PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

摘要

很多的点云语义分割技术需要对点云进行预处理操作，大多数研究人员通常在将这些数据输入深度网络架构之前将其转换为规则的 3D 体素网格或图像集合，但是这些预处理操作通常会大大增加性能消耗，内存开支和信息损失。PointNet 直接将原始点云数据作为输入，避免了预处理步骤的开销。但是由于 PointNet 网络只考虑了全局特征，直接暴力地将所有的点最大池化为了一个全局特征，因此丢失了每个点的局部信息，局部的点与点之间的联系并没有被网络学习到。本文通过自注意力特征聚合模块，逐点计算每个点的 k 个近邻，并通过一个自注意力机制来聚合近邻点之间的特征，使每个点都学习到局部特征，进而扩大每个点的感受野。实验结果表明，注意力聚合机制的方法相比于原始的 PointNet 网络在点云分类任务上拥有更高的精度。

关键词：点云分割；部件分割；点云分类

1 引言

智慧城市、虚拟现实 (VR)、增强现实 (AR)、自动驾驶作为新一轮的研究热点，极大程度上依赖于对点云分析技术的发展。由于点云的无序性，高维性和非结构性，传统的点云语义分割技术通常需要对点云进行预处理操作，大多数研究人员通常在将这些数据输入深度网络架构之前将其转换为规则的 3D 体素网格或图像集合，但是这些预处理操作通常会大大增加性能消耗，内存开支和信息损失。

出于这些原因，PointNet 直接将原始点云数据作为输入，避免了预处理步骤的开销。点云作为一种简单统一的三维数据结构，避免了网格的组合不规则性和复杂性，因此更容易被深度网络学习，但点云由于其自身的无序性和非结构化的性质，仍然需要保证其的排序不变性，PointNet 通过一个对称函数的映射来保证网络拥有这种性质。此外，还需要考虑到点云的刚性变换的各种特征的不变性。

PointNet 是一个统一的端到端的结构，它直接将原始点云数据作为输入，并输出每个输入点的语义标签。PointNet 网络的基本架构非常简单，因为在初始阶段，每个点都被相同且独立地处理。为了简单起见，每个点仅由其三个坐标 (x,y,z) 表示，也可以考虑增加更多的点的特征维度，通过计算法线或者点的 rgb 颜色和其他局部或全局特征来添加额外的维度。

如上所述，将这些学习到的最优值整合到整个形状的全局描述符中用于分类，或者用于预测每个点的标签分割。PointNet 输入格式很容易应用刚性或仿射变换，因为每个点都独立

变换。因此，我们可以添加一个与数据相关的空间转换网络 T-Net，该网络试图在 PointNet 处理数据之前对数据及其特征进行规范化，从而进一步改进结果。

PointNet 的主要贡献如下：

- 设计了一个新颖的深层网络架构来处理三维中的无序点集
- 设计的网络表征可以做三维图形分类、图形的局部分割以及场景的语义分割等任务
- 提供了完备的经验和理论分析来证明 PointNet 的稳定和高效。
- 充分的消融实验，证明网络各个部分对于表征的有效性。

2 相关工作

2.1 传统点云分割

传统的点云语义分割方法大致可分为基于特征聚类的算法 [1]，基于区域增长的算法 [2–5]，基于模型拟合的算法 [6,7] 等等，这三种类型的算法都有各自的适用场景，但放在场景点云中的效果并不理想：1) 基于特征聚类的方法对特征差异较大的点云场景效果较好，且较为稳定，但是其性能开销较大，而且无法对重叠混杂的点云对象进行有效的分割。2) 基于区域增长的算法较为稳定，但是对超参数敏感，参数调整困难。3) 基于模型拟合的算法处理速度较快，在简单几何点云场景的应用效果较好，但是在分割复杂形状的场景中效果较差。

2.2 基于体素的点云分割

基于体素的算法将原始点云数据转化为规则化的体素表示，然后将其输入到 CNN 神经网络中进行特征学习，如著名的 Voxnet 模型 [8]，该模型对稠密点云处理效果较好，但是在稀疏点云中使体素网格的排列密度低，占用内存过高，训练耗时长，且转化点云的同时也会丢失点云数据的信息，切割精度不高。为了解决这些问题，基于三维空间划分结构的 KD 树、八叉树的深度学习架构 Kd-net [9]、OctNet [10] 被提出来，这种算法将空间划分，使得计算资源能够被更多的分配到体素密度大的区域，能够构建更深的网络，但对噪点敏感，也未能涉及体素的几何结构。SEGcloud [11] 先将点云进行体素化之后通过 FCNN 进行粗粒度的体素预测，随同通过三线性插值和全连条件随机场输出细粒度的语义，这种方法的分割精度更高，但计算开销也大。而 PointGrid [12] 方法既有 Voxnet [8] 的简单性，相比于 PointNet 它也更加关注全局信息，同时相比 SEGcloud 计算开销更小，但它对上下文的学习不充足。

2.3 基于多视图的点云分割

基于多视图的算法最经典的是 MVCNN [13] 算法，这种算法将三维点云投影成二维图像，然后对图像的特征进行提取，并用池化层进行特征聚合，最后得到的特征输入到 CNN 中获得分割结果，尽管相比于传统语义分割算法相比 MVCNN 的效果更好，但多个图像会使几何信息冗余，同时点云在投影成二维图像的过程中会使信息丢失。为了改进信息丢失的问题，Guerry 等人相继提出了 SnapNet [14] 和 SnapNet-R [15] 算法,SnapNet 对点云的 RGB 图和深度图进行处理，获得密集的点标记，拥有更好的分割效果，而 SnapNet-R 则在 SnapNet 的基础上对

rgb 图进行了标记，能够保留更多的信息。SqueezeSeg [16] 和 SqueezeSegV2 [17] 在 SqueezeNet [18] 的基础上进行改进，SqueezeSeg 使用球面投影获取二维图像，再使用 SqueezeNet 提取特征，并在条件随机场中获取细粒度的语义，SqueezeSegV2 在 SqueezeSeg 的基础上使用域适应训练管道，进一步提升了分割精度，但与基于体素的算法类似，基于多视图的算法同样需要开销极大的预处理操作。

3 本文方法

3.1 本文方法概述

网络结构如图1所示，Pointnet 分类网络以 n 个点作为输入，进行输入和特征变换，然后通过最大池化对点特征进行聚合，网络的输出是 k 个类的分类分数。分割网络不同于分类网络，它将全局和局部特征连接，并输出每个点的预测分数，其中 mlp 是多层感知机。

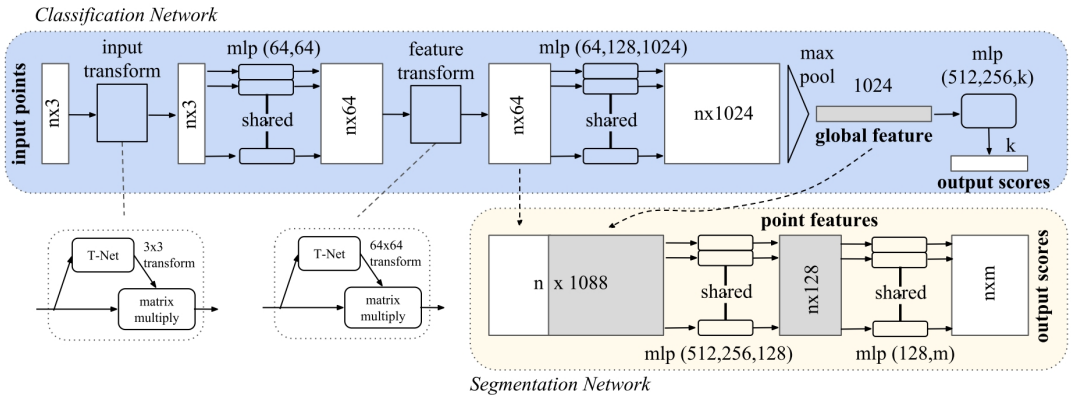

图 1. PointNet 结构

3.2 MaxPooling 层

为了实现网络的排序不变性，目前有三种方法:1) 将输入点云进行排序;2) 将输入点云作为一个序列来训练 RNN，并通过各种排列来增强训练数据;3) 使用一个简单的对称函数对每个点的信息进行聚合。对称函数以 n 个向量作为输入，并输出一个与输入顺序不变的新向量。排序并不是一个容易的方法，特别是在高维空间中的排序，并不能保证输出的顺序是稳定的。而如果使用 RNN 的方法来实现点云的排序不变性的话，实验证明了 RNN 在输入为几十的小长度序列上面有着较高的鲁棒性，但对于点云数据这种有着超过万亿级别的数据，其表现不佳，而且通常会带来极大的且不能接受的性能损耗。

所以 PointNet 使用对称函数来实现网络的排序不变性，通过对点集中的变换元素应用对称函数来近似定义在点集中的一般函数:

$$f(\{x_1, \ldots, x_n\}) \approx g(h(x_1), \ldots, h(x_n)), \tag{1}$$

where $f : 2^{\mathbb{R}^N} \to \mathbb{R}, h : \mathbb{R}^N \to \mathbb{R}^K$ and $g : \underbrace{\mathbb{R}^K \times \ldots \times \mathbb{R}^K}_{n} \to \mathbb{R}$ is a symmetric function.

这种方法首先将点云特征进行升维，利用了高维的特征空间的信息冗余性，使用最大池化来聚合特征，方法非常简单: 通过多层感知器 MLP 网络近似一个映射函数 h，通过单变量函数和最大池化函数的组合近似 g。原文的消融实验证明这是有效的，通过 h 的集合，PointNet 可以学习一些 f 来获取集合的不同性质，同时又防止了关键信息的丢失。

3.3 T-net 特征对齐模块

仅有排序不变性是不够的，当点云经过一定的几何变换比如刚性变换时，点云的语义标记必须也应该是不变的。因此，我们期望通过 PointNet 的点集学习到的表示对于这些变换是不变的，比如旋转不变性。一个自然的解决方案是在特征提取之前将所有输入集对齐到一个规范空间。PointNet 通过一个迷你网络 T-Net 来预测一个 3x3 的仿射变换矩阵，并直接将这个变换应用到输入点的坐标上。该网络本身类似于大网络，由点特征提取、最大池化和全连通层等基本模块组成。事实上，T-Net 还可以用在其它特征的对齐上面，比如图 1 中使用了第二个 T-Net 网络来预测一个 64x64 的变换矩阵。

4 复现细节

4.1 与已有开源代码对比

传统的 PointNet 网络虽然作为一个轻量级的网络，拥有处理速度快且对数据集缺失鲁棒性高的优点，但是由于 PointNet 网络只考虑了全局特征，直接暴力地将所有的点最大池化为了一个全局特征，因此丢失了每个点的局部信息，局部点与点之间的联系并没有被网络学习到。在分类和物体的 Part Segmentation 中，这样的问题还可以通过中心化物体的坐标轴来部分地解决，但在场景分割中，这就导致效果十分一般了。

为了进一步提高 PointNet 的分割精度和在场景点云的理解能力，本文引入了自注意力特征聚合模块，该模块将会逐点计算每个点的 k 个近邻，并通过一个自注意力机制来聚合近邻点之间的特征，使每个点都获得局部特征，扩大了每个点的感受野。新增的代码如下所示:

此外，原 PointNet 网络中试图用 T-Net 结构来实现点云数据的旋转不变性，而在实际实验中，T-net 的效果较小，但反而增加了一部分的计算量，所以在复现时选择不使用 T-net 特征对齐模块。

```
    @staticmethod
    def neighbors_gather(batch_pc, index, k):
        batch = batch_pc.size()[0]
        point_num = batch_pc.size()[1]
        feature_size = batch_pc.size()[2]
        gather_index = torch.repeat_interleave(index,
        repeats=feature_size, dim=2)
        gather_pc = batch_pc.gather(dim=1, index=gather_index).
        view(batch, point_num, k, feature_size)
        return gather_pc

```

```
    @staticmethod
    def self_attention_aggregate(batch_pc):
        batch = batch_pc.size()[0]
        point_num = batch_pc.size()[1]
        neigh_num = batch_pc.size()[2]
        feature_size = batch_pc.size()[3]
        f_shape = batch_pc.reshape((-1, neigh_num, feature_size))
        att_fc = torch.nn.Linear(feature_size,
        feature_size, bias=False, device='cuda:0')
        att_activate = att_fc(f_shape)
        att_score = torch.softmax(att_activate, dim=1)
        f_agg = f_shape * att_score
        f_agg = torch.sum(f_agg, dim=1)
        return f_agg.reshape((batch, point_num, feature_size))
```

4.2 实验环境搭建

原论文放出的源码所使用的框架是 tensorflow1.0，由于 tensorflow1.0 的调试困难，这里改用了 Python3.7+pytorch1.13.1+cu116 的版本，操作步骤如下：

- 下载源码

```
git clone https://github.com/fxia22/pointnet.pytorch
```

- 安装第三方库

```
cd pointnet.pytorch
pip install -e .
```

- 数据集下载和 Cython 程序编译

```
cd scripts
bash build.sh #build C++ code for visualization
bash download.sh #download dataset
```

4.3 ShapeNet 数据集介绍

ShpaeNet 是点云中一个比较常见的数据集，它能够完成部件分割任务，即部件知道这个点云数据大的分割，还要将它的小部件进行分割。它总共包括十六个大的类别，每个大的类别有可以分成若干个小类别（例如，飞机可以分成机翼，身体等小类别），总共有五十个小类别。其中数据集结构如图2所示

```
Airplane        02691156
Bag     02773838
Cap     02954340
Car     02958343
Chair   03001627
Earphone        03261776
Guitar  03467517
Knife   03624134
Lamp    03636649
Laptop  03642806
Motorbike       03790512
Mug     03797390
Pistol  03948459
Rocket  04099429
Skateboard      04225987
Table   04379243
```

```
-0.03614 0.06066 0.03030
0.02921 0.01067 0.05275
0.02921 0.01175 0.01386
0.01063 0.08777 0.22920
0.07625 -0.26221 0.14661
0.03465 0.03107 0.18211
0.01529 0.08325 0.07364
0.02010 0.05674 0.05885
0.01737 0.07218 0.05529
0.01249 0.10043 0.02607
0.02374 0.03648 0.05526
0.02170 0.04847 0.03865
0.02788 0.29411 0.03502
-0.04831 -0.08412 0.04088
-0.02758 -0.00760 -0.19385
0.01862 0.06757 0.26594
-0.04404 -0.04254 -0.19244
0.04988 -0.13456 -0.16585
0.02724 0.25457 -0.18455
-0.06670 -0.35122 0.10890
0.08575 -0.35122 0.09017
-0.08994 -0.35122 0.09243
0.02650 -0.35122 0.09636
0.04712 -0.35122 0.09507
0.02427 -0.35122 0.08014
-0.00566 -0.35122 0.09789
-0.01466 0.13214 -0.19596
```

图 2. ShapeNet 数据集格式

下载好数据集之后，数字文件夹里面放的都是每个大类的点云数据，shapenet 文件夹结构如图3所示

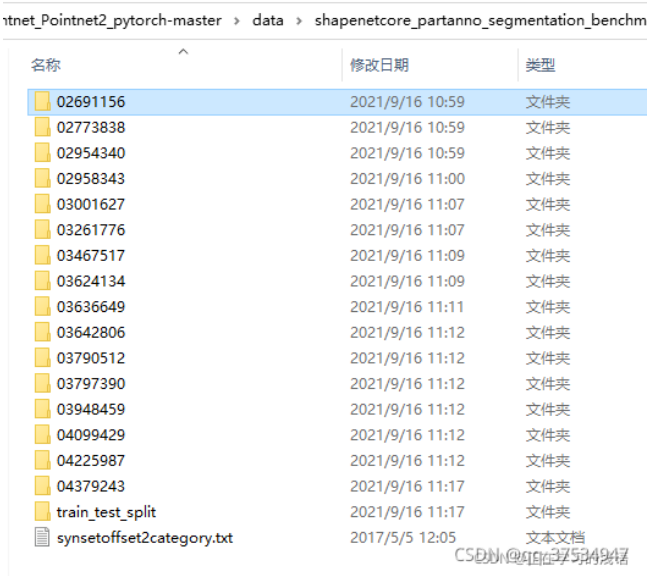

图 3. ShapeNet 文件夹结构

打开其中的文件夹，可以发现里面是很多 txt 文件。每个 txt 文件是一个点云数据，每一行表示一个点，每一列分别表示 xyz 坐标，每个点云数据由很多点组成。

4.4 创新点

改进后的 PointNet 有如下创新点：通过自注意力机制的特征聚合能够最大可能的保留在最大池化过程中的关键信息的丢失，同时使每个点都包含了一定的局部特征，对原网络中仅仅是对每个点表征，对局部结构信息整合能力太弱的问题进行了改进。注意力机制的公式如下所示：

$$Attention(K, Q, V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{2}$$

与用最大池化或者平均池化来聚合相邻点的方法不同，使用自注意力机制的聚合方法更加柔和，拥有更少的信息损失。

5 实验结果分析

本文在一个不完全的 ShapeNet 数据集上面进行测试，其中包含了 15011 个点云模型，其中 12137 个模型用于训练，2874 个模型用于测试，总计 16 个类别的模型，而由于某些类别的训练数据较少，会出现该类别的分类结果较差的情况。本文通过 89 个 epoch 进行测试，每个 epoch 选取 32 个模型为 1 个 batch，每个模型采样 2500 个点。

Method	mAC(%)	mIOUs(%)
PointNet	90.36	42.27
Att-aggregate	91.68	44.20

表 1. 注意力聚合方法和原始方法在部分 ShapeNet 数据集上平均准确度 (mAC) 和平均交并比 (mIOUs) 的对比

表1对比了原文方法和加入注意力聚合模块的方法，结果显示，注意力聚合机制的方法拥有更优的性能在点云分类的任务上面。而表2的结果也同样证明了这一点。在图4中，对 ShapeNet 的预测结果进行了可视化。

Method	mean	Airplane	Bag	Cap	Car	Chair	Earphone	Guitar	Knife	Lamp	Laptop	Motorbike	Mug	Pistol	Rocket	Skateboard	Table
PointNet	42.27	84.26	3.37	7.87	67.06	90.79	8.43	74.59	47.85	76.48	54.12	37.45	27.34	20.22	0.00	1.12	91.41
Att-aggregate	45.37	92.54	1.12	8.12	80.06	95.45	10.48	66.48	36.91	67.59	55.24	41.76	26.78	37.83	7.87	4.49	93.05

表 2. 注意力聚合方法和原始方法在部分 ShapeNet 数据集上各个类别的平均交并比 (mIOUs) 的对比

图 4. ShapeNet 预测结果可视化，在上的一行点云数据表示 ground truth，其下的一行点云表示模型预测值

6 总结与展望

正如前面所介绍的，虽然 PointNet 的轻量级以及其将点云直接作为输入的方式保证了 PointNet 的极快处理速度并且让 PointNet 对数据具有一定的鲁棒性，但显然 PointNet 仍有很大的局限性，PointNet 提取特征的方式是对所有点云数据提取一个全局特征，这也和目前在 2D 图像领域流行的能够逐层提取局部特征的 CNN 方式不一样。CNN 通过分层不断地使用卷积核扫描图像上的像素并做内积，使得越到后面的特征图感受野越大，同时每个像素包含的信息也越多。为了能达到近似 CNN 的效果，可以考虑通过特征聚合的方式和下采样过程模拟一次 CNN 的卷积操作，进而通过多次这样的操作来实现一种在点云数据上的类似 CNN 的卷积网络。

参考文献

[1] Qingming Zhan, Liang Yu, and Yubing Liang. A point cloud segmentation method based on vector estimation and color clustering. In *The 2nd International Conference on Information Science and Engineering*, pages 3463–3466. IEEE, 2010.

[2] Anh-Vu Vo, Linh Truong-Hong, Debra F Laefer, and Michela Bertolotto. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:88–100, 2015.

[3] Li Lun Huang, Wen Guo Li, Qi Le Yang, and Ying Chun Chen. Segmentation algorithm of three-dimensional point cloud data based on region growing. *Applied Mechanics and Materials*, 741:382–385, 2015.

[4] Renzhong Li, Yangyang Liu, Man Yang, and Huanhuan Zhang. Three-dimensional point cloud segmentation algorithm based on improved region growing. *Laser & Optoelectronics Progress*, 55(5):051502, 2018.

[5] Shuning Fan, Na Huang, Pengfei Fang, and Junjie Zhang. A 3d point cloud segmentation method based on local convexity and dimension features. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 5012–5017. IEEE, 2018.

[6] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[7] Bisheng Yang and Zhen Dong. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS journal of photogrammetry and remote sensing*, 81:19–30, 2013.

[8] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[9] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE international conference on computer vision*, pages 863–872, 2017.

[10] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017.

[11] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, pages 537–547. IEEE, 2017.

[12] Truc Le and Ye Duan. Pointgrid: A deep network for 3d shape understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9204–9214, 2018.

[13] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[14] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers & Graphics*, 71:189–198, 2018.

[15] Joris Guerry, Alexandre Boulch, Bertrand Le Saux, Julien Moras, Aurélien Plyer, and David Filliat. Snapnet-r: Consistent 3d multi-view semantic labeling for robotics. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 669–678, 2017.

[16] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[17] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 international conference on robotics and automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[18] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.