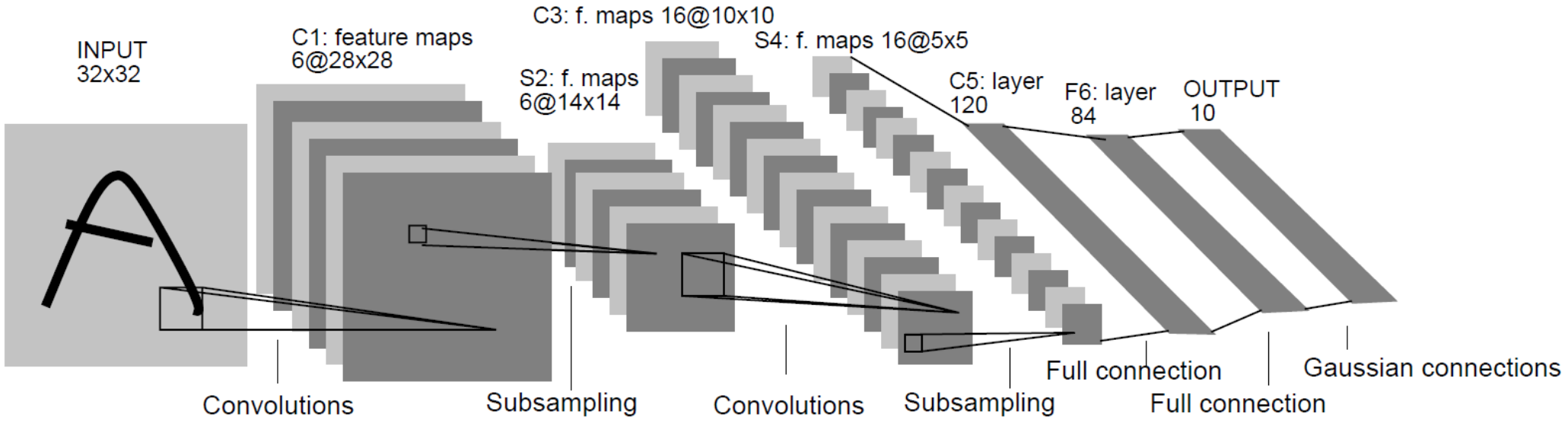
上节课回顾

- LeNet
- AlexNet
- ZF-Net
- VGG-Net

1. LeNet-5

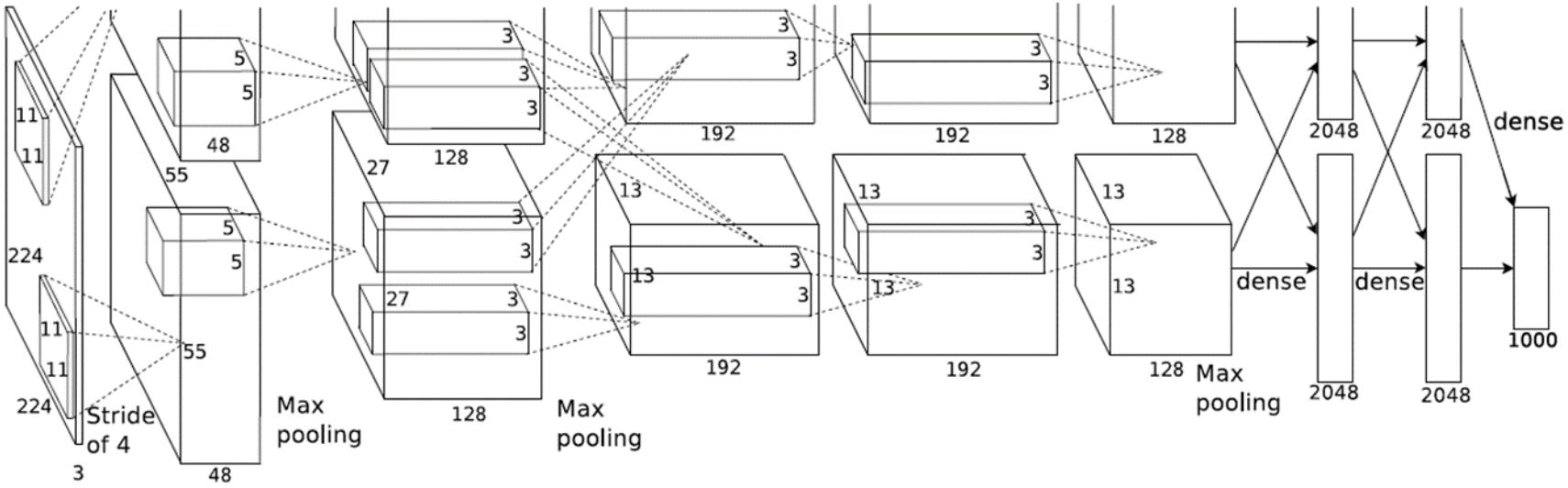


http://scs.ryerson.ca/~aharley/vis/conv/

输入尺寸：32*32
卷积层：2个
下采样层：2个
全连接层：2个
输出：10个类别（数字0-9的概率）

LeCun *et al.*, (1998) Gradient-Based Learning Applied to Document Recognition (Proc. IEEE 1998)

2. AlexNet

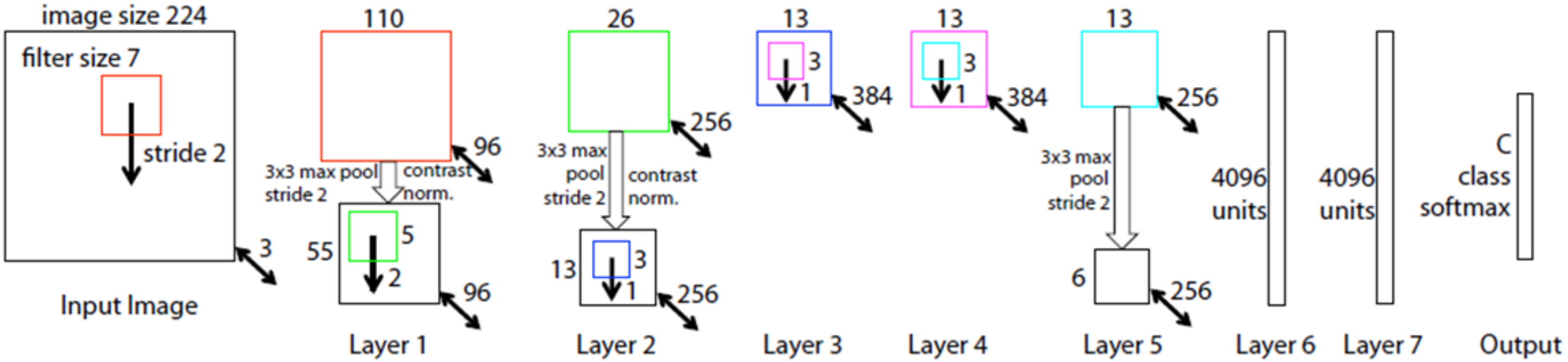


AlexNet架构

AlexNet由Alex Krizhevsky于2012年提出，夺得2012年ILSVRC比赛的冠军，top5预测的错误率为16.4%，远超第一名。AlexNet采用8层的神经网络，5个卷积层和3个全连接层(3个卷积层后面加了最大池化层)，6000万个参数和65万个神经元。

Krizhevsky A., Sutskever I., Hinton G. (2012) ImageNet classification with deep convolutional neural networks. In: NIPS

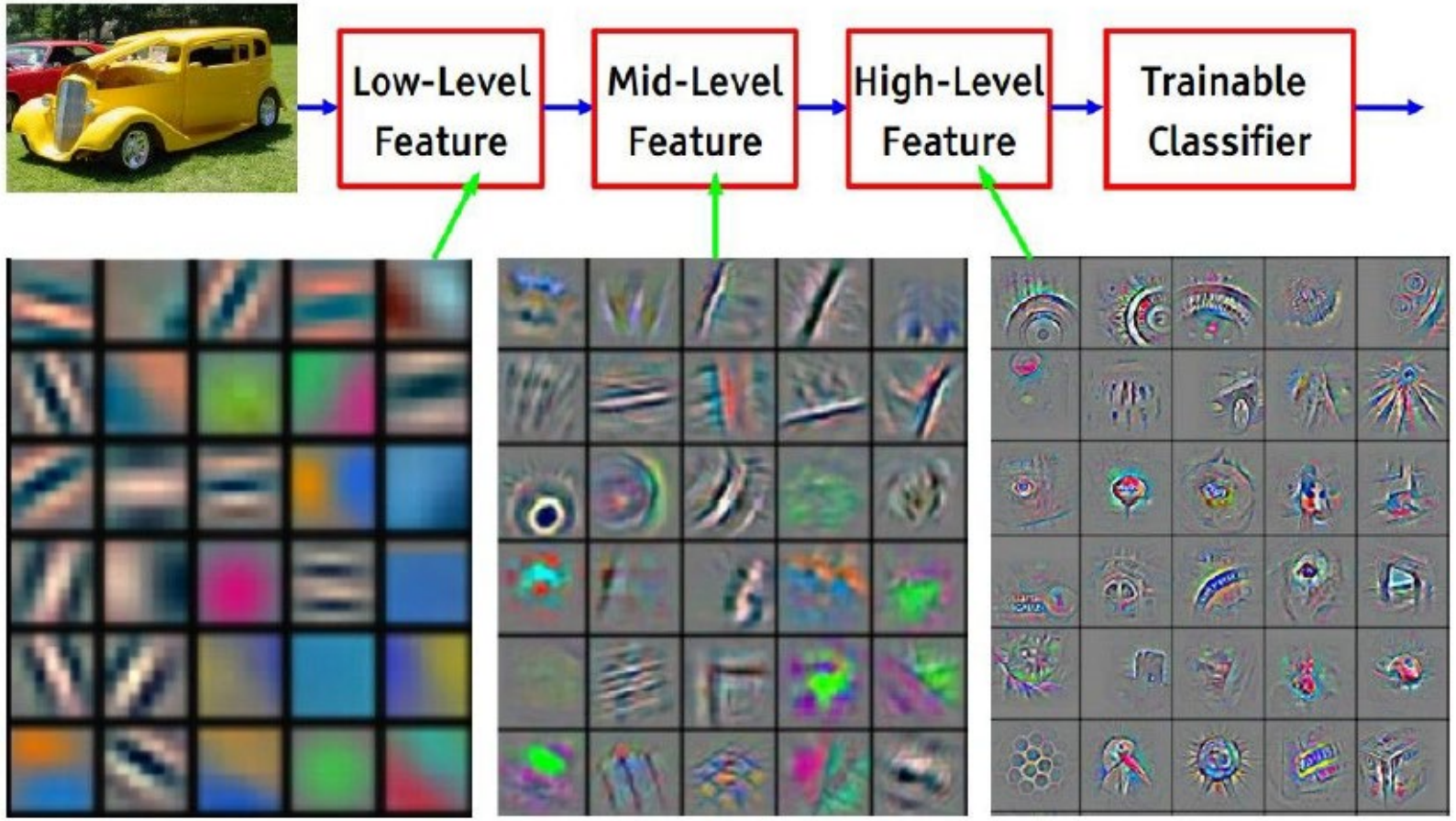
3. Zeiler&Fergus Net



Zeiler&Fergus Net

Zeiler M., Fergus R. (2014) Visualizing and understanding convolutional networks. In: ECCV.

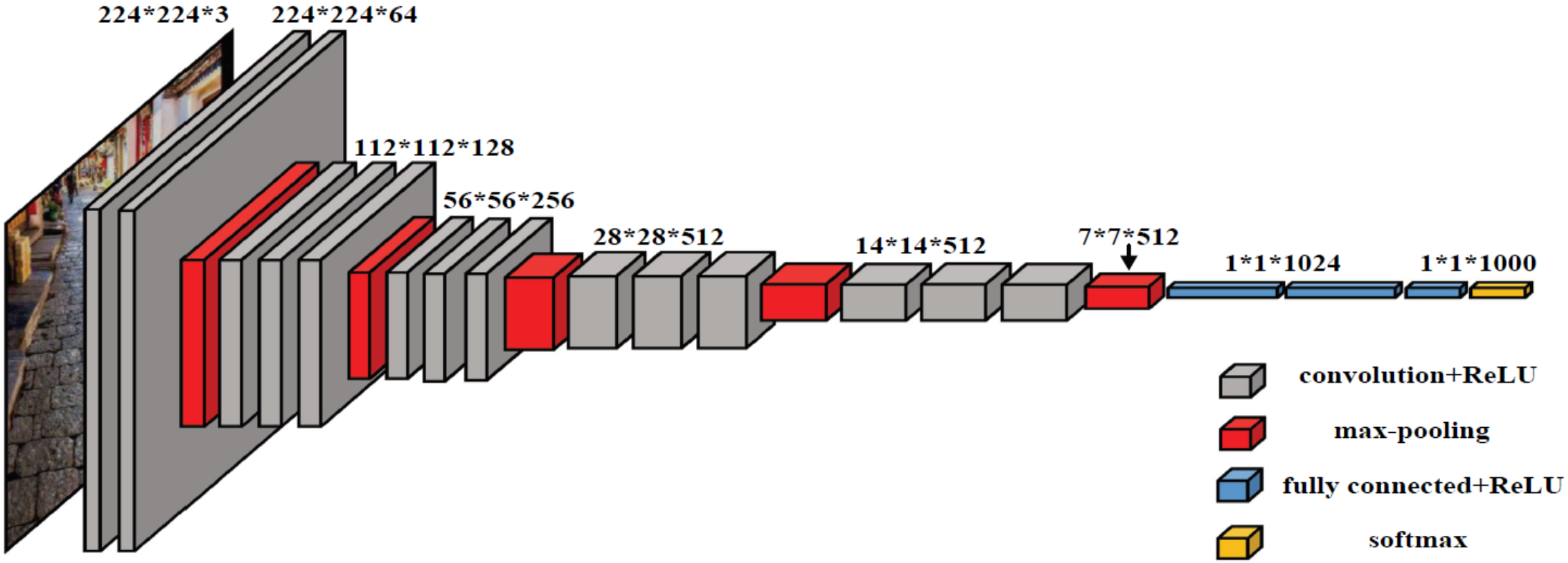
ZF-Net 可视化



https://github.com/vdumoulin/conv_arithmetic

https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

4. VGGNet



VGGNet一共有六种不同的网络结构，但是每种结构都有含有 5 组卷积，每组卷积都使用 3 x 3 的卷积核，每组卷积后进行一个 2 x 2 最大池化，接下来是三个全连接层。

Simonyan K., Zisserman A. (2015) Very deep convolutional networks for large scale image recognition. In: ICLR.

4. VGGNet

INPUT: [224x224x3] memory: 224*224*3=150K params: 0 (not counting biases)
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*64)*64 = 36,864
POOL2: [112x112x64] memory: 112*112*64=800K params: 0
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*128)*128 = 147,456
POOL2: [56x56x128] memory: 56*56*128=400K params: 0
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
POOL2: [28x28x256] memory: 28*28*256=200K params: 0
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512] memory: 14*14*512=100K params: 0
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512] memory: 7*7*512=25K params: 0
FC: [1x1x4096] memory: 4096 params: 7*7*512*4096 = 102,760,448
FC: [1x1x4096] memory: 4096 params: 4096*4096 = 16,777,216
FC: [1x1x1000] memory: 1000 params: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ~= 93MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters

VGG16 VGG19

ConvNet Configuration			
B	C	D	E
13 weight layers	16 weight layers	16 weight layers	19 weight layers
put (224 × 224 RGB image)			
conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool			
conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool			
conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool			
conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool			
conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool			
FC-4096			
FC-4096			
FC-1000			
soft-max			

Simonyan K., Zisserman A. (2015) Very deep convolutional networks for large scale image recognition. In: ICLR.

5. GoogLeNet

GoogLeNet（也称Inception）是2014年Christian Szegedy提出的一种全新的深度学习结构，在这之前的AlexNet、VGG等结构都是通过增大网络的深度（层数）来获得更好的训练效果，但层数的增加会带来很多负作用，比如overfitting、梯度消失、梯度爆炸等。Inception的提出则从另一种角度来提升训练结果：能更高效的利用计算资源，在相同的计算量下能提取到更多的特征，从而提升训练结果。

Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

5. GoogLeNet

那么，GoogLeNet是如何进一步提升性能的呢？

一般来说，提升网络性能最直接的办法就是增加网络深度和宽度，深度指网络层次数量、宽度指神经元数量。但这种方式存在以下问题：

（1）参数太多，如果训练数据集有限，很容易产生过拟合；

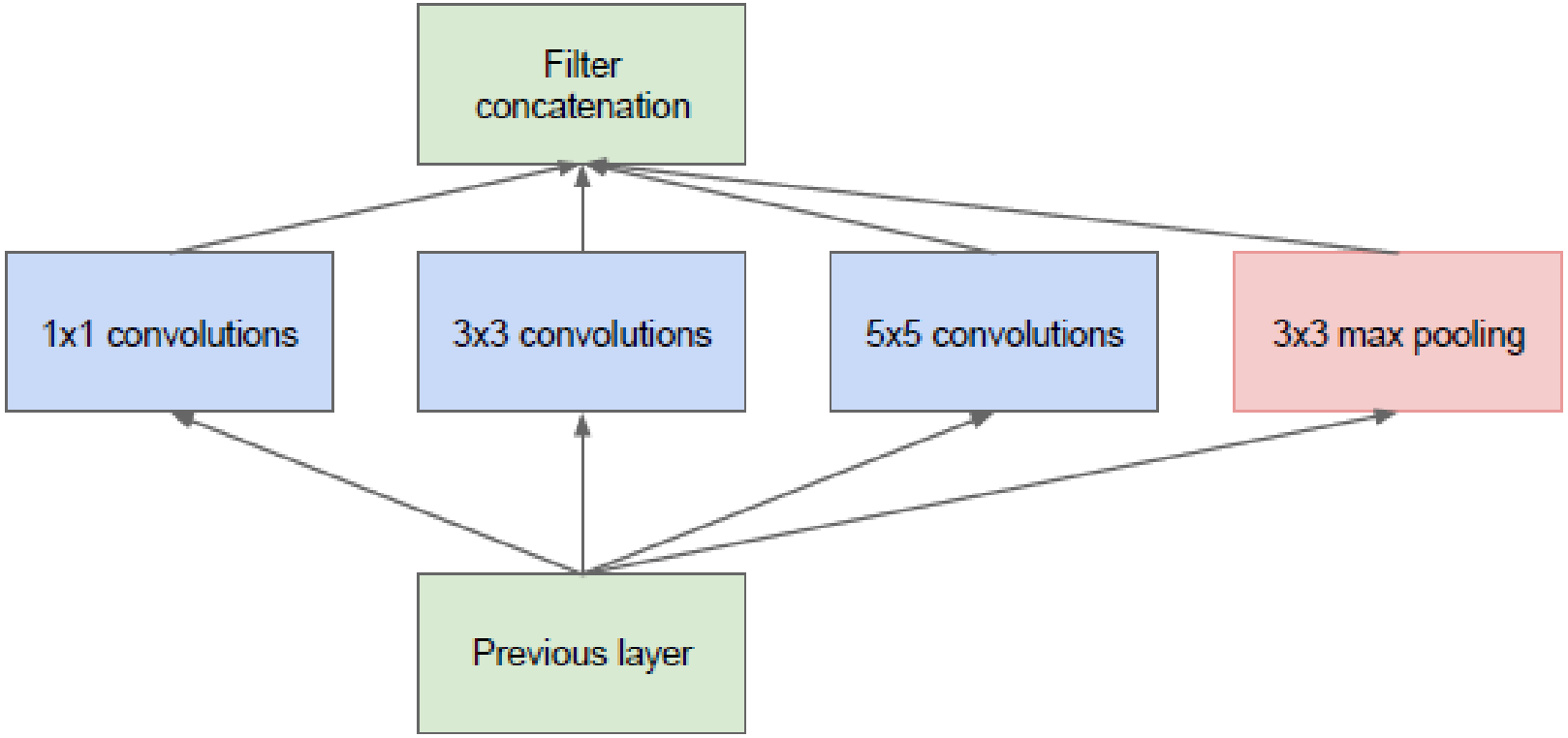
（2）网络越大、参数越多，计算复杂度越大，难以应用；

（3）网络越深，容易出现梯度消失问题，难以优化模型。

因此，GoogLeNet团队提出了Inception网络结构，就是构造一种“基础神经元”结构，来搭建一个稀疏性、高计算性能的网络结构。

Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

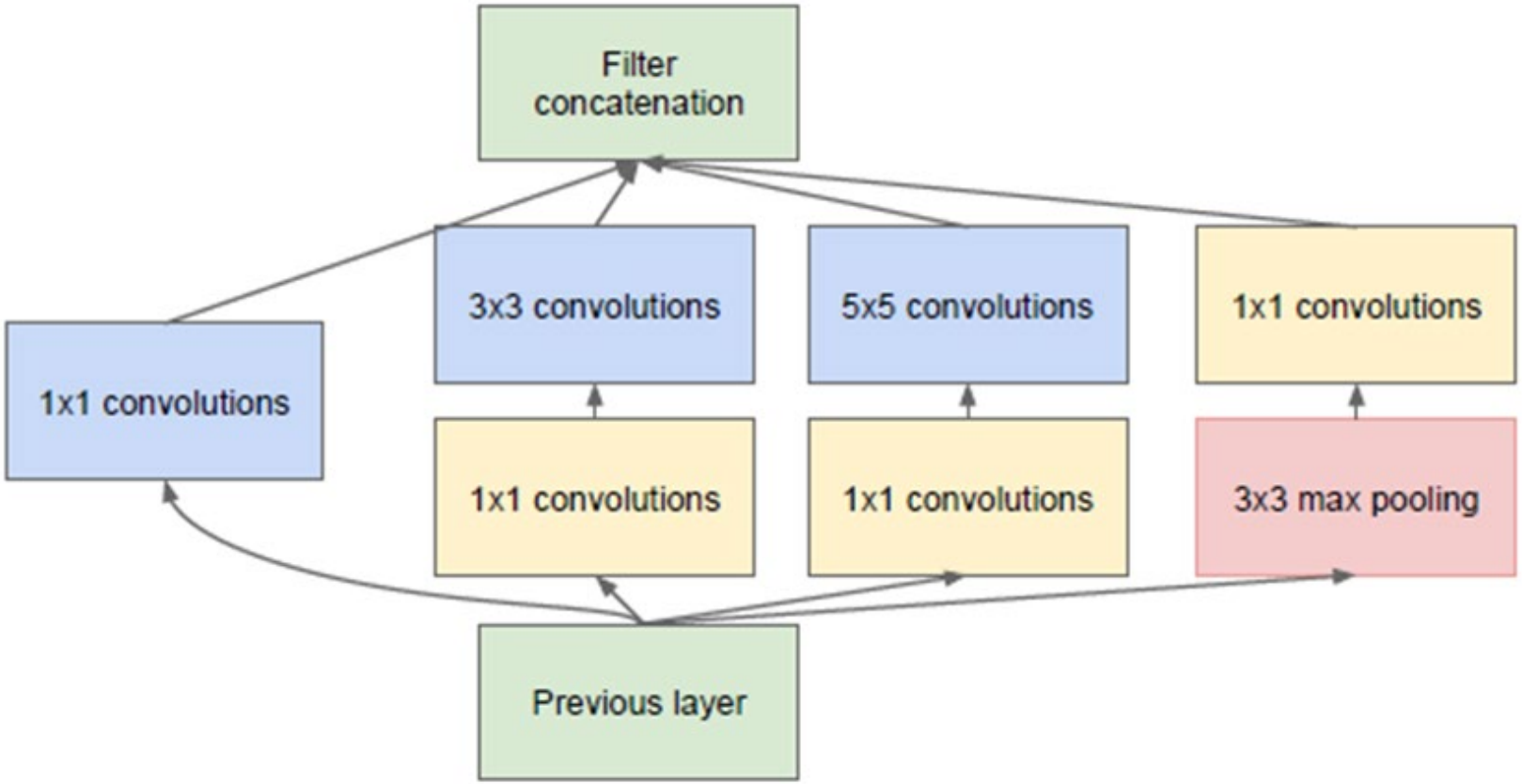
5. GoogLeNet



Inception模块

Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

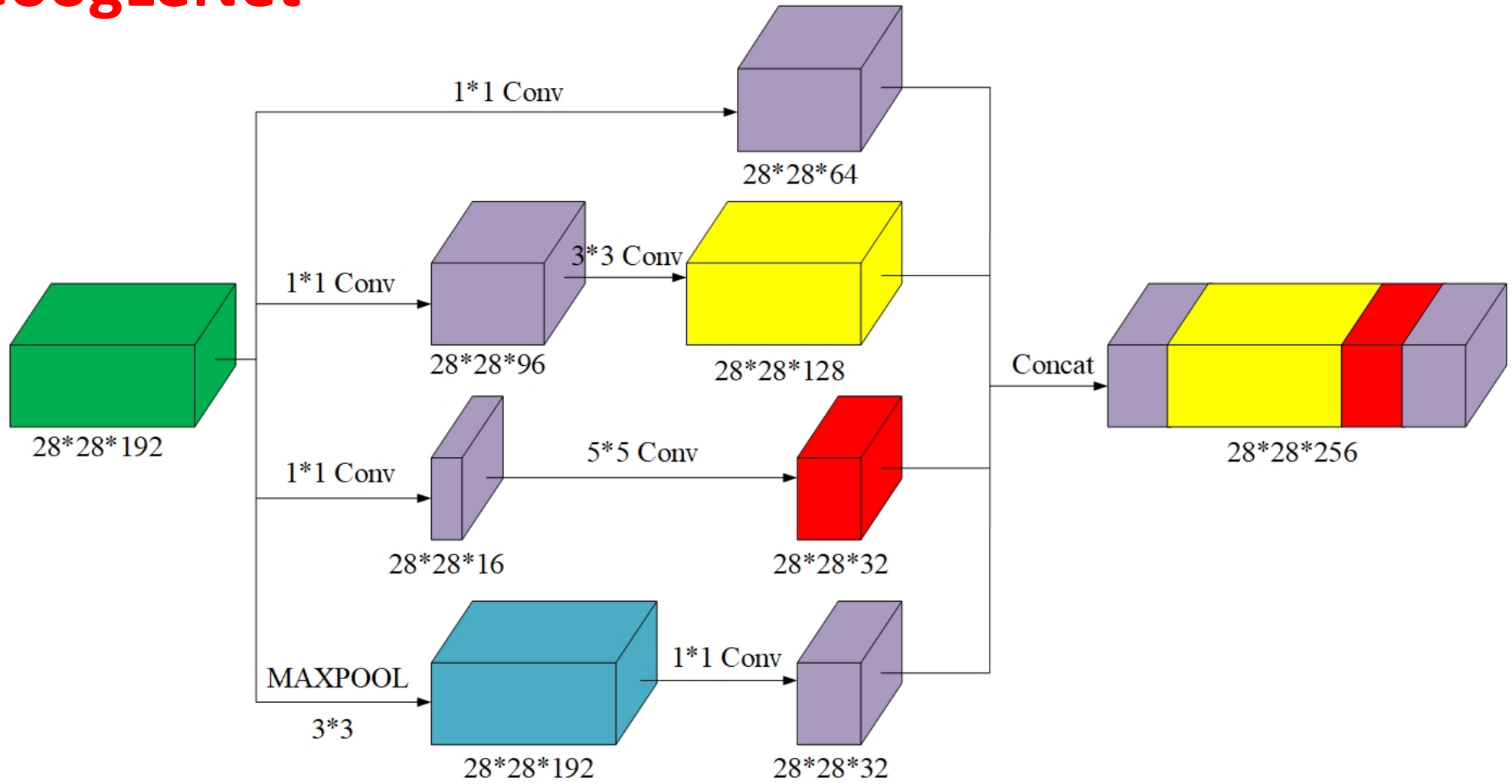
5. GoogLeNet



Inception降维模块

Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

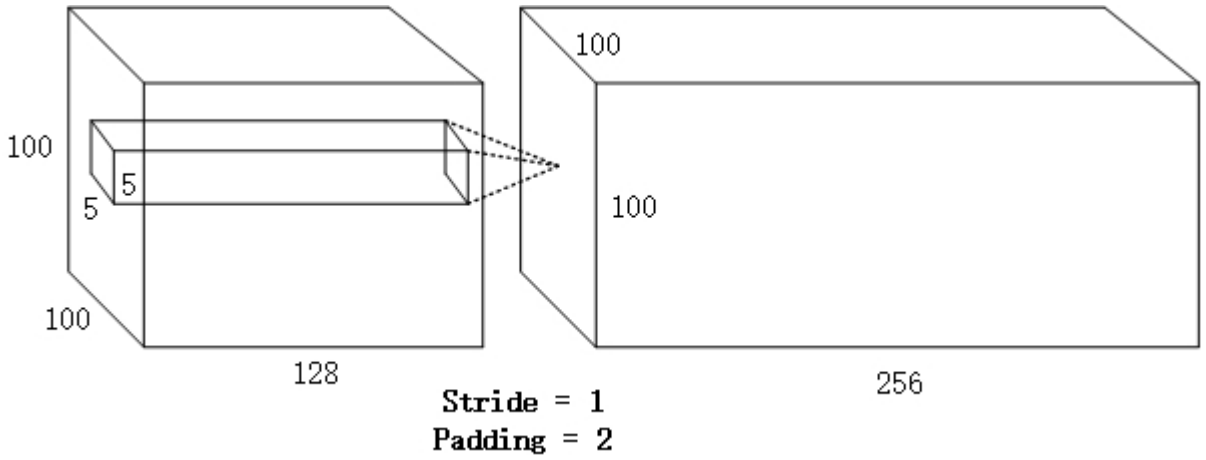
5. GoogLeNet



Filter Concatenation操作

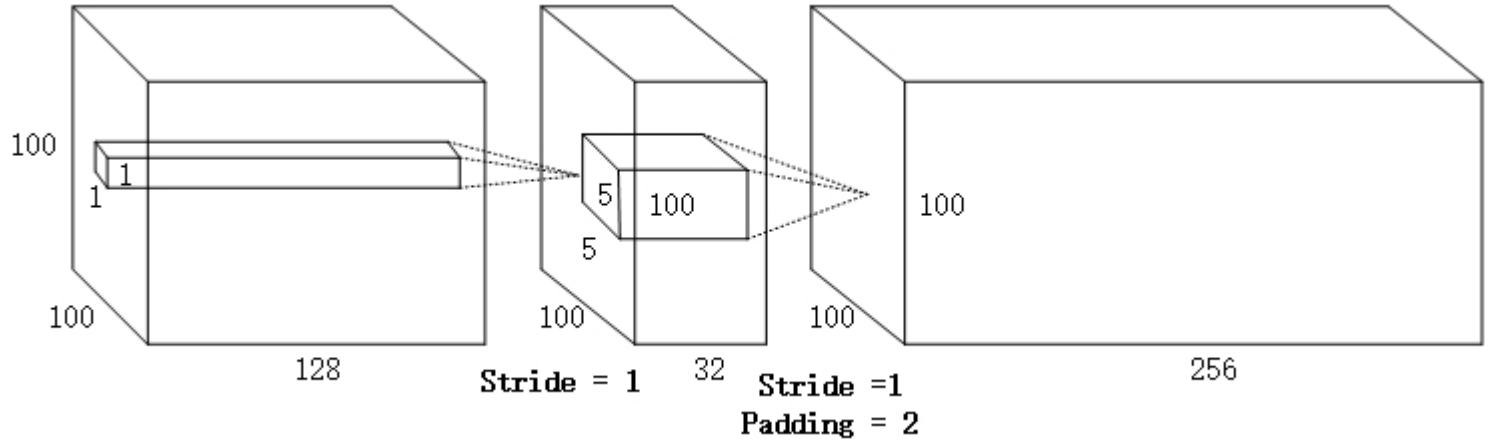
Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

5. GoogLeNet



Output:100 × 100 × 256
kernel parameter: 128 × 5 × 5 × 256 = 819200

Reduced four times

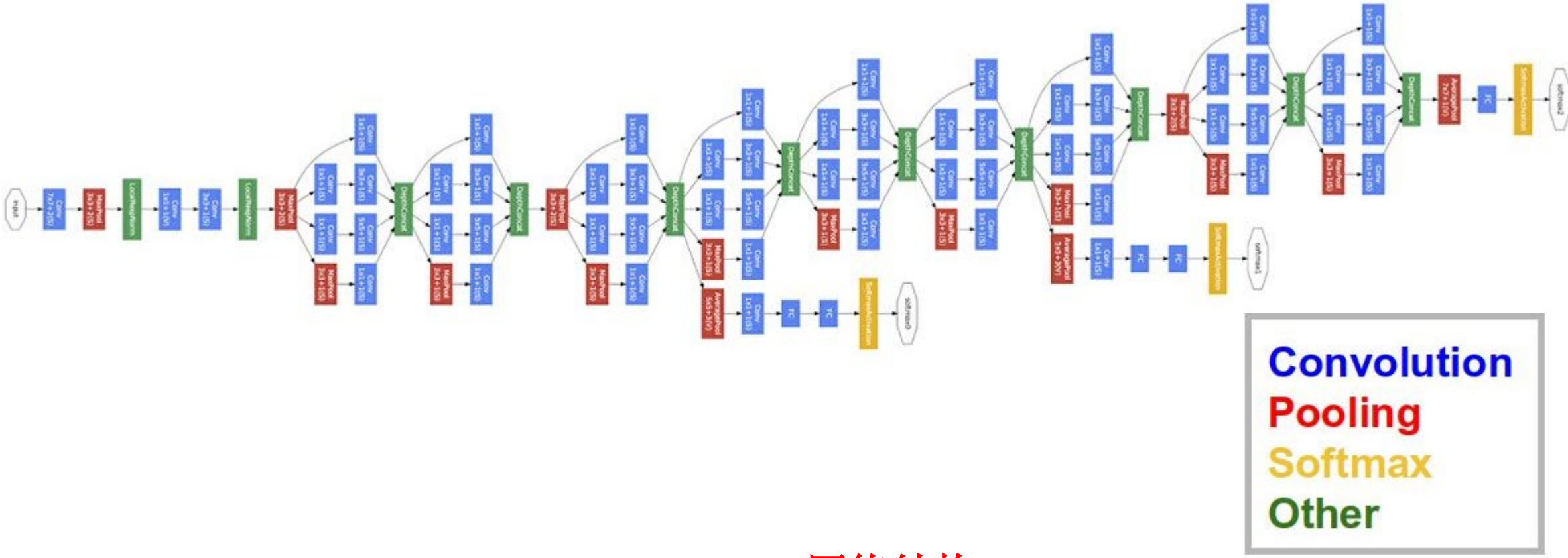


Output:100 × 100 × 256
kernel parameter:128 × 1 × 1 × 32 + 32 × 5 × 5 × 256 = 204800

1*1卷积核的降维（运算量）作用

Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

5. GoogLeNet



GoogLeNet网络结构

Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

5. GoogLeNet

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

网络的详细参数

Szegedy C., LiuW., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR.

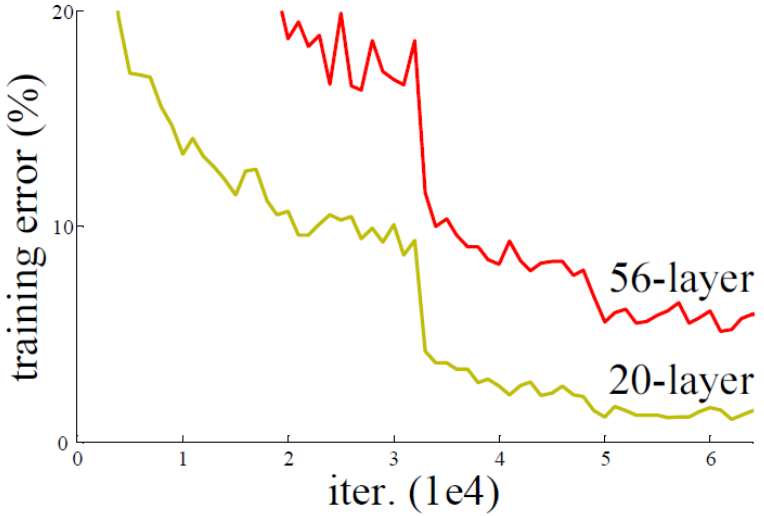
6. ResNet

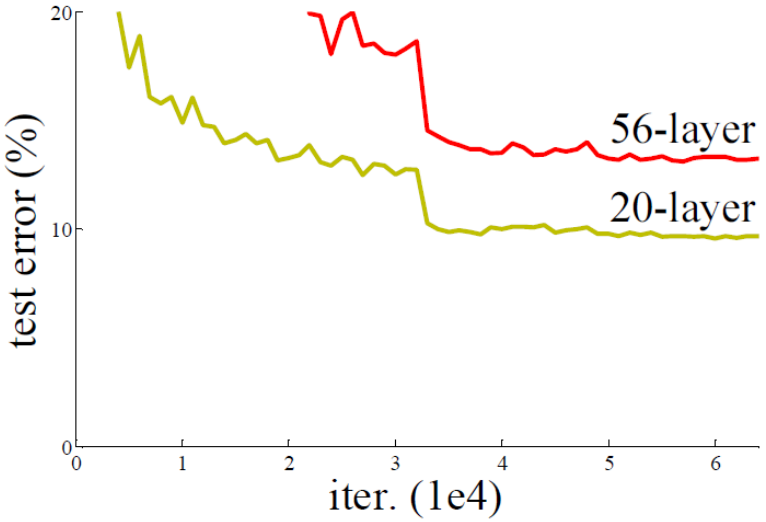
在深度重要性的推动下，出现了一个问题：学习更好的网络是否像堆叠更多的层一样容易？

这个问题的本质就是是否网络性能的增加只需要增加网络的深度（层数）？

He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR.

6. ResNet





20层和56层的“简单”网络在CIFAR-10上的训练误差（上）和测试误差（下）。更深的网络有更高的训练误差和测试误差。ImageNet有类似的情况。

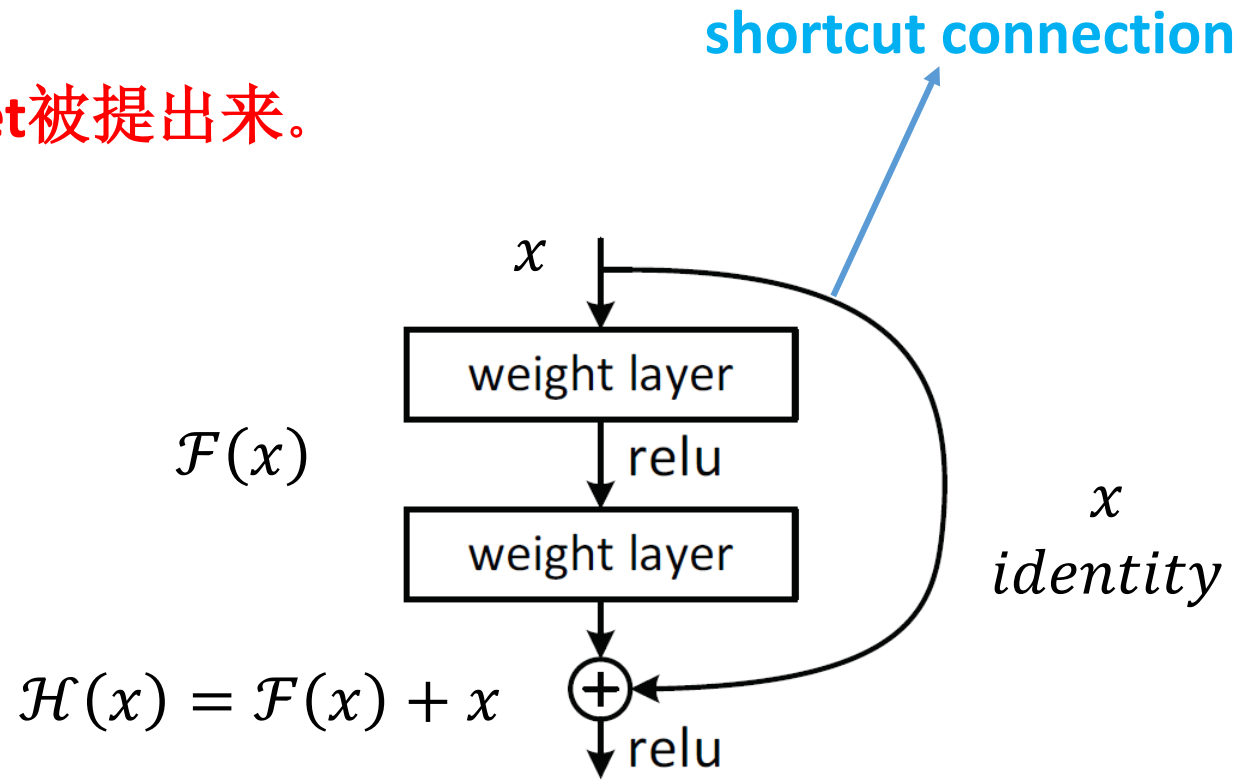
当更深的网络能够开始收敛时，暴露了一个退化问题：随着网络深度的增加，准确率达到饱和然后迅速下降。

He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR.

6. ResNet

为了解决这个问题，深度残差网络ResNet被提出来。

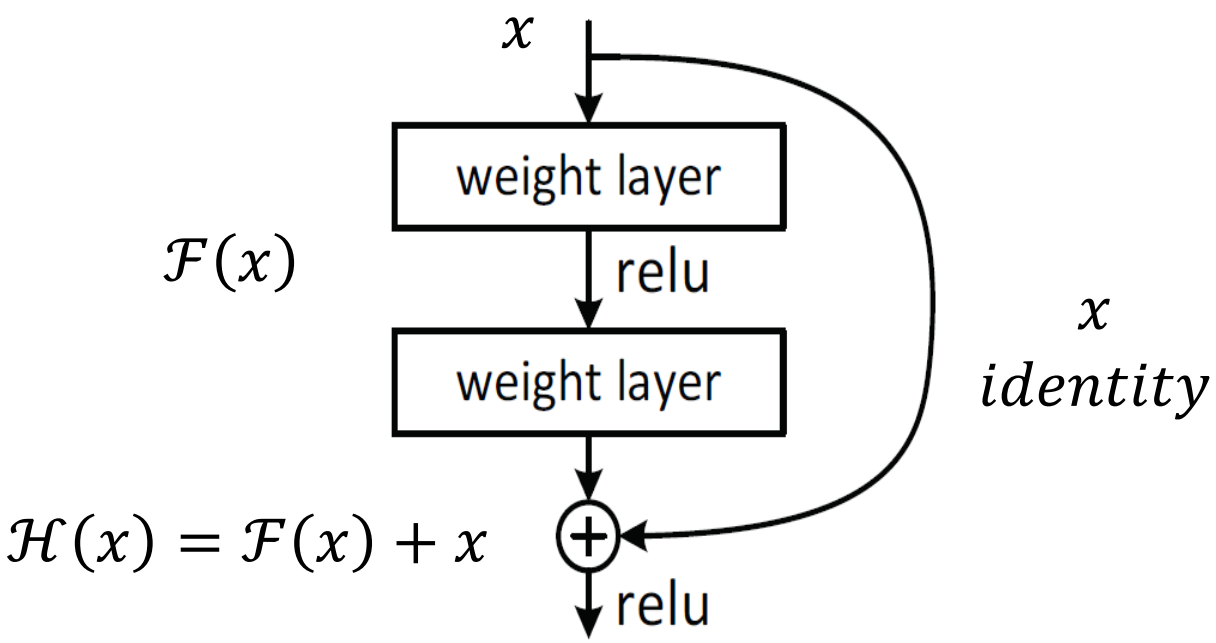
其中ResNet提出了两种mapping：一种是identity mapping，指的就是图中”曲线”，另一种residual mapping，指的就是除了”曲线“那部分，所以最后的输出是 $\mathcal{H}(x)=\mathcal{F}(x)+x$ 。identity mapping顾名思义，就是指本身，也就是公式中的x，而residual mapping指的是“差”，也就是$\mathcal{H}(x)-x$，所以残差指的就是$\mathcal{F}(x)$部分。



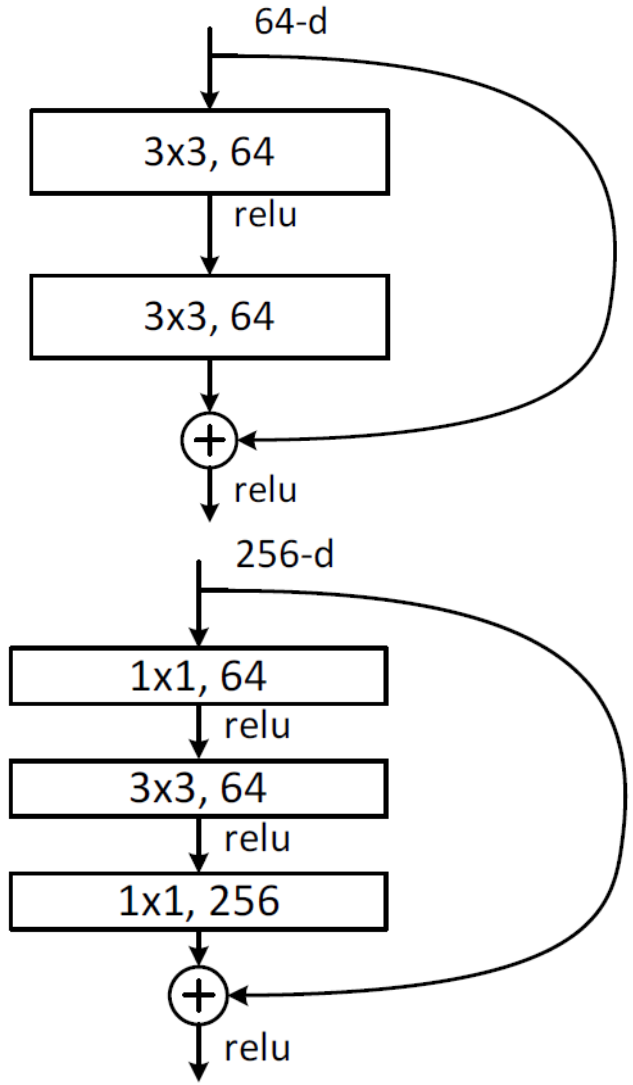
基本的残差学习模块

He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR.

6. ResNet

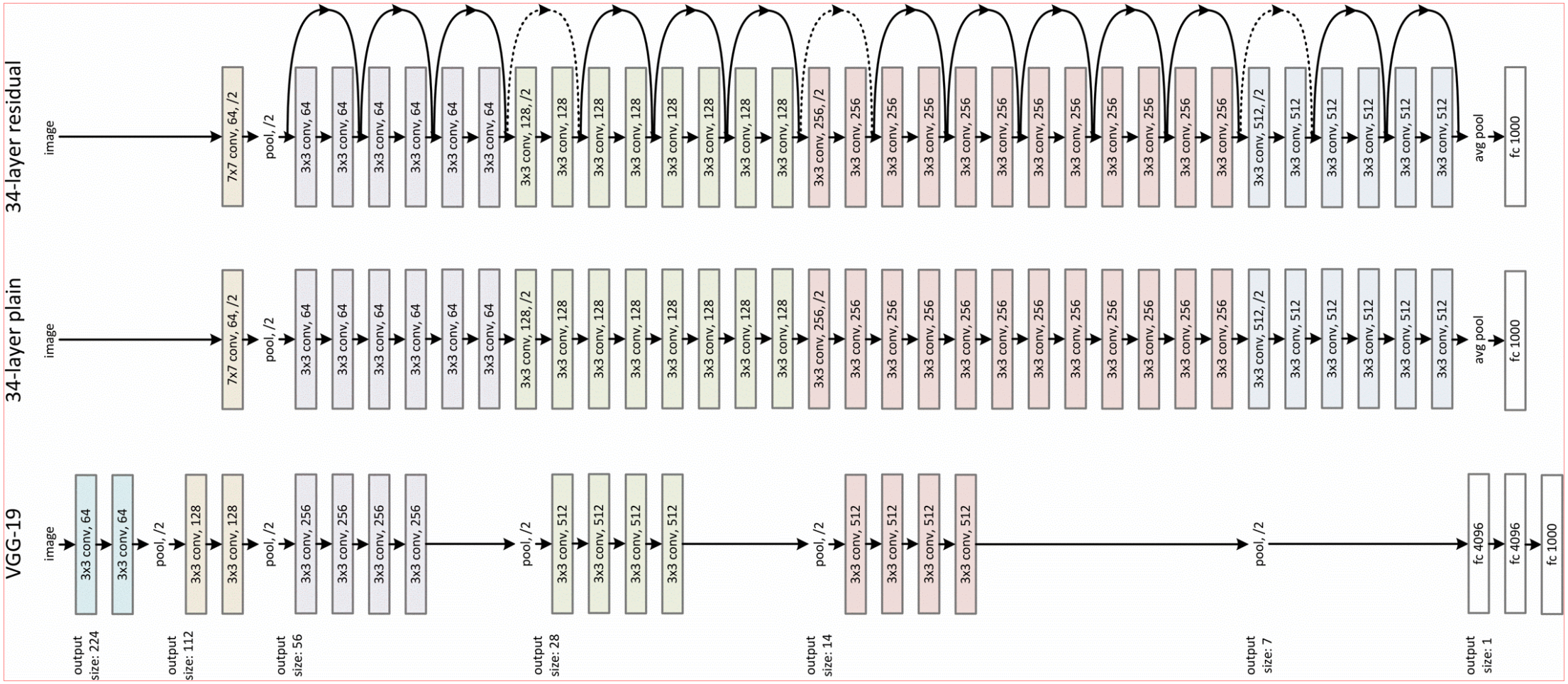


基本的残差学习模块



He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR.

6. ResNet



He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR.

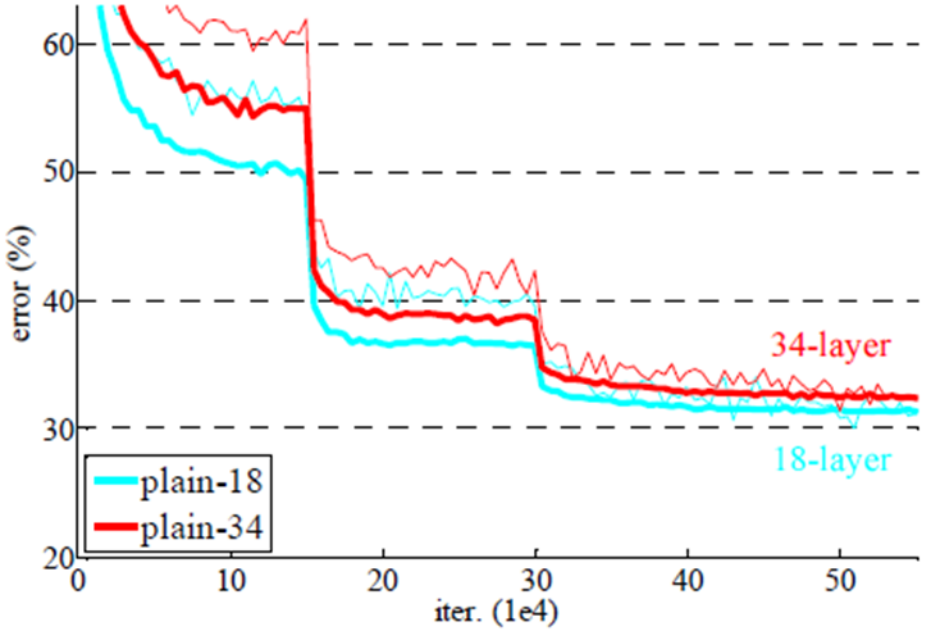
6. ResNet

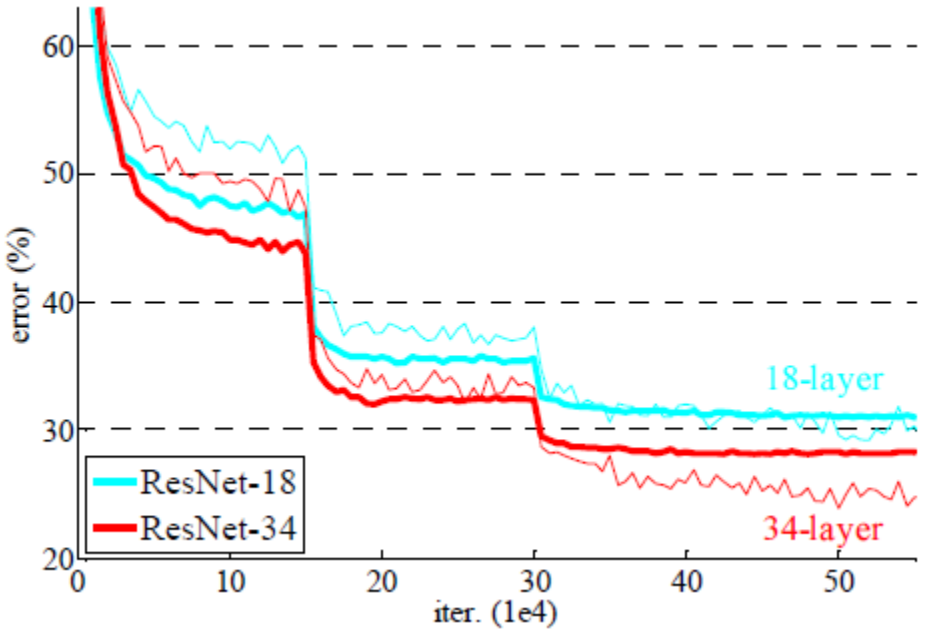
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR.

6. ResNet

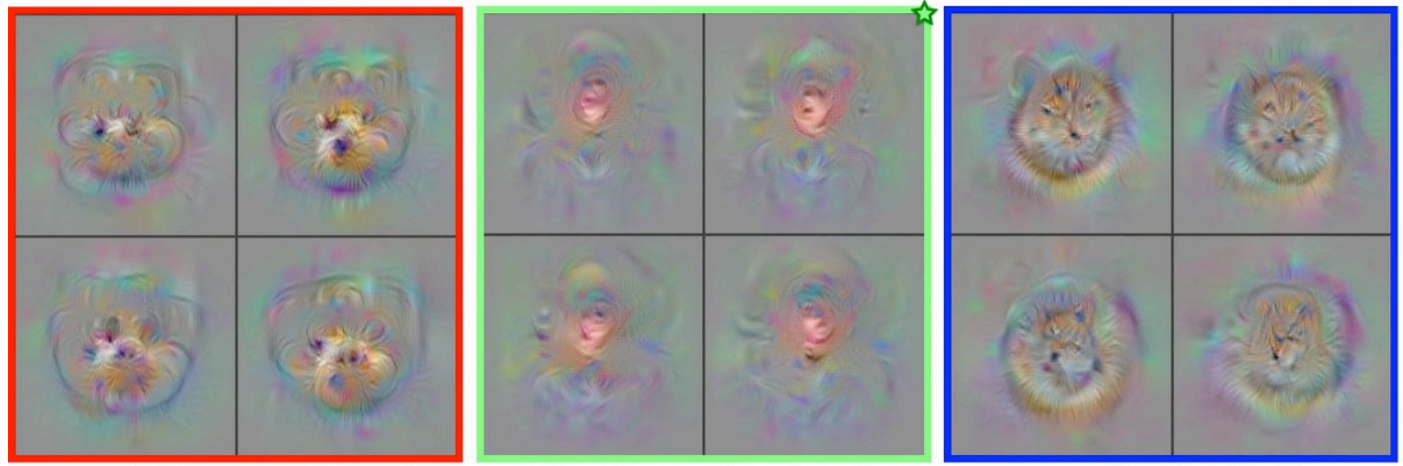
	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**





在ImageNet上训练。细曲线表示训练误差，粗曲线表示中心裁剪图像的验证误差。左：18层和34层的简单网络。右：18层和34层的ResNet。在本图中，残差网络与对应的简单网络相比没有额外的参数。

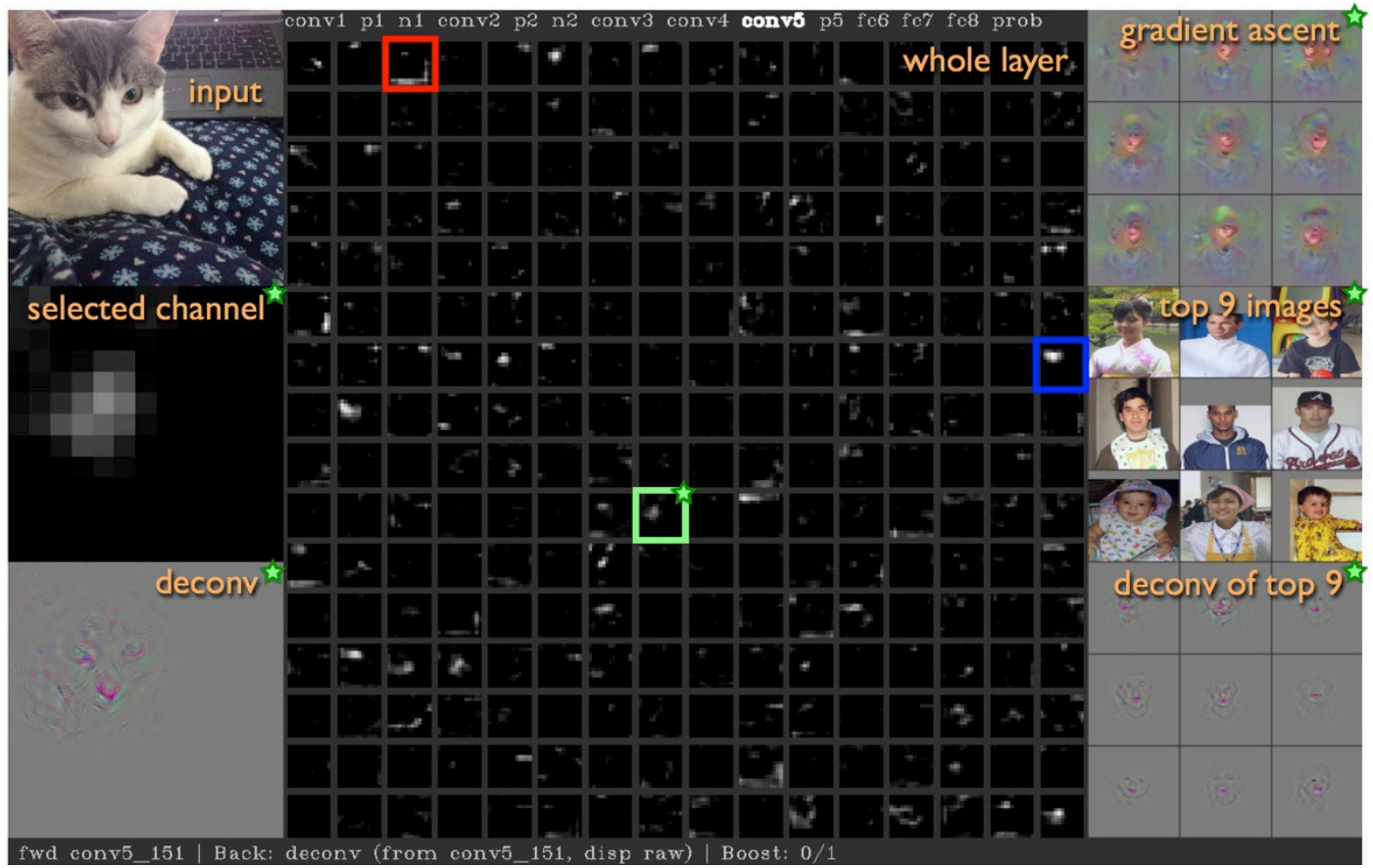
He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR.



conv5$_2$ (dog face + flower)

conv5$_{151}$ (human face + cat face)

conv5$_{111}$ (cat face)



参考资料

- 1. Stanford CS231n, 2016, 2017

- 2. Andrew Ng 机器学习与深度学习课程