

05 - Dynamic Graph CNN——一点云动态图卷积神经网络

2021年1月10日 13:11

文章地址: https://blog.csdn.net/weixin_39373480/article/details/88724518

其实这个文章之前已经读过了, 但是感觉不够深刻, 看看别人是怎么理解的吧。

简介: 主要提出一个EdgeConv层来获取局部特征, 解决PN没有的处理局部特征的问题。

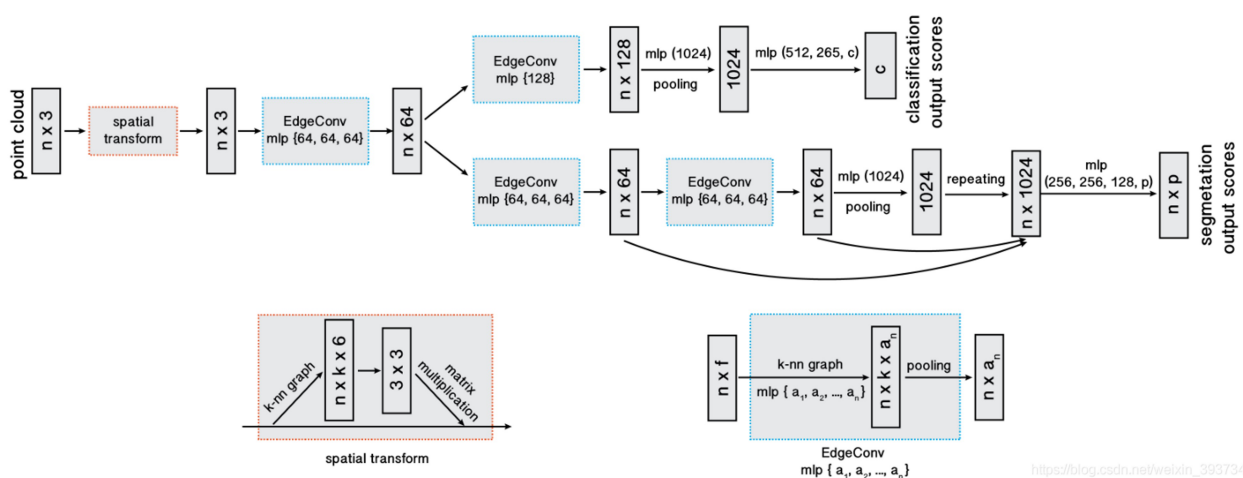
贡献:

- 提出一个新操作EdgeConv, 可以直接在点云上操作, 能在保证置换不变性的情况下获取局部几何信息;
- DGCNN可以在动态更新图的同时, 在语义上将点聚集起来。
- EdgeConv可以被集成, 嵌入到已知的点云处理框架中;

优缺点:

- 堆叠EC模块循环使用, 提取全局形状信息;
- 忽略了相邻点之间的向量方向, 最终还是损失了一部分局部几何信息 (这个是从哪里看出来的?)

网络架构:



和PN结构相似, 只是对其中的特征提取结构进行了修改, 变成了EdgeConv而已。

其中的EC的设计是文章的创新之处:

输入为 $n \times f$ 的张量,

通过MLP按照给定的每一层的输出的特征尺寸 $\{a_1, a_2, \dots, a_n\}$ 计算边缘特征,

最后沿着neighboring的边缘特征的方向做pooling操作生成新的 $n \times a_n$ 的张量。

EdgeConv层介绍

框架:

- 假设一个F维的点云有n个点，定义为: $X = \{x_1, \dots, x_n\}$ 属于 $R(F)$ 。最简单的情况下， $F=3$ ，也就是三维坐标。当然也可能包含额外的坐标。当然也可以包含额外的坐标，包含颜色，表面法线等等的信息。在一个DL中，后面的层都会接受前一层的输出，因此一般情况下，维度F也可以表示某一层的特征维度；
- 假设给定一个有向图 $G=(E, V)$ ，来表示点云的局部结构。在最简单的情况下，我们建立一个KNN的图G。假设距离点 x_i 最近的点 x_{j1}, \dots, x_{jk} 包含许多有向边缘 $(i, j_1), \dots, (i, j_k)$ ；
- 我们定义边缘特征为 $e_{ij} = h(x_i, x_j)$ ，其中 h 为F维度 * F维度 到F_维度，是一些使用一些可学习参数构成的非线性函数；
- 最后在EdgeConv操作上添加一个通道级的对称聚合操作

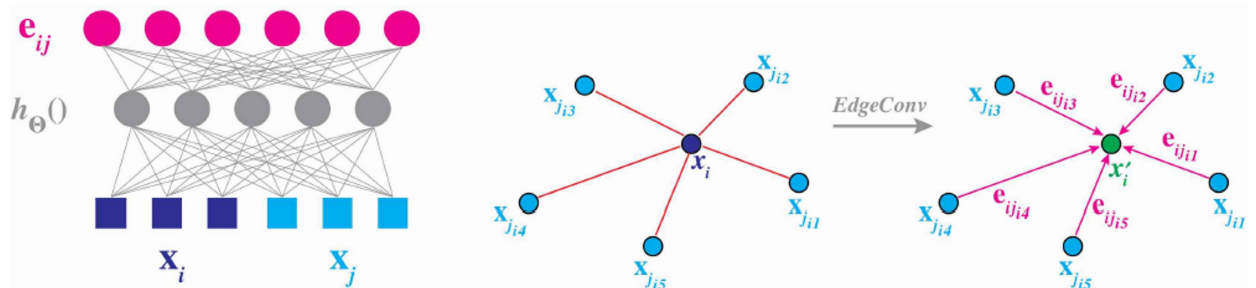


Figure 2. **Left:** An example of computing an edge feature, e_{ij} , from a point pair, x_i and x_j . In this example, $h_{\Theta}()$ is instantiated using a fully connected layer, and the learnable parameters are its associated weights and bias. **Right:** Visualize the EdgeConv operation. The output of EdgeConv is calculated by aggregating the edge features associated with all the edges emanating from each connected vertex.

图1：描述了一个关注点 x_i 和临近的一个点 x_j 的边缘特征的求解过程，图中的全连接操作只是一个示例。实际操作使用MLP。

图2：假设每一个关注点 x_i 周围 k 个邻近（here $k=5$ ）按照图例，需要按照图1的方式计算 x_i 和图中5个点的边缘特征。

图3：将计算出的5个点的边缘特征pooling到点数满意特征的尺寸上。

函数的选择：既考虑全局又考虑局部

$$h_{\vartheta}(x_i, x_j) = h_{\vartheta}(x_i, x_j - x_i)$$

动态CNN

- 每次重新计算每一层的图中的点在特征空间的最近邻点，是有用的。这也是动态图CNN与普通图CNN的不同之处；（想看下具体的代码是怎么操作的）

实现细节

- 整体的网络结构与PointNet的很类似，都使用了一个空间变换单元，最后计算了一个全局信息。
- 分类网络中包含了两个EdgeConv层，后面接上了一个池化操作和3个全连接层，然后得到分类结果。
- 分割网络使用了三个EdgeConv层，将全局特征和局部特征逐点拼接，最后每个点都会输出一个预测分数。
- 对于每个EdgeConv模块，每个都是共享边缘函数 $h_{\theta}(x_i, x_j) = h_{\theta}(x_i, x_j - x_i)$ ，而这个函数是对一个MLP实现的，聚合操作是max最大池化。
- KNN中的一个 k 值是一个超参，分类网络中 $k=20$ ，在分割网络中 k 为30

