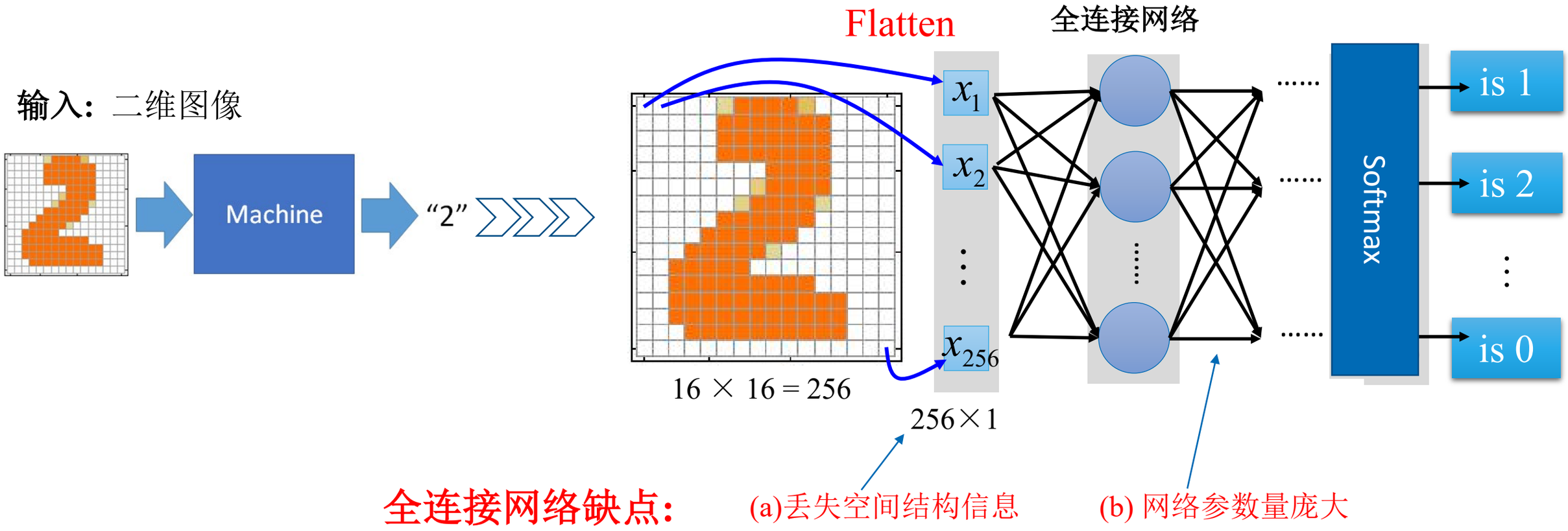
第8讲 深度学习基础1
——从卷积到卷积层

周文晖

杭州电子科技大学

上一讲的结束内容

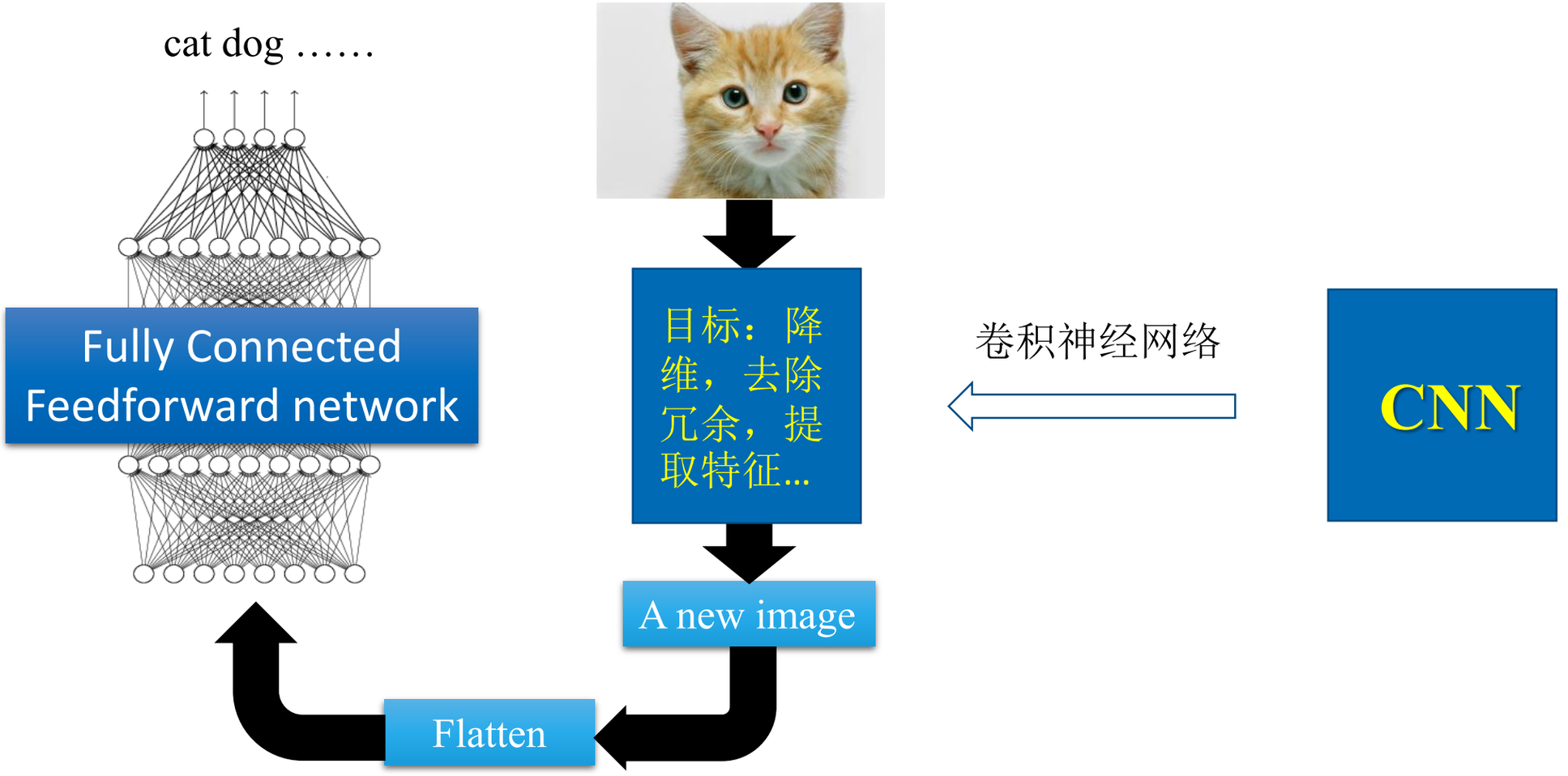
- 第一隐层神经元个数为 100，第一层权值数量为 256x100 = 25600



全连接网络缺点： (a)丢失空间结构信息　(b) 网络参数量庞大

实际场景中，往往需要更大的输入图像以及更深的网络结构

上一讲的结束内容

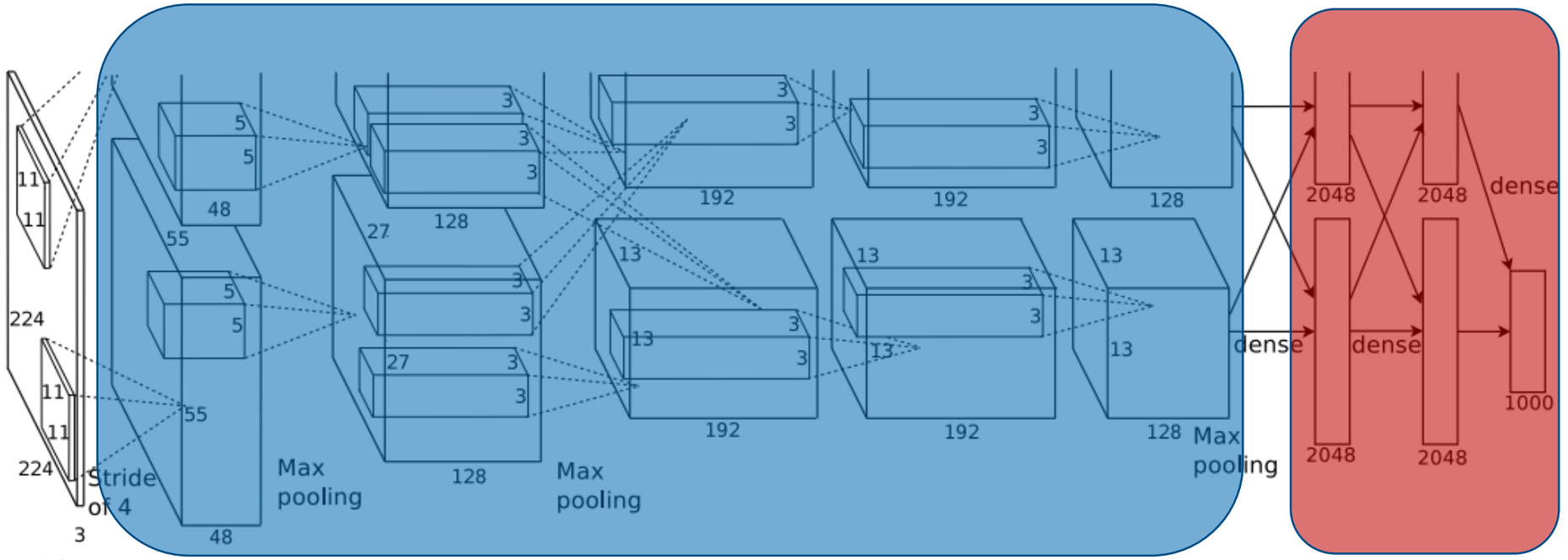

cat dog ……
Fully Connected Feedforward network
目标：降维，去除冗余，提取特征...
A new image
Flatten
卷积神经网络
CNN

Alexnet(2012)

模型总参数量：6千万

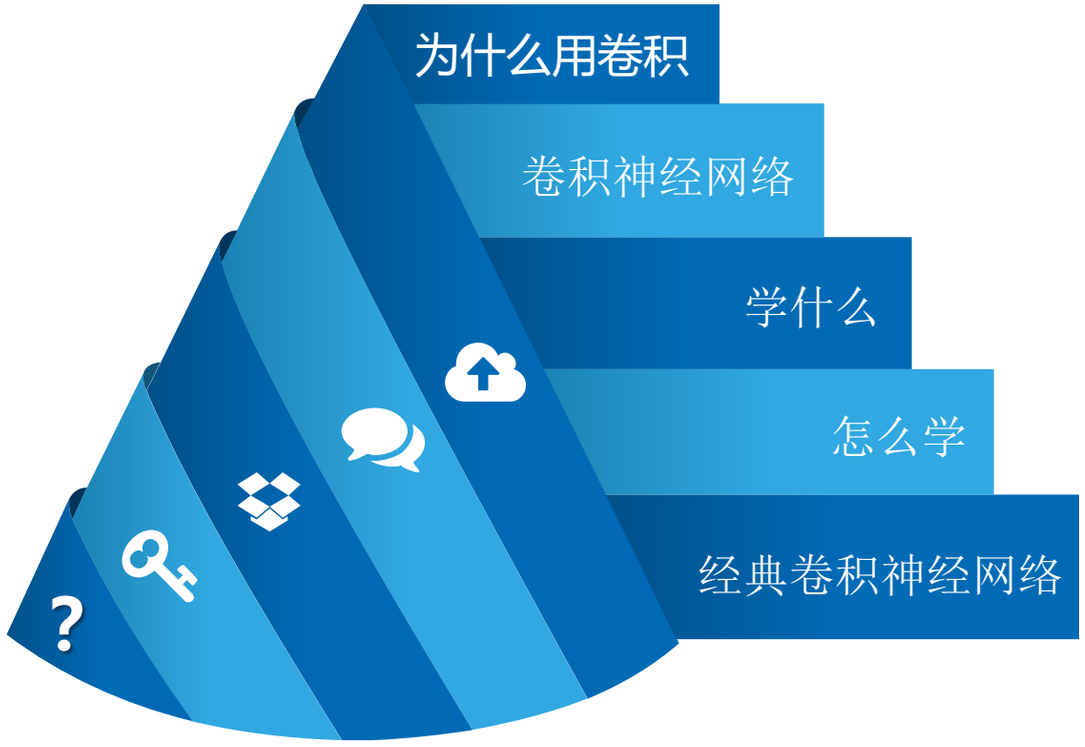
卷积层
（5%的参数, 95%的计算量）

全连接层
（95%的参数，5%的计算量）

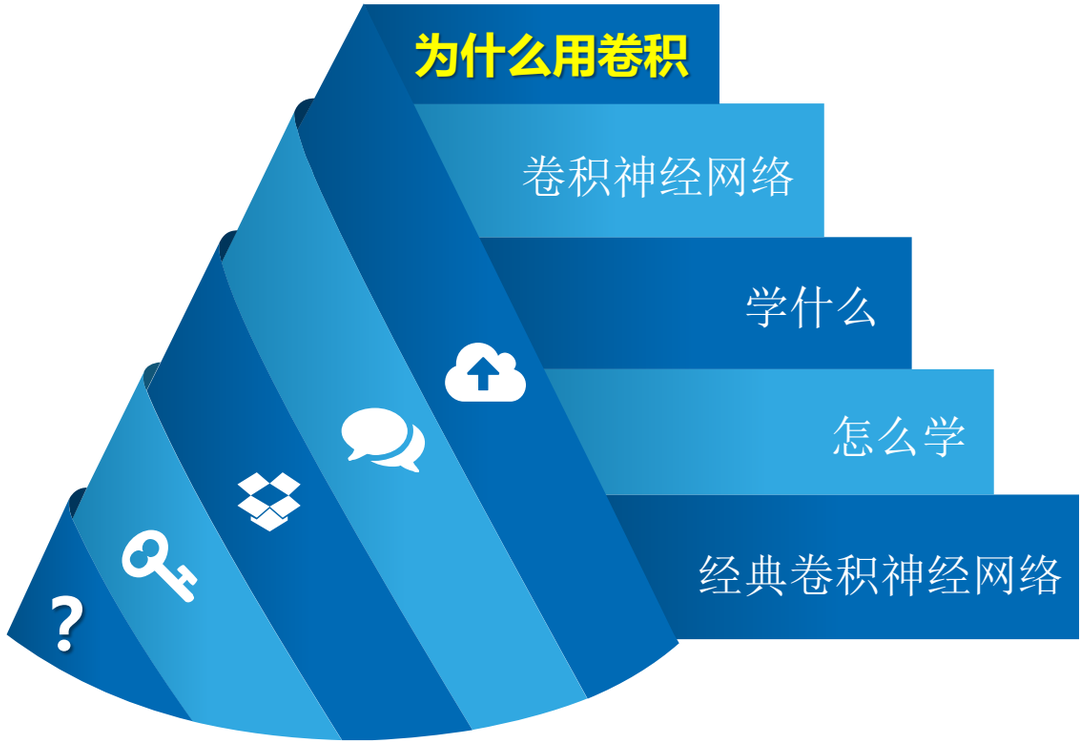


输入
224×224

卷积层输出
13×13



- **为什么要用卷积?**
  Convolution，…
- **卷积神经网络**
  Architecture, Pooling, …
- **学什么?**
  Feature extraction，…
- **怎么学?**
  优化算法, 数据增强, 数据归一化, …
- **经典卷积神经网络**
  AlexNet, VGG, ResNet, …



为什么要用卷积?

Convolution，…

卷积神经网络

Architecture, Pooling, …

学什么?

Feature extraction，…

怎么学?

优化算法, 数据增强, 数据归一化, …

经典卷积神经网络

AlexNet, VGG, ResNet, …

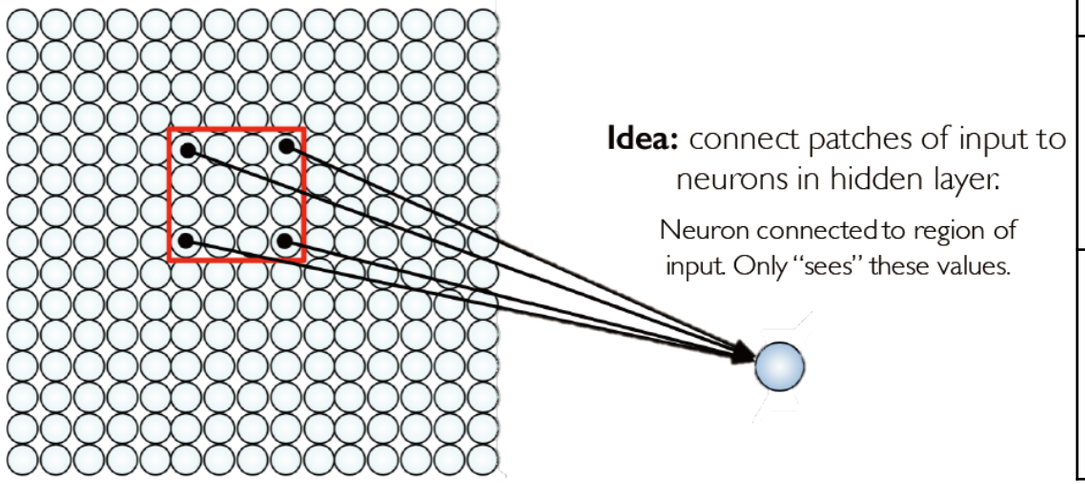
为什么用卷积？

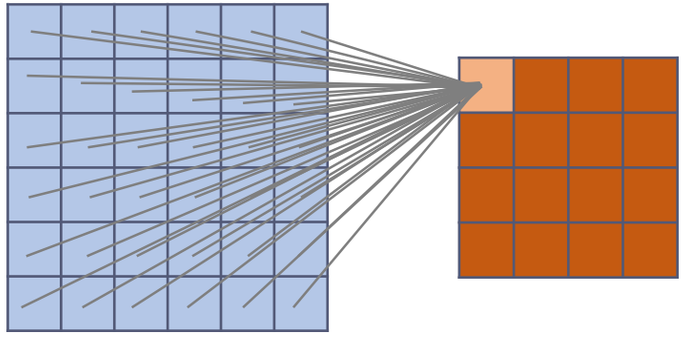
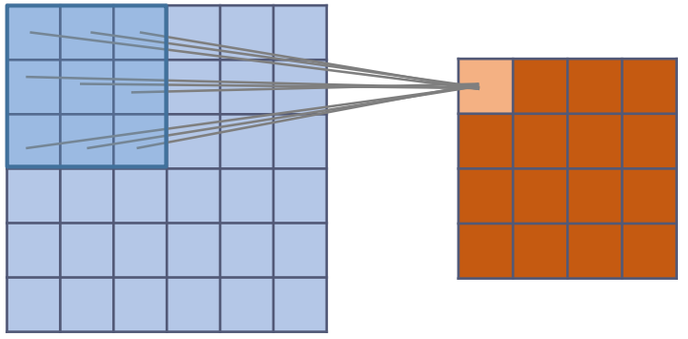
如何利用图像中的空间结构？

从神经元连接的角度理解卷积

输入：二维图像.

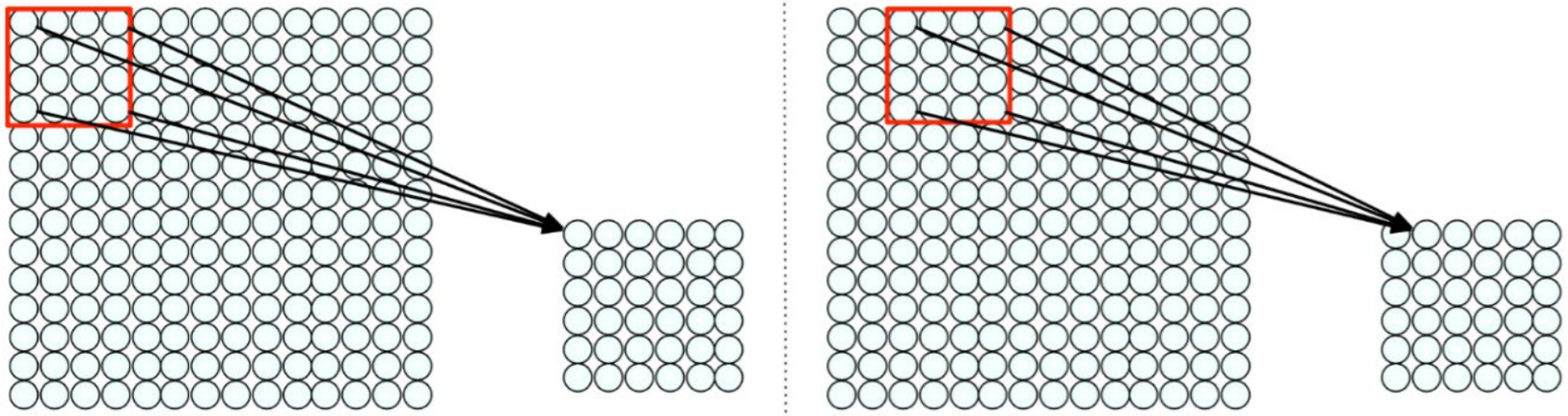
Array of pixel values



	全连接	卷积
局部连接		
权重共享	所有神经元之间的连接都使用不同权重。	输出层神经元共用同一组权重，进一步减少权重数量。
权重数量	$w_i \times h_i \times w_o \times h_o$	$f \times f$

为什么用卷积？

如何利用图像中的空间结构？——**从神经感受野角度理解卷积**



- 生物学上感受野（Receptive Field）机制：在视觉神经系统中，一个神经元的感受野是指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。

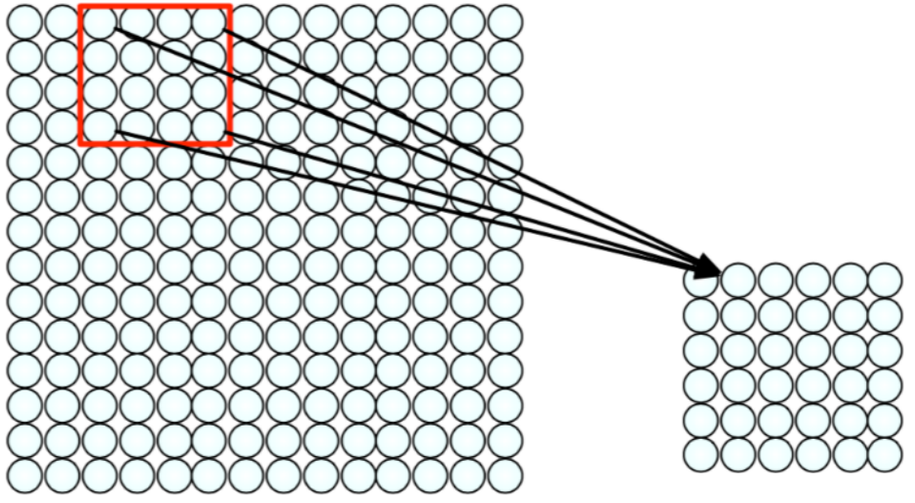
将输入图像中一个图像块作为下一层某个神经元的输入.

采用一个滑动窗口来定义神经元连接.

问题：如何通过对图像块**加权**来检测局部特征？

为什么用卷积？

卷积操作



4x4 filter: matrix of weights θ_{ij}

$$\sum_{i=1}^{4}\sum_{j=1}^{4}\theta_{ij}\,x_{i+p,j+q}+b$$

for neuron (p,q) in hidden layer

对于隐含层中的一个神经元:

- 仅“可见”输入的局部特征区域
- 计算加权和
- 偏置值

applying a window of weights
computing linear combinations
activating with non-linear function

为什么用卷积？

卷积操作——从信号处理的角度理解卷积

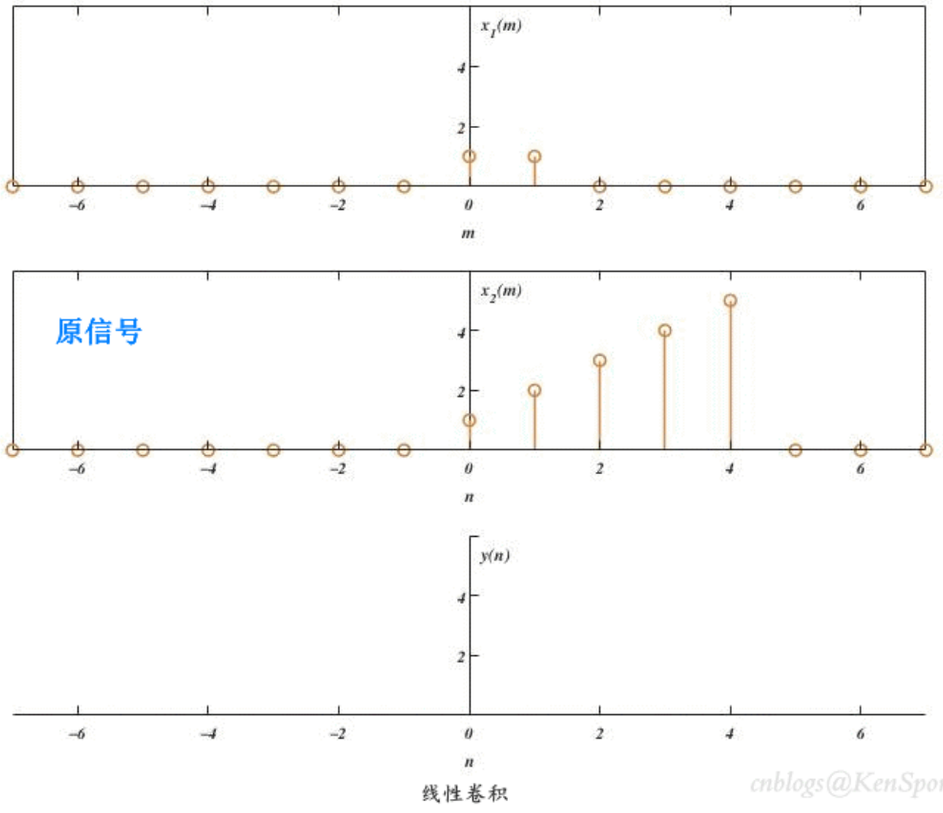
•卷积步骤

(1) 翻褶：

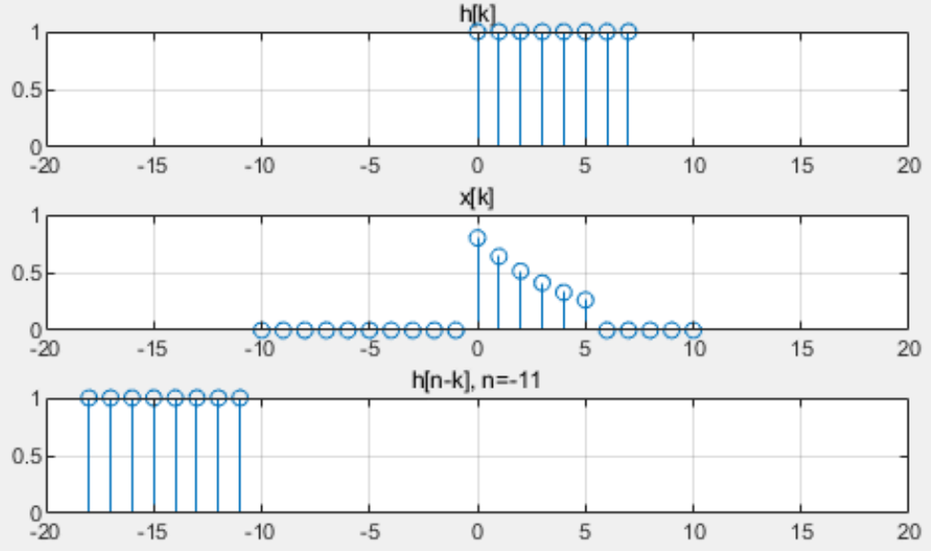
(2) 移位：

(3) 相乘：

(4) 相加：

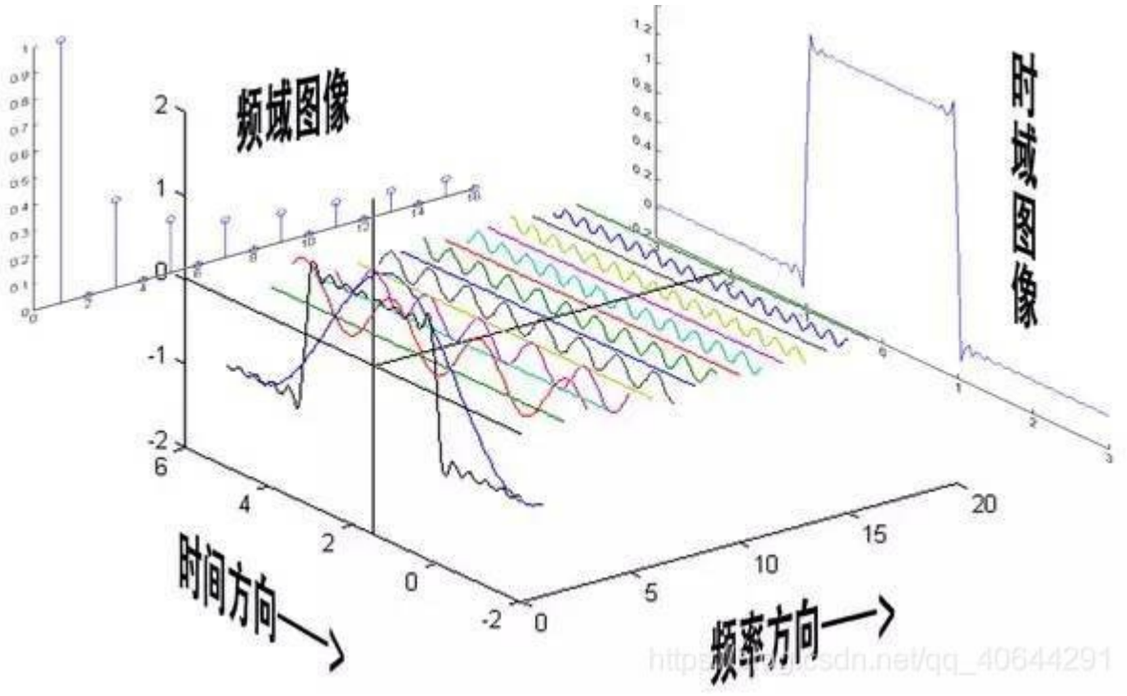


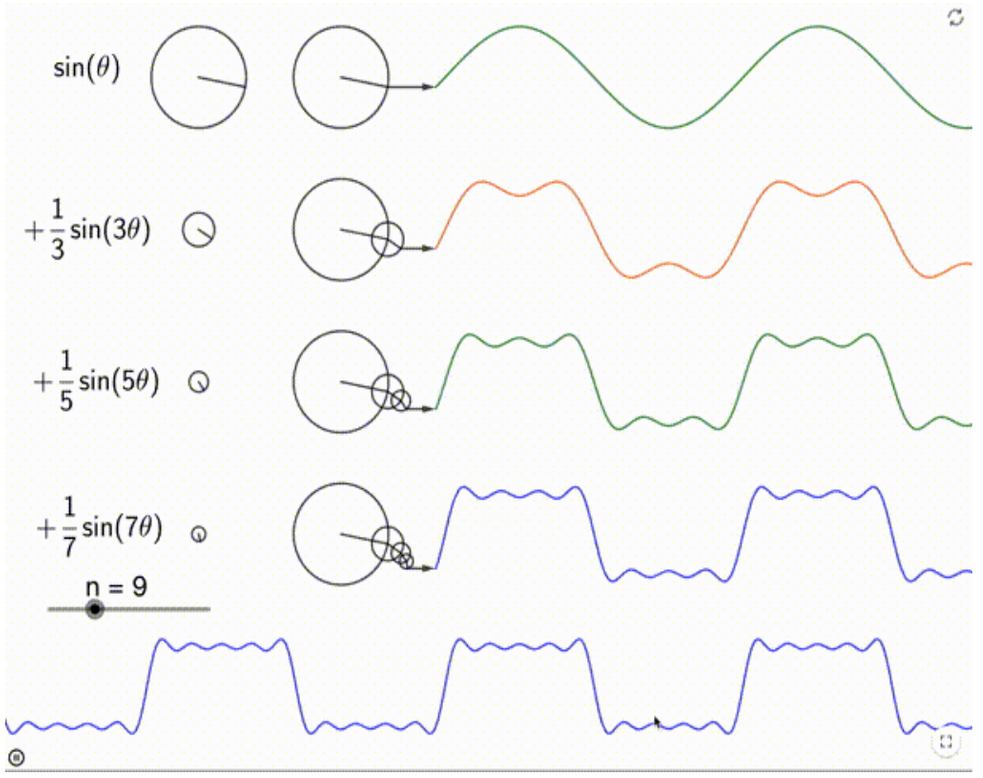
线性卷积



为什么用卷积？

卷积操作——**从信号处理的角度理解卷积**





为什么用卷积？

卷积操作——**从信号处理的角度理解卷积**

- **时域卷积定理**

若 $f_1(t) \leftrightarrow F_1(\omega),\ f_2(t) \leftrightarrow F_2(\omega)$

则 $f_1(t) * f_2(t) \leftrightarrow F_1(\omega) \cdot F_2(\omega)$

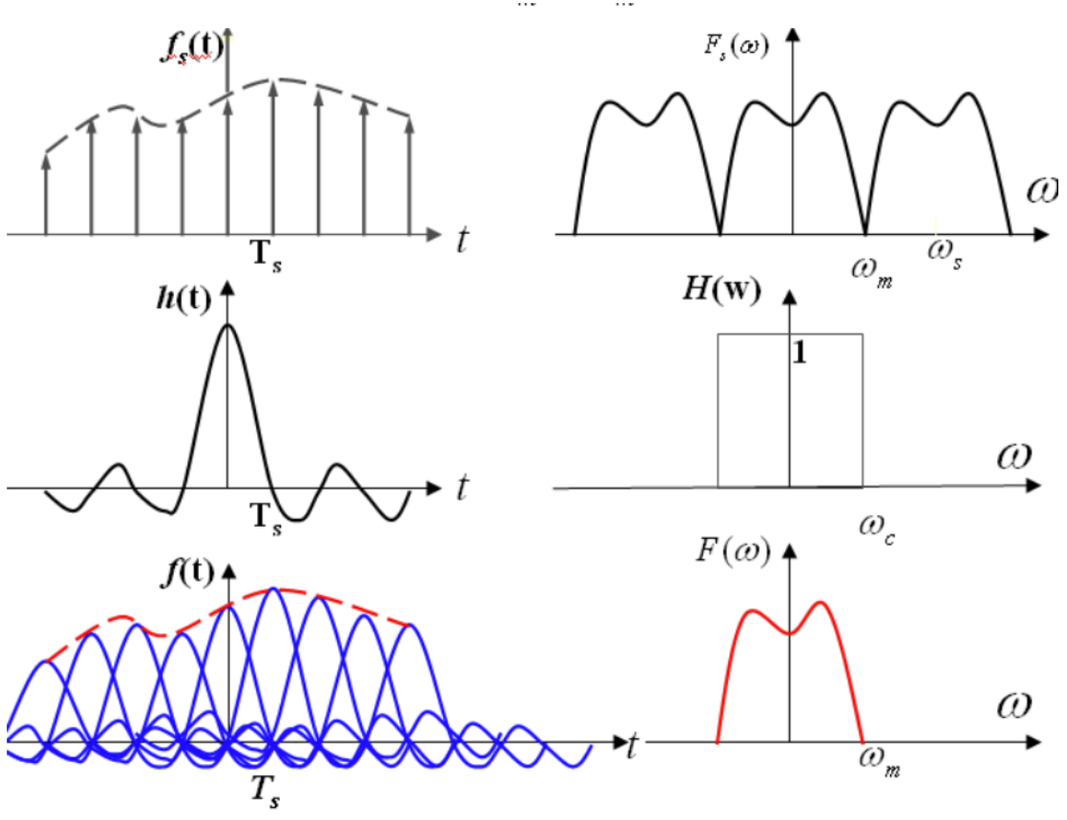
时域卷积对应频域频谱密度函数乘积。

- **频域卷积定理**

若 $f_1(t) \leftrightarrow F_1(\omega),\ f_2(t) \leftrightarrow F_2(\omega)$

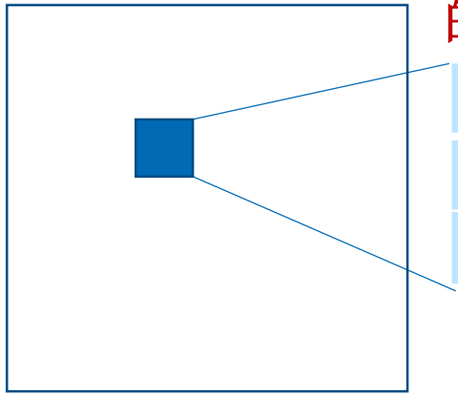
则 $f_1(t) \cdot f_2(t) \leftrightarrow \frac{1}{2\pi} F_1(\omega) * F_2(\omega)$

时间函数的乘积 ↔ 各频谱函数卷积的1/2π 倍。



为什么用卷积？

卷积操作——从图像处理中模板滤波的角度理解卷积



以 I_5 为中心的邻域区域

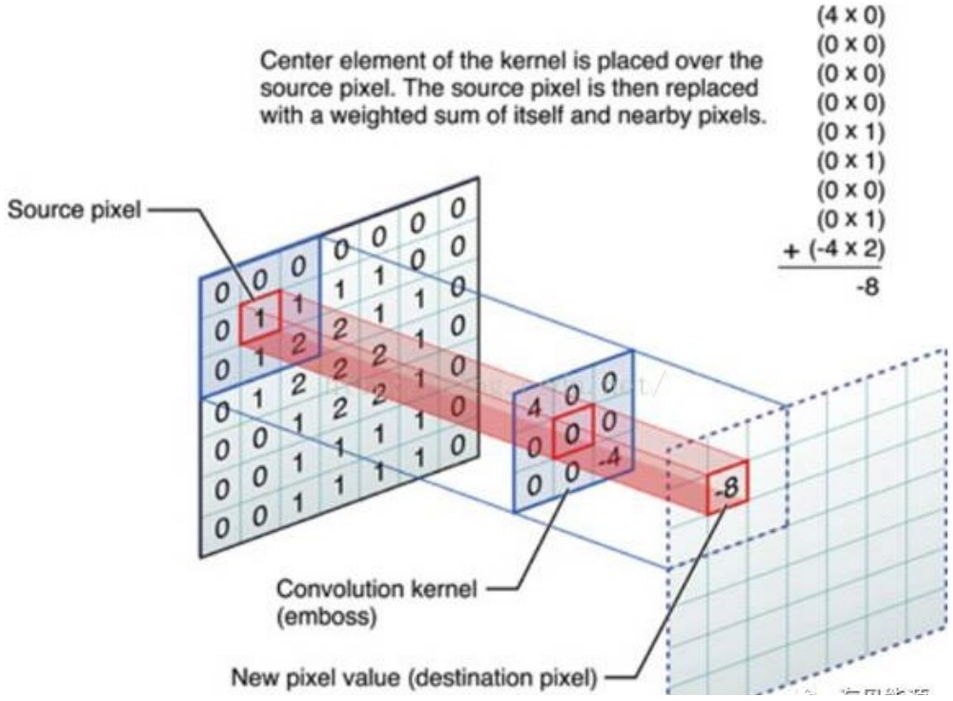
I_1	I_2	I_3
I_4	I_5	I_6
I_7	I_8	I_9

$*$

Kernel

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

卷积计算



$$I_5' = h_9 \cdot I_1 + h_8 \cdot I_2 + h_7 \cdot I_3 + h_6 \cdot I_4 + h_5 \cdot I_5 + h_4 \cdot I_6 + h_3 \cdot I_7 + h_2 \cdot I_8 + h_1 \cdot I_9$$

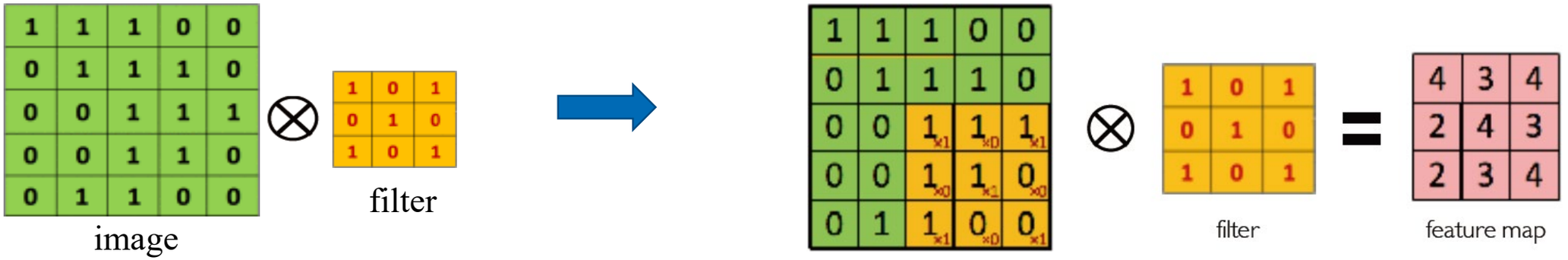
由于模板通常都是中心对称的，即可忽略模板以中心反转的过程，有

$$I_5' = h_1 \cdot I_1 + h_2 \cdot I_2 + h_3 \cdot I_3 + h_4 \cdot I_4 + h_5 \cdot I_5 + h_6 \cdot I_6 + h_7 \cdot I_7 + h_8 \cdot I_8 + h_9 \cdot I_9$$

为什么用卷积？

卷积算子

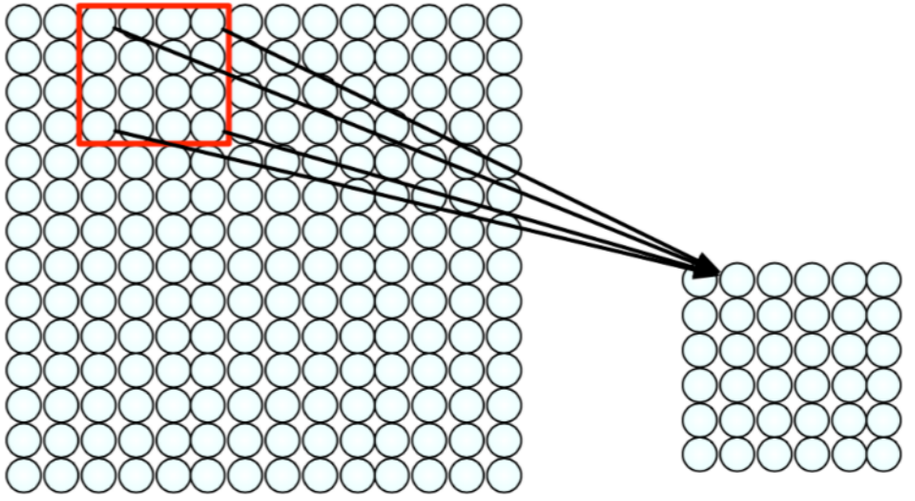
Suppose we want to compute the convolution of a 5x5 image and a 3x3 filter:

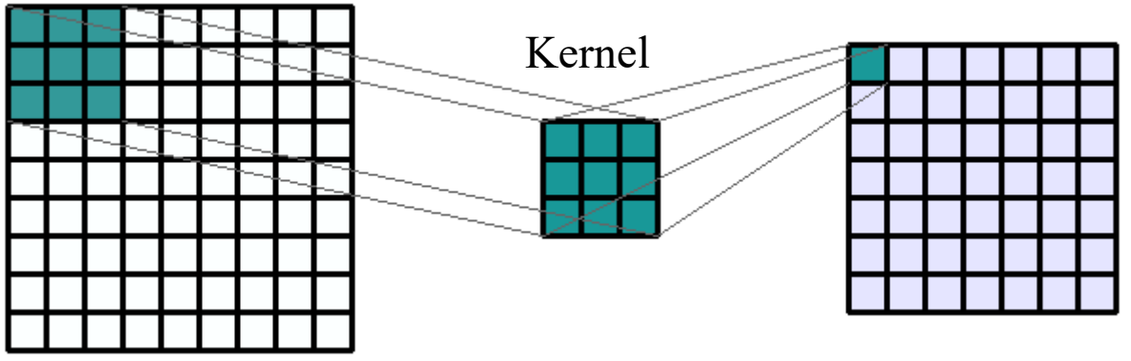


We slide the 3x3 filter over the input image, element-wise multiply, and add the outputs…

为什么用卷积？

卷积操作：提取语义特征





This “patchy” operation is **convolution**

- Filter of size 4×4 : 16 different weights
- Apply this same filter to 4×4 patches in input
- Shift by 2 pixels for next patch

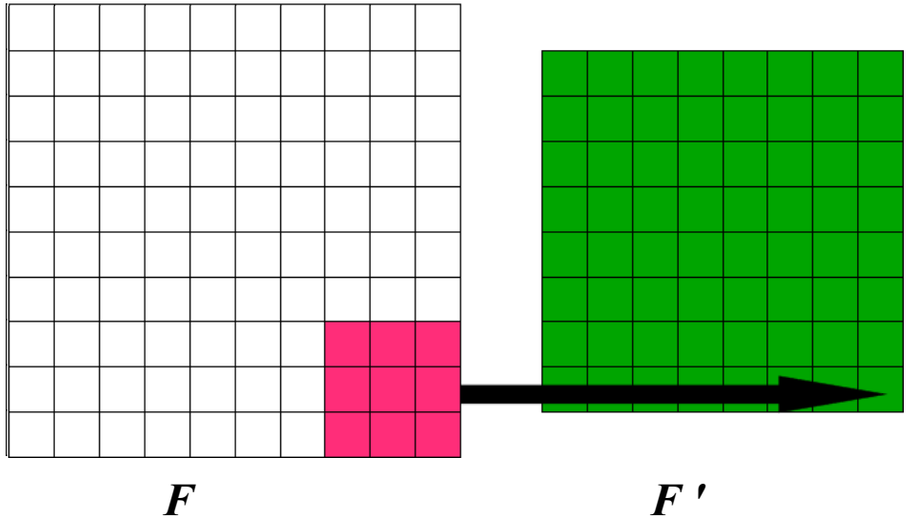
1) Apply a set of weights – a filter – to extract **local features**

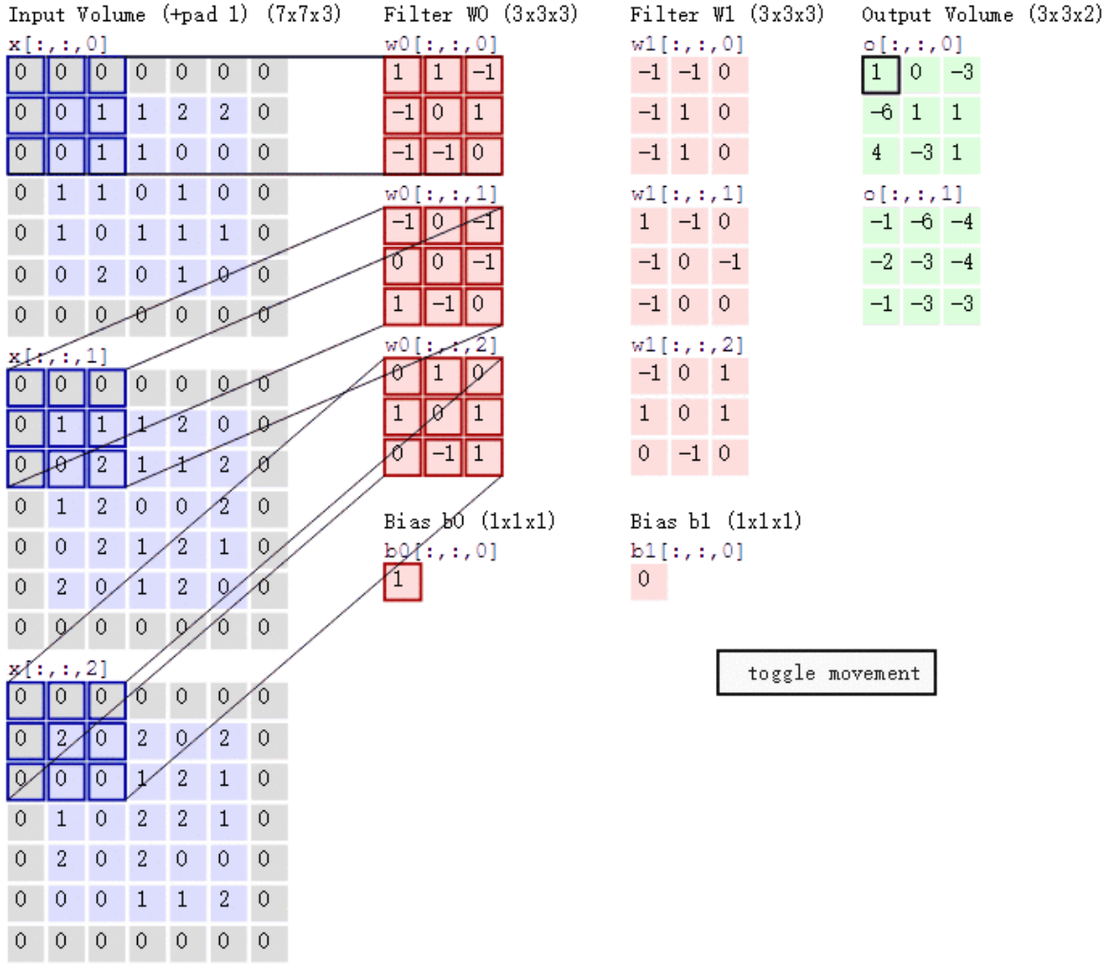
2) Use **multiple filters** to extract different features

3) **Spatially share** parameters of each filter

为什么用卷积？

卷积操作

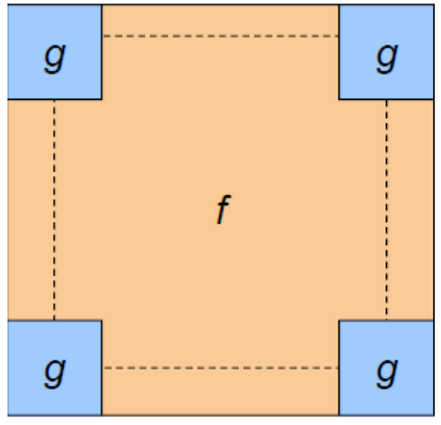




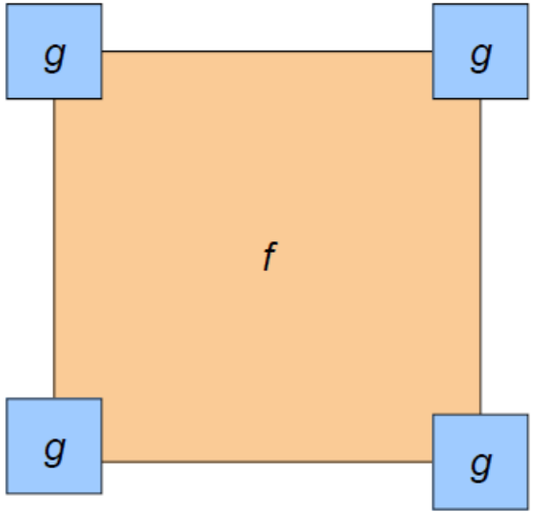
卷积滤波过程：遍历图像中所有像素，计算每个像素的邻域与模板的卷积值。

为什么用卷积？

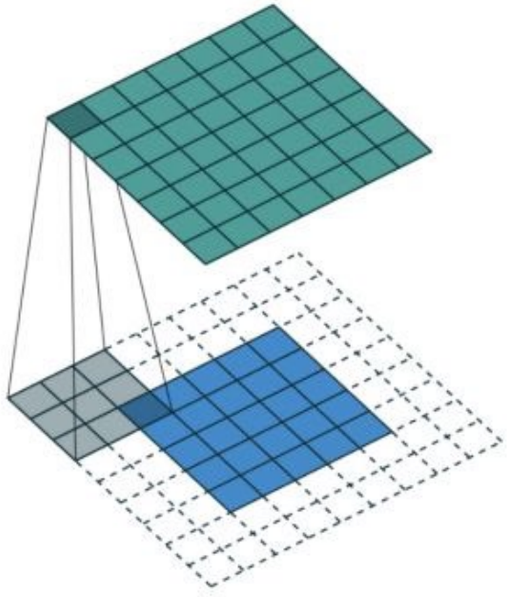
卷积的边缘问题：Padding（填充）



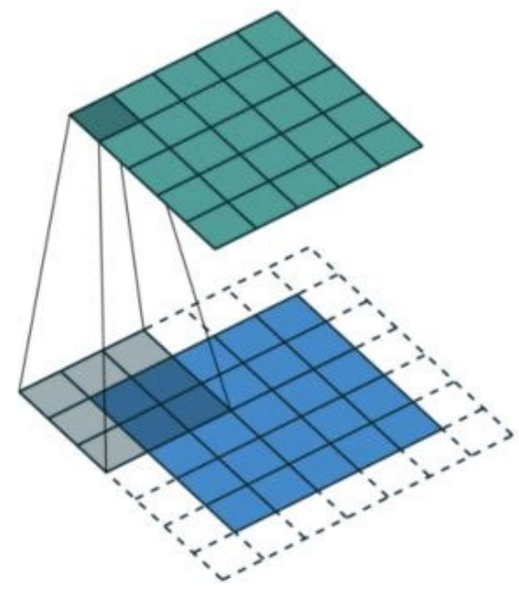
“Valid”Mode
without padding



“Same”Mode
Zero-padding



Full padding. Introduces zeros such that all pixels are visited the same amount of times by the filter. Increases size of output.



Same padding. Ensures that the output has the same size as the input.

为什么用卷积？ Feature Extraction with Convolution

Original

Sharpen

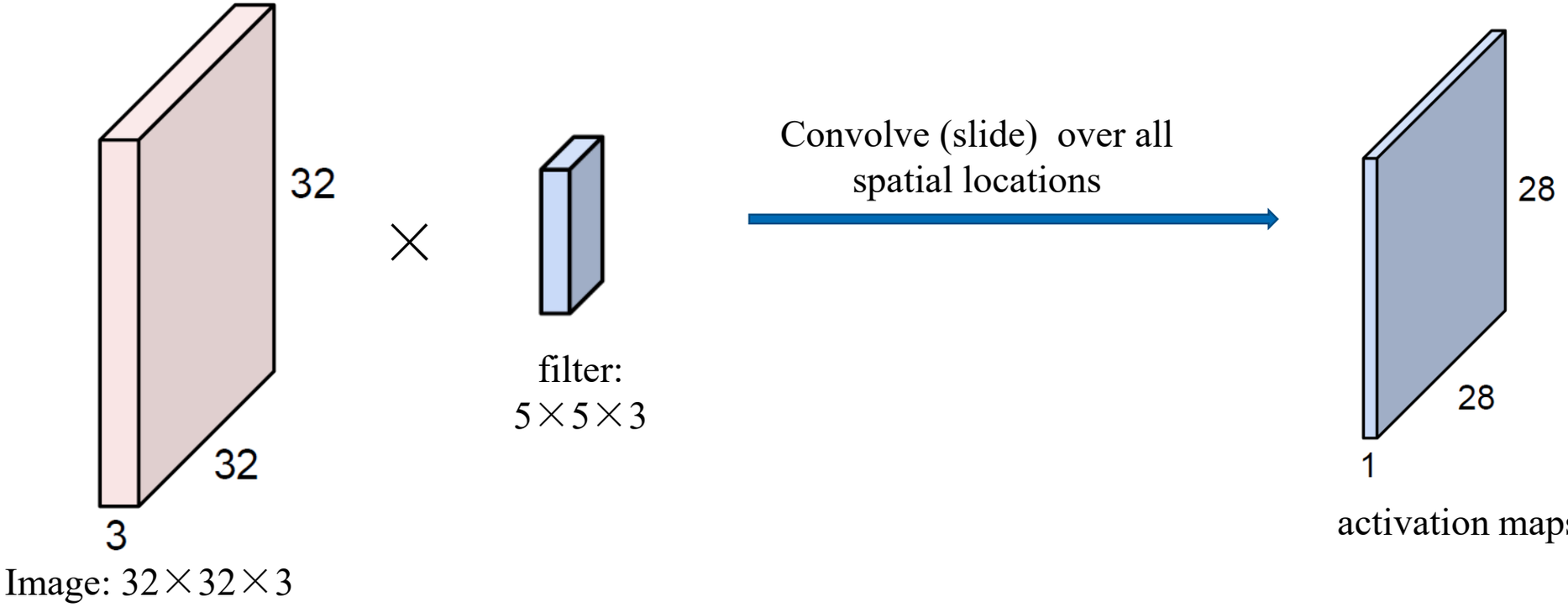
Edge Detect

“Strong” Edge Detect

原图 浮雕

为什么用卷积？

卷积层



Apply a set of weights – a filter – to extract **local features**

为什么用卷积？

卷积层

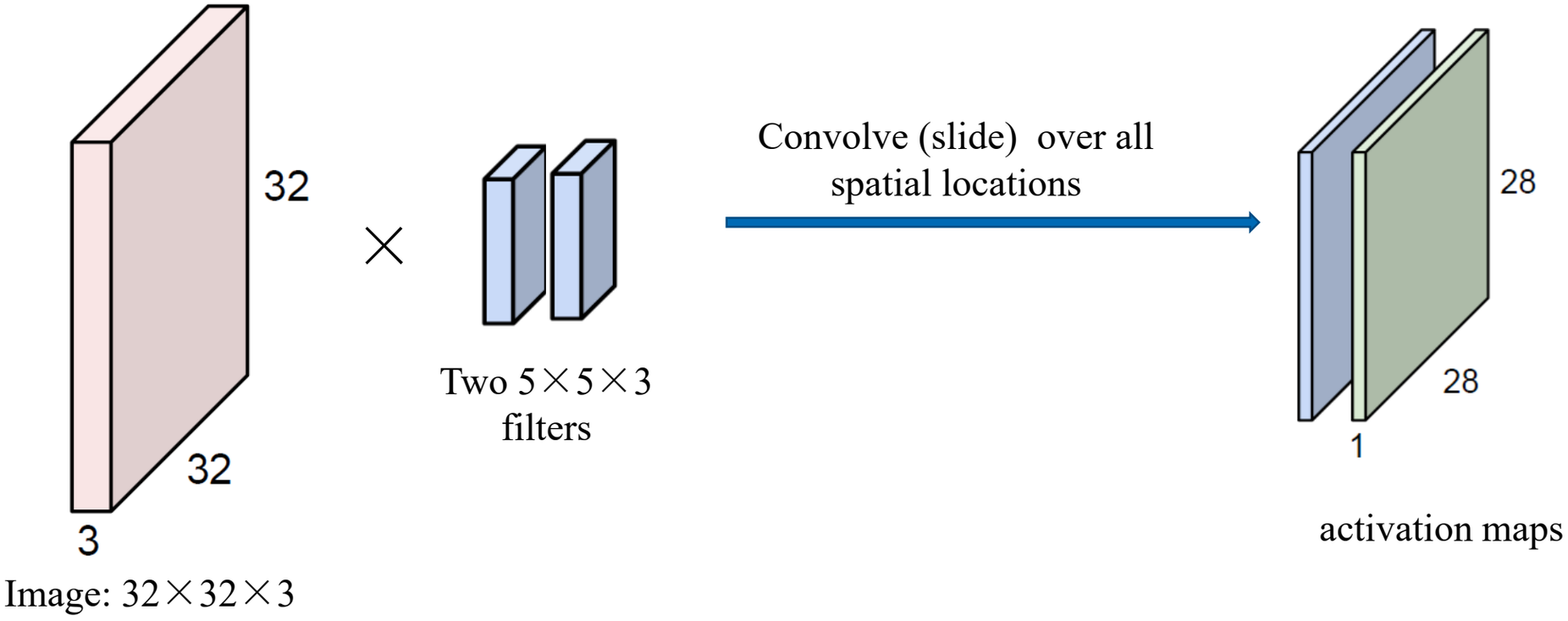


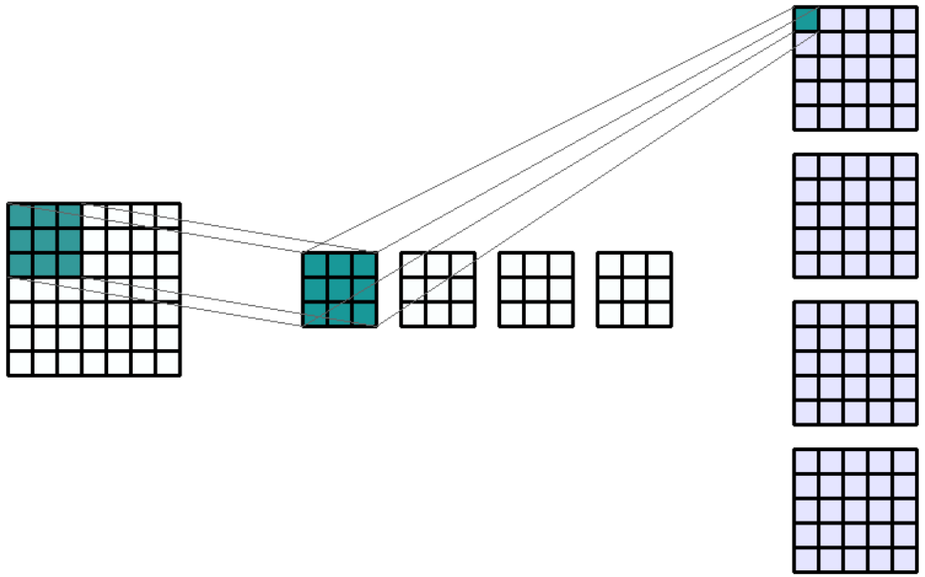
Image: 32×32×3

Use **multiple filters** to extract different features

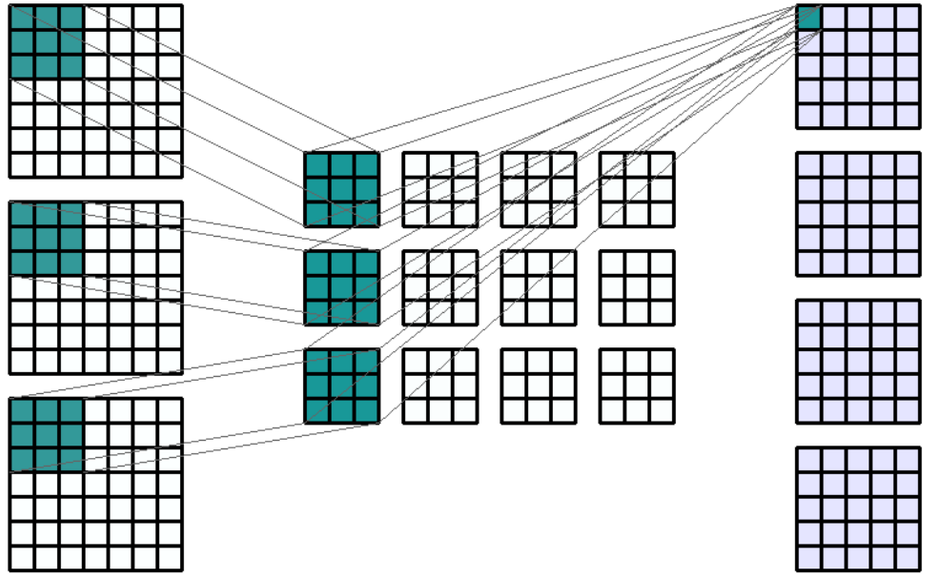
为什么用卷积？

输入输出通道

灰度图像（单通道输入）



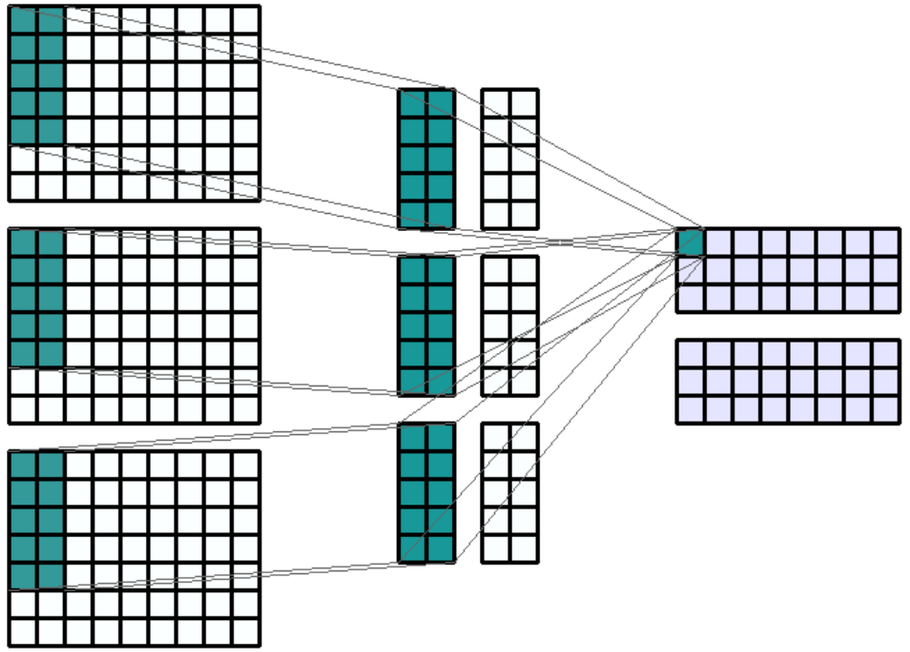
彩色图像（三通道输入）



为什么用卷积？

滤波器核大小

可以使用不同高度和宽度的滤波器核。在信号图像分析中通常是这种情况。

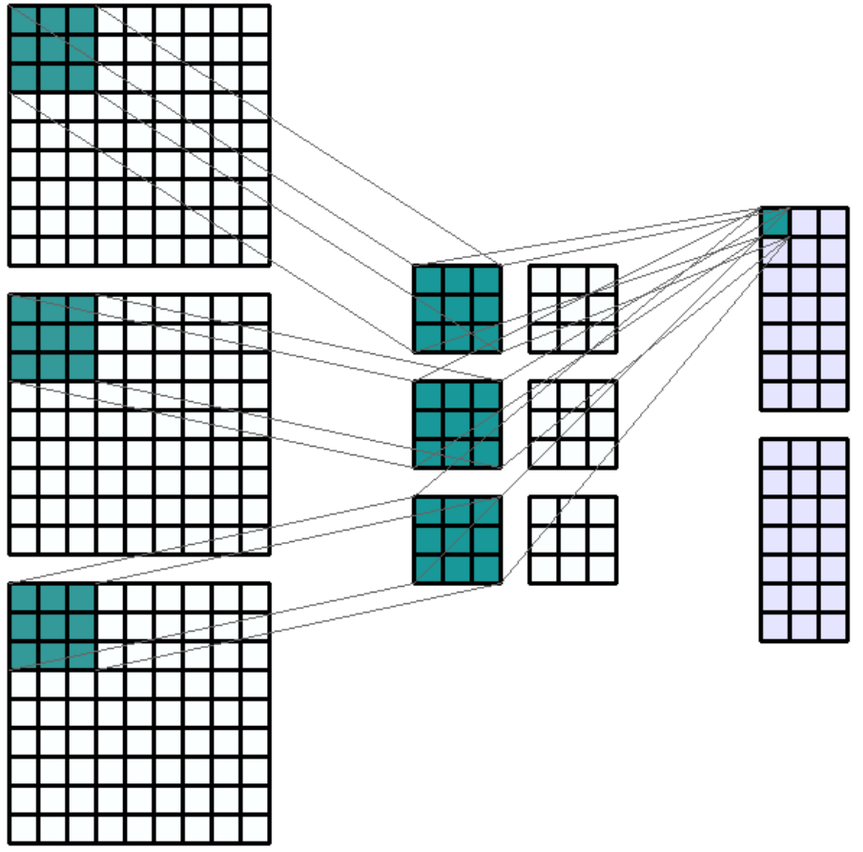


为什么用卷积？

步长 Strides

默认情况下，滤波器从左到右，从下到上，从一个像素移动到另一个像素，即步长为1。

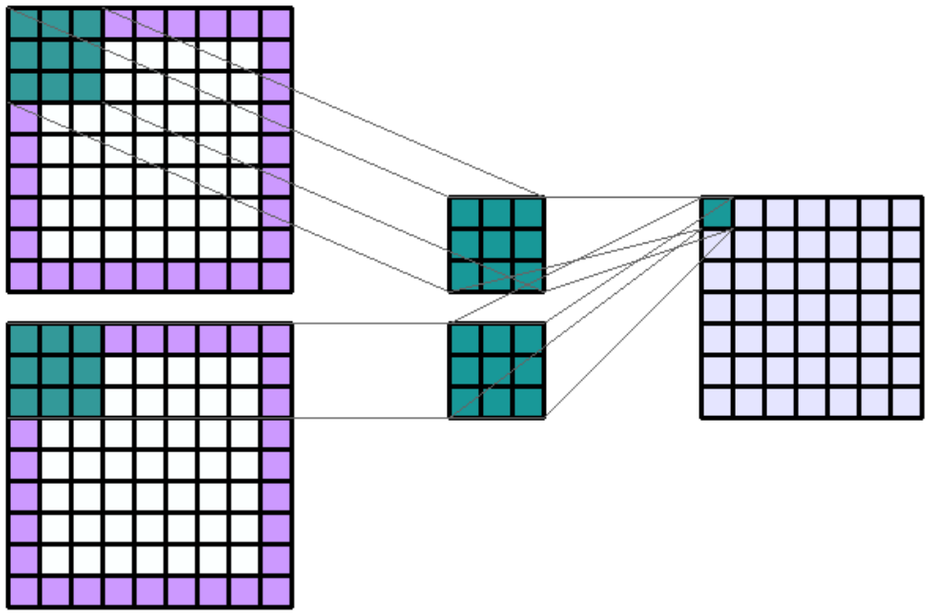
但步长可以改变，通常用于对输出通道进行下采样。例如，当步长为(1,3)时，两个数分别代表了垂直滑动和水平滑动步长值。这将产生水平下采样3的输出通道。



为什么用卷积？

Padding（填充）

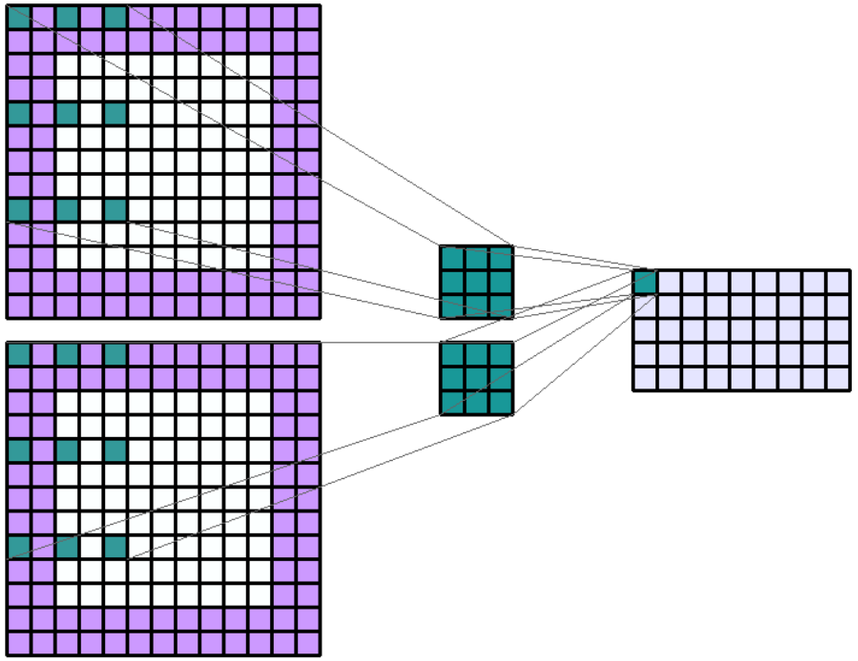
当希望输出大小与输入大小相等时，用Padding扩大输入图像



为什么用卷积？

Dilation（空洞卷积）

提升神经元的感受野。以（4，2）的Dilation卷积为例，那么输入通道上核的感受野会在垂直方向上扩大了4 *（3 -1）= 8，水平方向扩大了2 * (3–1)=4 （对于3乘3的核）。



为什么用卷积？

卷积层

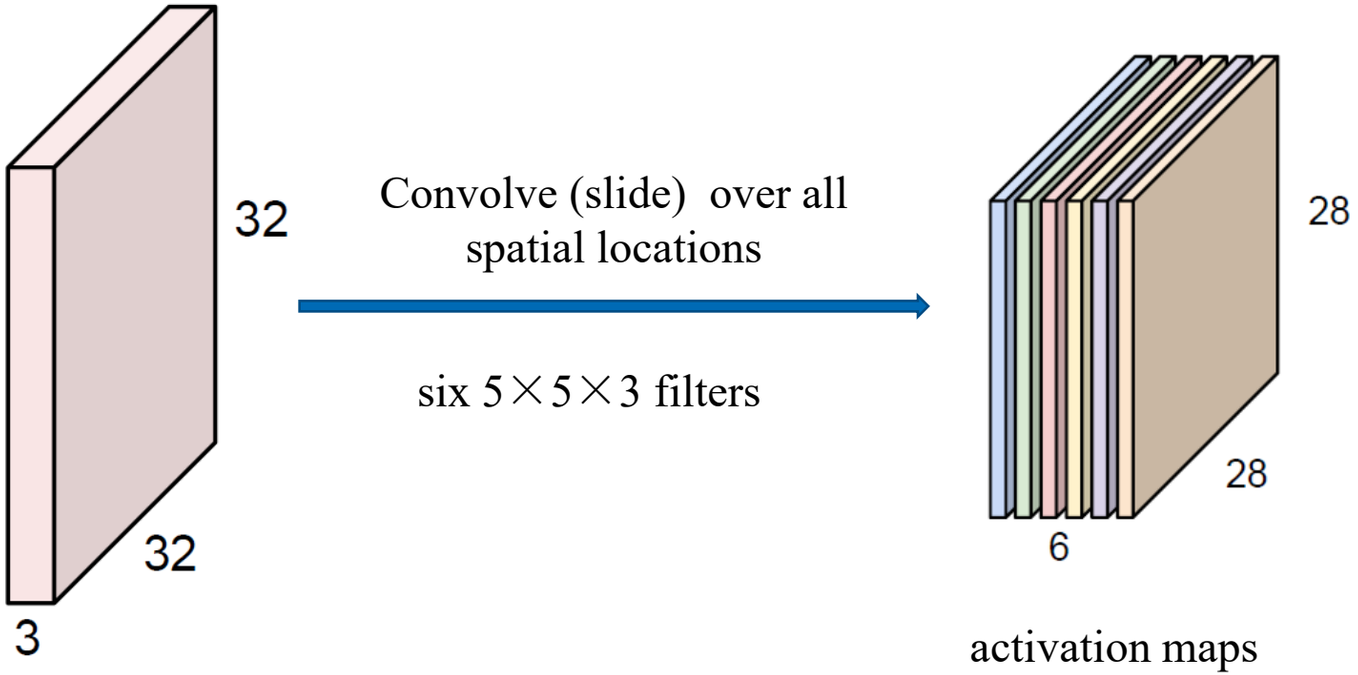


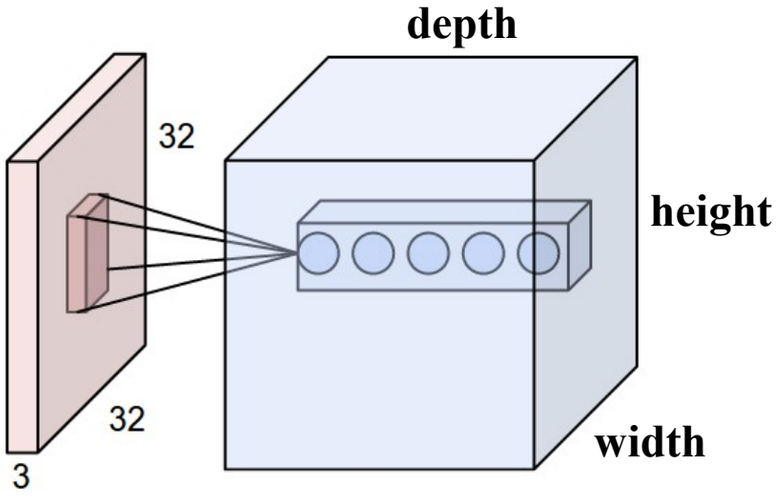
Image: 32×32×3

For example, if we had 6 5×5×3 filters, we'll get 6 separate activation maps.

We stack these up to get a new "image" of size 28×28×6.

为什么用卷积？

卷积层



Layer Dimensions:

$$h \times w \times d$$

where h and w are spatial dimensions

d (depth) = number of filters

Stride:

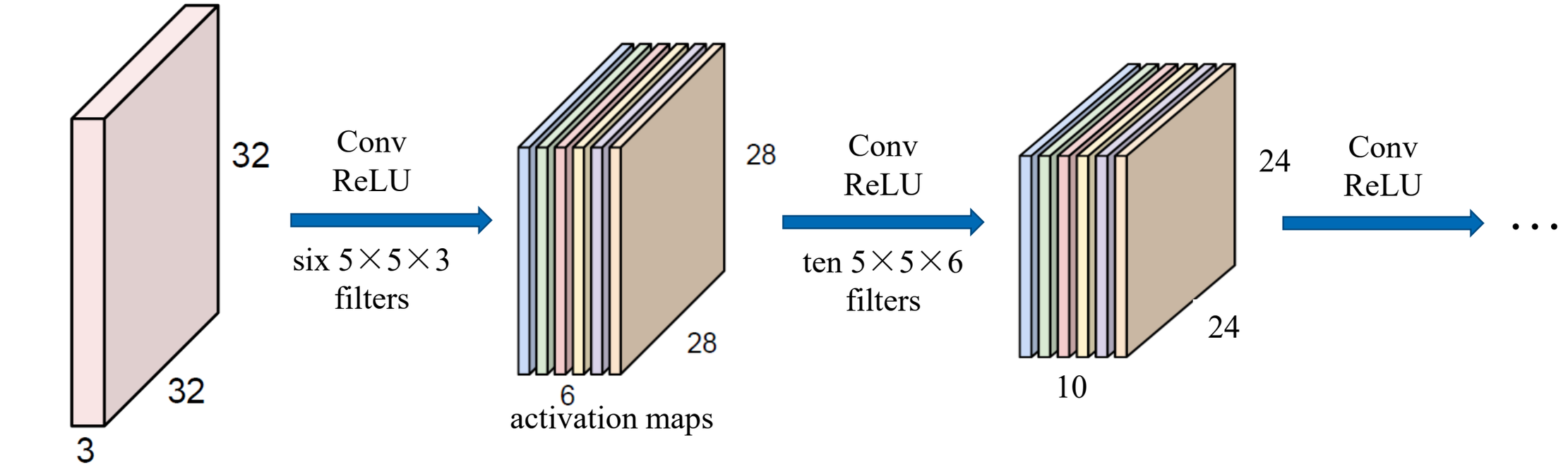
Filter step size

Receptive Field:

Locations in input image that a node is path connected to

为什么用卷积？

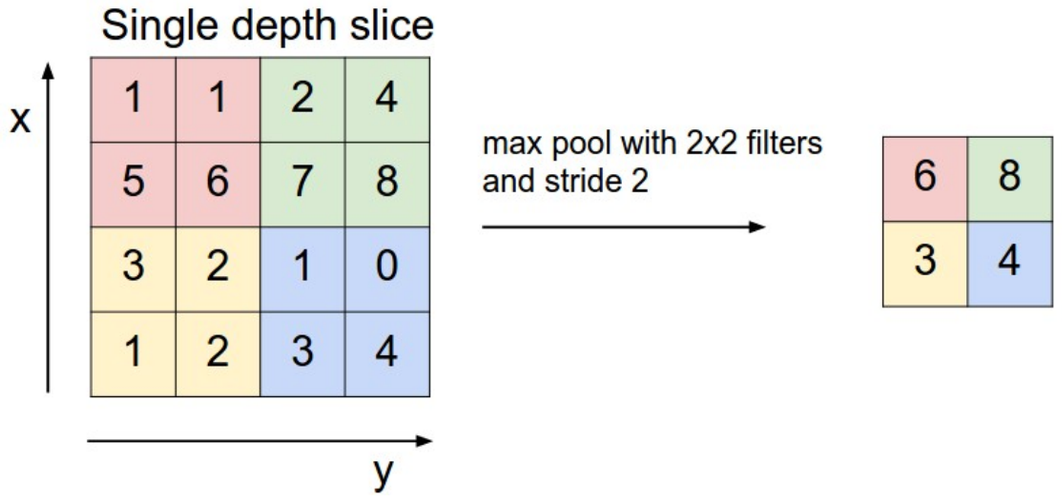
ConvNet is a sequence of Convolution layers, interspersed with activation functions.

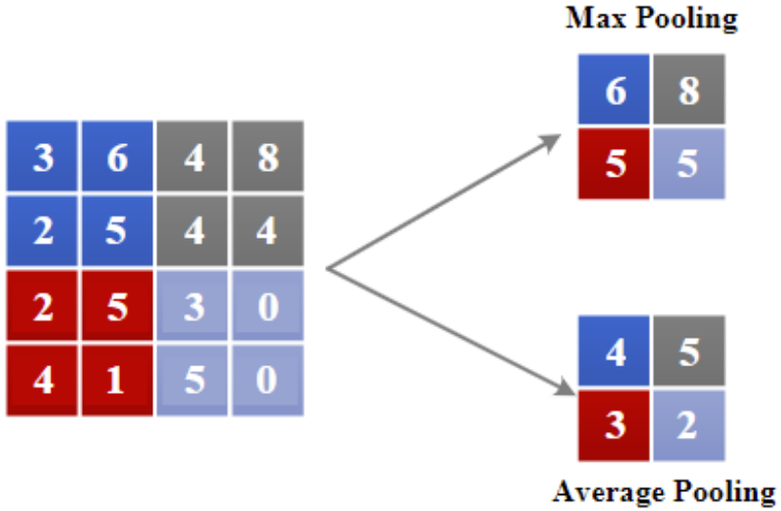


为什么用卷积？

池化（Pooling）

1) Reduced dimensionality
2) Spatial invariance





How else can we downsample and preserve spatial invariance?

为什么用卷积？

Property 1

- Some patterns are much smaller than the whole image

Property 2

- The same patterns appear in different regions.

Property 3

- Subsampling the pixels will not change the object

Convolution

Max Pooling

Convolution

Max Pooling

Flatten

Can repeat many times

Any Questions?