9.DenseNet

DenseNet是2017年CVPR的Best paper，论文地址，[Densely Connected Convolutional Networks]

随着CNN网络层数的不断增加，gradient vanishing和model degradation问题出现在了人们面前, BatchNormalization的广泛使用在一定程度上缓解了gradient vanishing的问题，而ResNet和Highway Networks通过构造恒等映射设置旁路，进一步减少了gradient vanishing和model degradation的产生。

本文亮点：
1.相比ResNet拥有更少的参数数量；
2.旁路（Bypass）加强了特征的重用；
3.网络易于训练，并具有一定的正则效果；
4.缓解了Gradient Vanishing和Model Degradation的问题

9.1 Dense Connectivity

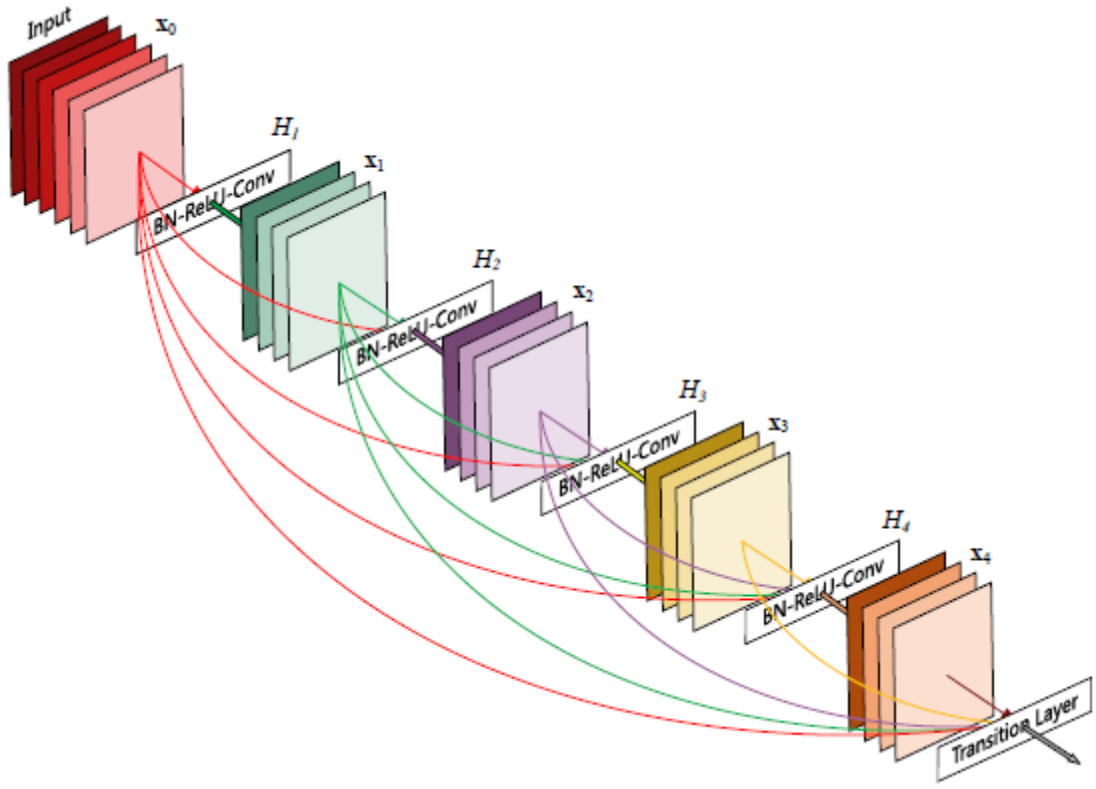

图33 Dense block

ResNet的一个最主要的优势便是梯度可以流经恒等函数来到达靠前的层，但恒等映射和非线性变换输出的叠加方式是相加，这在一定程度上破坏了网络中的信息流。为了进一步优化信息流的传播，DenseNet提出了图33的网络结构。

8.2 Pooling Layers

由于在DenseNet中需要对不同层的feature map进行cat操作，所以需要不同层的feature map保持相同的feature size，这就限制了网络中Down sampling的实现。为了使用Down sampling，作者将DenseNet分为多个Denseblock，如下图所示：

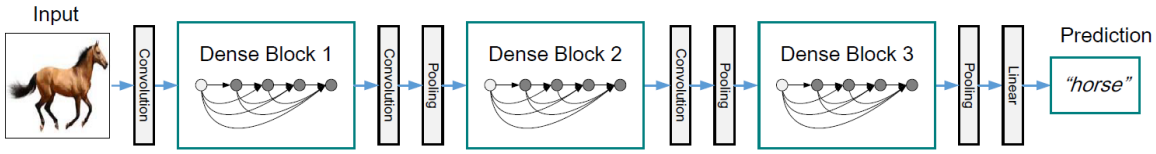

图34 DenseNet

在不同Denseblock之间设置transition layers实现Downsampling，transition layer由BN+conv（1×1）+2×2 average-pooling组成。

8.3 Growth Rate

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

图35 DenseNet网络结构

在同一个Denseblock中的每一层都与之前所有层相关联，如果我们把feature看作是一个Denseblock的全局状态，那么每一层的训练目标便是通过现有的全局状态，判断需要添加给全局状态的更新值。因而每个网络层输出的特征图数量K又称为Growth rate，同样决定着每一层需要给全局状态更新的信息的多少。

8.4 Bottleneck Layers

虽然DenseNet接受较少的k，也就是feature map的数量作为输出，但由于不同层feature map之间由cat操作组合在一起，最终仍然会是feature map的channel较大而成为网络的负担。作者在这里使用1×1 Conv(Bottleneck)作为特征降维的方法来降低channel数量，以提高计算效率。经过改善后的非线性变换变为BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3)，使用Bottleneck layers的DenseNet被作者称为DenseNet-B。在实验中，作者使用1×1卷积生成channel数量为4k的feature map。

8.5 Compression

为了进一步优化模型的简洁性，我们同样可以在transition layer中降低feature map的数量。若一个Denseblock中包含m个feature maps，那么我们使其输出连接的transition layer层生成⌊θm⌋个输出feature map。其中θ为Compression factor， 当θ=1时，transition layer将保留原feature维度不变。

pytorch实现