModelNet40数据集上的点云分类任务准确率，并与其他点云分类网络进行了比较。第一条虚线上方的是经典的深度学习点云分类方法，包括PointNet[7]、PointNet++[8]以及RS-CNN[9]等，第三条虚线下即为本文所提出的网络。第一条虚线上方倒数四个网络都是输入数据中不仅包含位置数据，还包含法向量的经典方法，有O-CNN[33]、Spec-GCN[34]以及SO-Net[35]等，PPFConvMixer网络分类准确率都超过了这些模型。第一条与第二条虚线之间是一些基于Transformer的方法，但是其网络架构做了更多特殊设计和归纳偏置。PPFConvMixer即使没有类似的特殊网络设计，分类精度仍然优于PCT[28]和Point Transformer[27]。第二条虚线与第三条虚线间的方法是一些基于标准Transformer模型设计的点云分类网络。可以发现PPFConvMixer比所有标准Transformer模型构建的点云分类网络效果都要更好。

“#params(M)”列中记录了网络参数的数量，可以看到PPFConvMixer的参数量为2.62M，并不是很大。总之，PPFConvMixer的总体性能优于表中的Point-BERT[29]等其他点云分类模型。这些实验结果表明，高效的输入表示对于实现Transformer的卓越性能至关重要，这一结论在点云领域依旧成立。

表1 在ModelNet40公共数据集上的形状分类结果
（nor：法向量，“-”：未知）

网络模型	输入数据格式	输入点数	#params（M）	总体分类精度（%）
PointNet[7]	xyz	1024	3.50	89.2
PointNet++[8]	xyz	1024	1.48	90.7
PointCNN[21]	xyz	1024	0.60	92.2
DGCNN[13]	xyz	1024	1.84	92.9
DensePoint[36]	xyz	1024	0.67	92.8
RS-CNN[9]	xyz	1024	1.41	92.9
PAConv(*DGC)[22]	xyz	1024	-	93.6
O-CNN[33]	xyz, nor	1024	-	90.6
Spec-GCN[34]	xyz, nor	1024	2.05	91.8
SO-Net[35]	xyz, nor	1024	-	92.5
SpiderCNN[37]	xyz, nor	1024	-	92.4
PCT[28]	xyz	1024	2.88	93.2
Point Transfromer[27]	xyz	1024	-	93.7
NPCT[28]	xyz	1024	1.36	91.0
Transformer[29]	xyz	1024	-	91.4
Transformer-CoCo[29]	xyz	1024	-	92.1
Point-BERT[29]	xyz	1024	-	93.2
PPFConvMixer	xyz, nor	1024	2.62	**93.8**

4.2 部件分割

（1）数据集

部件分割是细粒度形状识别的一项挑战性任务。ShapeNet数据集是一个具有丰富标注的大规模点云数据集，广泛应用于计算机视觉和机器人研究。其中，其中，ShapeNet Parts数据集常用来做三维点云的部件分割任务。ShapeNet Parts数据集总共包括16个大的类别，包含16881个点云模型，如飞机、座椅、桌子等。每个大的类别又可以分成若干个小类别，总共可分为50个小类别，如一个桌子的点云模型可以分割成桌面、桌腿等小类别部件。每个点云形状模型可划分为2-5个部件。ShapeNet Parts数据集中划分了13998个训练数据，2874个测试数据。

（2）实现细节

使用AdamW优化器进行300个epoches的训练，*batchsize*大小为64。与形状分类实验设置类似，使用warmup+Cosine Annealing的学习率控制策略，warmup的*epoch*为10，最大学习率为0.0005，最小学习率为1e-6。权重衰减系数、ConvMixer深度*depth*和特征维度分别为0.5、4、368。

（3）实验结果

本小节构建PointConvMixer点云分割网络在ShapeNet Parts数据集上进行了部件分割实验，以类

别平均交并比($mIOU_C$)和实例平均交并比($mIOU_I$)作为评价指标,并与经典的点云分割网络的性能进行了对比,评估结果如表2所示。由表中的实验结果可以看出,PointConvMixer的$mIOU_C$优于经典的PointNet[7]、PointNet++[8]及DGCNN[13],分别高出3.51%、2.05%和1.57%,比标准Transformer、Transformer-OcCo[29]模型高出0.48%。另外,PointConvMixer的$mIOU_I$指标为85.4%,优于经典的Kd-Net[12]、PointNet[7]、PointNet++[8]和DGCNN[13],分别高出3.1%、1.7%、0.3%和0.2%。且PointConvMixer分割模型的$mIOU_I$也优于标准NPCT[28]、Transformer以及添加了OcCo预训练方法的Transformer-OcCo[29]网络,分别高出0.2%、0.3%和0.3%,稍低于Point-BERT[29]的85.6%。

表2 ShapeNet Parts数据集上的部件分割结果("–":未知)

网络模型	$mIOU_I$(%)	$mIOU_C$(%)
Kd-Net[12]	–	82.3
PointNet[7]	80.39	83.7
PointNet++[8]	81.85	85.1
DGCNN[13]	82.33	85.2
Transformer[29]	83.42	85.1
Transformer-CoCo[29]	83.42	85.1
Point-BERT[29]	**84.11**	**85.6**
PointConvMixer	83.90	85.4

值得注意的是,PointConvMixer使用简单的卷积架构ConvMixer取代了Transformer中的关键模块,但在形状分类和部件分割方面的性能都优于原有Transformer。这表明Transformer的卓越性能不仅归功于其架构,而且至少部分归功于对点云数据进行基于Patch的预处理,以降低计算复杂度。

4.3 消融实验

为了验证网络各参数设置的合理性,并验证分析模块的有效性,本文设计了以下不同配置的实验:

(1)尝试了随机缩放+随机平移的数据增强方法,说明了数据增强对性能提升的作用。

(2)对比了将特征表示输入卷积网络前,分别采用最大池化和平均池化的分类效果,验证了最大池化降维的必要性。

(3)讨论了PPF局部特征描述子的使用对结果造成的影响。

(4)比较了在使用K近邻算法聚合k个近邻点并计算点云的PPF特征表示时,不同的k值所造成的性能差异。

(5)比较了不同层数(即不同深度)的ConvMixer层对网络性能的影响。

前三项设置的消融实验结果如表3所示,设定基线为模型A,其分类准确率较低,仅为91.5%。表中DA、Max、Average分别表示数据增强、最大池化和平均池化策略。实验结果表明,使用最大池化方法进行降维时,点云分类的准确率(Acc.)较高,在[2/3,3/2]范围内随机缩放和在[-0.2,0.2]范围内平移的数据增强方法能有效优化性能。此外,PPF描述子的引入还带来了1.9%的精度提升。这说明PPF是一种非常有效的三维局部特征描述子。通过计算局部点对特征,PPF特征表示包含了丰富的三维点几何信息。

表3 PPFConvMixer在ModelNet40基准上的消融研究结果

模型	DA	Max	Average	With PPF	Acc.(%)
A		✓			91.5
B	✓	✓			91.9
C	✓		✓	✓	92.4
D	✓	✓		✓	**93.8**

对K近邻算法中的局部大小k选取不同值时,在ModelNet40数据集上所得总体分类准确率如表4所示。可以看出,当K近邻算法中的k=64时,点云分类精度最高。推测如果k太小,那么在局部邻域中选取的点就会太少,无法获得丰富的邻域信息。如果k过大,但在查询半径0.25m的区域内没有那么多点,那么算法就会随机重复选取区域内的点,以保证邻域内的点数满足要求。最后导致邻域中重复出现无效信息,也降低了性能。

使用不同深度ConvMixer层对ModelNet40数据集进行点云分类的准确率结果如表5所示。该表显示了ConvMixer堆叠层数*depth*对PPFConvMixer网络性能的影响,当*depth*设置为4时,可以获得最佳效果。

表4 ModelNet40基准上不同k值的总体分类准确率

k	Acc.(%)
32	93.2
48	93.3
64	**93.8**
80	93.4

表5 使用不同深度ConvMixer层的ModelNet40分类准确率

depth	Acc.(%)
2	91.9
3	92.8
4	**93.8**
5	93.2

这是由于点云本身携带的信息有限，网络并非设置得越深越好，层数越深会带来梯度不稳定和网络退化问题，反而会导致网络性能下降。

5 结论

受文献[15]对Transformer有效性来源的探索，本文以Point-BERT[29]为研究基线，采用一致的操作来分割点云Patch，用更简单、轻便的ConvMixer架构取代了Transformer部分，提出了点云处理架构PointConvMixer。PointConvMixer在形状分类和部分分割实验中的表现都优于基于Transformer的模型，这表明Transformer的性能至少部分归功于将点云数据预处理成三维Patch的方式。

此外，本文进一步引入了PPFs作为编码点云局部特征的有效方法，提出了PointConvMixer的改进版本PPFConvMixer模型，在ModelNet40数据集上实现了更高的点云分类精度。这表明高效的标记化输入表示也是点云识别任务的一个关注方向。然而，基于PPFs的表示法不适用于使用多级采样获取密集特征图的点云分割任务。研究点云分类和分割任务的通用优化方法也是未来可探索的方向。

参考文献（References）：

[1] Qi C R，Liu W，Wu C，et al. Frustum pointnets for 3D object detection from RGB-D data[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2018：918-927.

[2] Tang Y，Chen M，Wang C，et al. Recognition and localization methods for vision-based fruit picking robots：a review [J]. Frontiers in Plant Science，2020，11：510.

[3] Kästner L，Frasineanu V C，Lambrecht J. A 3D-deep- learning-based augmented reality calibration method for robotic environments using depth sensor data[C]// 2020 IEEE International Conference on Robotics and Automation（ICRA），2020：1135-1141.

[4] Qi C R，Su H，Niessner M，et al. Volumetric and multi-view CNNs for object classification on 3D data[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2016：5648-5656.

[5] Roveri R，Rahmann L，Oztireli C，et al. A network architecture for point cloud classification via automatic depth images generation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2018：4176-4184.

[6] Sarkar K，Hampiholi B，Varanasi K，et al. Learning 3D shapes as multi-layered height-maps using 2D convolutional networks[C]// Proceedings of the European Conference on Computer Vision（ECCV），2018：71-86.

[7] Qi C R，Su H，Mo K，et al. PointNet：deep learning on point sets for 3D classification and segmentation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2017：652-660.

[8] Qi C R，Yi L，Su H，et al. Pointnet++：deep hierarchical feature learning on point sets in a metric space[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems，2017，30：5105-5114.

[9] Liu Y，Fan B，Xiang S，et al. Relation-shape convolutional neural network for point cloud analysis[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition，2019：8895-8904.

[10] Hornik K. Approximation capabilities of multilayer feedforward networks[J]. Neural Networks，1991，4(2)：251-257.

[11] Riegler G，Osman Ulusoy A O，Geiger A. OctNet：learning deep 3D representations at high resolutions[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2017：3577- 3586.

[12] Klokov R，Lempitsky V. Escape from cells：deep kd-networks for the recognition of 3D point cloud models[C]// Proceedings of the IEEE International Conference on Computer Vision，2017：863-872.

[13] Wang Y，Sun Y，Liu Z，et al. Dynamic graph CNN for learning on point clouds[J]. ACM Transactions on Graphics，2019，38(5)：1-12.

[14] Zhao H，Jiang L，Fu C W，et al. PointWeb：enhancing local neighborhood features for point cloud processing[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition，2019：5565 -5573.

[15] Trockman A，Kolter J Z. Patches are all you need?[DB /OL]. arXiv：2201.09792，2022.

[16] Drost B，Ulrich M，Navab N，et al. Model globally，match locally：efficient and robust 3D object recognition[C]// 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition，2010：998- 1005.

[17] Su H，Maji S，Kalogerakis E，et al. Multi-view convolutional neural networks for 3D shape recognition[C]// Proceedings of the IEEE International Conference on Computer Vision，2015：945-953.

[18] Yu T，Meng J，Yuan J. Multi-view harmonized bilinear network for 3D object recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2018：186-194.

[19] Feng Y，Zhang Z，Zhao X，et al. GVCNN：group-view convolutional neural networks for 3D shape recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2018：264-272.

[20] Yang Z，Wang L. Learning relationships for multi-view 3D

object recognition[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 7505-7514.

[21] Li Y, Bu R, Sun M, et al. PointCNN: convolution on x-transformed points[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, 31: 828-838.

[22] Xu M, Ding R, Zhao H, et al. PAConv: position adaptive convolution with dynamic kernel assembling on point clouds[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 3173-3182.

[23] Tolstikhin I, Houlsby N, Kolesnikov A, et al. MLP-Mixer: an all-MLP architecture for vision[C]// Proceedings of the 35nd International Conference on Neural Information Processing Systems, 2021, 34: 24261-24272.

[24] Touvron H, Bojanowski P, Caron M, et al. ResMLP: feedforward networks for image classification with data-efficient training[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(4): 5314-5321.

[25] Liu H, Dai Z, So D, et al. Pay attention to MLPs[J]. Advances in Neural Information Processing Systems, 2021, 34: 9204-9215.

[26] Yan X, Zheng C, Li Z, et al. PointANSL: robust point clouds processing using nonlocal neural networks with adaptive sampling[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 5589-5598.

[27] Zhao H, Jiang L, Jia J, et al. Point transformer[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021: 16259-16268.

[28] Guo M H, Cai J X, Liu Z N, et al. PCT: point cloud transformer[J]. Computational Visual Media, 2021, 7: 187-199.

[29] Yu X, Tang L, Rao Y, et al. Point-BERT: pre-training 3D point cloud transformers with masked point modeling[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022: 19313-19322.

[30] Wu Z, Song S, Khosla A, et al. 3D shapeNets: a deep representation for volumetric shapes[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015:1912-1920.

[31] Liu X, Han Z, Liu Y S, et al. Point2Sequence: learning the shape representation of 3D point clouds with an attention-based sequence to sequence network[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(01):8778-8785.

[32] Goyal A, Law H, Liu B, et al. Revisiting point cloud shape classification with a simple and effective baseline[C]// International Conference on Machine Learning(PMLR),2021: 3809-3820.

[33] Wang P S, Liu Y, Guo Y X, et al. O-CNN: octree-based convolutional neural networks for 3D shape analysis[J]. ACM Transactions on Graphics, 2017, 36(4):1-11.

[34] Wang C, Samari B, Siddiqi K. Local spectral graph convolution for point set feature learning[C]// Proceedings of the European Conference on Computer Vision (ECCV), 2018: 52-66.

[35] Li J, Chen B M, Lee G H. SO-NET: self-organizing network for point cloud analysis[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition2018: 9397-9406.

[36] Liu Y, Fan B, Meng G, et al. DensePoint: learning densely contextual representation for efficient point cloud processing[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 5239-5248.

[37] Xu Y, Fan T, Xu M, et al. SpiderCNN: deep learning on point sets with parameterized convolutional filters[C]// Proceedings of the European Conference on Computer Vision (ECCV), 2018:87-102.

编辑:赵志军