
CNN神经网络

Steven Tang



主要课程内容

什么是特征提取？

卷积的概念理解

卷积层，池化层，全连接层

简单的卷积神经网络框架

卷积神经网络

卷积神经网络(Convolutional Neural Networks, CNN), CNN可以有效的降低**反馈神经网络**(传统神经网络)的复杂性, 常见的CNN结构有LeNet-5、AlexNet、ZFNet、VGGNet、GoogleNet、ResNet等等, 其中在LVSVC2015冠军ResNet是AlexNet的20多倍, 是VGGNet的8倍; 从这些结构来讲CNN发展的一个方向就是层次的增加, 通过这种方式可以利用增加的非线性得出目标函数的近似结构, 同时得出更好的特征表达, 但是这种方式导致了网络整体复杂性的增加, 使网络更加难以优化, 很容易过拟合。

CNN的应用主要是在**图像分类**和**物品识别**等应用场景应用比较多

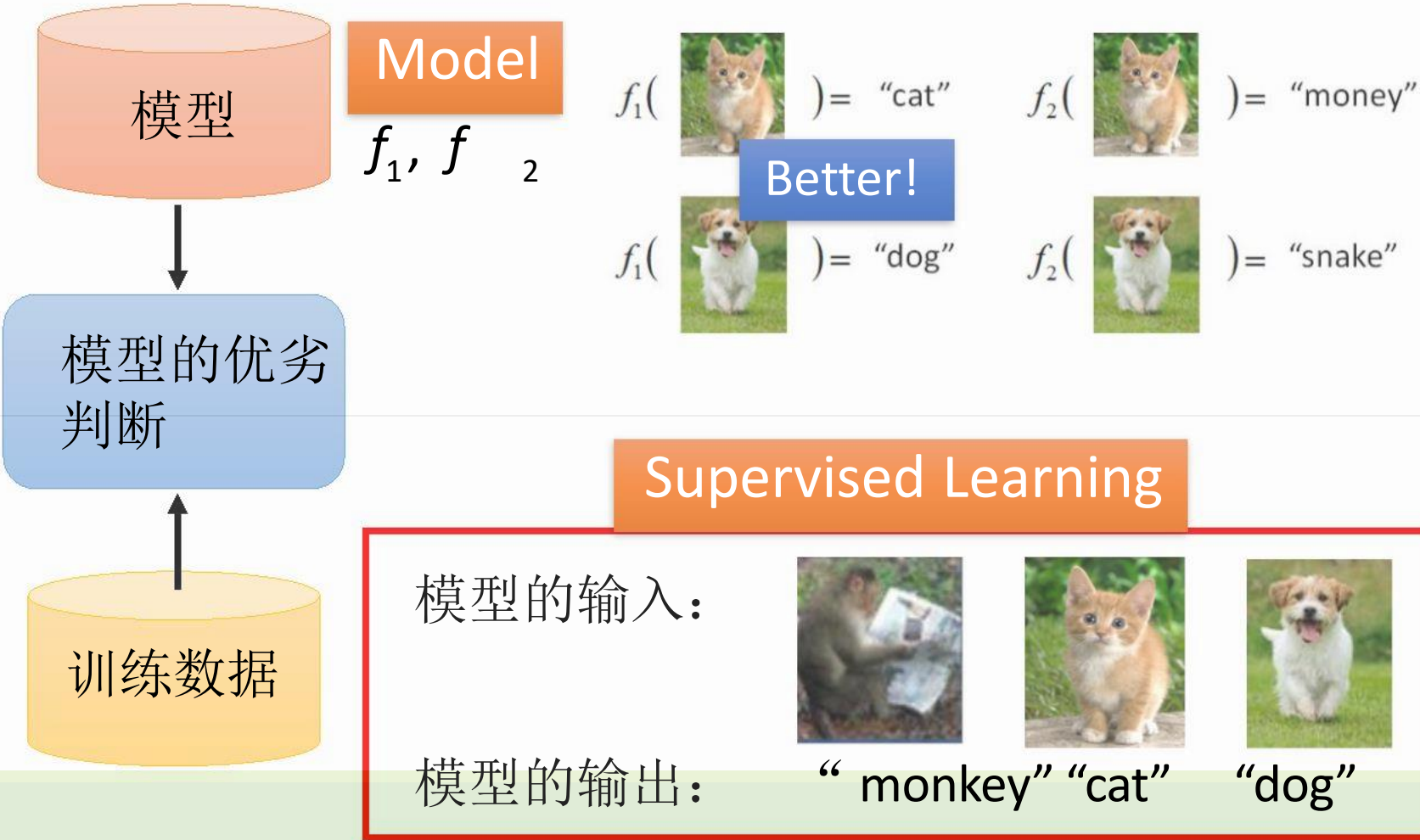
人大脑识别图片过程

人的大脑在识别图片的过程中，会由不同的皮质层处理不同方面的数据，比如：颜色、形状、光暗等，然后将不同皮质层的处理结果进行合并映射操作，得出最终的结果值，第一部分实质上是一个局部的观察结果，第二部分才是一个整体的结果合并。

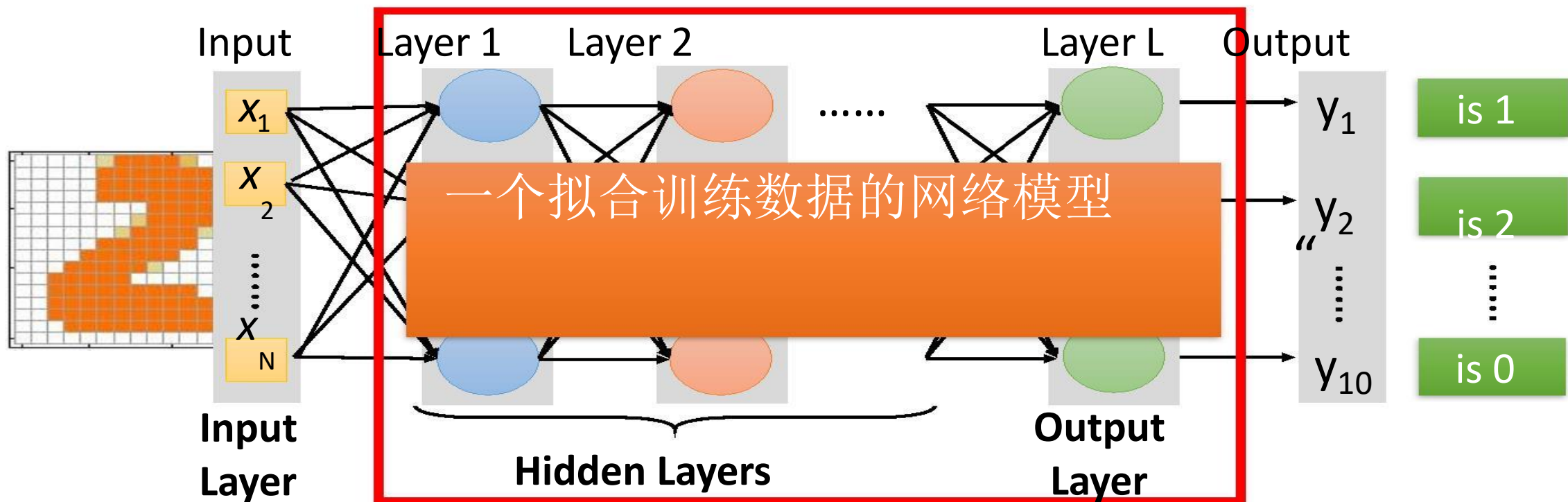


整体模型

图像识别问题 f  = “cat”

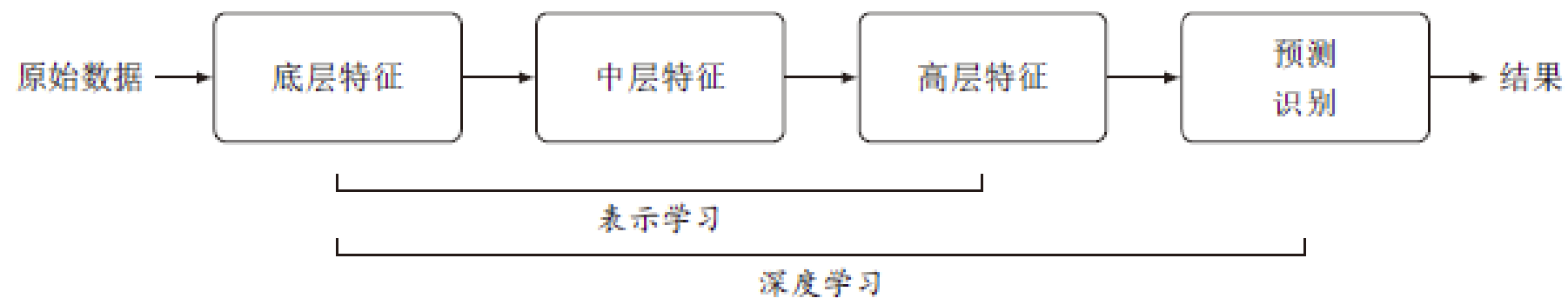


传统神经网络模型

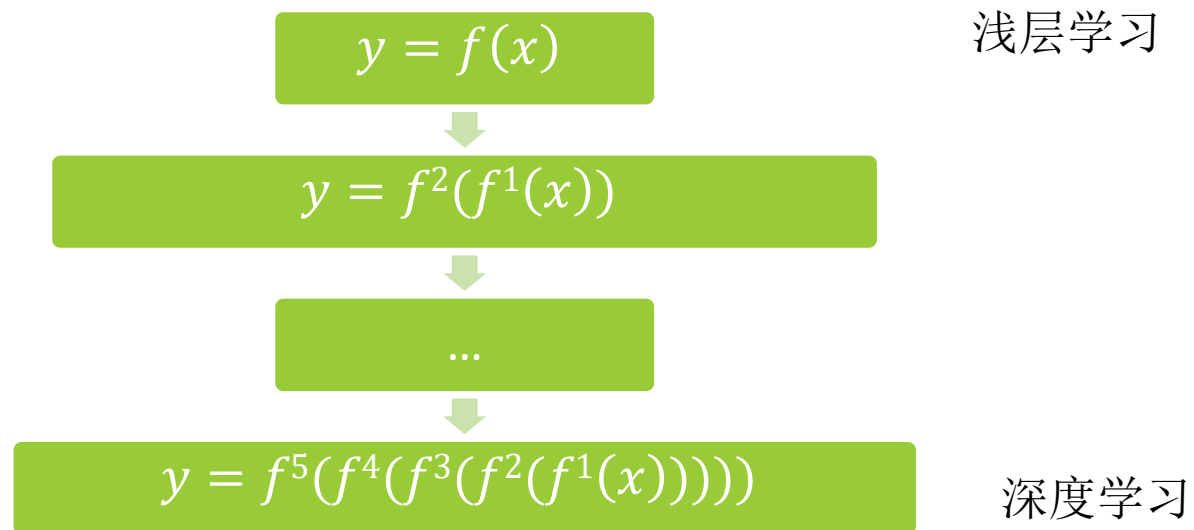


You need to decide the network structure to let a good function in your function set.

深度学习=表示学习+浅层学习



深度学习的数学描述



当 $f^1(x) = \sigma(W^1x)$ 时为神经网络！

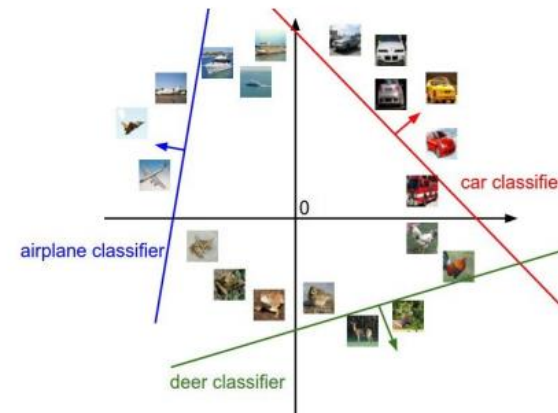
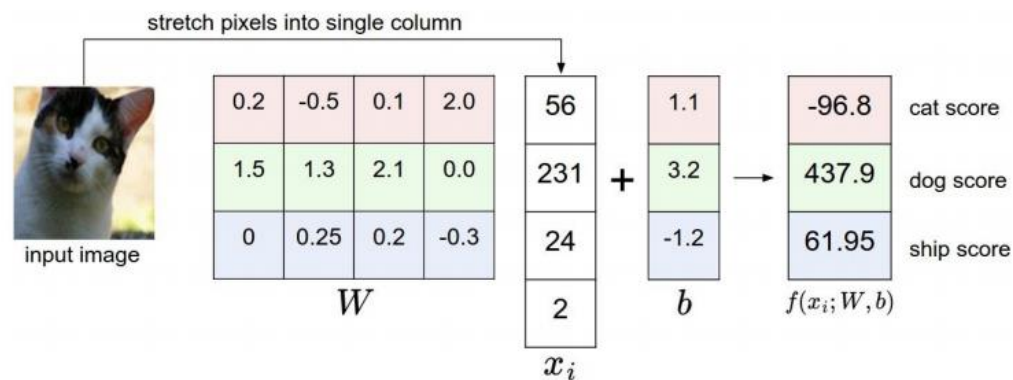
图像分类



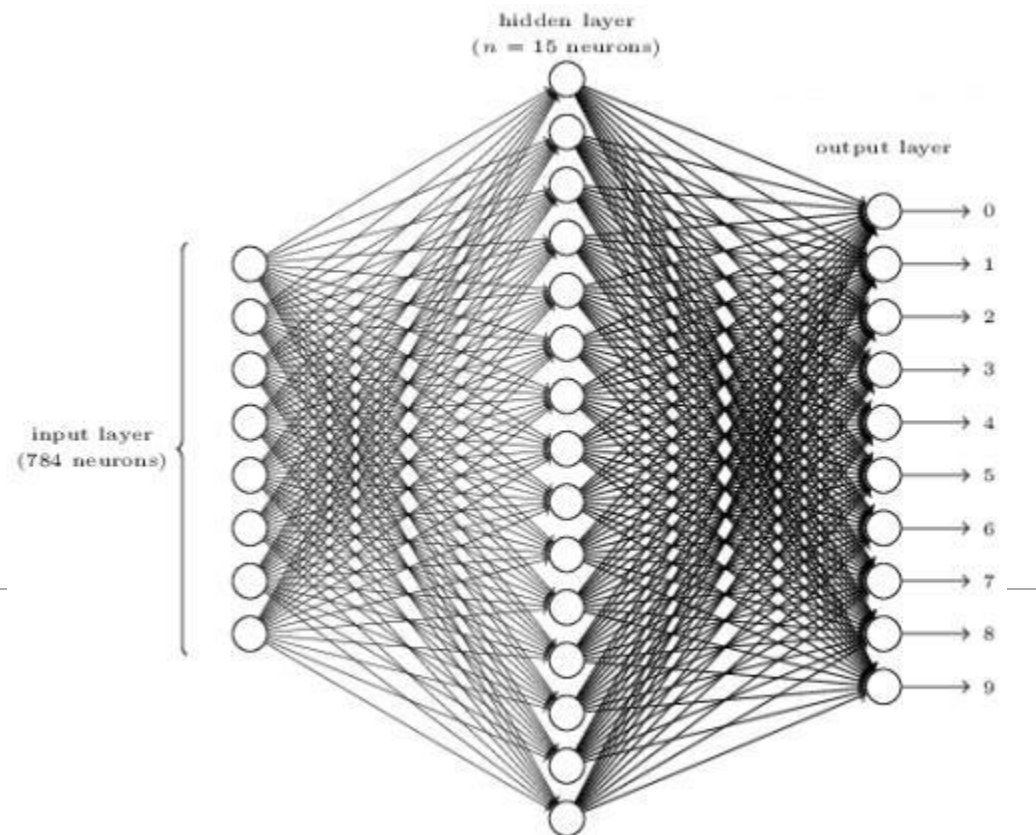
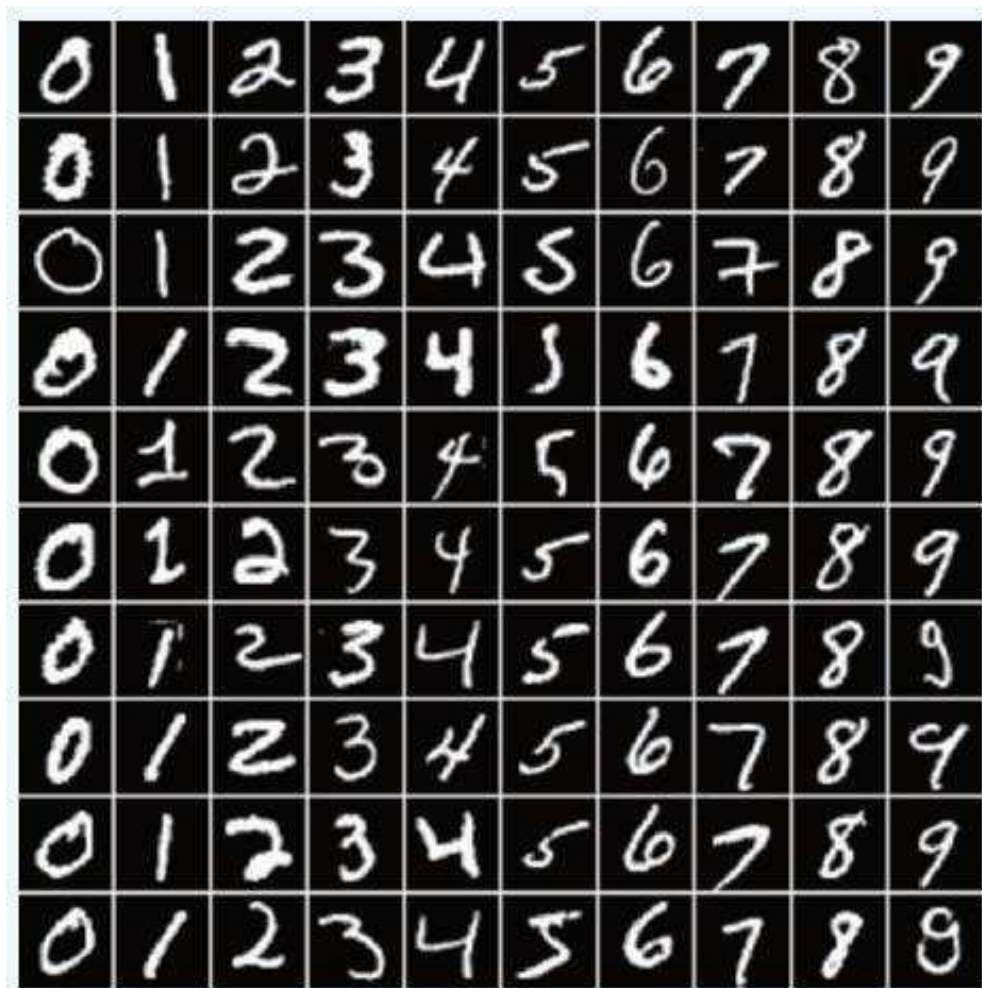
[32x32x3]
array of numbers 0...1
(3072 numbers total)

image parameters
 $f(\mathbf{x}, \mathbf{W})$

10 numbers, indicating
class scores



传统神经网络模型 (MNIST 数据集)



中间测试

$$6+9=2;$$

$$10+13=1$$

$$8+66=4$$

$$3+5=?$$

另一个测试

将下列六个汉字分为两组

1.中 2.国 3.人 4.口 5.日 6.本

卷积神经网络和图像分类

- 有很多图形特征比图像本身要小的多

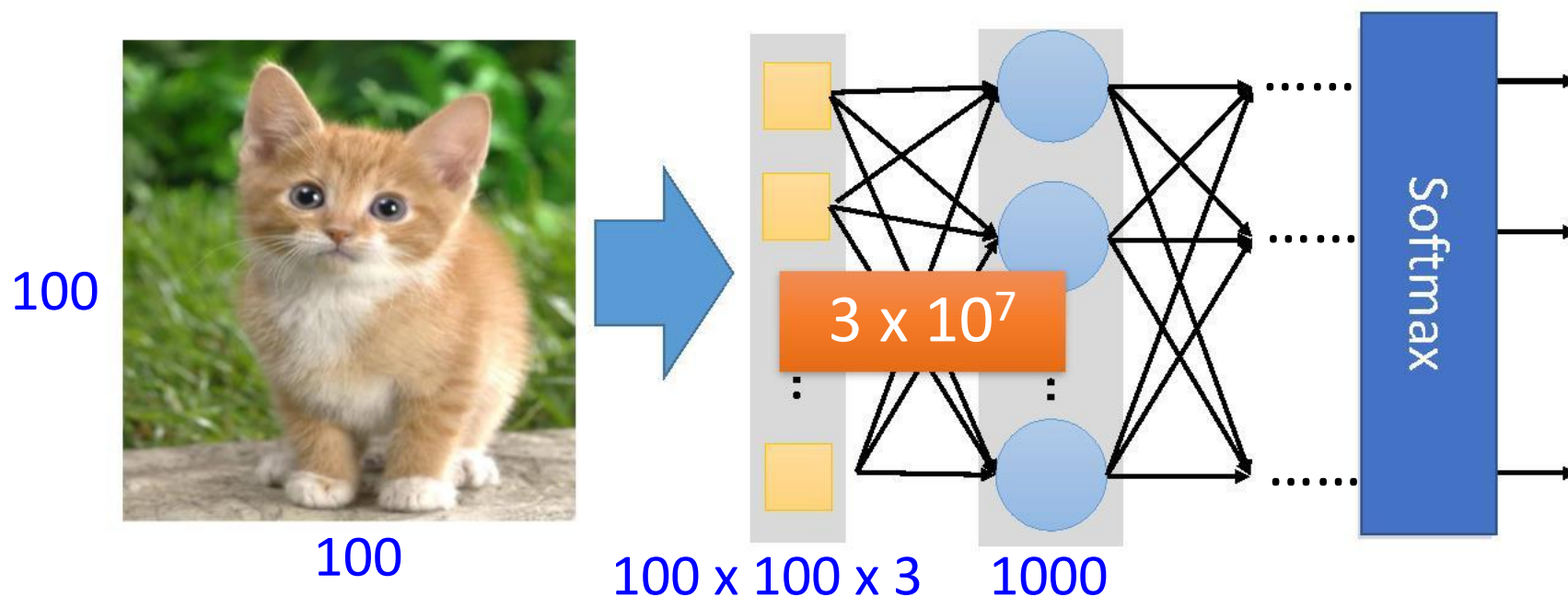
单个神经元并不需要考虑整个图像。

连接一小块区域能够减少参数



卷积神经网络和图像分类

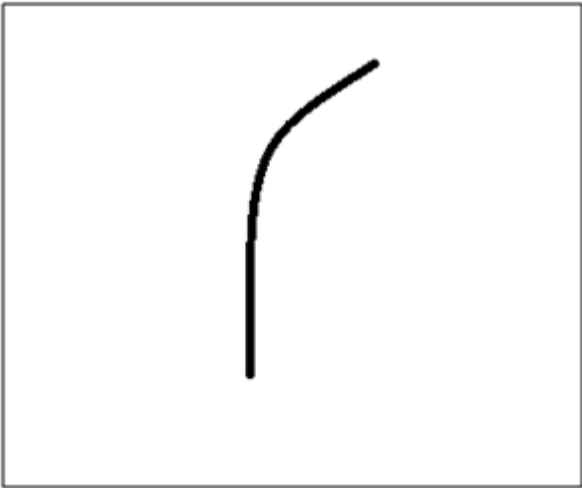
传统神经网络参数过多



直观图解深度神经网络（第一层）

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

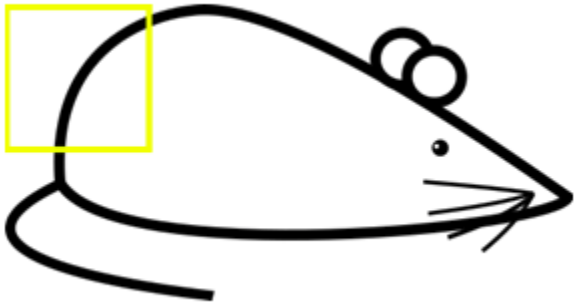
Pixel representation of filter



Visualization of a curve detector filter



Original image



Visualization of the filter on the image

直观图解深度神经网络（第一层）



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = $(50 \times 30) + (50 \times 30) + (50 \times 30) + (20 \times 30) + (50 \times 30) = 6600$ (A large number!)



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

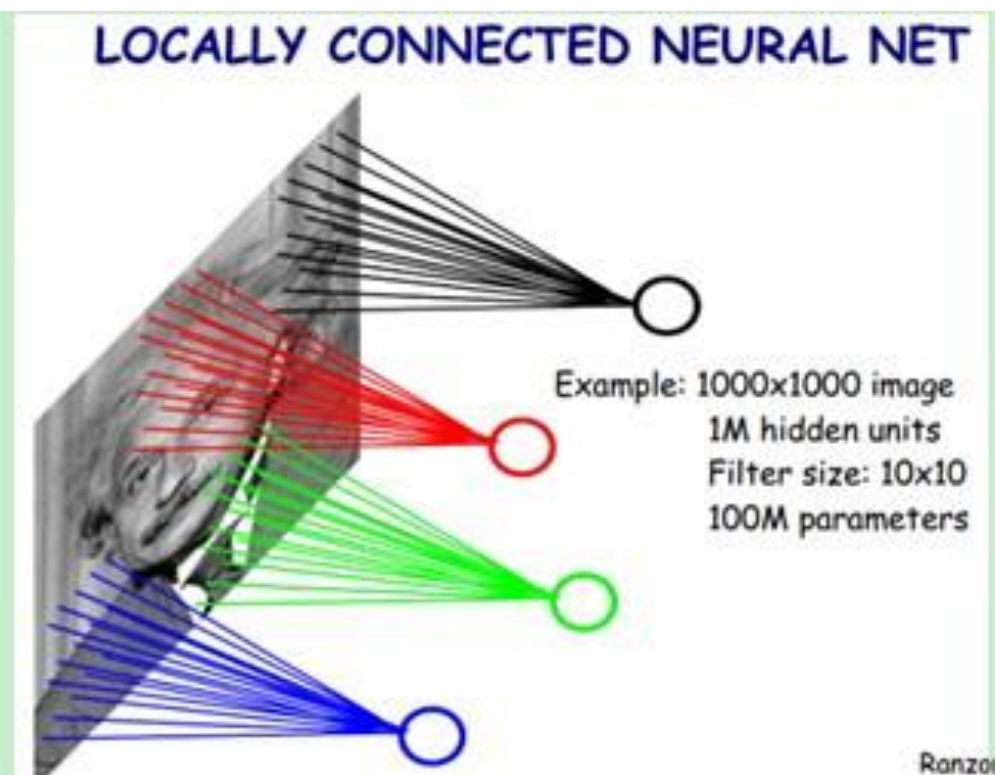
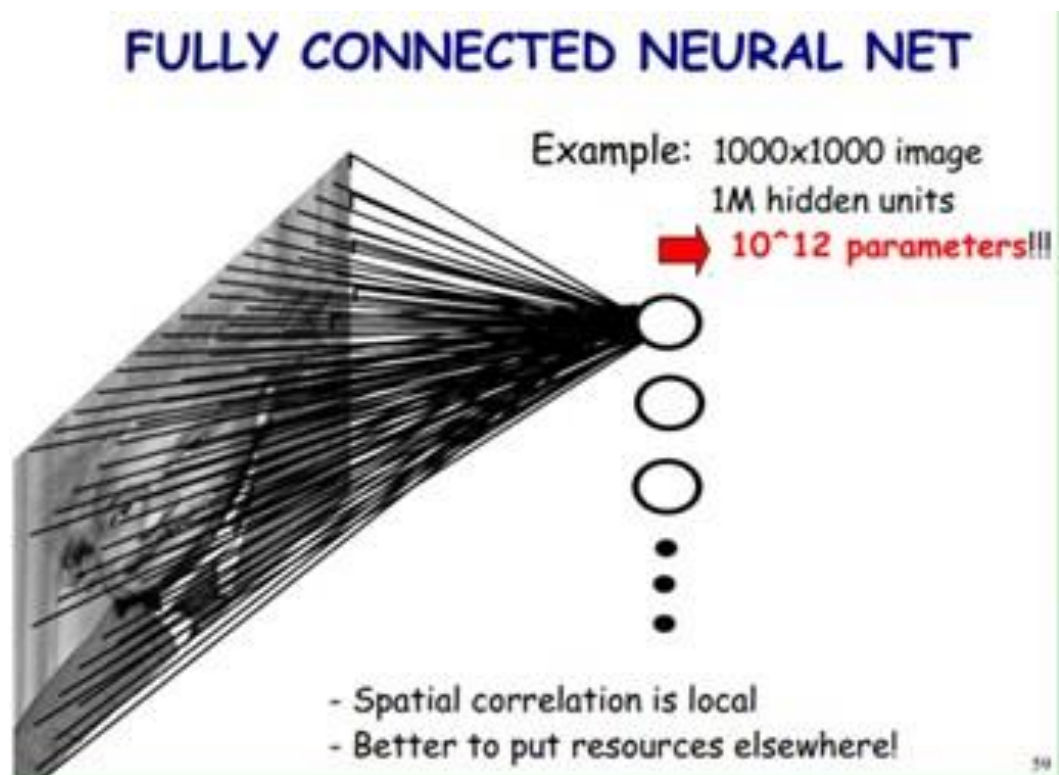
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

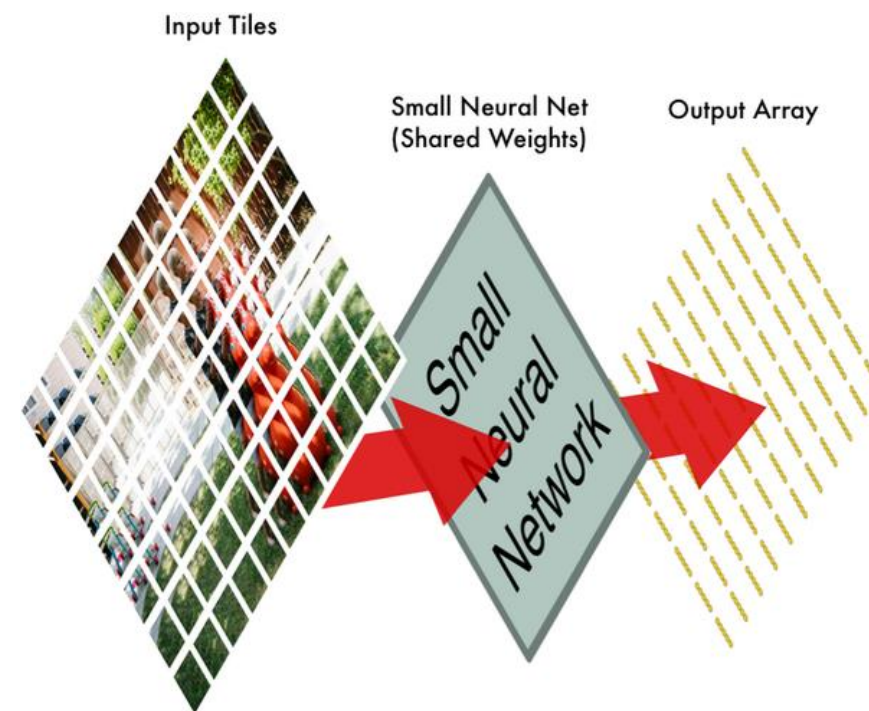
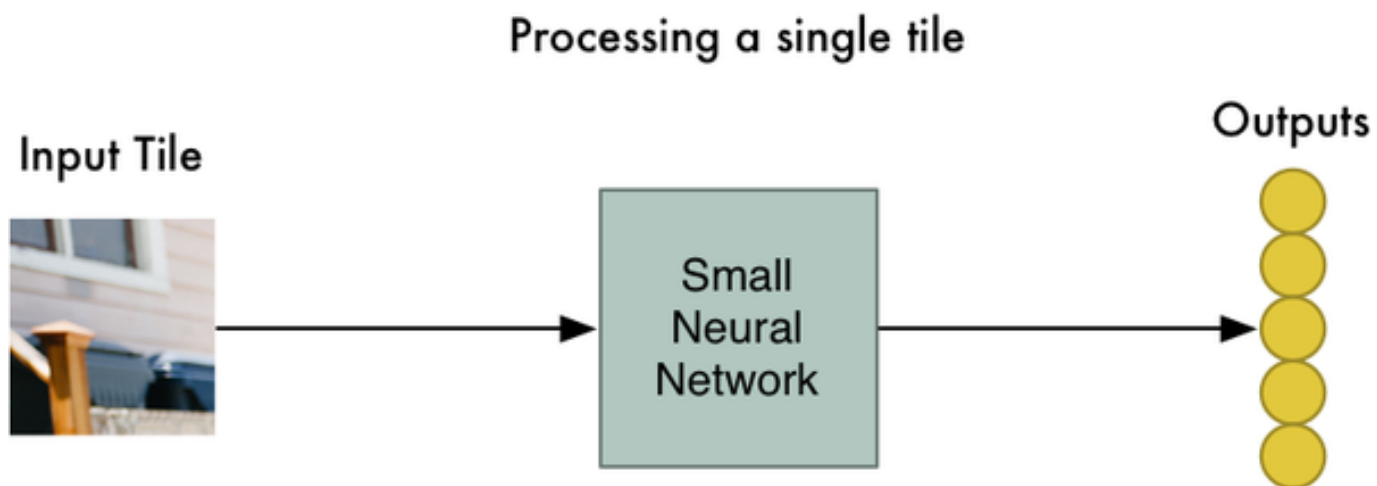
理解深度神经网络（第二层）：局部感知和权值共享



理解深度神经网络（第二层）：局部感知和权值共享



理解深度神经网络（第二层）：权值共享



卷积神经网络-CONV Layer

参数共享机制：假设每个神经元连接数据窗的权重是固定的

固定每个神经元的连接权重，可以将神经元看成一个模板；也就是每个神经元只关注一个特性

需要计算的权重个数会大大的减少

一组固定的权重和不同窗口内数据做内积：卷积

卷积神经网络

全连接网络

- 权重矩阵的参数非常多

卷积神经网络

- 生物学上感受野

卷积神经网络有三个结构上的特性：

- 局部连接
- 权重共享

卷积

卷积经常用在信号处理中，用于计算信号的延迟累积。

假设一个信号发生器每个时刻 t 产生一个信号 x_t ，其信息的衰减率为 w_k ，即在 $k-1$ 个时间步长后，信息为原来的 w_k 倍

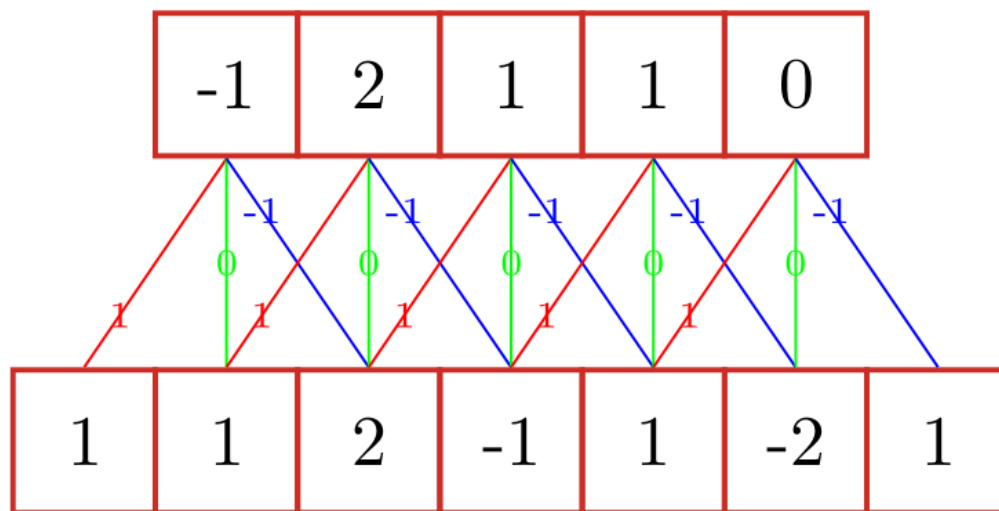
- 假设 $w_1 = 1, w_2 = 1/2, w_3 = 1/4$

时刻 t 收到的信号 y_t 为当前时刻产生的信息和以前时刻延迟信息的叠加

$$\begin{aligned}y_t &= 1 \times x_t + 1/2 \times x_{t-1} + 1/4 \times x_{t-2} \\&= w_1 \times x_t + w_2 \times x_{t-1} + w_3 \times x_{t-2} \\&= \sum_{k=1}^3 w_k \cdot x_{t-k+1}.\end{aligned}$$

滤波器 (filter) 或卷积核 (convolution kernel)

一维卷积



滤波器: $[-1, 0, 1]$

二维卷积

1	1	1	1	1
-1	0	-3	0	1
2	1	1	-1	0
0	-1	1	2	1
1	2	1	1	1

$\times -1$ $\times 0$ $\times 0$
 $\times 0$ $\times 0$ $\times 0$
 $\times 0$ $\times 0$ $\times 1$

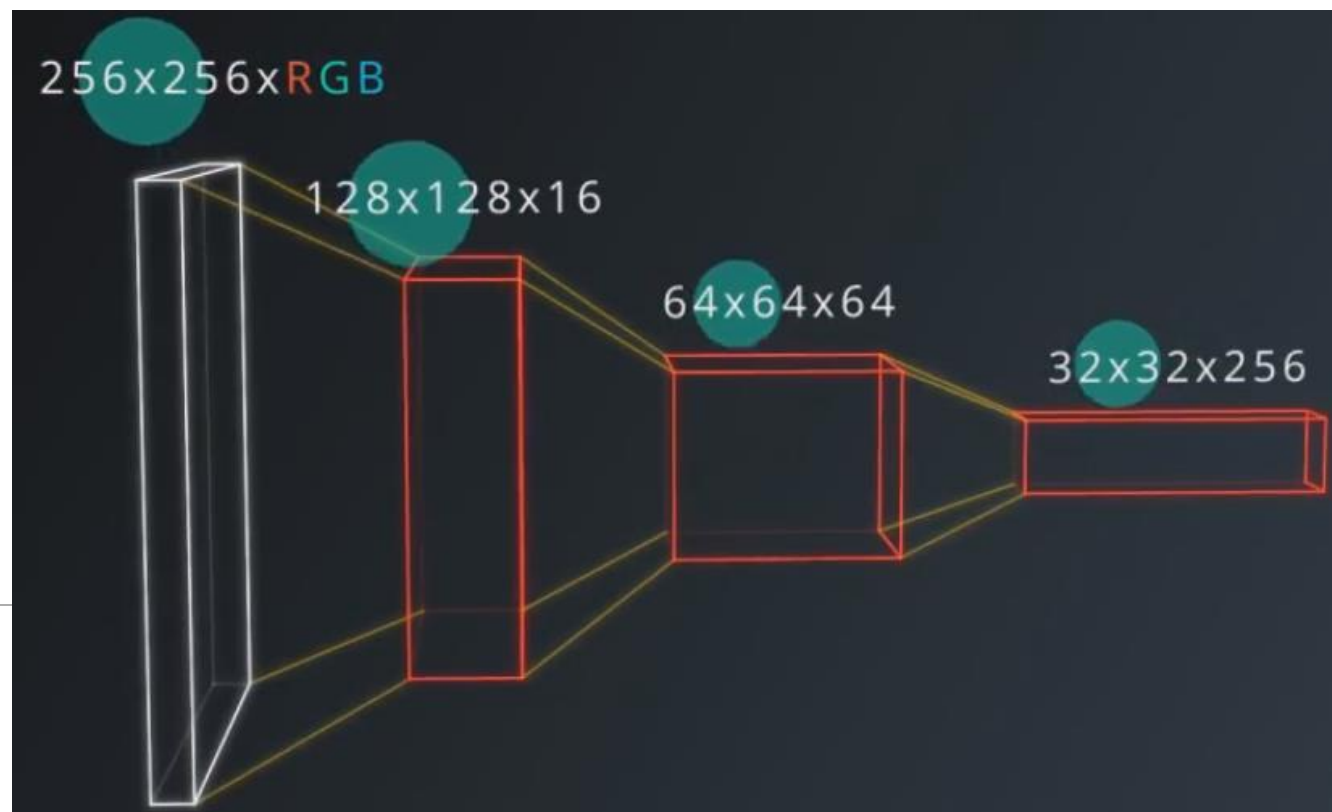
 \otimes

1	0	0
0	0	0
0	0	-1

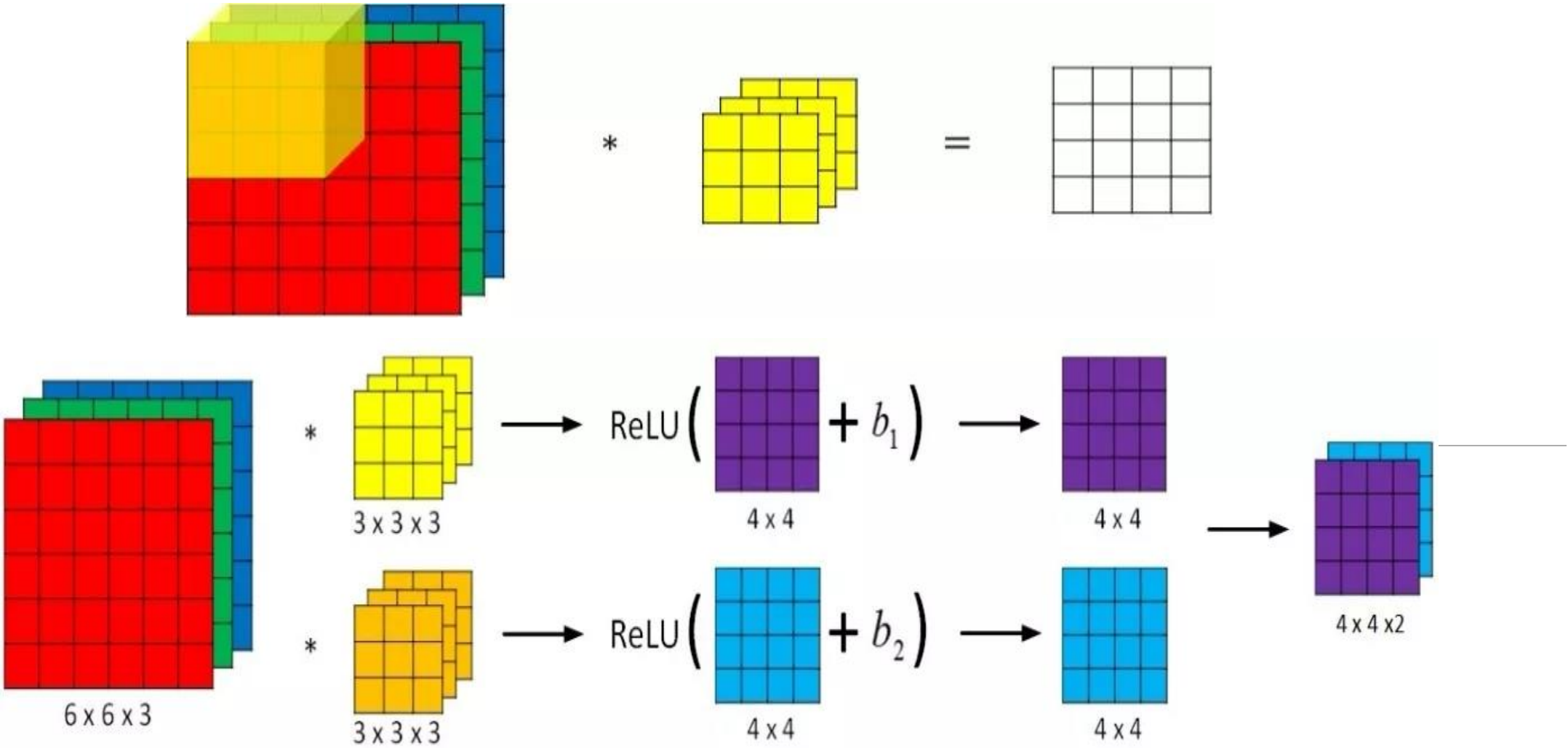
 $=$

0	-2	-1
2	2	4
-1	0	0

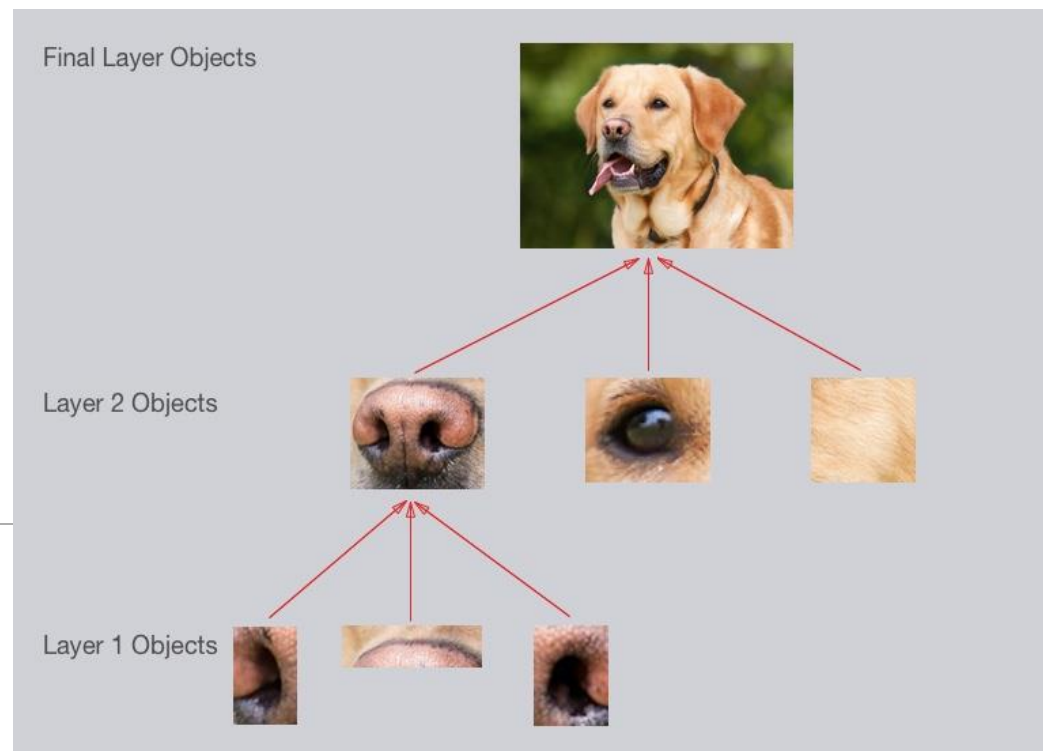
卷积网络概述



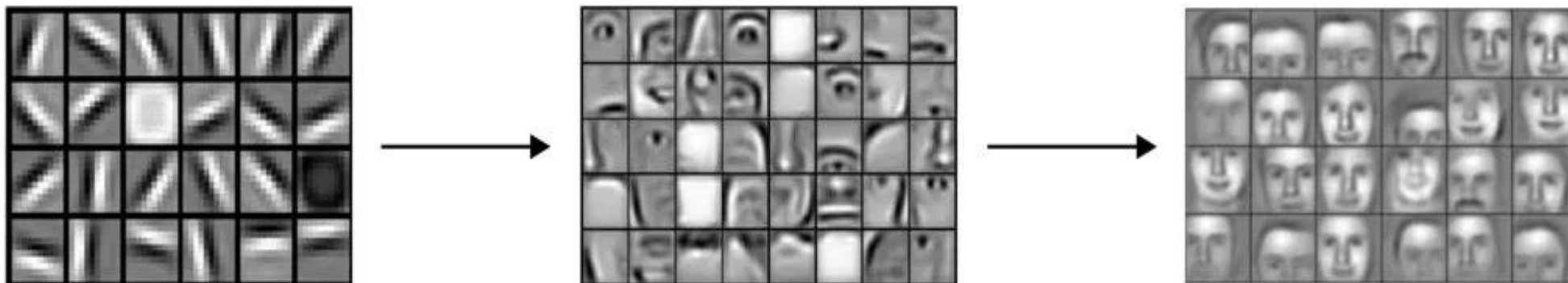
卷积网络概述



3.2 理解深度神经网络（第三层）：特征抽象



理解深度神经网络（第三层）：特征抽象



特征可视化

<http://scs.ryerson.ca/~aharley/vis/conv/flat.html>

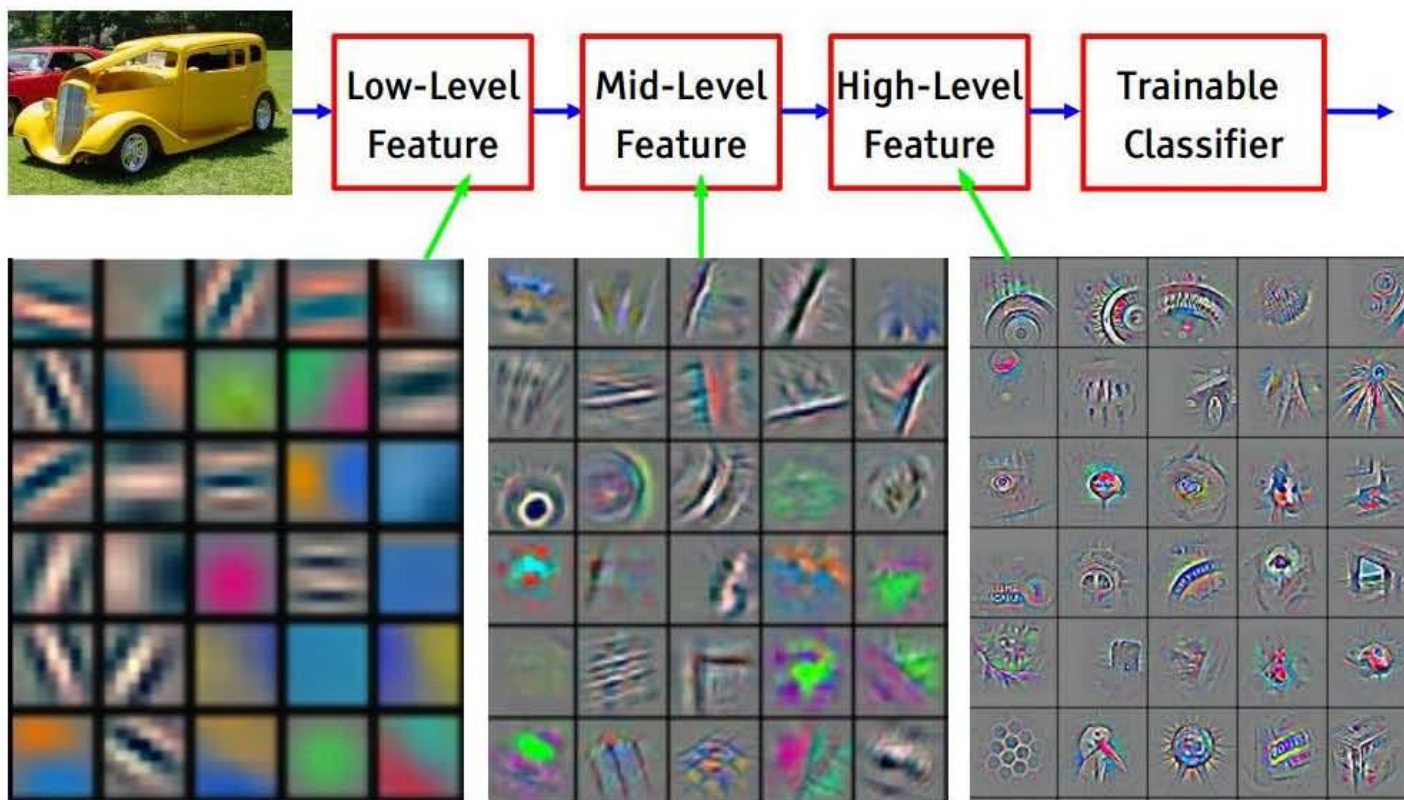
<https://distill.pub/2018/building-blocks/>

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

理解深度神经网络（第三层）：特征抽象

Preview

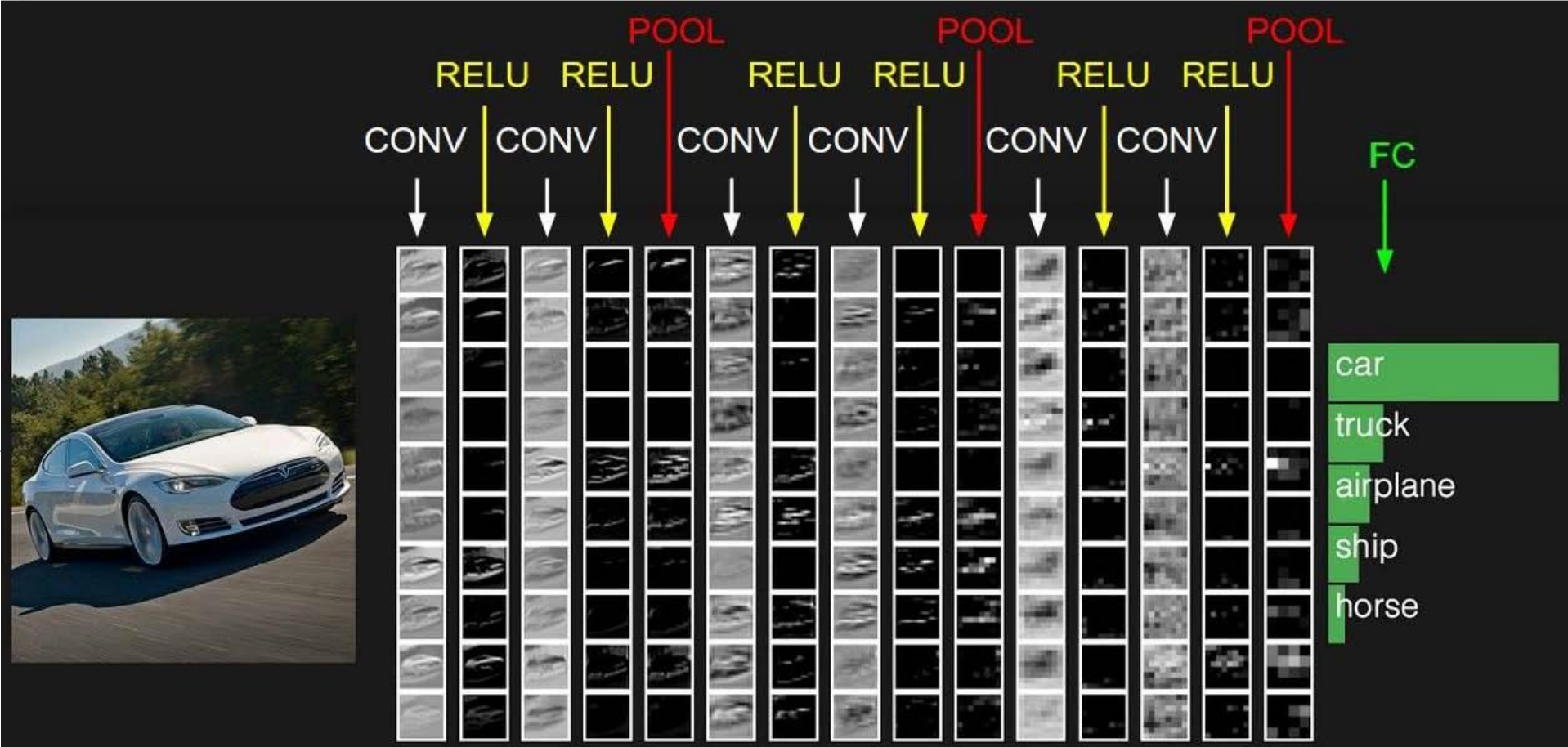
[From recent Yann LeCun slides]



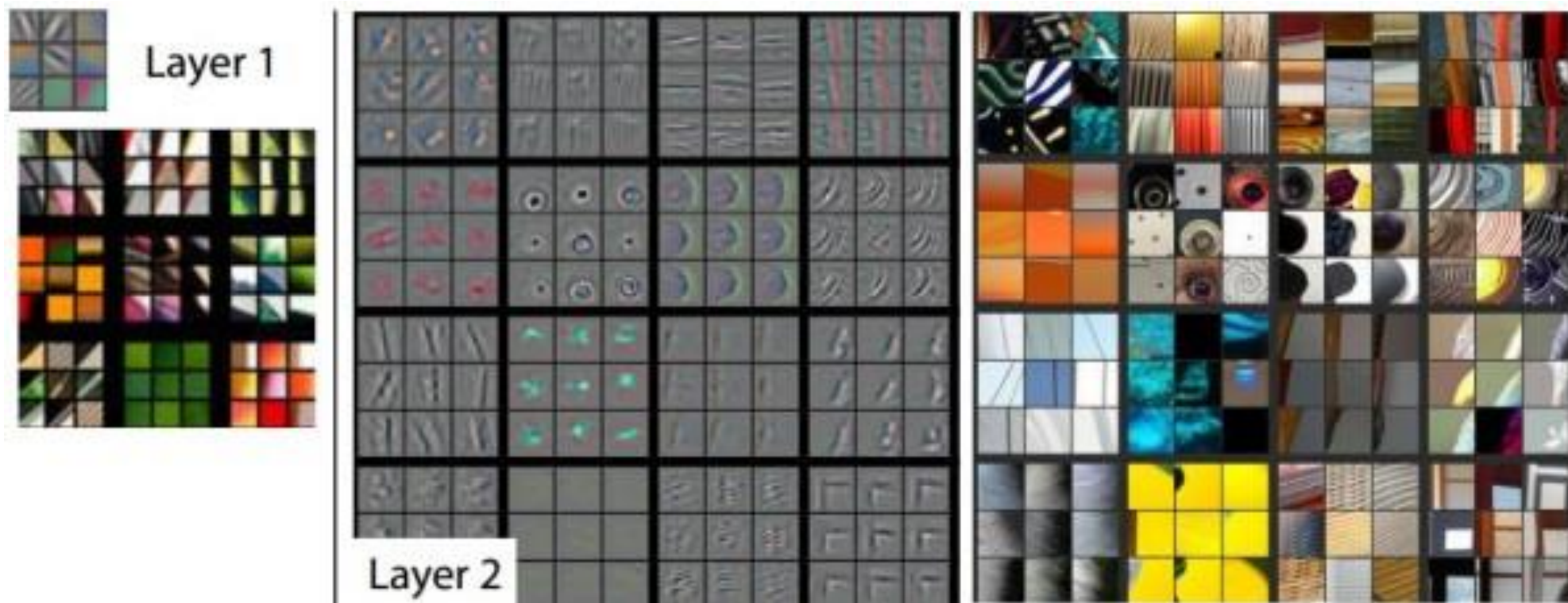
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

理解深度神经网络（第三层）：特征抽象

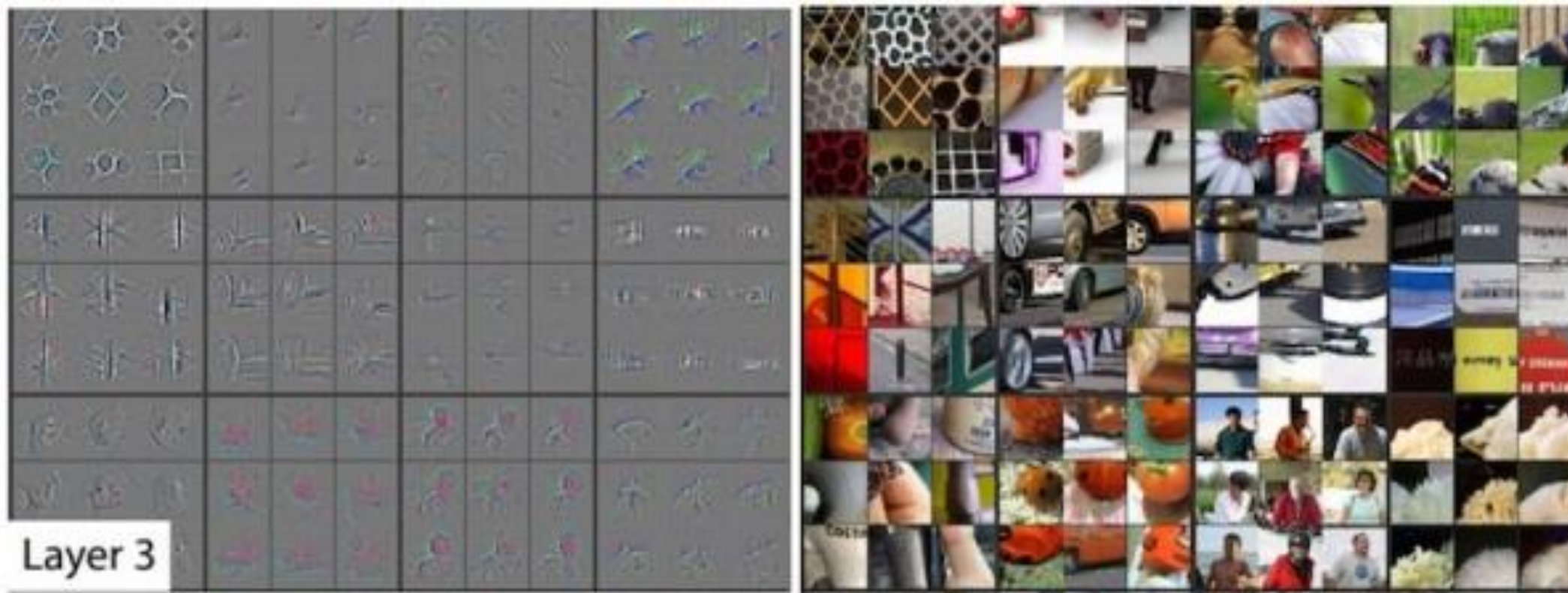
preview:



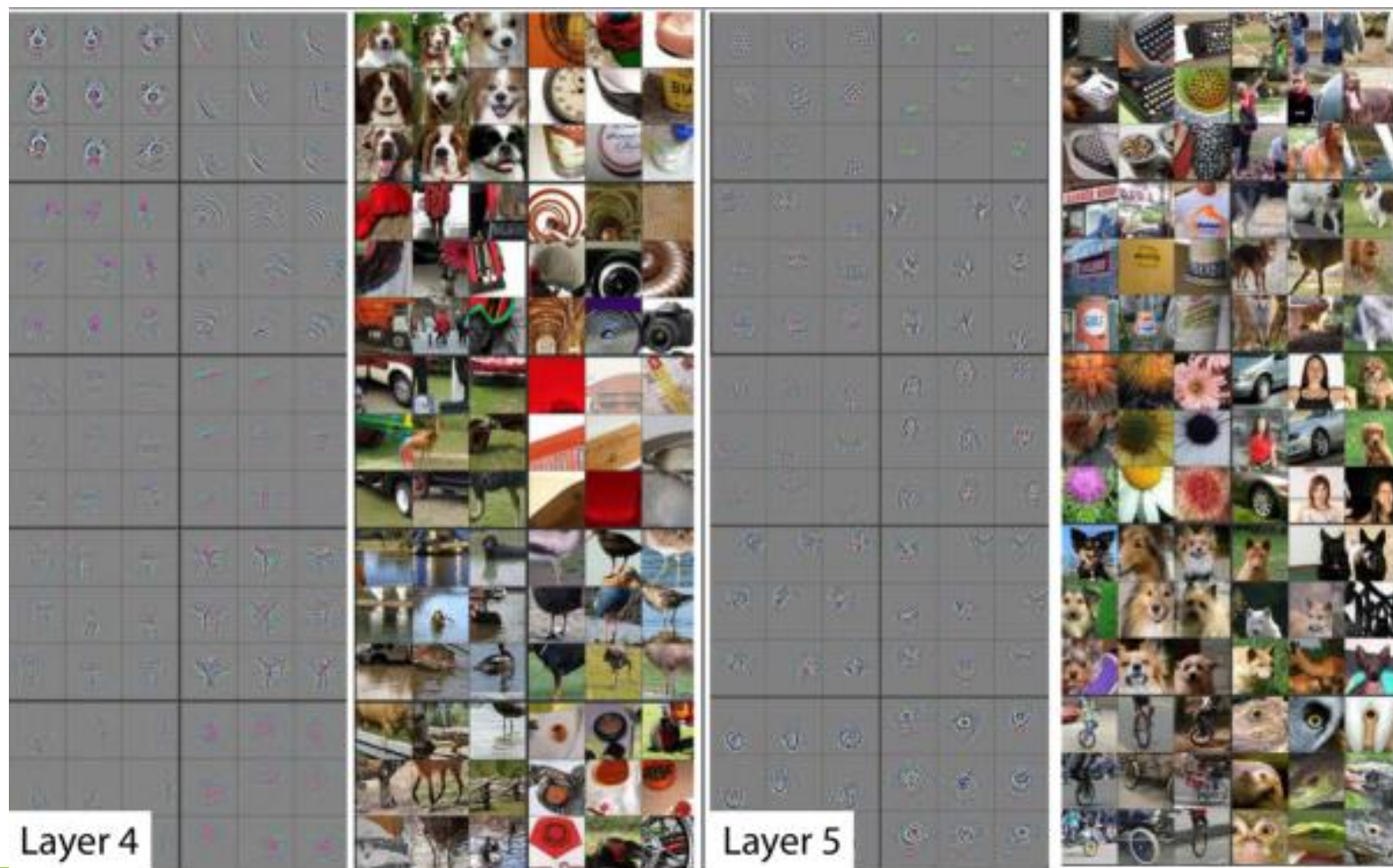
卷积神经网络可视化



卷积神经网络可视化

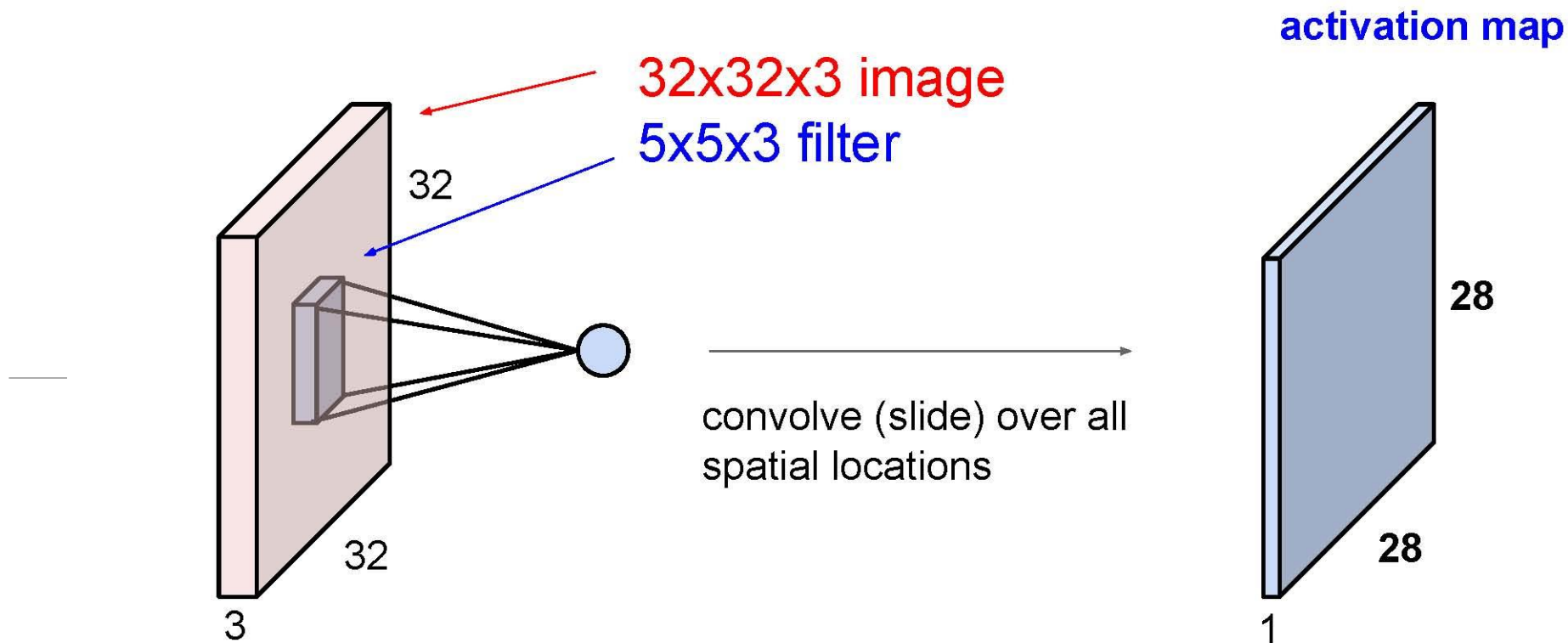


卷积神经网络可视化



卷积层（卷积核和特征图）

A closer look at spatial dimensions:



卷积层（卷积核和特征图）

INPUT IMAGE

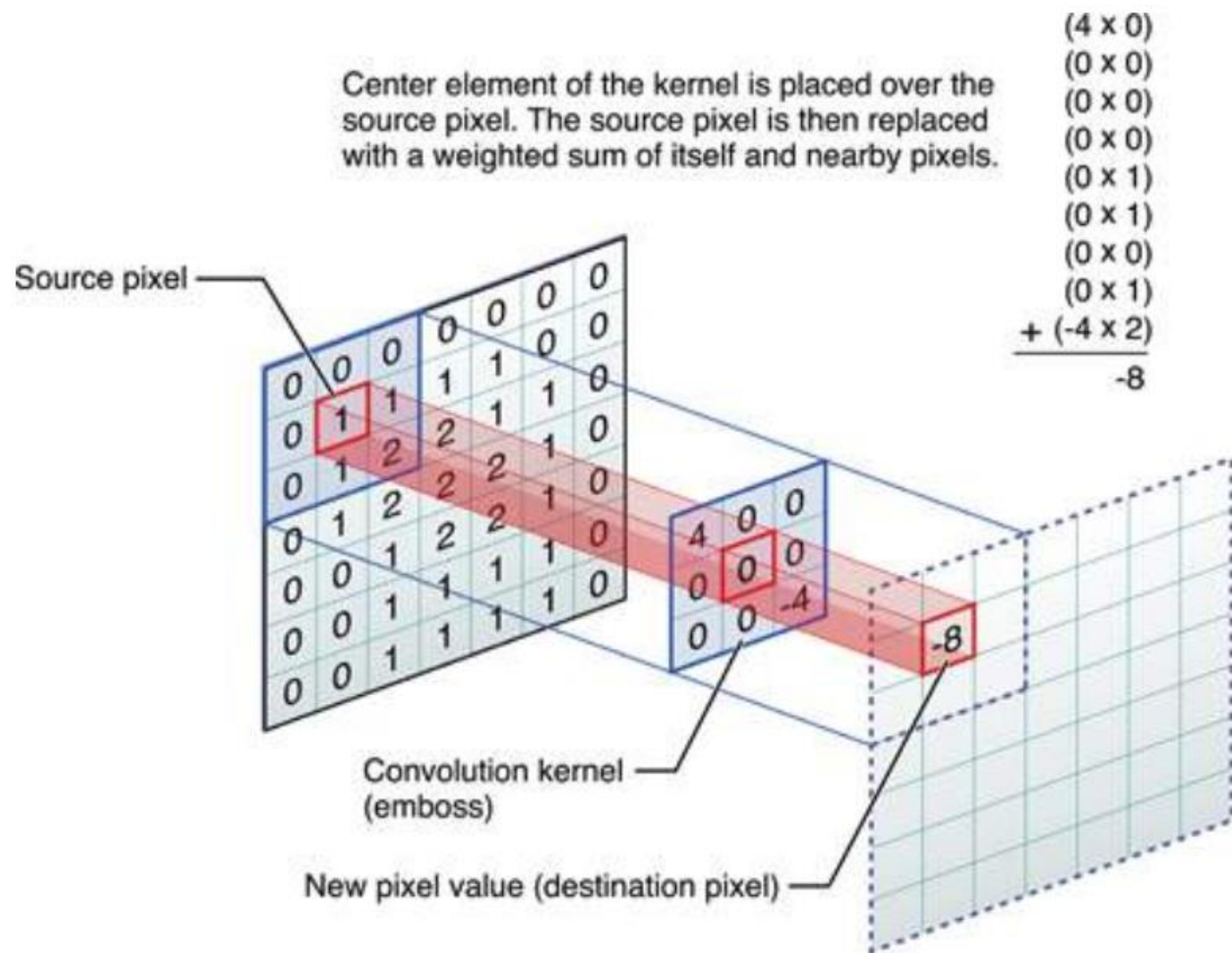
18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

WEIGHT

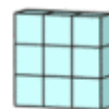
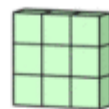
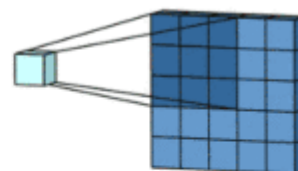
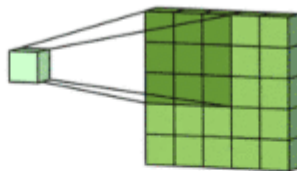
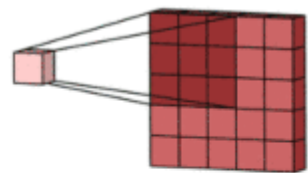
1	0	1
0	1	0
1	0	1

429

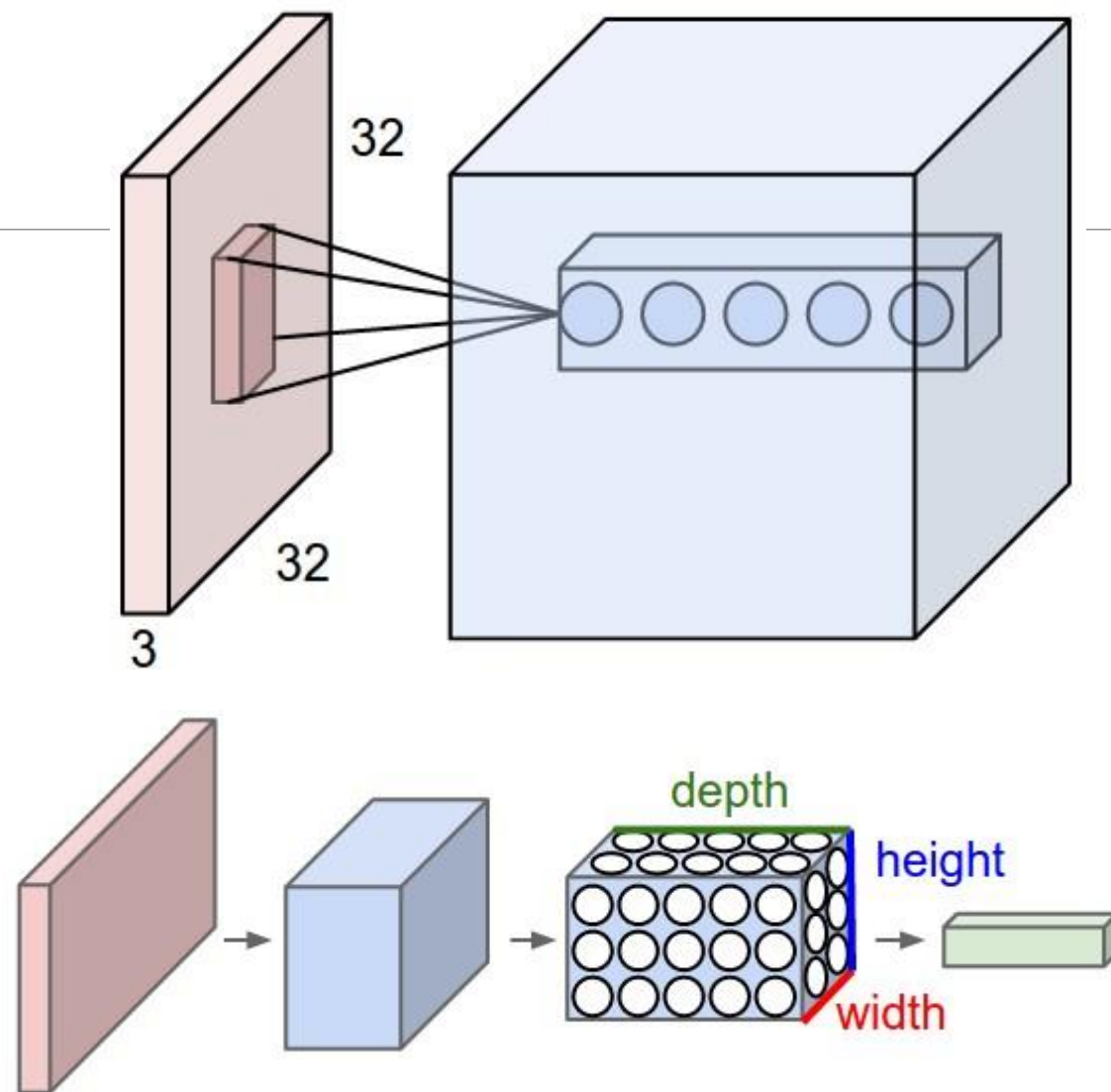
卷积的理解



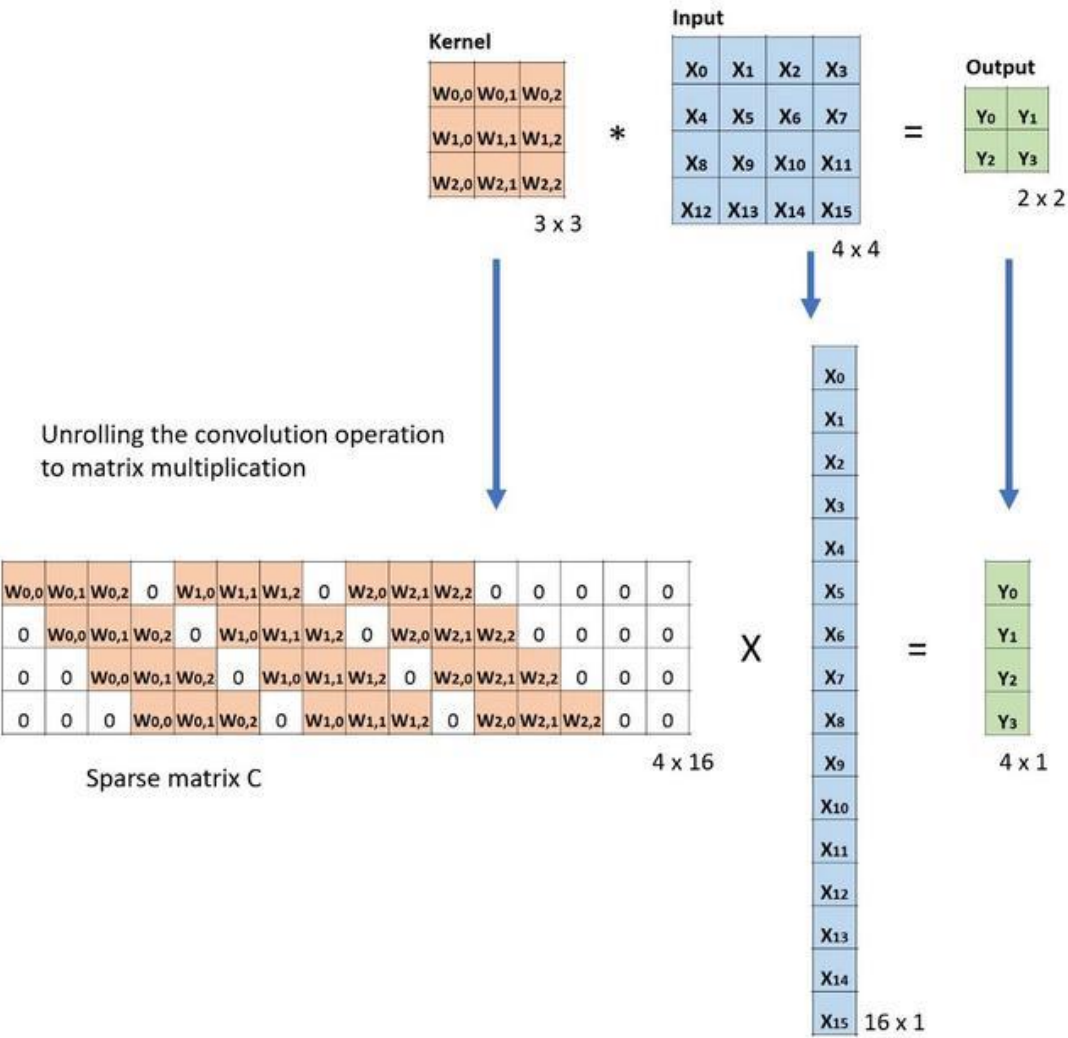
多通道卷积



卷积核的长宽高

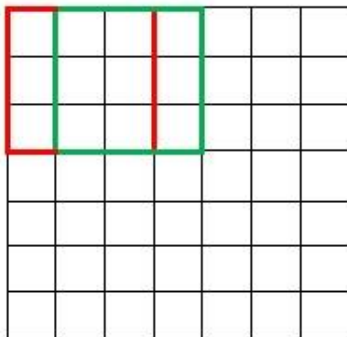


将卷积看成矩阵乘法

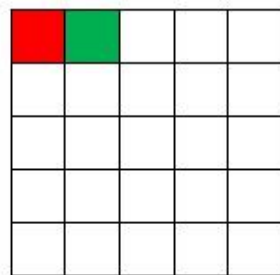


Padding: Valid, 以及stride

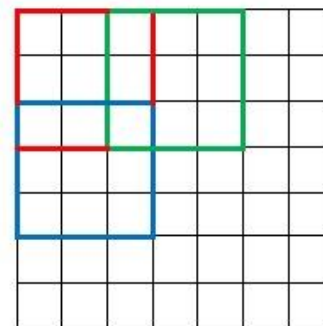
7 x 7 Input Volume



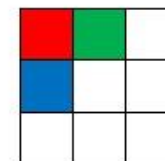
5 x 5 Output Volume



7 x 7 Input Volume

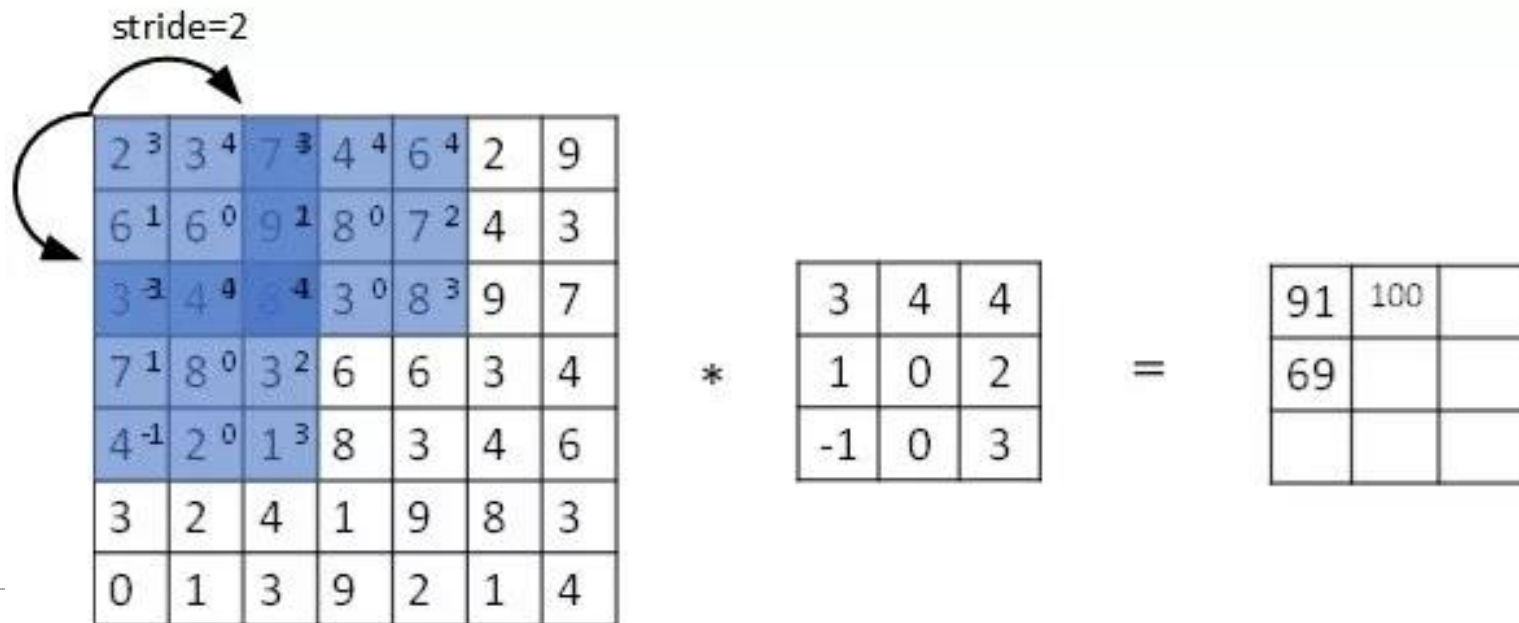


3 x 3 Output Volume



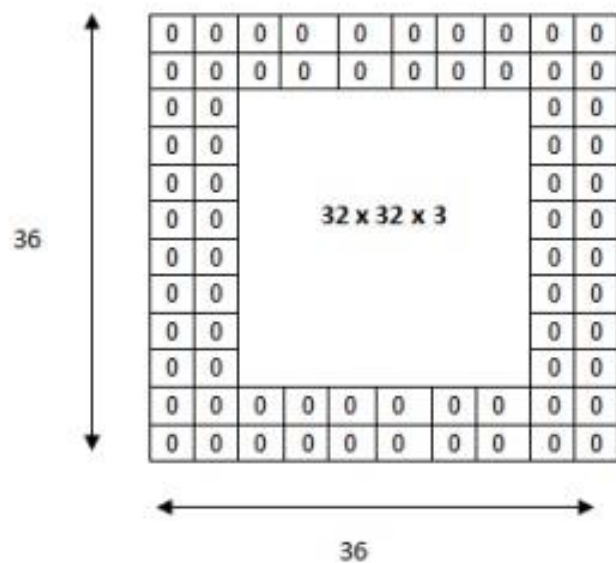
Stride: 移动滤波器平移时的像素的数量，当stride=1，输出的尺寸和输入的尺寸大体相同，stride=2时候，输出尺寸相当于输入的一半。

stride



Stride: 移动滤波器平移时的像素的数量，当stride=1，输出的尺寸和输入的尺寸大体相同，stride=2时候，输出尺寸相当于输入的一半。

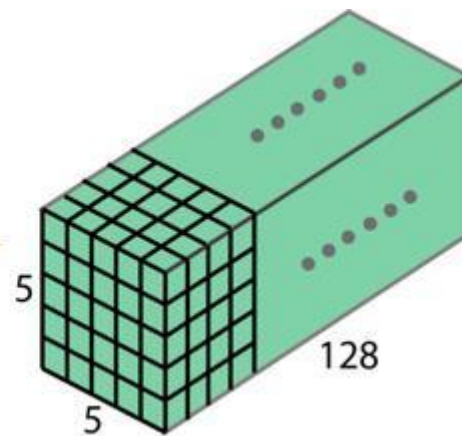
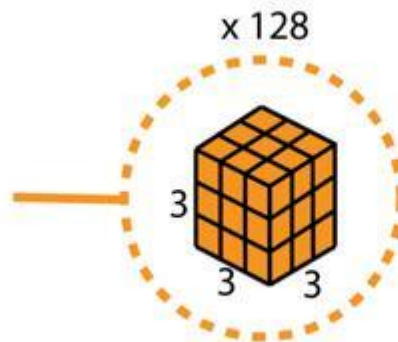
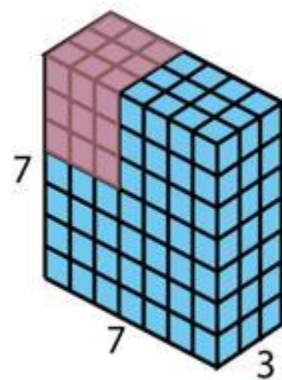
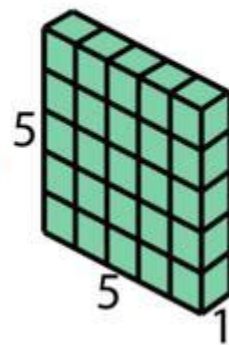
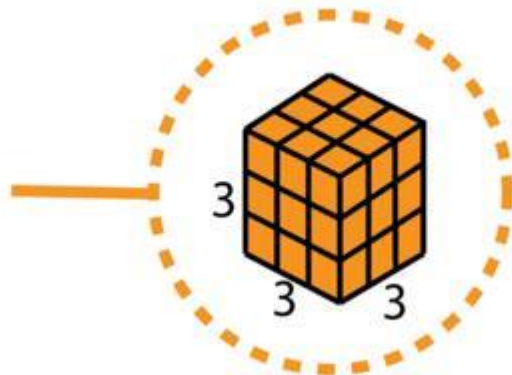
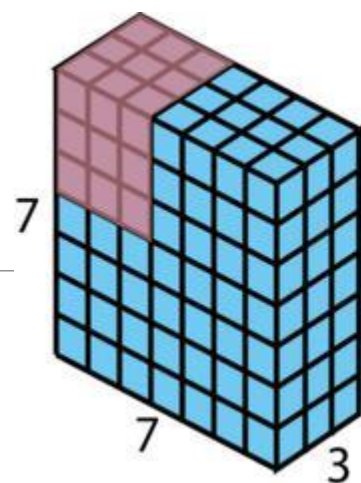
Padding: Same



The input volume is 32 x 32 x 3. If we imagine two borders of zeros around the volume, this gives us a 36 x 36 x 3 volume. Then, when we apply our conv layer with our three 5 x 5 x 3 filters and a stride of 1, then we will also get a 32 x 32 x 3 output volume.

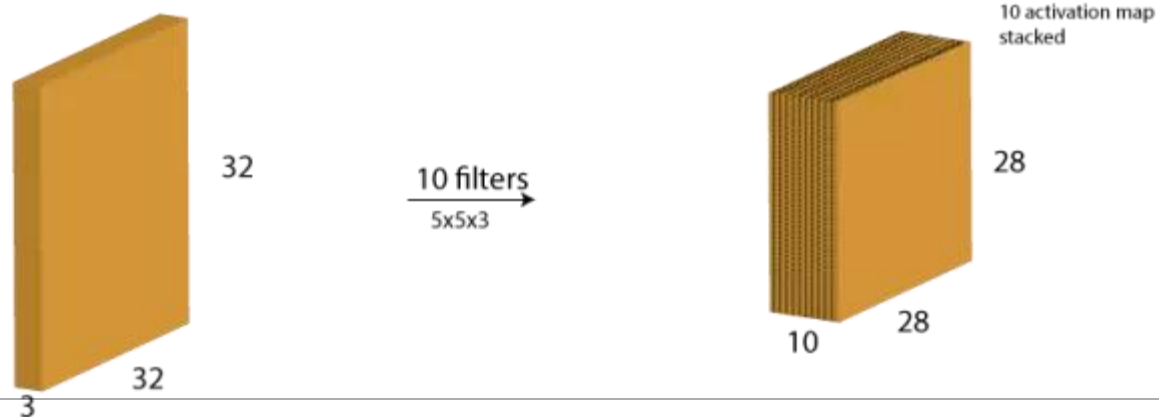
当你把 $5 \times 5 \times 3$ 的过滤器用在 $32 \times 32 \times 3$ 的输入上时，会发生什么？输出的大小会是 $28 \times 28 \times 3$ 。

如何维持输出的维度仍然为： $32 \times 32 \times 3$ ？

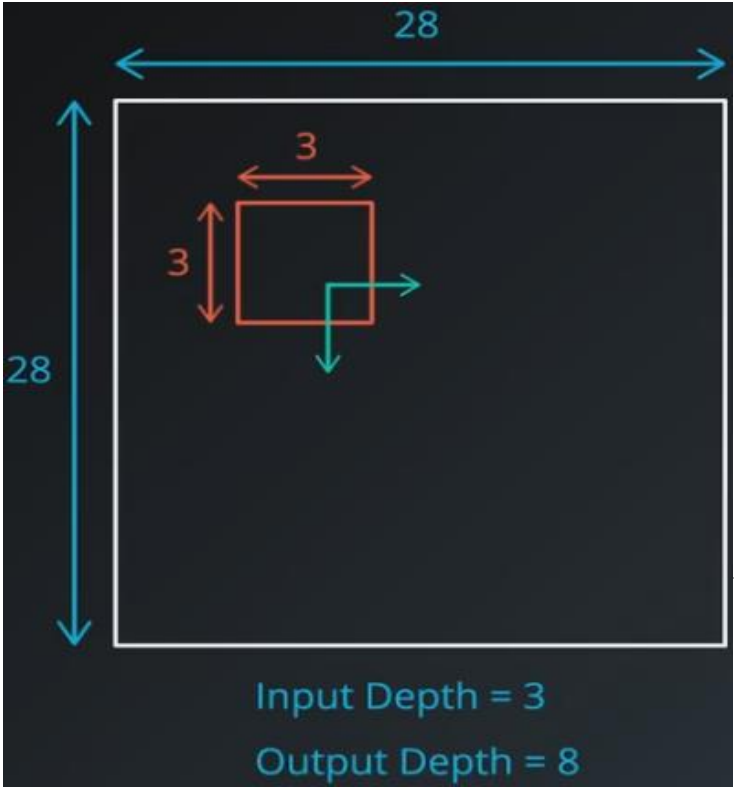


卷积层（卷积核和特征图）

每一个过滤器的输出被堆叠在一起，形成卷积图像的纵深维度。假设我们有一个 $32*32*3$ 的输入。我们使用 $5*5*3$ ，带有 valid padding 的 10 个过滤器。输出的维度将会是 $28*28*10$

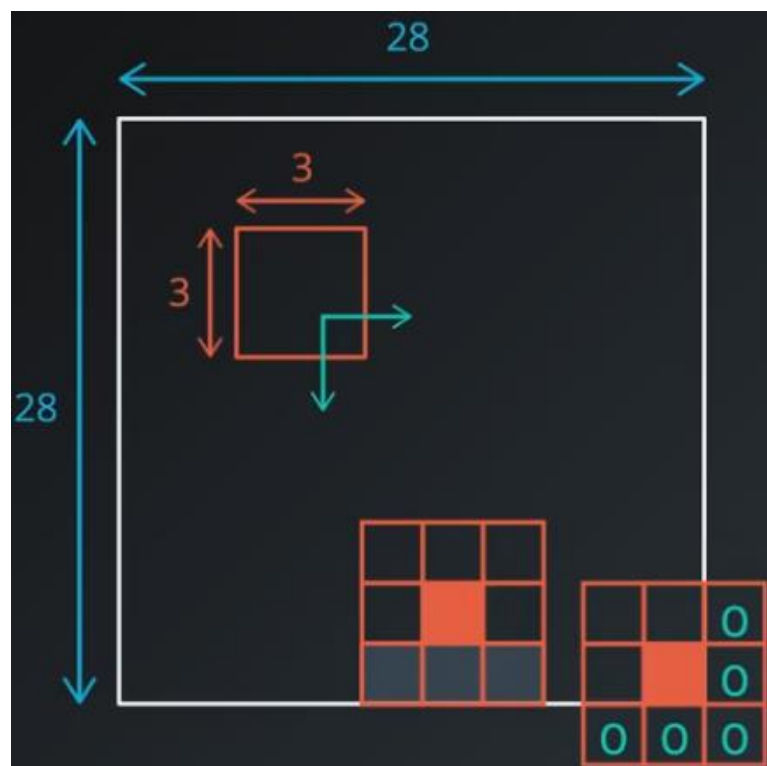


特征图尺寸练习



输出				
填充	步幅	宽	高	深度
Same	1			
Valid	1			
Valid	2			

特征图尺寸答案



输出				
填充	步幅	宽	高	深度
Same	1	28	28	8
Valid	1	26	26	8
Valid	2	13	13	8

卷积神经网络输出shape练习

H=高度, W=宽度, D=深度

Input shape=32*32*3(H*W*D)

20个滤波器, 滤波器shape=8*8*3

高和宽的stride=2

Padding size=1

求输出的shape:应该为 H*W*D

公式如下:

$$\text{new_height} = (\text{input_height} - \text{filter_height} + 2 * P) / S + 1$$

$$\text{new_width} = (\text{input_width} - \text{filter_width} + 2 * P) / S + 1$$

卷积神经网络输出shape答案

H=高度, W=宽度, D=深度

Input shape=32*32*3(H*W*D)

20个滤波器, 滤波器shape=8*8*3

高和宽的stride=2

Padding size=1

答案:

$\text{new_height} = (\text{input_height} - \text{filter_height} + 2 * P) / S + 1$

输出高度= $(32-8+2*1)/2 + 1 = 14$

$\text{new_width} = (\text{input_width} - \text{filter_width} + 2 * P) / S + 1$

宽度=14

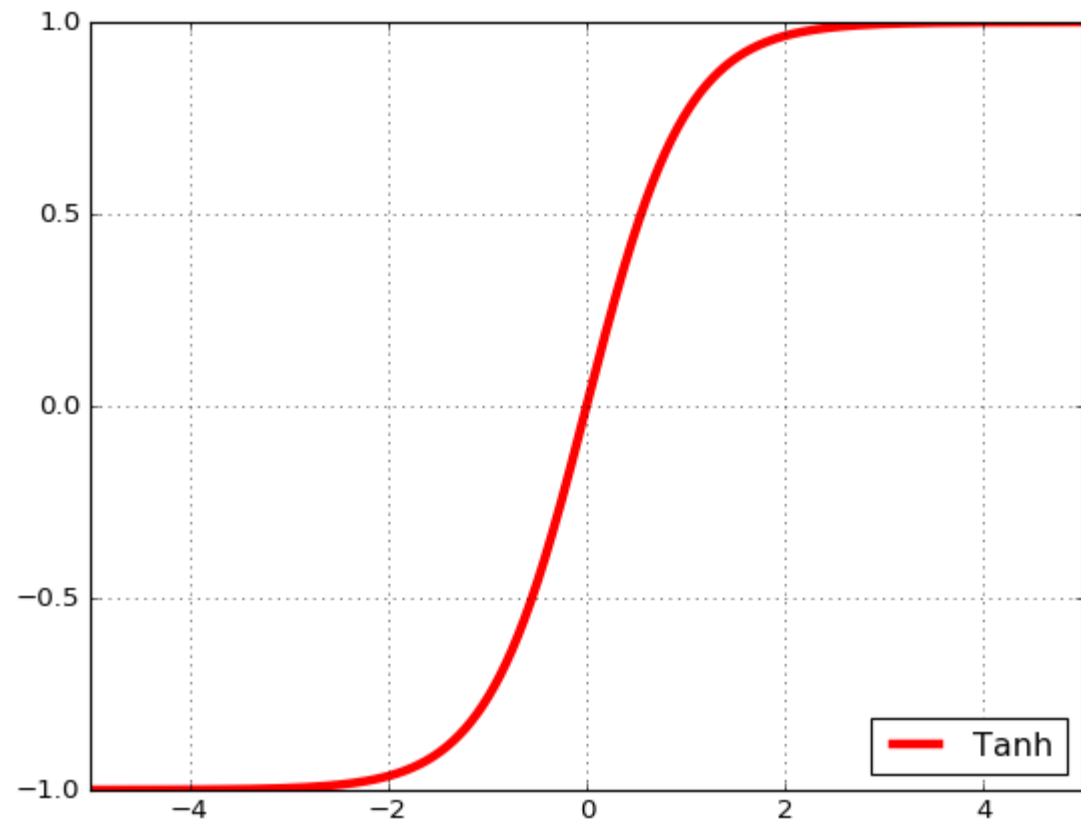
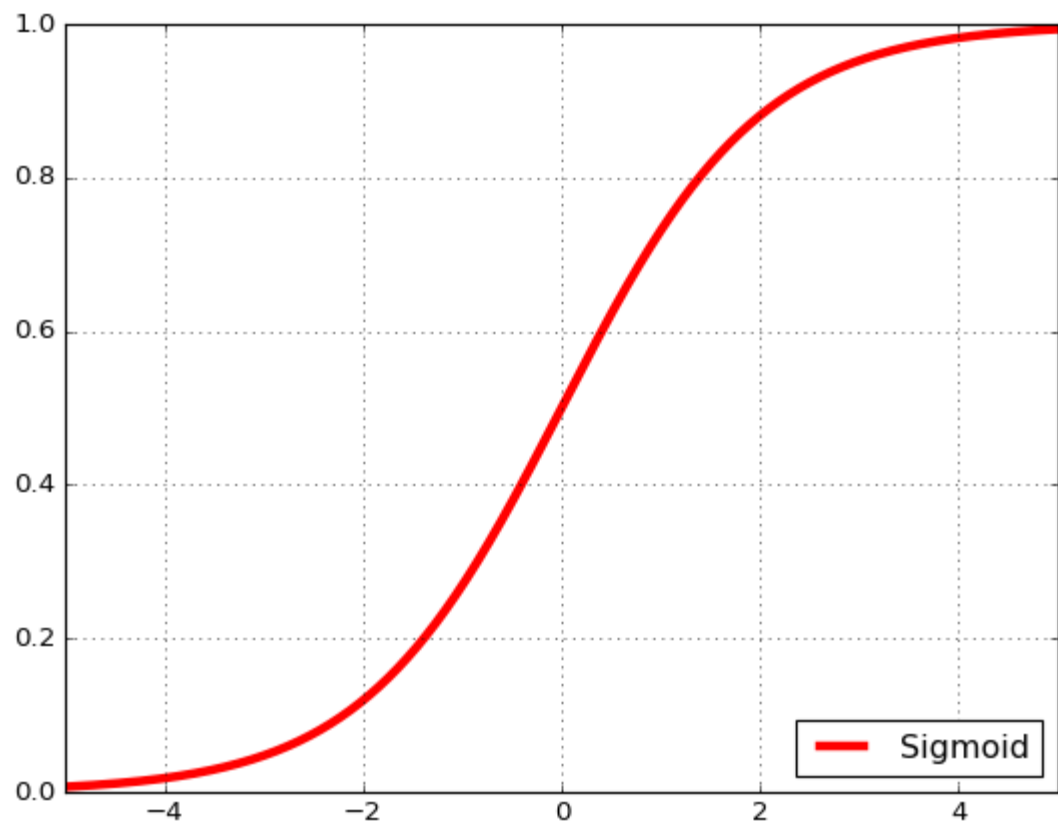
深度=20

卷积神经网络-ReLU Layer

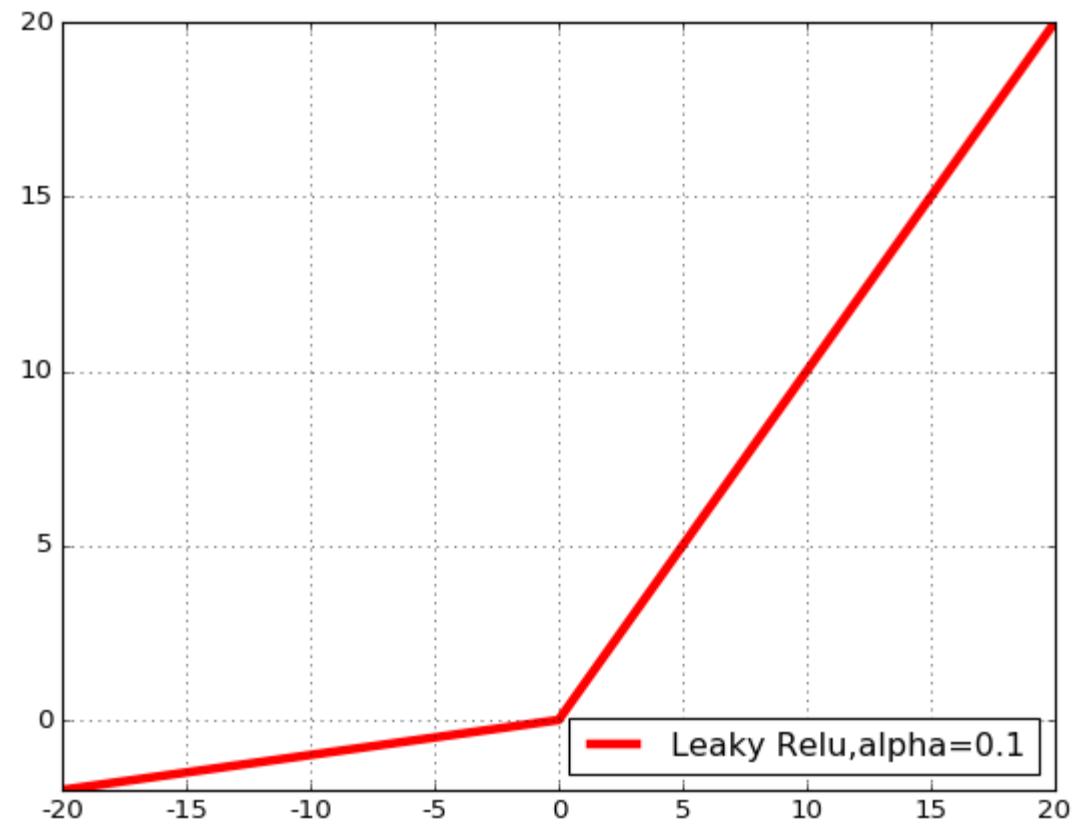
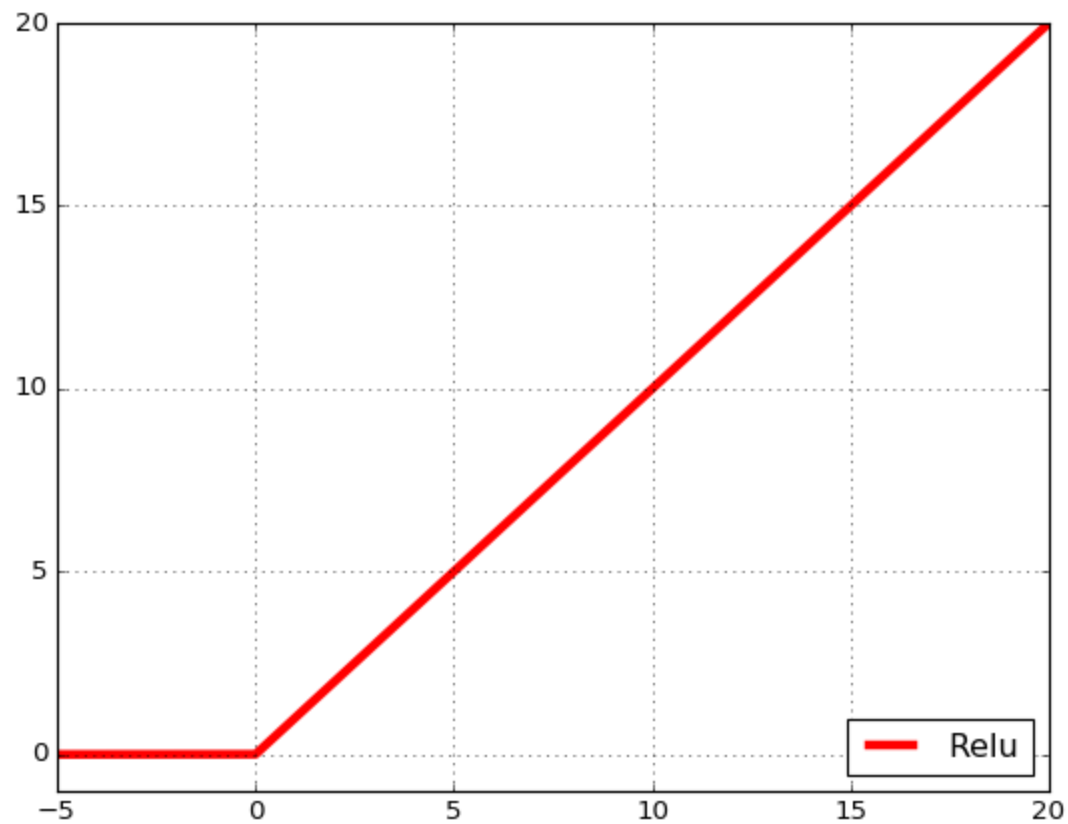
常用非线性映射函数

- Sigmoid(S形函数)
- Tanh(双曲正切,双S形函数)
- ReLU
- Leaky ReLU
- ELU
- Maxout

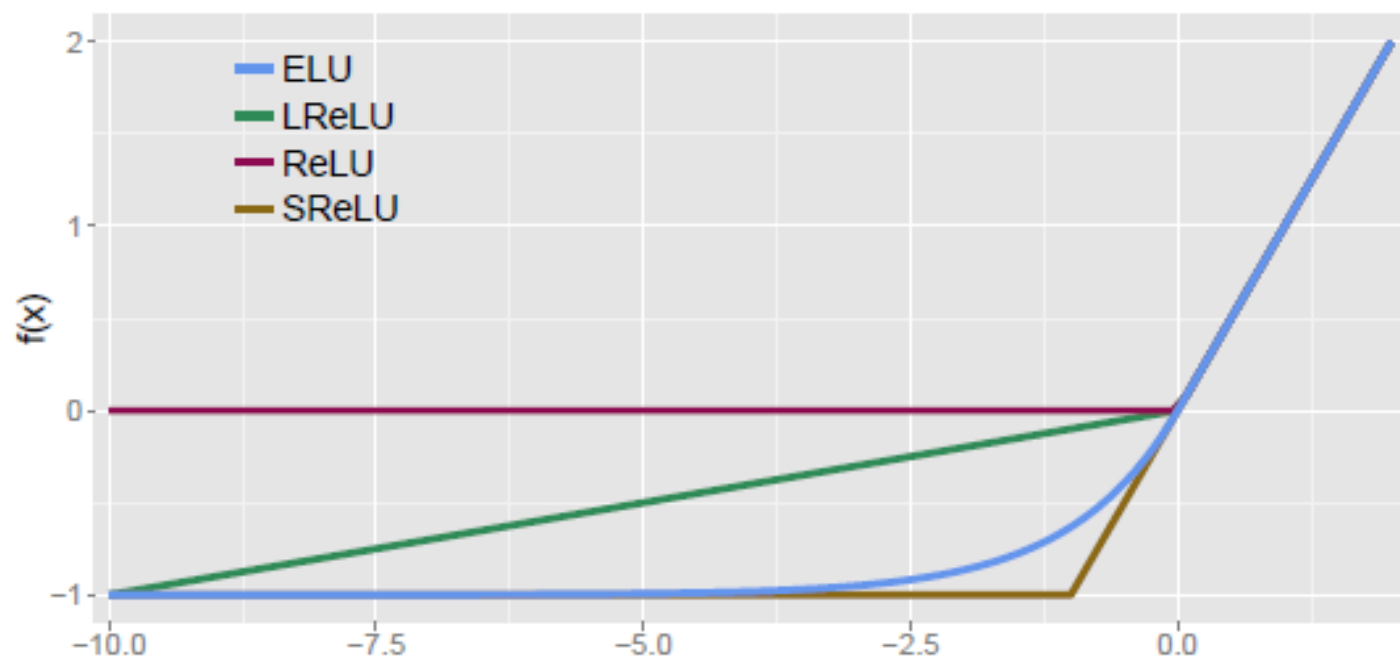
卷积神经网络-ReLU Layer



卷积神经网络-ReLU Layer



卷积神经网络-ReLU Layer



The *exponential linear unit* (ELU) with $0 < \alpha$ is

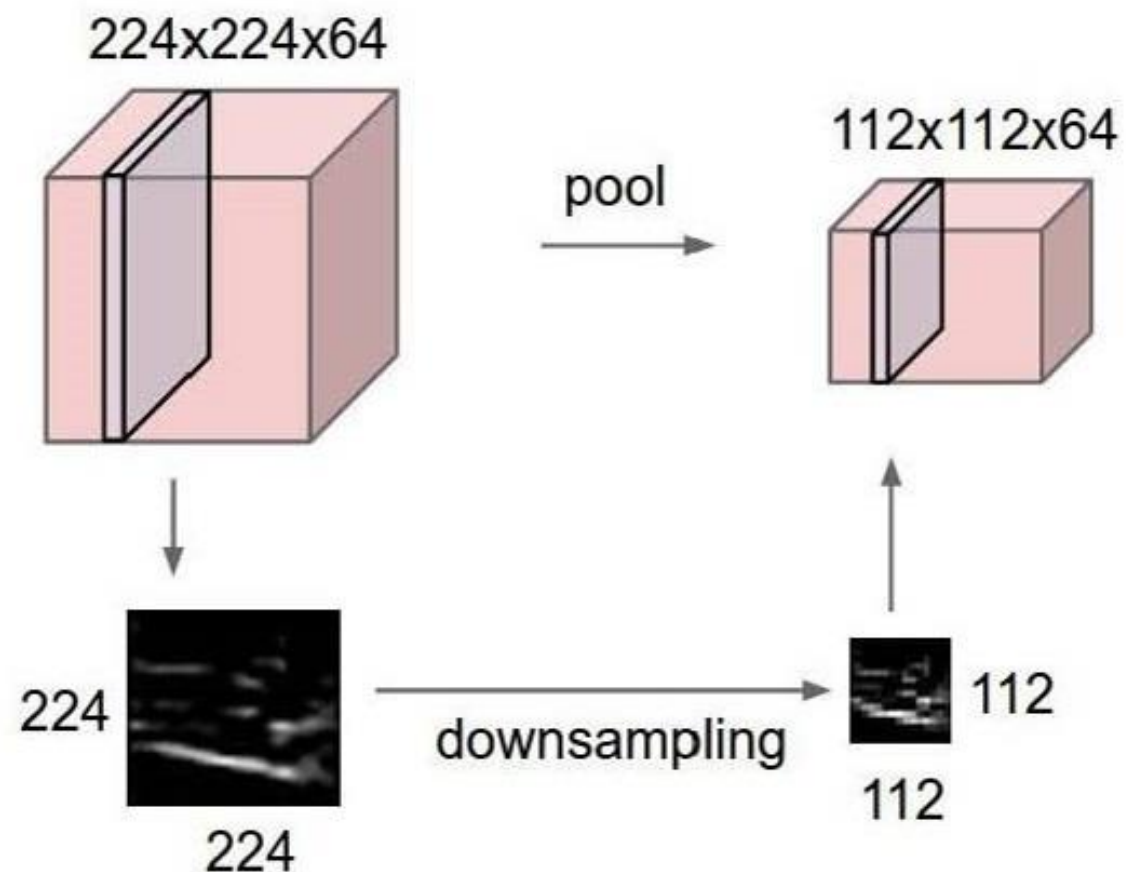
$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha (\exp(x) - 1) & \text{if } x \leq 0 \end{cases}, \quad f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \leq 0 \end{cases}.$$

卷积神经网络-ReLU Layer

激励层建议

- CNN尽量不要使用sigmoid，如果要使用，建议只在全连接层使用
- 首先使用RELU，因为迭代速度快，但是有可能效果不佳
- 如果使用RELU失效的情况下，考虑使用Leaky ReLu或者Maxout，此时一般情况都可以解决啦
- tanh激活函数在某些情况下有比较好的效果，但是应用场景比较少

池化层 (pooling layer)

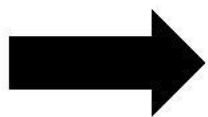


池化层 (pooling layer)

- 对图像进行下采样并不会改变图像中特征的相对位置

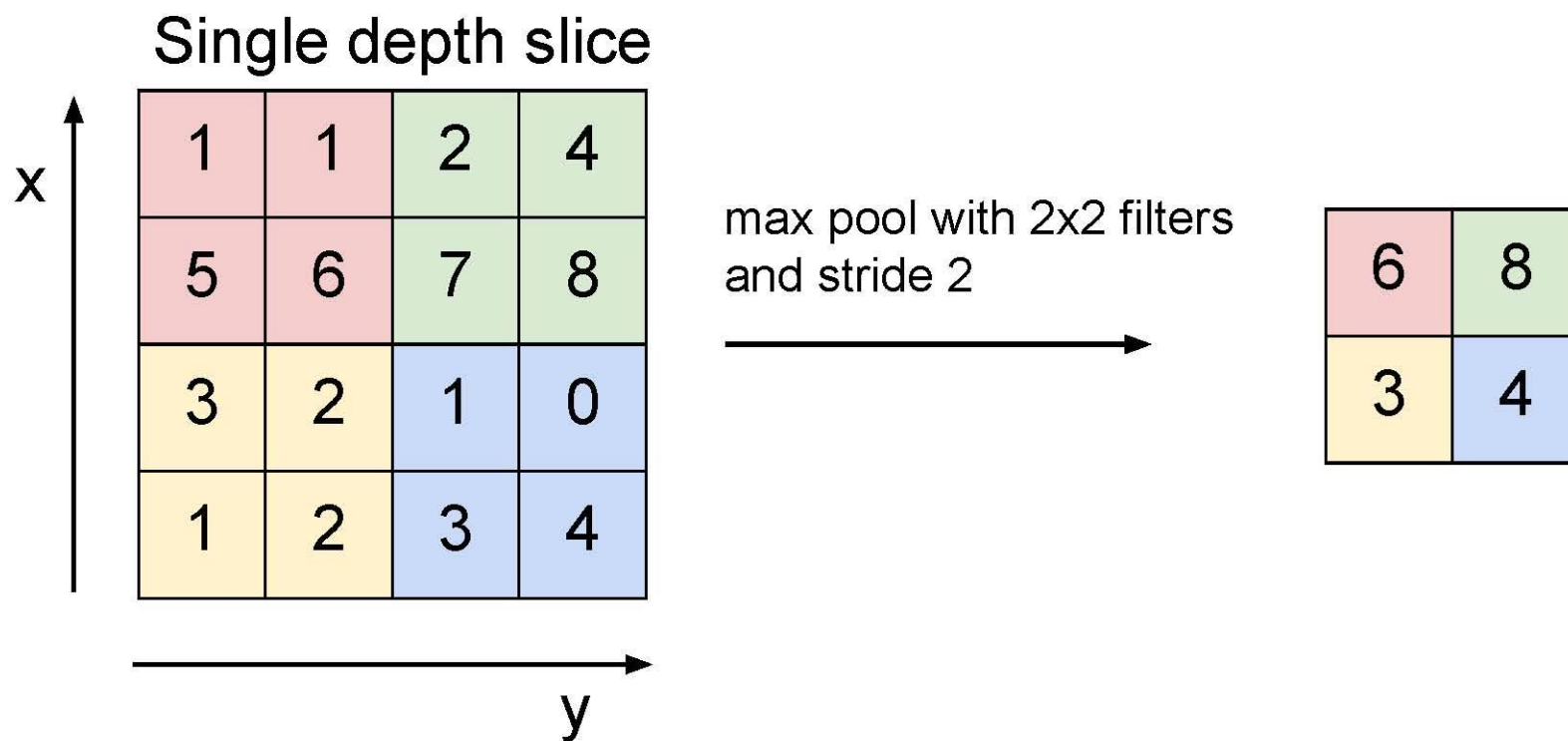


我们能够通过下采样降低图像的分辨率

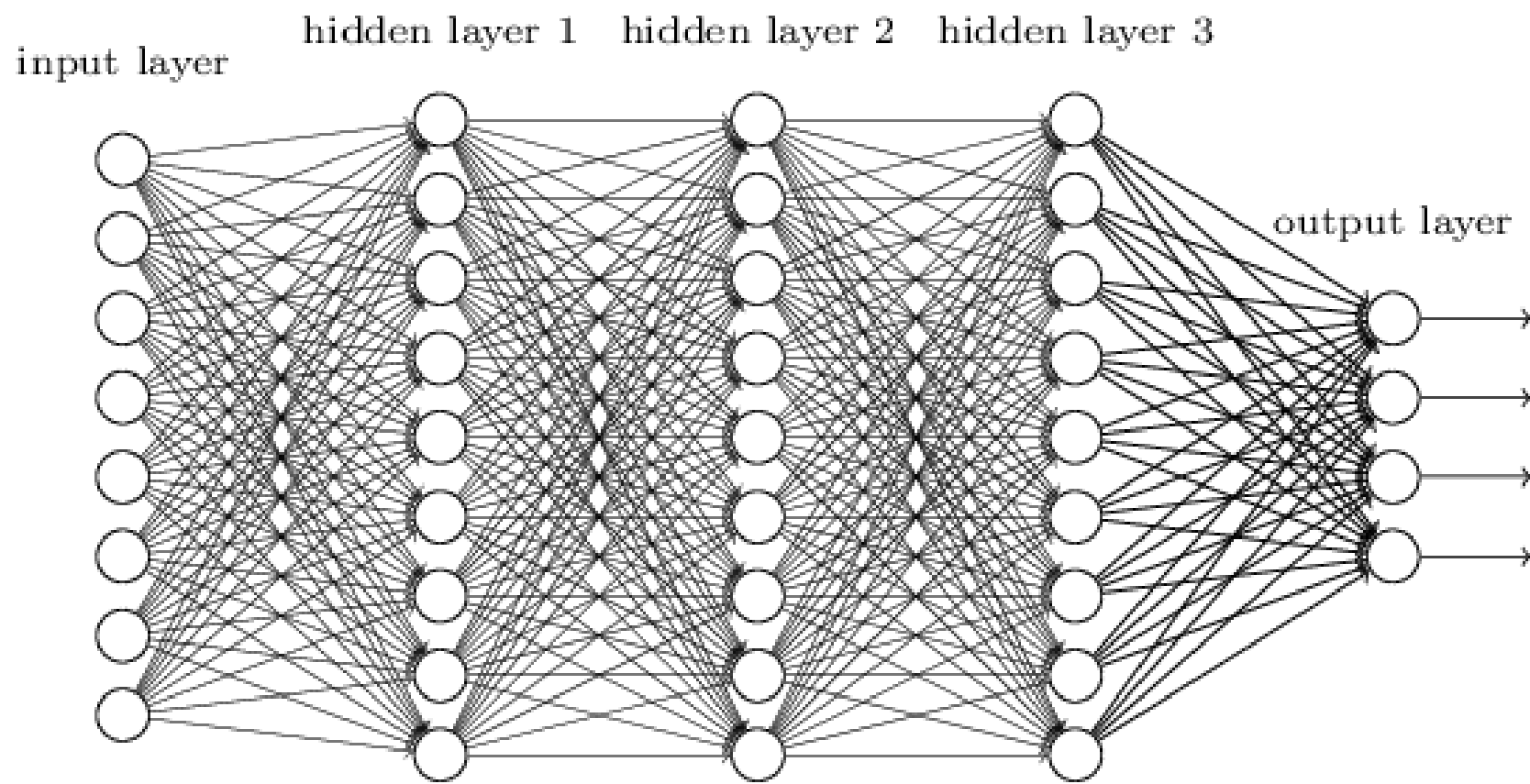


这样网络需要处理的参数就更少。

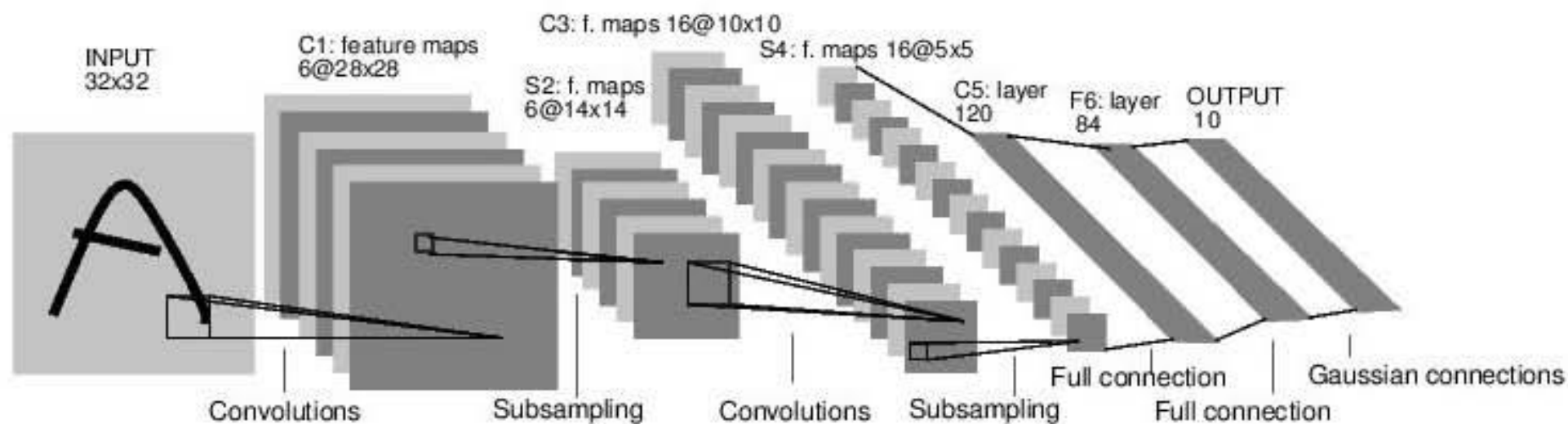
MAX POOLING



全连接层 (FC layer)



Convolutional Neural Networks



[LeNet-5, LeCun 1980]