

# 深度学习：从理论到实践

## 第一章：深度学习理论（二） ——卷积神经网络（上）



# 课程大纲

---

✓ 基本概念

✓ 发展历程

✓ 网络特点

# 课程大纲

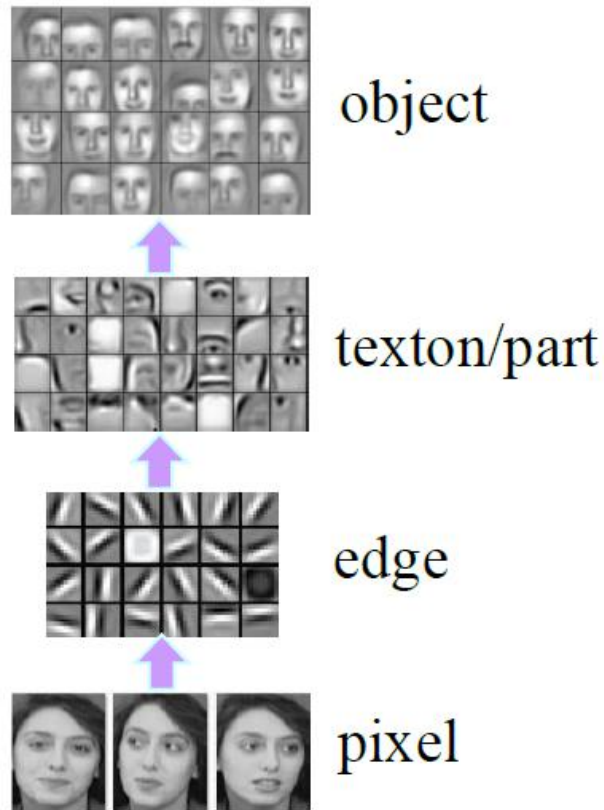
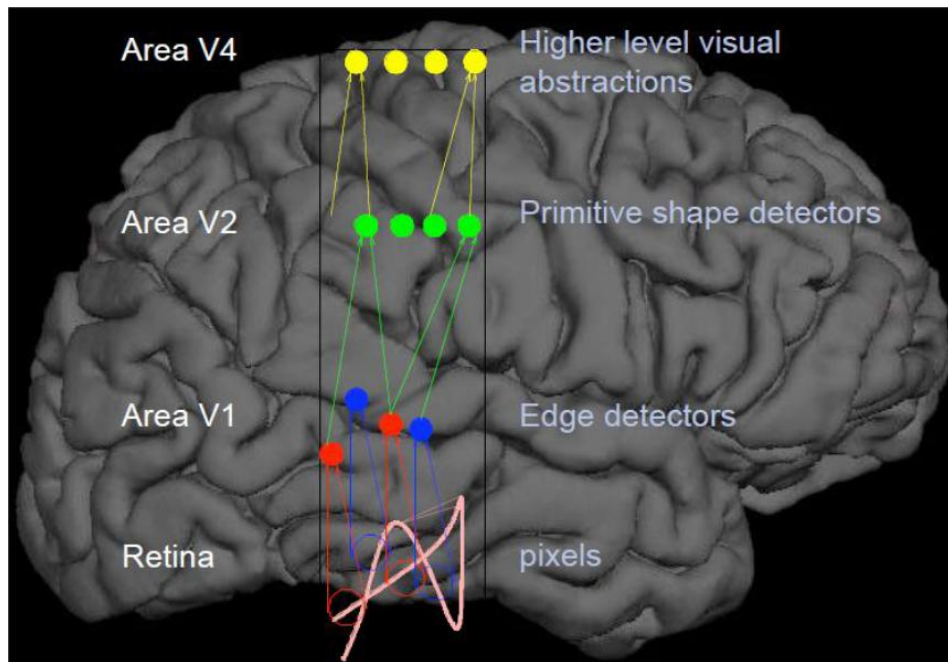
---

✓ 基本概念

✓ 发展历程

✓ 网络特点

## 人类视觉系统的层次结构



## 感受野与局部感受野

局部感受野+人工神经网络→局部连接

### ➤ 生物神经网络中的感受野：

- 1962年，Hubel和Wiesel通过对猫视觉皮层细胞的研究，提出了**感受野**（receptive field）的概念。

### ➤ 视觉系统**局部感受野**：

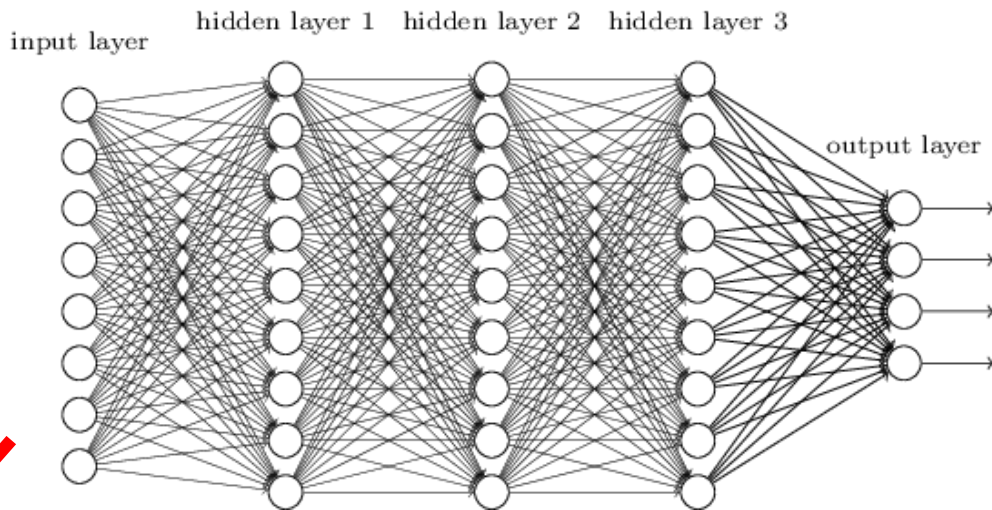
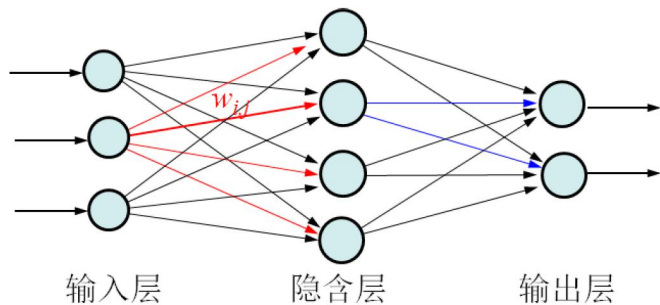
- 人对外界的认知是从局部到全局的。视觉皮层的神经元就是局部接受信息的（即只响应某些特定区域的刺激）。

### ➤ 从生物神经网络到人工神经网络：

- 1984年，日本学者Fukushima基于感受野概念提出的神经认知机，是感受野概念在人工神经网络领域的首次应用。

## 从神经网络到卷积神经网络：why CNN?

(convolutional neural network, CNN)



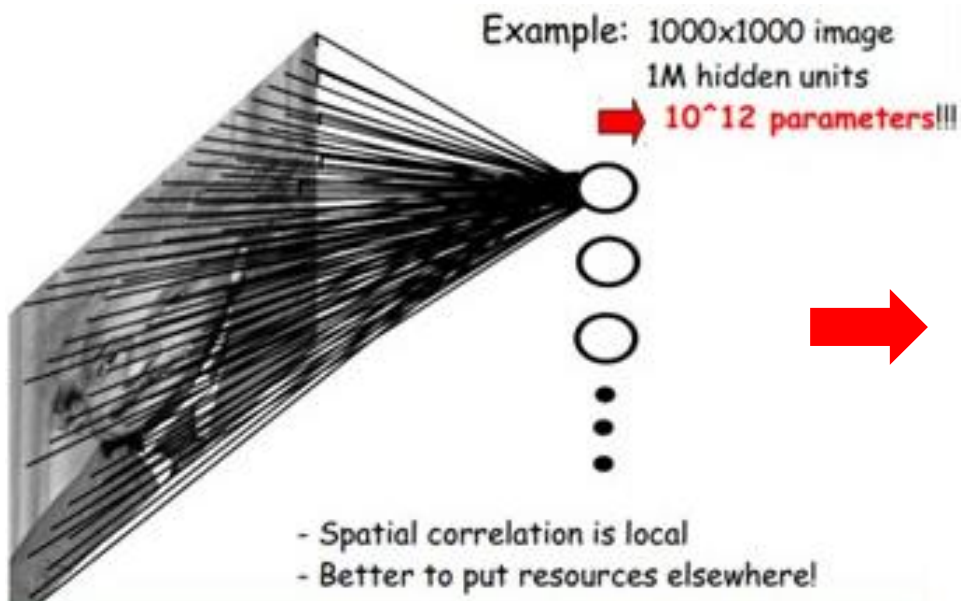
参数数目巨大

难以训练 — 梯度消失/爆炸

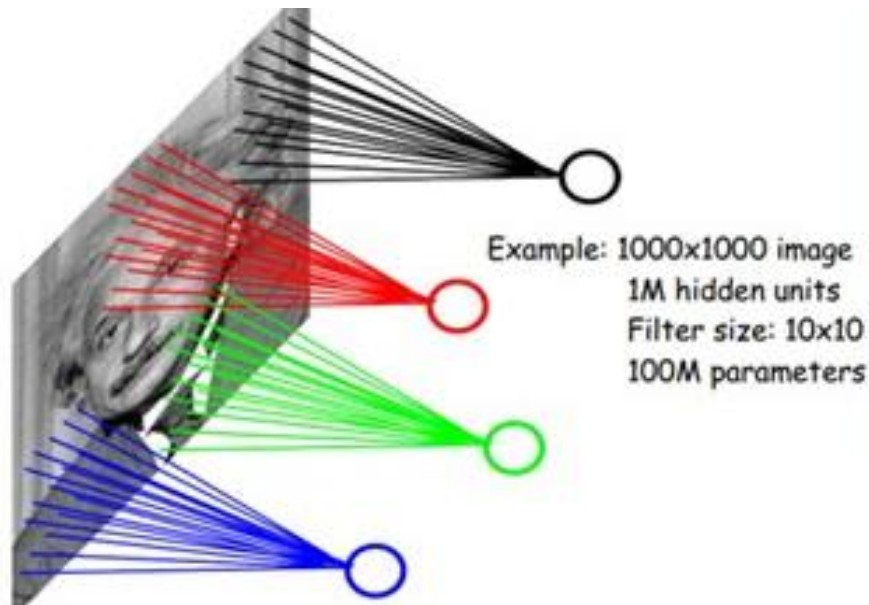


## 从神经网络到卷积神经网络：why CNN?

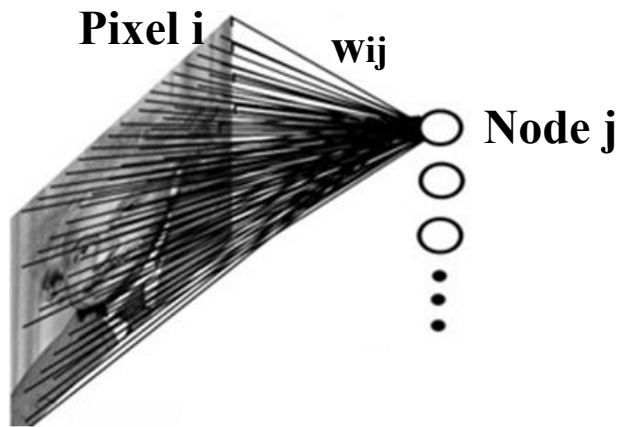
全连接



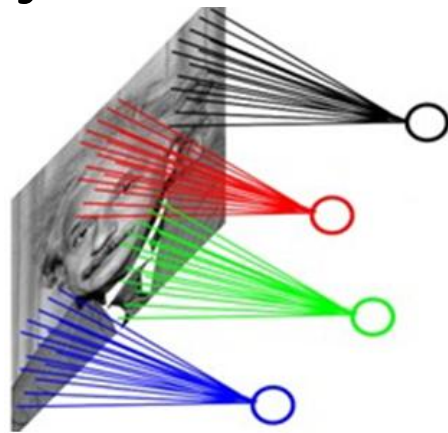
局部连接



## 从神经网络到卷积神经网络：why CNN?



$H \times W$ 的图像，与一组 $H \times W$ 个参数计算加权和，得到一个输出节点的值。



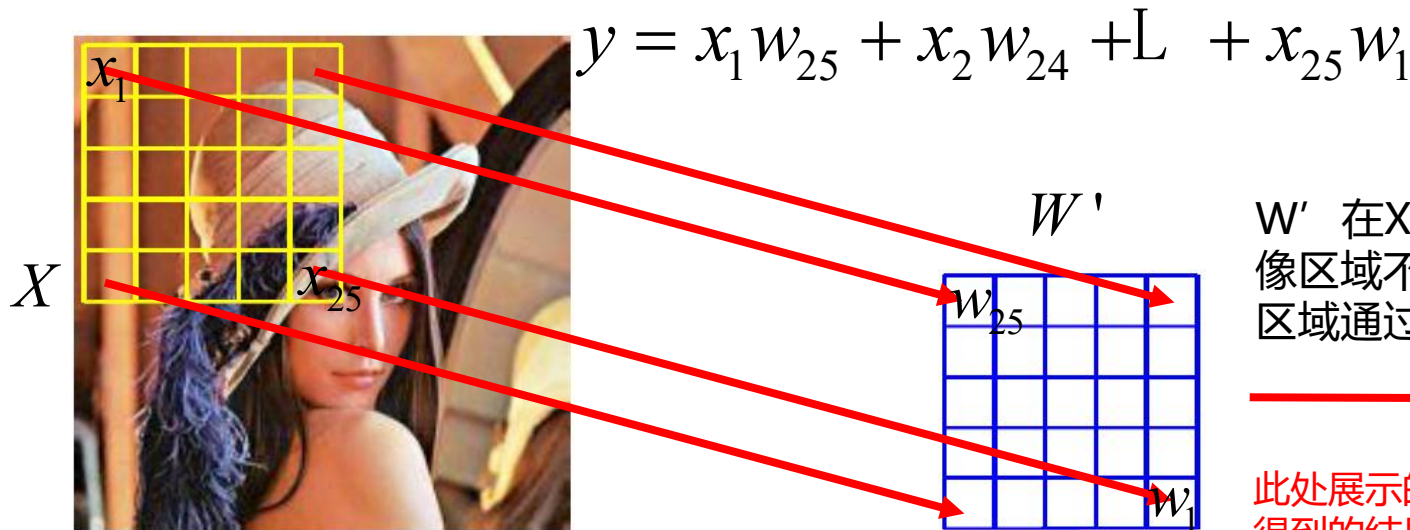
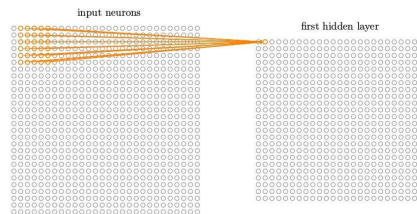
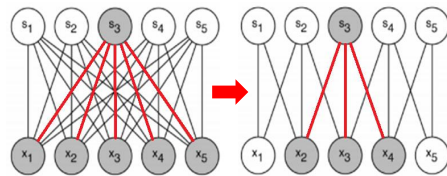
$H \times W$ 的图像，与四组 $h \times w$ 个参数计算加权和，得到四个输出节点的值。每组参数在一次计算中只与图像中 $h \times w$ 个像素产生关联。虽然计算的范围相比于左图有所缩减，但是计算方式是相同的。



# 基本概念

## 局部连接的实现方式：从一维推广到二维

最直接的方式：输入矩阵与参数矩阵对应位置元素相乘再相加（加权和），数学上称之为“**相关**”运算，记为  $Y = X \circ W'$



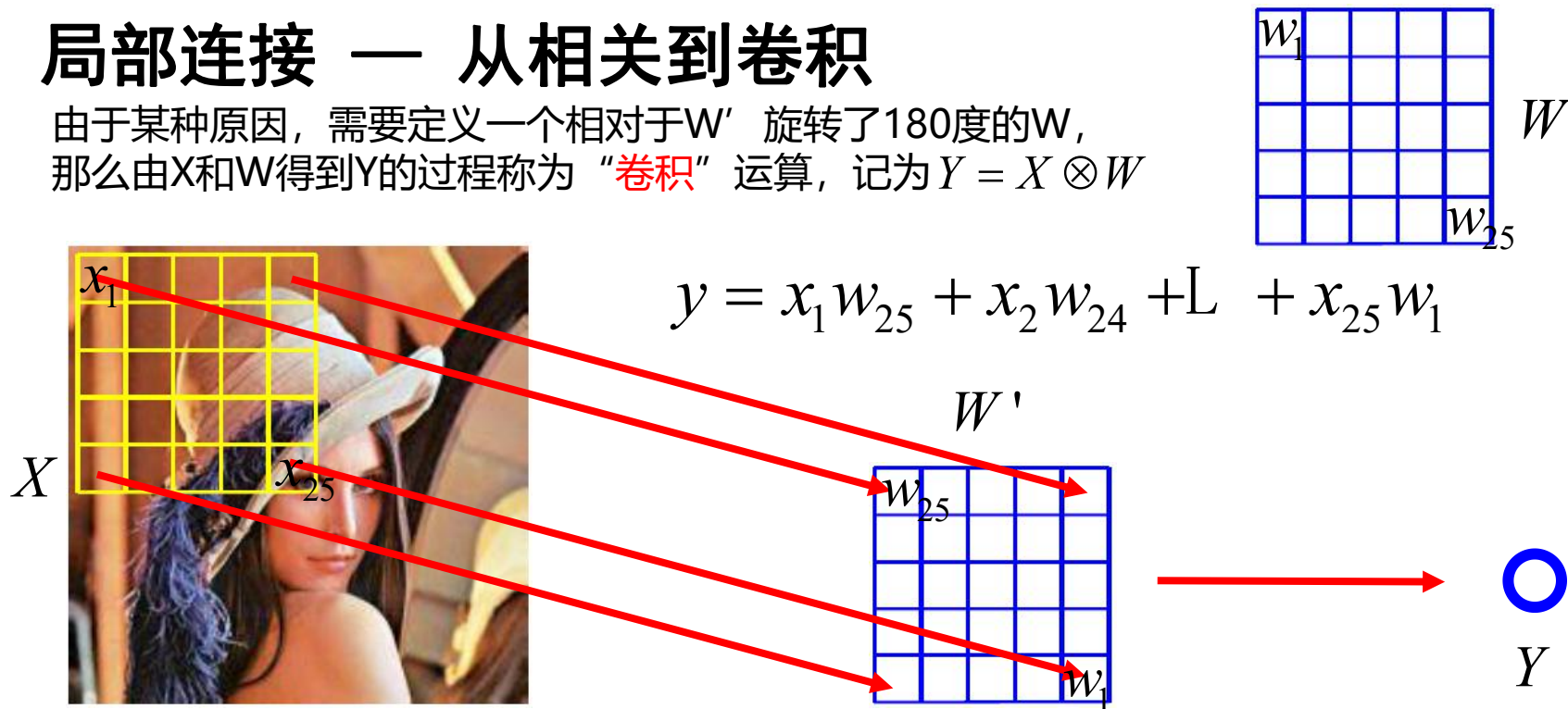
$W'$  在  $X$  上滑动，每次覆盖的图像区域不同，就可以得到不同区域通过局部连接得到的响应

→   $Y$

此处展示的只是  $W'$  滑动到一个位置得到的结果，并非整个过程的结果

## 局部连接 — 从相关到卷积

由于某种原因，需要定义一个相对于 $W'$  旋转了180度的 $W$ ，那么由 $X$ 和 $W$ 得到 $Y$ 的过程称为“卷积”运算，记为 $Y = X \otimes W$



## 局部连接 — 从相关到卷积

思考:  $C = A \otimes B$   
对于任意一组变量A和B, 卷积是如何进行的?

“卷积” = 1. W旋转180度得到W'  
2. X与W' 进行 “相关” 运算 (对应位置元素加权和)

$Y = X \otimes W$  中, W称为 “卷积核”

卷积核W在图像X上的滑动得到了整幅图像的响应 (通过局部连接得到的输出)

“卷积” 相比于 “相关” 的优势: 对称性  $X \otimes W = W \otimes X$

关于卷积运算与相关运算的对比可参考<https://www.zhihu.com/question/32067344?sort=created>  
关于二者更多、更本质的区别, 可以参考数字信号处理中的内容, 但是即使不知道这些内容也不影响学习卷积神经网络, 因为卷积在这里只是一种局部连接的计算方式。

## 卷积 - 数学描述

一维卷积

$$w(x) * f(x) = \int_{-\infty}^{\infty} w(u)f(x-u) du$$

二维卷积

$$w(x, y) * f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(u, v)f(x-u, y-v) du dv$$

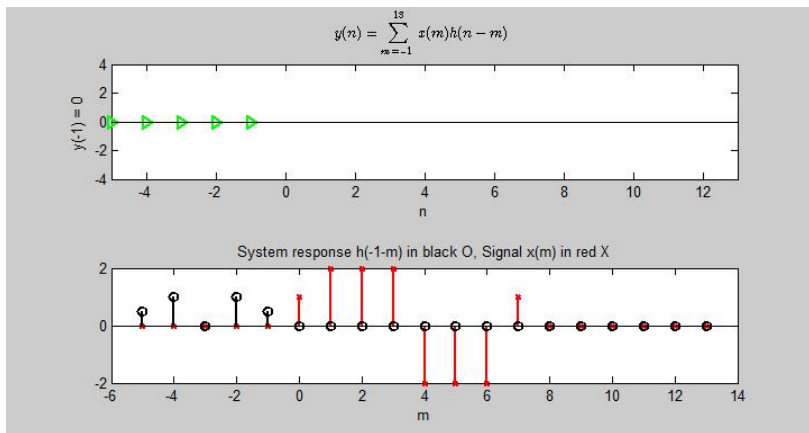


二维离散卷积

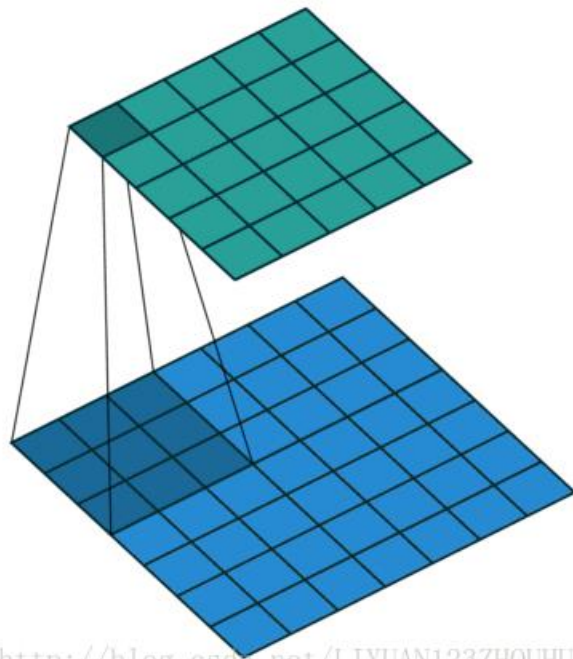
$$W(x, y) * F(x, y) = \sum_u \sum_v W(u, v)F(x-u, y-v)$$

# 基本概念

## 卷积 - 示意图



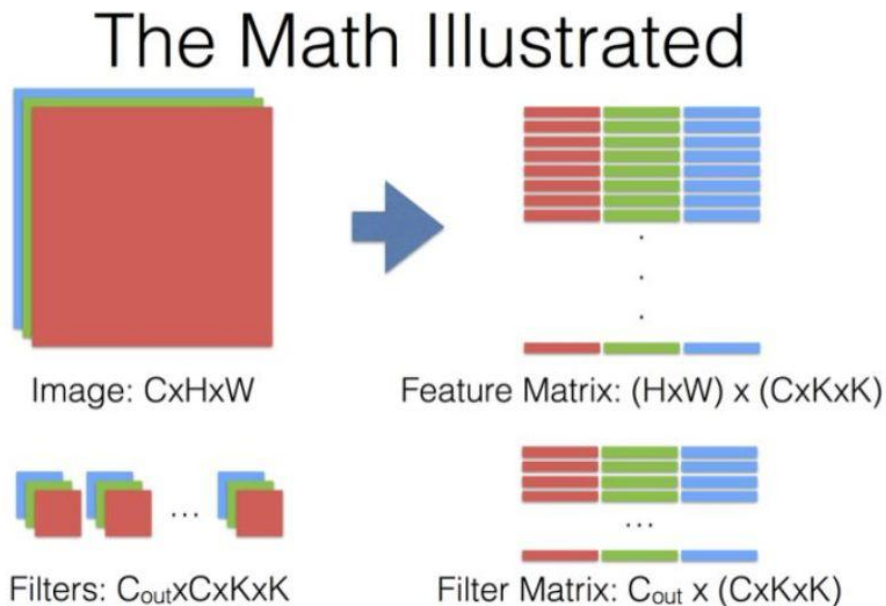
本质：计算信号（如图像）  
对某种特定模式的响应



<http://blog.csdn.net/LIYUAN123ZHOUHUI>

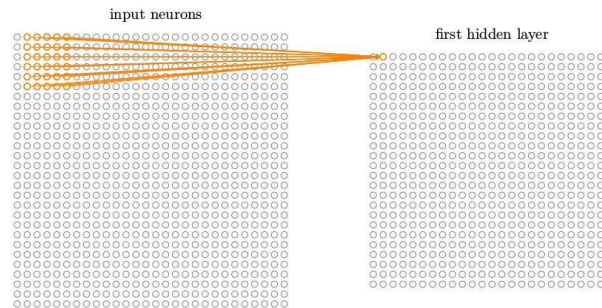
关于二维卷积的动态演示可参考<http://blog.csdn.net/liyuan123zhouhui/article/details/60139790>  
关于卷积的具体介绍可参考[http://blog.sina.com.cn/s/blog\\_7445c2940102wmrp.html](http://blog.sina.com.cn/s/blog_7445c2940102wmrp.html)

## 卷积 - 深度学习框架中的实际实现（以caffe为例）



## 卷积 - 作用

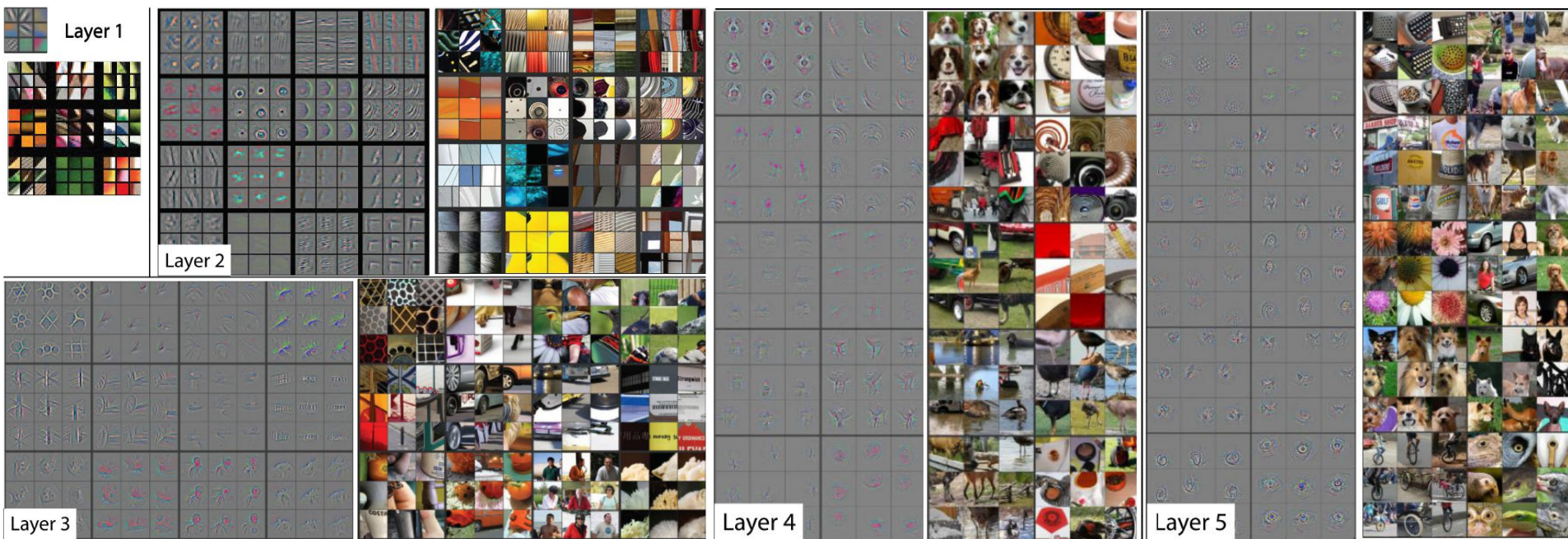
- 类似于数字图像处理中的各类算子（Sobel, Robert, Laplacian）。
- 用于提取特征，因此将得到的结果称为**特征图**（feature map）。
- 每个卷积核的初始值通常设定为随机数，其具体的功能（负责提取的特征，如直线、角点和颜色等）完全由网络学习来决定，而确定卷积核各个元素的值正是网络学习最主要的目的。





# 基本概念

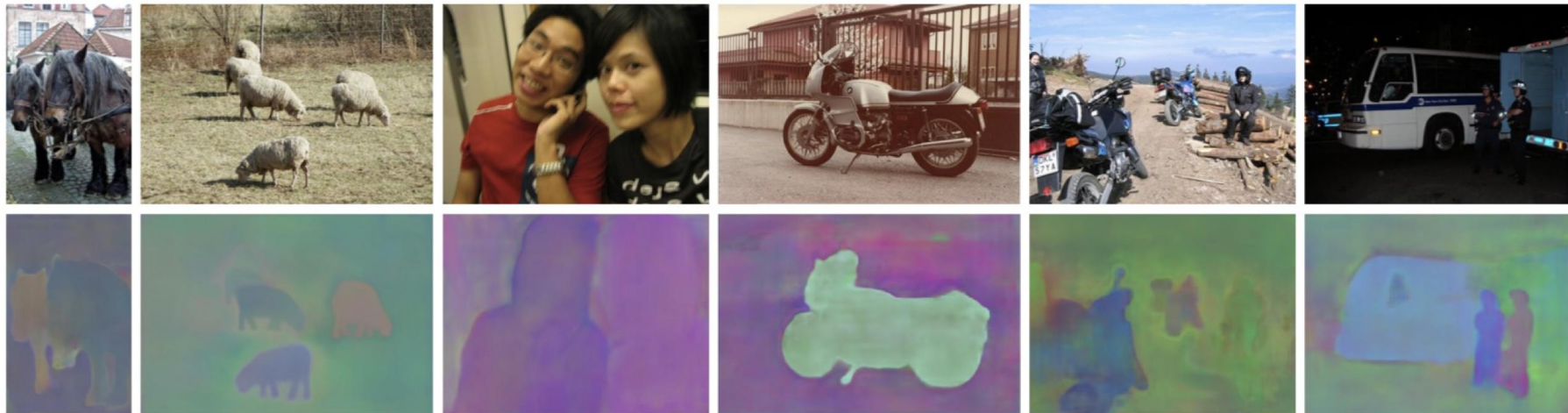
## 卷积 - 作用





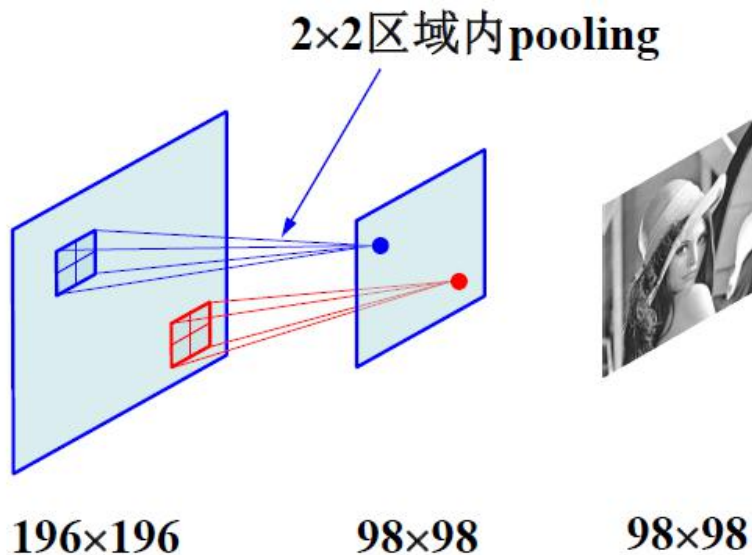
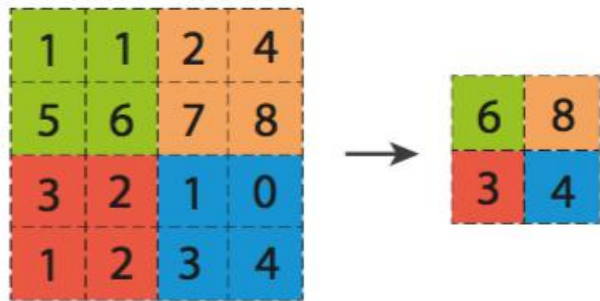
# 基本概念

## 卷积 - 作用



## 池化操作 (pooling)

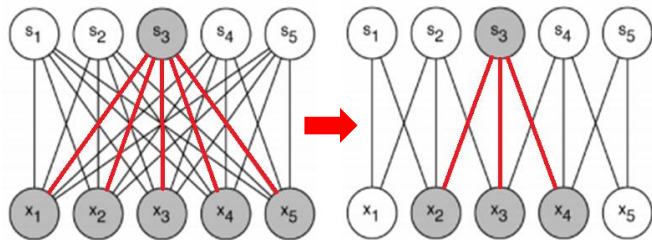
- 图像区域内关于某个特征的统计聚合。
- 两种方式：区域内取最大值或平均。
- 是一种下采样操作。



## 卷积神经网络与多层感知器的对比

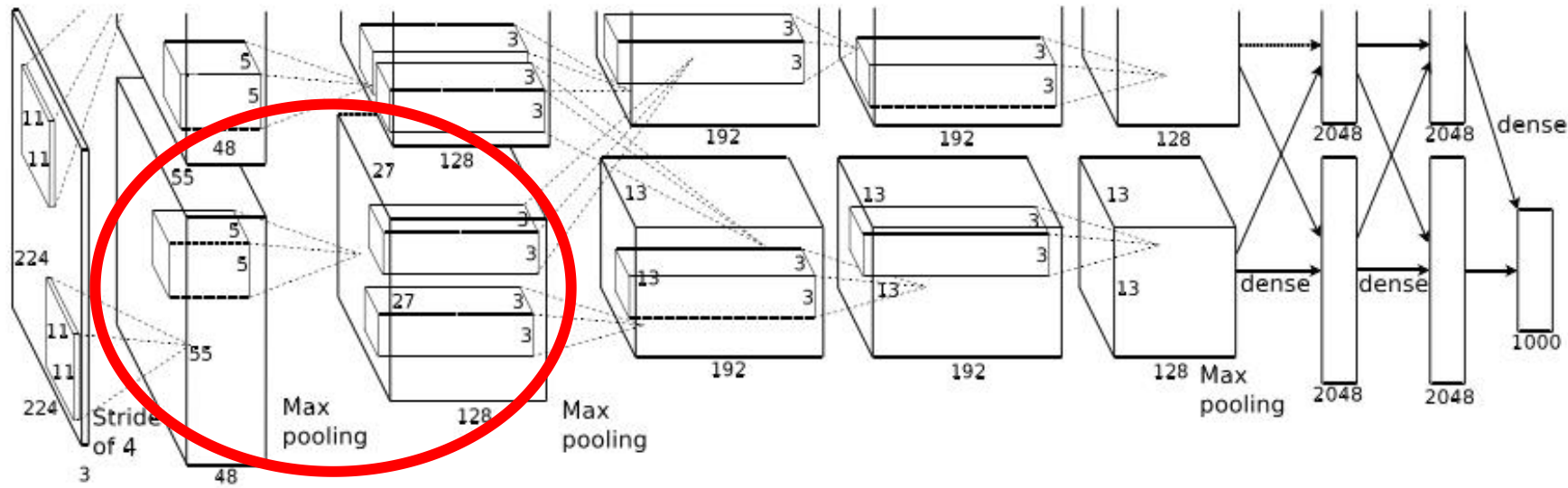
	多层感知器	卷积神经网络
隐含层数目	通常只有一层	多层
连接方式	全连接	局部连接、 <b>稀疏连接</b>
基本操作	乘法、加法、非线性映射	乘法、加法、非线性映射、 <b>池化</b>
相邻两层之间的参数数目	前一层点数*后一层节点数	<b>前一层特征图数目*卷积核尺寸*后一层特征图数目</b>
计算负担	一般较高	卷积操作容易高效实现

与图像本身的尺寸无关  
参数数目大幅度降低



# 基本概念

## 卷积神经网络与多层感知器的对比

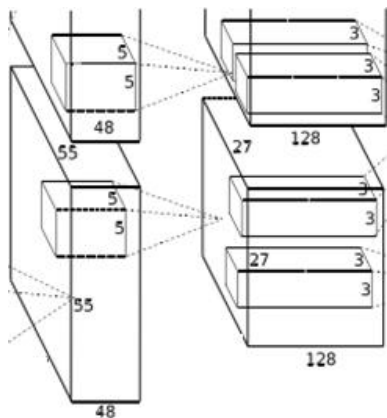


$$48 \times 5 \times 5 \times 128 = 153600$$

# 基本概念

## 补充：多通道的卷积如何实现？

$$\begin{aligned} y_{11}^n = & r_{11}w_{22}^n + r_{12}w_{21}^n + r_{21}w_{12}^n + r_{22}w_{11}^n \\ & + g_{11}v_{22}^n + g_{12}v_{21}^n + g_{21}v_{12}^n + g_{22}v_{11}^n \\ & + b_{11}u_{22}^n + b_{12}u_{21}^n + b_{21}u_{12}^n + b_{22}u_{11}^n \end{aligned}$$



$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & g_{11} & g_{12} & g_{13} \\ b_{31} & g_{21} & r_{11} & r_{12} & r_{13} \\ & g_{31} & r_{21} & r_{22} & r_{23} \\ & & r_{31} & r_{32} & r_{33} \end{bmatrix} \otimes \begin{bmatrix} u_{11}^n & u_{12}^n \\ u_{21}^n & v_{11}^n & v_{12}^n \\ & v_{21}^n & w_{11}^n & w_{12}^n \\ & & w_{21}^n & w_{22}^n \end{bmatrix} \Rightarrow \begin{bmatrix} y_{11}^n & y_{12}^n \\ y_{21}^n & y_{22}^n \end{bmatrix} \quad Y^n = X \otimes W^n$$

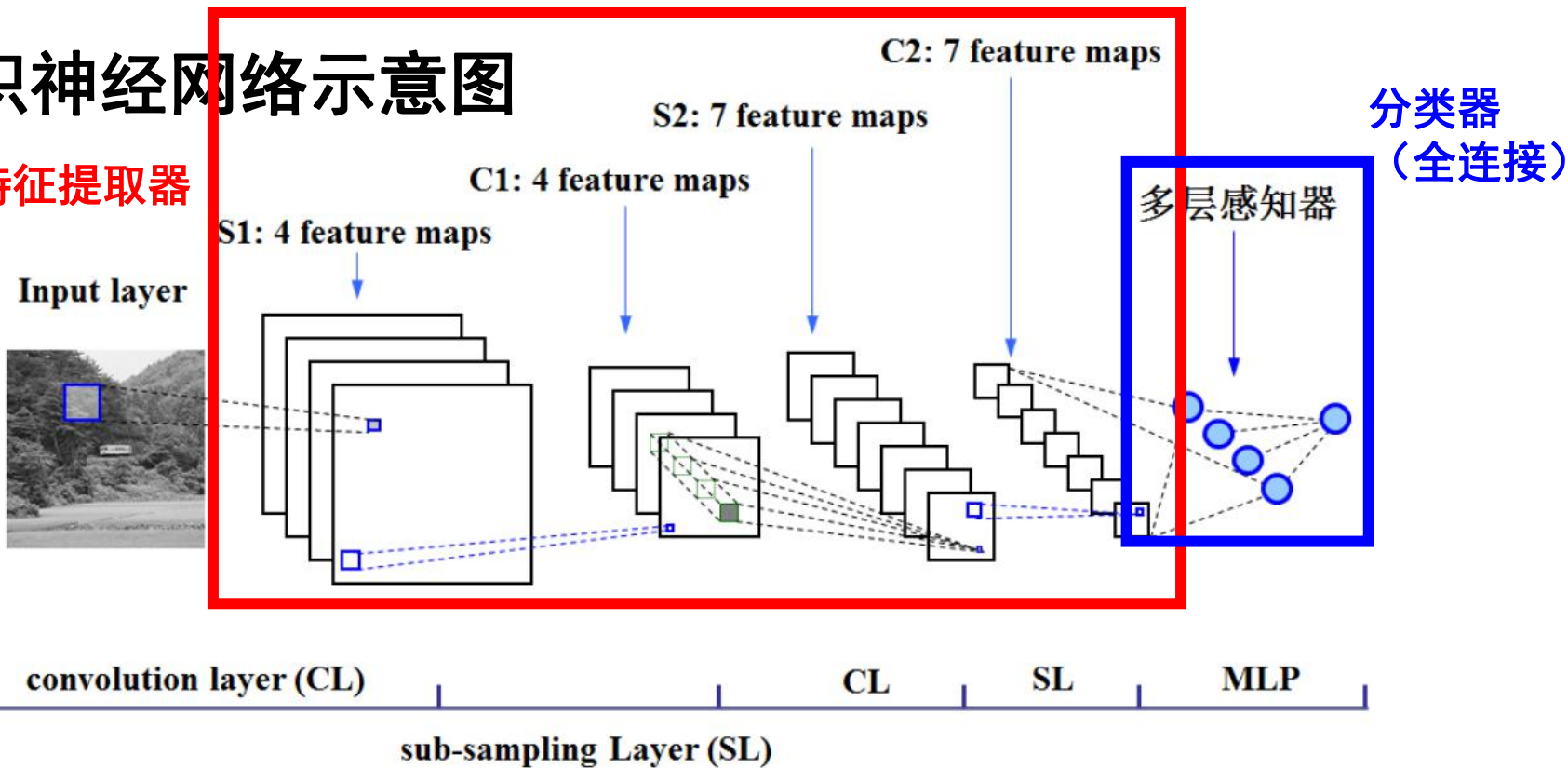
$X$ ：尺寸为 $3 \times 3$ 的RGB三通道图像  $W^n$ ：一个尺寸为 $2 \times 2$ 的三通道卷积核

如果有 $N$ 个卷积核 $\{W^n\}_1^N$ ，则输出有 $N$ 个通道 $\{Y^n\}_1^N$ 。

# 基本概念

## 卷积神经网络示意图

特征提取器



# 课程大纲

---

✓ 基本概念

✓ 发展历程

✓ 网络特点

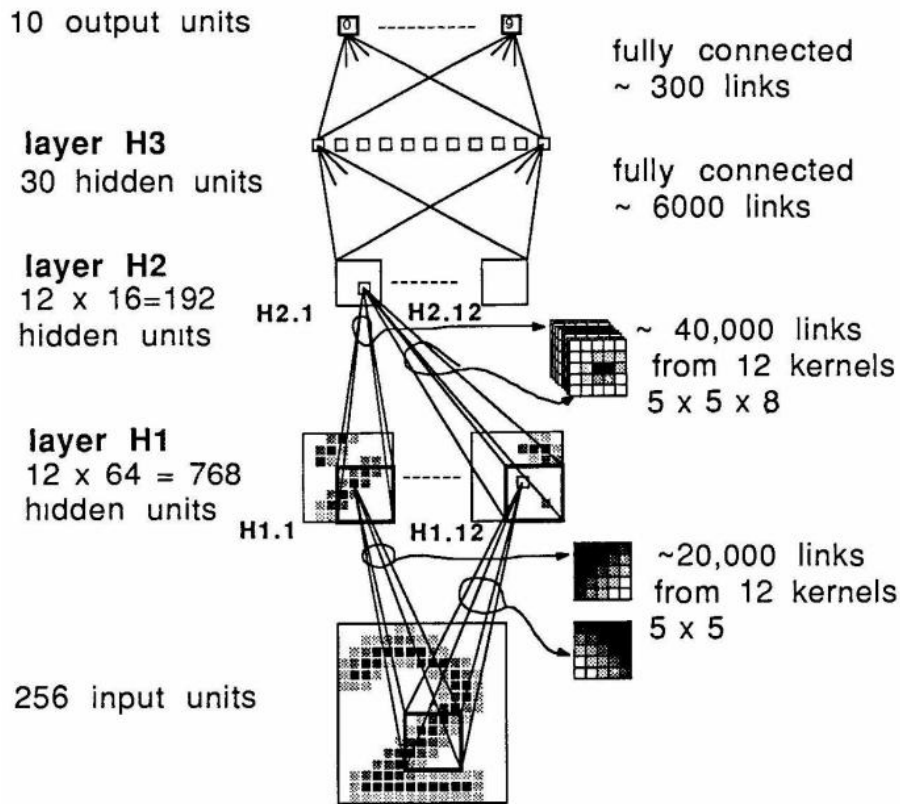


# 发展历程

## LeNet — 图像分类

- 1989年由Yann LeCun提出。
- 应用于手写数字（邮政编码）识别。
- 卷积神经网络最初的形式，  
也是最基本的形式，目前仍在沿用。

Y. LeCun et al., Backpropagation applied to handwritten zip code recognition. Neural Computation 1989.





# 发展历程

## LeNet — 图像分类

- 1989年由Yann LeCun提出。
- 应用于手写数字（邮政编码）识别。
- 卷积神经网络最初的形式，  
也是最基本的形式，目前仍在沿用。

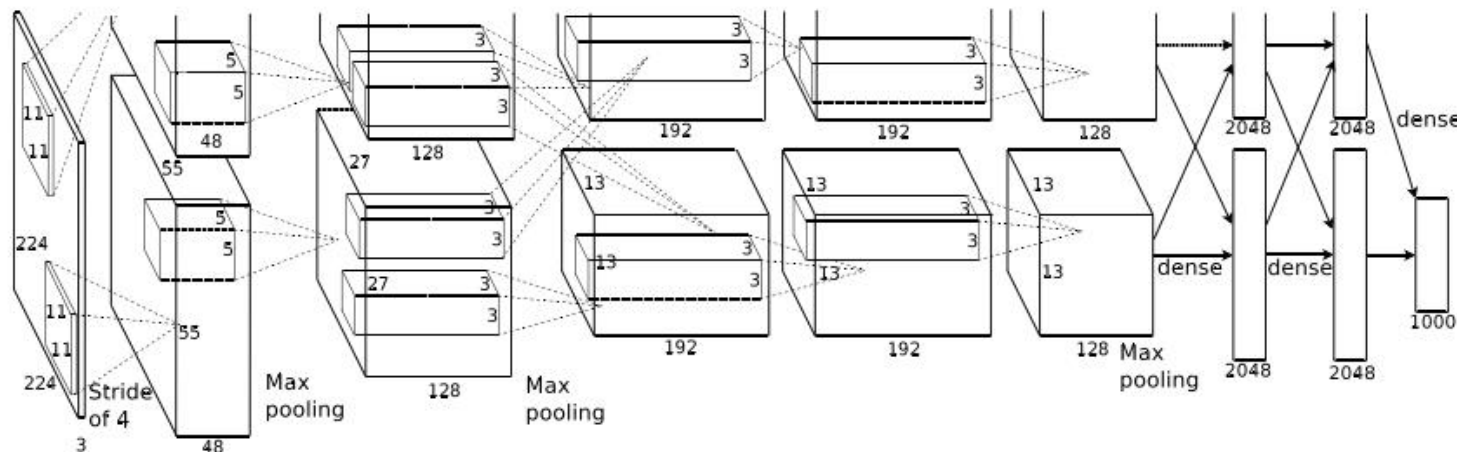
Y. LeCun et al., Backpropagation applied to handwritten zip code recognition. Neural Computation 1989.



Hinton是DBN的发明者，  
但却凭借CNN获奖。

## AlexNet — 图像分类

- LeNet虽然取得了一定的效果，但并未受到广泛关注。
- 2012年的ImageNet比赛中，Alex Krizhevsky等对其进行了成功的应用。



## AlexNet的重要意义

- 证实了卷积神经网络可以利用大规模训练数据得到优秀的分类效果，从此以后卷积神经网络在计算机视觉领域便得到飞速发展和应用。
- 奠定了GPU在深度学习中的重要作用（同时也使NVIDIA公司得到了巨大的发展）。
- 多种网络设置和训练的技巧，如ReLU、Dropout等，使深层网络的训练成为可能。
- 网络学习到的特征对分类效果具有一定的解释性。

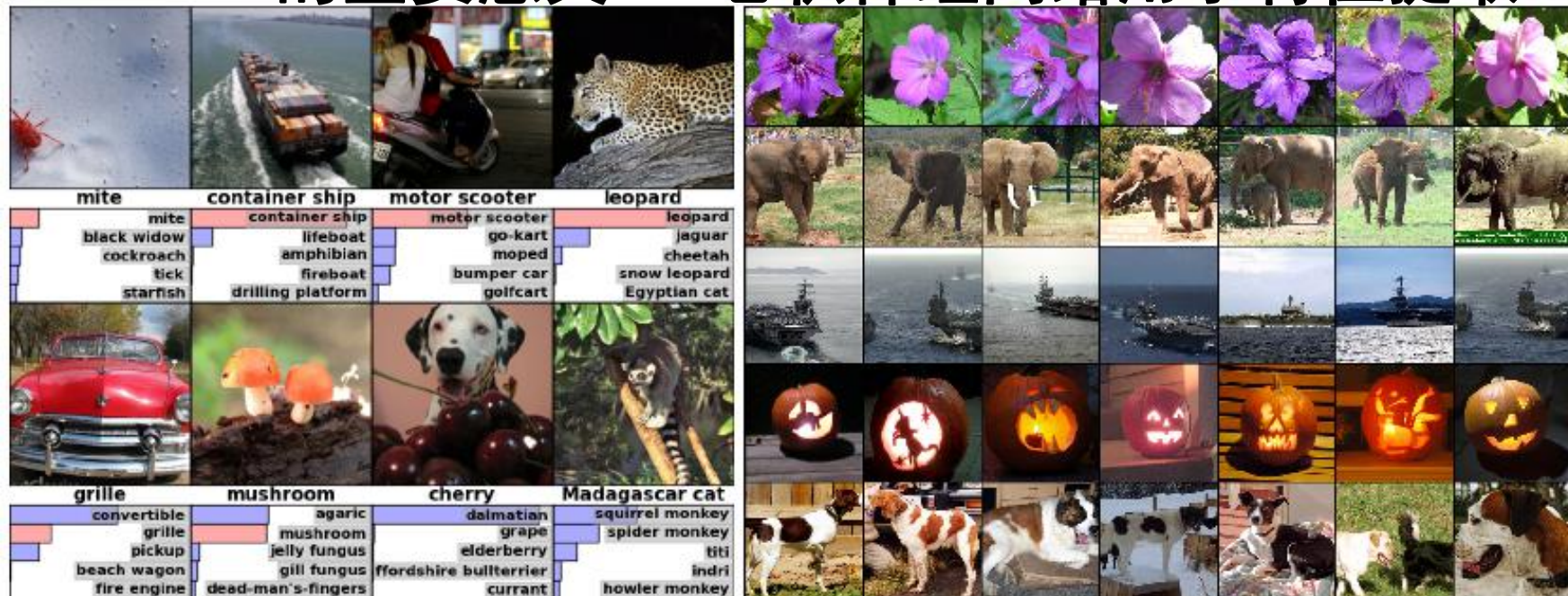
## AlexNet的重要意义 — 卷积神经网络用于特征提取

学习到的卷积核



# 发展历程

## AlexNet的重要意义 — 卷积神经网络用于特征提取



A. Krizhevsk et al., ImageNet Classification with Deep Convolutional Neural Networks. NIPS 2012.



# 发展历程

## R-CNN — 从图像分类到目标检测

目标检测的主要流程：

image



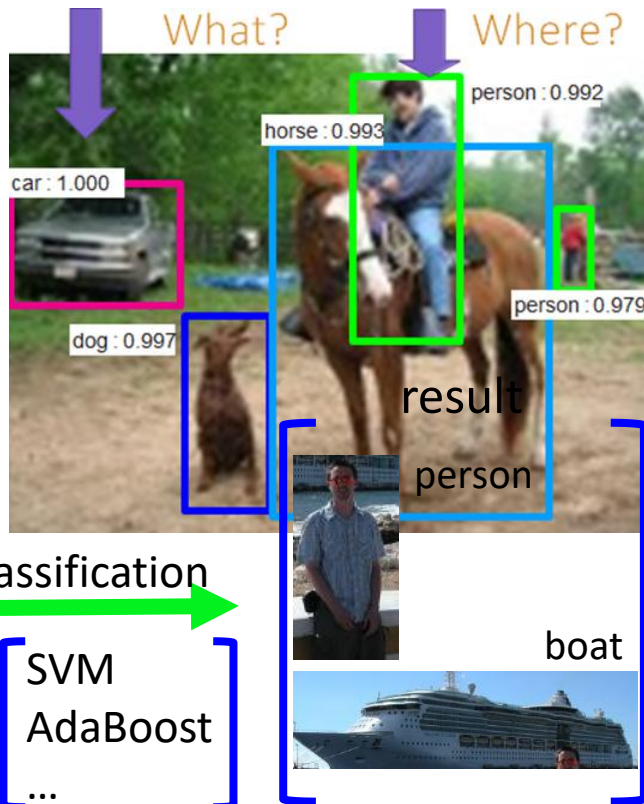
Objectness  
estimation  
→

object proposals



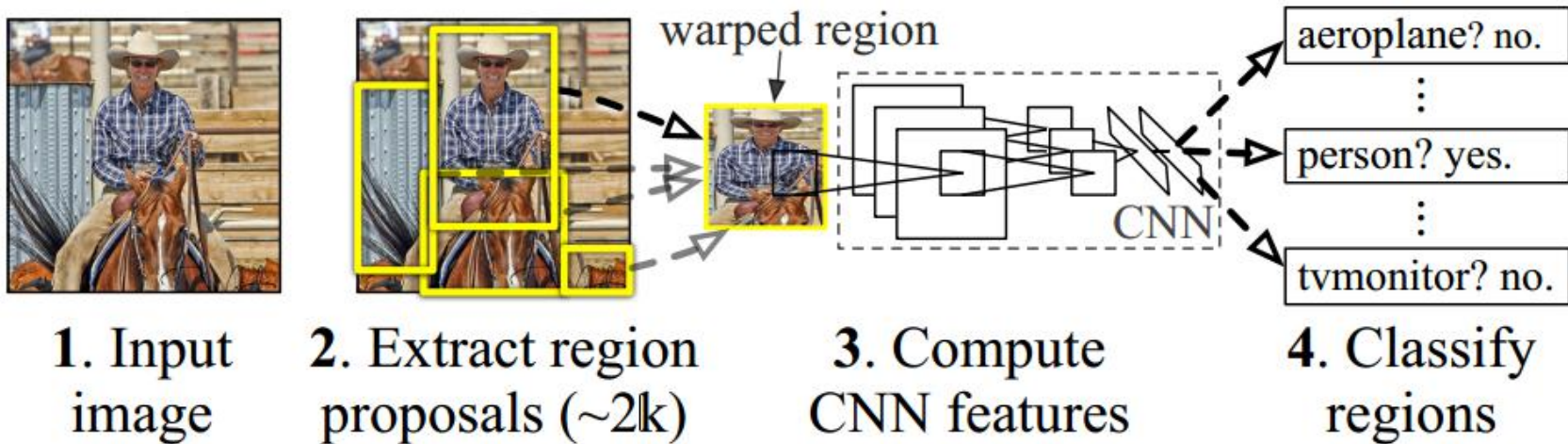
Classification  
→

SVM  
AdaBoost  
...



## R-CNN — 从图像分类到目标检测

*R-CNN: Regions with CNN features*



R. Girshick et al., Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR 2014.

## R-CNN的重要意义

- 证实了可以将图像的局部作为基本研究对象，将原问题转化为图像分类问题，利用卷积神经网络以图像分类的方式进行解决。
- 引出了卷积神经网络在计算机视觉中极为丰富的各类应用：
  - fast R-CNN, faster R-CNN, SSD, Mask R-CNN.....
  - 图像分割、图像描述、目标解析.....

R. Girshick et al., Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR 2014.



## R-CNN的重要意义

Ross B. Girshick et al. "Deformable Part Models are Convolutional Neural Networks." Technical report.

- 在使用CNN之前，目标检测主要采用的方法是DPM (Deformable parts models)，精度只能到43%。
- Ross B. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." CVPR 2014. (66.0%, 47s)
- Ross B. Girshick. "Fast r-cnn." CVPR 2015. (66.9%, 0.22s)
- Shaoqing Ren, Kaiming He, Ross B. Girshick et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." NIPS 2015. (73.2%, 0.196s including all steps)

# R-CNN

## RCNN (ECCV2014)

- selective search,  
根据颜色, 边缘,  
1. 纹理等等快速的找到的可能存在的目标候选框

1.1 475张, VOC2007上的检测结果从DPM HSC的34.3%直接提升到了66%(mAP)

1.2 Proposal归一化到 $227 \times 227$ , CNN只对一个图片ROI图片的提特征, 分类还是SVM, 最终有对分类好的proposal的回归

1.3 问题在于每一个图像块进来都要用CNN算一下特征, 其实整张图算一次就好了

## Fast RCNN (ICCV2015)

- 加入SPPnet,  
end to end 训练, 使用了回归

1.1 35张, Map70%, 仍然网络外部给Proposal

1.2 ROI pooling: 类似于SPP, 但只有一种 $7 \times 7$ 网格, 下采样得到 $49 \times 512$ 维度的特征 (只有全连接层才对Size有要求)

1.3 损失函数使用了多任务损失函数(multi-task loss), 将边框回归直接加入到CNN网络中训练

1.4 SPP对任意输入的Feature Map 做了金字塔Pooling: 对Map 划成 $4 \times 4$ ,  $2 \times 2$ ,  $1 \times 1$ 三种网格, 然后做pooling: 得到固定的FC输入:  $(16+4+1) \times \text{channels}$  维度

## Faster RCNN (NIPS2015)

- 使用网络直接产生召回率高的Proposals: RPN网络

1.1 5FPS, mAP73.2%

1.2 加入了9种 anchors (3种尺度, 3种比例), 总共输出20000~proposals

1.3 输入的特征proposal接入到ROI Pooling

## YOLO (CVPR2016)

- 变为回归问题来做

1.1 45FPS, mAP57.9%

1.2 整张图划为 $7 \times 7$ 网格, 每一个格子预测两个目标, 输出的结果有置信度+坐标位置

1.3 并没有使用Region proposal,  $7 \times 7$ 比较粗燥, 小目标不好

## SSD (ECCV2015)

- YOLO +  
1. Proposal + 多尺度

1.1 58FPS, mAP73.9%

1.2 整张图 $8 \times 8$ 网格+anchors+FCN

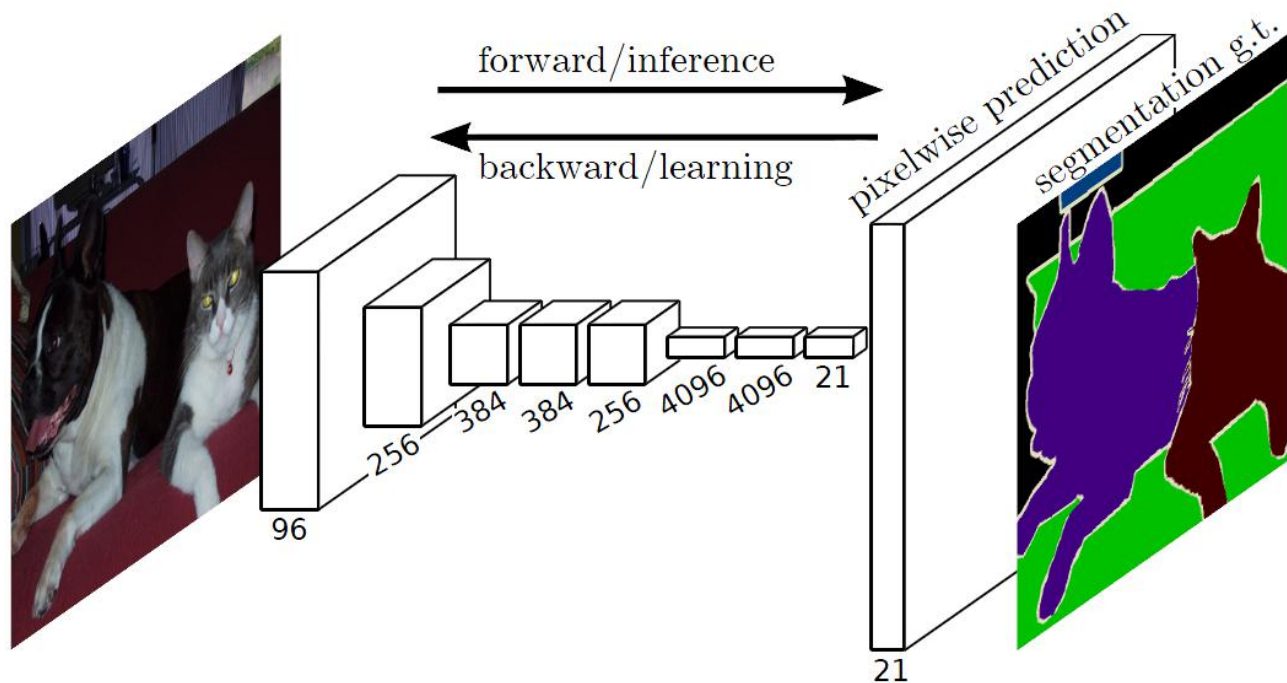
1.3 不同层的feature map  $3 \times 3$ 滑动窗感受野不同, 作为不同尺度的检测

## FCN

- 在FCN出现之前的图像分割，一般将图像的局部作为基本研究对象，如像素、超像素、图像块（矩形区域或者不规则区域）等，通过对它们进行分类来实现图像分割。
- FCN则采用“**整幅图像输入，整幅分割结果输出**”这种“端到端”（end-to-end）式的结构，能够接受任意尺寸的输入图像。
- 第一个“端到端”式的图像分割模型。

# 发展历程

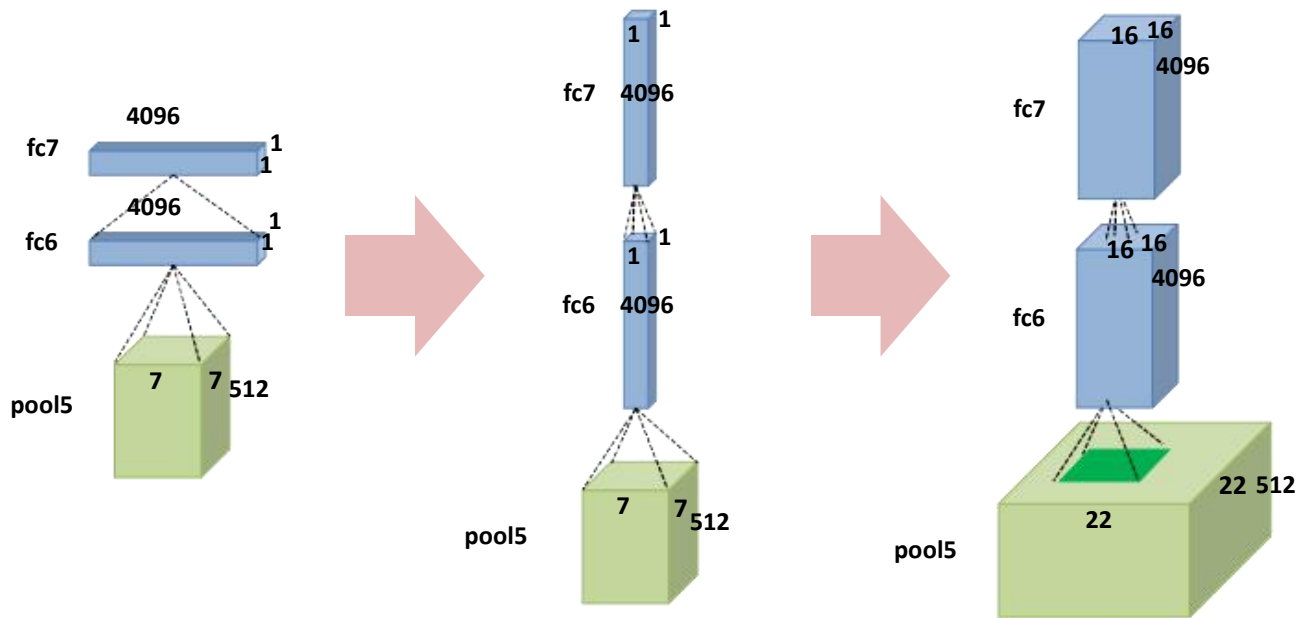
## FCN



J. Long et al., Fully convolutional networks for semantic segmentation. CVPR 2015.

# 发展历程

## FCN



**Fully connected  
layers**

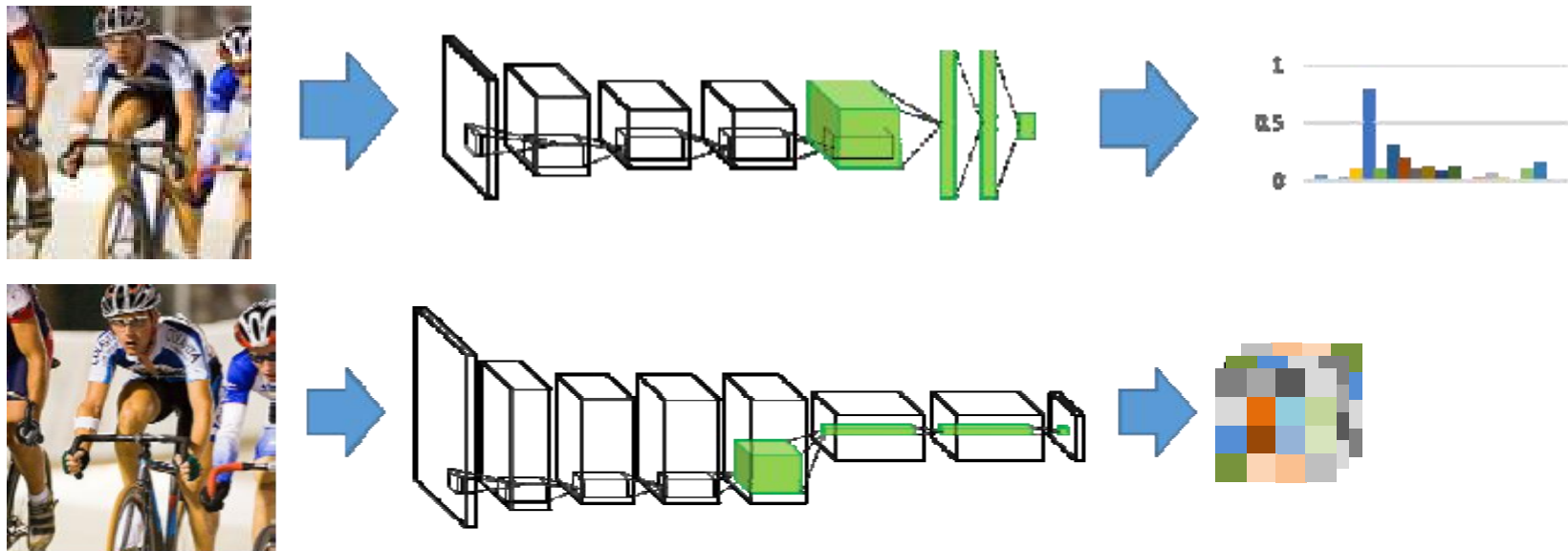
**Convolution  
layers**

**For the larger Input field**

J. Long et al., Fully convolutional networks for semantic segmentation. CVPR 2015.

# 发展历程

## FCN



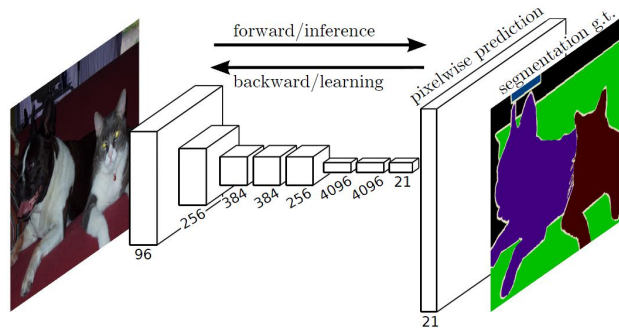
## FCN的重要意义

- 充分保留了全局的语义信息。
- 减少了重复计算（如果采用图像块，图像块之间会有overlap），提高了训练和测试的效率。
- 去除了对输入图像尺寸的限制（因为没有全连接层，每层的节点数目于参数数目不再直接相关）。
- 全卷积网络成为许多后续网络模型的重要基础：Deeplab, CRFRNN...
- “端到端”式结构成为后续网络设计的一项重要标准。

# 发展历程

## Conv-Deconv Net

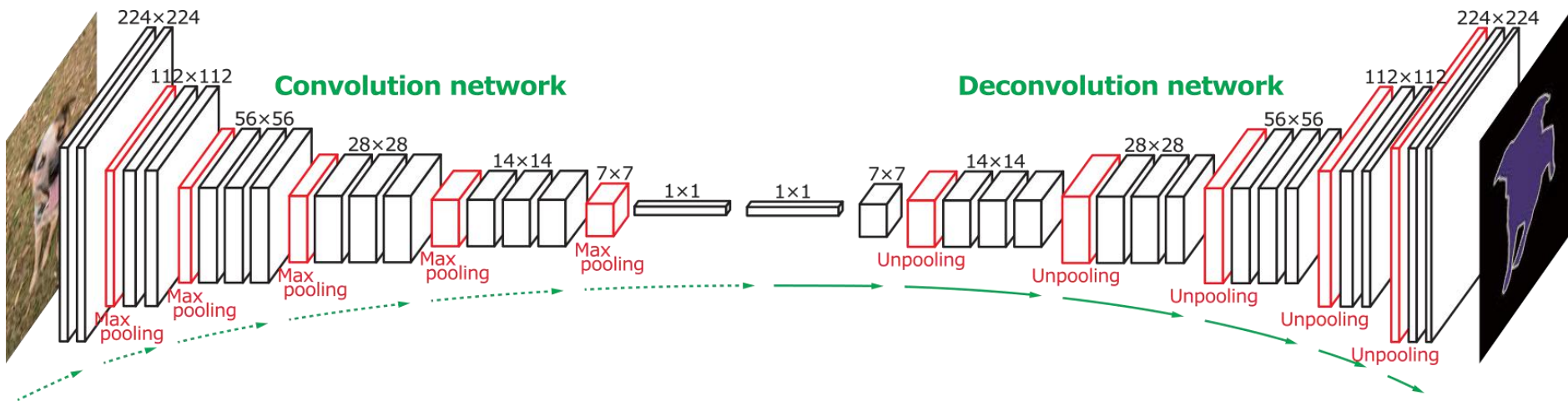
- FCN中特征图经过逐级下采样，尺寸大大减小，最后经过一次性的上采样恢复输入尺寸。
- 整个过程损失了太多的细节信息。
- Conv-Deconv Net采用与卷积神经网络对称的结构，通过逐级上采样来恢复特征图的尺寸。





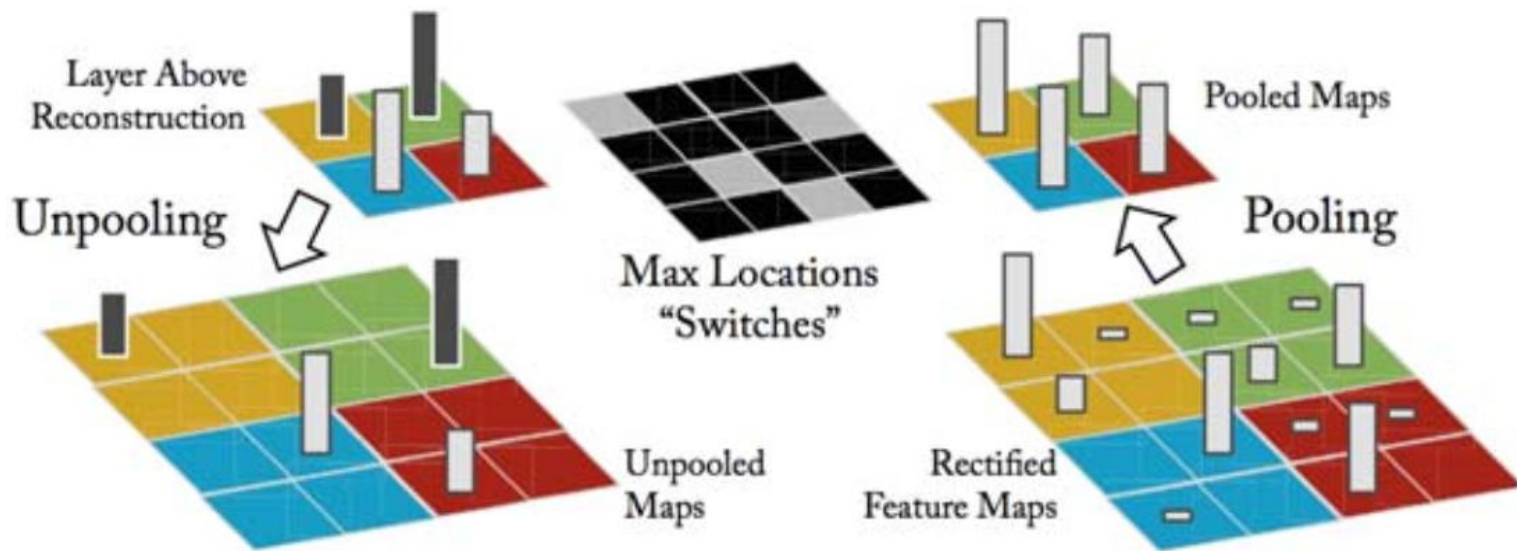
# 发展历程

## Conv-Deconv Net



H. Noh et al., Learning deconvolution network for semantic segmentation. ICCV 2015.

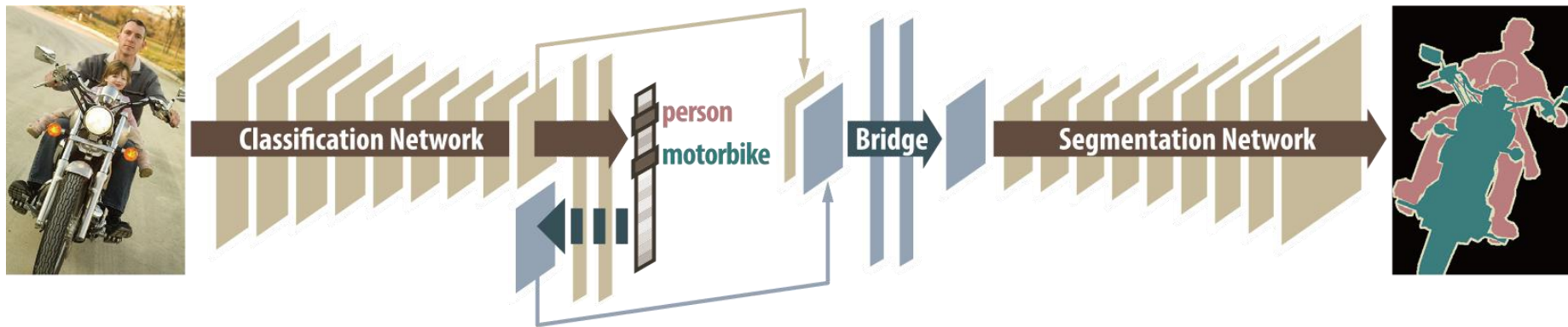
## Conv-Deconv Net中的下采样与上采样



## Conv-Deconv Net的重要意义

- 逐级的上采样，实现了以“由粗到精”的方式恢复细节信息。
- 弥补了全卷积网络的不足（一次性高倍数上采样）。
- 这种“沙漏形”的对称网络结构成为众多像素级标注问题的基础。

## Conv-Deconv形式的各种模型

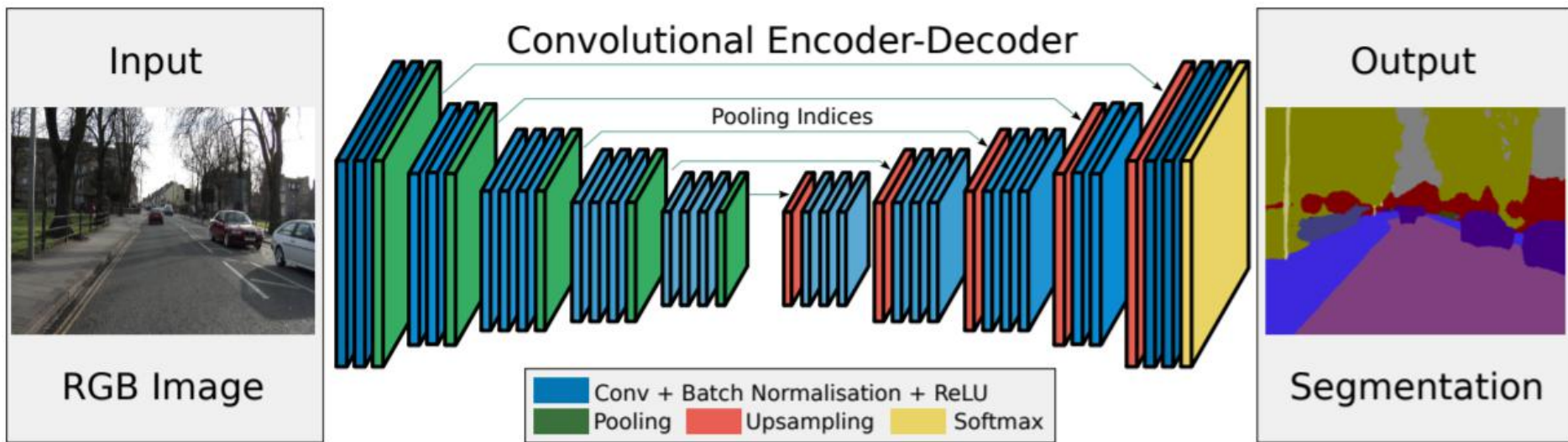


S. Hong et al., Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation. NIPS 2015.

# 发展历程

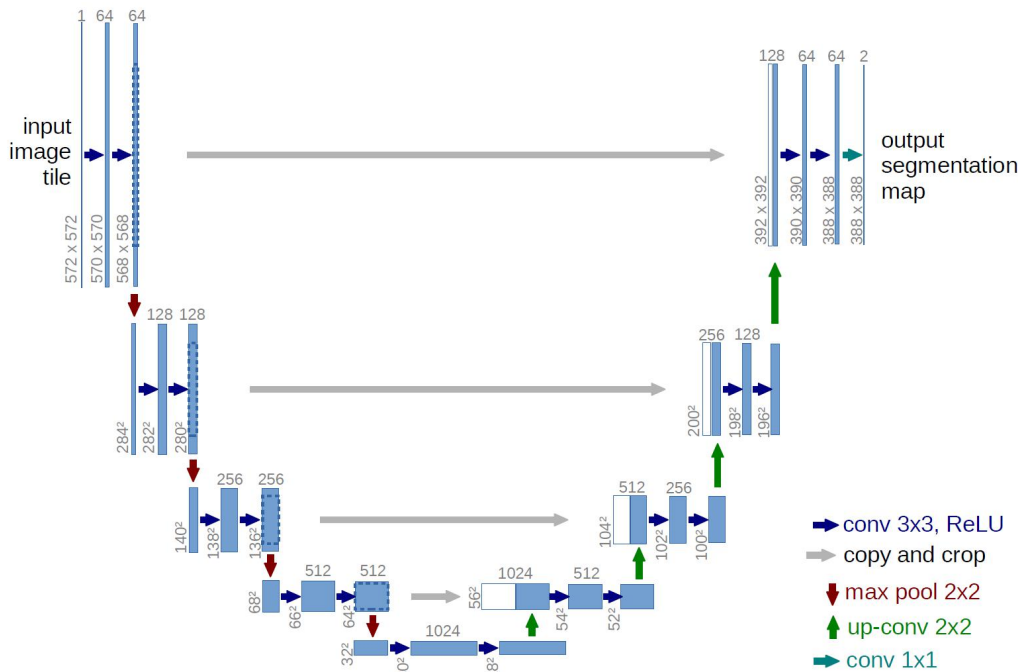
## Conv-Deconv形式的各种模型 (SegNet)

相比于Conv-Deconv模型，  
最大区别在于去掉了  
中间7\*7的池化



V. Badrinarayanan et al., SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, arXiv 2016.

## Conv-Deconv形式的各种模型（U-Net）

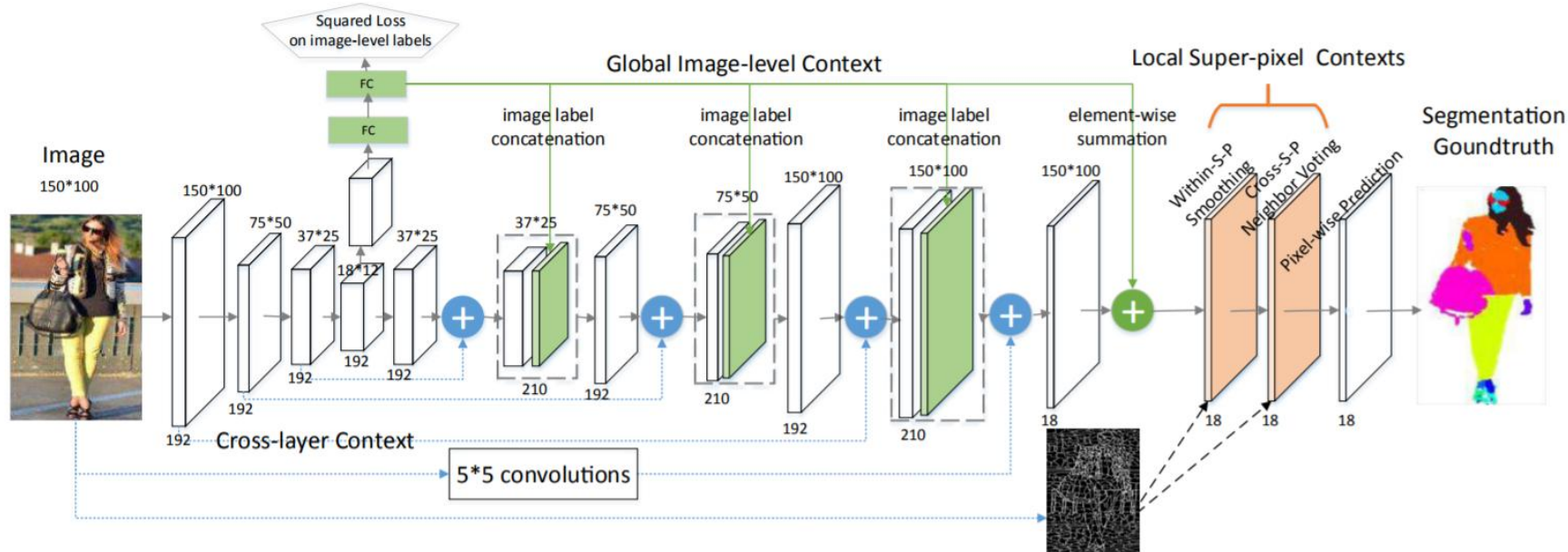


两阶段的feature map结合起来,  
更好地保留了细节信息

## 面试中常被问到的问题：如何提高分割结果中边界部分的精度？

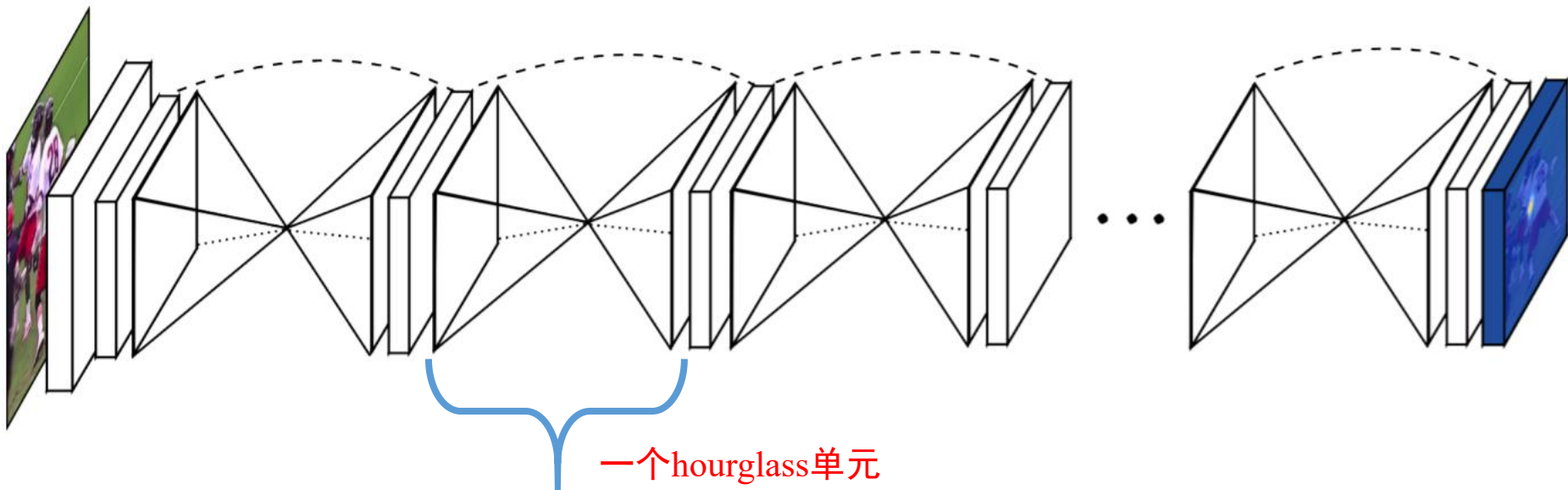
O. Ronneberger et al., U-Net:  
Convolutional Networks for  
Biomedical Image Segmentation.  
MICCAI 2015.

## Conv-Deconv形式的各种模型



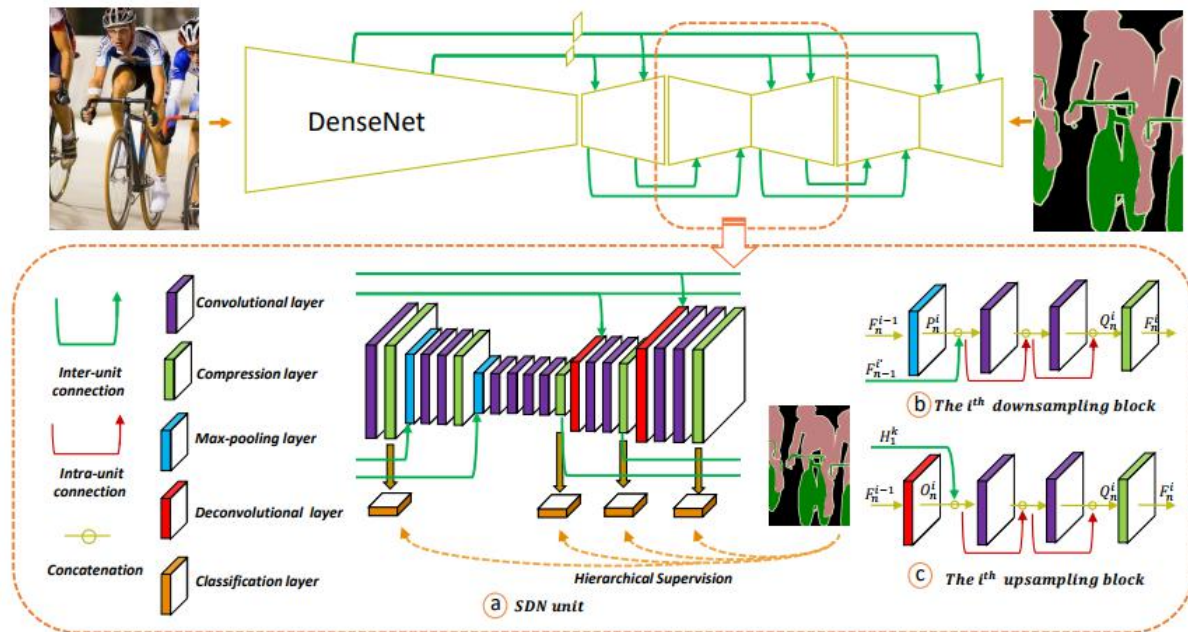


## Conv-Deconv形式的各种模型



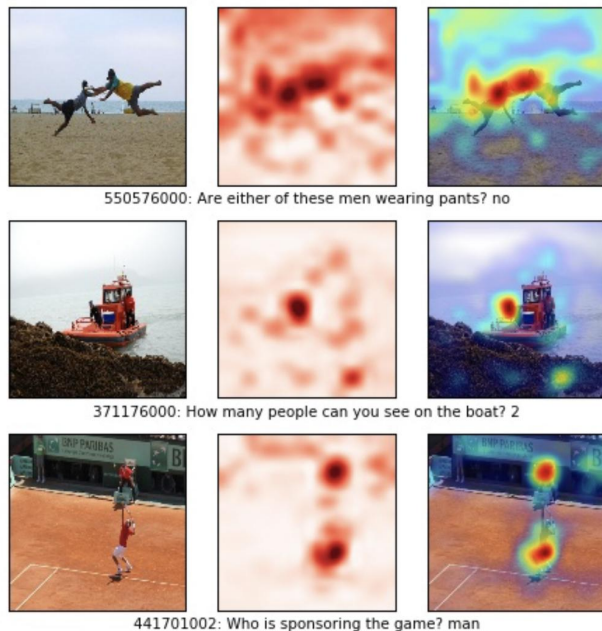
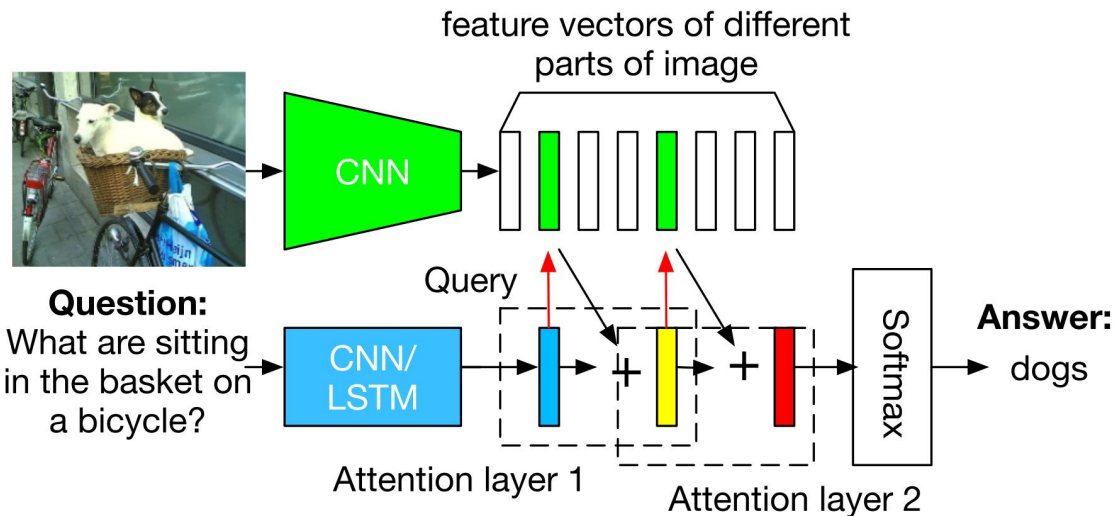
A. Newell et al., Stacked Hourglass Networks for Human Pose Estimation. arXiv 2016.

## Conv-Deconv形式的各种模型

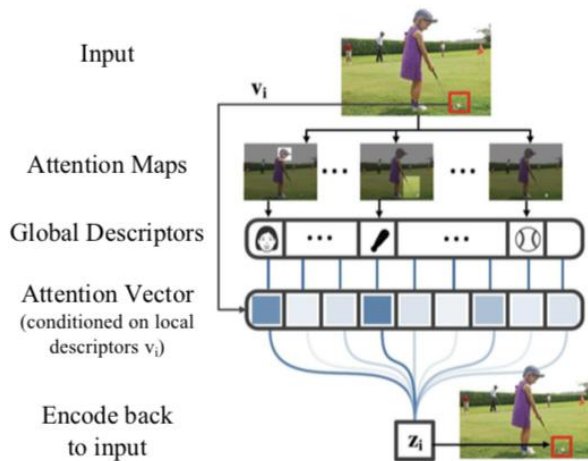


# 发展历程

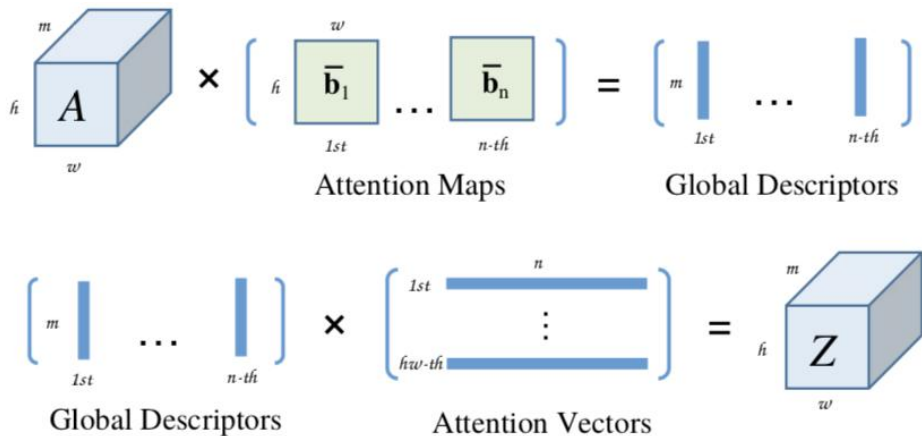
## Attention网络



## Attention网络



(a) Double Attention Method



The diagram visualizes the Double Attention Operation using tensor notation. It shows the multiplication of a 3D tensor  $A$  (with dimensions  $m$ ,  $f_i$ , and  $w'$ ) by a sequence of 2D tensors  $\bar{b}_1, \dots, \bar{b}_n$  (each with dimensions  $f_i$  and  $w'$ ). The result is a sequence of 2D tensors (Global Descriptors) with dimensions  $m$  and  $f_i$ . This sequence is then multiplied by a 2D tensor (Attention Vectors) with dimensions  $f_i$  and  $n$ . The final result is a 3D tensor  $Z$  (with dimensions  $m$ ,  $f_i$ , and  $w'$ ).

(b) Visualization of the Double Attention Operation

# 课程大纲

---

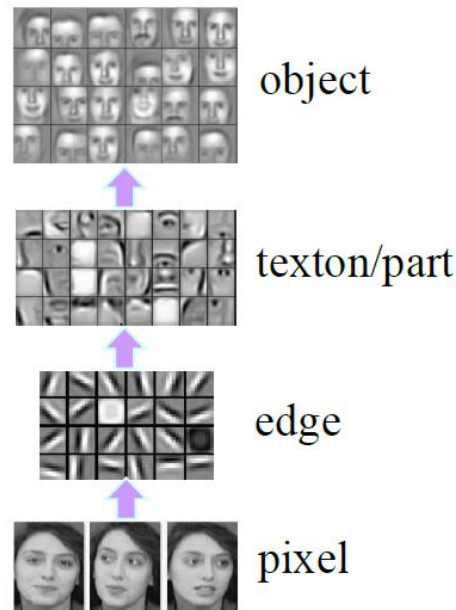
✓ 基本概念

✓ 发展历程

✓ 网络特点

## 层次化的特征学习

- 直接从图像底层开始进行学习。随着网络层数的加深，能够学习到更高层的语义信息。
- 不必事先提取人为设计的特征，比如Gabor纹理特征、多尺度小波特征、SIFT特征、HOG特征等。
- 人为设计特征的缺点：这些特征具有一些参数，如尺度、梯度方向、频域划分等，其泛化能力不强。



思考：自动学习的特征优于人为设计的特征



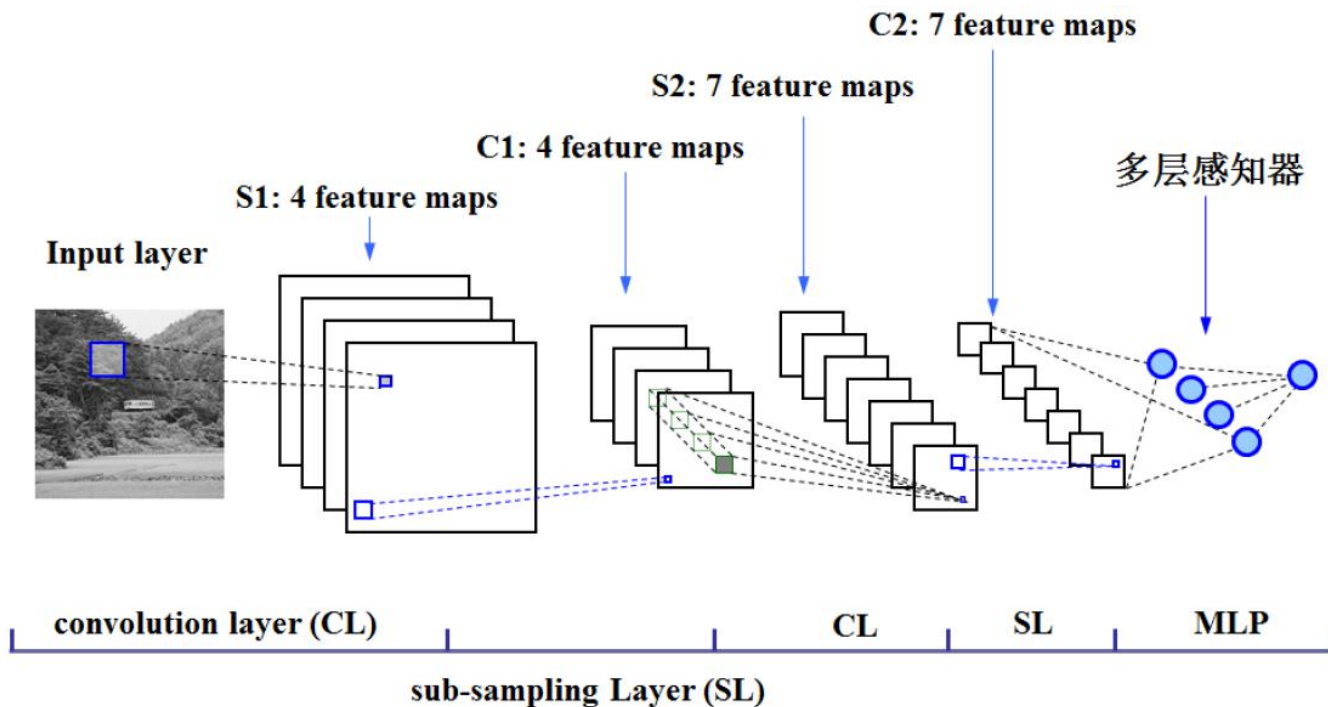
## 局部感受野（局部连接）

- **图像的空间相关性**：对图像而言，局部邻域内的像素联系较紧密，距离较远的像素相关性较弱。
- 每个神经元没必要对全局图像进行感知，只需要对局部区域进行感知。
- 在更高层将局部信息综合起来，以获得全局信息。
- 极大地降低了网络参数的数目。
- 增加网络层数能够保证节点具有更大范围的实际感受野（相对于输入图像而言）。



# 网络特点

## 局部感受野（局部连接）

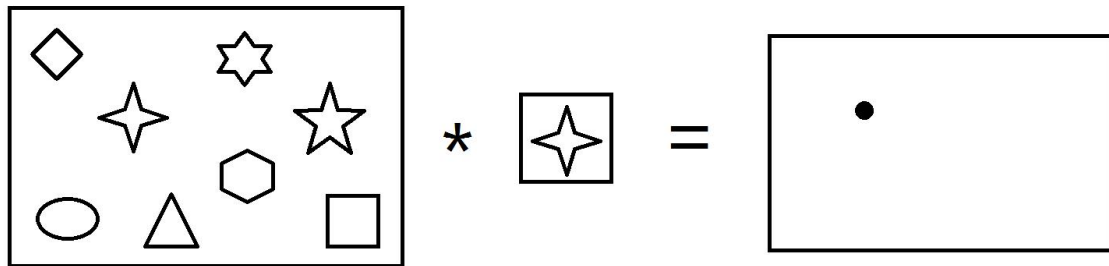


思考：实际感受野如何计算？

## 权值共享（卷积）

- 从图像任意局部区域内连接到同一类型的隐含结点，其权重保持不变。
- 实现：卷积（卷积核在图像或特征图上滑动）。
- 每个卷积核可以看成学习一种特征的滤波器。
- 也是一种降低网络参数数目的手段。

**思考：【面试真题】卷积层和全连接层的参数数目及计算负担的对比。**  
**（网络配置信息见下页）**



包含多种模式的合成图像 卷积核

响应结果

每种卷积核负责寻找图像中的某种特定模式

# 网络特点

## 池化 (pooling)

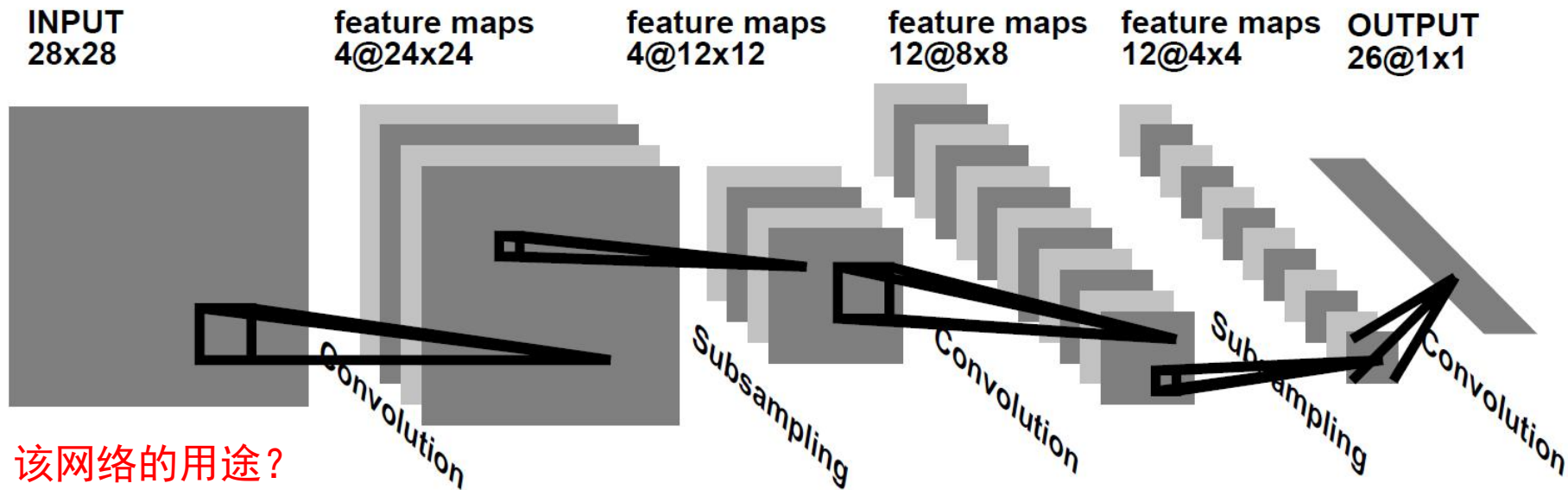
- 降低特征图的尺寸，突出更强烈的响应。
- 使卷积神经网络具有**平移不变性**和**畸变不变性**。
- 每次进行下采样时，特征图数目相应增加  
=> **牺牲空间细节信息，获得更多的语义信息。**
- 增加高层特征图中节点的实际感受野。
- 也是一种降低网络参数数目的手段。

name	kernel size	output size
input	-	$224 \times 224 \times 3$
conv1-1	$3 \times 3$	$224 \times 224 \times 64$
conv1-2	$3 \times 3$	$224 \times 224 \times 64$
pool1	$2 \times 2$	$112 \times 112 \times 64$
conv2-1	$3 \times 3$	$112 \times 112 \times 128$
conv2-2	$3 \times 3$	$112 \times 112 \times 128$
pool2	$2 \times 2$	$56 \times 56 \times 128$
conv3-1	$3 \times 3$	$56 \times 56 \times 256$
conv3-2	$3 \times 3$	$56 \times 56 \times 256$
conv3-3	$3 \times 3$	$56 \times 56 \times 256$
pool3	$2 \times 2$	$28 \times 28 \times 256$
conv4-1	$3 \times 3$	$28 \times 28 \times 512$
conv4-2	$3 \times 3$	$28 \times 28 \times 512$
conv4-3	$3 \times 3$	$28 \times 28 \times 512$
pool4	$2 \times 2$	$14 \times 14 \times 512$
conv5-1	$3 \times 3$	$14 \times 14 \times 512$
conv5-2	$3 \times 3$	$14 \times 14 \times 512$
conv5-3	$3 \times 3$	$14 \times 14 \times 512$
pool5	$2 \times 2$	$7 \times 7 \times 512$
fc6	$7 \times 7$	$1 \times 1 \times 4096$
fc7	$1 \times 1$	$1 \times 1 \times 4096$

# 网络特点

## 池化（pooling）与参数数目

思考：【面试真题】如果去掉池化层，将会带来什么后果？



## 池化（pooling）与平移不变性和畸变不变性

- 在图像识别和图像分类任务中，当图像中的某种特征被检测出（即该特征对卷积核具有较高响应），其具体位置便变得不再重要，只需要大致保持与其他特征的相对位置即可。
- 经过池化之后，能够更好地进行“非极大值抑制”（即抑制其他不明显特征响应的干扰），更加凸显该特征。
- 降低特征图的尺寸后，网络对平移和畸变的敏感性也相应降低。

思考：池化操作带来的平移不变性 vs. 卷积操作中卷积核滑动的“平移不变性”。

## 池化（pooling）与平移不变性和畸变不变性

- 有利于应对实际问题中各种不同类型、不同质量的图像。
- 通常还需专门对输入图像进行平移、畸变等处理来提高网络的泛化性能——**数据增广**



关于卷积神经网络的平移不变性和畸变不变性的具体介绍可以参考技术报告

Y. LeCun and Y. Bengio, **Convolutional Networks for Images, Speech and Time-Series.**

以及知乎回答<https://www.zhihu.com/question/36980971/answer/94840350>

1. Y. LeCun et al., Backpropagation applied to handwritten zip code recognition. Neural Computation 1989.
2. Y. LeCun et al., Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE 1998.
3. A. Krizhevsk et al., ImageNet Classification with Deep Convolutional Neural Networks. NIPS 2012.
4. J. Long et al., Fully convolutional networks for semantic segmentation. CVPR 2015.
5. Y. LeCun and Y. Bengio, Convolutional Networks for Images, Speech and Time-Series. Technical report.
6. Y. Bengio et al, Deep learning. <https://github.com/HFTrader/DeepLearningBook> .



# 课后作业

1. 列举两个尺寸不同的矩阵（如 $5 \times 5$ 和 $3 \times 3$ ），手算二者的卷积。具体计算结果可以利用Matlab的conv2d函数来验证。
2. 给出P19中 $Y^n$ 的其他三个元素的计算公式。
3. 总结从R-CNN到Fast R-CNN以及Faster R-CNN的关键改进。感兴趣的话可以进一步总结到Mask R-CNN及Mask<sup>X</sup> R-CNN的关键改进。
4. 对比DeconvNet（P39）和SegNet（P43）的主要区别。
5. P51的模型中，C2层的一个节点对应到输入图像的感受野是多大？

# 个人建议-深度学习如何入门?

目前大多数深度学习代码都是公开的，配好环境之后，按照readme.txt的指示直接就能运行，所以跑代码本身不是问题。因此，真正需要考虑的主要有两点：

1. 自己对哪个方向感兴趣(分类? 检测? 分割? )。比如你对检测感兴趣，那么R-CNN这一系列的代码肯定都得感受感受。
2. 技术的发展路线。比如，要研究DeconvNet，它是基于VGG实现的，其中很多关于VGG的内容并不会太过细致深入地介绍(默认读这篇文章的人已经充分了解VGG)，所以肯定要事先学好VGG。虽然现在网络种类特别多，但是大多数都是在已有模型基础上改造的，比如加入多尺度、BN、multi-task，以及多级子网络cascade等等，所以关键还是学好基础的几种模型，比如Alexnet, VGG, ResNet，这三个是使用频率特别高的，尤其是后两种，所以肯定得掌握。在此基础上，再看到其他基于它们的工作，就可以把关注点放在做了哪些改进上(而不是还在纠结里面说的VGG或者ResNet本身的一些细节)，并且最好能总结整理哪些改进具有哪些特定的效果，这些结论以后自己也有可能用到。





深蓝学院  
shenlanxueyuan.com

感谢各位聆听!  
Thanks for Listening

