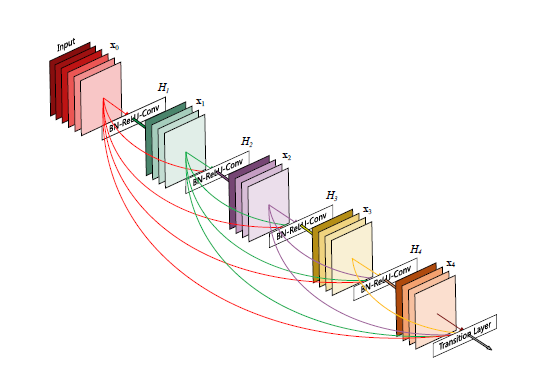
Densenet的基本结构

DenseNet 是一种具有密集连接（dense connection）的卷积神经网络。在该网络中，任何两层之间都有直接的连接，该网络连接类似于一个握手游戏，在一个组（深度网络模型）中的每个成员（feature layers）都必须要进行一次握手（connection），网络每一层的输入都是前面所有层输出的并集，而该层所学习的特征图也会被直接传给其后面所有层作为输入。下面展示DenseNet的一个示意图。



Why DensNet?

Densenet的实质就是在前传的时候将当前层（假设为第N层）与前面的（N-1）层建立连接，这样乍看起来并没有什么好处，反而让层与层之间的连接变多了。但是这样却提高了数据流动的效果。传统卷积网络证明了网络的性能随深度的增加会有一定的提升，但是深度卷积神经网络又引来了一个新的问题，越深的网络越难训练，而且随着深度的增加会出现梯度消失的问题。梯度消失会造成网络前面的层的参数不能得到训练和优化。所以当网络加深时又会面临这样的问题。而Densenet 通过连接前面的层让每一层都与loss function直接的联系(direct access)。这样就在一定程度上避免了因为网络深度过深而造成的在模型反传优化的时候存在的梯度消失不能优化的问题。

Densenet另一个极大的好处就是feature reuse，传统的卷积神经网络就是feed-forward前传结构，上一层的输出作为下一层的输入。随着网络的不断加深，网络前面的feature layers对最终结果不起作用，而DenseNet 通过skip connenction 将不同阶段的feature map通过级联（concatenate）作为当前输入。这样就直接实现了feature reuse。另外这样还可以起到利用多尺度下的特征的效果（utilize multi-level features in CNNs）。

下面我们来讲解一些DenseNet和ResNet的区别。

ResNet



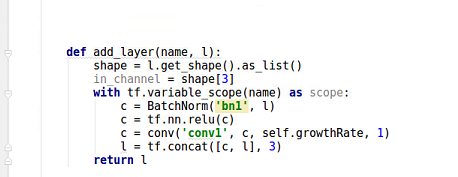
DenseNet



代表非线性函数（卷积操作（conv）,非线性激活（leaky ReLU）,BatchNorm等）。从公式上我们可以看到DenseNet和ResNet最大的区别就是ResNet只在相邻的两个层之间建立连接和提供数据流动的接口。而DensNet则是将当前层（第层）和前面的-1层都进行连接。换言之，当前层的输入和前面的每一层的输出都有直接的关系。

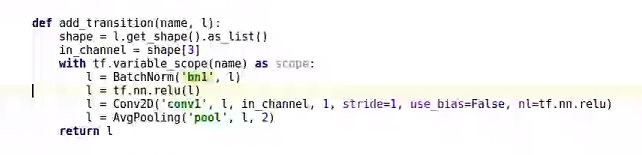
Growth rate（k）

既然DenseNet网络让每一层都有连接，那么每一层的特征图（feature maps）将会比原始模型增加很多，为了控制网络参数量的大小，DenseNet通过一个Growth rate的值进行控制，通俗的来讲就是指下一层比当前层多出的特征图的通道数(channels)。下面代码上是这样实现的。

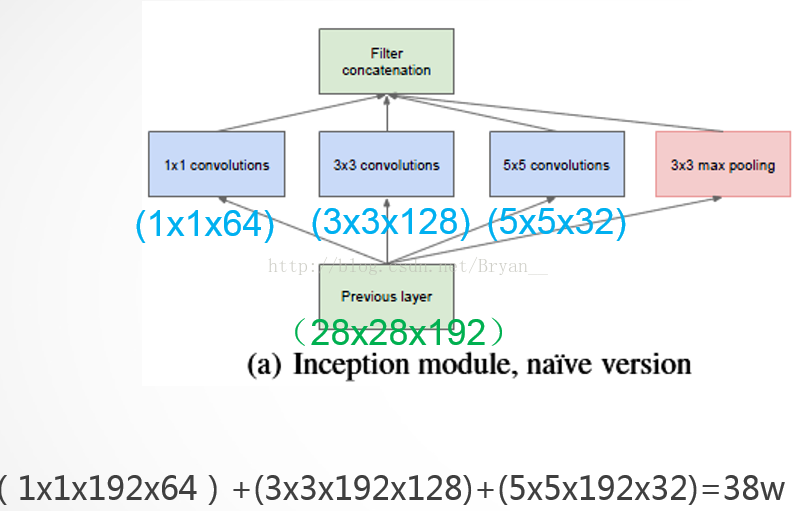


Compression

为了减少网络的参数量和提高网络的紧凑性。DenseNet中采用了transition layer来控制网络参数，设置一个额外的参数， 在这里 起到了一个random dropout的作用，这样既节省了参数，又可以起到防止模型过拟合的问题。



在使用密集连接时候的网络显得比较冗余，这样会不会增大参数量呢，DenseNet很好的处理了这个问题。它将每一层都设计的特别窄，在Dense Block中的每一个单元实际上都是一个 bottleneck layer，其中包括一个 1x1 conv和一个3x3 conv。这个创新点最早在GoogleNet inception 结构中被采用。



通过这样的网络结构来降低参数量。