目录

[AlexNet 2](#_Toc66357955)

[ZFNet 2](#_Toc66357956)

[VGGNet 3](#_Toc66357957)

[GoogLeNet 4](#_Toc66357958)

[ResNet 6](#_Toc66357959)

[其他网络 7](#_Toc66357960)

[Network in Network (NiN) [Lin et al. 2014] 7](#_Toc66357961)

[Identity Mappings in Deep Residual Networks [He et al. 2016] 8](#_Toc66357962)

[Wide Residual Networks [Zagoruyko et al. 2016] 8](#_Toc66357963)

[Aggregated Residual Transformations for Deep Neural Networks (ResNeXt) [Xie et al. 2016] 8](#_Toc66357964)

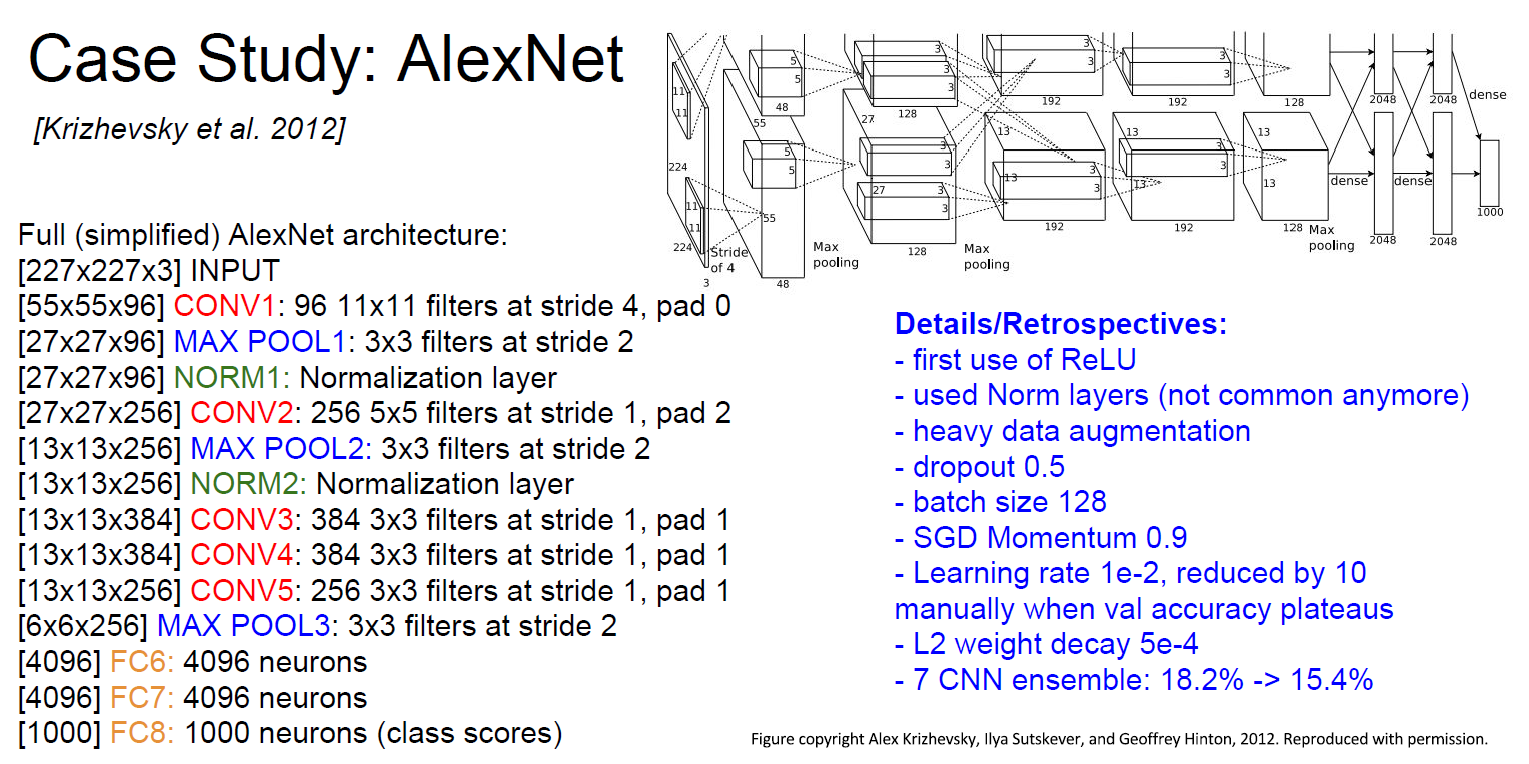
[Deep Networks with Stochastic Depth [Huang et al. 2016] 9](#_Toc66357965)

[FractalNet: Ultra-Deep Neural Networks without Residuals [Larsson et al. 2017] 9](#_Toc66357966)

[Densely Connected Convolutional Networks [Huang et al. 2017] 10](#_Toc66357967)

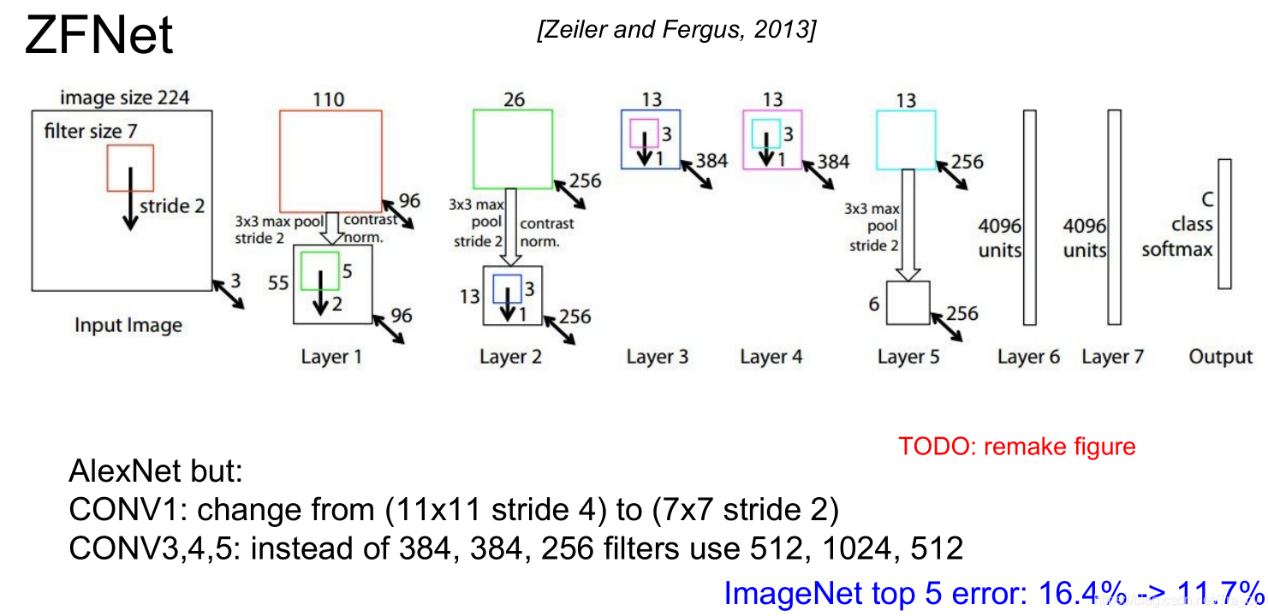
[SqueezeNet: AlexNet-level Accuracy With 50x Fewer Parameters and <0.5Mb Model Size [Iandola et al. 2017] 10](#_Toc66357968)

# AlexNet



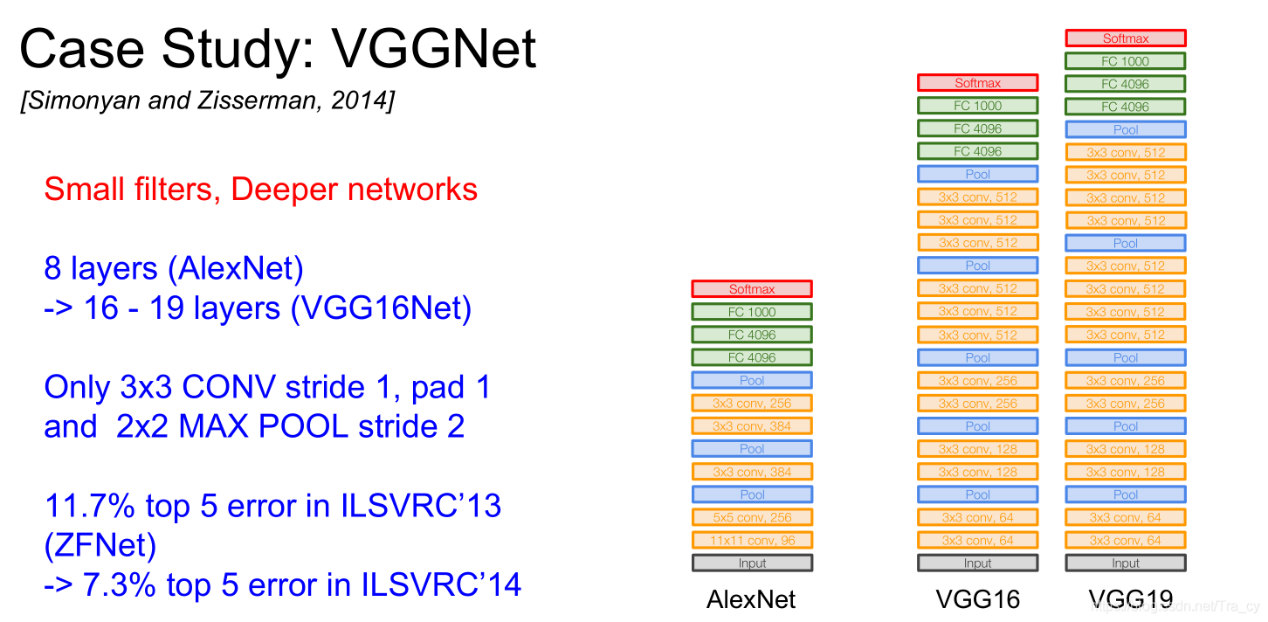
输入：227\*227\*3的图像矩阵。  
第一层（CONV1）：96个步长为4，大小为11\*11的卷积核，经过第一层后的输出大小为55\*55\*96，参数个数为11\*11\*3\*96。 提示：（227-11）/4+1=55  
第二层（POOL1）：3\*3的卷积核，步长为2，输出大小为27\*27\*96，池化层没有参数（仅有运行规则，没有参数）。 提示：（55-3）/2+1=27

# ZFNet

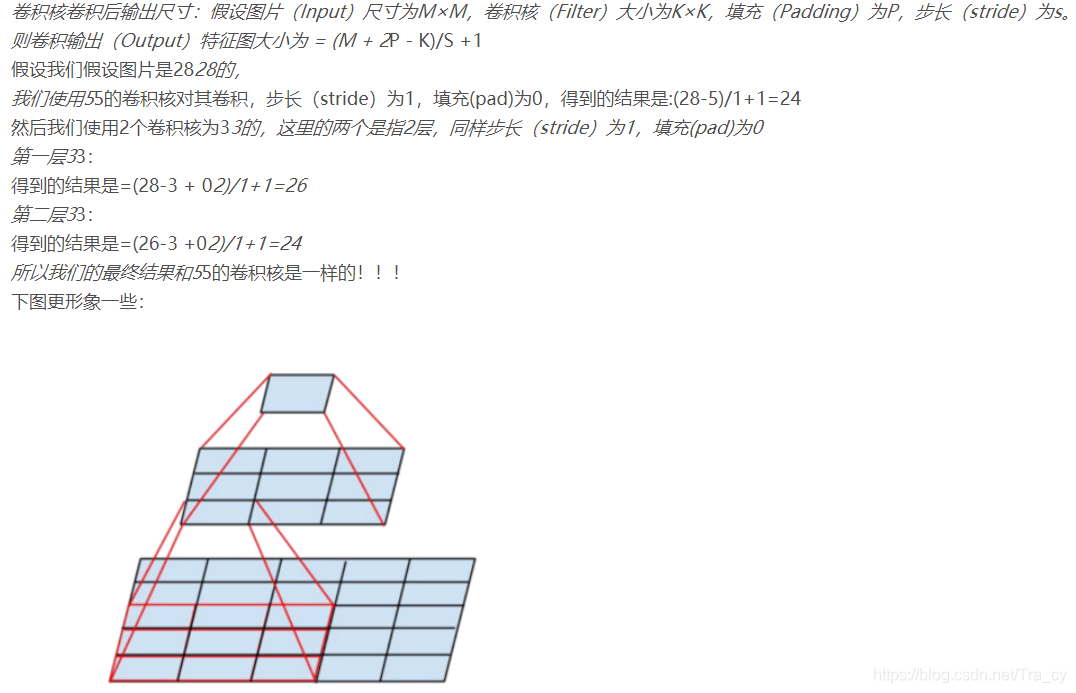


和AlexNet有相同的层数，相同的基本结构，优化了超参。如步长、卷积核数量等超参做了一些改进，降低了错误率，但是基本的思想是一致的。

# VGGNet



VGG的特点是：小卷积核，深网络。  
使用小卷积核的原因：保证网络精度的情况下，减少参数，两个3x3的堆叠卷基层的有限感受野是5x5；三个3x3的堆叠卷基层的感受野是7x7，原因如下：



使用3个3x3卷积核来代替一个7x7的卷积核的优势：

1 相同的感受野

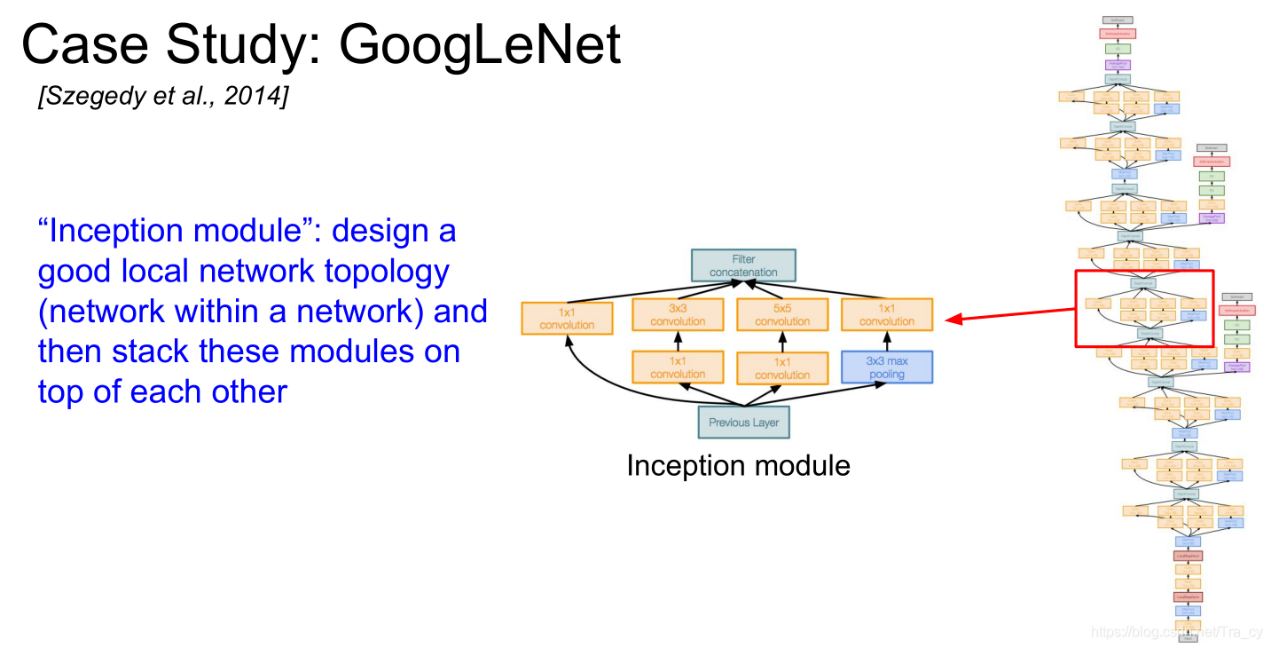
2 虽然网络更深，但是有更多的非线性

3 更少的参数：3\*（3^2\*C\*2）VS 7^2\*C^2 C为通道数

# GoogLeNet

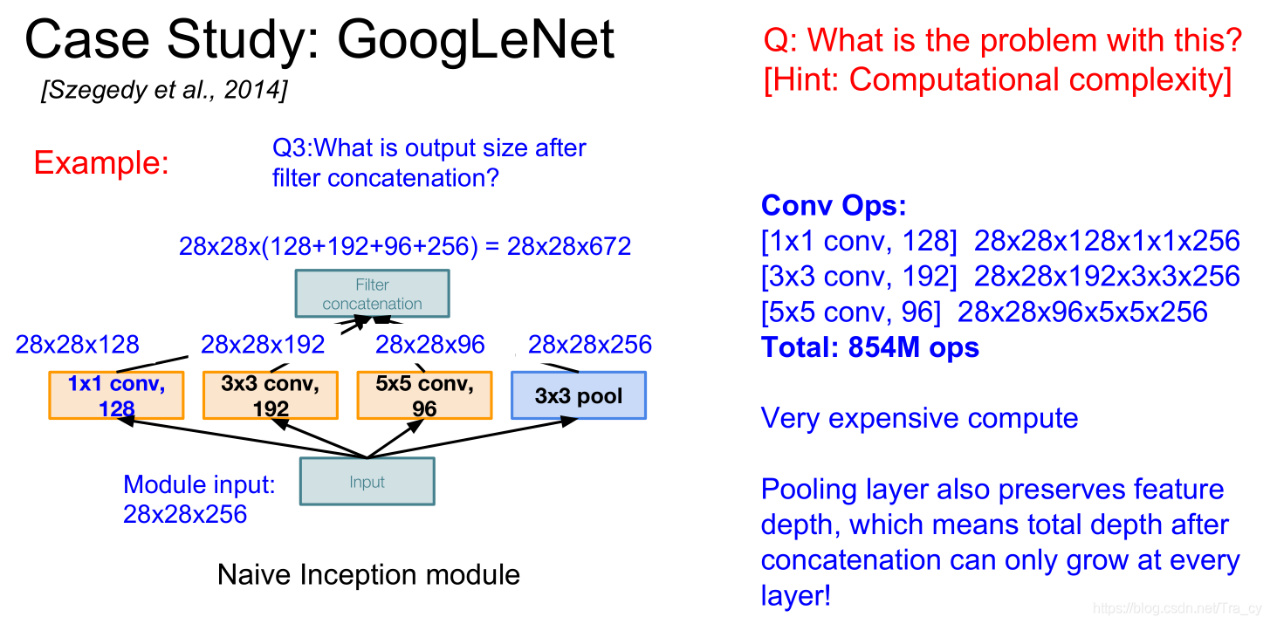
特点：深层的网络、计算有效性、没有全连接层、参数少（500万，比AlexNet少了12倍）、拥有一个inception layer。

“Inception module"：设计一个好的网络拓扑，然后将这些模块彼此堆叠。

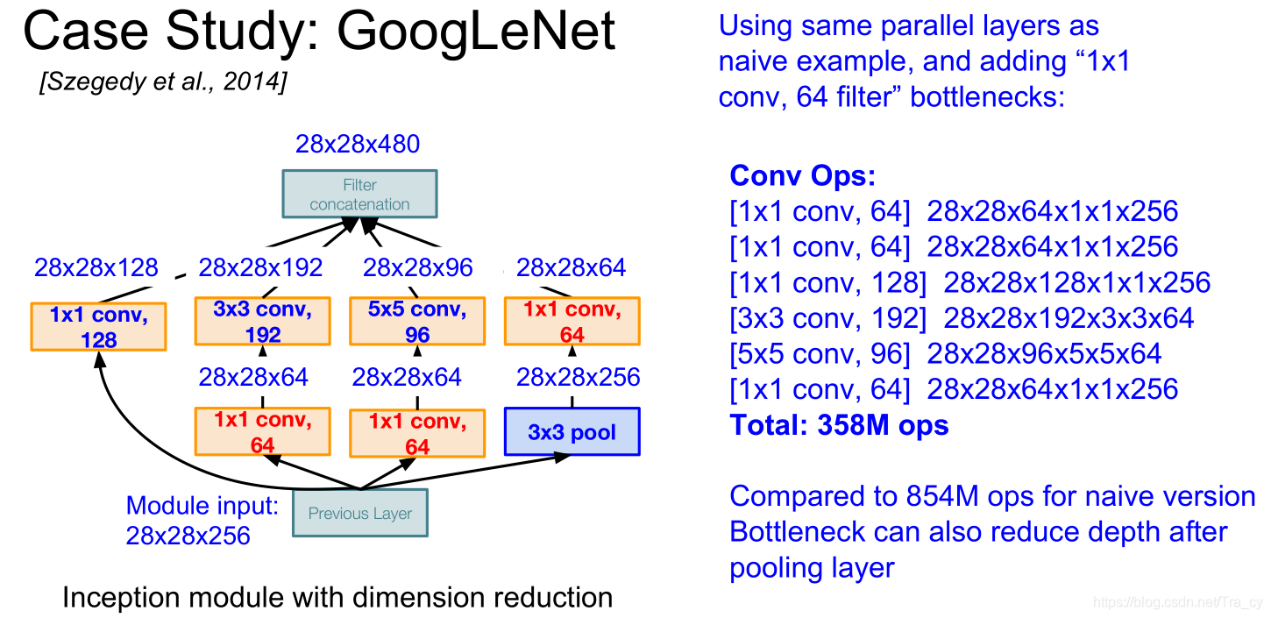


工作原理：对本层的输入并行地使用不同的filter操作，得到不同的输出，再把这些输出合在一起，最后得到一个张量输出，这个输出再作为下一层的输入。

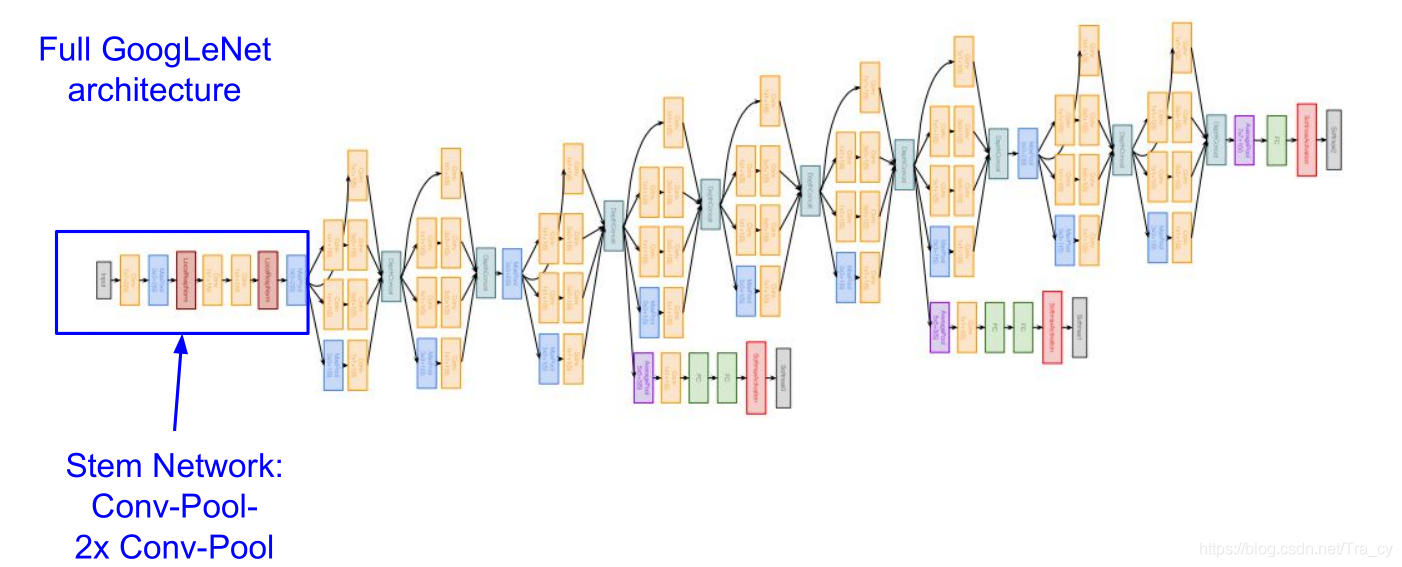
GoogLeNet存在的问题是计算复杂性的问题：例如下面的整个过程，输入是28\*28\*256大小，输出是28\*28\*672，计算成本很大，而且随着网络深度的增加，这个计算成本一直在增大。（在适当部分做0填充，以保持28x28 ->28x28）



针对上述问题，GoogLeNet的解决方法是增加一层bottleneck层，在卷积运算之前降低特征图的维度。如下图，在3\*3和5\*5的卷积层之前，以前池化层之后，都增加了一层bottleneck层，降低了深度。



GoogLeNet的整个架构：

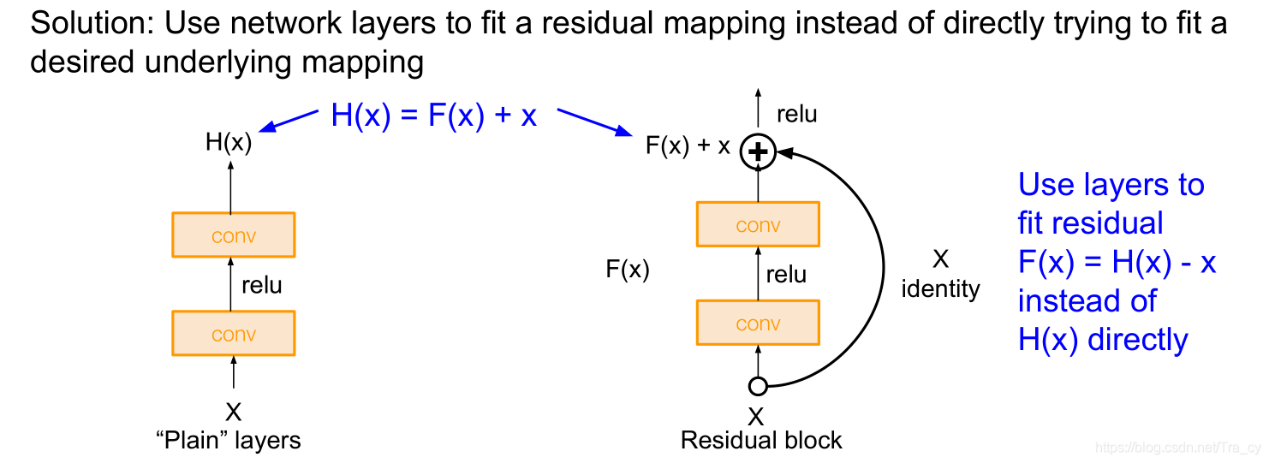


# ResNet

<http://zh.d2l.ai/chapter_convolutional-neural-networks/resnet.html> 多参考这篇文章！！！

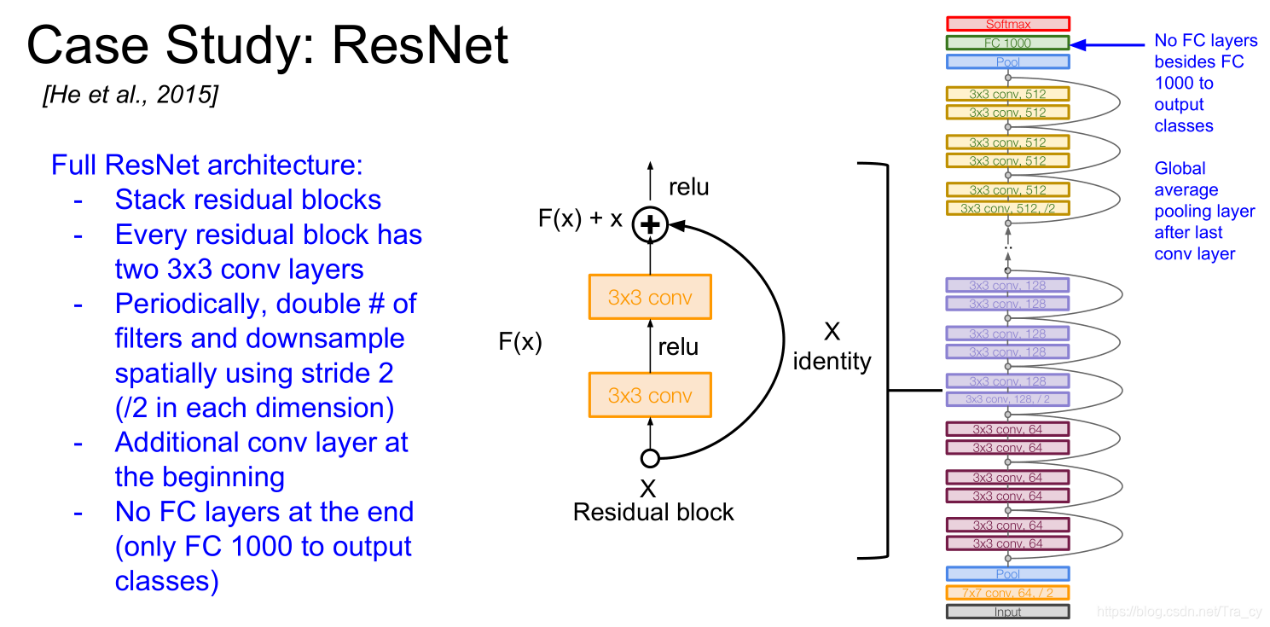
残差网络是一个使用残差层连接的非常深的网络。当我们在卷积神经网络上堆叠越来越多的层时，会出现深层的网络不管在训练集还是测试集上都不如一些浅层的网络，残差网络的创造者提出这是一个优化问题，相比于浅的网络，更深的网络更难优化。然而，一个更深的网络至少要能达到一个较浅网络的效果，我们可以构造一个恒等映射来证实。

于是提出了解决方案：让网络的每一层都尝试学习一些所需函数的底层映射，使用残差网络去拟合残差映射，而不是直接去拟合底层映射，简单地说，就是将H(x)进行分解，如果我们直接学习H(x)可能会导致很深的网络，而学习F(x)可能会比较容易实现，这里的F(x)就是指的残差。

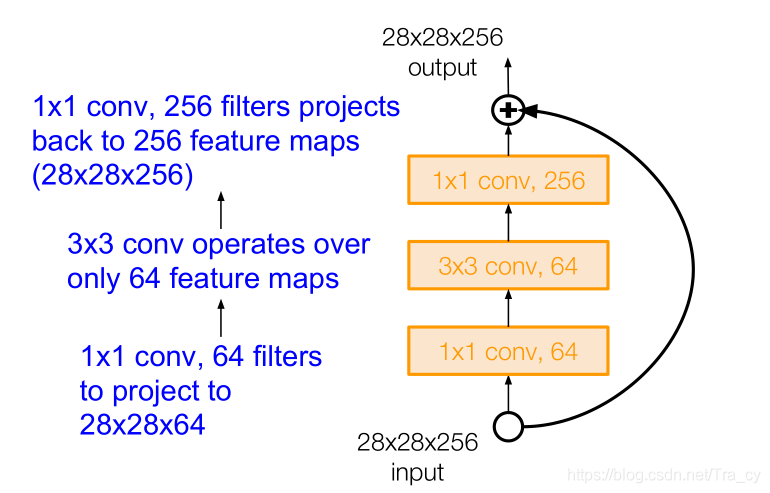


完整的残差网络如下图所示：

* 将所有这些残差块堆叠在一起组成一个很深的网络
* 每个残差块有两个3\*3的卷积层
* 使用两倍数量的卷积核，步长为2进行下采样
* 网络输入的地方有一个7\*7的卷积层
* 网络在最后有一个全局的池化层，它对所有的东西在空间上取平均，网络没有全连接层，只有一个FC1000作为输出。



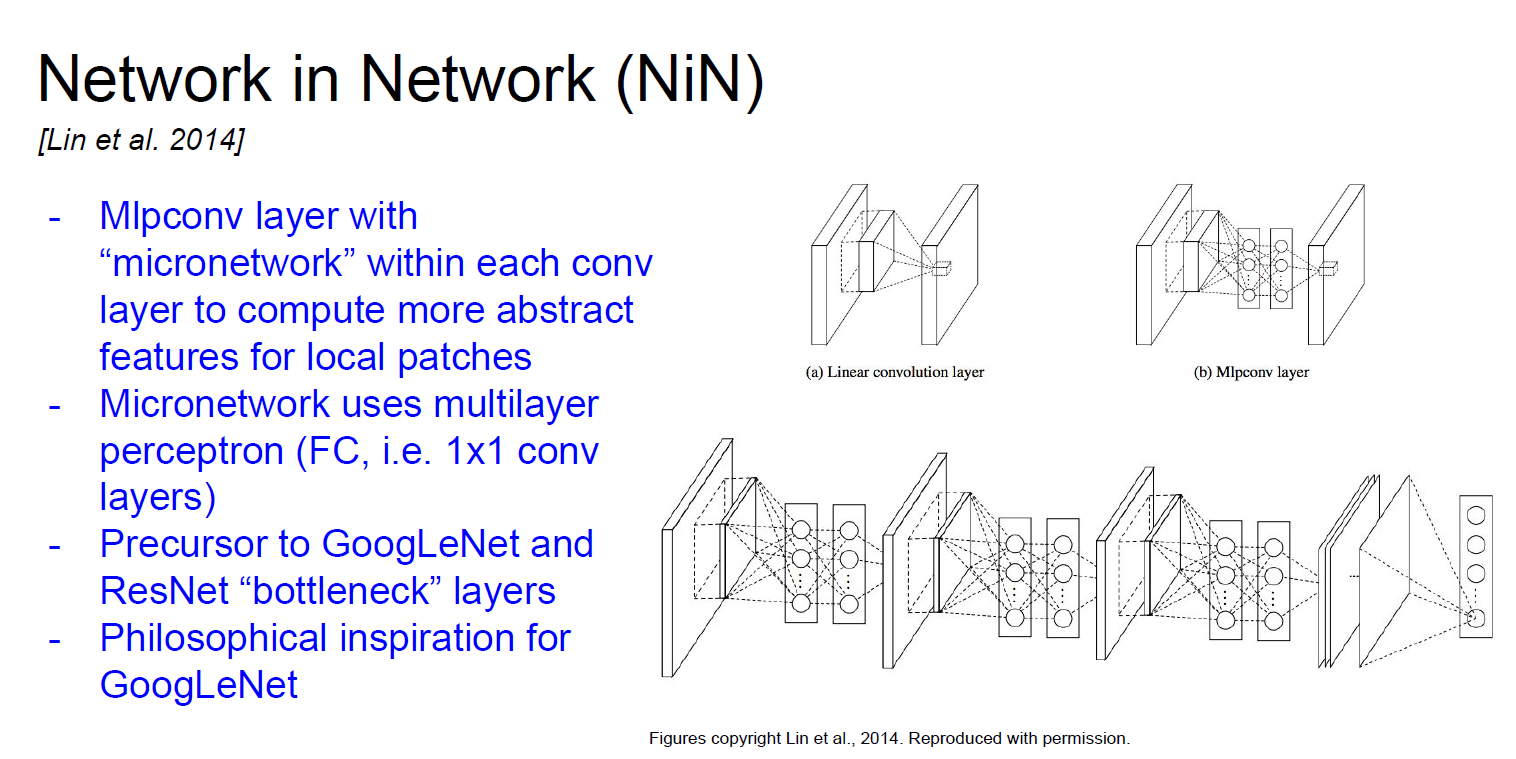
对于较深的网络，也可以使用和GoogLeNet一样的”bottleneck“来提高计算效率，也就是在每一层的输入后都加上一层1\*1的卷积层来降低深度



# 其他网络

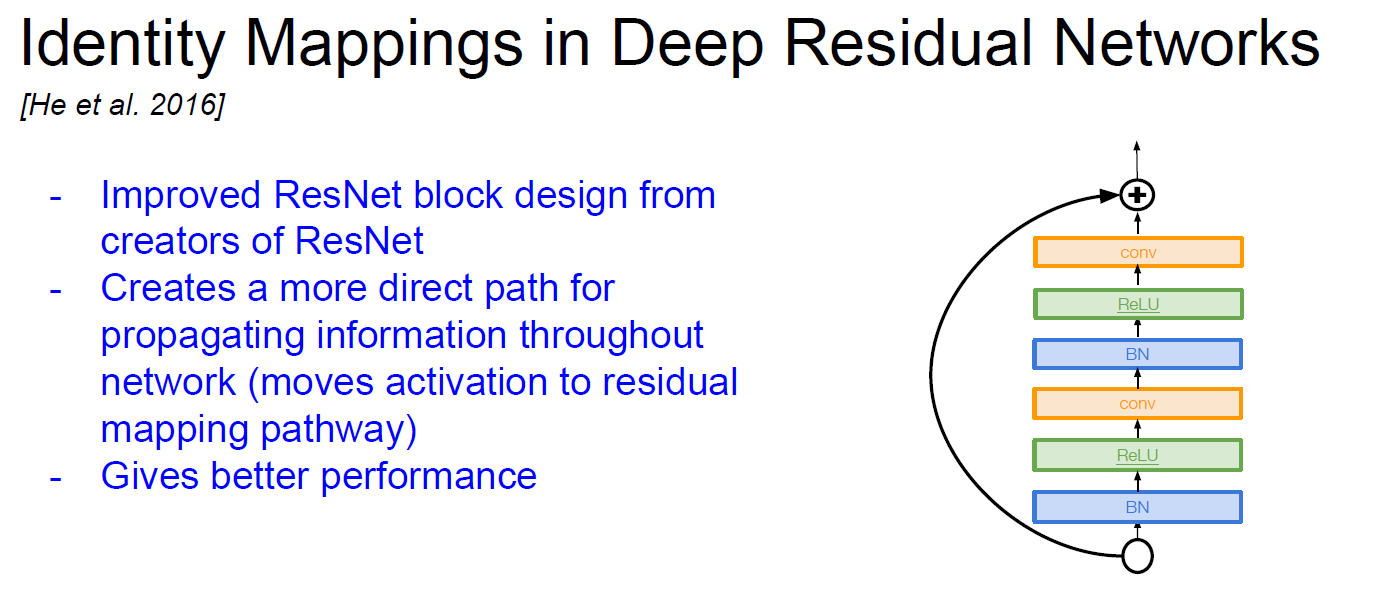
## Network in Network (NiN) [Lin et al. 2014]

在GoogLeNet和ResNet“bottleneck”层的前身，在conv层之后使用微网络获得更抽象的特征。



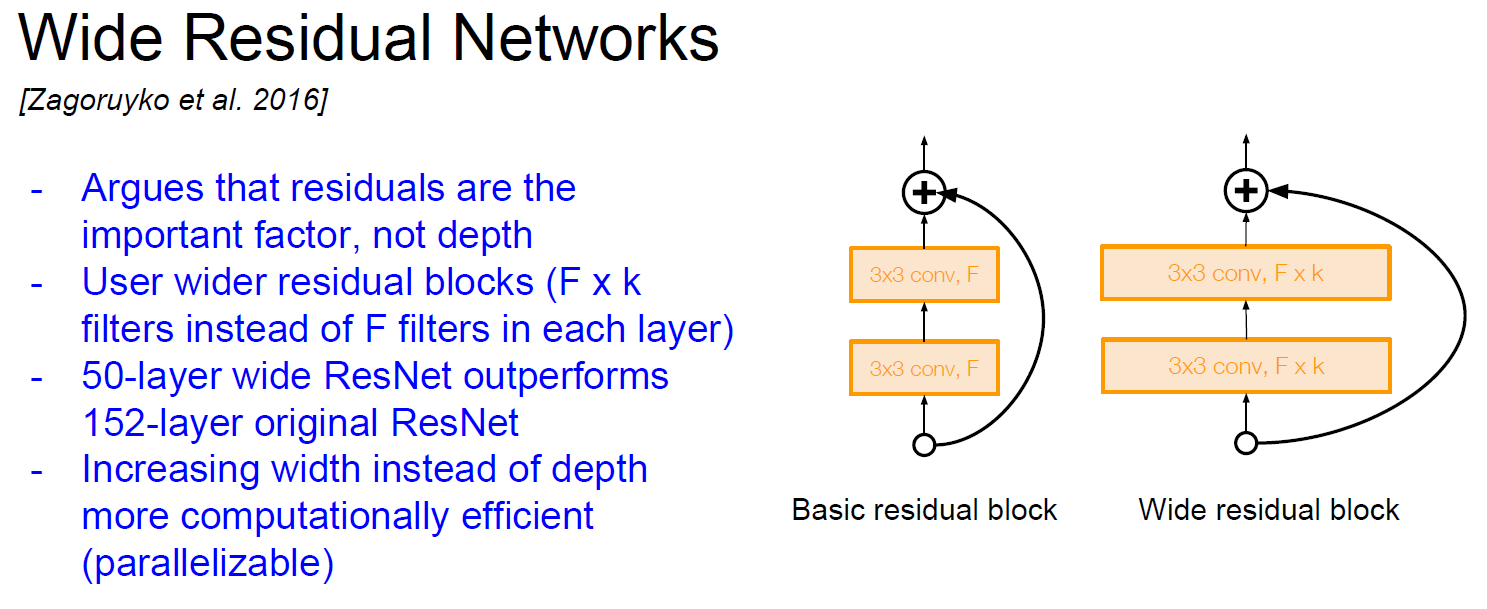
## Identity Mappings in Deep Residual Networks [He et al. 2016]

改进ResNet块的设计，调整了ResNet块的层数，取得更好的表现



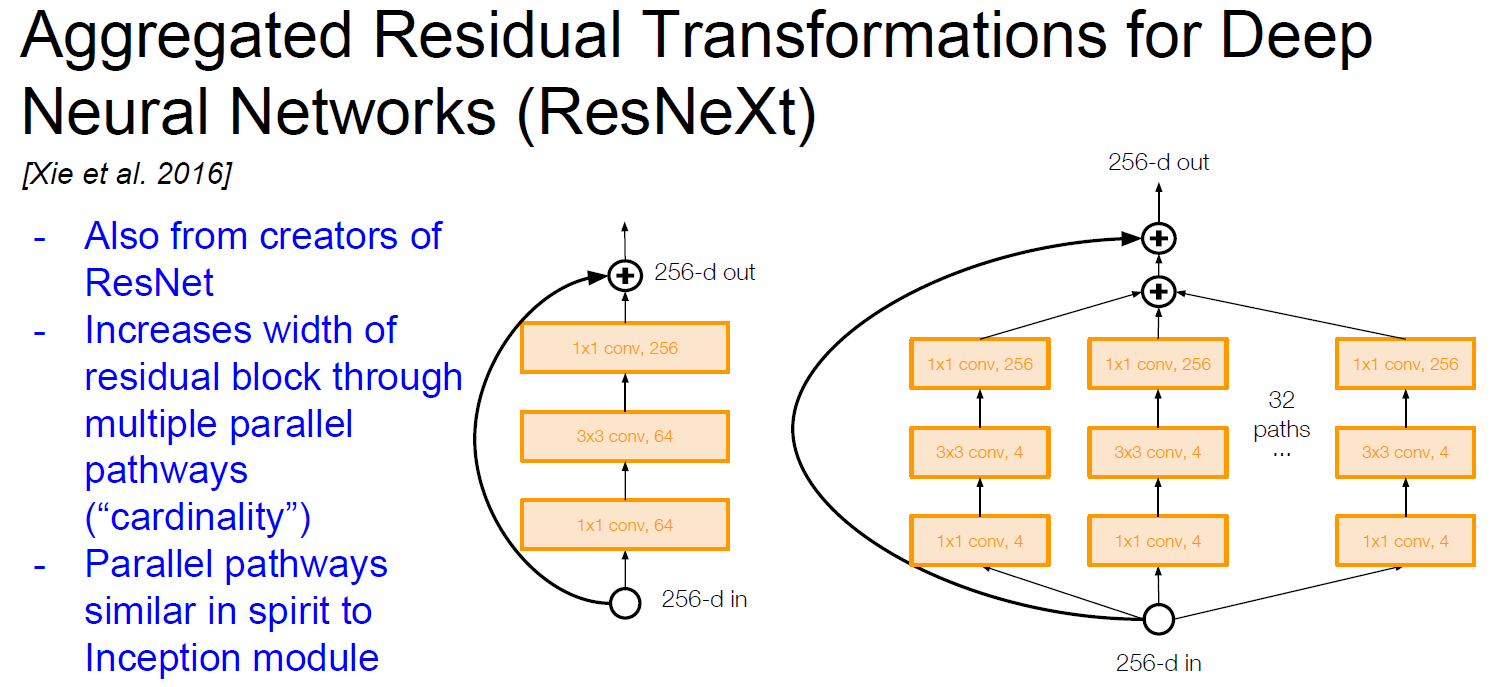
## Wide Residual Networks [Zagoruyko et al. 2016]

强调残差部分而不是深度，因此在残差块中有更多的过滤器，使网络更浅，从而更并行化。



## Aggregated Residual Transformations for Deep Neural Networks (ResNeXt) [Xie et al. 2016]

通过多个平行路径来增加残差块的宽度， 平行路径类似于Inception module

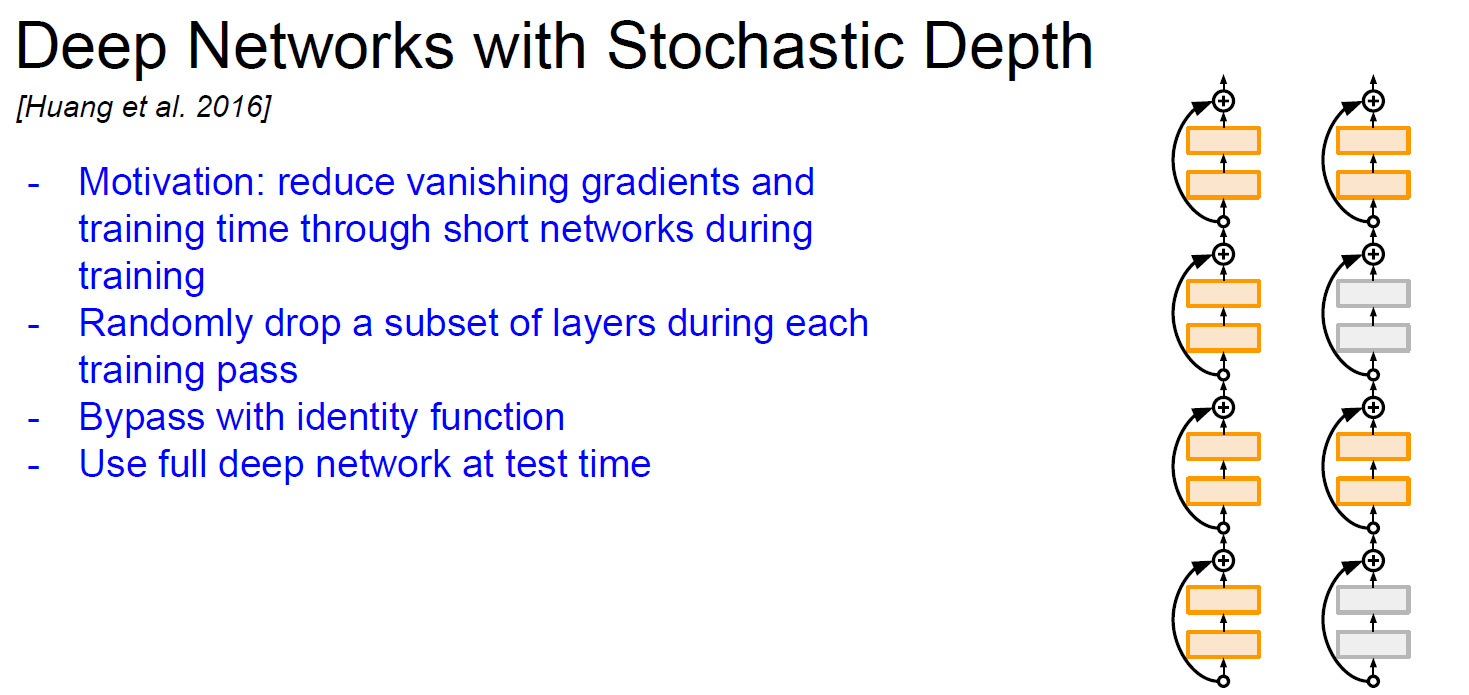


## Deep Networks with Stochastic Depth [Huang et al. 2016]

随机深度

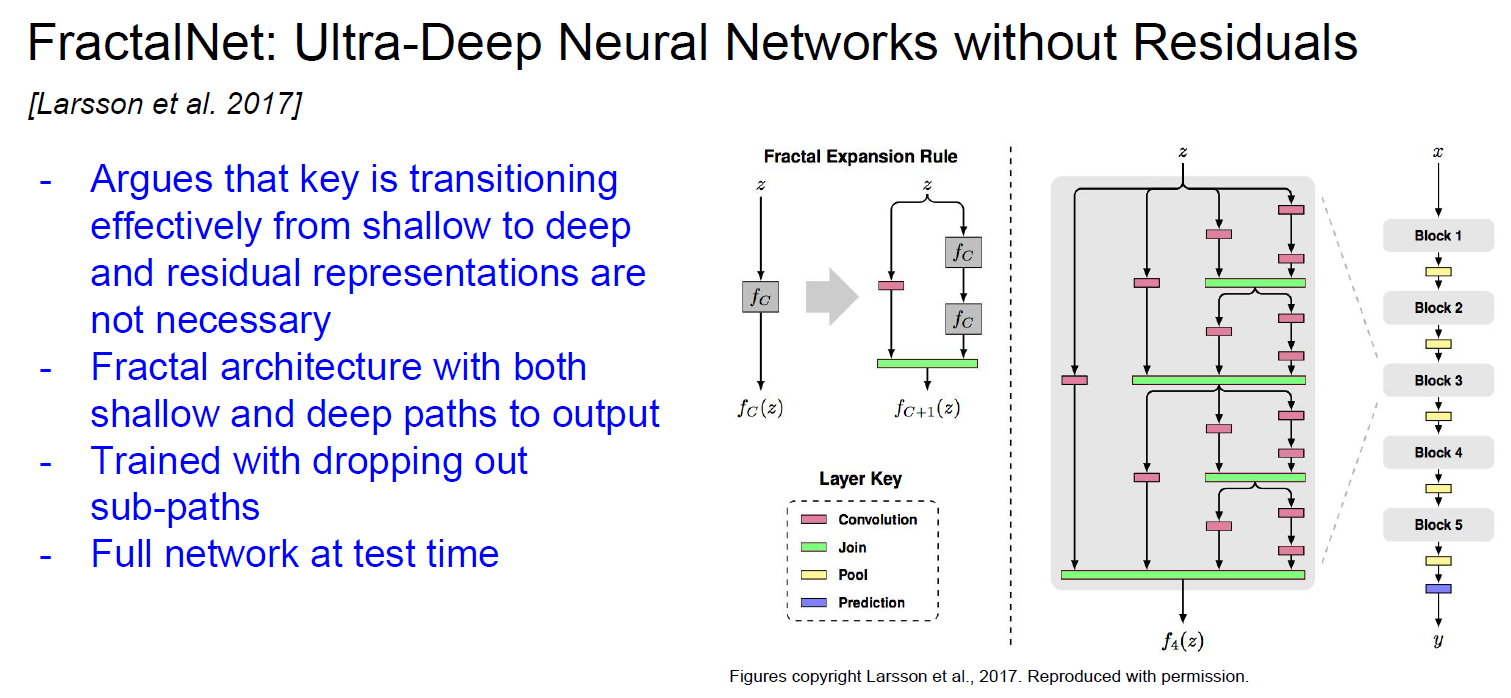
在训练过程中删除部分图层（不更新参数，但正常使用），使网络更短，效率更好

类似于dropout



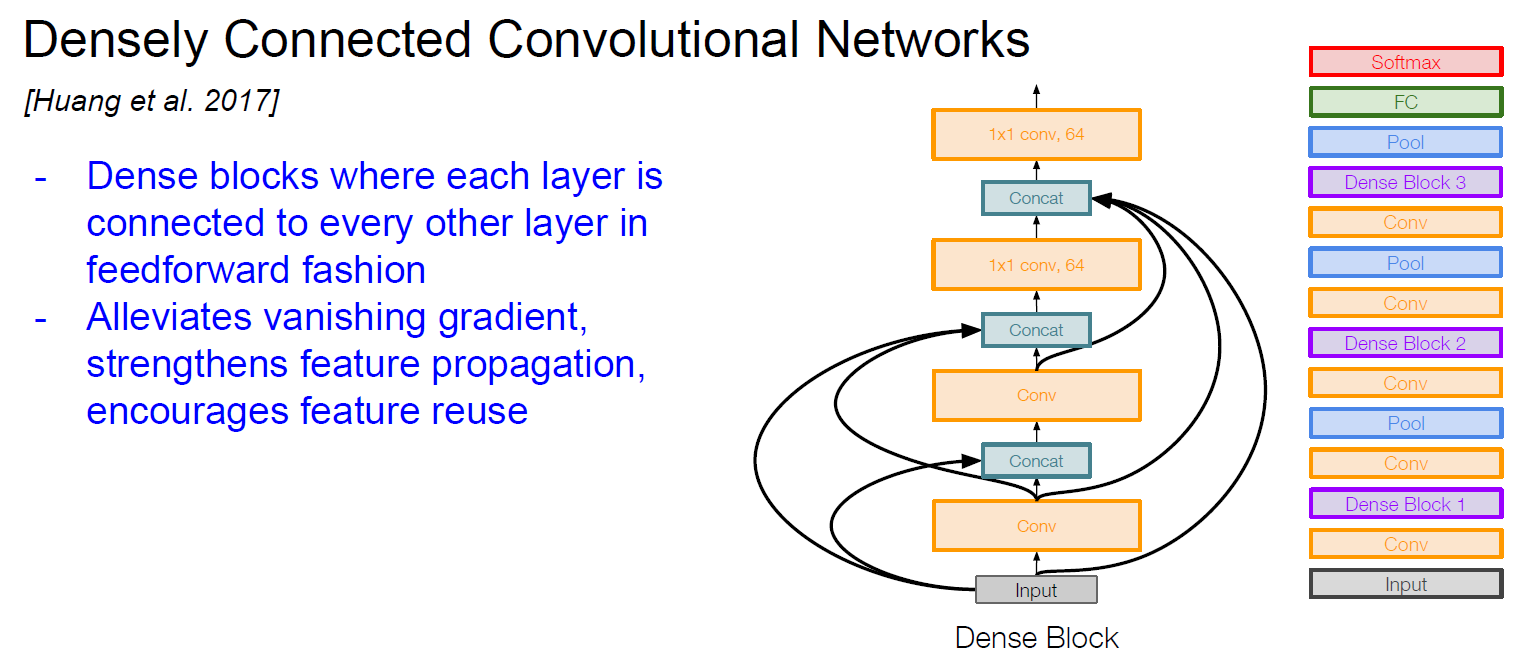
## FractalNet: Ultra-Deep Neural Networks without Residuals [Larsson et al. 2017]

从浅到深，不需要残差映射。输出的路径有浅路径和深路径。



## Densely Connected Convolutional Networks [Huang et al. 2017]

DenseNet。使用密集块，其中每一层都以前馈方式连接到每一层。减轻消失梯度，加强特征传播，鼓励特征重复使用



## SqueezeNet: AlexNet-level Accuracy With 50x Fewer Parameters and <0.5Mb Model Size [Iandola et al. 2017]

没有讲，需要用到的时候再去看

