法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

第6课　图像分类

计算机视觉

主讲人：叶梓

上海交通大学博士
主要研究方向：机器学习、深度学习、人工智能

本章结构

- ILSVRC竞赛
- 常见的CNN
  - Alexnet
  - VGG
  - GoogLeNet（Inception V1、V2、V3）
  - ResNet、ResNeXt
- 应用案例：
  - VGG
  - ResNet

图片分类

判断图片中是否有某个物体

- 一个图对应一个标签
- 性能指标
  - Top1 error → 前1中1
  - Top5 error → 前5中1

Steel drum



输出：
Scale
T-shirt
Steel drum
Drumstick
Mud turtle

✓

输出：
Scale
T-shirt
Giant panda
Drumstick
Mud turtle

✗

ILSVRC竞赛

ImageNet Large Scale Visual Recognition Challenge

- 多个任务
  - 图像分类（Image Classification ）
    - 1000个分类
    - 训练集（1.2M）、验证集（50K）、测试集（150K）
  - 场景分类（Scene Classification）
  - 物体检测（Object Detection）
  - 物体定位（Object Localization）
  - 场景解析（Scene parsing）
- URL
  - http://image-net.org/challenges/LSVRC/2017/

ILSVRC竞赛

ImageNet数据集

- 根据WordNet组织的图片集
  - 100,000+个词/词组（synsets）
  - 8,000+个名词
- 为一个名词提供平均1000张图片
  - 总共14,197,122张图片
  - 支持21,841个synsets
- URL
  - http://image-net.org/

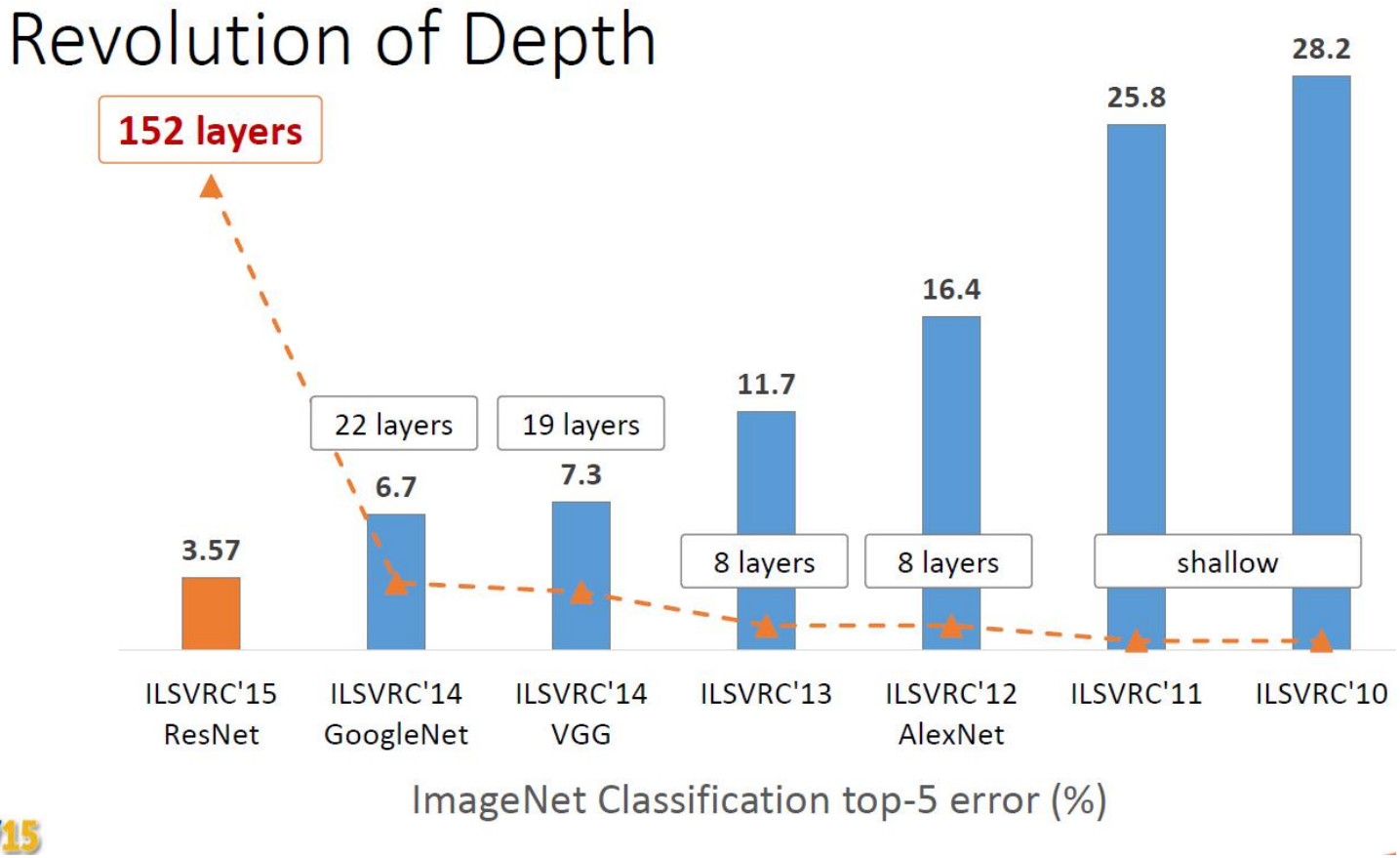
ILSVRC竞赛

ImageNet Large Scale Visual Recognition Challenge

- 多个任务
  - 图像分类（Image Classification ）
  - 场景分类（Scene Classification）
    - 来自MIT的Places2数据集（图片10M+、分类400+）
    - 365个场景分类
    - 训练集（8M）、验证集（36K）、测试集（328K）
  - 物体检测（Object Detection）
  - 物体定位（Object Localization）
  - 场景解析（Scene parsing）

卷积神经网络（CNN）

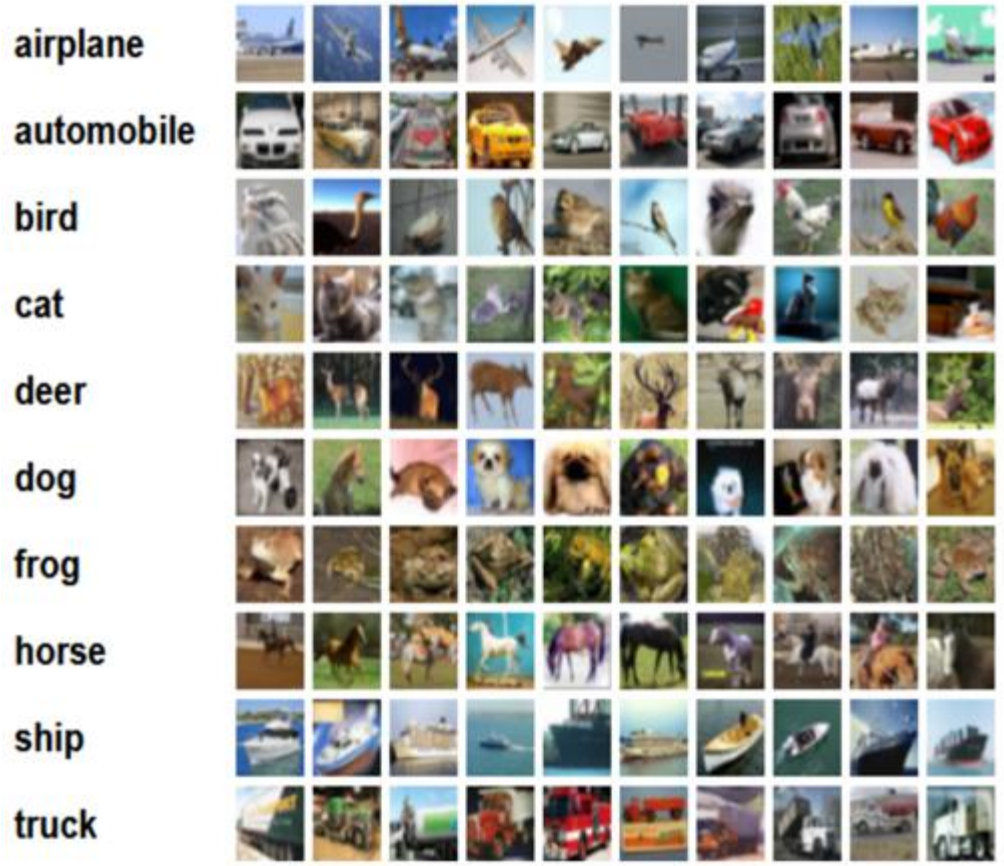
ImageNet性能进化



CIFAR-10数据集

collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

- 该数据集共有60000张彩色图像，这些图像的分辨率是32*32，分为10个类，每类6000张图。
- 其中有50000张用于训练，构成了5个训练批，每一批10000张图；另外10000用于测试，单独构成一批。
- 测试批的数据里，取自10类中的每一类，每一类随机取1000张。

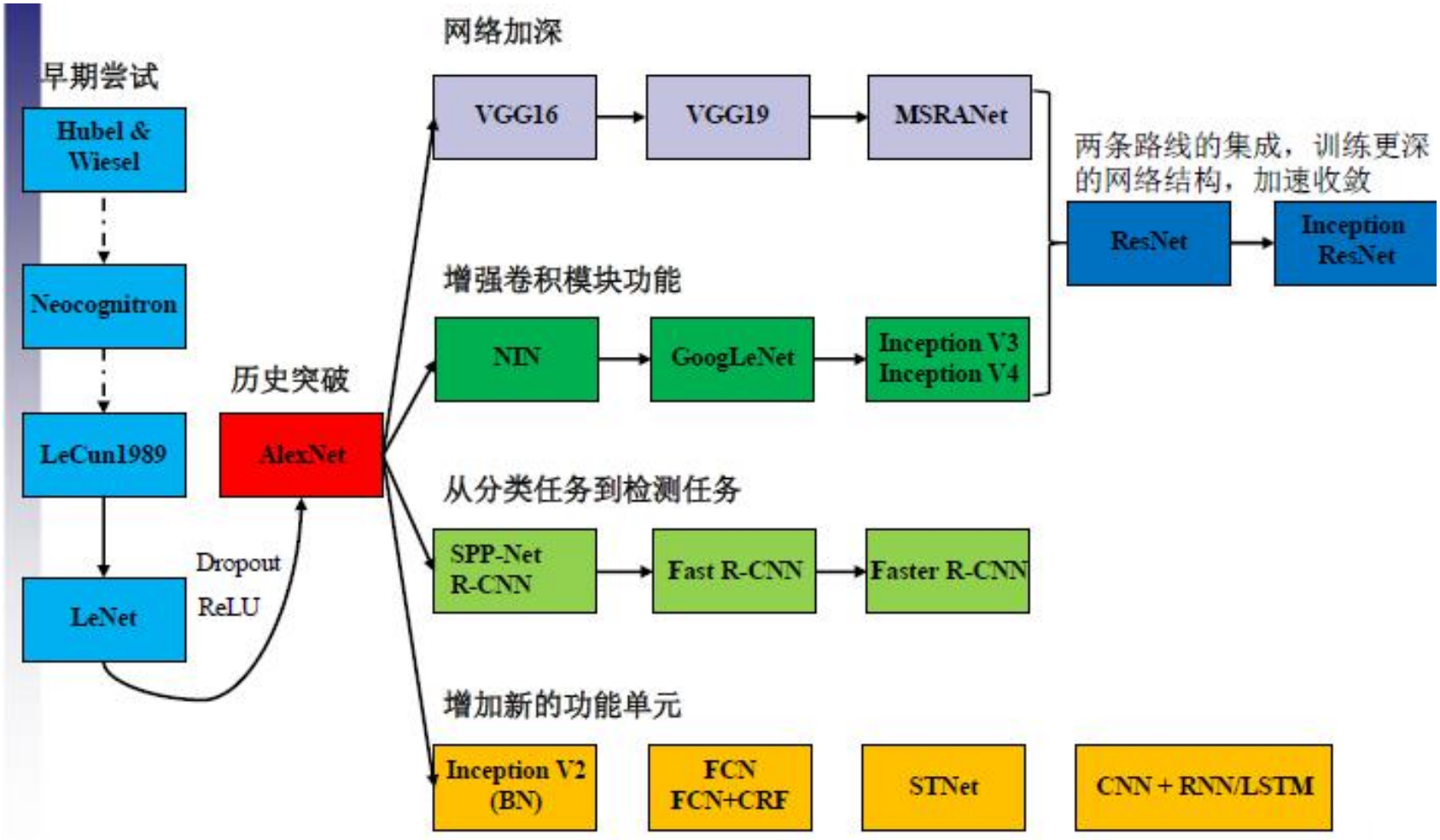


卷积神经网络（CNN）

网络进化

- 网络：AlexNet → VGG → GoogLeNet → ResNet
- 深度：8 → 19 → 22 → 152
- VGG结构简洁有效
  - 容易修改，迁移到其他任务中去
  - 高层任务的基础网络
- 性能竞争网络
  - GoogLeNet：Inception v1 → v4
    - Split-transform-merge
  - ResNet：ResNet1024 → ResNeXt
    - 深度、宽度、基数(cardinality)

CNN结构的演化


早期尝试
Hubel & Wiesel
Neocognitron
LeCun1989
LeNet
历史突破
AlexNet
Dropout
ReLU
网络加深
VGG16
VGG19
MSRANet
增强卷积模块功能
NIN
GoogLeNet
Inception V3
Inception V4
从分类任务到检测任务
SPP-Net
R-CNN
Fast R-CNN
Faster R-CNN
增加新的功能单元
Inception V2
(BN)
FCN
FCN+CRF
STNet
CNN + RNN/LSTM
两条路线的集成，训练更深的网络结构，加速收敛
ResNet
Inception ResNet

AlexNet网络

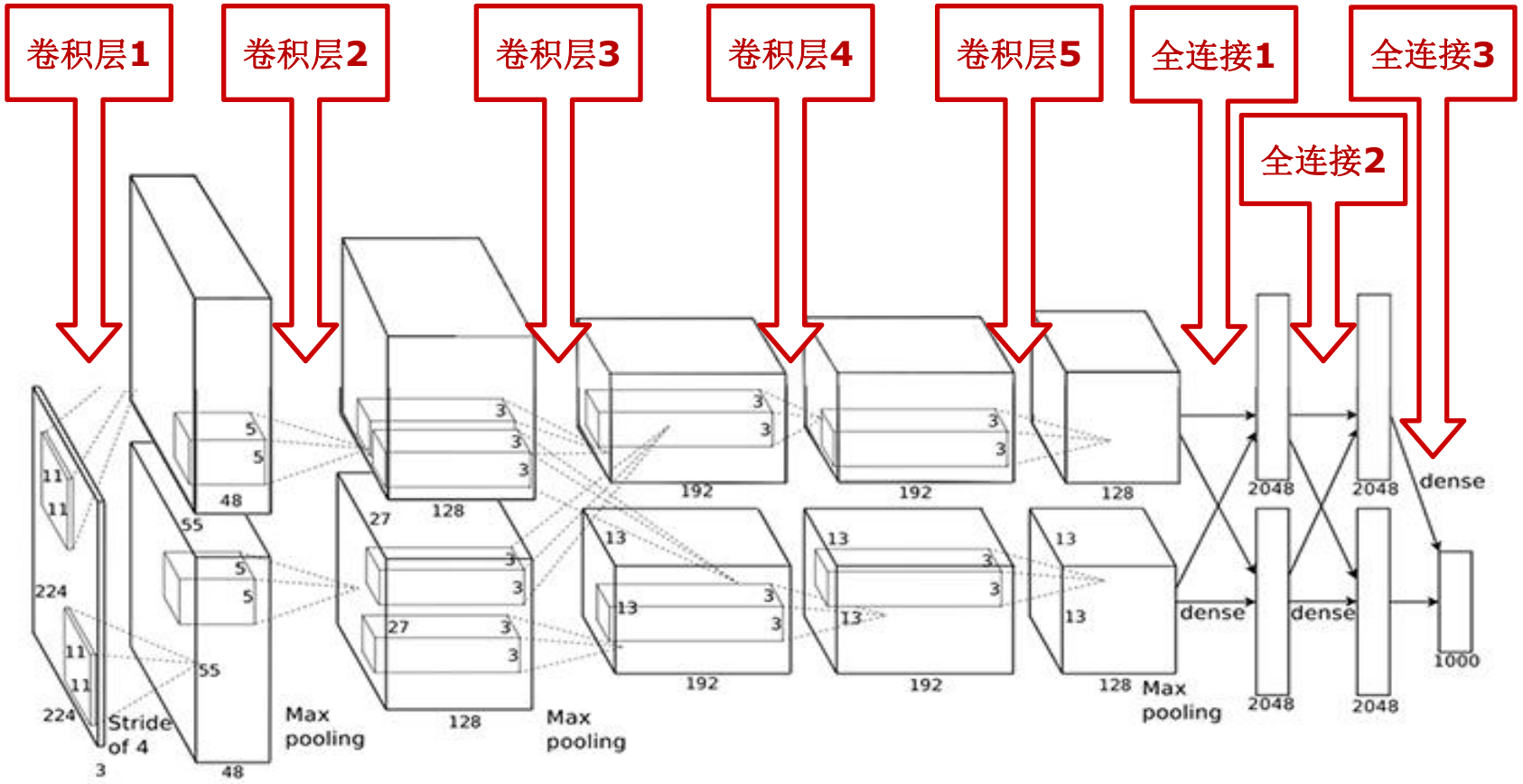
AlexNet网络

- ImageNet-2012竞赛第一
- 标志着DNN深度学习革命的开始
  - 5个卷积层 + 3个全连接层
  - 60M个参数 + 650K个神经元
  - 2个分组 → 2个GPU（3GB）
    - 使用两块GTX 580 GPU训练了5~6天
  - 新技术
    - ReLU非线性激活
    - Max pooling池化
    - Dropout regularization

还包括后来不被认可的LRN。

AlexNet网络

AlexNet网络



卷积神经网络（CNN）

AlexNet网络



- 输入层: 224 x 224 x 3
- 卷积层1: 96 x 11 x 11 x 3 → LRN → MP → ReLU
- 卷积层2: 256 x 5 x 5 x 48 → LRN → MP → ReLU （通道独立）
- 卷积层3: 384 x 3 x 3 x 256 → ReLU （通道合并，双GPU交互）
- 卷积层4: 384 x 3 x 3 x 192 → MP → ReLU （通道独立）
- 卷积层5: 256 x 3 x 3 x 192 → ReLU （通道独立）
- 全连接层1: 4096 → ReLU
- 全连接层2: 4096 → ReLU
- 全连接层3: 1000 → ReLU
- Softmax层: 1000
- Loss: 标签概率的log值之和 $argmax_w \left\{ \frac{1}{N} \sum -log\left(p(f(x,w) = y(x))\right) \right\}$

AlexNet网络

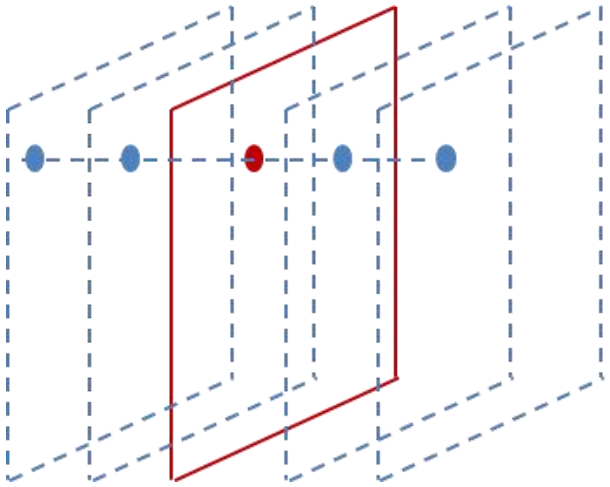

输入图像 （大小 227×227×3）
卷积层 1
227×227×3
卷积核大小 11×11，数量 48 个，步长 4
激活函数（relu）
池化（kernel size=3, stride=2）
标准化
卷积核大小 11×11，数量 48 个，步长 4
激活函数（relu）
池化（kernel size=3, stride=2）
两台 GPU 同时训练，即共 96 个核。
输出特征图像大小：（227 - 11）/ 4 + 1 = 55，即 55×55×96
输出特征图像大小：（ 55 - 3）/ 2 + 1 = 27，即 27×27×96
卷积层 2
27×27×96
卷积核大小 5×5，数量 128 个，步长 1
激活函数（relu）
池化（kernel size=3, stride=2）
标准化
卷积核大小 5×5，数量 128 个，步长 1
激活函数（relu）
池化（kernel size=3, stride=2）
输入特征图像先扩展 2 个像素，即大小 31×31
输出特征图像大小：（ 31 - 5）/ 2 + 1 = 27，即 27×27×256
输出特征图像大小：（ 27 - 3）/ 2 + 1 = 13，即 13×13×256
卷积层 3
13×13×256
卷积核大小 3×3，数量 192 个，步长 1
激活函数（relu）
卷积核大小 3×3，数量 192 个，步长 1
激活函数（relu）
输入特征图像先扩展 1 个像素，即大小 15×15
输出特征图像大小：（ 15 - 3）/1 + 1 = 13，即 13×13×384
卷积层 4
13×13×384
卷积核大小 3×3，数量 192 个，步长 1
激活函数（relu）
卷积核大小 3×3，数量 192 个，步长 1
激活函数（relu）
输入特征图像先扩展 1 个像素，即大小 15×15
输出特征图像大小：（ 15 - 3）/1 + 1 = 13，即 13×13×384
卷积层 5
13×13×384
卷积核大小 3×3，数量 128 个，步长 1
激活函数（relu）
池化（kernel_size=3, stride=2）
卷积核大小 5×5，数量 128 个，步长 1
激活函数（relu）
池化（kernel_size=3, stride=2）
输入特征图像先扩展 1 个像素，即大小 15×15
输出特征图像大小：（ 13 - 3）/1 + 1 = 13，即 13×13×256
输出特征图像大小：（ 13 - 3）/ 2 + 1 = 6，即 6×6×256
全连接 6
6×6×256
2048 个神经元
dropout
2048 个神经元
dropout
共 4096 个神经元
输出 4096×1 的向量
全连接 7
4096×1
2048 个神经元
dropout
2048 个神经元
dropout
共 4096 个神经元
输出 4096×1 的向量
全连接 8
4096×1
1000 个神经元
输出 1000×1 的向量

AlexNet网络

- 局部响应归一化(LRN)
- Local Response Normalization
  - 神经元的侧抑制机制
  - 某个位置（x,y）上沿通道方向上的归一化
  - n为邻域值，N为通道数
  - 超参数： k=2, n=5, α=0.0001, β=0.75

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} \left(a^j_{x,y}\right)^2 \right)^{\beta}$$

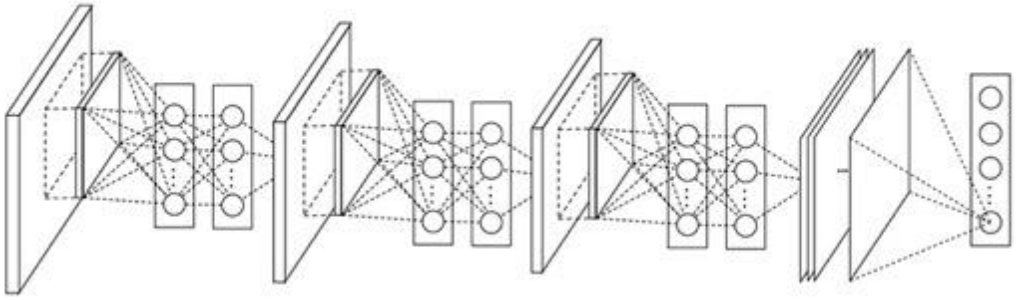
提出VGGNet的论文认为此技术既没有用，又浪费开销。



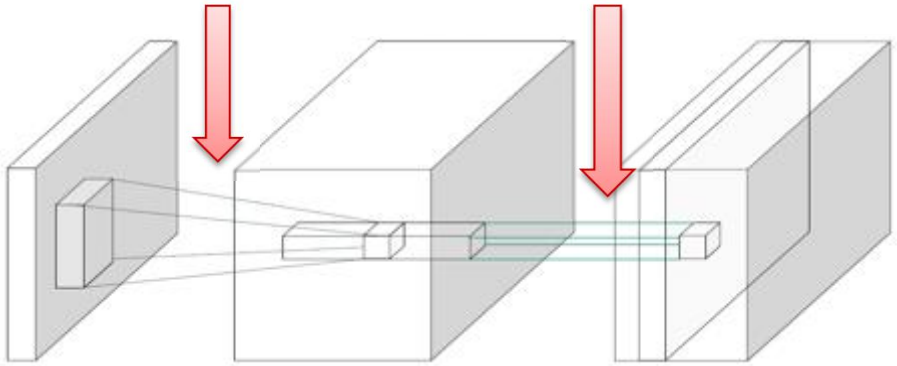
Network-in-Network网络

Network-in-Network网络（NiN）

- 提高CNN的局部感知区域的非线性



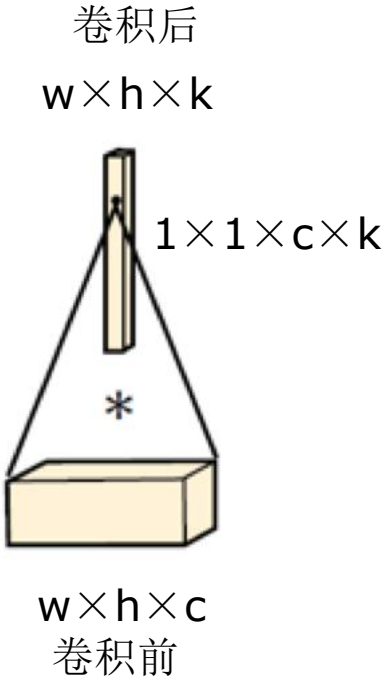
- 卷积层→1x1卷积层→Max池化层



源自2014年ICLR的一篇论文，Alexnet网络参数大小是230M，采用NiN的算法才29M。

1×1卷积

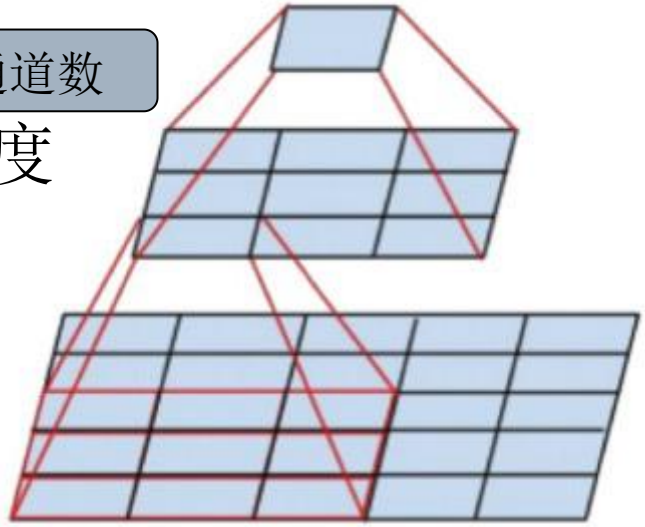
- 1×1卷积，实现的是"同一个像素点上"的各个通道的值得线性组合。
- 这里的"全连接"指的是卷积前通道数与卷积后通道数之间的。
  - $f(a_{11}x_1+a_{12}x_2+\ldots+a_{1c}x_c)=y_1$
  - $f(a_{21}x_1+a_{22}x_2+\ldots+a_{2c}x_c)=y_2$
  - ……
  - $f(a_{k1}x_1+a_{k2}x_2+\ldots+a_{kc}x_c)=y_k$
- 1×1的卷积有两个方面的作用
  - 实现跨通道的交互和信息整合
  - 进行卷积核通道数的降维和升维



VGG网络

VGG网络

- 提出目的是为了探究在大规模图像识别任务中，卷积网络深度对模型精确度有何影响。
- 一个大卷积核分解成连续多个小卷积核
  - 核分解：7x7核 → 3个3x3核（由ReLU连接）
  - 参数数量：$49C^2$ → $27C^2$
- 减少参数，降低计算，增加深度
- ImageNet-2014竞赛第二
- 网络改造的首选基础网络



VGG网络

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	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	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	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	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	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					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

- 从左至右每一列代表着深度增加的不同的模型，从上至下代表模型的深度；
- 其中：conv<滤波器大小>-<通道数>
- 其中E（即VGG19）是较为常见的VGG架构；
- 随着层数的增加A→E，参数增加的并不是很多

GoogLeNet

GoogLeNet出现之前，主流的网络结构突破大致是网络更深（层数），网络更宽（神经元数），但现在看来这纯粹是增大网络的缺点。

GoogLeNet网络

- ImageNet-2014竞赛第一
- 进化顺序
  - Inception V1 → Inception V2 → Inception V3 → Inception V4
- 为了提升性能
  - 减少参数，降低计算
  - 增加宽度、深度

Inception：开端

GoogLeNet

Inception V1网络

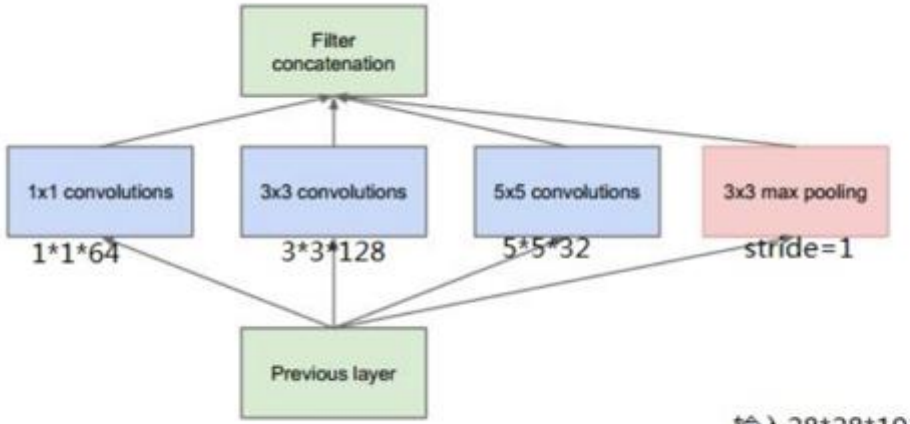
- 核心组件Inception Architecture
  - Split-Merge→1x1卷积, 3x3卷积, 5x5卷积, 3x3池化
    - 增加网络对多尺度的适应性
    - 增加网络宽度
  - Bottleneck Layer→使用NiN的1x1卷积进行特征降维
    - 大幅降低计算量
- 取消全连接
  - 参数量大，减负
- 辅助分类器
  - 解决前几层的梯度消失问题

Inception V1

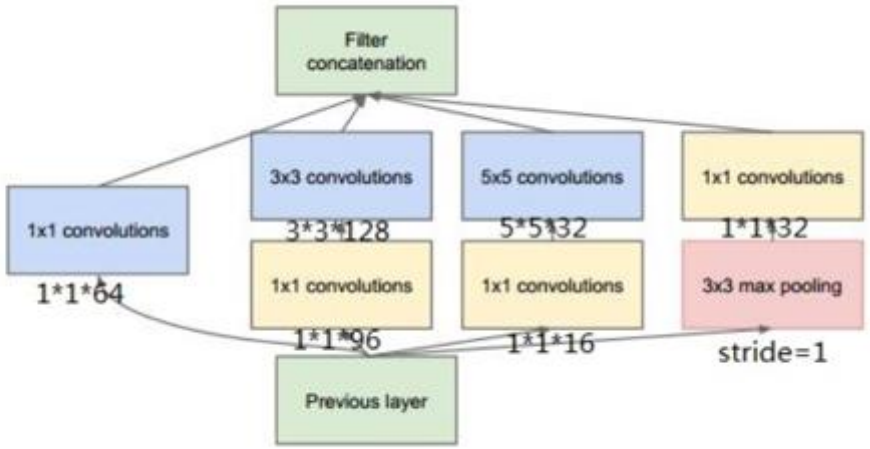
```
concatenated_tensor = tf.concat(3, [branch1, branch2, branch3, branch4])
```

Inception V1网络

- 核心组件Inception Architecture （稀疏连接结构）
- 1*1，3*3，5*5的卷积和3*3的pooling组合在一起
- 亮点是从NIN中引入了1*1 卷积核



(a) Inception module, naïve version



(b) Inception module with dimension reductions

Inception V1

- 上图中：假设前一层的输出为28*28*192，则，
- a的weights数量为：
  1*1*192*64+3*3*192*128+5*5*192*32=387072
- a的输出feature map大小为：
  28*28*64+28*28*128+28*28*32+28*28*192=28*28*416
- b的weights数量为：
  1*1*192*64+(1*1*192*96+3*3*96*128)+(1*1*192*16+5*5*16*32)+1*1*192*32=163328
- b的输出feature map大小为：
  28*28*64+28*28*128+28*28*32+28*28*32=28*28*256
- 可以看出1*1 conv一方面减少了weights，另一方面降低了dimension。

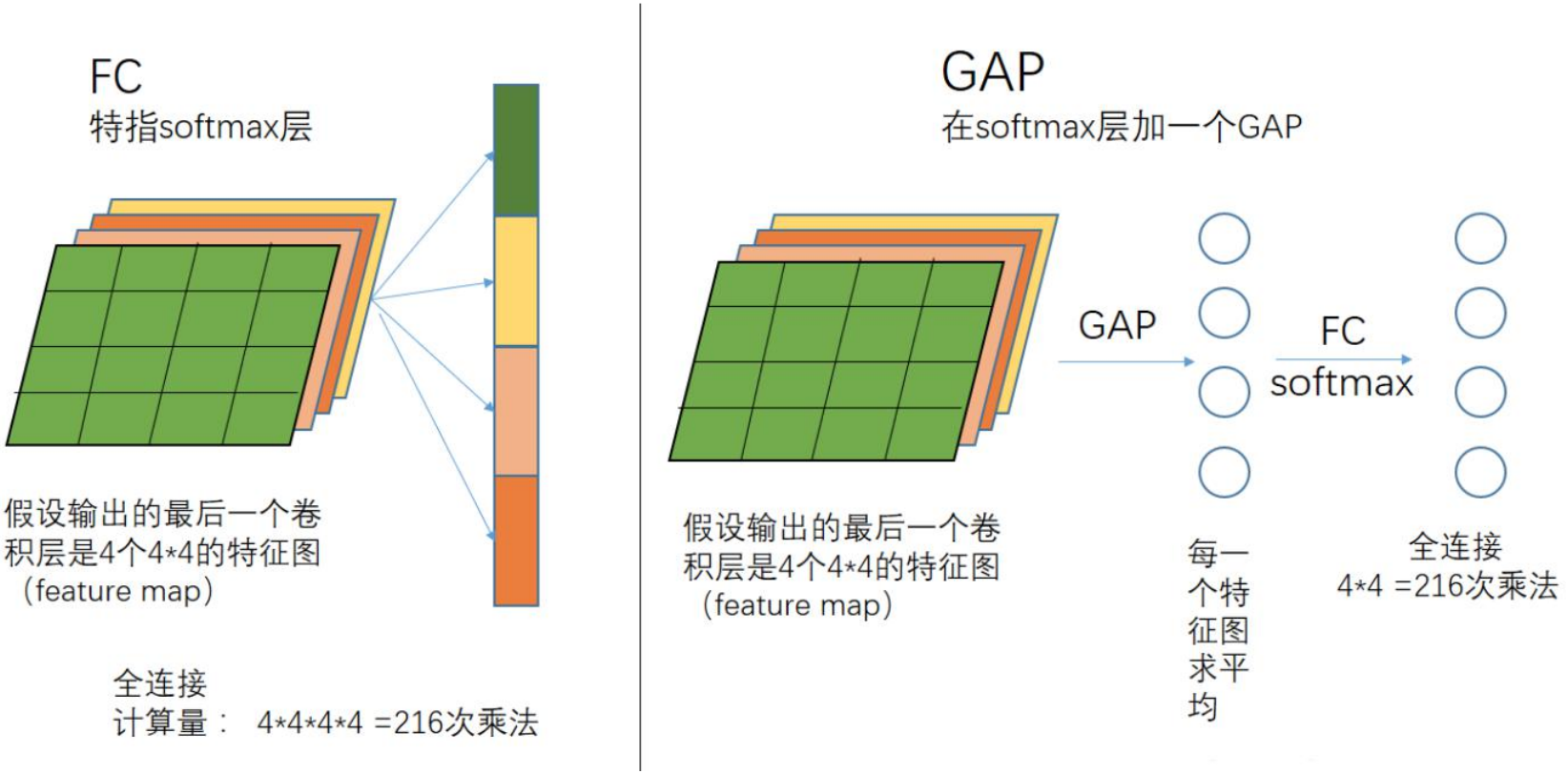
Inception V1

Inception V1网络

- 取消全连接层
  - 本质上是一个全尺寸的卷积层
  - 全连接层占用了大量参数
    - AlexNet: 58.6M (6x6x256x4096 + 4096x4096 + 4096x1000)
    - VGG: 72M (7x7x256x4096 + 4096x4096 + 4096x1000)
  - 由全局平均池化替代（Global average pooling）
    - 输入：7x7x1024
    - 输出：1x1x1024
  - 一大趋势

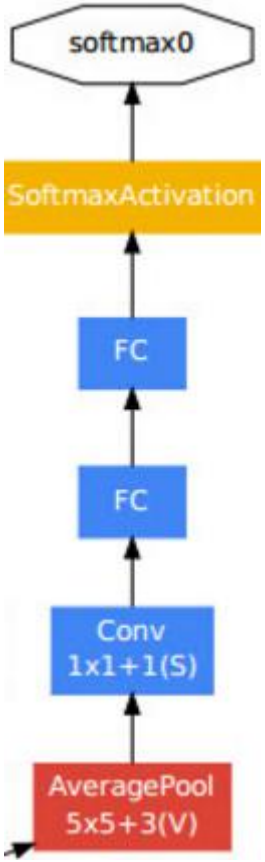
全局平均池化

- 全局平均池化就没有size，它针对的是整张feature map.



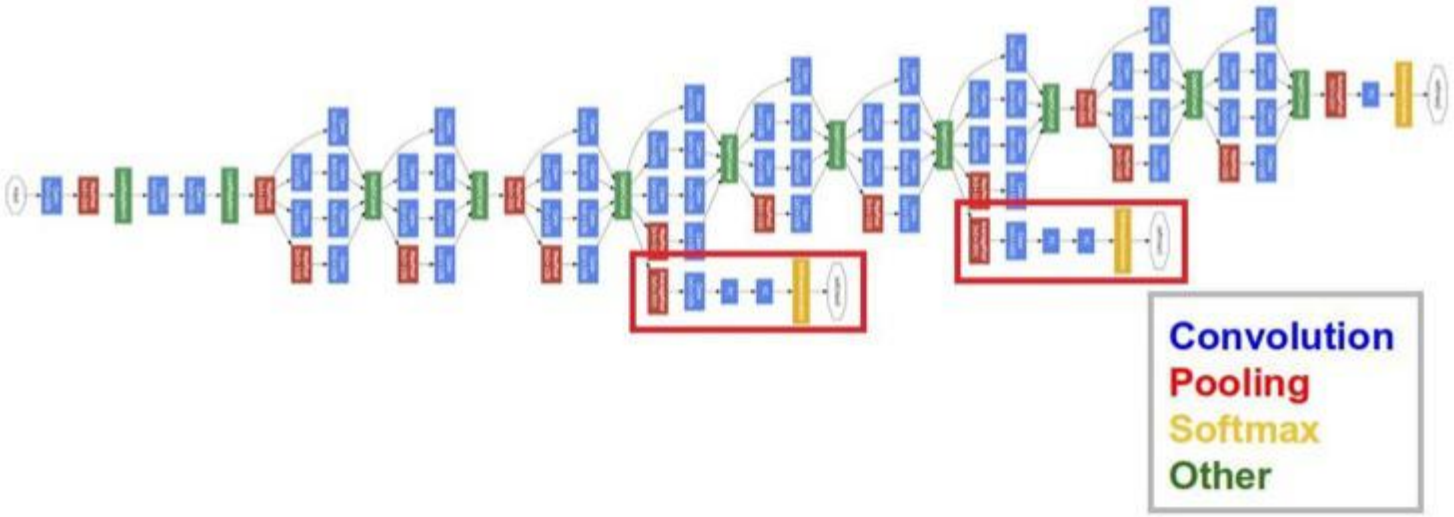
GoogLeNet

Inception V1网络



2个辅助分类器

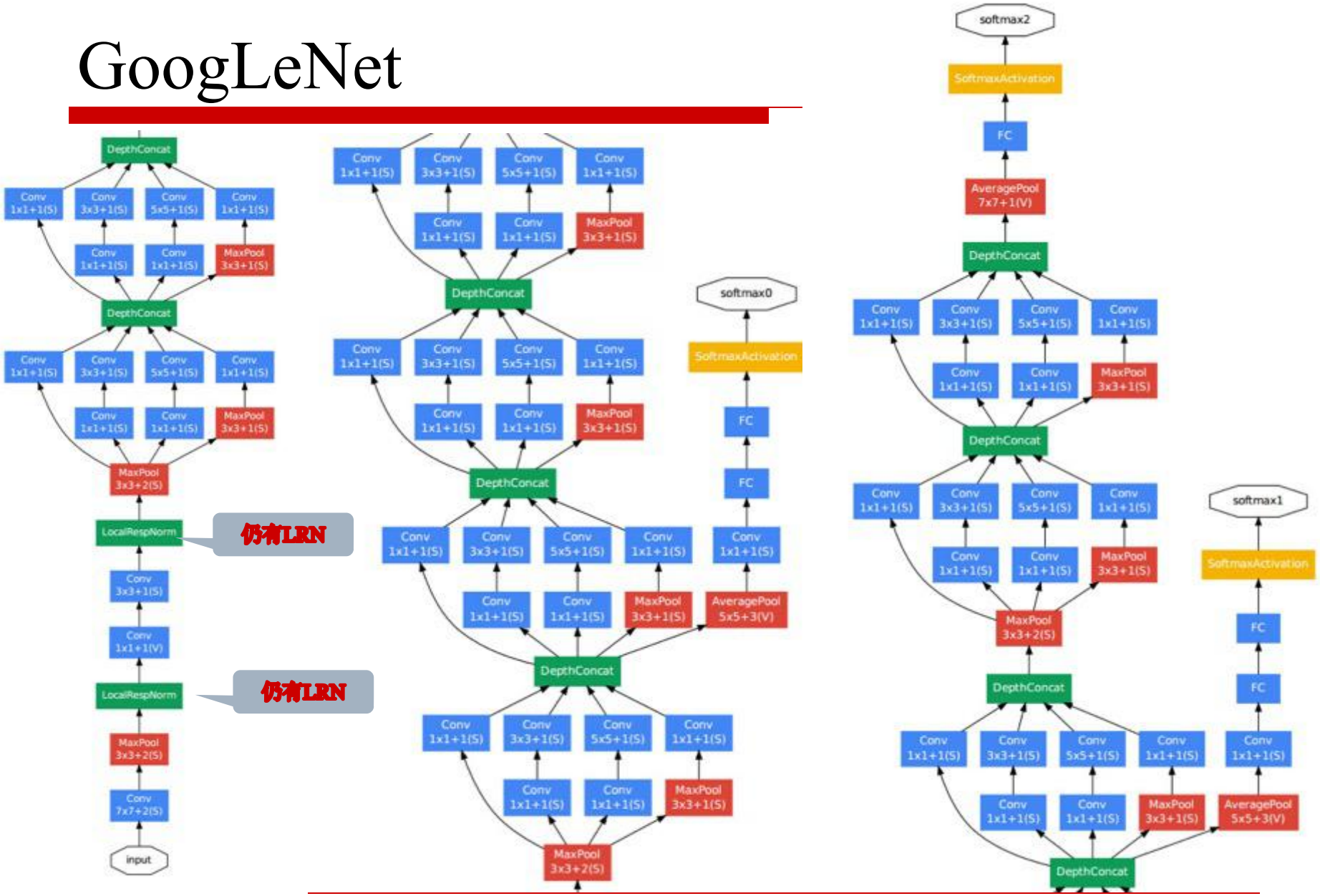
- 深网络中，梯度回传到最初几层，存在严重消失问题
- 有效加速收敛
- 测试阶段不使用



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								

GoogLeNet

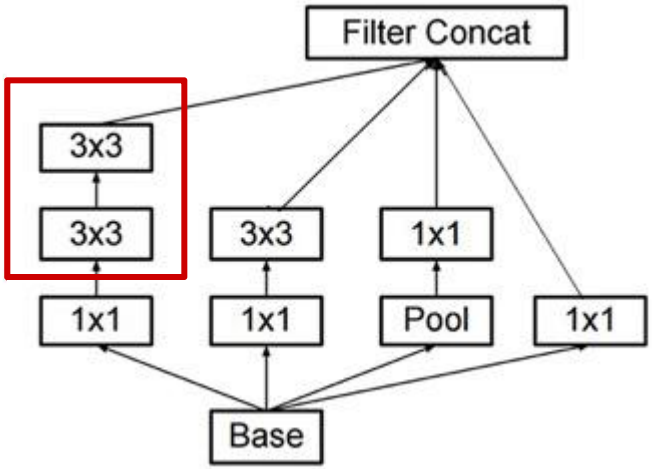

input
Conv 7x7+2(S)
MaxPool 3x3+2(S)
Conv 1x1+1(V)
Conv 3x3+1(S)
LocalRespNorm
仍有LRN
LocalRespNorm
仍有LRN
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
MaxPool 3x3+1(S)
softmax0
SoftmaxActivation
FC
AveragePool 5x5+3(V)
softmax1
softmax2
AveragePool 7x7+1(V)

Inception V2网络

Inception V2网络

- 核心组件
  - Batch Normalization（批归一化）
    - 白化：使每一层的输出都规范化到N(0, 1)
    - 解决Internal Covariate Shift问题
    - 允许较高学习率
    - 取代部分Dropout
  - 5x5卷积核→2个3x3卷积核

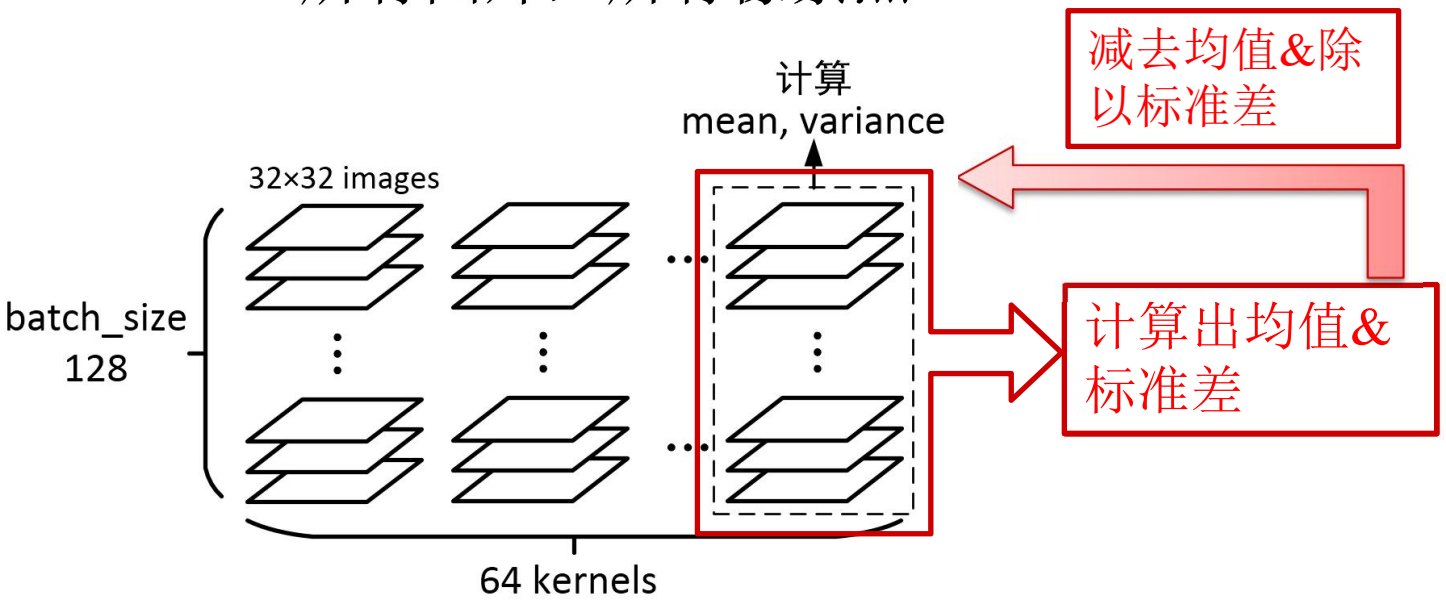
最新的论文认为BN对解决Internal Covariate Shift没有帮助，它起作用的原理是使"解空间"更加平滑。



Inception V2网络

Inception V2网络

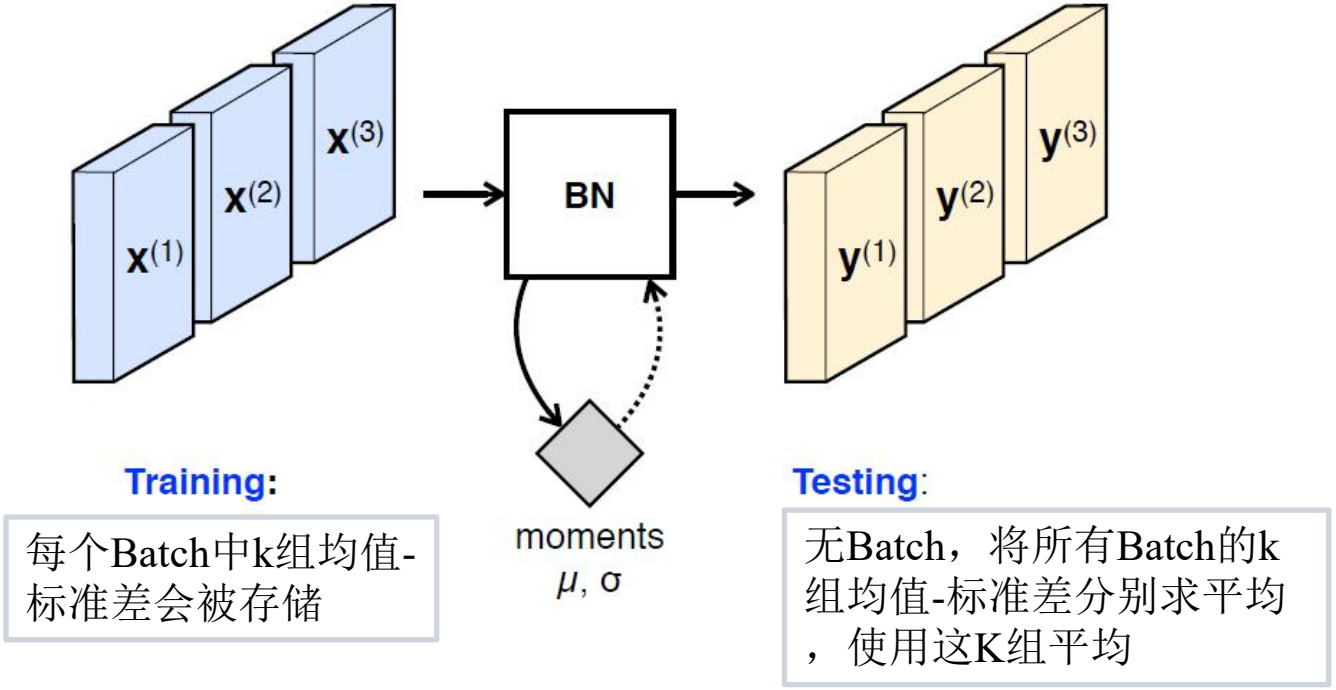
- Batch Normalization批归一化
  - 在batch范围内，对每个特征通道分别进行归一化
  - 所有图片，所有像素点



Inception V2网络

Inception V2网络

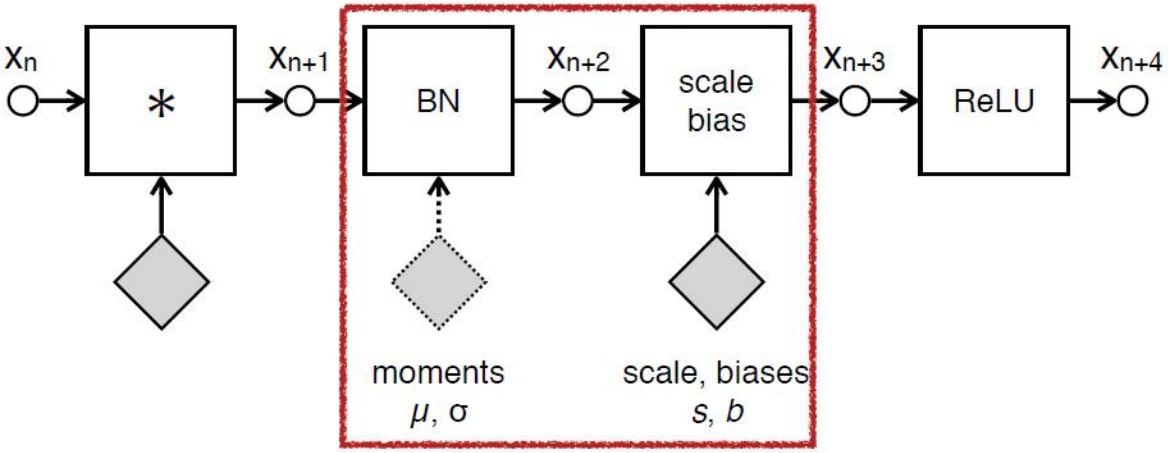
- 训练阶段→实时计算
- 测试阶段→使用固定值（对训练求平均）



Inception V2网络

Inception V2网络

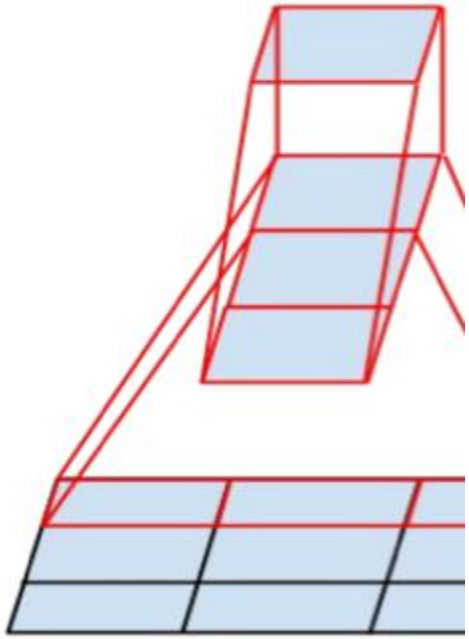
- 位置：卷积→BN→ReLU
- 配对使用scale & shift
  - 添加一组逆算子：scale乘子, bias偏置
  - 这组参数需要学习 $y^{(k)} = \gamma^{(k)}\widehat{x}^{(k)} + \beta^{(k)}$



Inception V3网络

Inception V3网络

- 核心组件
  - 非对称卷积：
  - N x N 分解成 1 x N → N x 1
  - 降低参数数量和计算量

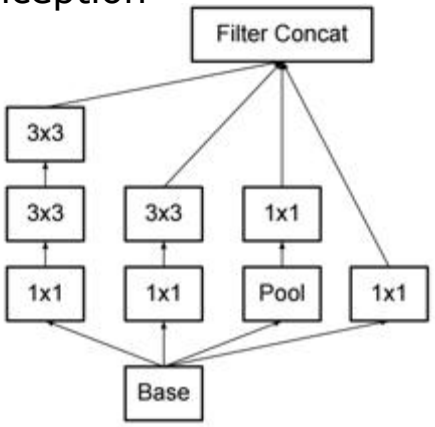


- 在中度大小的feature map上使用效果才会更好，对于m*m大小的feature map,建议m在12到20之间。

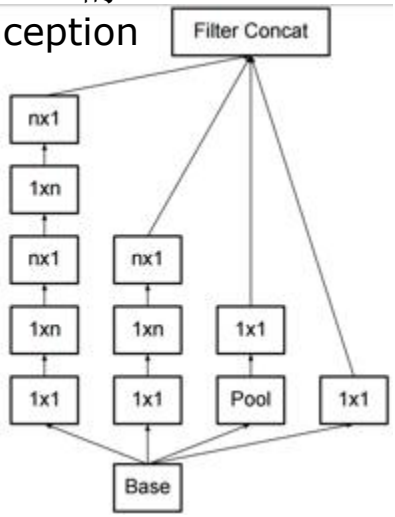
Inception V3网络

- Inception V3优化了Inception Module的结构，V3有三种不同的结构，如下图。
- 这些Inception Module只在网络的后部出现，前部还是普通的卷积层。

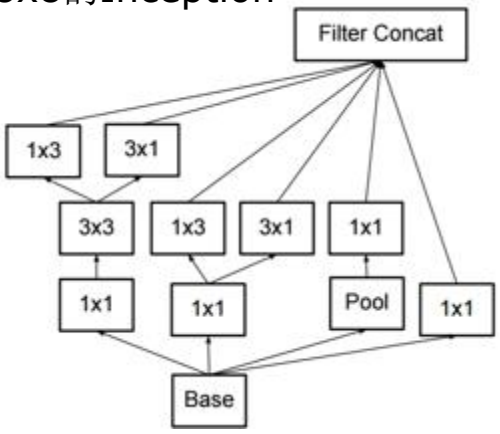
- 35x35的Inception



- 17x17的Inception



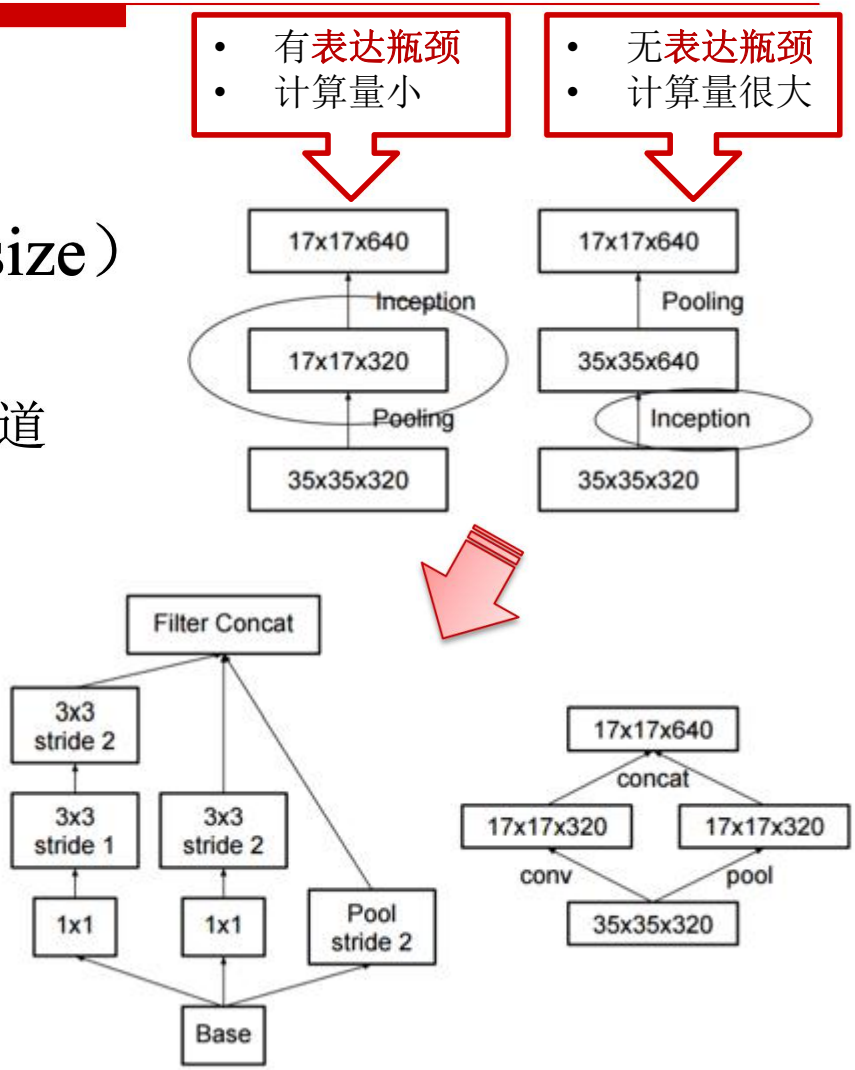
- 8x8的Inception



Inception V3网络

Inception V3网络

- 高效的降尺寸（Grid size）
  - 避免表达瓶颈
    - 降尺寸前增加特征通道
  - 2个并行分支
    - 卷积分支+池化分支
    - 串接分支结果

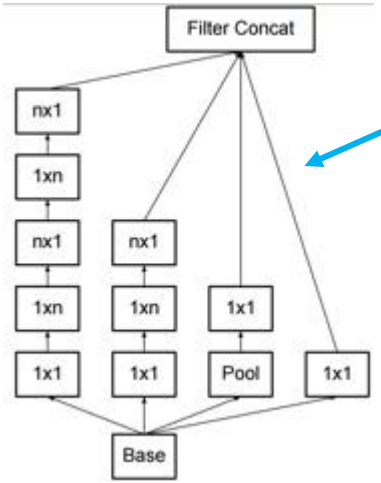


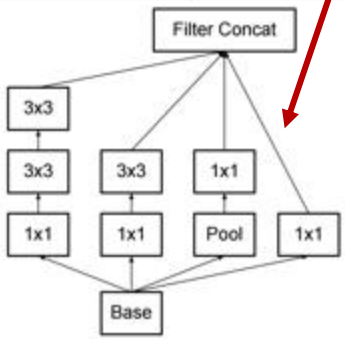
Inception V3网络

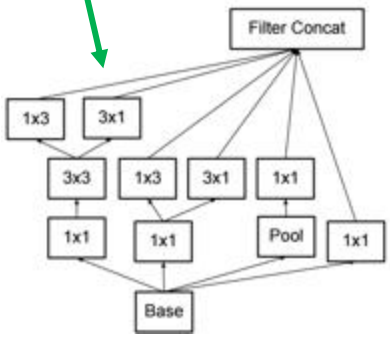
Inception V3网络

- 不增加计算量
  - 避免表达瓶颈
  - 增强结构（表达力）
    - 宽度、深度

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 5	35×35×288
5×Inception	As in figure 6	17×17×768
2×Inception	As in figure 7	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000



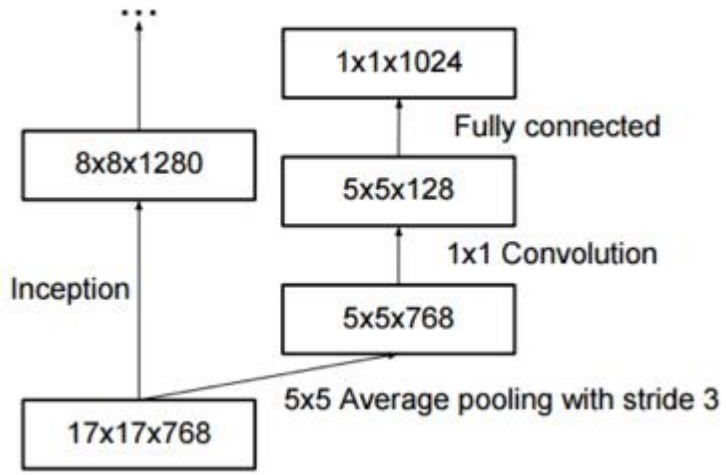




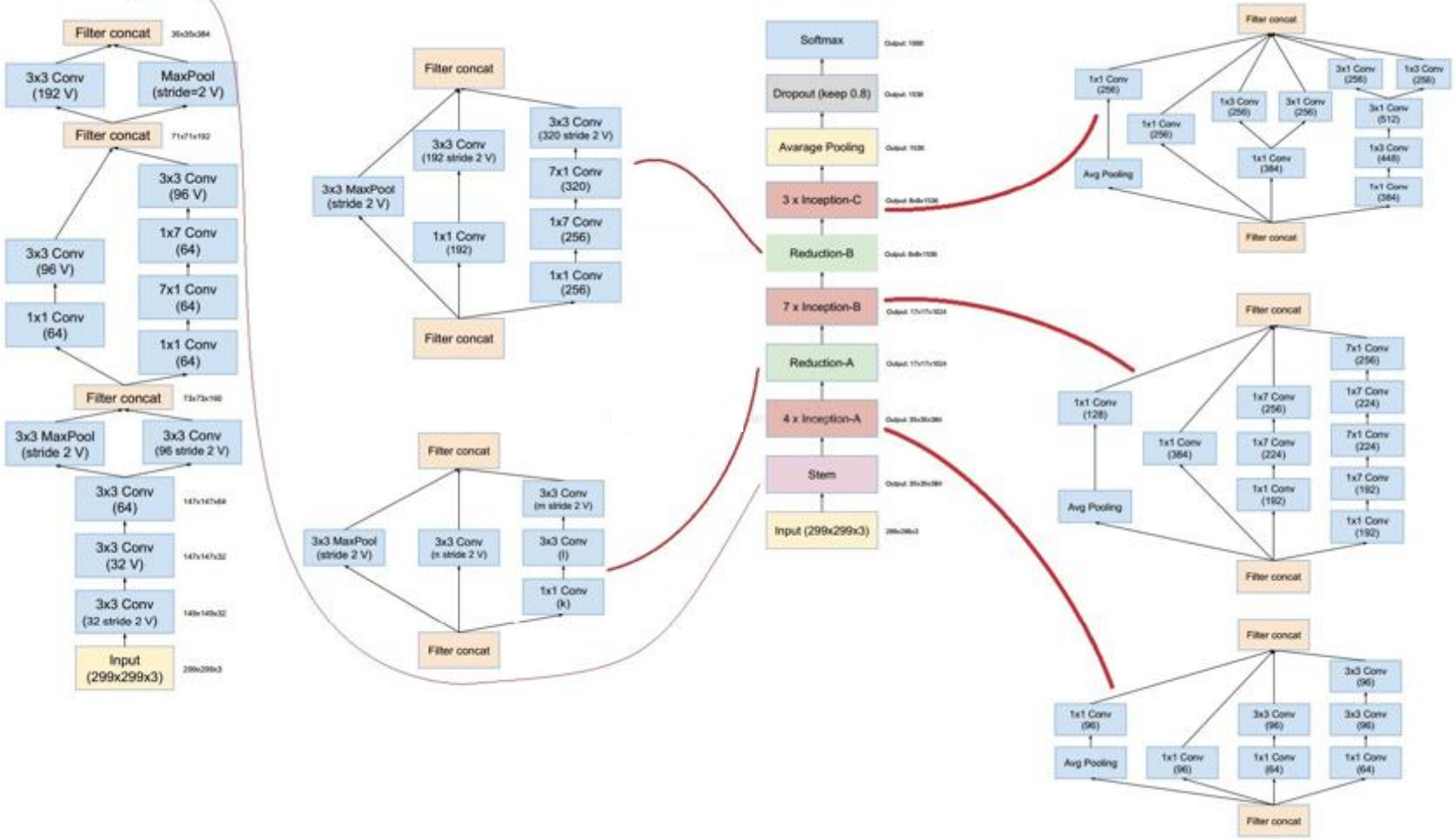
Inception V3网络

Inception V3网络

- 取消浅层的辅助分类器
  - 完全无用
- 深层辅助分类器只在训练后期有用
  - 加上BN和Dropout，主分类器Top1性能提升0.4%
  - 正则化作用
  - 用在最后一层17x17后



Inception V4网络



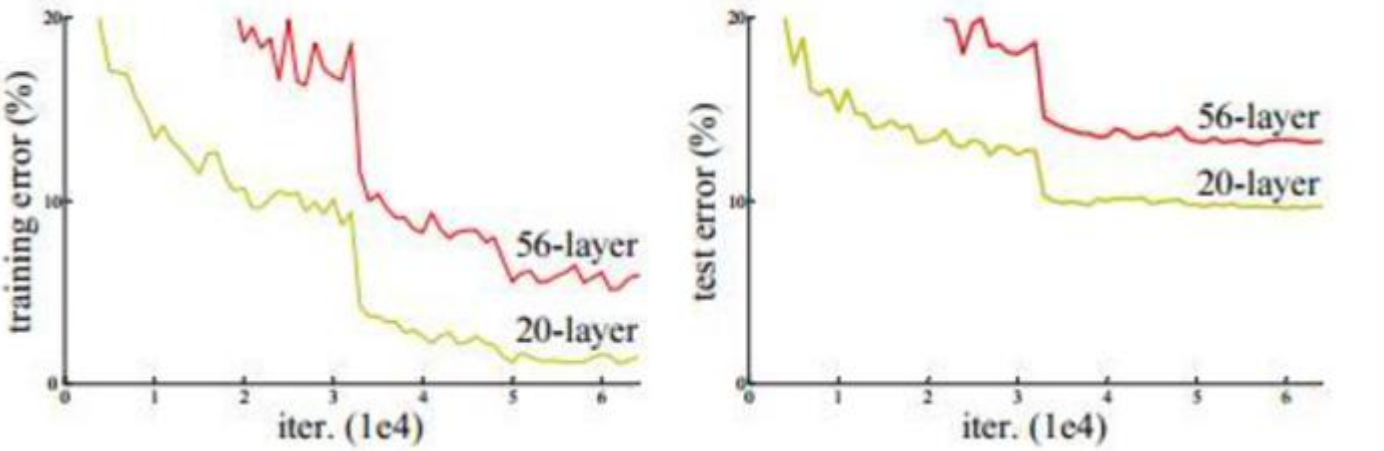
总结一下Inception

- 代替人工确定卷积层中的过滤器类型或者确定是否需要创建卷积层和池化层；
- 不需要人为的决定使用哪个过滤器，是否需要池化层等，由网络自行决定这些参数；
- 即：预先给网络添加所有可能值，将输出连接起来，让网络自己学习它需要什么样的参数。

- Inception 网络有个问题：网络的超参数设定的针对性比较强，当应用在别的数据集上时需要修改许多参数，因此可扩展性一般。

ResNet残差网络

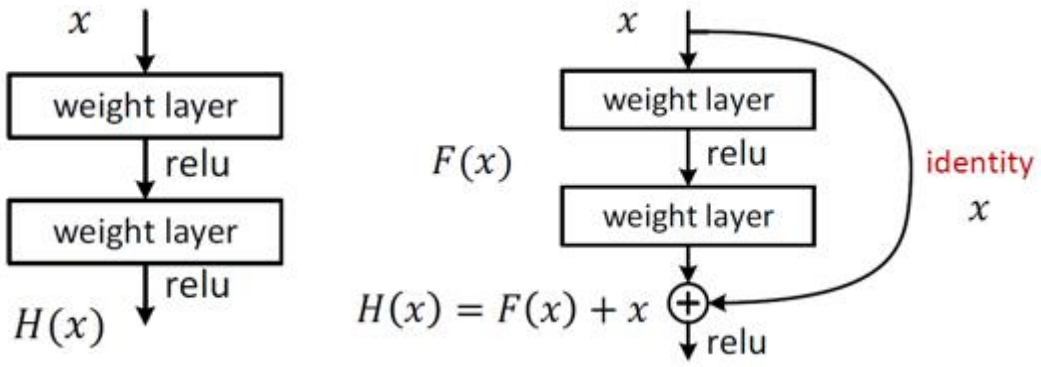
- *Is learning better networks as easy as stacking more layers?*
  - *vanishing/exploding gradients*
  - *degradation*



ResNet残差网络

ResNet残差网络

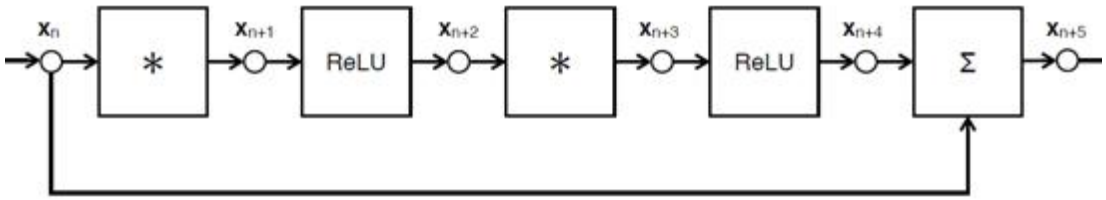
- 核心组件Skip/shortcut connection
  - Plain net: 可以拟合出任意目标映射H(x)
  - Residual net
    - 可以拟合出任意目标映射F(x)，H(x)=F(x)+x
    - F(x)是残差映射，相对于identity来说

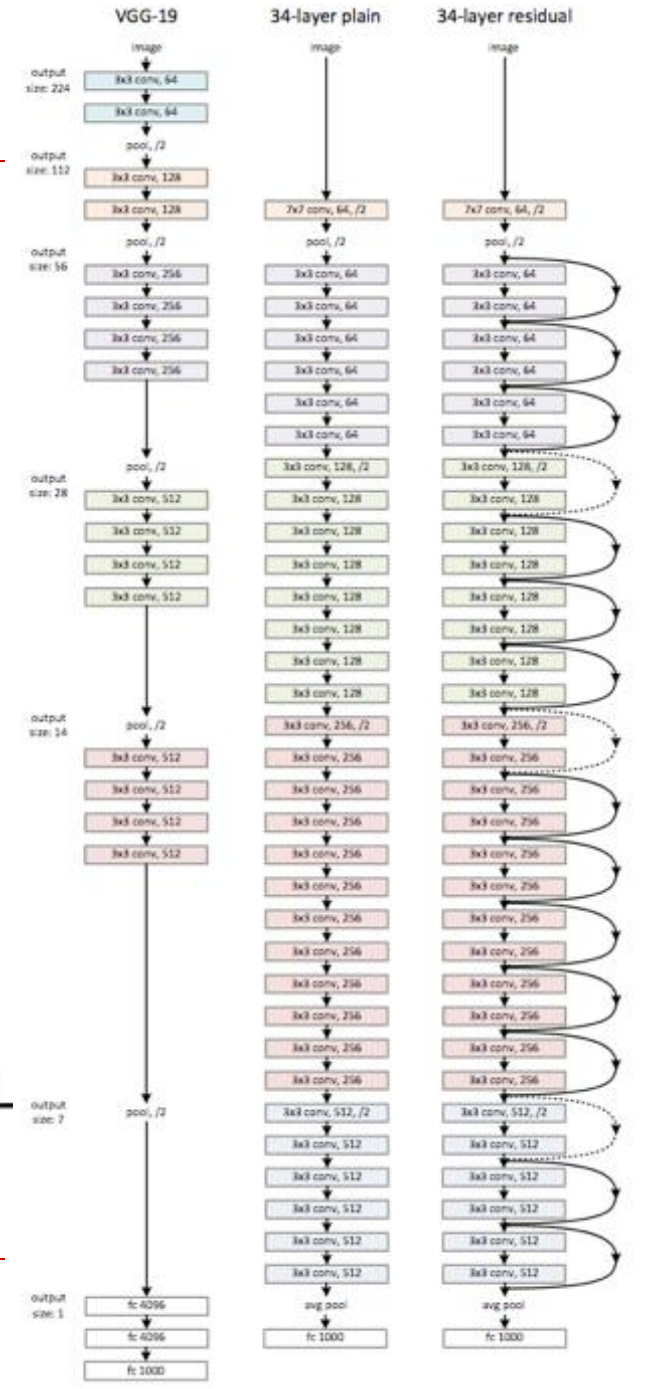


ResNet残差网络

ResNet残差网络

- 其他设计
  - 全是3x3卷积核
  - 卷积步长2取代池化
  - 使用Batch Normalization
  - 取消
    - Max池化
    - 全连接层
    - Dropout

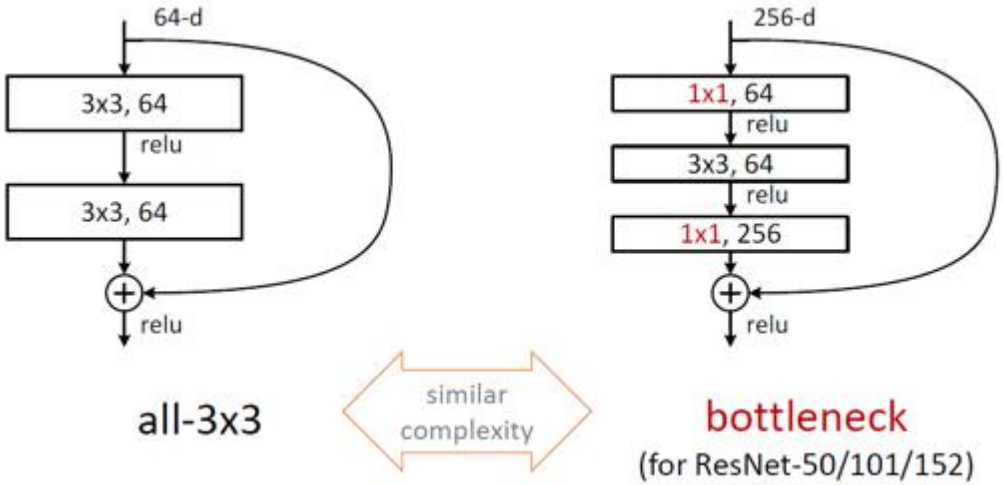




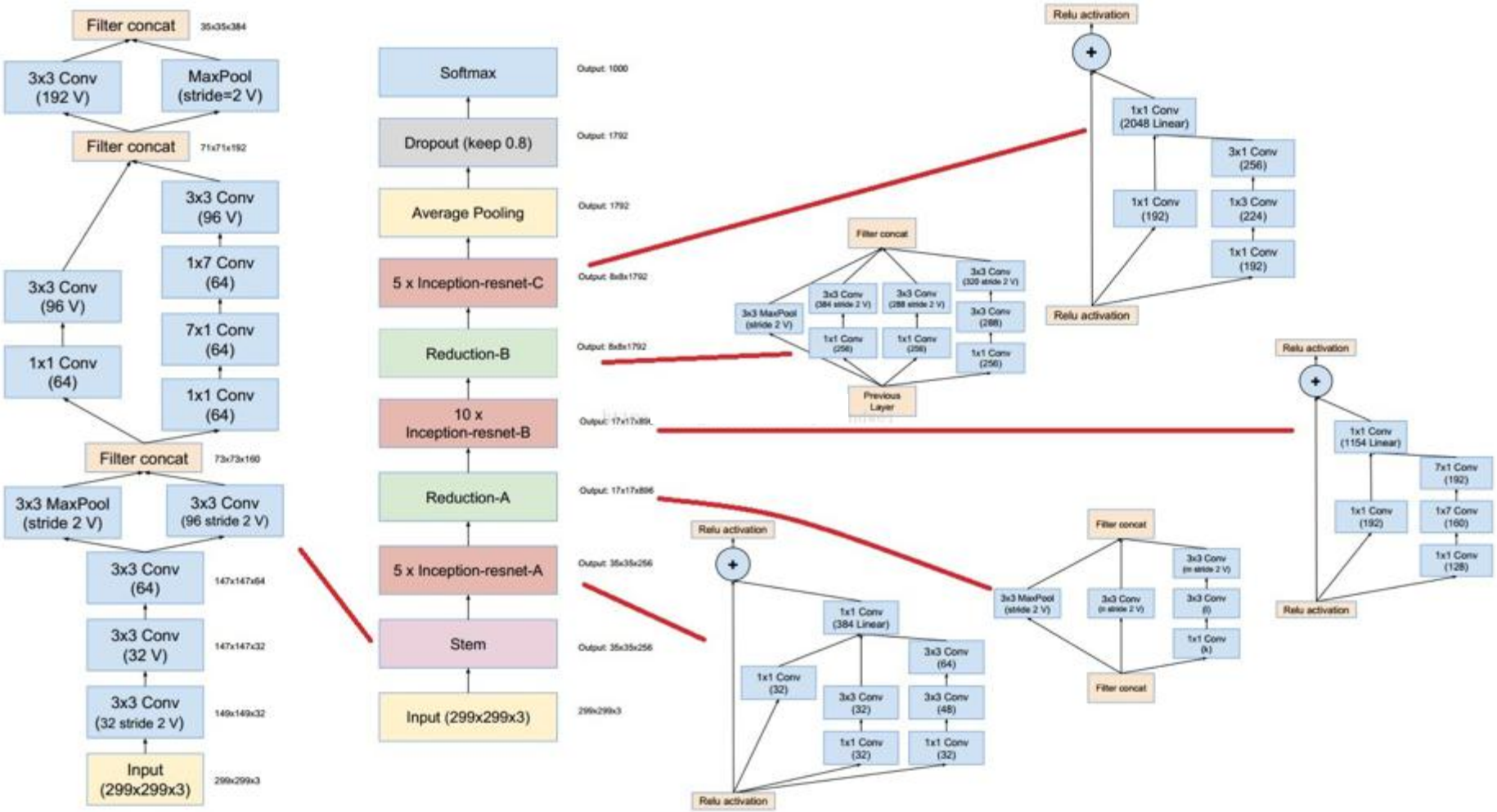
ResNet残差网络

ResNet残差网络

- 更深网络：根据Bootleneck优化残差映射网络
  - 原始：3x3x256x256→3x3x256x256
  - 优化：1x1x256x64→3x3x64x64→1x1x64x256



Inception-ResNet网络



ResNet残差网络

ResNet残差网络

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

ResNeXt网络

ResNeXt网络

- 提出"深"和"宽"之外的第3个维度：
- cardinality（基数）
  - 采用Split-Transform-Aggregate策略
    - 将卷积核按通道分组，形成32个并行分支
    - 低维度卷积进行特征变换
    - 加法合并
- 同参数规模下，增加结构，提高模型表达力
  - 100层ResNeXt = 200层ResNet
  - ILSVRC-2016竞赛第2

ResNeXt网络

resnet的参数量为：

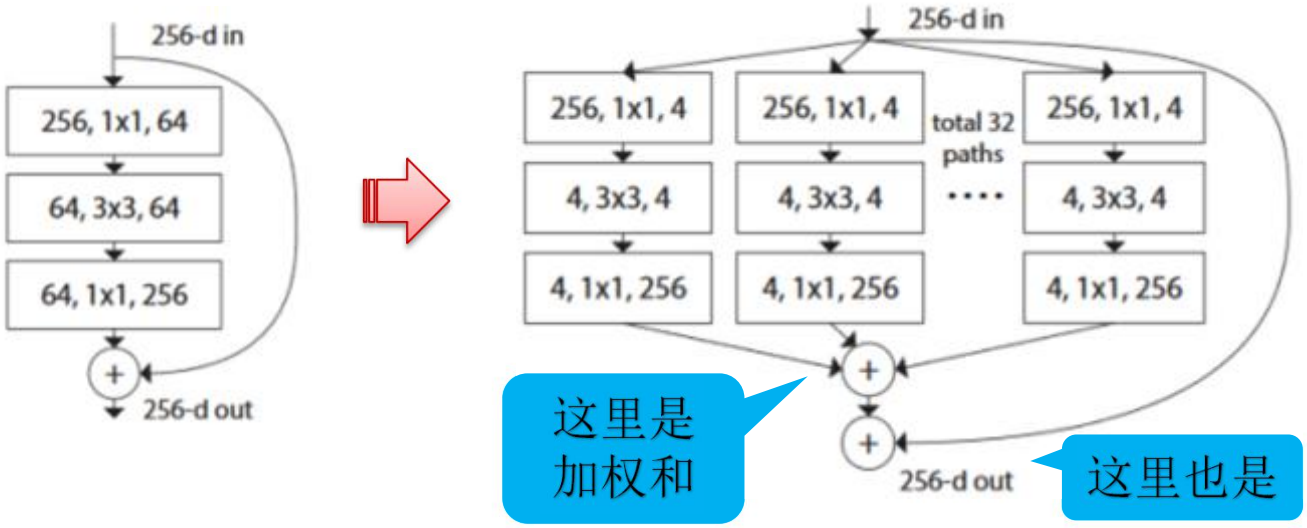
256*64+3*3*64*64+64*256=70k

ResNeXt的参数量为：

C*(256*d+3*3*d*d+d*256) 当C取32，d=4时，上式也等于70k。

ResNeXt网络

- 还记得AlexNet吗？ ResNeXt实际上就是将group convolution引进了ResNet中，以获得更少的参数。
- 32x4d块结构



ResNeXt网络

ResNeXt网络

- 打破或deeper，或wider的常规思路，ResNeXt则认为可以引入一个新维度，称之为cardinality。
- 增加cardinality基数可以不断提高性能

	setting	top-1 error (%)
ResNet-50	1 × 64d	23.9
ResNeXt-50	2 × 40d	23.0
ResNeXt-50	4 × 24d	22.6
ResNeXt-50	8 × 14d	22.3
ResNeXt-50	32 × 4d	**22.2**
ResNet-101	1 × 64d	22.0
ResNeXt-101	2 × 40d	21.7
ResNeXt-101	4 × 24d	21.4
ResNeXt-101	8 × 14d	21.3
ResNeXt-101	32 × 4d	**21.2**

Cardinality: the size of the set of transformations

ResNeXt网络

ResNeXt网络

- 32x4d结构参数
  - 32个分支
  - 每分支4通道
    - Bottleneck width

stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		$\mathbf{25.5}\times10^6$	$\mathbf{25.0}\times10^6$
FLOPs		$\mathbf{4.1}\times10^9$	$\mathbf{4.2}\times10^9$

	224×224		320×320 / 299×299	
	top-1 err	top-5 err	top-1 err	top-5 err
ResNet-101 [14]	22.0	6.0	-	-
ResNet-200 [15]	21.7	5.8	20.1	4.8
Inception-v3 [39]	-	-	21.2	5.6
Inception-v4 [37]	-	-	20.0	5.0
Inception-ResNet-v2 [37]	-	-	19.9	4.9
ResNeXt-101 (**64 × 4d**)	20.4	5.3	**19.1**	**4.4**

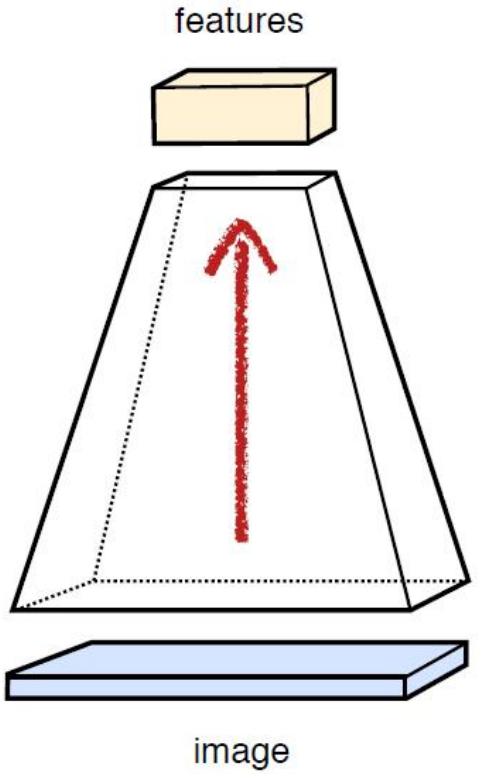
典型CNN的参数与效果对比

模型名	AlexNet	VGG	GoogLeNet v1	ResNet
时间	2012	2014	2014	2015
层数	8	19	22	152
Top-5错误	16.4%	7.3%	6.7%	3.57%
Data Augmentation	+	+	+	+
Inception(NIN)	–	–	+	–
卷积层数	5	16	21	151
卷积核大小	11,5,3	3	7,1,3,5	7,1,3,5
全连接层数	3	3	1	1
全连接层大小	4096,4096,1000	4096,4096,1000	1000	1000
Dropout	+	+	+	+
Local Response Normalization	+	–	+	–
Batch Normalization	–	–	–	+

CNN设计准则

CNN设计准则

- 避免信息瓶颈
  - 卷积过程中
    - 空间尺寸H x W逐渐变小
    - 输出通道数C逐渐变多
  - H x W x C要缓慢变小

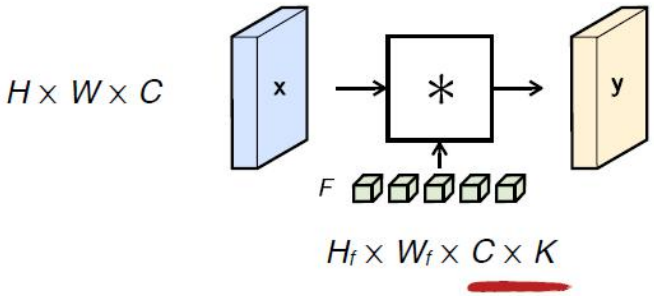


CNN设计准则

CNN设计准则

- 通道（卷积核）数量保持在可控范围内
  - 输入通道数量C
  - 输出通道数量K
  - 参数数量 $H_f \times W_f \times C \times K$
  - 操作数量 $\frac{H \times H_f}{\text{stride}} \times \frac{W \times W_f}{\text{stride}} \times C \times K$

从这儿也可以看出：卷积核的大小对总计算量的影响

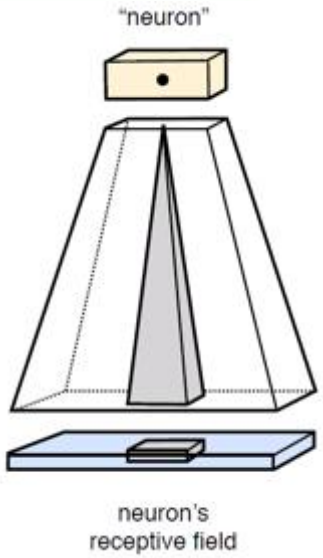


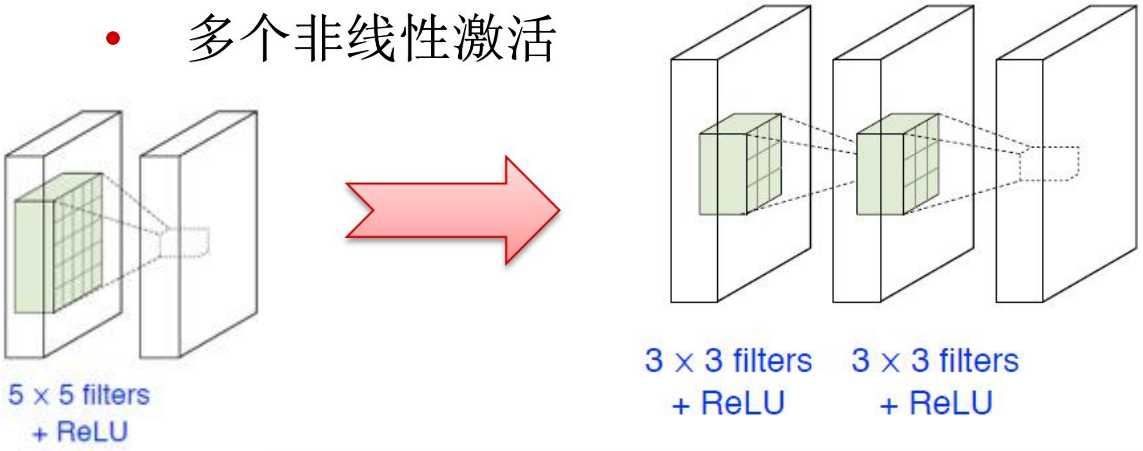
$$\text{complexity} \propto C \times K$$

CNN设计准则

CNN设计准则

- 感受野要足够大
  - 卷积是基于局部图片的操作
  - 捕捉大尺寸内容
  - 多个小尺寸卷积核 vs 一个大尺寸卷积核
    - 参数少，计算快
    - 多个非线性激活

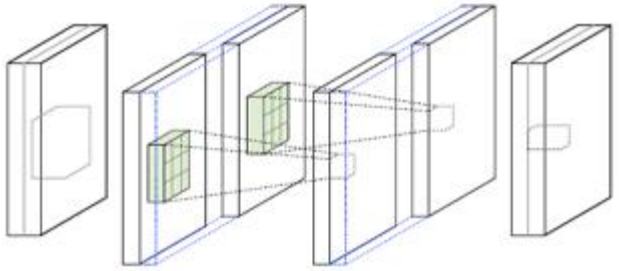




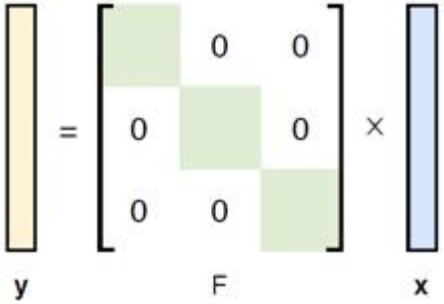
CNN设计准则

CNN设计准则

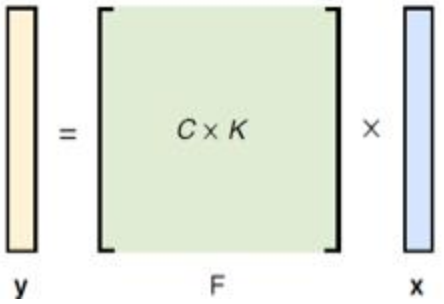
- 分组策略→降低计算量
  - G组M/G个滤波器 vs M个滤波器



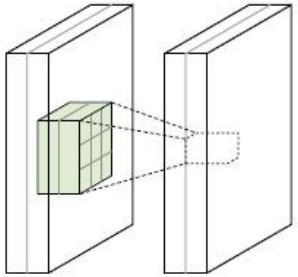
通道拆分 分组卷积 特征合并



complexity: $C \times K / G$



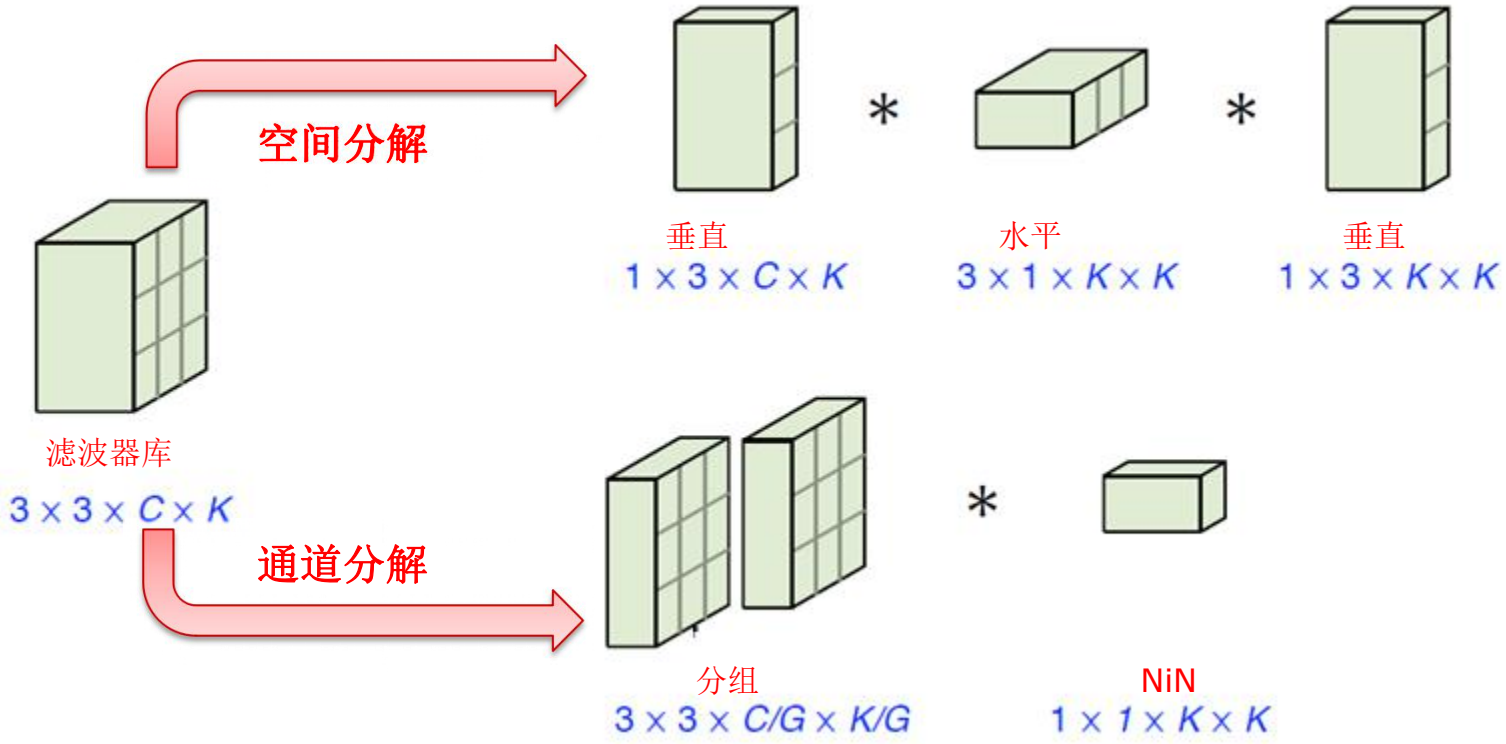
complexity: $C \times K$



CNN设计准则

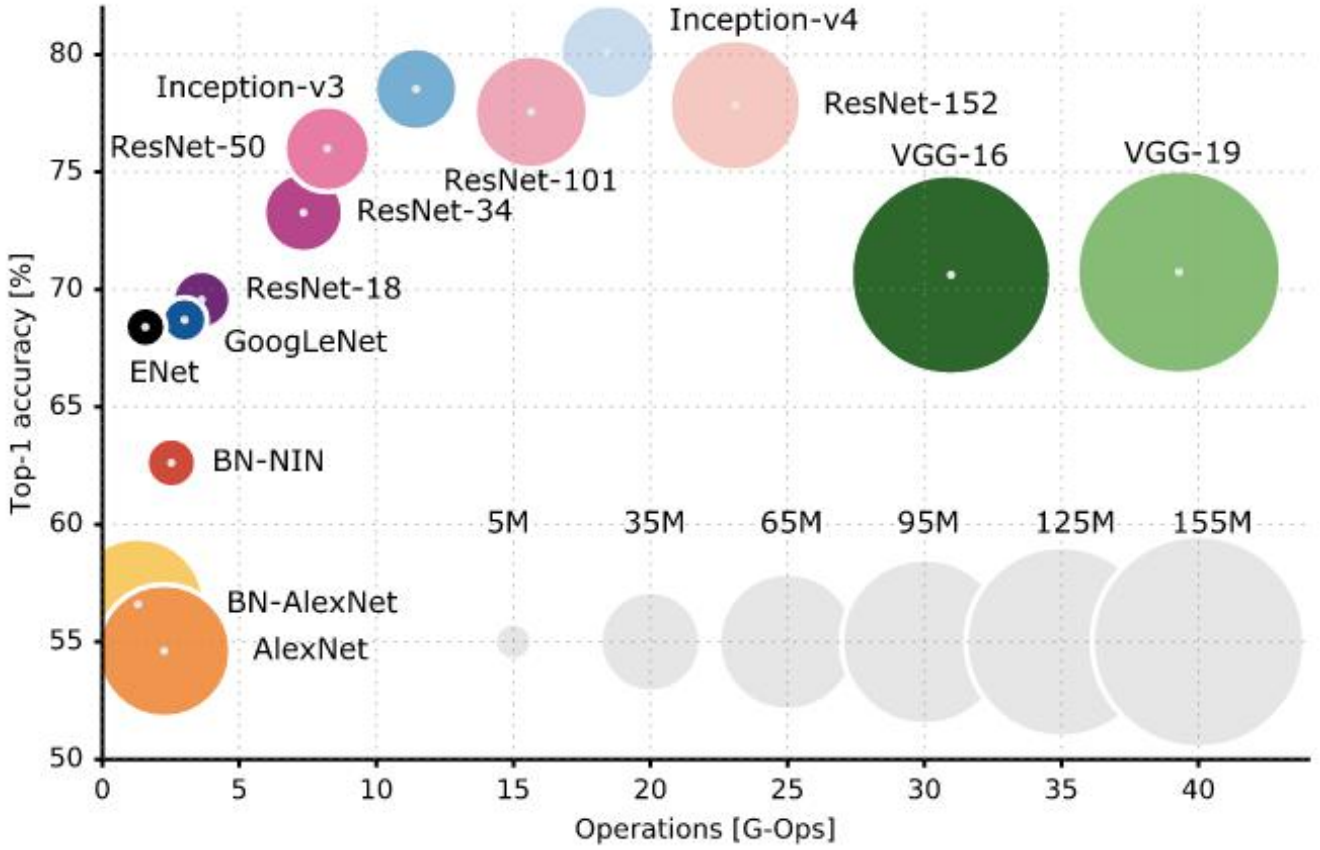
CNN设计准则

- 低秩分解→降低参数&计算量



ImageNet分类

准确性 vs. 执行效率



演示环节

- VGG网络
- ResNet网络

参考资料

- http://baijiahao.baidu.com/s?id=1601882944953788623&wfr=spider&for=pc
- https://blog.csdn.net/guoyunfei20/article/details/78122504
- https://raw.githubusercontent.com/titu1994/Inception-v4/master/Architectures/Inception-v4.png

问答互动

在所报课的课程页面，

1、点击“全部问题”显示本课程所有学员提问的问题。

2、点击“提问”即可向该课程的老师和助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

– 微信公众号：**小象学院**

