

# 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。

□ 课程详情请咨询

■ 微信公众号：小象

■ 新浪微博：ChinaHadoop



# 第4课 图像分类（上）

---

## Image Classification

主讲人：张宗健

悉尼科技大学博士

主要研究方向： 计算机视觉、视觉场景理解、图像&语言、深度学习  
图像检索CbIR、Human ReID等

# 本章结构

---

- 图片分类 (Image Classification)
- ILSVRC 竞赛 (ImageNet Large Scale Visual Recognition Challenge)
- 卷积神经网络 (Convolutional Neural Network)
- 应用案例：
  - 超深的残差神经网络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/2016/index>

# 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）

---

## 网络进化

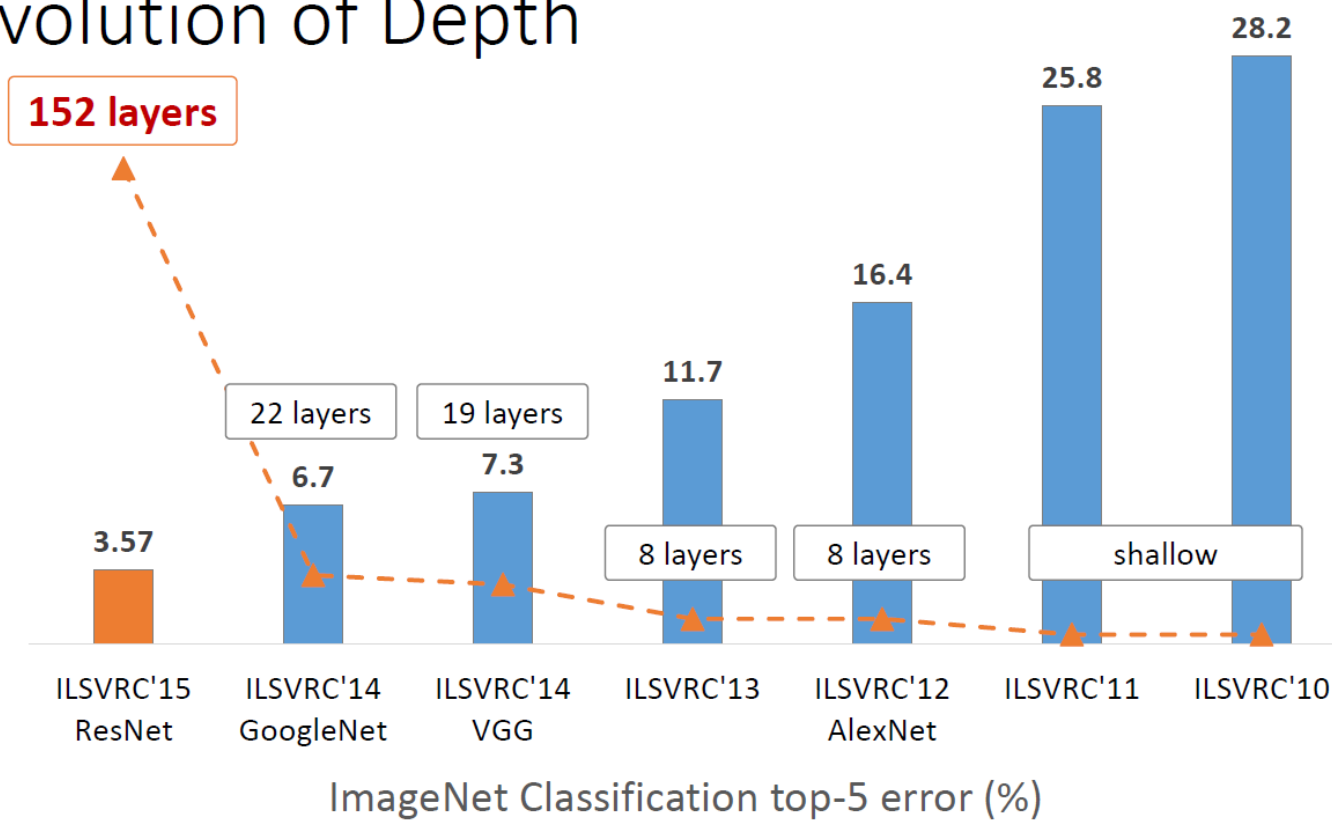
- 网络：AlexNet  $\rightarrow$  VGG  $\rightarrow$  GoogLeNet  $\rightarrow$  ResNet
- 深度：8  $\rightarrow$  19  $\rightarrow$  22  $\rightarrow$  152
- VGG、ResNet结构简洁有效
  - 容易修改，迁移到其他任务中去
  - 高层任务的基础网络
- 性能竞争网络
  - GoogLeNet：Inception v1  $\rightarrow$  v4
    - Split-transform-merge
  - ResNet：ResNet152  $\rightarrow$  ResNeXt
    - 深度、宽度、基数(cardinality)



# 卷积神经网络（CNN）

## ImageNet性能进化

### Revolution of Depth

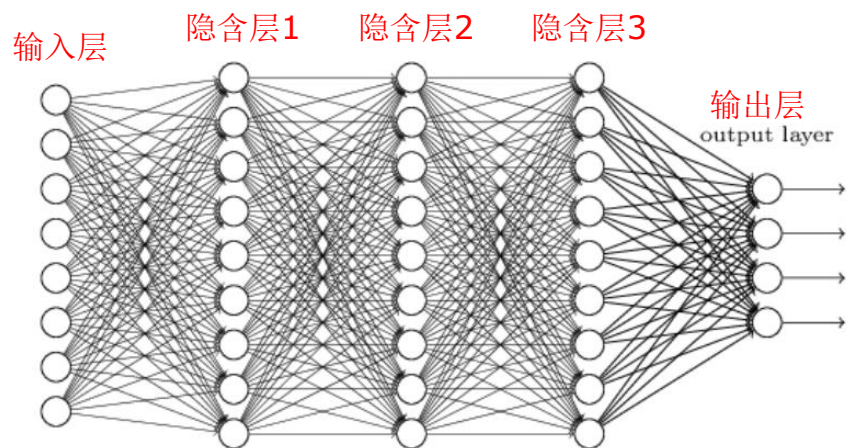
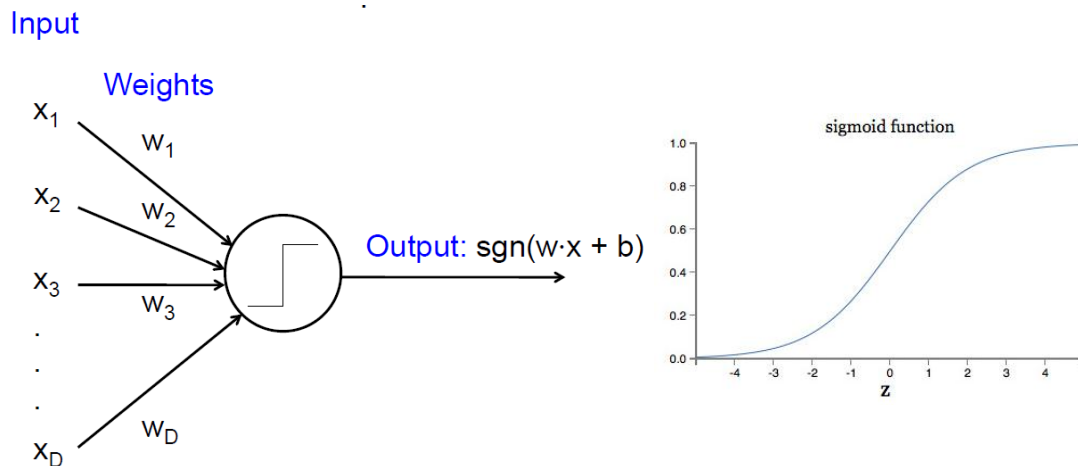


ICCV15

# 卷积神经网络 (CNN)

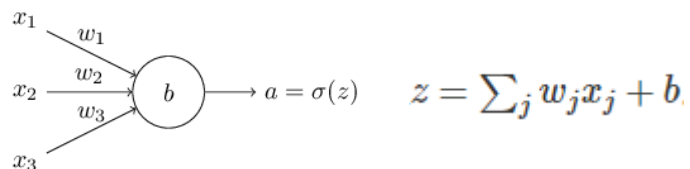
## 基础神经网络

- 神经元
  - 输入向量  $x$
  - 权重向量  $w$
  - 偏置标量  $b$
  - 激活函数 sigmoid
- 浅网络
  - 3~5层
  - 优化
    - 梯度下降
    - BP后向传播 (链式规则)



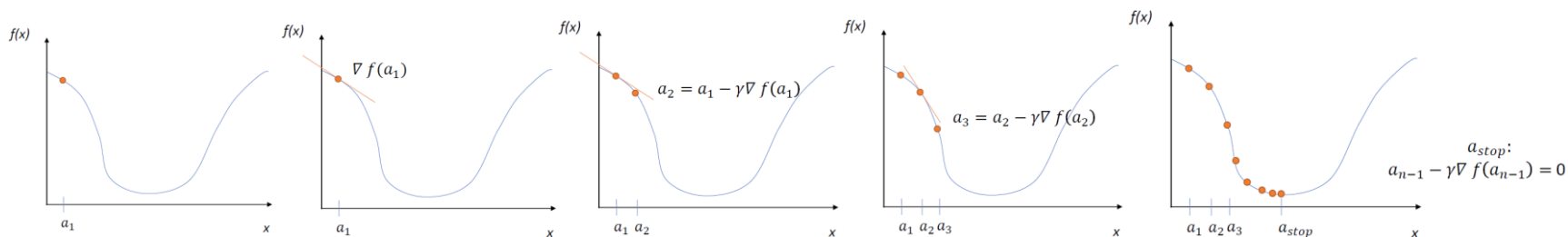
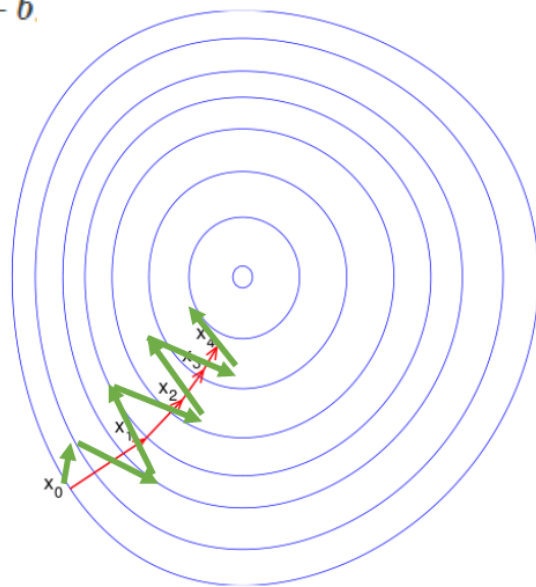
# 卷积神经网络 (CNN)

## 基础神经网络



- 梯度下降优化

- 交叉熵  $C = -\frac{1}{n} \sum_x \sum_j [y_j \ln a_j^L + (1 - y_j) \ln(1 - a_j^L)]$
- 批量梯度下降
- 随机梯度下降
  - 学习率/步长
  - 扰动  $\rightarrow$  动量算法 (momentum)



# 卷积神经网络（CNN）

---

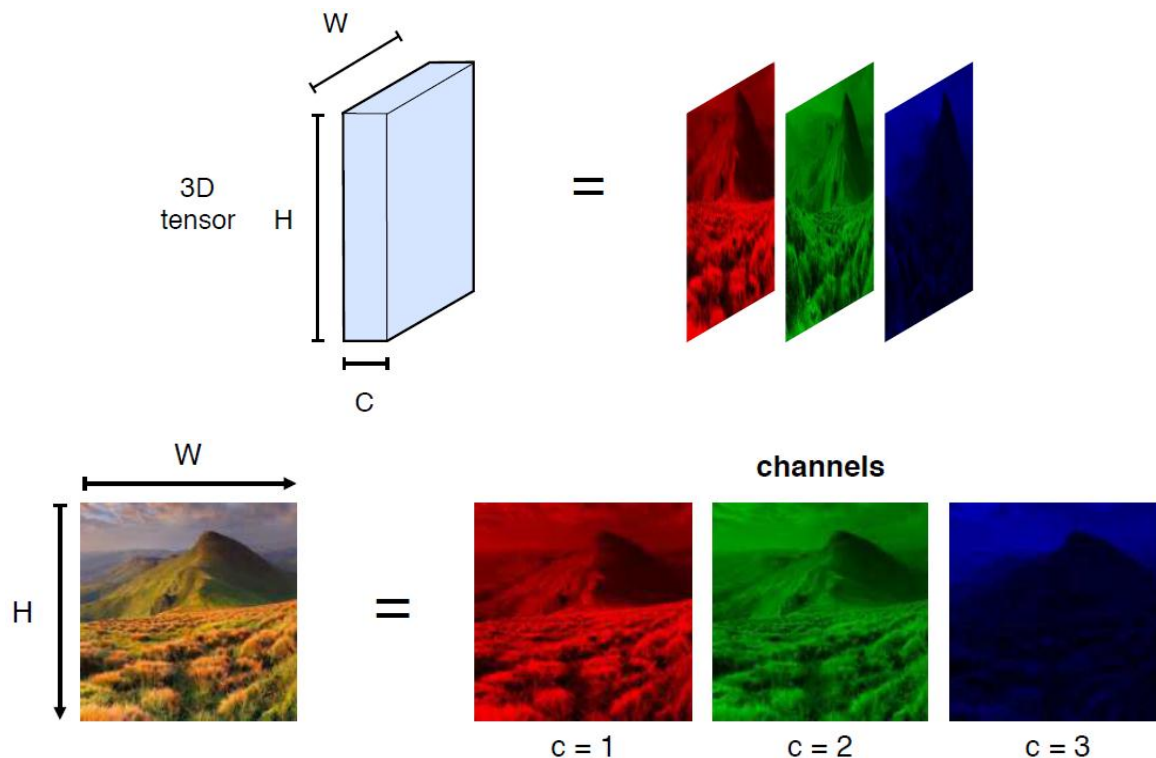
## 构建CNN的基本层

- 卷积层（Convolutional layer）
  - 激活函数（Sigmoid, ReLU, ...）
- 池化层（Pooling layer）
  - 平均池化（Average pooling）
  - 最大化池化（Max pooling）
- 全连接层（Fully-Connected layer）
- 批归一化层（Batch Normalization layer）

# 卷积神经网络 (CNN)

## CNN卷积层

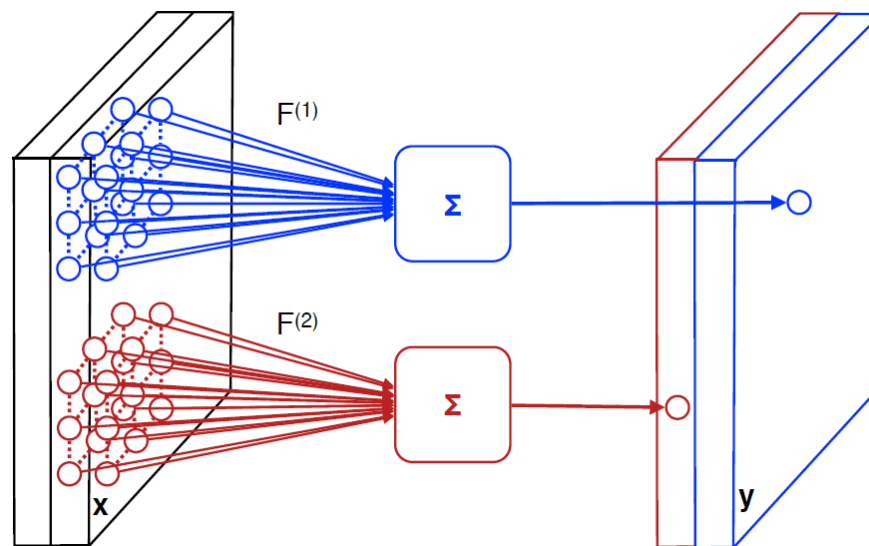
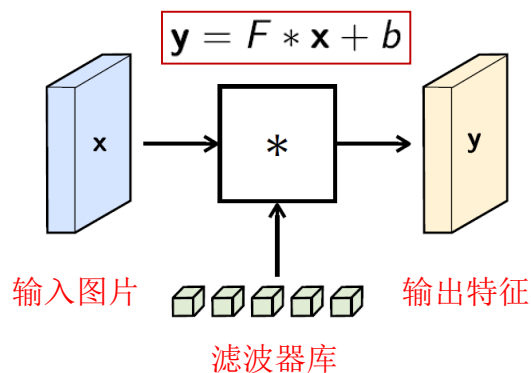
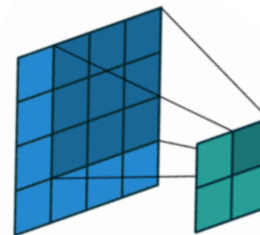
- 3通道 (RGB) 输入图片  $\rightarrow$  3D tensor



# 卷积神经网络（CNN）

## CNN卷积层

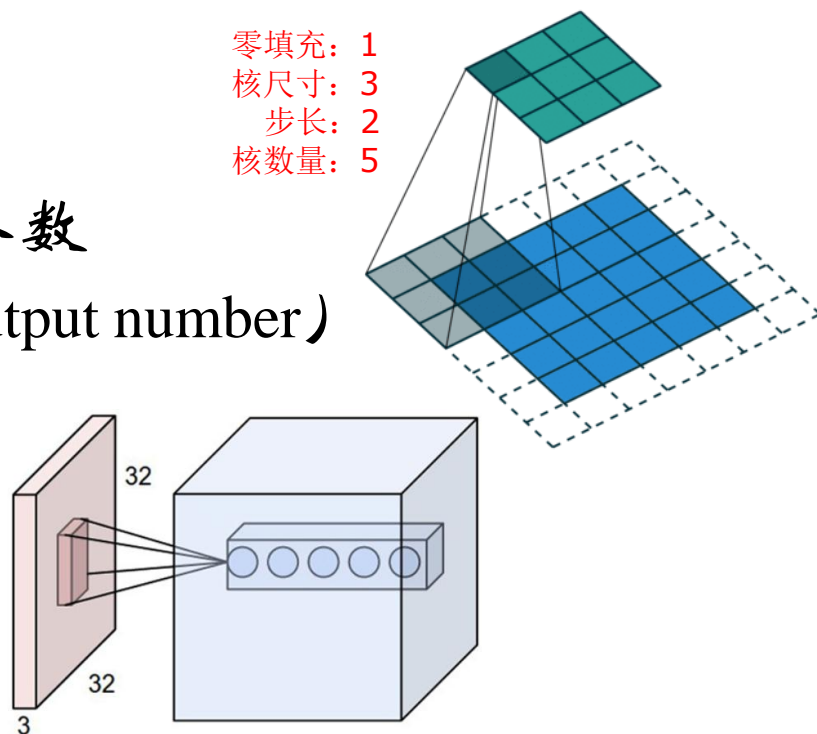
- 3D滤波器/卷积核
  - 以**扫描窗**的方式，对图像做卷积
  - 每层含有**多个核**，每个核对应一个输出通道
  - 提取**局部**特征
  - **权重参数需要自学习**



# 卷积神经网络（CNN）

## CNN卷积层

- 3D滤波器/卷积核的超参数
  - 滤波器/卷积核数量（output number）
  - 核尺寸（kernel size）
  - 步长（stride）
  - 零填充（zero padding）
- 尺寸计算（W, H, D）
  - $W = (W - \text{size} + 2 * \text{padding}) / \text{stride} + 1$
  - $H = (H - \text{size} + 2 * \text{padding}) / \text{stride} + 1$
  - $D = \text{output number}$

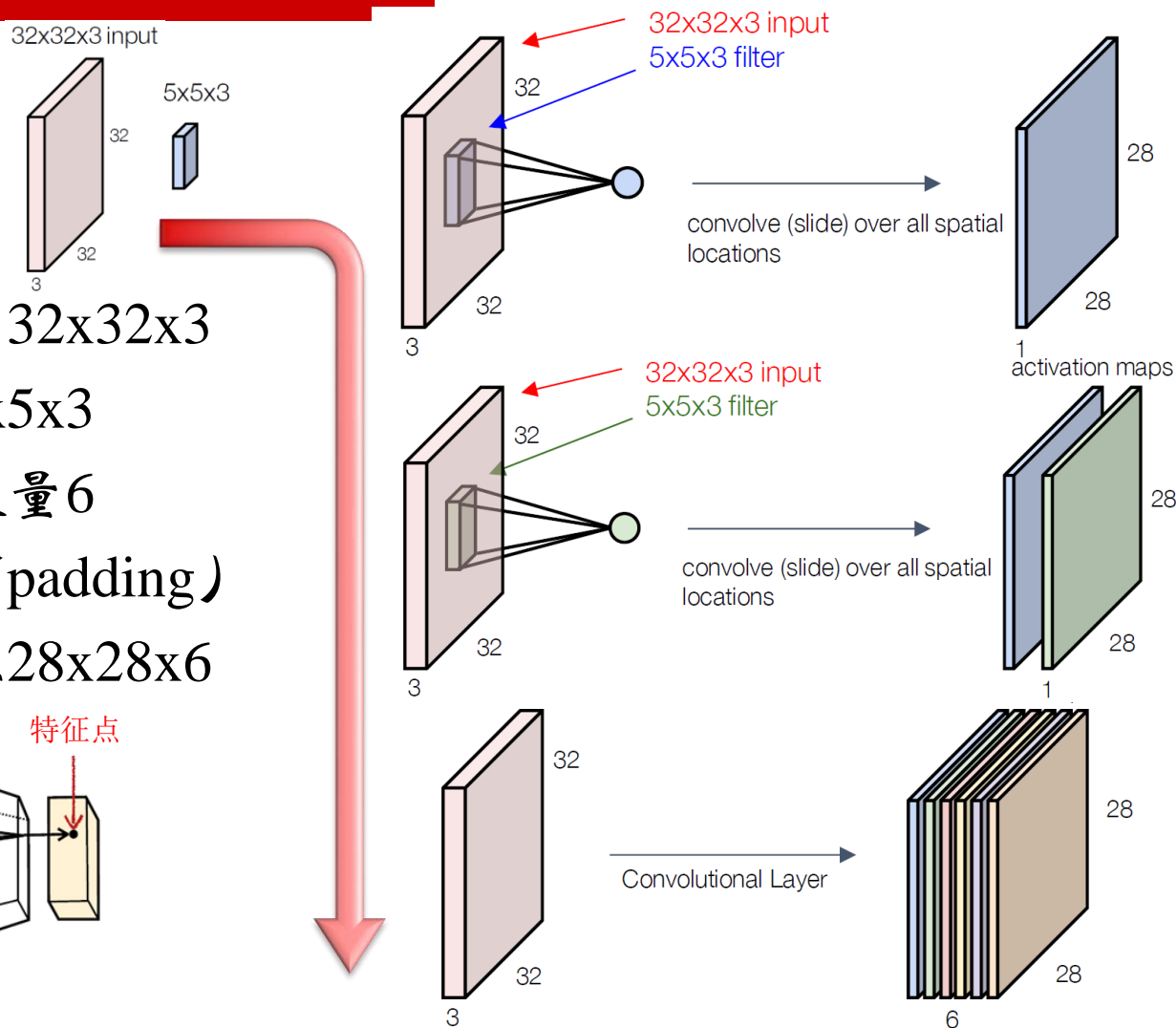
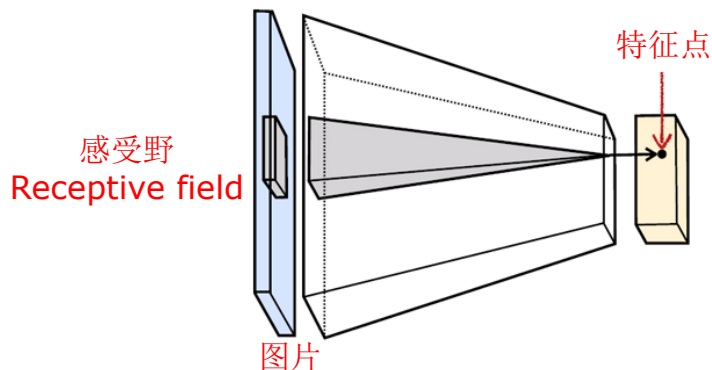


# 卷积神经网络 (CNN)

## CNN 卷积层

- 原理演示

- 输入图片  $32 \times 32 \times 3$
- 卷积核  $5 \times 5 \times 3$
- 卷积核数量 6
- 无补零 (padding)
- 输出特征  $28 \times 28 \times 6$

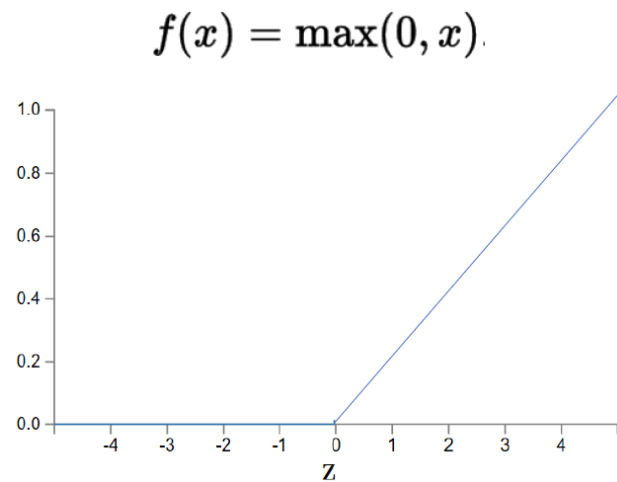
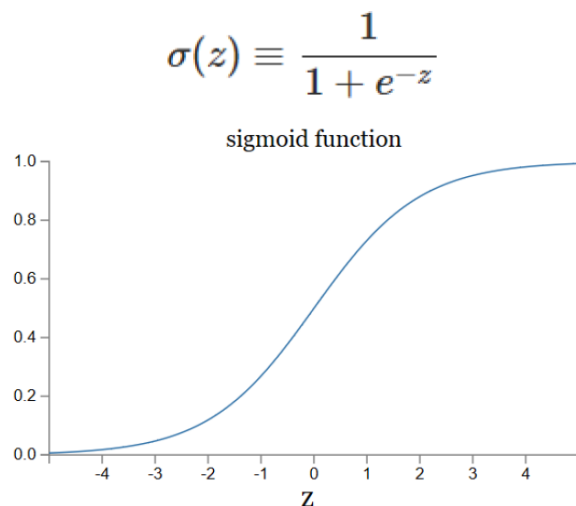




# 卷积神经网络（CNN）

## CNN卷积层

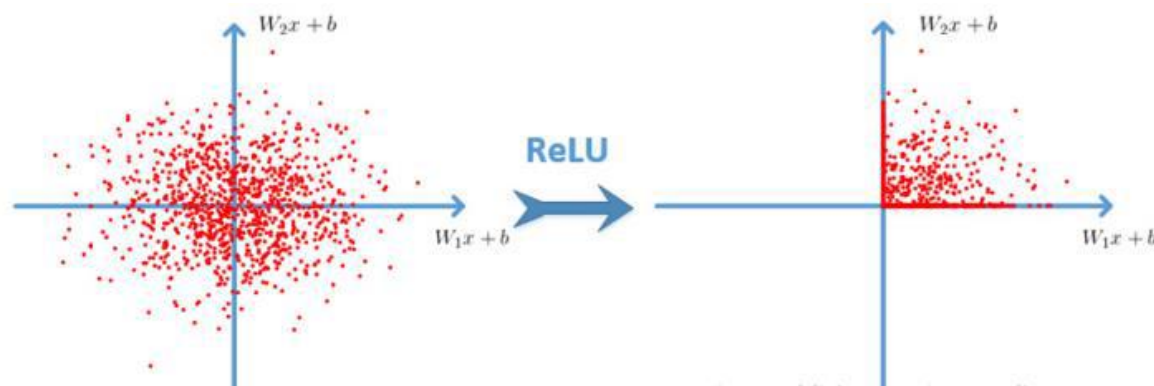
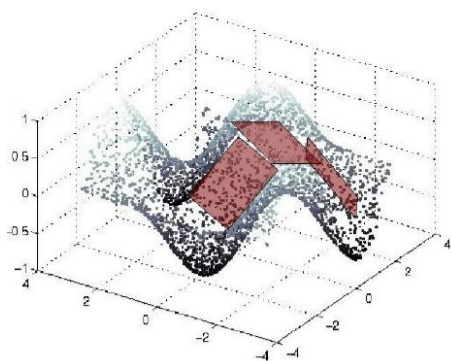
- 非线性激活函数
  - Sigmoid
  - **ReLU** (Rectified Linear Unit)



# 卷积神经网络（CNN）

## CNN卷积层

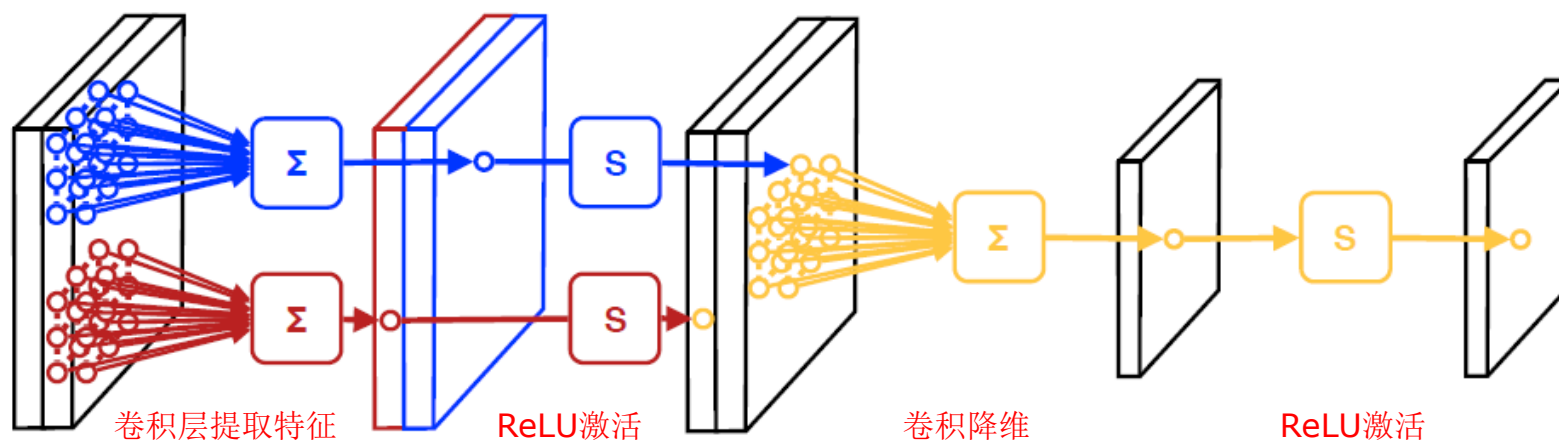
- **ReLU**激活函数
  - 分段线性函数
  - 无饱和问题，明显减轻梯度消失问题
  - 深度网络能够进行优化的关键



# 卷积神经网络（CNN）

## CNN卷积层

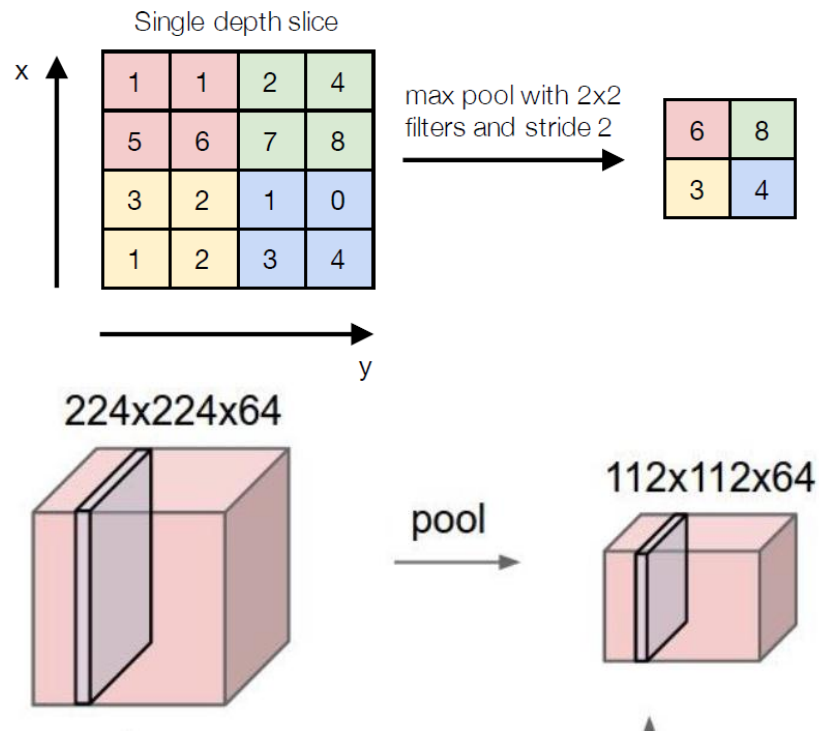
- 组合简例
  - 卷积步长大于1，有降维作用



# 卷积神经网络（CNN）

## CNN池化层

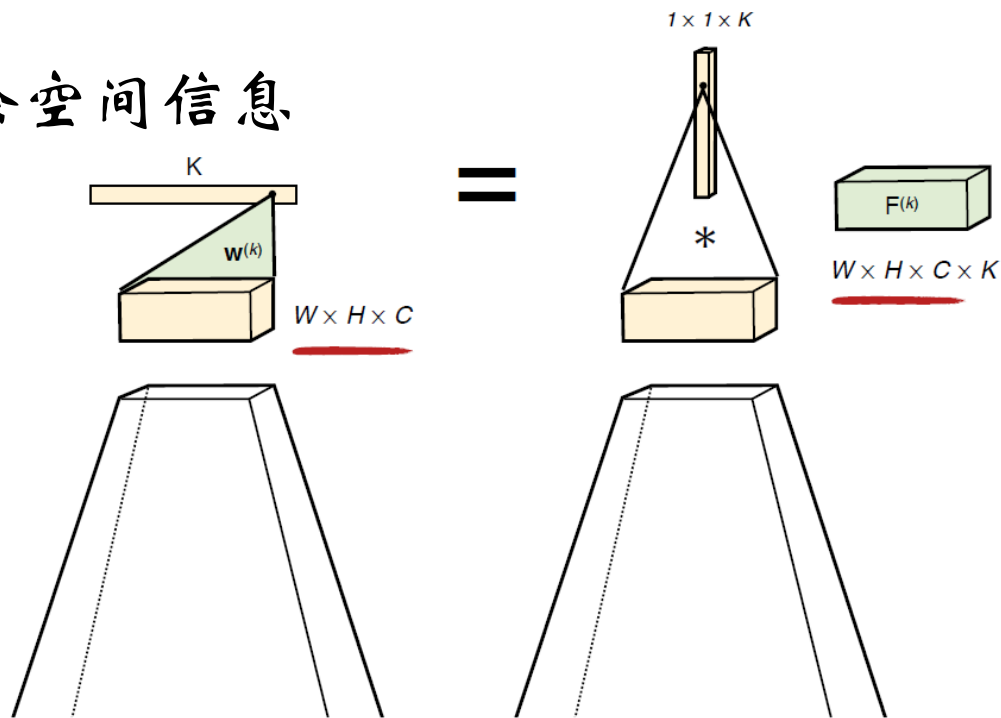
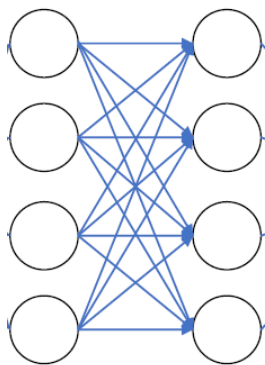
- 作用：特征融合，降维
- 无参数需要学习
- 超参数
  - 尺寸（size）
  - 步长（step）
  - 计算类别
    - 最大化池化（Max pooling）
    - 平均池化（Average pooling）



# 卷积神经网络 (CNN)

## CNN全连接层

- 作用：推理器，分类器
- 普通神经网络
- 全局感受野，去除空间信息
- 需要学习参数
- 等效于全局卷积



# 卷积神经网络（CNN）

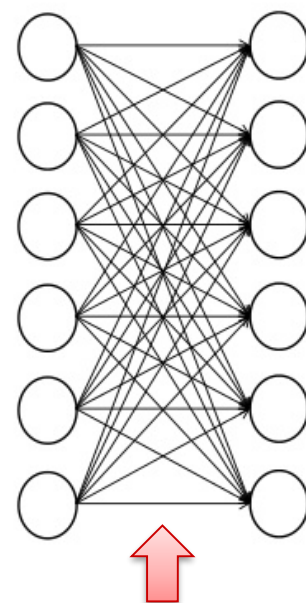
## CNN-Softmax层

- 指数归一化函数
  - 将一个实数值向量压缩到  $(0, 1)$
  - 所有元素和为1

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K.$$

- 最后一个全连接层对接1000-way的softmax层
- 得出1000类标签的概率值
- 取log值后，用于构建loss

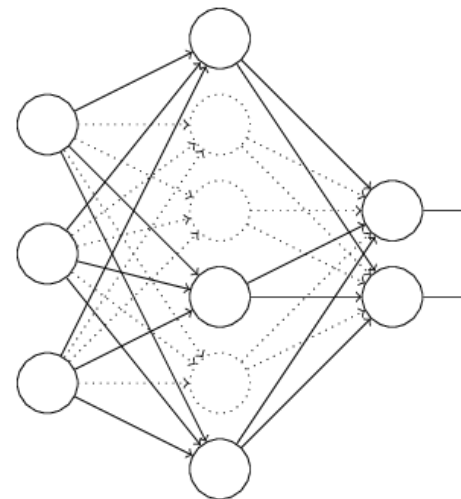
全链接输出 概率值



# 卷积神经网络（CNN）

## 工程技巧tricks

- 图像像素中心化
  - (R, G, B) 减去各自通道的均值
- 防过拟合，提高泛化能力
  - 数据增强x10
    - 256x256中提取中心和四角的224x224子图片x5
    - 水平翻转x2
  - Dropout随机失活
    - 训练中，随机让一些神经元的输出设为0
    - 失活率0.5
  - Weight decay权重衰减（L2正则）
$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2$$



# 卷积神经网络（CNN）

---

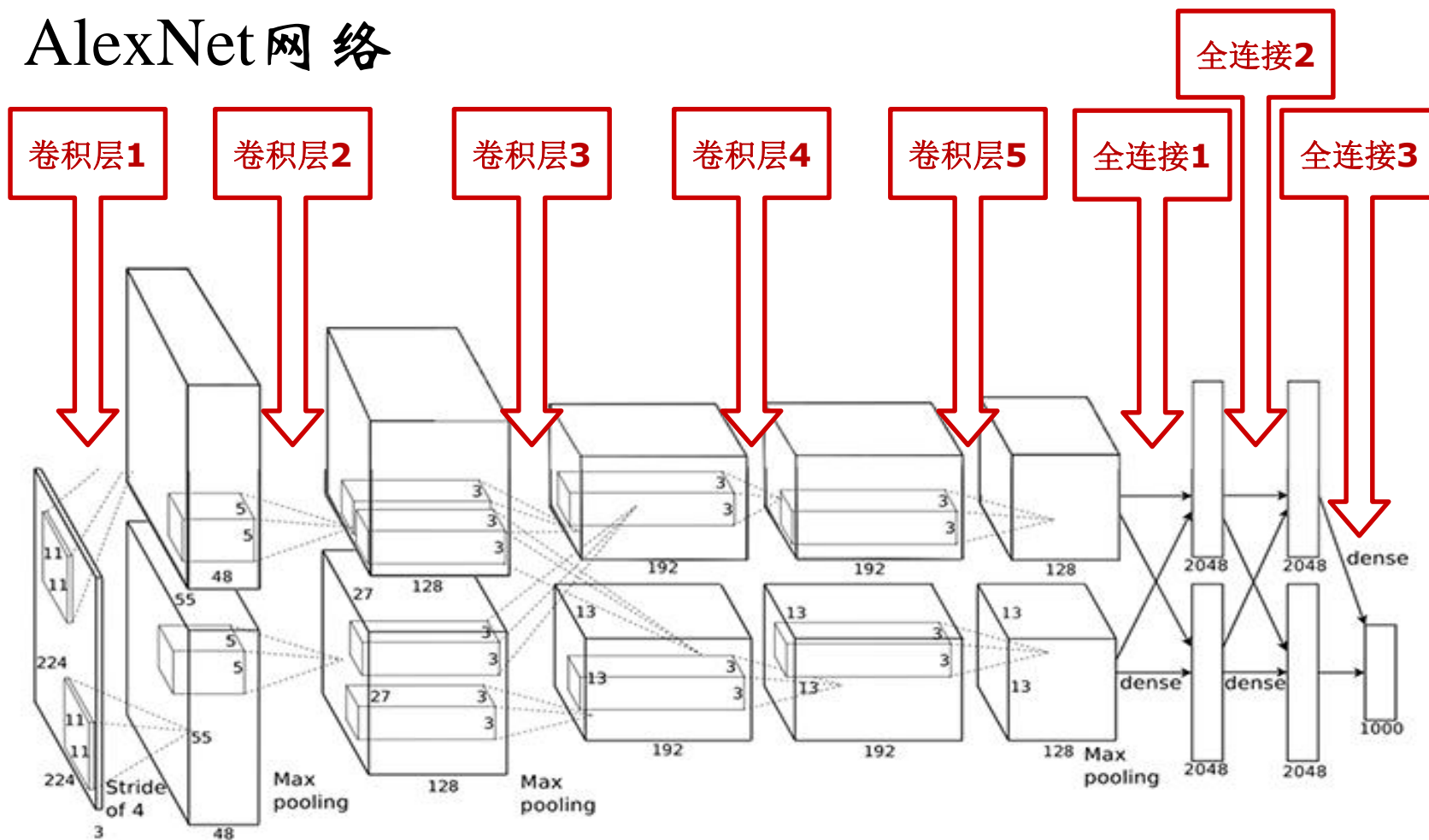
## AlexNet网络

- 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）
    - 训练时长一周，50x加速
  - 新技术
    - ReLU非线性激活
    - Max pooling池化
    - Dropout regularization



# 卷积神经网络（CNN）

## AlexNet网络

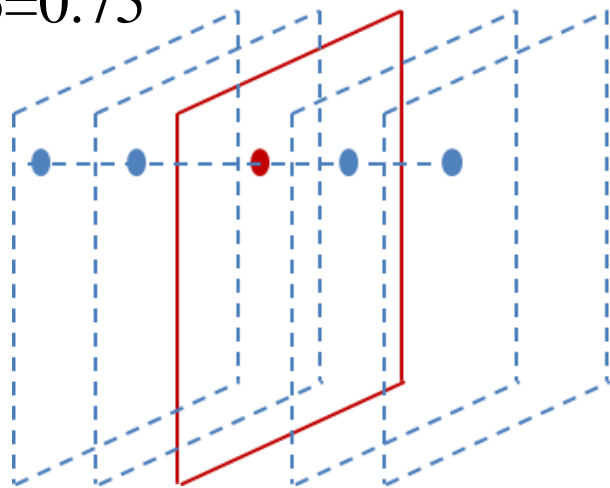


# 卷积神经网络（CNN）

## AlexNet网络

- 局部响应归一化（Local Response Normalization）
  - 神经元的侧抑制机制
  - 某个位置（x,y）上跨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,  $\alpha=0.0001$ ,  $\beta=0.75$

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



# 卷积神经网络（CNN）

## AlexNet网络

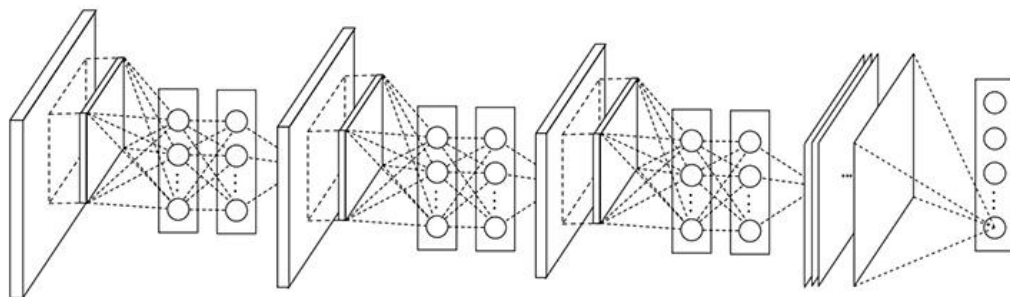
- 输入层:  $224 \times 224 \times 3$
- 卷积层1:  $96 \times 11 \times 11 \times 3 \rightarrow \text{LRN} \rightarrow \text{MP} \rightarrow \text{ReLU}$
- 卷积层2:  $256 \times 5 \times 5 \times 48 \rightarrow \text{LRN} \rightarrow \text{MP} \rightarrow \text{ReLU}$ （通道独立）
- 卷积层3:  $384 \times 3 \times 3 \times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



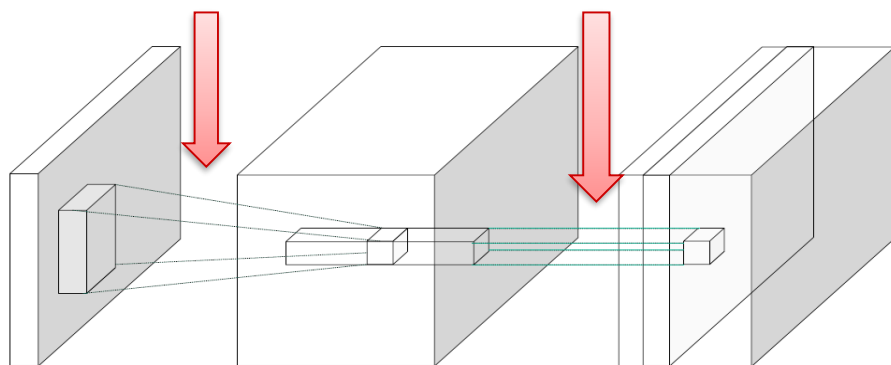
# 卷积神经网络（CNN）

## Network-in-Network网络（NiN）

- 提高CNN的局部感知区域（Bottleneck Layer）



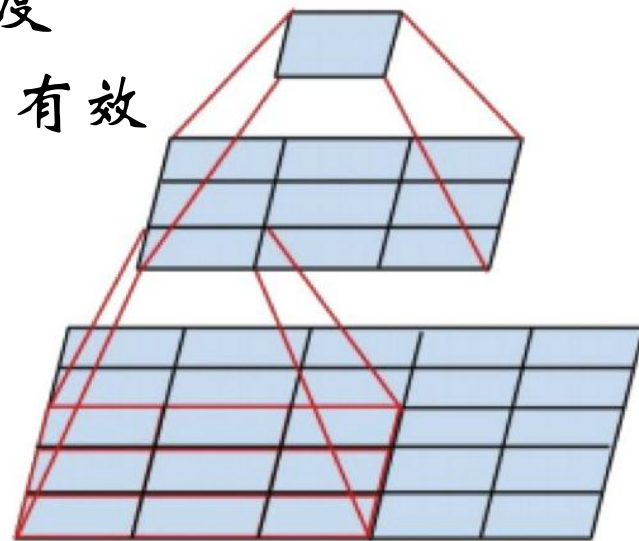
- 卷积层  $\rightarrow$   $1 \times 1$  卷积层  $\rightarrow$  Max池化层



# 卷积神经网络（CNN）

## VGG网络

- 一个**大**卷积核**分解**成连续多个**小**卷积核
  - 核分解：  $7 \times 7$ 核  $\rightarrow$  3个  $3 \times 3$ 核（由ReLU连接）
  - 参数数量：  $49C^2 \rightarrow 27C^2$
- 减少参数，降低计算，增加深度
- 继承AlexNet结构特点：简单，有效
- ImageNet-2014竞赛第二
- 网络改造的首选基础网络

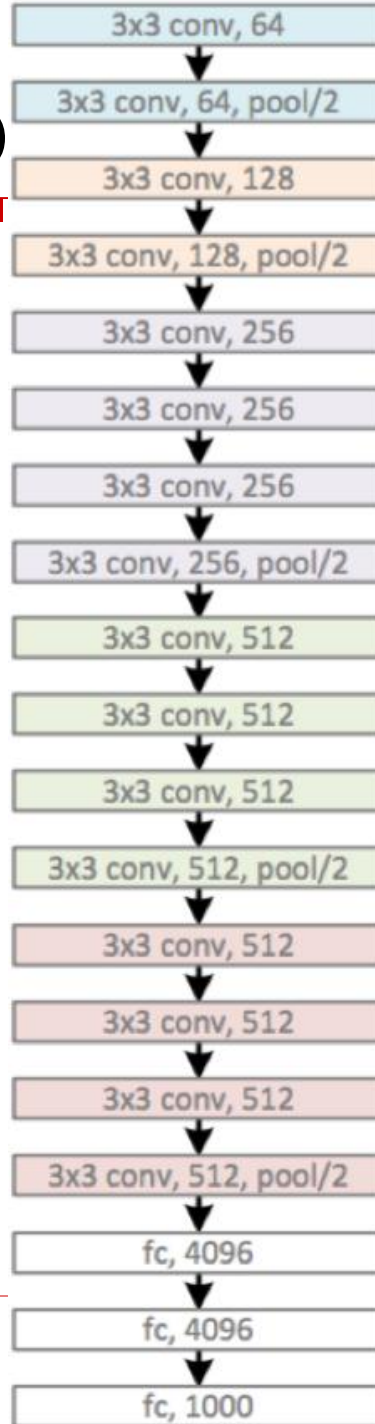


# 卷积神经网络 (CNN)

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 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	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 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
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



# 第4课 图像分类（下）

---

## Image Classification

主讲人：张宗健

悉尼科技大学博士

主要研究方向： 计算机视觉、视觉场景理解、图像&语言、深度学习  
图像检索CbIR、Human ReID等

# 卷积神经网络（CNN）

---

## GoogLeNet网络

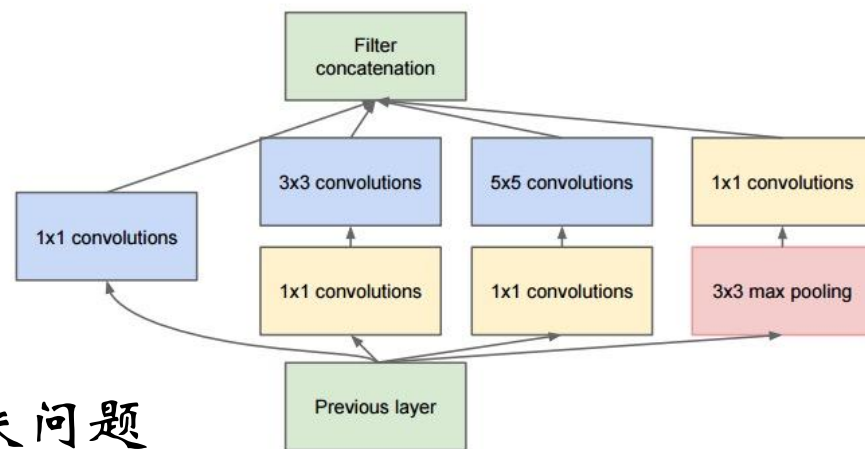
- ImageNet-2014竞赛第一
- 进化顺序
  - Inception V1 → Inception V2 → Inception V3 → Inception V4
- 为了提升性能
  - 减少参数，降低计算
  - 增加宽度、深度



# 卷积神经网络（CNN）

## Inception V1 网络

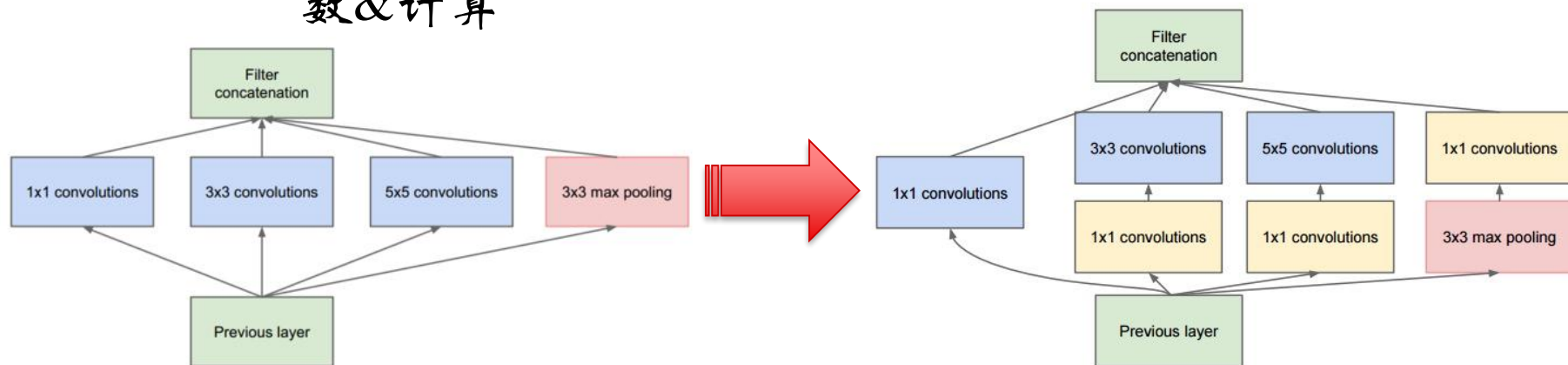
- 核心组件 **Inception Architecture**
  - Split-Merge  $\rightarrow$  1x1 卷积, 3x3 卷积, 5x5 卷积, 3x3 池化
    - 增加网络对 **多尺度** 的适应性
    - 增加网络 **宽度**
  - Bottleneck Layer  $\rightarrow$  使用 **NiN** 的 1x1 卷积进行特征降维
    - 大幅降低 **计算量** 10x
- 取消全连接
  - 参数量大, 减负
- 辅助分类器
  - 解决前几层的梯度消失问题



# 卷积神经网络（CNN）

## Inception V1 网络

- 核心组件 **Inception Architecture**（稀疏连接结构）
  - 提供 **多尺度** 特征：输出通道多尺度（感受野）化
    - 首个 Split-Merge 思想
    - 串接合并所有分支输出
  - **Bottleneck Layer** 的 1x1 卷积解决多尺度带来的高额参数&计算



# 卷积神经网络（CNN）

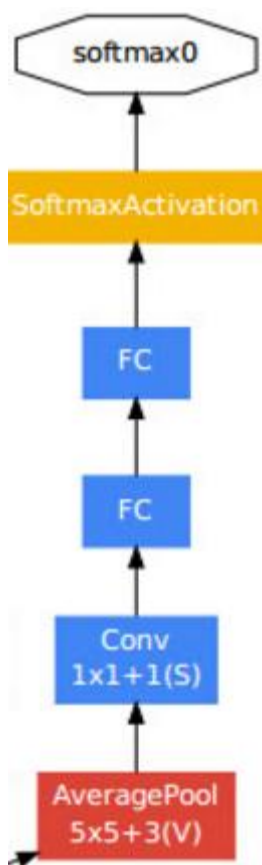
---

## 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 ( $6 \times 6 \times 256 \times 4096 + 4096 \times 4096 + 4096 \times 1000$ )
    - VGG: 72M ( $7 \times 7 \times 256 \times 4096 + 4096 \times 4096 + 4096 \times 1000$ )
  - 由全局平均池化替代（Global average pooling）
    - 输入：7x7x1024
    - 输出：1x1x1024
- 一大趋势

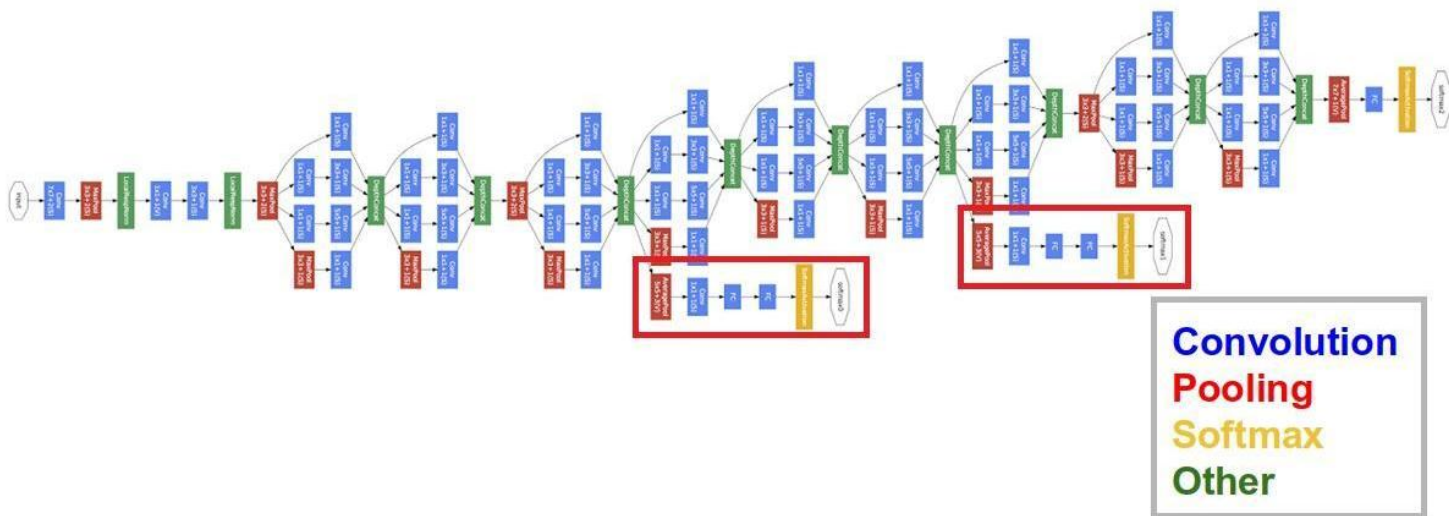
# 卷积神经网络 (CNN)

## Inception V1 网络



2个辅助分类器

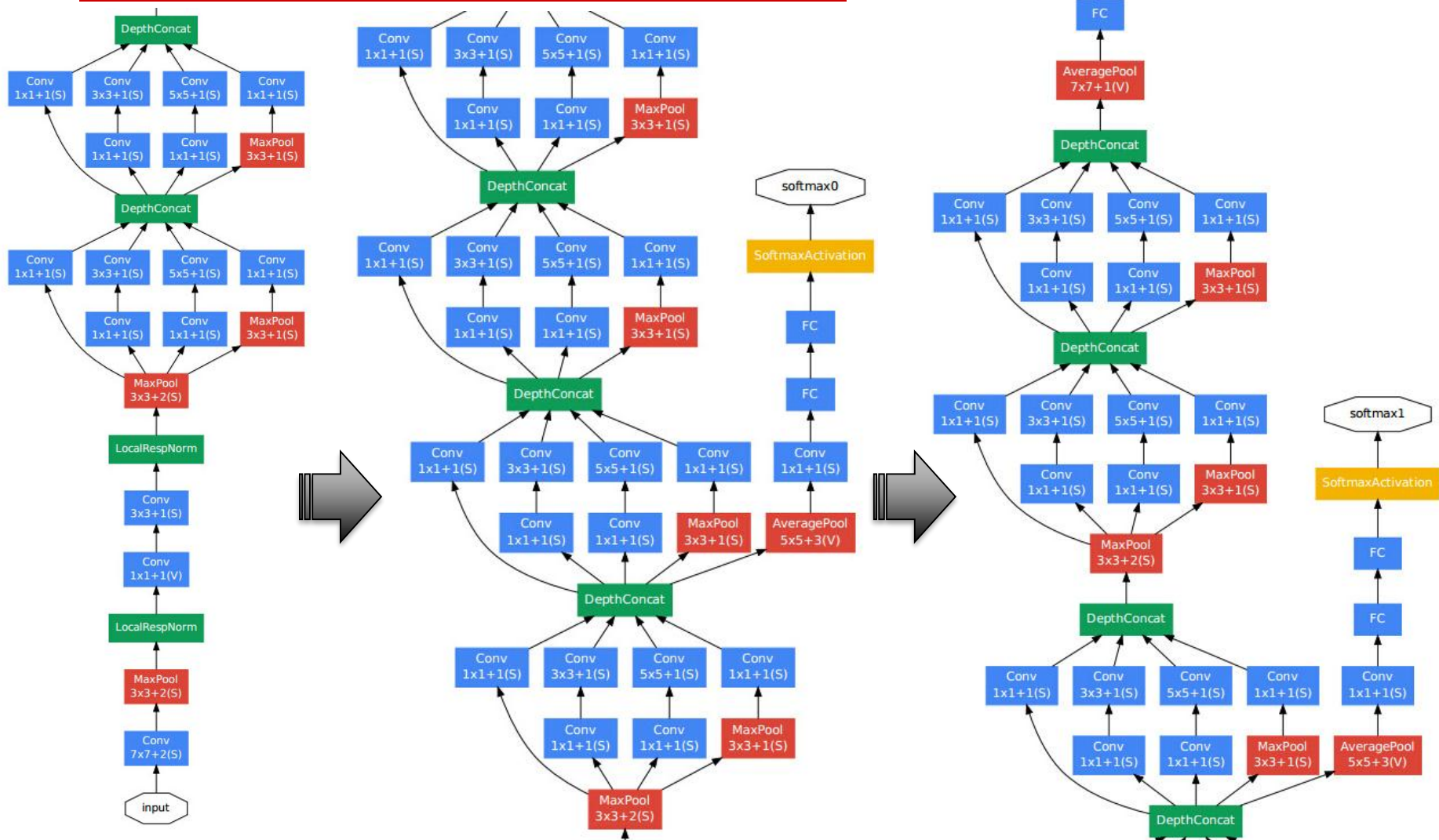
- 深网络中，梯度回传到最初几层，存在严重消失问题
- 有效加速收敛
- 测试阶段不使用



# 卷积神经网络 (CNN)

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								

# 卷积神经网络 (CNN)

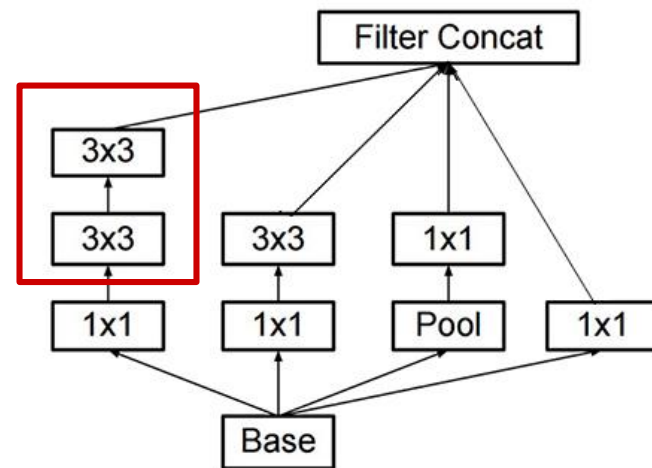




# 卷积神经网络（CNN）

## Inception V2网络

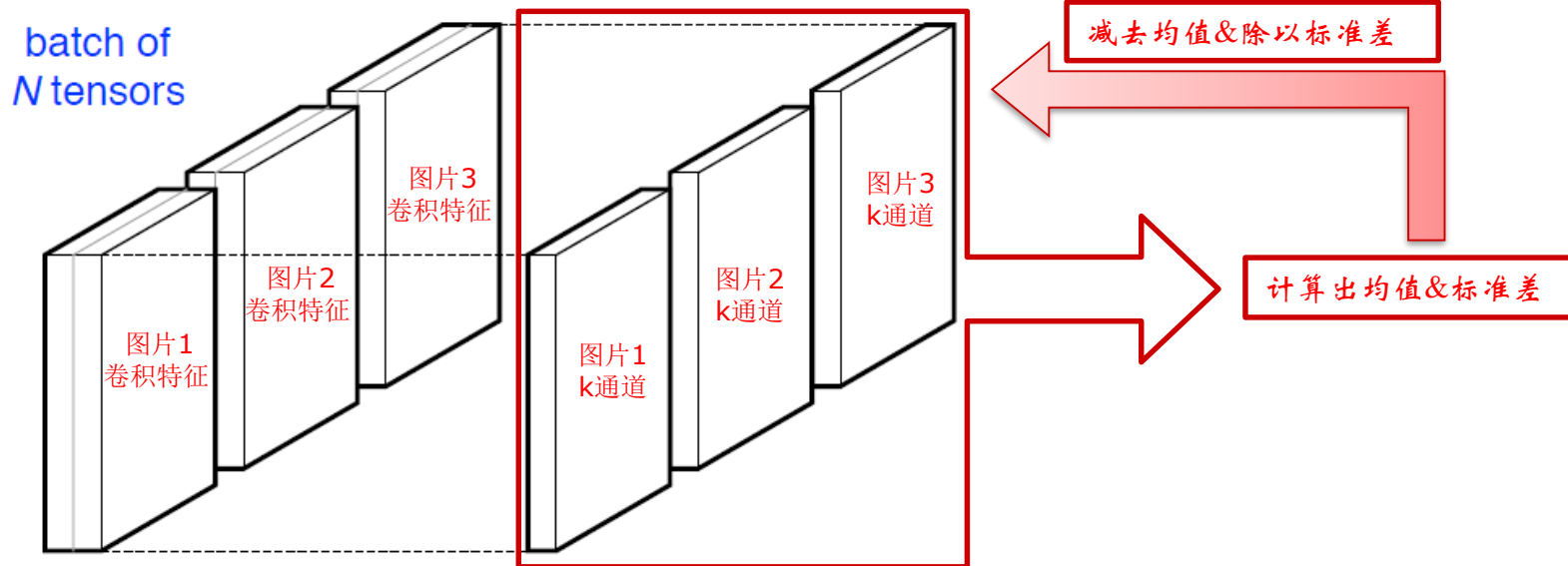
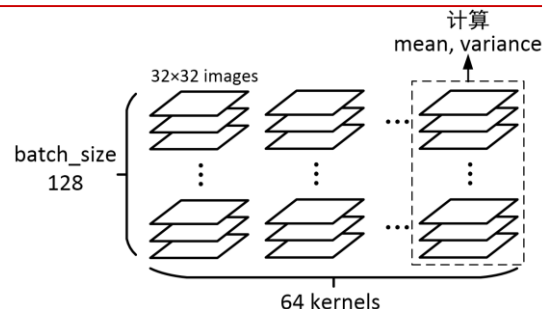
- 核心组件
  - Batch Normalization（批归一化）
    - 解决Internal Covariate Shift问题（内部neuron的数据分布发生变化）
    - 白化：使每一层的输出都规范化到 $N(0, 1)$
    - 允许较高学习率
    - 取代部分Dropout
  - 5x5卷积核→2个3x3卷积核



# 卷积神经网络（CNN）

## 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

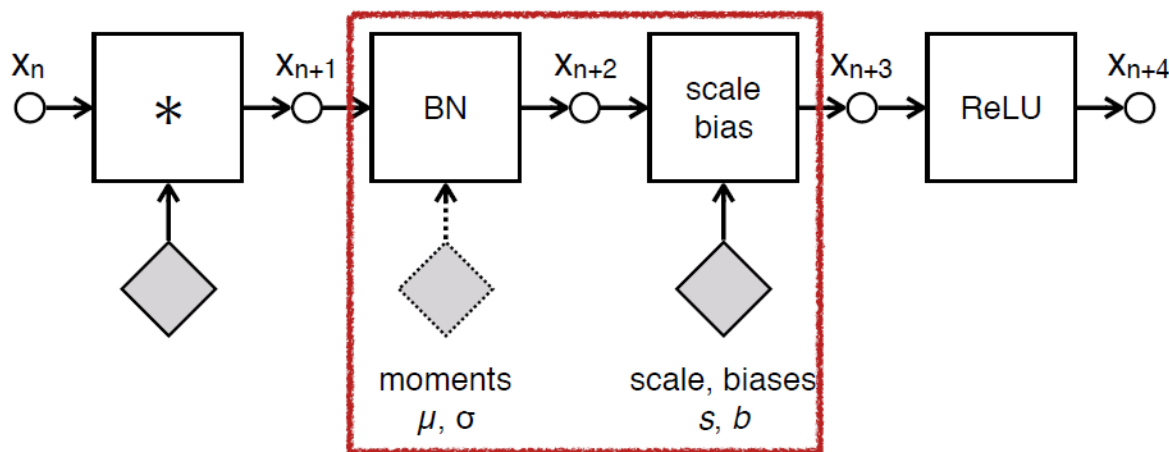




# 卷积神经网络（CNN）

## Inception V2网络

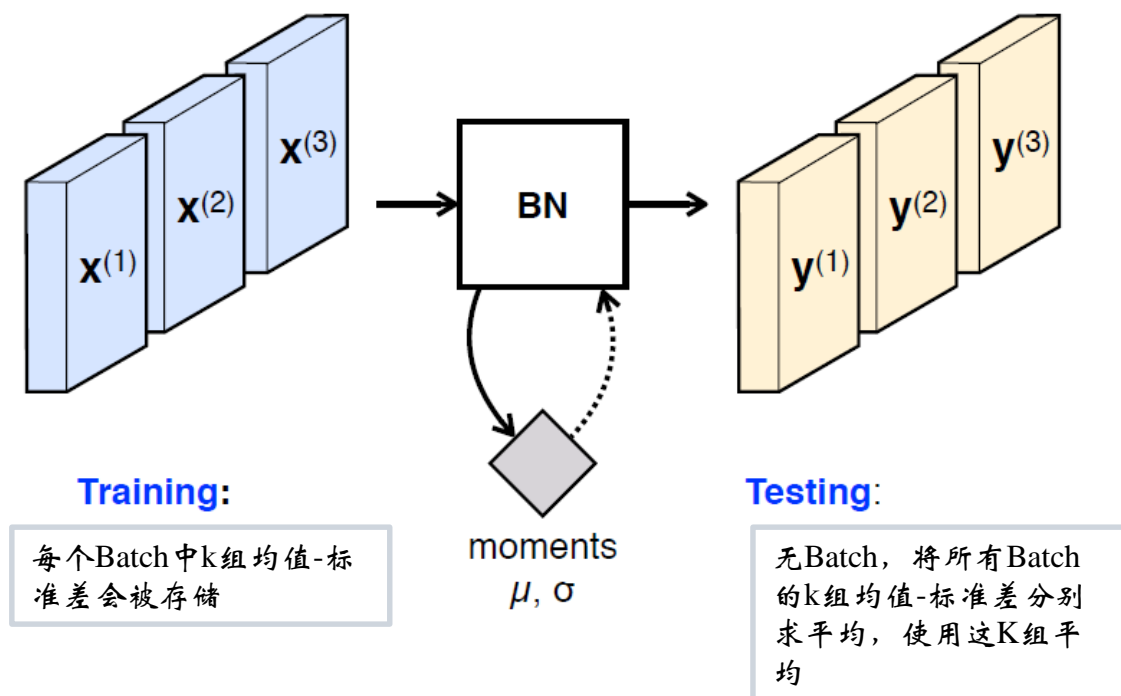
- 位置：卷积→BN→ReLU
- 配对使用scale & shift
  - 添加一组逆算子：scale乘子, bias偏置
  - 这组参数需要学习  $y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)}$



# 卷积神经网络（CNN）

## 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）

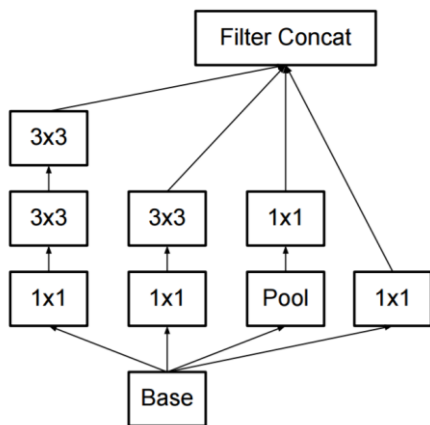
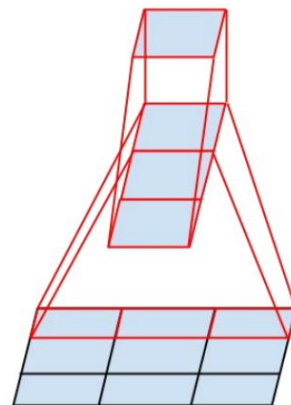


# 卷积神经网络（CNN）

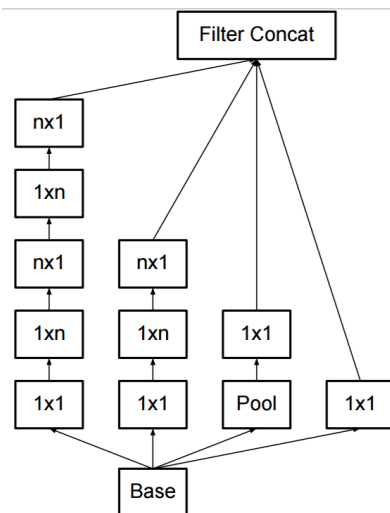
## Inception V3网络

- 核心组件

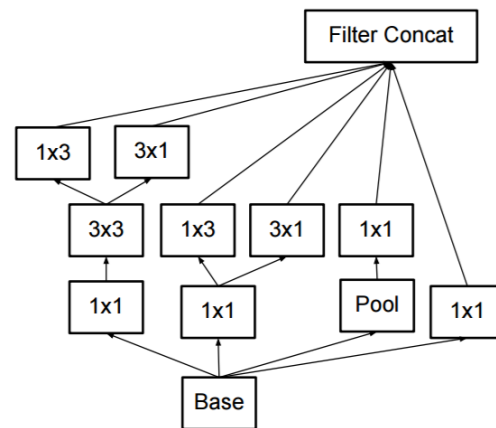
- 非对称卷积： $N \times N$  分解成  $1 \times N \rightarrow N \times 1$
- 降低参数数量和计算量



- 分辨率35x35上使用



- 分辨率17x17上使用
- $n=7$



- 最低分辨率8x8上使用
- 增加特征维度

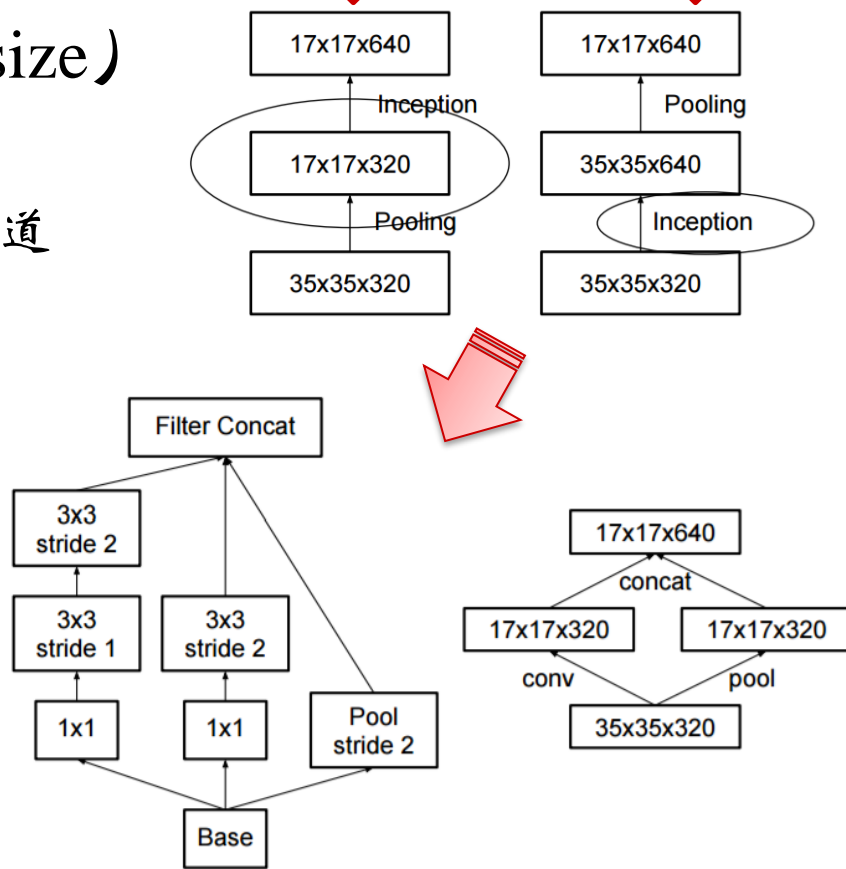
# 卷积神经网络 (CNN)

## 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

- 有表达瓶颈
- 计算量小

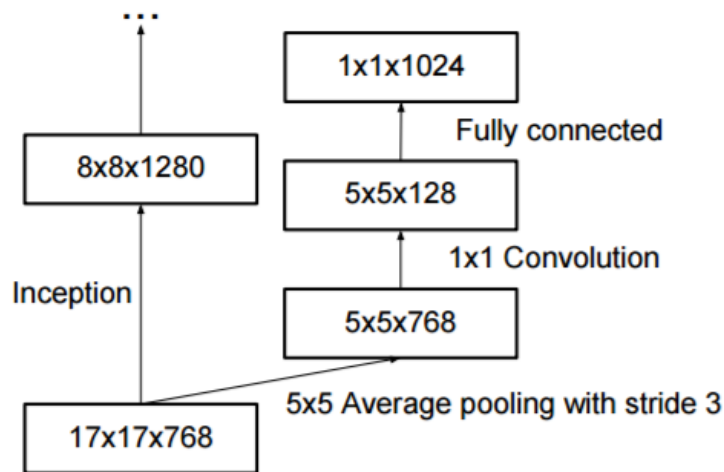
- 无表达瓶颈
- 计算量很大



# 卷积神经网络（CNN）

## 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

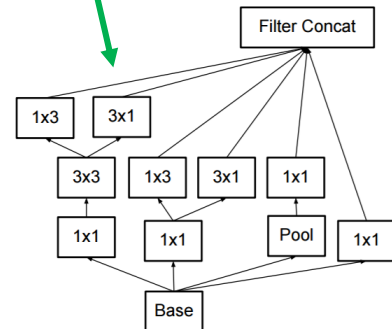
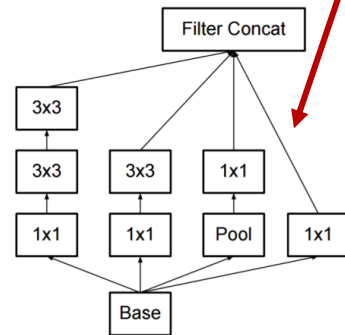
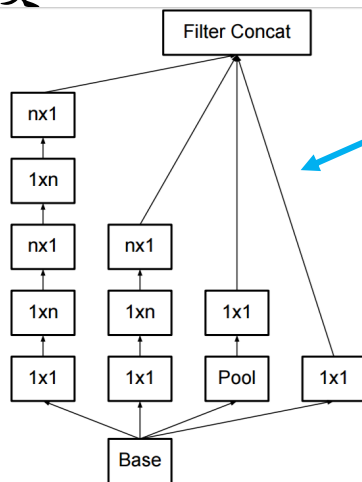


# 卷积神经网络（CNN）

## 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

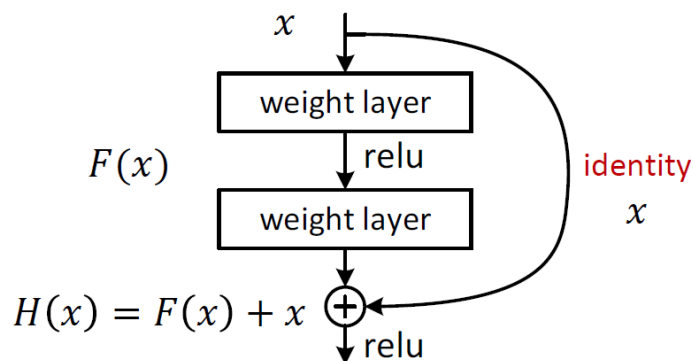
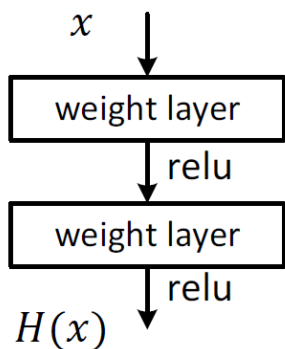
type	patch size/stride or remarks	input size
conv	$3 \times 3/2$	$299 \times 299 \times 3$
conv	$3 \times 3/1$	$149 \times 149 \times 32$
conv padded	$3 \times 3/1$	$147 \times 147 \times 32$
pool	$3 \times 3/2$	$147 \times 147 \times 64$
conv	$3 \times 3/1$	$73 \times 73 \times 64$
conv	$3 \times 3/2$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



# 卷积神经网络（CNN）

## ResNet残差网络

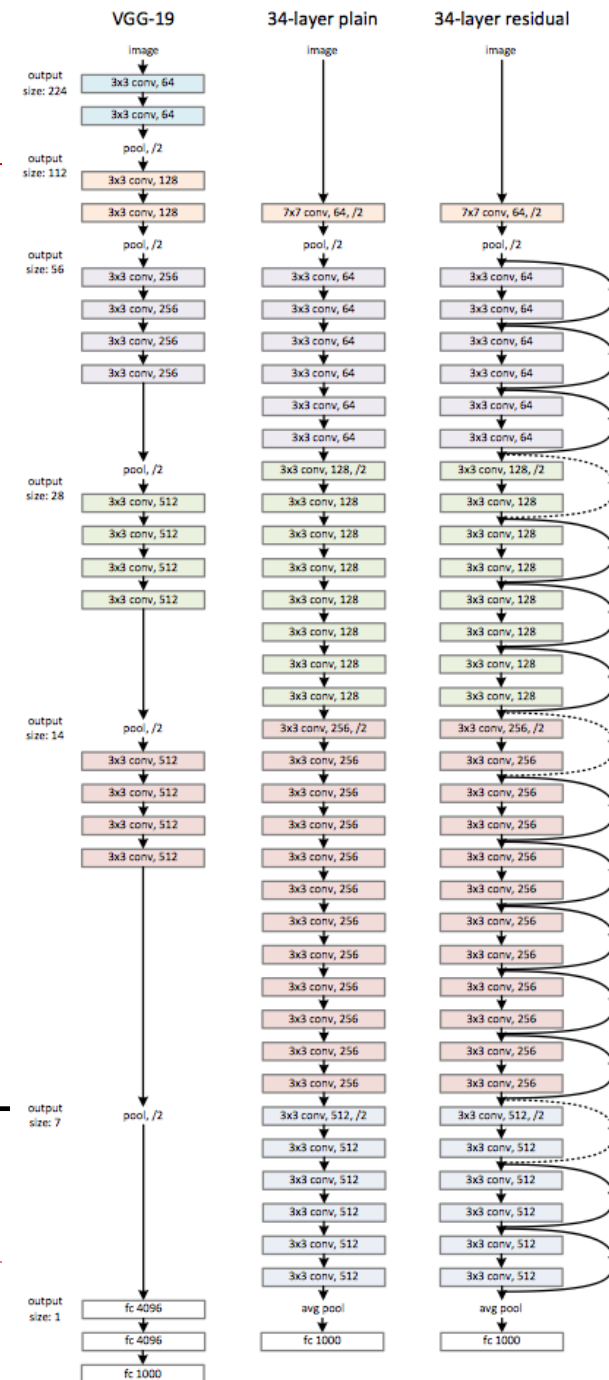
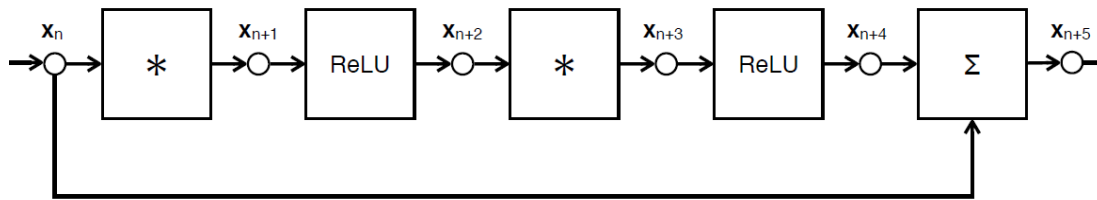
- 核心组件Skip/shortcut connection
  - Plain net: 可以拟合出任意目标映射 $H(x)$
  - Residual net
    - 可以拟合出任意目标映射 $F(x)$ ,  $H(x)=F(x)+x$
    - $F(x)$ 是残差映射, 相对于identity来说
    - 当 $H(x)$ 最优映射接近identity时, 很容易捕捉到小的扰动



# 卷积神经网络 (CNN)

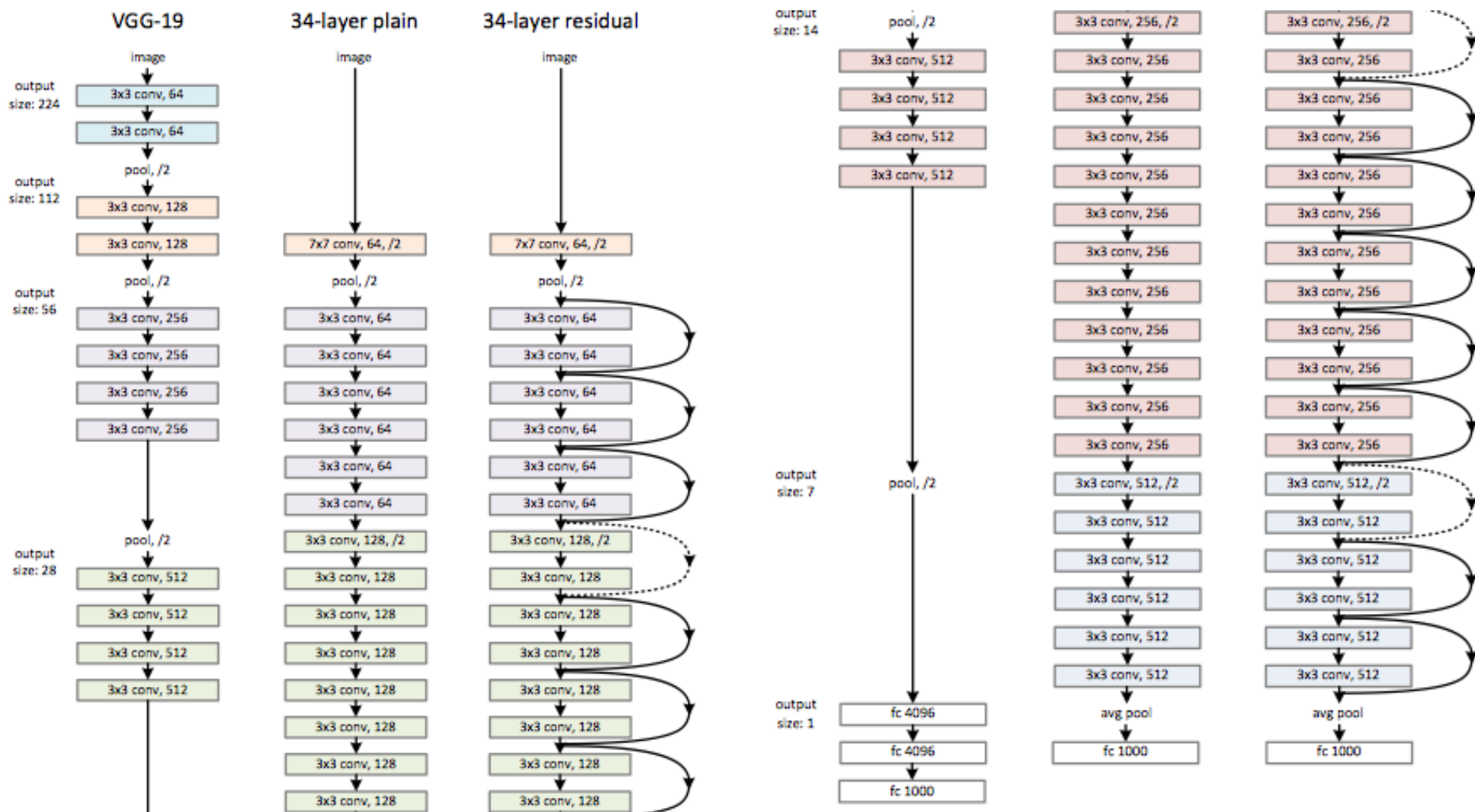
## ResNet残差网络

- 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





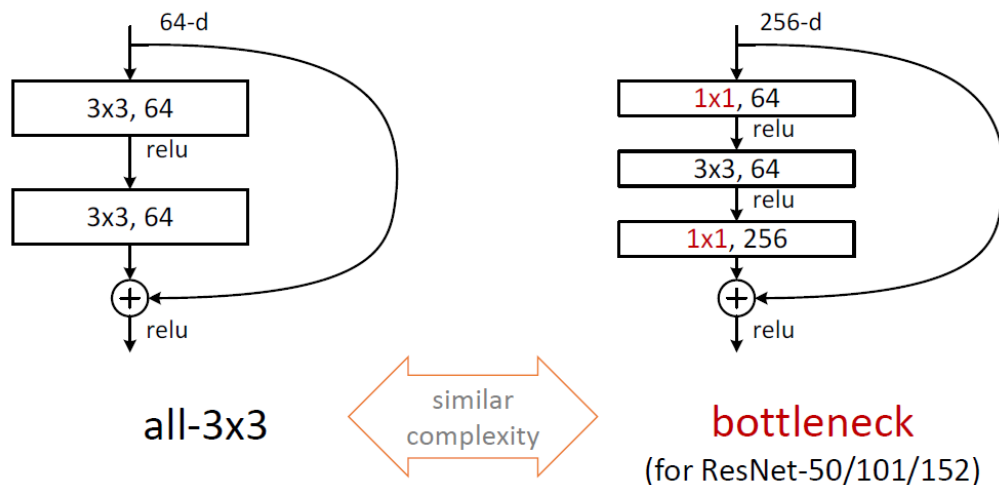
# 卷积神经网络 (CNN)



# 卷积神经网络（CNN）

## ResNet残差网络

- 更深网络：根据**Bootleneck**优化残差映射网络
  - 原始： $3 \times 3 \times 256 \times 256 \rightarrow 3 \times 3 \times 256 \times 256$
  - 优化： $1 \times 1 \times 256 \times 64 \rightarrow 3 \times 3 \times 64 \times 64 \rightarrow 1 \times 1 \times 64 \times 256$



# 卷积神经网络（CNN）

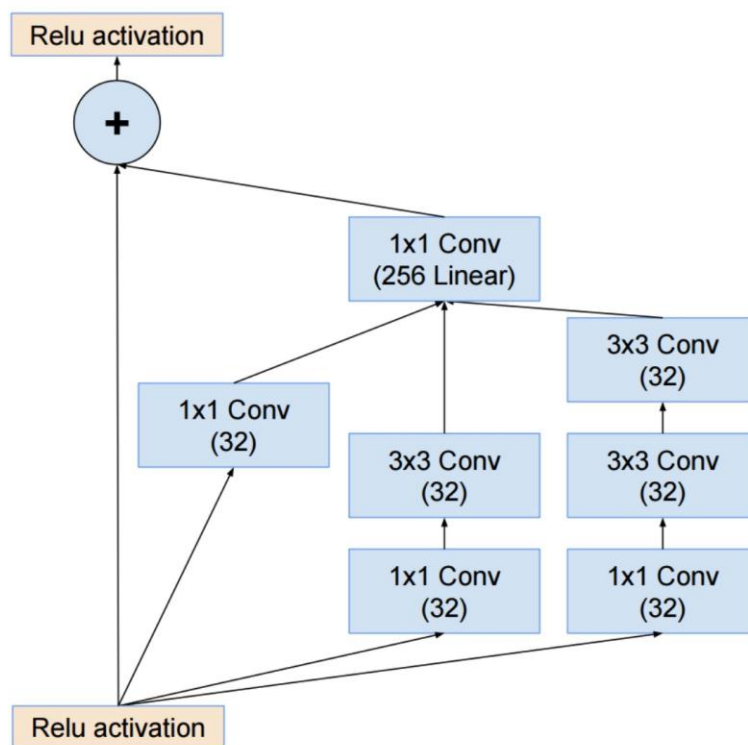
## ResNet残差网络

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				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

# 卷积神经网络（CNN）

## Inception V4网络

- 引入残差到Inception Architecture



# 卷积神经网络（CNN）

---

## ResNeXt网络

- 提出第3个DNN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

# 卷积神经网络（CNN）

## ResNeXt网络

- 32x4d块结构

- ResNet卷积

- $256\text{in} \rightarrow 1 \times 1, 64 \rightarrow 3 \times 3, 64 \rightarrow 1 \times 1, 256 \rightarrow 256\text{out}$

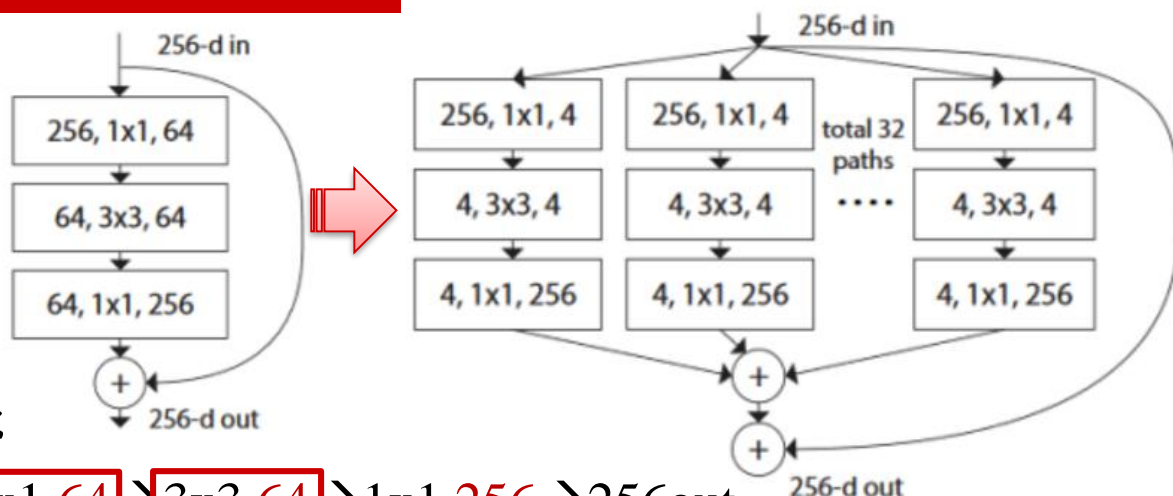
- ResNeXt卷积

- 分支1:  $256\text{in} \rightarrow 1 \times 1, 4 \rightarrow 3 \times 3, 4 \rightarrow 1 \times 1, 256 \rightarrow 256\text{out}$

- 分支2:  $256\text{in} \rightarrow 1 \times 1, 4 \rightarrow 3 \times 3, 4 \rightarrow 1 \times 1, 256 \rightarrow 256\text{out}$

- .....

- 分支32:  $256\text{in} \rightarrow 1 \times 1, 4 \rightarrow 3 \times 3, 4 \rightarrow 1 \times 1, 256 \rightarrow 256\text{out}$



# 卷积神经网络（CNN）

## ResNeXt网络

- 在计算复杂度固定的情况下
  - ResNet-50: ~4.1B FLOPs
  - ResNet-50: ~7.8B FLOPs
- 增加cardinality基数会不断提高性能

	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	<b>22.2</b>
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	<b>21.2</b>

# 卷积神经网络（CNN）

## ResNeXt网络

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

	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

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
		$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128, C=32 \\ 1\times 1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256, C=32 \\ 1\times 1, 512 \end{bmatrix} \times 4$
		$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512, C=32 \\ 1\times 1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 1024 \\ 3\times 3, 1024, C=32 \\ 1\times 1, 2048 \end{bmatrix} \times 3$
		global average pool	global average pool
	1×1	1000-d fc, softmax	1000-d fc, softmax
# params.		25.5×10 <sup>6</sup>	25.0×10 <sup>6</sup>
FLOPs		4.1×10 <sup>9</sup>	4.2×10 <sup>9</sup>



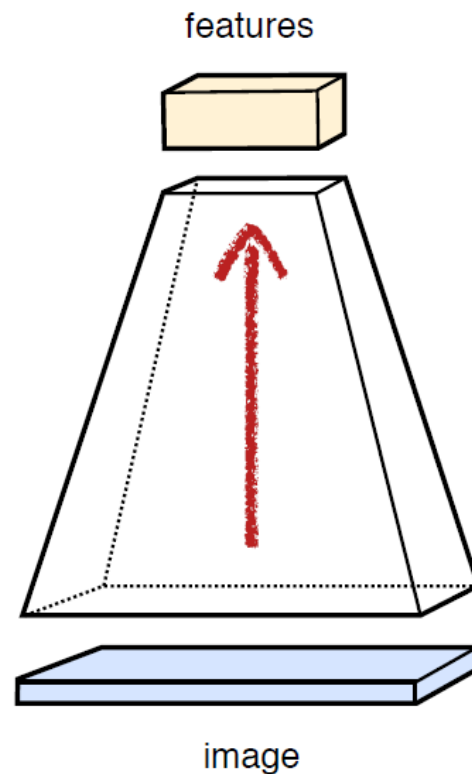
# 卷积神经网络（CNN）

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	—	—	—	+

# 卷积神经网络（CNN）

## CNN设计准则

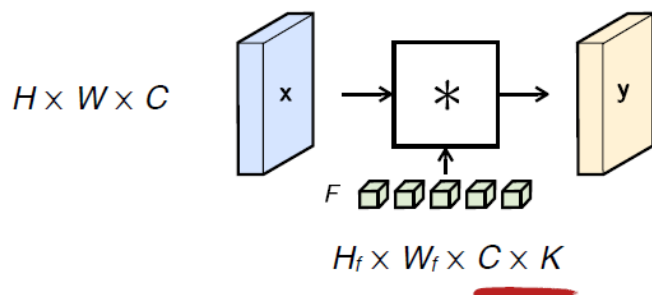
- 避免信息瓶颈
  - 卷积过程中
    - 空间尺寸 $H \times W$ 会变小
    - 输出通道数 $C$ 会变多
  - 数据量 $H \times W \times C$ 要缓慢变小



# 卷积神经网络（CNN）

## 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  $C$
  - 输入通道数量  $H$
  - 参数数量  $H_f \times W_f \times \underline{C \times K}$
  - 操作数量  $\frac{H \times H_f}{\text{stride}} \times \frac{W \times W_f}{\text{stride}} \times \underline{C \times K}$

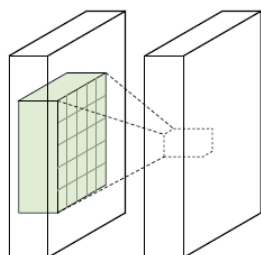


$$\text{complexity} \propto \underline{C \times K}$$

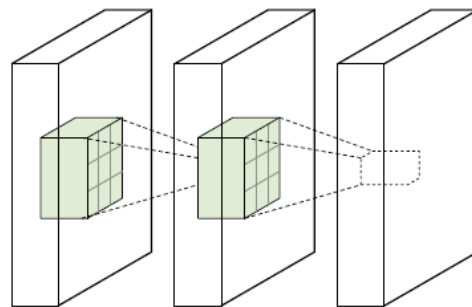
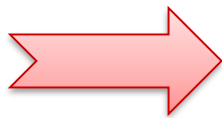
# 卷积神经网络 (CNN)

## CNN设计准则

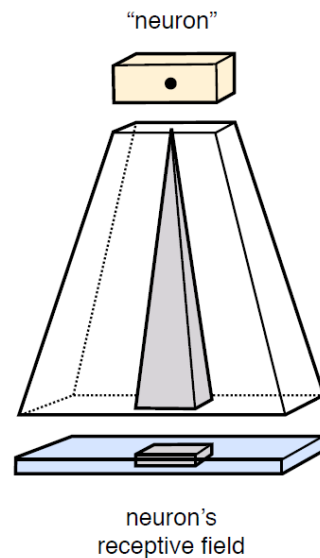
- 感受野要足够大
  - 卷积是基于局部图片的操作
  - 捕捉大尺寸内容
  - 多个小尺寸卷积核 vs 一个大尺寸卷积核
    - 参数少，计算快
    - 多个非线性激活



5 × 5 filters  
+ ReLU



3 × 3 filters  
+ ReLU      3 × 3 filters  
+ ReLU



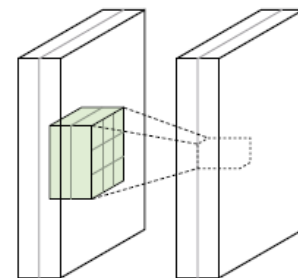
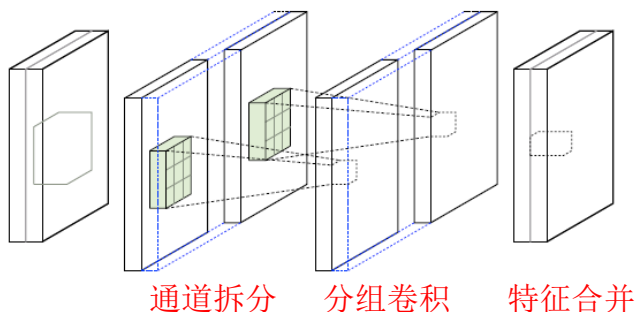
# 卷积神经网络 (CNN)

## 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**



$$\begin{matrix} \text{y} \end{matrix} = \begin{bmatrix} \text{green} & 0 & 0 \\ 0 & \text{green} & 0 \\ 0 & 0 & \text{green} \end{bmatrix} \times \begin{matrix} \text{x} \end{matrix}$$

$F$

complexity:  $C \times K / G$

$$\begin{matrix} \text{y} \end{matrix} = \begin{bmatrix} C \times K \end{bmatrix} \times \begin{matrix} \text{x} \end{matrix}$$

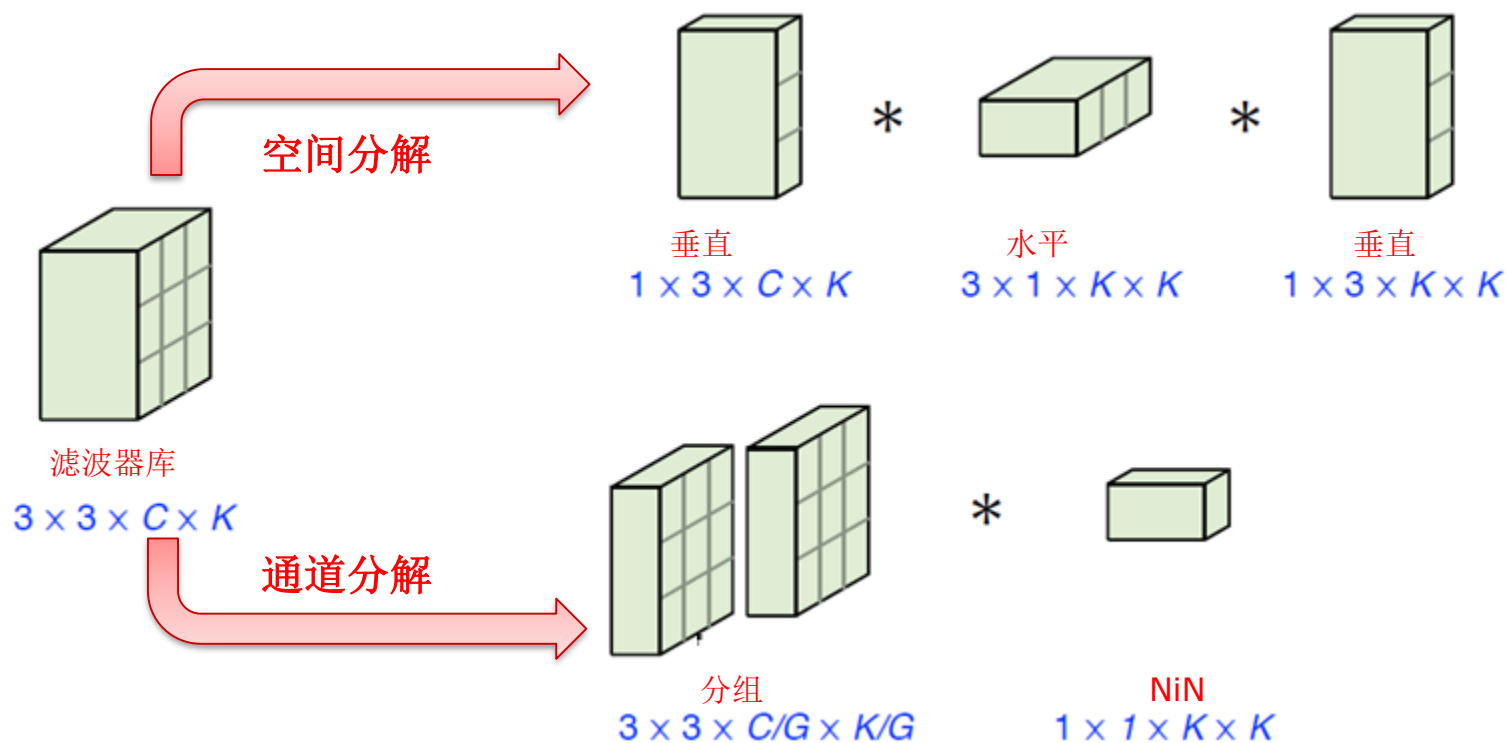
$F$

complexity:  $C \times K$

# 卷积神经网络 (CNN)

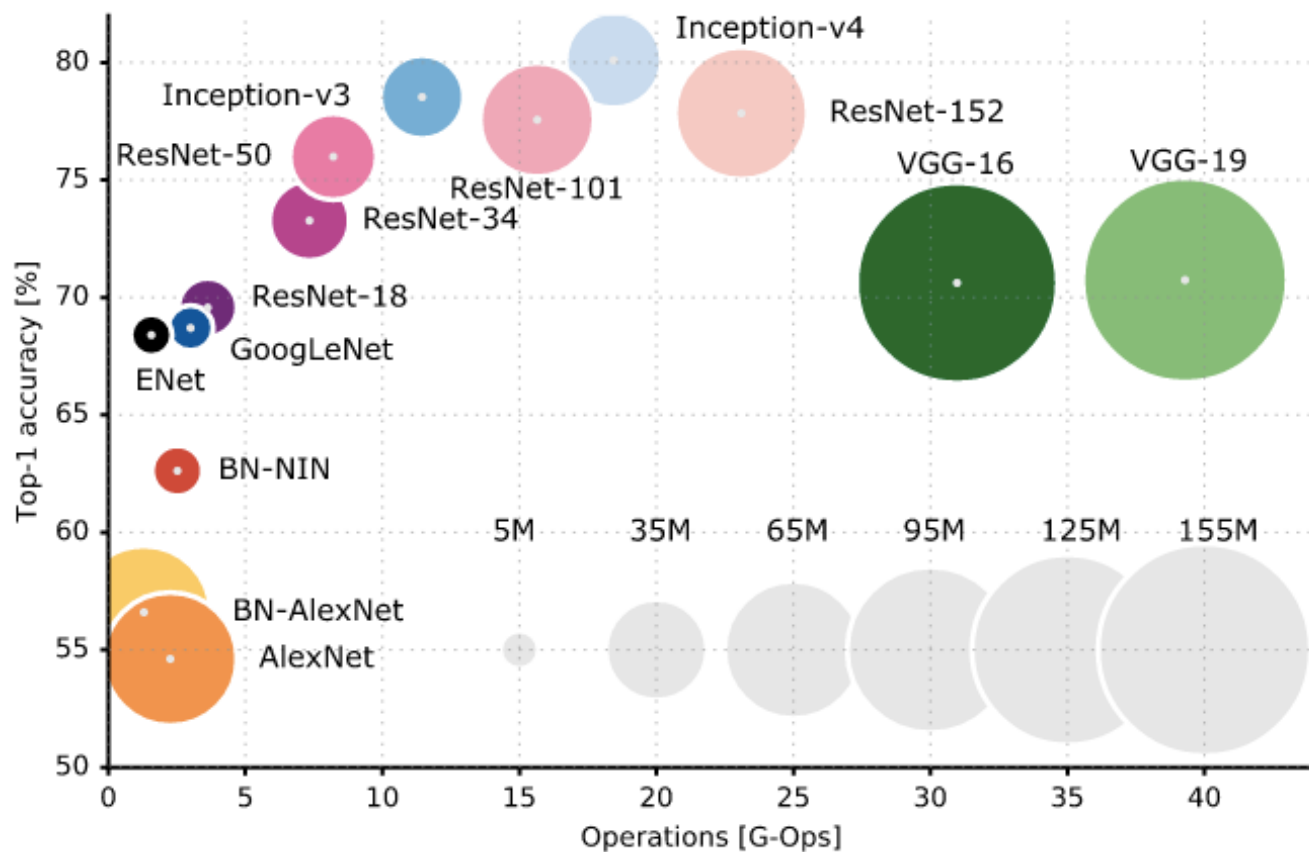
## CNN设计准则

- 低秩分解  $\rightarrow$  降低参数&计算量



# ImageNet分类

## 准确性 vs. 执行效率



# 场景分类

## 性能对比

	Validation Set of Places365		Test Set of Places365	
	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.
Places365-AlexNet	53.17%	82.89%	53.31%	82.75%
Places365-GoogLeNet	53.63%	83.88%	53.59%	84.01%
Places365-VGG	<b>55.24%</b>	84.91%	<b>55.19%</b>	85.01%
Places365-ResNet	54.74%	<b>85.08%</b>	54.65%	<b>85.07%</b>



GT: cafeteria  
top-1: cafeteria (0.179)  
top-2: restaurant (0.167)  
top-3: dining hall (0.091)  
top-4: coffee shop (0.086)  
top-5: restaurant patio (0.080)



GT: creek  
top-1: forest broadleaf (0.307)  
top-2: forest path (0.208)  
top-3: creek (0.086)  
top-4: rainforest (0.076)  
top-5: cemetery (0.049)



GT: crosswalk  
top-1: crosswalk (0.720)  
top-2: plaza (0.060)  
top-3: street (0.055)  
top-4: shopping mall indoor (0.039)  
top-5: bazaar outdoor (0.021)



GT: greenhouse indoor  
top-1: greenhouse indoor (0.479)  
top-2: greenhouse outdoor (0.055)  
top-3: botanical garden (0.044)  
top-4: assembly line (0.025)  
top-5: vegetable garden (0.022)



# 本章小结

---

- CNN
  - 基础层
  - AlexNet
  - VGG
  - Inception V1  $\rightarrow$  V4
  - ResNet  $\rightarrow$  ResNeXt
- 代码实例
  - ResNet

# 演示环节

---

- Github
  - <https://github.com/349zzjau>
- 百度网盘
  - <http://pan.baidu.com/s/1gfpCCwj>
- 演示内容
  - Windows 7 下 TensorFlow 的配置
  - ResNet网络
    - 结构图<http://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006>

# 疑问

---

□ 问题答疑：<http://www.xxwenda.com/>

■ 可邀请老师或者其他人回答问题

## Q & A

小象账号：349zzjau

课程名：基于深度学习的计算机视觉

课后调查问卷<http://cn.mikecrm.com/ANB98i9>

# 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
- 新浪微博：ChinaHadoop

