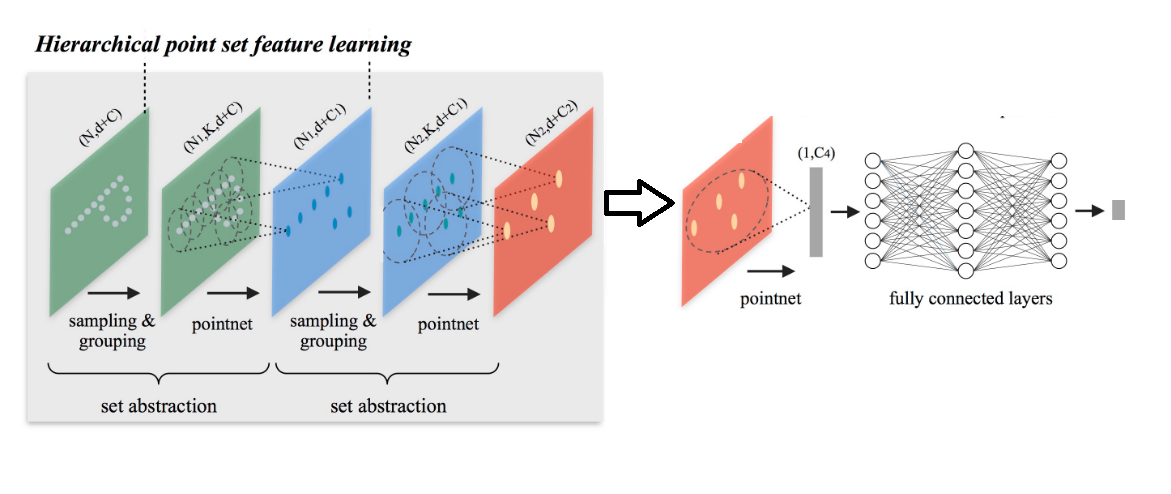
网络的创新点或者说实现的难点（优点）。

输入为点云,输出的label值其实包含非常大的噪声的,(这里说的噪声是指输出值其实并不完全依赖输入的信息,还有额外的信息,但是输入里并没有,且整体类似混沌系统.) 输入的点云由于点与点之间也是远比图像像素点之间关系复杂. 所以,在这样情况下,给模型训练带来很大的困难.

如何让模型自动摒弃这些噪声信号,学习到输出label里依赖点云结构的信息. 这是成功的关键. 经过对数据滑动窗口发现, 虽然单时刻的label之间差异变化大,但是连续一段时间窗口内的label的均值却比较固定. 在原始的pointnet++模型, 模型是单独孤立去处理不同的样本,没有时间窗的概念. 所以, 在使用pointnet++模型时需要加入一个时间窗时序关系.每一批次的样本输入时候, 同一批次的样本保证有先后的时序关系,同时在训练loss上也是整体考虑一个批次样本的误差. 从而学习出来的模型,能够抛弃噪声部分,比较准确地预测滑动窗内的均值.

网络结构

模型结构:

其中

N1 : 采用中心点数目

radius: 邻域的半径.

K: 在邻域内抽样的点数

d: 每个点的位置维度,3维空间的点,d就是3

C: 每个点的属性维度

在我们模型里,采用2个set abstraction

第一个set abstraction

N1 = 512

C = 原始数据输入的维度

领域半径: 分别有0.1, 0.2, 0.4

对应这3个领域内分别抽样K = 16, 32, 128

这3个领域后面分别接了个 mlp 的参数是 :

mlp1 = [32, 32, 64]

mlp2 = [64, 64, 128]

mlp3 = [64, 96, 128]

以mlp1为例,时对应邻域半径为0.1,抽样16的样本, mlp有3层,前向层节点数分别32,32,64. 其他的同理.

第二个set abstraction

N1 = 128

领域半径: 分别有 0.2, 0.4, 0.8

对应这3个领域内分别抽样K = 32, 64, 128

输入的属性维度C=320

这3个领域后面分别接了个 mlp 的参数是 :

mlp1 = [64, 64, 128]

mlp2 = [128, 128, 256]

mlp3 = [128, 128, 256]

最后接了一个全局pointnet

全局pointnet,意味半径是大到足够包含所有的点,K也是所有的点数

属性维度640 + 3

mlp为256, 512, 1024

最后再接一个输出mlp

[1024,512,256,1]

两个数据集对应的模型的差异仅仅只是输入数据属性维度不一样.其他的参数一样.

损失函数, 对一个batch内样本的预测值和label计算mse后再在batch间平均. 训练epoch在60到100左右, 优化器采用adam‘

在原始的pointnet++模型, 模型是单独孤立去处理不同的样本,没有时间窗的概念. 所以, 在使用pointnet++模型时需要加入一个时间窗时序关系.每一批次的样本输入时候, 同一批次的样本保证有先后的时序关系,同时在训练loss上也是整体考虑一个批次样本的误差. 从而学习出来的模型,能够抛弃噪声部分,比较准确地预测滑动窗内的均值.

时间窗体现在batch内， 也就是按一个滑动窗的方式去取一个批次的数据。每个时刻的点云的所有点看成一个样本，而不是点云内部某个点看成一个样本。然后通过batch机制把一个时间窗的内的样本都一起读进去处理。

参数是调参时候，在经验值的基础上微调下得到的。 整体上，mlp的节点数越多，模型复杂性越高，能力越强。 抽样数K也是，但是抽样数设置太大了，很容易过拟合。 领域半径则比较微妙，这个和样本的属性取值有密切关系，设置起来也比较困难。为了减低这个多样性，往往对点的属性归一化到[-1，1]之间，这时候用经验值0.1,0.2,0.4一般问题不大。 一般是取的意义是这个半径内的点有较强相关性。模型的第一层往往只是分析到局部细节特征，所以领域取比较小，在这个领域里要求点的特征是高度相关的。 模型的后面几层，模型会逐步抽象语义，能识别到些组件层面的概念，所以领域就相对大很多。但是为什么是0.1,0.2,0.4这个呈2倍关系，这个是经验的。