一种基于注意力机制的三维点云物体识别方法

钟 诚,周浩杰,韦海亮
(数学工程与先进计算国家重点实验室,江苏 无锡 214000)

摘 要:三维点云数据通常具备无序排列的结构。在三维点云数据处理领域,深度学习模型通常会利用最大池化等对称操作来处理点云的排列不变性。最大池化方法一方面会破坏点云的信息结构,使得局部信息与全局信息难以交互。另一方面,最大池化方法对点云信息过度压缩,得到的特征对局部细节描述不足。针对上述问题,提出了 AttentionPointNet 的网络结构。该网络利用注意力机制,使每个点与点云其余部分进行特征交互,实现了局部与全局信息的综合。为降低最大池化造成的信息损失,提出了一种稀疏卷积方法来替代池化操作。这种方法利用大步长的稀疏卷积实现全局信息的提取。在 ModelNet40 数据集上,AttentionPointNet 取得了 87.2% 的准确率。不使用池化层,完全采用卷积层实现的模型取得了 86.2% 的分类准确率。

关键词:注意力机制;点云;物体识别;池化;稀疏卷积
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2020)04-0041-05
doi:10.3969/j.issn.1673-629X.2020.04.008

A 3D Point Cloud Object Recognition Method Based on Attention Mechanism

ZHONG Cheng, ZHOU Hao-jie, WEI Hai-liang
(State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi 214000, China)

Abstract: 3D point cloud data usually has an unordered structure. In the field of point cloud data processing, deep learning models usually use the symmetry operations such as maximum pooling to deal with the permutation invariance of point clouds. On the one hand, this approach often destroys local information of point cloud data. On the other hand, the maxpooling method over-compresses point cloud information, and the extracted features are insufficiently described for local details. Aiming at those problems, we propose a network structure called AttentionPointNet which uses the attention mechanism to make each point interact with the rest of the point cloud to achieve the integration of local and global information. In order to reduce the information loss caused by the maximum pooling, we propose a sparse convolution to replace the pooling layer, which uses large stride sparse convolution to extract global information. On the ModelNet40 dataset, AttentionPointNet achieves 87.2% classification accuracy. The model, which only uses convolution layers to replace maxpooling layer, achieves 86.2% classification accuracy.

Key words: attention mechanism; point cloud; object recognition; pooling; sparse convolution

1 概 述

三维点云是指一个三维坐标系统中一组向量的集合。这些向量通常以 X,Y,Z 三维坐标的形式表示,一般用于描述物体的外貌形状。深度学习方法已经被广泛应用于图像识别、文本处理等领域。但目前而言,使用深度学习方法提取三维点云数据的特征仍然存在诸多障碍。其中最主要的原因在于点云的排列不变性。用于表示同一物体的 n 个点云数据点有 $n!$ 种排列方式,而点云的高层语义特征不能因为点云排列顺序的变化而改变。这意味着过去应用在网格状数据(如图像、文本)上的卷积方法难以应用在三维点云数据上。Charles R. Qi 等人提出的 PointNet[1] 为三维点云数据的处理打开了一扇新的大门。PointNet 将对称函数应用到三维点云的处理过程中,凭借最大池化的方法提取三维点云的高层特征,借此刷新了多项基准数据集的记录。最大池化的方法虽然在高层语义特征的提取

收稿日期:2019-05-09 **修回日期**:2019-09-11 **网络出版时间**:2019-12-18
基金项目:国家科技部重点研发计划项目(2018ZX01028101)
作者简介:钟 诚(1992-),男,硕士研究生,研究方向为计算机视觉、模式识别;周浩杰,副研究员,研究方向为分布式计算、数据智能;韦海亮,高级工程师,研究方向为分布式计算。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20191218.1113.050.html

上有很大优势,但是采用这种方法会造成三维点云局部几何特征的缺失,给局部信息的提取造成困难。自然语言处理问题同样需要妥善解决局部信息与全局信息的关系。自然语言处理中,解决这一问题的常用方法有:卷积网络、循环卷积网络[2]以及注意力机制[3]。其中注意力机制可以不依赖卷积的堆叠,实现局部与全局信息的整合。这一特点满足三维点云数据处理过程的需求,所以利用注意力机制提取点云信息是一种可行的技术路线。

文中的主要贡献分为两点:在点云特征提取中引入注意力机制,为点云局部与全局信息的整合提供一种可行方法;使用稀疏卷积层替代最大池化层,减少池化过程的信息损失。

2 相关工作

三维点云的特征提取技术在与点云相关的应用中发挥着重要作用。与深度学习特征有关的主流的点云特征提取方法主要可以分为三类:体素化方法[4]、多视图方法、最大池化方法。

体素化方法可以视作二维卷积在三维空间上的拓展与应用。体素化方法结构简单,但是在三维空间构造网格对内存资源消耗大,并且点云在空间中的分布往往是稀疏的,直接对空网格进行卷积会造成不必要的计算资源浪费。所以单纯的体素化方法一般难以完成高分辨率点云解析任务以及大规模点云处理任务[5]。多视图方法通过将三维点云投影到二维空间使用二维卷积方法完成三维数据的特征提取。多视图方法在三维目标检测方面有着独到的优势,但是在如三维目标分割等应用场景中效果不佳[6]。因为投影的过程造成了深度信息的损失。近年来,直接使用三维点云数据作为输入利用最大池化方法处理三维点云逐渐成为了研究热点。如 PointNet[1]等可以利用最大池化方法完成三维点云特征的高层语义信息提取,然而使用最大池化操作的缺陷在于模型会丧失感知局部信息的能力。针对上述问题,研究者设计出了能够描述局部特征的三维点云特征提取网络如 PointNet++[7],基于 Octree 的方法[8]、PointCNN[9]等。这些方法大都按照:分层、局部特征的提取、全局特征的提取、特征聚合等步骤处理点云信息。其中,为了构建全局关系往往需要对点云进行分层,分层的过程又可能会引入新的信息损失。比如文献[10]中的分层网络会将三维点云切割的过细,产生难以利用的点云碎片。在一些应用中,直接结合三维点云的局部与全局信息可能起到更好的特征处理效果。

自然语言中处理上下文关系的一些方法可以为三维点云的处理提供参考。自然语言处理通常利用循环卷积网络对上下文的信息进行抽取,解决局部与全局信息的整合问题。但是循环卷积网络的基本结构是一个递归模型。递归模型一般难以实现并行化并且循环卷积网络对全局信息的感知相对较弱,要逐步递归才能获得全局信息,一般要使用双向循环卷积网络。卷积网络只能获取局部信息,需要通过层叠来增大感受野。文献[11]提出了仅依赖注意力机制即可提取文本的全局信息的方法。笔者认为利用注意力机制同样能够在三维点云上获取全局信息。不同于直接构造特征提取层的方法,文中基本的思路是构建如下编码方案:

$$y_t = f(x_t, \boldsymbol{A}, \boldsymbol{B})$$

其中,$x_t \in X$ 表示原始点云数据,y_t 表示编码后的点云数据,$\boldsymbol{A}, \boldsymbol{B}$ 表示两个序列矩阵。若取 $\boldsymbol{A} = \boldsymbol{B} = \boldsymbol{X}$,则得到结果 $y_t = f(x_t, \boldsymbol{X}, \boldsymbol{X})$。$y_t$ 为 x_t 与全局 $\boldsymbol{X}$ 点云数据交互后的编码结果,这个结果既包含本地信息又包含全局信息,提取特征的问题转化成为确定编码函数 f 的问题。

3 方 法

本节结合注意力机制的基本概念介绍在点云中应用注意力机制的方法。针对最大池化层的缺陷,本节从理论上分析了池化层与卷积层的异同,给出了一种替换池化层的解决方案。

3.1 注意力机制

注意力机制的一般化定义如式(1)所示。

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{1}$$

其中,$\boldsymbol{Q} \in R^{n \times d_k}, \boldsymbol{K} \in R^{m \times d_k}, \boldsymbol{V} \in R^{m \times d_v}$。

不妨取 $\boldsymbol{q}_t \in \boldsymbol{Q}$,则对单个输入向量 $\boldsymbol{q}_t$ 求得的编码结果可以表示为:

$$\text{Attention}(\boldsymbol{q}_t, \boldsymbol{K}, \boldsymbol{V}) = \sum_{s=1}^{m} \frac{1}{Z} \exp\left(\frac{\langle \boldsymbol{q}_t, \boldsymbol{k}_s \rangle}{\sqrt{d_k}}\right)\boldsymbol{v}_s \tag{2}$$

其中,Z 为归一化因子,q, k, v 为 query, key, value 的简写。$\sqrt{d_k}$ 起调节作用使得送入激活函数的内积不至于过大。式(2)中的 $\exp\left(\frac{\langle \boldsymbol{q}_t, \boldsymbol{k}_s \rangle}{\sqrt{d_k}}\right)\boldsymbol{v}_s$ 是注意力机制的核心部分。$\exp\left(\frac{\langle \boldsymbol{q}_t, \boldsymbol{k}_s \rangle}{\sqrt{d_k}}\right)\boldsymbol{v}_s$ 主要用于衡量 $\boldsymbol{q}_t$ 与 $\boldsymbol{v}_s$ 的相似度。整个公式可以理解为寻找 $\boldsymbol{q}_t$ 与 $\boldsymbol{v}_s$ 间的一种非线性映射关系。式(2)的大体流程是将 $\boldsymbol{q}_t$ 与各 $\boldsymbol{k}_s$ 内积,再通过 softmax 的方式评估 $\boldsymbol{q}_t$ 与 $\boldsymbol{v}_s$ 的相似度,最终得到的结果通过加权求和得到一个 $\boldsymbol{d}_v$ 的向量。观察计算过程,除 $\boldsymbol{q}_t$ 外输入矩阵的其他部分也会影响向量 $\boldsymbol{d}_v$ 的计算结果,所以 $\boldsymbol{d}_v$ 不仅与 $\boldsymbol{q}_t$ 有关,也与输入矩阵中的其他部分产生关联,$\boldsymbol{d}_v$ 可以作为 $\boldsymbol{q}_t$ 在全局向量中

的表示。

在 Attention 机制基础之上,谷歌提出的 Multi – Head Attention 机制用于进一步提升模型的编码能力[11]。文中使用的注意力机制模型主要基于 Multi – Head Attention。相比基础的模型, Multi – Head Attention 机制有两点不同。一是将 Q,K,V 经矩阵参数进行映射,再送入 Attention 模型;二是将原始输入进行多次不共享参数的 Attention 操作,输出结果拼接。这两点改进能提升模型的描述能力。具体的模型如下所示:

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^{Q},\boldsymbol{K}\boldsymbol{W}_i^{K},\boldsymbol{V}\boldsymbol{W}_i^{V}) \tag{3}$$

其中, $\boldsymbol{W}_i^{Q} \in \mathbb{R}^{d_k \times \tilde{d}_k}, \boldsymbol{W}_i^{K} \in \mathbb{R}^{d_k \times \tilde{d}_k}, \boldsymbol{W}_i^{V} \in \mathbb{R}^{d_v \times \tilde{d}_v}$

最终输出的特征表示为:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\cdots,\text{head}_h)$$

3.2　在点云中使用注意力机制

注意力机制在点云数据与文本数据方面的应用密切相关又存在不同。应用于自然语言领域的注意力机制模型通常由一组编码器与解码器构成。编码器负责构建上下文关系,对词句进行映射,形成特征;解码器负责解释特征,对语义信息进行还原。文中应用注意力机制的主要目的是通过编码功能实现局部信息与全局信息的聚合,避免使用卷积堆叠逐层增加感受野的方法,进而绕过点云层次划分问题。所以文中主要利用注意力机制的编码部分,通过编码即可得到可用于点云分类的特征向量。

文中基于 Multi – Head Attention 机制对点云进行编码,需要在序列内部应用 Attention ,寻找序列内部的联系综合局部与全局信息。为实现这一目的,文中利用自注意力机制(Self Attention)[3]实现模型。所谓的自注意力机制就是将式(1)中的输入 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 设为同一矩阵 $\boldsymbol{X}$。具体的模型如下所示:

$$Y = \text{MultiHead}(\boldsymbol{X},\boldsymbol{X},\boldsymbol{X}) \tag{4}$$

其中, $\boldsymbol{X}$ 为三维点云集合, Y 为对点云的编码结果。

对某一点云 $\boldsymbol{x}_t \in \boldsymbol{X}$ 而言,其编码过程如式(5)所示:

$$\text{Attention}(\boldsymbol{x}_t,\boldsymbol{X},\boldsymbol{X}) = \sum_{s=1}^{m} \frac{1}{Z}\exp\left(\frac{\langle \boldsymbol{x}_t,\boldsymbol{x}_s\rangle}{\sqrt{d}_k}\right)\boldsymbol{x}_s \tag{5}$$

容易观察到每次对 $\boldsymbol{x}_t$ 进行编码的过程中,其他的点云也作为变量,影响到 $\boldsymbol{x}_t$ 的输入结果,得到的编码结果既蕴含本地信息又包括全局信息。直接使用点云作为输入提取到的信息仅包含点特征,而点特征的描述能力较弱,难以囊括邻近的几何信息。为了更好地获取局部几何信息,文中参照边缘卷积[12]的概念对点云输入重新进行构造。记原始的点云集合为 $\boldsymbol{X}$, $\boldsymbol{x}_t \in \boldsymbol{X}$,记 $\boldsymbol{x}_t$ 的最近邻为 $\boldsymbol{x}_{tn}$。$\boldsymbol{x}_t$ 的最近邻向量为 $(\boldsymbol{x}_t,\boldsymbol{x}_{tn})$。输入向量 $\boldsymbol{q}_t$ 记为 $\boldsymbol{q}_t = (\boldsymbol{x}_t,\boldsymbol{x}_m)$。这样做是为注意力机制提供了一个直观的可解释的作用机制。$\boldsymbol{q}_t$ 的编码结果可以视作其余向量在输入空间中的线性组合。按上述方法构造出的最近邻向量 $\boldsymbol{q}_t$ 完全可以用于重新生成原始的点云数据 $\boldsymbol{x}_t$。这意味着输入向量构造方案不会破坏原始三维点云的信息。

三维点云难以应用卷积的一个主要原因在于点云的排列不变性。观察注意力机制公式(3),在该式中如果将 $\boldsymbol{K},\boldsymbol{V}$ 按行打乱顺序,那么经过 Attention 模块编码得到的结果还是一样的。这在自然语言处理中会造成词序混乱的问题,所以在用注意力机制处理自然语言时通常还需要增加一个标注位置信息的模块。然而这个问题在三维点云的处理中反而成为一个优点,因为三维点云的输入本身是无序的,若 $\boldsymbol{K},\boldsymbol{V}$ 的顺序对编码结果不造成影响,那么说明这种编码方案可以适应三维点云的序列无关性, Attention 模块起到类似对称函数的作用。

3.3　对最大池化方法的替换

PointNet 中最关键的点是最大池化方法能够描述三维点云的分布,并且池化层大小与模型的性能密切相关。然而最大池化操作的缺点在于局部信息的损失较大。虽然增加池化层的宽度可以逼近三维点云的空间分布,但是真实的应用场景中不可能无限制增加池化层宽度。局部信息的损失同样会为感知三维点云的精细结构造成困难。注意到最大池化操作是对点云空间的一种降采样。使用其他的采样方法有可能也能够达成相同的效果。

文献[13]指出池化层可以用卷积的形式进行表示,这表明在处理点云数据时同样可能用卷积方法对池化层进行替换。

为了阐明两者的区别与联系,下面对二者进行对比。设 f 是原始数据经过一系列网络处理后得到的特征表示(feature map)。f 可以用一个三维矩阵 $W \times H \times N$ 表示。其中 W 、H 、N 分别为 feature map 的宽度、高度以及特征的通道数。在这个三维结构下,池化操作可以看成是一个用 p – norm 当作激活函数的卷积操作。当 p 趋向于正无穷时,就是最大池化操作。卷积式的池化层的公式表示如下:

$$s_{i,j,u}(f) = \left(\sum_{h=\lfloor k/2\rfloor}^{\lfloor k/2\rfloor}\sum_{w=\lfloor k/2\rfloor}^{\lfloor k/2\rfloor} |f_{g(h,w,i,j,u)}|^{P}\right)^{1/p} \tag{6}$$

其中, k 为池化层大小, r 为步长, $g(h,w,i,j,u) = (r\cdot i + h, r\cdot j + w, u)$ 为 f 到池化层的映射函数。

卷积层的公式定义如下:

$$c_{i,j,o}(f) = \sigma\left(\sum_{h=\lfloor k/2\rfloor}^{\lfloor k/2\rfloor}\sum_{w=\lfloor k/2\rfloor}^{\lfloor k/2\rfloor}\sum_{u=1}^{N}\theta_{h,w,u,o}\cdot f_{g(h,w,i,j,u)}\right) \tag{7}$$

其中, θ 为卷积权重, σ 为激活函数, $o \in [1,M]$

为输出通道。

通过观察上式不难发现,池化层可以视作一个特征级别的卷积。在式(7)中,当卷积步长取值非常大时,局部信息对卷积结果的影响相对较小[13]。此时卷积层对全局特征的提取性能较好。在目标分类的实验中同样证明了这一点:当取较大卷积核的步长时分类结果更好。

4 实 验

4.1 数据集

文中在 ModelNet40 数据集[14]上评估模型的三维形状分类性能。所有的点云数据来自对 40 个类别,共计 12 311 个 CAD 模型的采样。实验取其中 9 843 个模型用于训练,使用 2 468 个进行测试。对每个训练/测试模型实例抽取 2 048 个采样点。

4.2 AttentionPointNet 结构

图 1 所示为 AttentionPointNet 的基本结构。

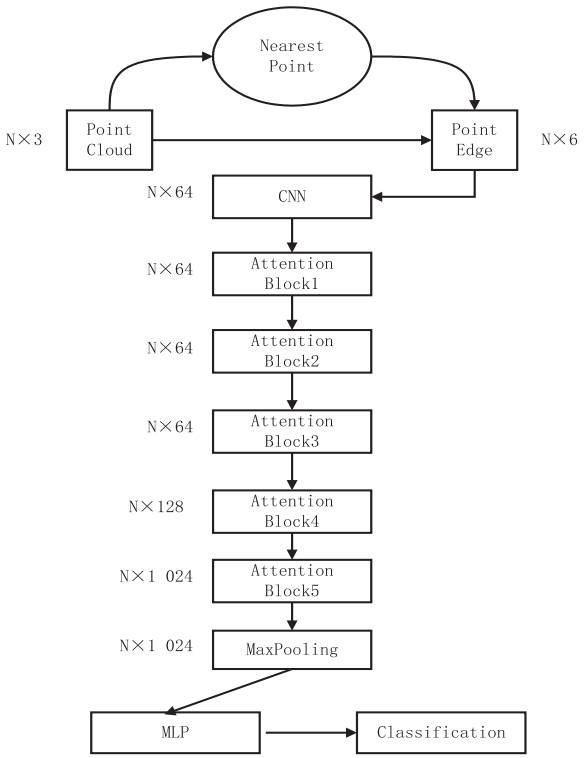

图 1 AttentionPointNet 模型

其中最基本的结构为 Attention Block。文中的 Attention Block 模块(见图 2)可以分为三个部分。

第一部分主要完成输入向量 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 的线性映射。实际的模型采用自注意力机制,所以 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 的取值均设置为三维点云的最近邻向量矩阵 $\boldsymbol{X}$。实验中采用长度为 64 的全连接层实现映射工作。为避免混淆,方便概念解释,下文仍使用 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 代指 Attention Block 中的三部分输入(query,key,value)。模型中的第二部分为 Multi - Head Attention 模块。这一模块主要完成特征的融合工作。$\boldsymbol{Q}$ 构成原始输入,$\boldsymbol{K}$ 构成特征向量空间,$\boldsymbol{V}$ 作为欲表征的结果,训练 head = Attention($\boldsymbol{QW}^Q,\boldsymbol{KW}^K,\boldsymbol{VW}^V$) 中的 $\boldsymbol{W}^Q$、$\boldsymbol{W}^K$、$\boldsymbol{W}^V$ 等参数矩阵。第三部分为一个残差模块。该模块将输入与 $\boldsymbol{V}$ 的值连接可以提升模型的分类性能,提高模型的分类准确率。

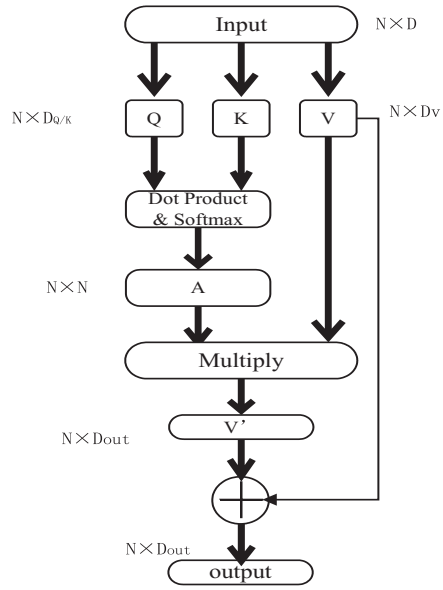

图 2 Attention Block 结构

4.3 AttentionPointNet 实现

AttentionPointNet 采用 Tensorflow[15]构建。整体的结构包括一层卷积层、五个 Attention Block 模块、一层最大池化层和多个三层连接层。第一层的卷积层卷积核大小为(1,1,64)。五个 Attention Block 的通道维度大小分别为 64、64、64、128、1 024。最大池化层宽度为 1 024。使用 Adam(adaptive moment estimation)算法作为优化算法,初始学习率为 0.001。每 20 轮学习率减半。学习轮数 300 轮。表 1 为不同模型在 ModelNet40 数据集上的分类结果。

表 1 ModelNet40 数据集分类结果

方法	ModelNet40 Classification (Accuracy)	ModelNet40 Retrieval (mAP)
POINTNET	89.2	86.0
POINTNET++	90.7	-
VOXNET	85.9	83.0
3DSHAPENETS	84.7	77.3
OURS(baseline)	87.2	84.8

4.4 去除最大池化层的 AttentionPointNet

现有的大多数基于 PointNet 的模型依赖于最大池化层对空间分布的拟合。文中提出的 Attention PointNet 对每个三维点云进行编码,编码的信息中包含了其他点的空间分布信息。对于三维点云的分类这一问题,除了使用最大池化的方法,其他的降采样方法

理应能取得类似的结果,比如经过特殊设计的卷积方法。在 ModelNet40 数据集上,文中分别改变卷积的步长,测试实验性能用于寻找最大池化层的替换品。

实验中取采样点个数为 256,batchsize 设为 128,训练轮数设为 300,对 Attention Block5 的输出结果($N \times 1\ 024$)进行卷积。输出的通道维度为 128。表 2 为用不同步长时替代最大池化层时,在 ModelNet40 数据集上的结果。

表 2　不同卷积核在 ModelNet40 数据集上的分类结果

Stride Size	Accuracy	Average Accuracy
16	75.2	69.8
32	81.2	76.8
64	80.5	76.0
128	84.2	78.8
160	86.0	82.2
200	85.9	82.8
256	85.6	81.8
512	85.2	81.3

实验的结果同样证明在三维点云数据处理中,池化层可以由卷积层进行代替,使用较大的步长能更好地提取点云的全局特征,取得更优的分类效果。并且当步长的取值超过一定阈值,泛化性能基本保持不变。

5　结束语

针对三维点云的全局与局部信息整合问题,提出了利用注意力机制对三维信息进行整合的方法。目前利用深度学习技术处理三维点云的主流方法大都依赖最大池化层的表征能力。结合注意力机制,使用大步长的卷积方法对点云的高层次信息进行抽取,取得了与使用最大池化方法类似的结果。这表明除了池化方法,卷积方法也具备三维点云的高层信息提取的潜力。使用注意力机制的好处是能够一步到位捕捉到三维点云局部与全局的联系。但是相比长程的、全局性的依赖,在三维点云处理中有部分问题比如点云的分割问题,更加依赖于局部结构。在这种情况下,文中使用的最近邻向量构造方法就不太合适。而使用 K 近邻描述局部特征可能是更为合理的方案。

参考文献:

[1] QI C R, SU H, MO K, et al. Pointnet: deep learning on point sets for 3d classification and segmentation[C]//IEEE conference on computer vision and pattern recognition. Honolulu, HI, USA: IEEE, 2017: 652-660.

[2] CHO K, VAN MERRIËNBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]//Conference on empirical methods in natural language processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1724-1734.

[3] LIN Z, FENG M, SANTOS C N, et al. A structured self-attentive sentence embedding[C]//International conference on learning representations. [s. l.]: [s. n.], 2017: 232-240.

[4] MATURANA D, SCHERER S. Voxnet: a 3d convolutional neural network for real-time object recognition[C]//2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). Hamburg, Germany: IEEE, 2015: 922-928.

[5] WANG P, LIU Y, GUO Y, et al. O-cnn: Octree-based convolutional neural networks for 3d shape analysis[J]. ACM Transactions on Graphics, 2017, 36(4): 72.

[6] SU H, MAJI S, KALOGERAKIS E, et al. Multi-view convolutional neural networks for 3d shape recognition[C]//IEEE conference on computer vision. Venice, Italy: IEEE, 2015: 945-953.

[7] QI C R, YI L, SU H, et al. Pointnet++: deep hierarchical feature learning on point sets in a metric space[C]//Advances in neural information processing systems. Long Beach, USA: IEEE, 2017: 5099-5108.

[8] VO A V, TRUONG-HONG L, LAEFER D F, et al. Octree-based region growing for point cloud segmentation[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2015, 104: 88-100.

[9] LI Y, BU R, SUN M, et al. PointCNN: convolution on X-transformed points[C]//Advances in neural information processing systems. Montréal, Canada: IEEE, 2018: 820-830.

[10] LANDRIEU L, SIMONOVSKY M. Large-scale point cloud semantic segmentation with superpoint graphs[C]//IEEE conference on computer vision and pattern recognition. Salt Lake City, Utah, USA: IEEE, 2018: 4558-4567.

[11] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in neural information processing systems. Long Beach, USA: IEEE, 2017: 5998-6008.

[12] WANG Y, SUN Y, LIU Z, et al. Dynamic graph CNN for learning on point clouds[J]. ACM Transactions on Graphics, 2019, 36(1): 321-334.

[13] SPRINGENBERG J T, DOSOVITSKIY A, BROX T, et al. Striving for simplicity: the all convolutional net[C]//International conference on learning representations. San Diego, CA, USA: IEEE, 2015: 732-740.

[14] WU Z, SONG S, KHOSLA A, et al. 3d shapenets: a deep representation for volumetric shapes[C]//IEEE conference on computer vision and pattern recognition. Venice, Italy: IEEE, 2015: 1912-1920.

[15] ABADI M, BARHAM P, CHEN J, et al. Tensorflow: a system for large-scale machine learning[C]//12th USENIX symposium on operating systems design and implementation. Savannah, GA, USA: ACM, 2016: 265-283.