

# ResNet总结

---

- 什么是退化问题？

随着网络逐渐变深，准确率达到饱和后迅速下降，本质是一个优化问题

- 什么实验现象表明是退化问题而不是过拟合问题？

在增加层数后，训练集的准确率也出现退化问题

- 如何理解残差连接？为什么残差连接可以解决退化问题？

拟合输出与输入的残差。

为了构建一种恒等映射，使得深层网络的训练误差不会大于对应的浅层网络。但让多个非线性层去拟合这种恒等映射是很难优化的，利用残差学习，如果恒等映射是最优的，可以通过权重重置0来得到，如果最优函数和恒等映射接近（恒等映射加一个扰动），残差学习会在恒等映射的基础上很容易的学习到这个扰动，比直接学习最优函数本身要容易

- 在网络结构上如何实现残差连接？（分维度不变和维度变化两种情况）

维度不变时使用恒等shortcut，维度变化时通过零填充或者投影shortcut（1\*1conv）

- 从参数量和计算量上，残差连接有什么优势？

不会引入额外的参数，计算量很小

- bottleneck结构是怎样的？其中1x1卷积的作用是什么？为什么要设计 bottleneck结构？

1x1→3x3→1x1

先降维后升维，减少3\*3conv维度从而减少计算量

- BN和conv，relu的位置关系是什么？

conv-BN-relu

- BN是为了解决什么问题？为什么可以解决这个问题？

要解决的问题：内部协变量偏移（internal covariate shift） 定义：指在训练过程中每一层的输入分布随着前一层的参数变化而变

每层的输入分布发生变化有两个部分：一是随着训练增加神经网络中某些神经元总是很活跃（这部分神经元之间依赖性很强）而某些却很难激活或是激活值很低，如果总是那一部分神经元激活那就限制了模型的泛化能力。二是，深层次的神经元的输入相比较于前面浅层的神经元更难激活，因为随着前面分布偏移的累积使得后面神经元更难激活，例如输入落入sigmoid激活函数的两端导致梯度消失。

为了解决神经元输入分布偏移的情况，对每个神经元的带权输入（一个mini-batch上）进行标准化，把发生偏移的带权输入拉回到同一分布上。首先这样做解决了上述中第一个分布发生变化的问题，使得每个神经元都会适当的激活，同时标准化使得带权输入分布在0附近更容易激活神经元解决梯度消失问题。但只做标准化容易把之前学习到的特征丢弃掉，这时候就引入了伸缩因子和平移因子（这两个参数是需要学习的），相当于对标准化后的带权输入又做了个线性变换。这样使得所有神经元带权输入都有一个同一的分布，但又保持了相对差异。

由此可以说BN让神经网络中各个神经元的训练更加均衡，也就避免不同参数取值相差很大（想象多项式回归，如果参数取值很大或很小那学习出的曲线震荡幅度很大就容易过拟合而我们希望学习的曲线更平滑）。同时由于每个神经元保持了相对差异，这样不同神经元之间的依赖性很低，模型也就没有那么复杂了就一定程度避免了过拟合。

- 训练和测试时，BN的计算过程分别是怎样的？

在训练时，计算一个batch的均值和标准差，做归一化，然后进行scale和shift，同时学习scale和shift参数

在测试时，利用训练集的均值和标准差做归一化，并进行scale和shift，保留部分特征

- 有哪些训练和测试时计算方式不同的层？（有两个）

BN、dropout

- 加入BN后，网络结构和训练超参数需要做出哪些调整

增大学习率、去掉dropout、减少weight decay、加快学习率下降的速度