



中国科学技术大学
University of Science and Technology of China

PointNet++

中国科学技术大学

2023 年 12 月 19 日

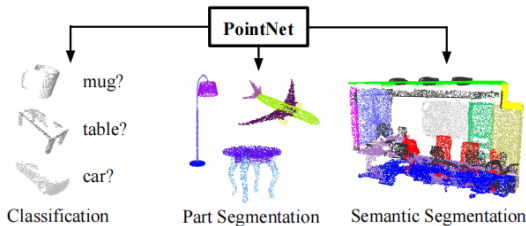


- 1 介绍
- 2 PointNet
- 3 PointNet++
- 4 实验复现
- 5 MRG 方法实现



- 1 介绍
- 2 PointNet
- 3 PointNet++
- 4 实验复现
- 5 MRG 方法实现

点云是一种重要的几何数据结构。由于其不规则的格式，大多数研究人员将这些数据转换为规则的 3D 体素网格或图像集合。然而，这使得数据不必要地庞大，并造成问题。研究人员设计了一种新型的神经网络 PointNet，它直接以点云作为输入，考虑了输入点的排列不变性。PointNet 提供了一个统一的架构，用于对象分类，物体分割，场景分割。



研究者设计了一个深度学习框架，直接使用无序点集作为输入。点云表示为一组 3D 点 $\{P_i | i = 1, \dots, n\}$ ，其中每个点 P_i 是其 (x, y, z) 坐标加上诸如颜色、法线等的额外特征通道的向量。

对于对象分类任务，Pointnet 为所有 k 个候选类输出 k 个分数。对于语义分割，输入可以是用于部分区域分割的单个对象，或者是用于对象区域分割的 3D 场景。Pointnet 将为 n 个点中的每个点生成 m 个语义类别分数，共输出 $n \times m$ 个分数。

3D 计算机视觉

Volumetric CNNs: 将三维卷积神经网络应用于体素化 3D 视觉学习。然而，由于 3D 数据的稀疏性和三维卷积的计算成本，体素表示受到其分辨率的限制。FPNN 和 Vote3D 提出了处理稀疏性问题的特殊方法，然而，处理非常大的点云仍然是一个挑战。**Multiview CNNs:** 将三维点云渲染成二维图像，然后应用二维卷积网络对其进行分类。通过设计良好的图像学习网络，这种方法在形状分类和检索任务上取得了主要的性能。然而，将它们扩展到场景分割或其他 3D 任务，如点云补全是非常困难的。**Feature-based DNNs:** 将三维数据转换为一个向量，然后利用全连接网络对形状进行分类。这样提取到特征的代表能力有很大的约束，因为将无序的点云转化成了有序的向量。

无序性，与图像中的像素阵列或体素网格中的体素阵列不同，点云是一组没有特定顺序的点，输入为 N 个点的网络需要对 $N!$ 保持输出不变性。交互性，点云中的这些点来自于具有相同距离度量的欧几里得空间。这意味着点与点之间不是孤立的，相邻的点形成了一个相互关联的子集，这意味着模型需要从附近的点捕获局部结构的语义特征。仿射性，点云是一个完整的仿射对象，即对整个点云进行仿射变换如旋转和平移后仍和原始点云语义特征保持一致。



- 1 介绍
- 2 PointNet
- 3 PointNet++
- 4 实验复现
- 5 MRG 方法实现

应用点对称函数来近似点集上的一般函数，通过多层感知器网络近似 h ，通过单变量函数和最大池函数的组合近似 g

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n))$$

$$\forall \epsilon > 0, \exists h, \gamma, \quad st. \left| f(S) - \gamma \left(\max_{x_i \in S} \{h(x_i)\} \right) \right| < \epsilon$$



在计算全局点云特征向量之后，通过将全局特征与每个点特征拼接起来反馈给每个点。然后我们基于拼接的点特征提取新的逐点特征，其中包含了局部信息和全局信息。

通过一个 T-net 预测仿射变换矩阵，并直接将此变换应用于输入点云，其中 T-net 只由简单的全连接层组成。这一思想也可以进一步扩展到特征空间的对齐。我们可以对逐点特征应用另一个对齐网络，使用同样的方法预测特征仿射变换矩阵，以对齐来自不同输入点云的特征。为了将特征变换矩阵约束为接近正交矩阵，将正则化项添加到训练损失中。

$$L_{reg} = \|I - AA^T\|_F^2, \quad (2)$$

PointNet 只能捕捉点云的全局特征，不能捕获欧氏空间点集的局部结构特征，从而限制了其识别细粒度模式的能力和对复杂场景的泛化能力。



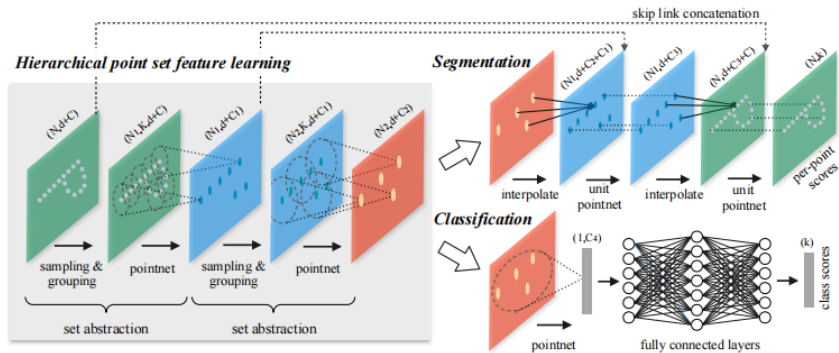
- 1 介绍
- 2 PointNet
- 3 PointNet++**
- 4 实验复现
- 5 MRG 方法实现

使用分层特征学习代替 PointNet 中使用单个最大池化聚合点云特征。具体来说，每一个抽象层由一个采样层，一个分组层和一个学习层组成。

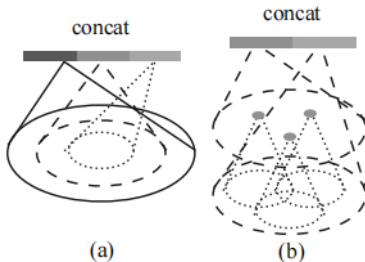
采样层，给定输入特征点集，使用迭代最远点采样（FPS）来选择特征点集的子集，与随机采样相比，在相同的质点数下，该方法对整个点集的覆盖率更高。

分组层，使用球查询搜索到质心点的特征距离在半径内的所有点（在实现中设置上限 K ），与 kNN 相比，球查询的局部邻域保证了固定的区域尺度，从而使局部区域特征在空间上更具泛化性，且利于学习层中的 PointNet 提取特征。

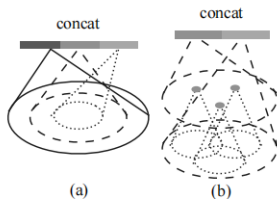
学习层，使用 PointNet 学习逐点特征，局部区域中的点的坐标首先被转换成相对于质心点的局部坐标系，通过 PointNet 得到该局部区域的全局特征作为下一个采样层的输入。



多尺度分组 (MSG), 如图 (a) 所示, 在每一个抽象层中使用具有不同尺度的分组层, 然后根据 PointNets 提取每个尺度的特征。不同尺度的特征被连接以形成多尺度特征。



多分辨率分组 (MRG)，由于上面的 MSG 计算开销较大，因为它在每个质心点的大规模邻域中运行 PointNet。通常由于在底层时质心点的数量相当大，因此时间成本是高昂的。MRG 的层特征是两个向量的拼接。一个向量（图中左侧）是通过使用单尺度分组 (SSG) 汇总每个子区域的特征而获得的。另一个矢量（右）是通过使用单个 PointNet 直接处理局部区域中的所有原始点获得的特征。当局部区域的密度低时，左侧向量比右侧向量更不可靠，因为在计算左侧向量时子区域点云稀疏导致采样不足。在这种情况下，右侧向量的权重应该更高。相反，当局部区域的密度高时，左侧矢量提供了更精细的信息，此时应该提高左侧向量的权重。



使用基于特征距离的插值传播方法逆向生成前一个抽象层所有点的近邻特征，再与前一个抽象层的分组学习特征拼接为逐点分割特征，使用 PointNet 的部分仿射架构进行特征学习，进行下一轮特征传播，直到传播至原始点云。在插值中，使用 k 个最近邻的反距离加权平均特征。

$$f^{(j)}(x) = \frac{\sum_{i=1}^k w_i(x) f_i^{(j)}}{\sum_{i=1}^k w_i(x)} \quad \text{where} \quad w_i(x) = \frac{1}{d(x, x_i)^p}, j = 1, \dots, C$$



- 1 介绍
- 2 PointNet
- 3 PointNet++
- 4 实验复现**
- 5 MRG 方法实现



实验环境：

RTX 3090, Pytorch 1.10.1, Python 3.7

参数设置：

分类任务：batchsize 为 24, epoch 为 200, 学习率为 0.001, optimizer 为 Adam

零件分割任务：batchsize 为 16, epoch 为 251, 学习率为 0.001, optimizer 为 Adam

语义分割任务：batchsize 为 16, epoch 为 32, 学习率为 0.001, optimizer 为 Adam

ModelNet40(用于分类任务):

包含了 40 个类别（如飞机，汽车等）的 CAD 模型数据。训练集有 9843 个点云数据，测试集有 2468 个点云数据。

ShapeNet(用于部件分割任务):

包含 14 个大类别（如飞机，椅子等）和 55 个小类别的 CAD 模型数据。训练集有 14007 个点云数据，测试集有 2874 个点云数据。

S3DIS(用于语义分割任务):

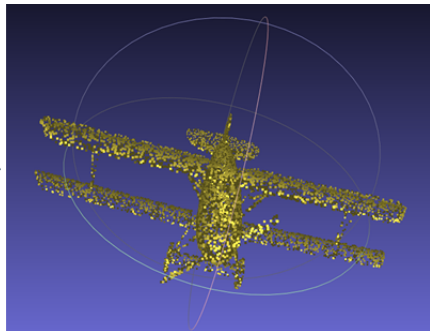
是大规模室内三维空间数据集，包含了 6 个建筑物，共计 271 个房间。房间内包含 13 类物体（如天花板、地板、门、沙发等）。

ModelNet40(分类任务): 源数据 (txt 文件) 进行可视化:

airplane_0001.txt - 记事本

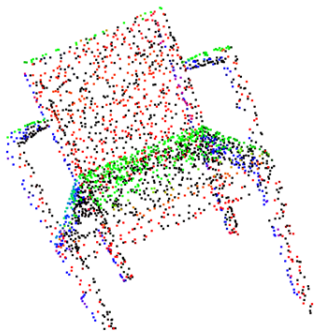
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
-0.098790,-0.182300,0.163800,0.829000,-0.557200,-0.048180  
0.994600,0.074420,0.010250,0.331800,-0.939500,0.085320  
0.189900,-0.292200,-0.926300,0.239000,-0.178100,-0.954500  
-0.989200,0.074610,-0.012350,-0.816500,-0.250800,-0.520100  
0.208700,0.221100,0.565600,0.837600,-0.019280,0.545900  
0.404700,0.080140,-0.002638,0.002952,0.973500,-0.228800  
-0.043890,-0.115500,-0.412900,-0.672800,0.679900,-0.291600  
-0.525000,0.158200,0.303300,0.000000,-0.413200,0.910600  
-0.191700,-0.160900,0.657400,0.228400,-0.071910,0.970900  
-0.267300,-0.265700,-0.804700,-0.992800,0.095000,0.073630  
0.694500,-0.135400,0.304400,0.023350,-0.973600,0.227200  
0.186300,-0.376900,0.426500,0.562700,-0.531300,0.633300
```



ShapeNet(部件分割任务) 源数据 (txt 文件) 进行可视化:

```
1a6f615e8b1b5ae4dbbc9440457e303e.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
0.130150 -0.049690 0.249140 0.705900 -0.571200 0.418800 15.000000
0.213540 0.004660 0.213130 0.947700 0.319300 0.000000 13.000000
-0.177380 -0.250870 -0.179690 0.991200 -0.130300 0.023200 14.000000
0.109680 0.138340 0.222770 -0.854400 -0.116900 -0.506300 15.000000
-0.132090 0.038620 0.124130 -0.346000 0.938200 -0.000025 13.000000
-0.021210 -0.033030 0.213740 -0.048360 -0.000789 0.998800 13.000000
0.074840 -0.028000 0.218390 -0.048300 0.040230 0.998000 13.000000
0.077390 -0.021120 0.218510 -0.048100 0.080840 0.995600 13.000000
0.136580 -0.006940 0.221390 0.964700 -0.079060 0.251400 13.000000
0.007410 -0.037800 0.215120 -0.027640 -0.093780 0.995200 13.000000
-0.047030 -0.042480 0.212480 -0.045920 -0.135800 0.989700 13.000000
0.139740 -0.017430 0.221540 0.964700 -0.079060 0.251400 13.000000
```



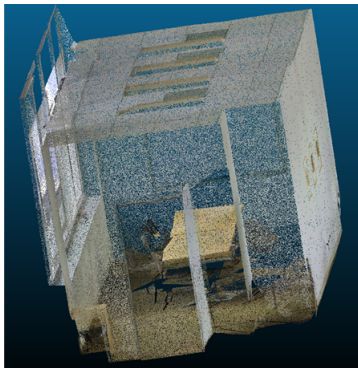
S3DIS(语义分割任务)

源数据 (txt 文件) 进行可视化:

conferenceRoom_1.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
-15.609 39.505 2.214 71 64 54  
-15.634 39.518 2.198 68 64 52  
-15.622 39.514 2.195 70 61 52  
-15.621 39.510 2.215 72 65 55  
-15.606 39.505 2.211 71 63 52  
-15.657 39.524 2.213 76 70 58  
-15.549 39.484 2.206 63 53 44  
-15.639 39.520 2.197 70 64 52  
-15.562 39.489 2.206 68 58 48  
-15.542 39.481 2.206 63 53 44  
-15.648 39.525 2.187 69 63 51  
-15.603 39.507 2.192 68 60 49  
-15.603 39.503 2.213 71 63 50  
-15.597 39.502 2.208 72 62 50  
-15.565 39.492 2.195 63 54 45  
-----
```



在 ModelNet40 数据集上进行分类：

方法	Accuracy(%)
Subvolume	89.2
MVCNN	90.1
pointnet (without normal)	90.4
pointnet (with normal)	91.8
pointnet2_ssg (without normal)	92.3
pointnet2_ssg (with normal)	92.5
pointnet2_msg (without normal)	91.9
pointnet2_msg (with normal)	93.1
pointnet2_mrg (without normal)	92.2
pointnet2_mrg (with normal)	92.7

比较 pointnet 和 pointnet++ 各个模型的显存占用:

方法	Accuracy(%)	显存占用(MB)
pointnet (without normal)	90.4	3053
pointnet (with normal)	91.8	3025
pointnet2_ssg (without normal)	92.3	4671
pointnet2_ssg (with normal)	92.5	4673
pointnet2_msg (without normal)	91.9	13671
pointnet2_msg (with normal)	93.1	13671
pointnet2_mrg (without normal)	92.2	10101
pointnet2_mrg (with normal)	92.7	10101

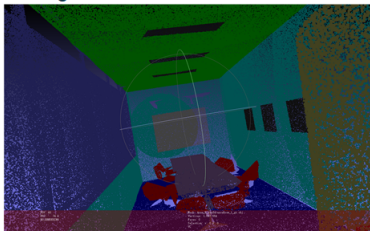
在 ShapeNet 数据集上进行部件分割：

方法	mIoU(%)
Yi	81.4
SSCNN	84.7
pointnet (without normal)	83.3
pointnet (with normal)	84.4
pointnet2_ssg (without normal)	85.1
pointnet2_ssg (with normal)	85.5
pointnet2_msg (without normal)	85.2
pointnet2_msg (with normal)	85.5

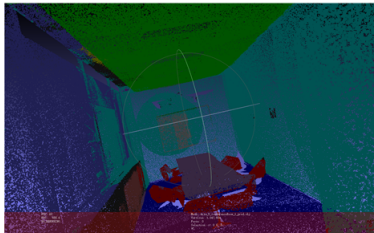
在 S3DIS 数据集上进行语义分割：

方法	Accuracy(%)	Class avg IoU(%)
pointnet	78.6	41.9
pointnet2_ssg	83.4	53.9
pointnet2_msg	83.0	63.2

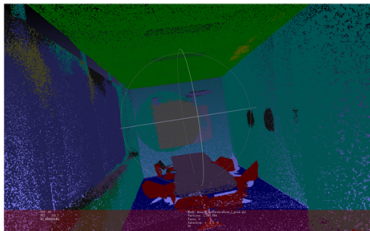
➤ ground truth



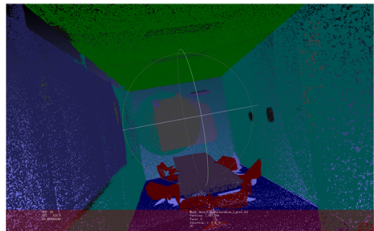
➤ pointnet



➤ pointnet2_msg



➤ pointnet2_ssg





- 1 介绍
- 2 PointNet
- 3 PointNet++
- 4 实验复现
- 5 MRG 方法实现**

MSG

为了提取多尺度的模式，一种简单有效的方法是针对每层的点云都采用多个尺寸的局部区域进行特征聚合

MRG

MSG 方法计算成本较高，替代方法 MRG 通过连接不同层特征，在减少计算成本的同时仍保留多尺度模式提取能力

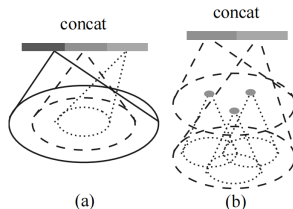


Figure 3: (a) Multi-scale grouping (MSG); (b) Multi-resolution grouping (MRG).

图: MSG 与 MRG

论文并没有给出 MRG 方法的源码, 我们根据论文中给出的网络结构实现了使用 MRG 的分类 pointnet++

右图是论文中的 pointnet++ 的网络结构

Branch 1: $SA(512, 0.2, [64, 64, 128]) \rightarrow SA(64, 0.4, [128, 128, 256])$

Branch 2: $SA(512, 0.4, [64, 128, 256])$ using $r = 0.4$ regions of original points

Branch 3: $SA(64, 128, 256, 512)$ using all original points.

Branch 4: $SA(256, 512, 1024)$.

图: MRG 分类网络结构

SA 为 Set Abstraction; (N,D) 表示输入点数为 N , 特征维度为 D ; C 表示连接操作

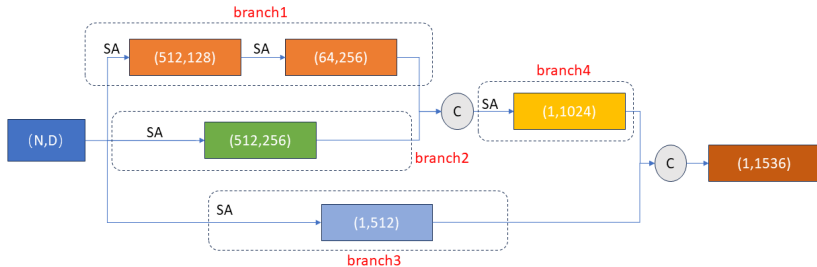


图: 特征提取过程示意图



谢谢!