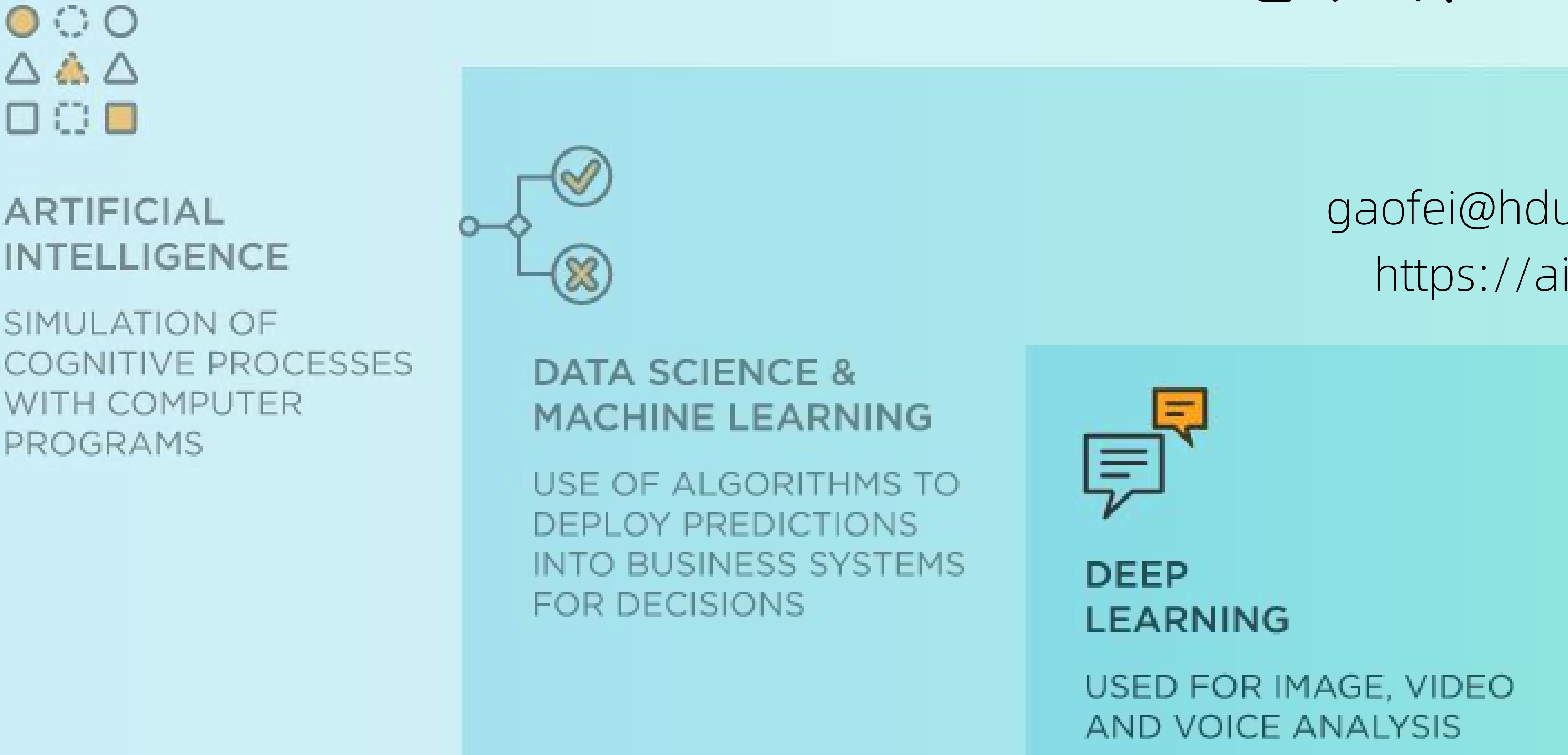
卷积神经网络

Fei Gao

gaofei@hdu.edu.cn

https://aiart.live/



让机器更好的理解和服务人类

图像信息

任务：理解图像内容

方法：卷积神经网络

序列信息

任务：理解语音/文字/视频

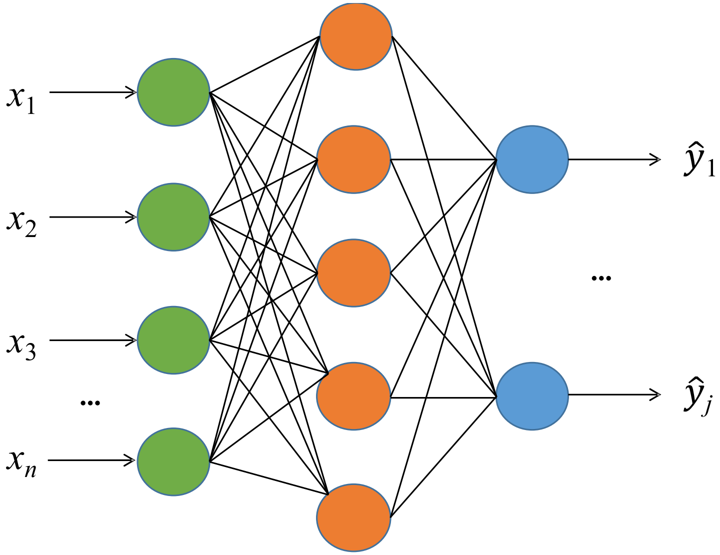
方法：循环神经网络

一个例子

- 计算机视觉

http://novel.ict.ac.cn/aics

输入图像

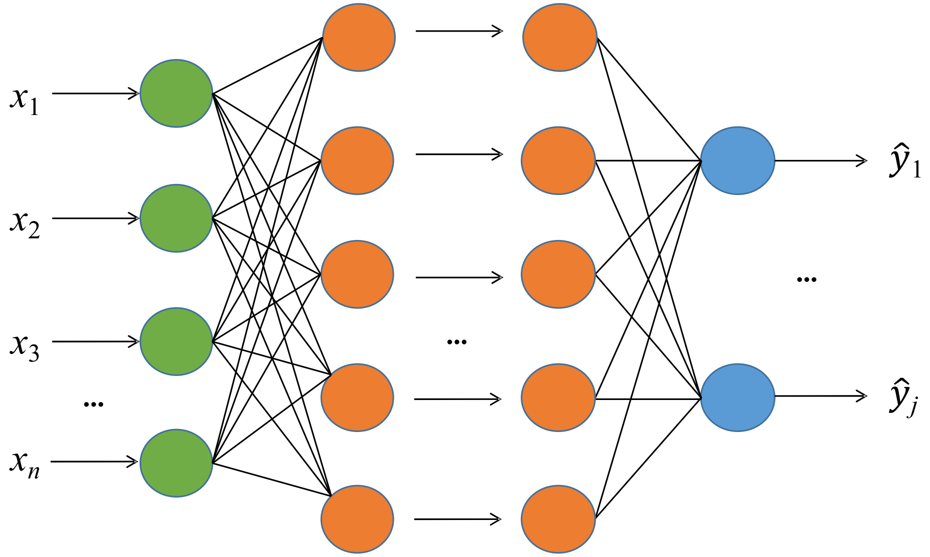


- 输入图像大小为 32x32，输入数据量为 32x32x3 = 3072
- 隐层神经元个数为 100，第一层权值数量为 3072 x100 = 307200

一个例子

- 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。

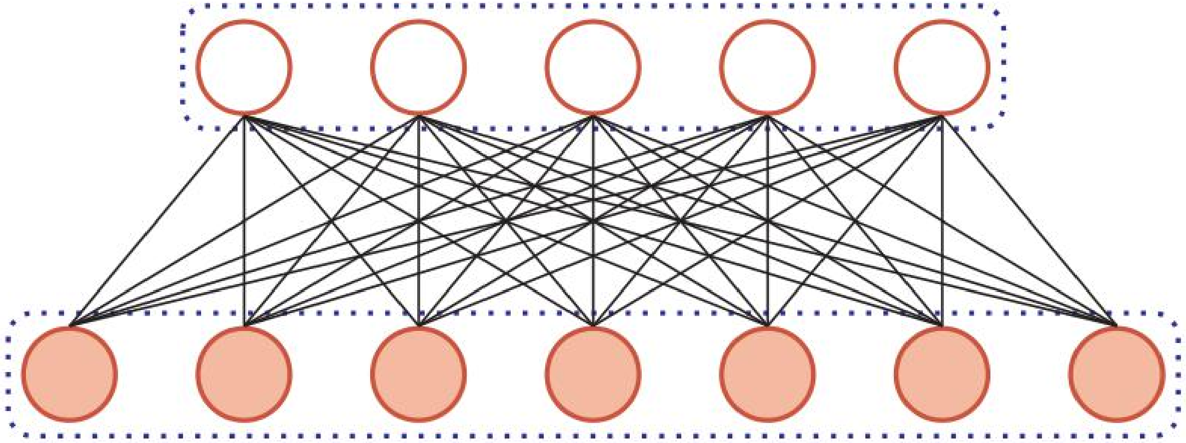
输入图像



- 输入图像大小为 1024x1024，第一层隐层神经元个数为 1000
- 第一层权重数量级为 10^9，过多的参数会导致**过拟合**
- **卷积神经网络**可以有效减少权重数量

全连接前馈神经网络

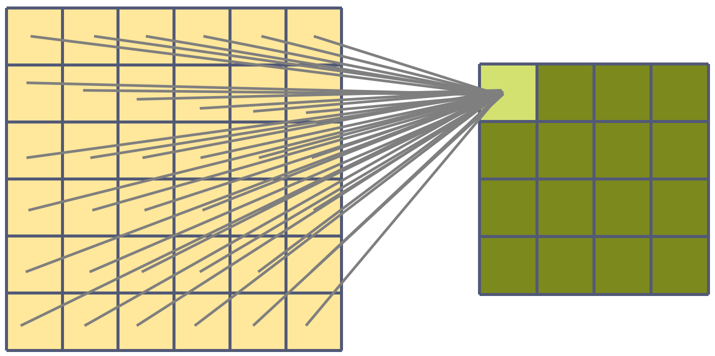
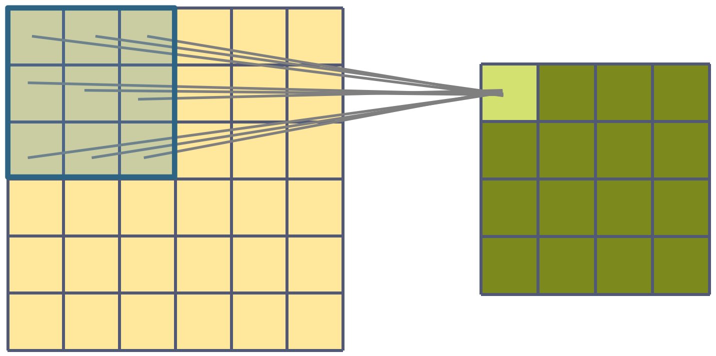
- 权重矩阵的参数非常多



- 局部不变性特征
  - 自然图像中的物体都具有局部不变性特征，比如尺度缩放、平移、旋转等操作不影响其语义信息。
  - 而全连接前馈网络很难提取这些局部不变特征。

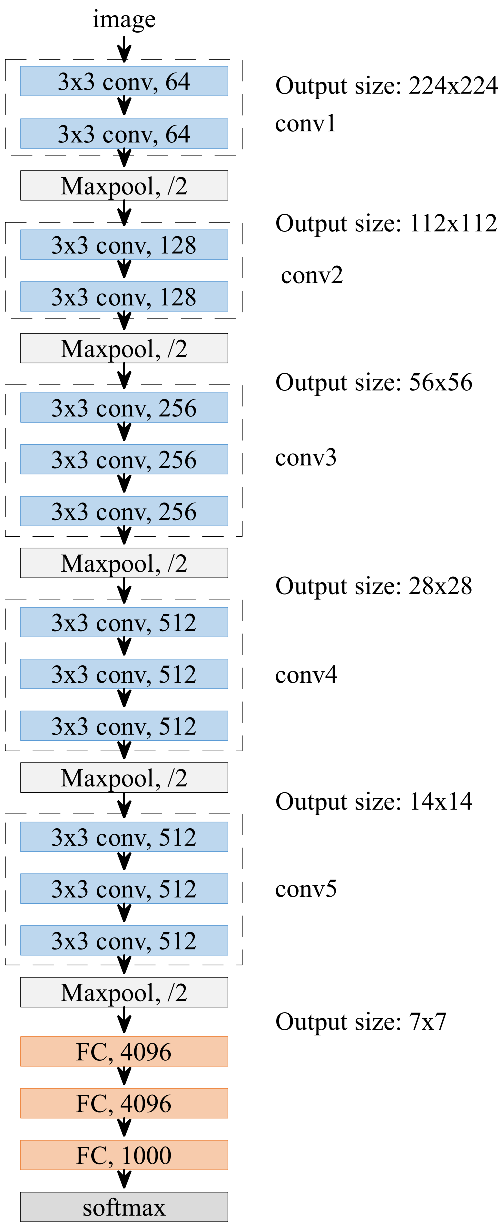
卷积神经网络（CNN）

http://novel.ict.ac.cn/aics

	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	$w_i \times h_i \times w_o \times h_o$	$f \times f$

CNN组成

- VGG16
  - 卷积层（conv）
  - 池化层（max pool）
  - 全连接层 （FC）
  - Softmax



http://novel.ict.ac.cn/aics

卷积层

http://novel.ict.ac.cn/aics

• 卷积层如何检测特征

- 检测复杂边缘
- 将权重作为参数，在训练中学习。

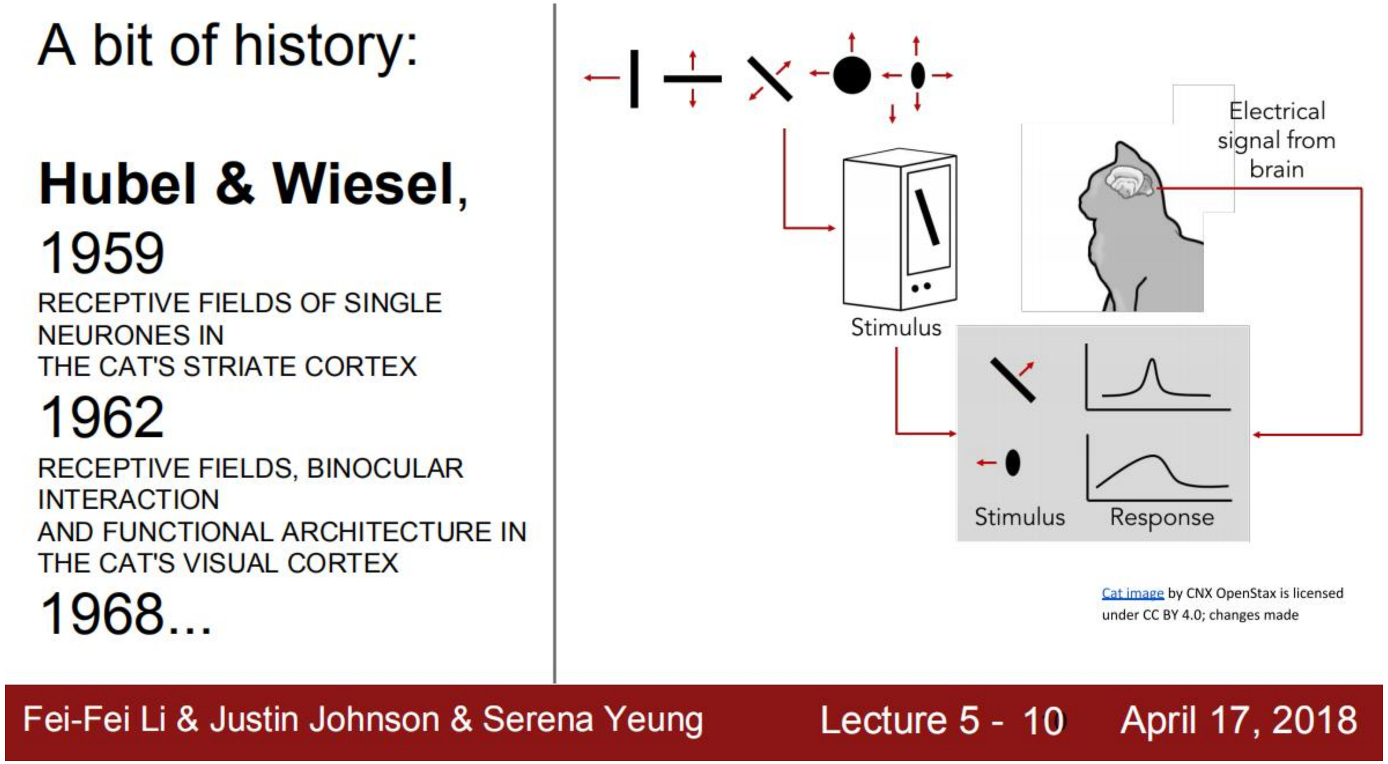
w0	w1	w2
w3	w4	w5
w6	w7	w8

filter/kernel

> ➢ 卷积神经网络的两个重要特征：<u>**局部连接、权重共享**</u>
>
> 可有效减少权重参数，避免过拟合，为增加卷积层数提供可能。

卷积神经网络

- 卷积神经网络
  - 生物学上**局部感受野**（Receptive Field）
- 卷积神经网络有两个结构上的特性：
  - 局部连接
  - 权重共享



卷积神经网络

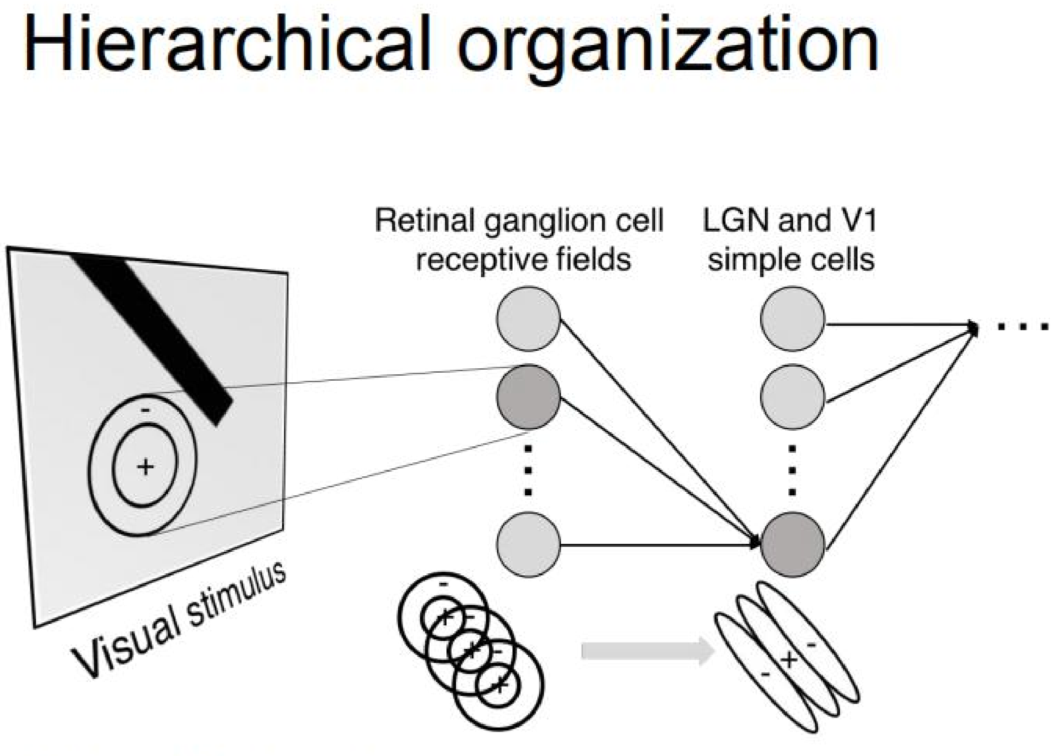
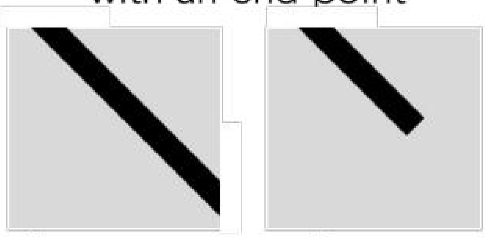


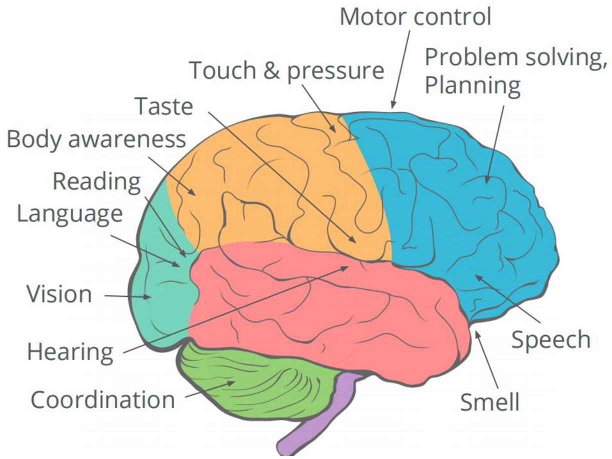
Illustration of hierarchical organization in early visual pathways by Lane McIntosh, copyright CS231n 2017

Simple cells:
Response to light orientation

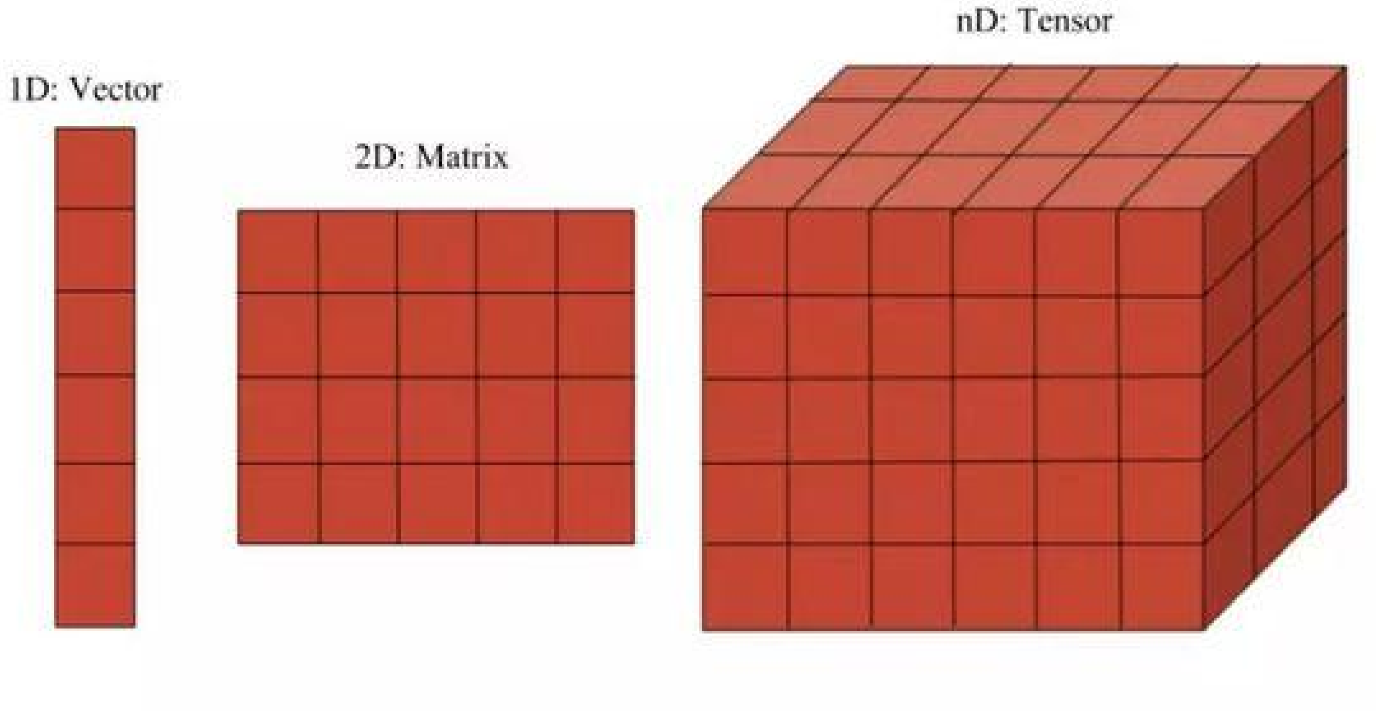
Complex cells:
Response to light orientation and movement

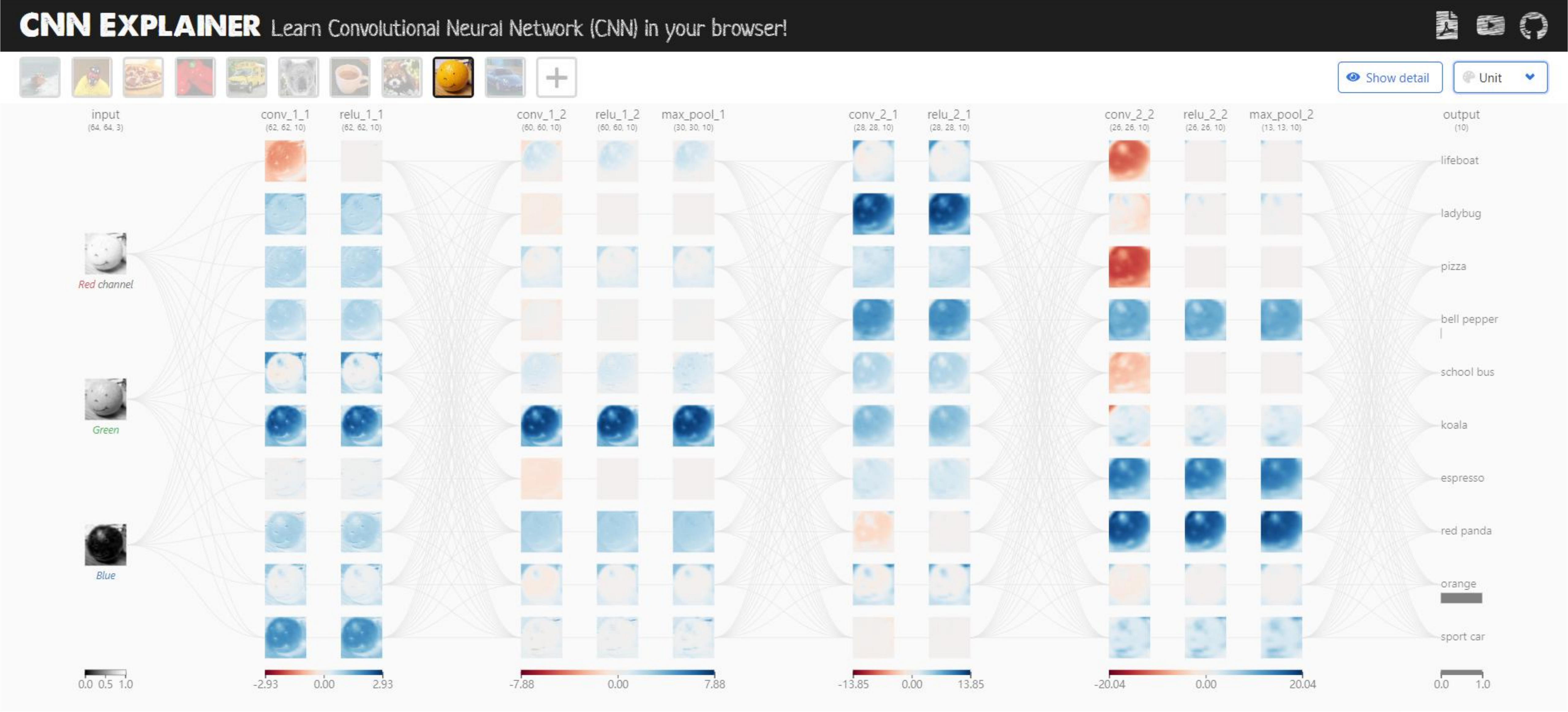
Hypercomplex cells:
response to movement with an end point





- 数据表示：矩阵，张量（Tensor）





https://poloclub.github.io/cnn-explainer/

卷积层

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xt，其信息的衰减率为wk，即在k−1个时间步长后，信息为原来的wk 倍
    - 假设w1 = 1,w2 = 1/2,w3 = 1/4
- 时刻t收到的信号yt 为当前时刻产生的信息和以前时刻延迟信息的叠加

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

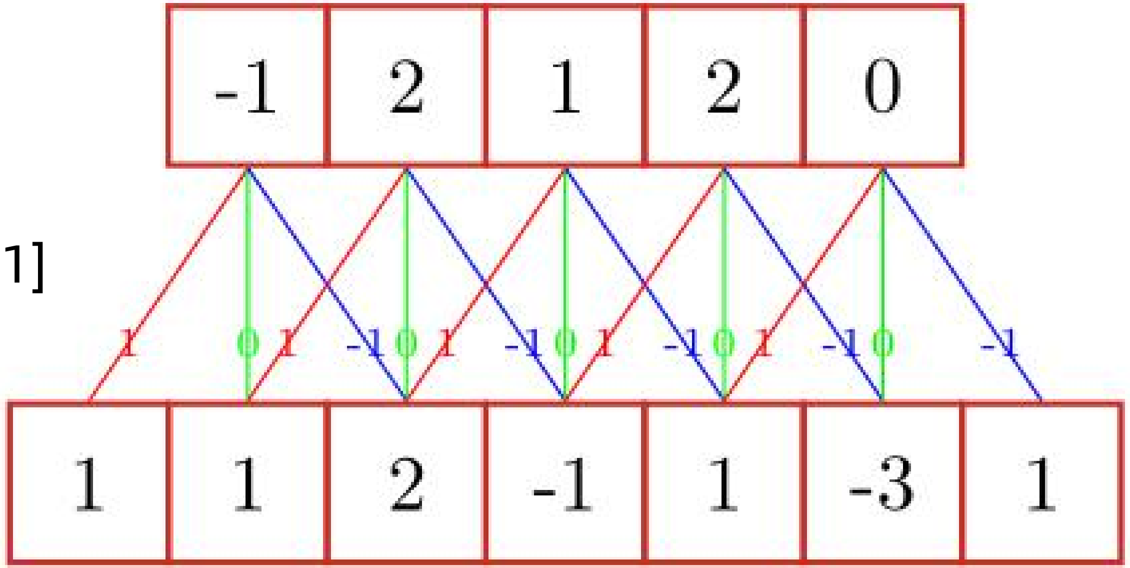
滤波器（filter）或卷积核（convolution kernel）

卷积

- 卷积经常用在信号处理中，用于计算信号的延迟累积。
  - 给定一个收入信号序列 x 和滤波器 w 卷积的输出为：

$$y_t = \sum_{k=1}^{m} w_k x_{t-k+1}$$

Filter: [-1,0,1]

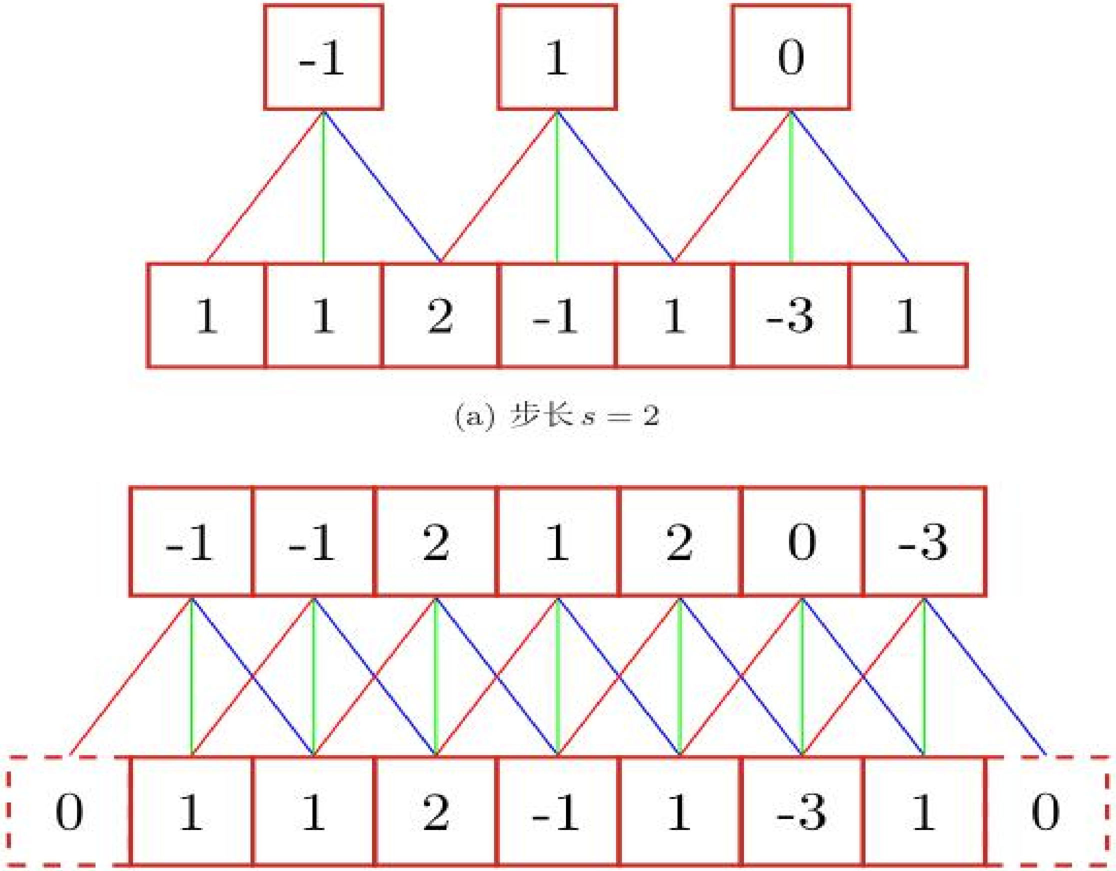


卷积扩展

- 引入滤波器的滑动步长s和零填充p

窄卷积

等宽卷积



(a) 步长 $s = 2$

(b) 零填充 $p = 1$

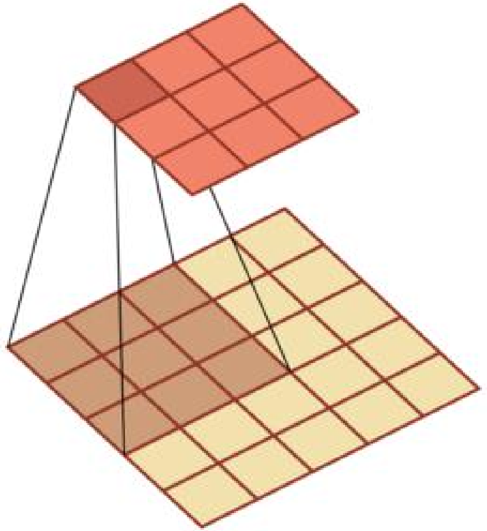
两维卷积

- 在图像处理中，图像是以二维矩阵的形式输入到神经网络中，因此我们需要二维卷积。

$$\mathbf{y} = \mathbf{w} \otimes \mathbf{x},$$

$$y_{ij} = \sum_{u=1}^{m} \sum_{v=1}^{n} w_{uv} \cdot x_{i-u+1,j-v+1}.$$

$$\begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1_{\times -1} & 1_{\times 0} & 1_{\times 0} \\ \hline -1 & 0 & -3_{\times 0} & 0_{\times 0} & 1_{\times 0} \\ \hline 2 & 1 & 1_{\times 0} & -1_{\times 0} & 0_{\times 1} \\ \hline 0 & -1 & 1 & 2 & 1 \\ \hline 1 & 2 & 1 & 1 & 1 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & -2 & -1 \\ \hline 2 & 2 & 4 \\ \hline -1 & 0 & 0 \\ \hline \end{array}$$



卷积层

- 数学：
  - 卷积运算

$$y(n)=\sum_{i=-\infty}^{\infty}x(i)h(n-i)=x(n)*h(n)$$

（“*”表示卷积）

  - 神经网络：实际为计算矩阵内积(相关系数)；

2 1	3 0	1 1	5	2	3
7 4	4 -3	5 2	2	3	1
3 3	9 0	6 -1	0	4	2
0	6	4	7	1	2
4	1	0	8	0	6
7	0	2	1	6	3

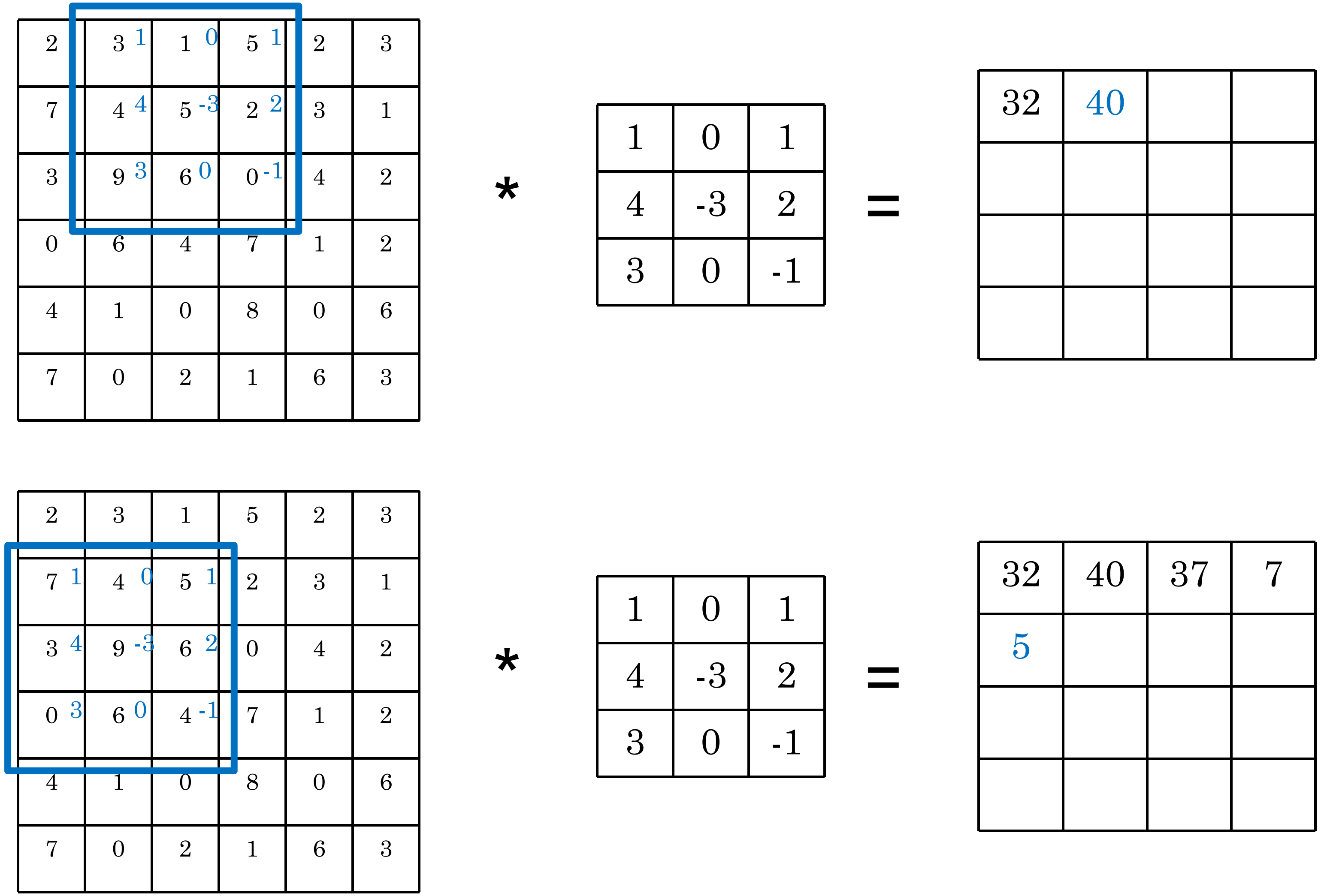
*

1	0	1
4	-3	2
3	0	-1

=

32			

卷积层



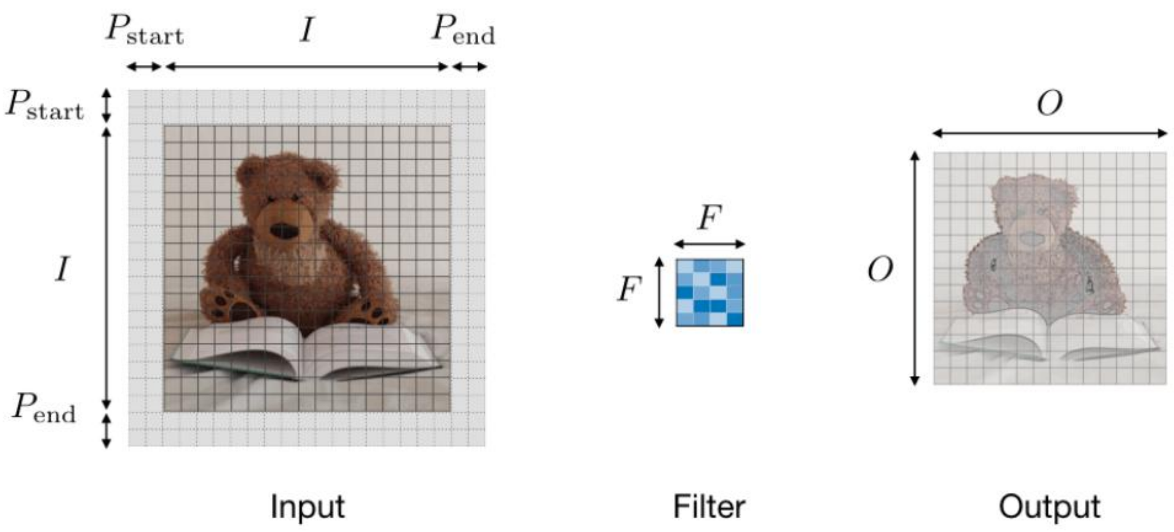
2 3 1 1 0 5 1 2 3
7 4 4 5 -3 2 2 3 1
3 9 3 6 0 0 -1 4 2
0 6 4 7 1 2
4 1 0 8 0 6
7 0 2 1 6 3
*
1 0 1
4 -3 2
3 0 -1
=
32 40
2 3 1 5 2 3
7 1 4 0 5 1 2 3 1
3 4 9 -3 6 2 0 4 2
0 3 6 0 4 -1 7 1 2
4 1 0 8 0 6
7 0 2 1 6 3
*
1 0 1
4 -3 2
3 0 -1
=
32 40 37 7
5

卷积层

- 离散卷积的边缘效应

Parameter compatibility in convolution layer — By noting I the length of the input volume size, F the length of the filter, P the amount of zero padding, S the stride, then the output size O of the feature map along that dimension is given by:

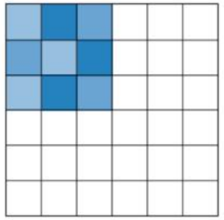
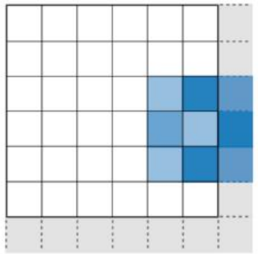
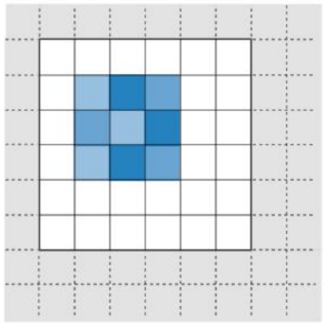
$$\boxed{O = \frac{I - F + P_{\text{start}} + P_{\text{end}}}{S} + 1}$$



Remark: often times, $P_{\text{start}} = P_{\text{end}} \triangleq P$, *in which case we can replace* $P_{\text{start}} + P_{\text{end}}$ *by* $2P$ *in the formula above.*

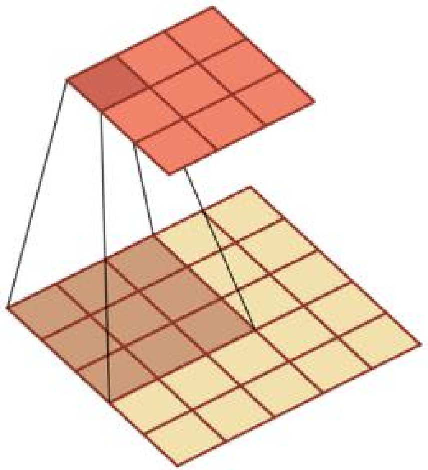
卷积层

- 离散卷积的边缘效应
  - Zero-Padding，edge-padding，reflect-padding

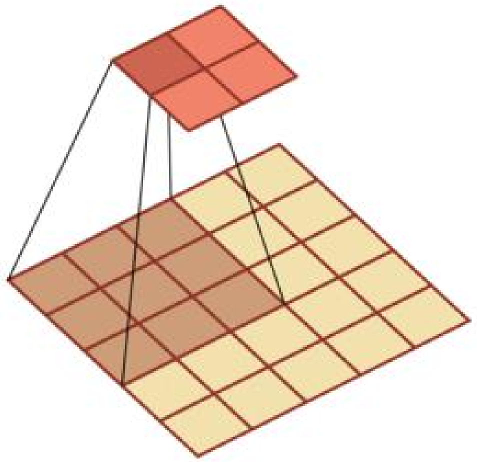
Mode	Valid	Same	Full
Value	$P = 0$	$P_{\text{start}} = \left\lfloor \frac{S\lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rfloor$ $P_{\text{end}} = \left\lceil \frac{S\lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rceil$	$P_{\text{start}} \in [\![0, F-1]\!]$ $P_{\text{end}} = F - 1$
Illustration			
Purpose	• No padding • Drops last convolution if dimensions do not match	• Padding such that feature map size has size $\left\lceil \frac{I}{S} \right\rceil$ • Output size is mathematically convenient • Also called 'half' padding	• Maximum padding such that end convolutions are applied on the limits of the input • Filter 'sees' the input end-to-end

二维卷积

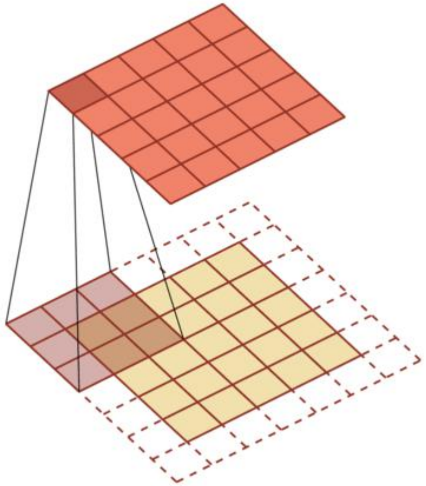
步长1，零填充0



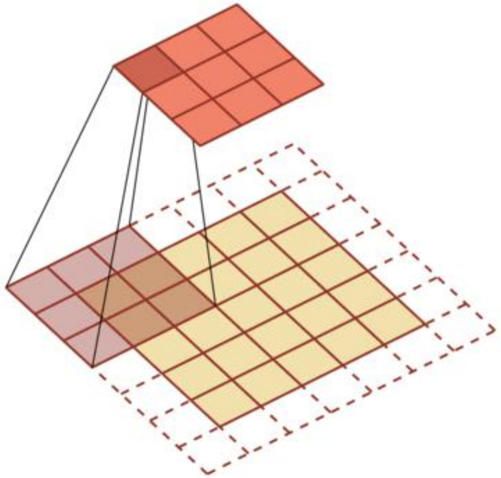
步长2，零填充0



步长1，零填充1

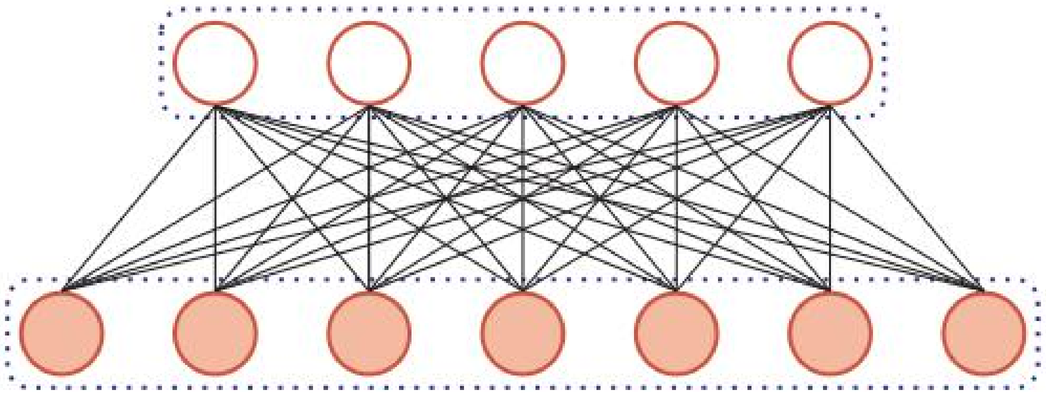


步长2，零填充1

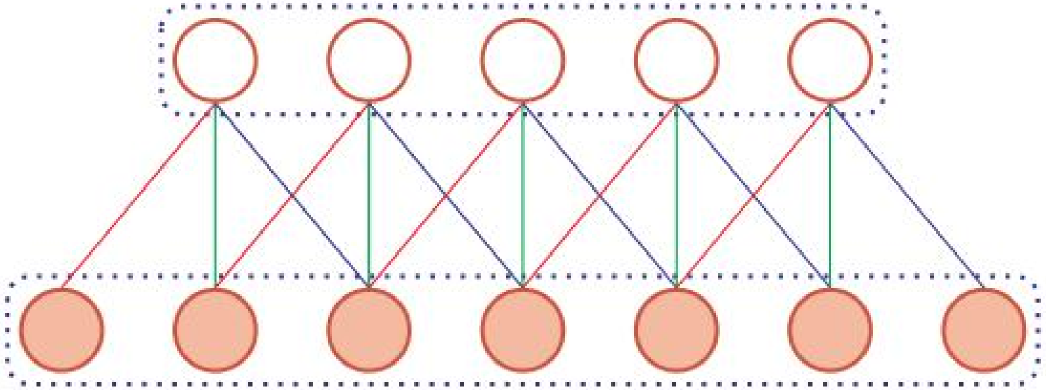


卷积神经网络

- 用卷积层代替全连接层



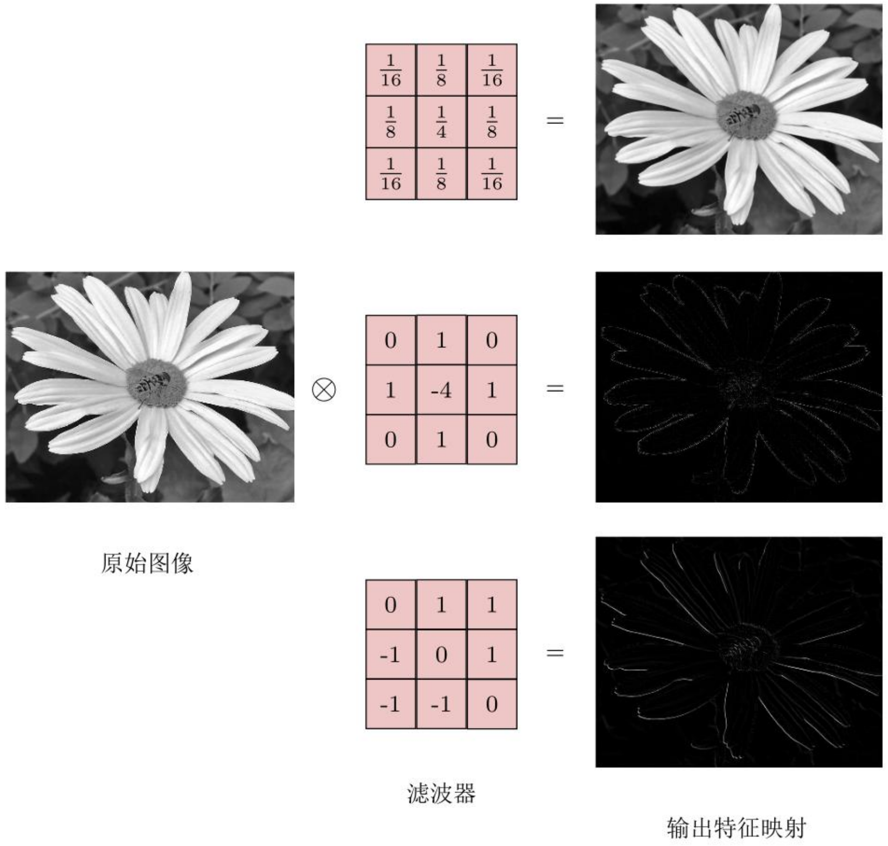
(a) 全连接层



(b) 卷积层

卷积作为特征提取器

- 特征映射（Feature Map）：图像经过卷积后得到的特征。
  - 卷积核看成一个特征提取器



卷积层

- 卷积层如何检测特征

  - 检测垂直边缘

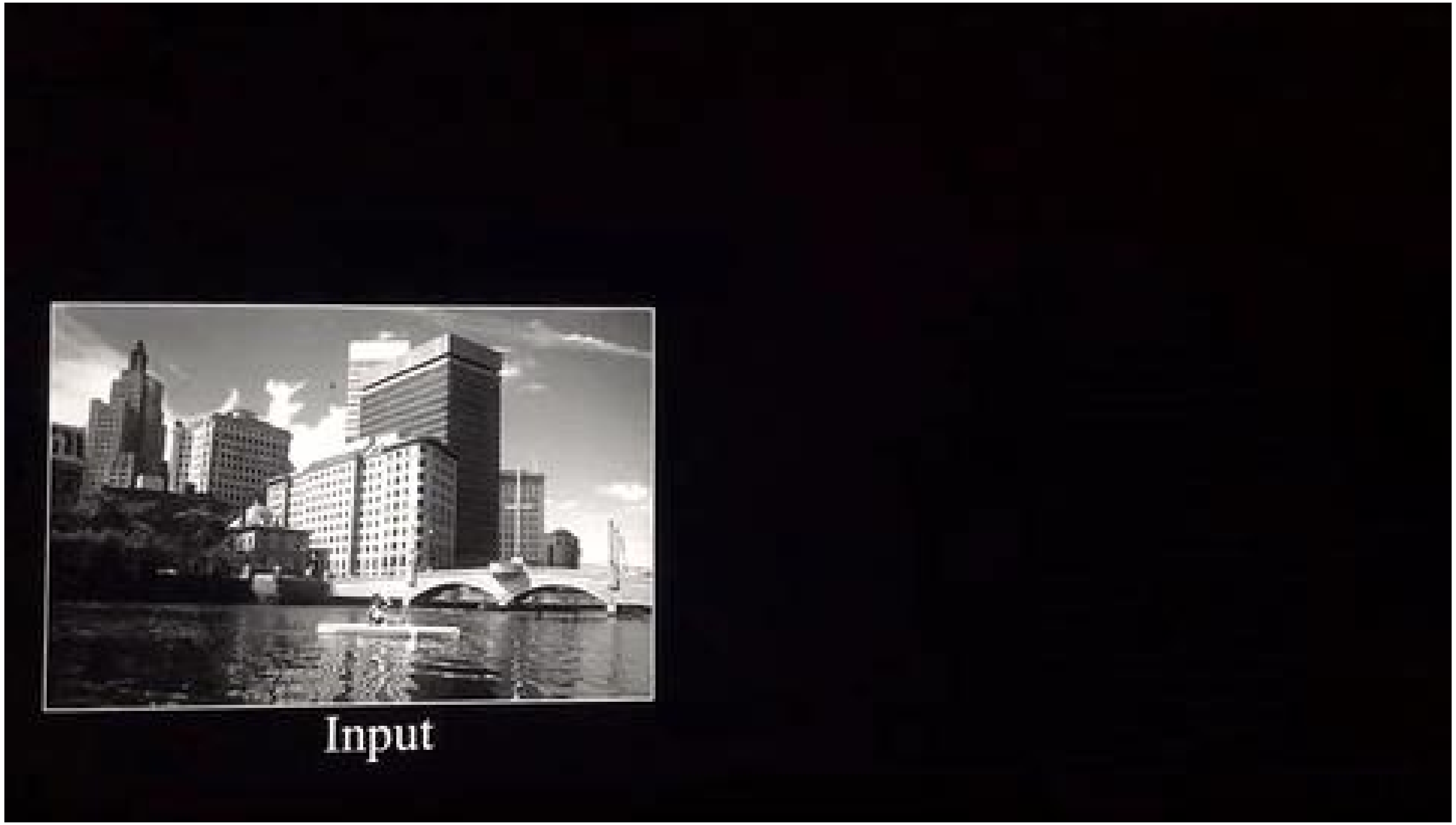
$$
\begin{bmatrix}
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0
\end{bmatrix}
*
\begin{bmatrix}
1 & 0 & -1 \\
1 & 0 & -1 \\
1 & 0 & -1
\end{bmatrix}
=
\begin{bmatrix}
0 & 30 & 30 & 0 \\
0 & 30 & 30 & 0 \\
0 & 30 & 30 & 0 \\
0 & 30 & 30 & 0
\end{bmatrix}
$$

  - 检测对角线边缘

$$
\begin{bmatrix}
10 & 10 & 10 & 10 & 10 & 0 \\
10 & 10 & 10 & 10 & 0 & 0 \\
10 & 10 & 10 & 0 & 0 & 0 \\
10 & 10 & 0 & 0 & 0 & 0 \\
10 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
*
\begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & -1 \\
0 & -1 & -1
\end{bmatrix}
=
\begin{bmatrix}
0 & 10 & 30 & 30 \\
10 & 30 & 30 & 10 \\
30 & 30 & 10 & 0 \\
30 & 10 & 0 & 0
\end{bmatrix}
$$

卷积层

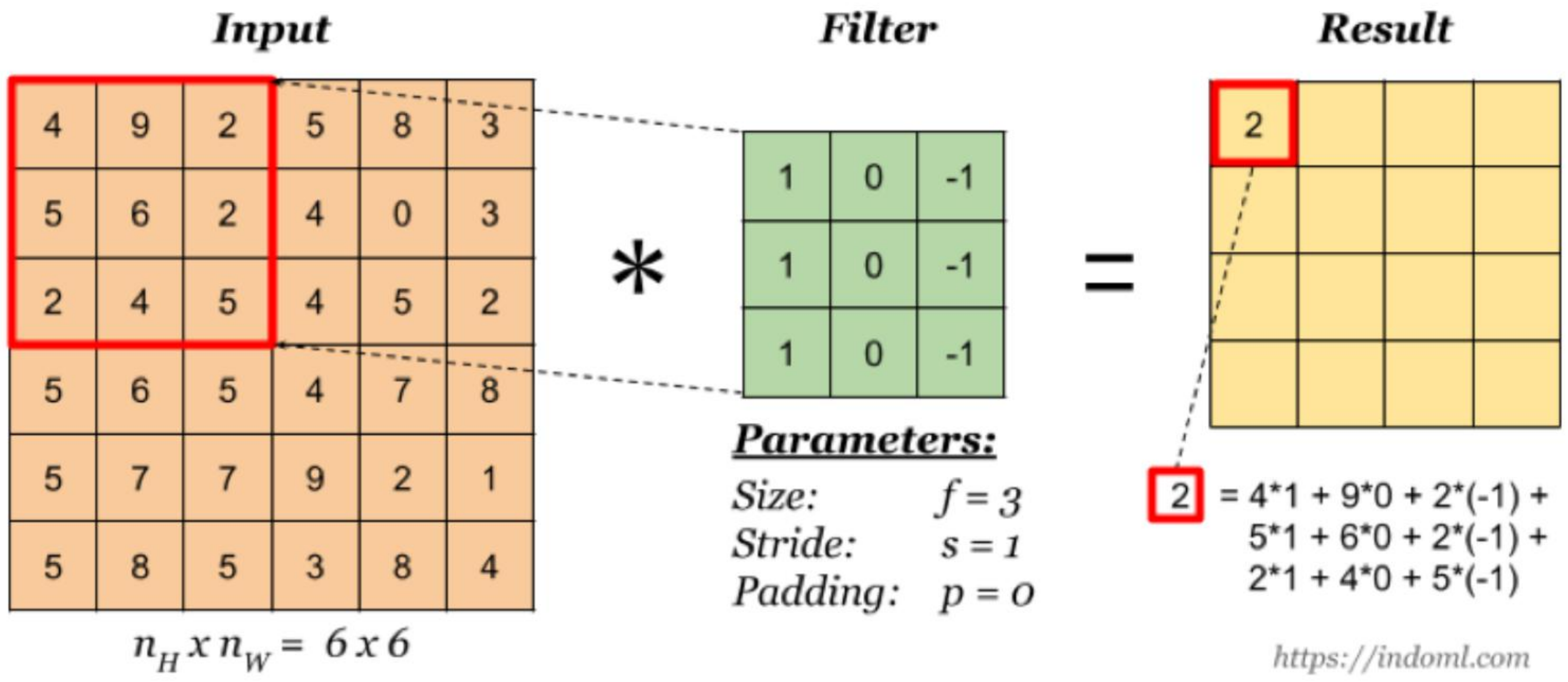
- 基本操作单元：卷积层



卷积层

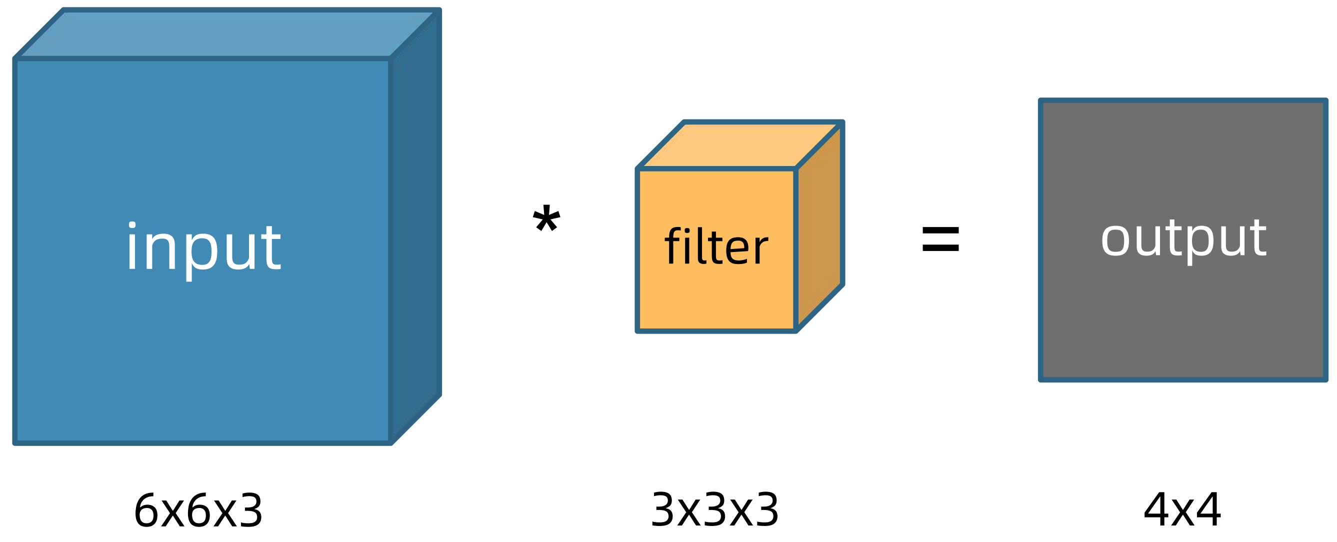
- 基本操作单元：卷积层

卷积核 (或称滤波器，filter/kernel)

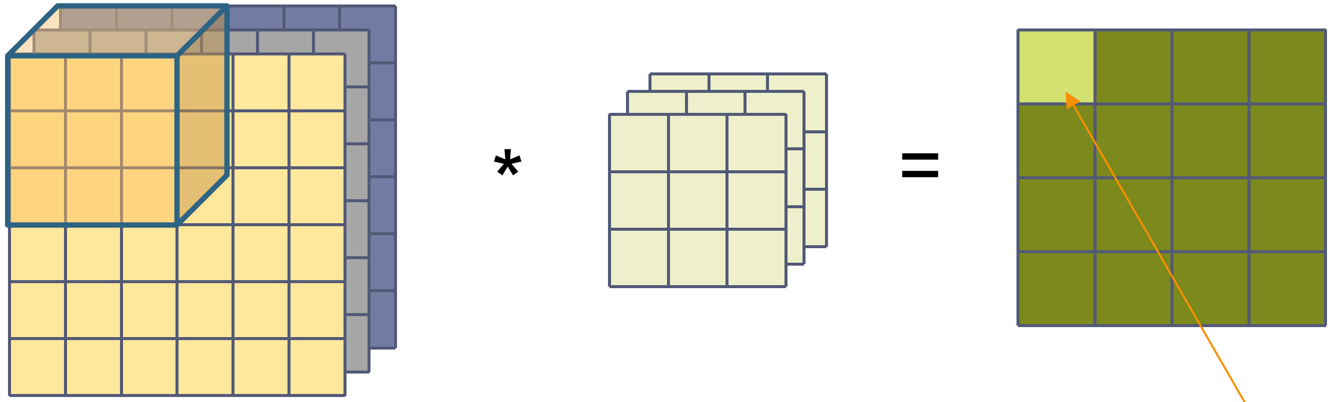


卷积层

- 多输入特征图单输出特征图卷积运算



卷积层



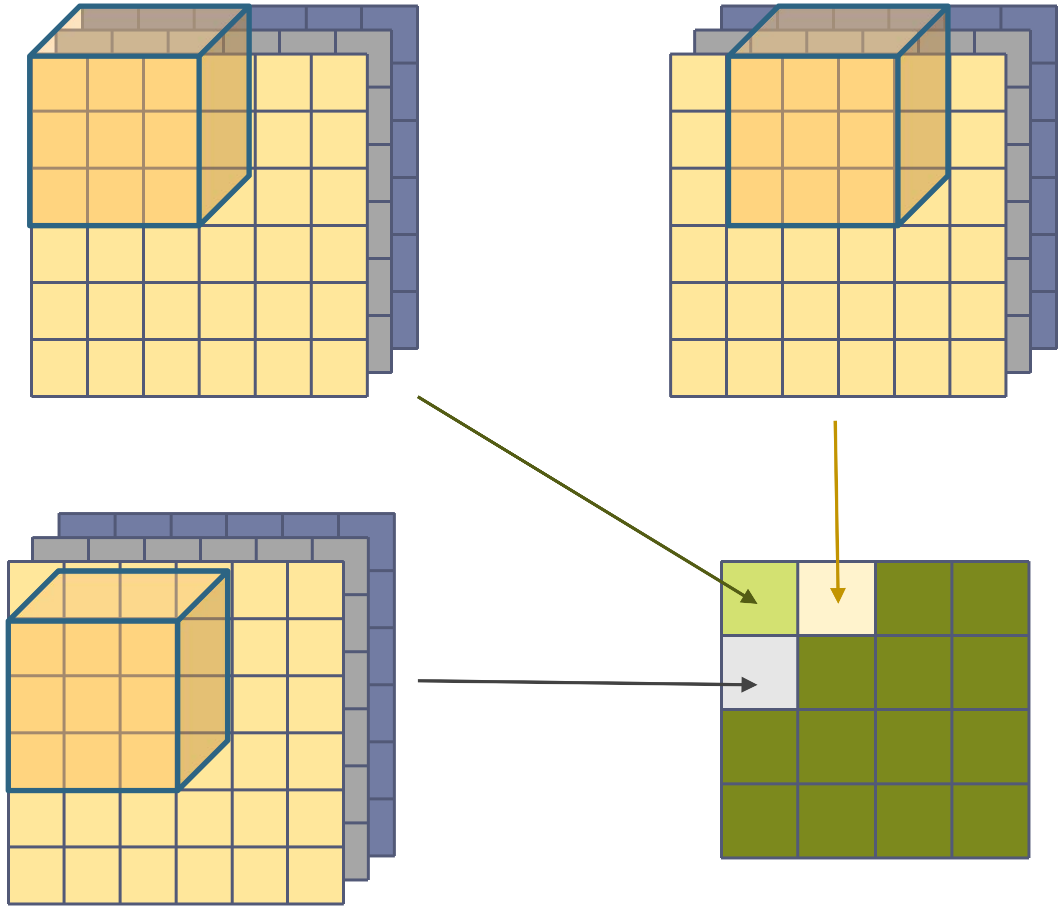
C = 0: $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 1 & 2 \end{bmatrix}$ $\begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$

C = 1: $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix}$ * $\begin{bmatrix} 1 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & 0 & 1 \end{bmatrix}$

C = 2: $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$ $\begin{bmatrix} 1 & -1 & -1 \\ -1 & -1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$

$$= \begin{matrix} 2-2-1+2 \\ + \\ 0-2+0+2 \\ + \\ (-1)+0+0+2 \end{matrix} = 2$$

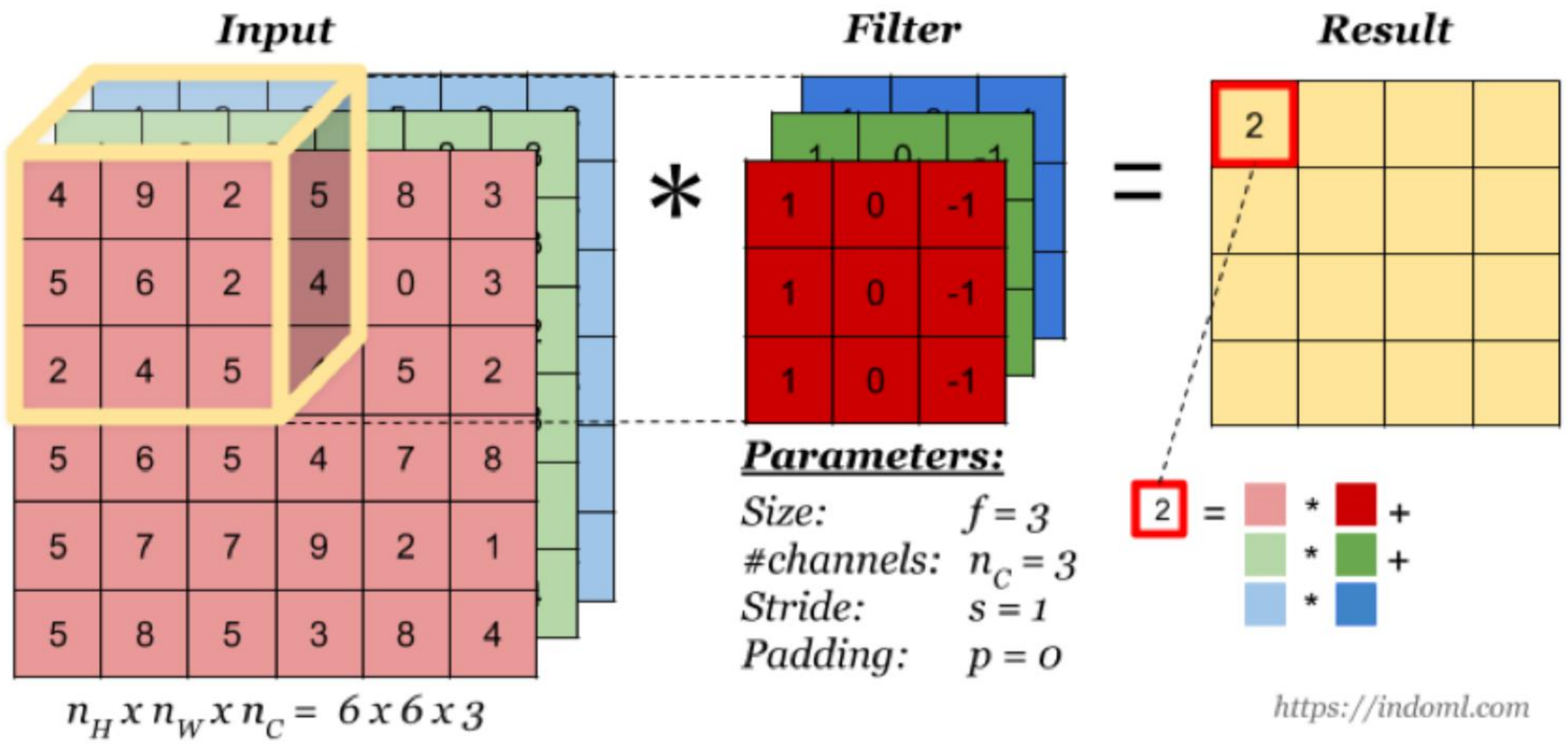
卷积层



卷积层

- 基本操作单元：卷积层

多通道卷积

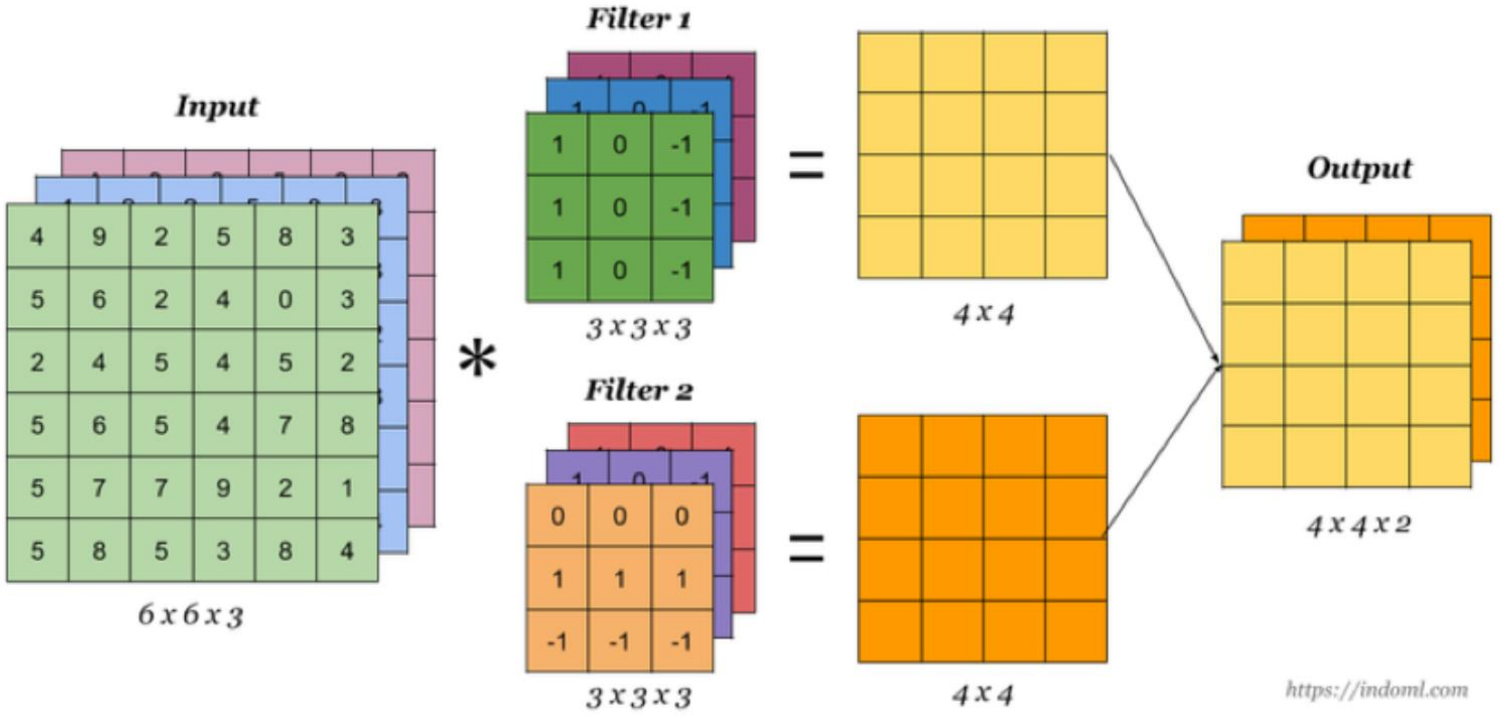


卷积层

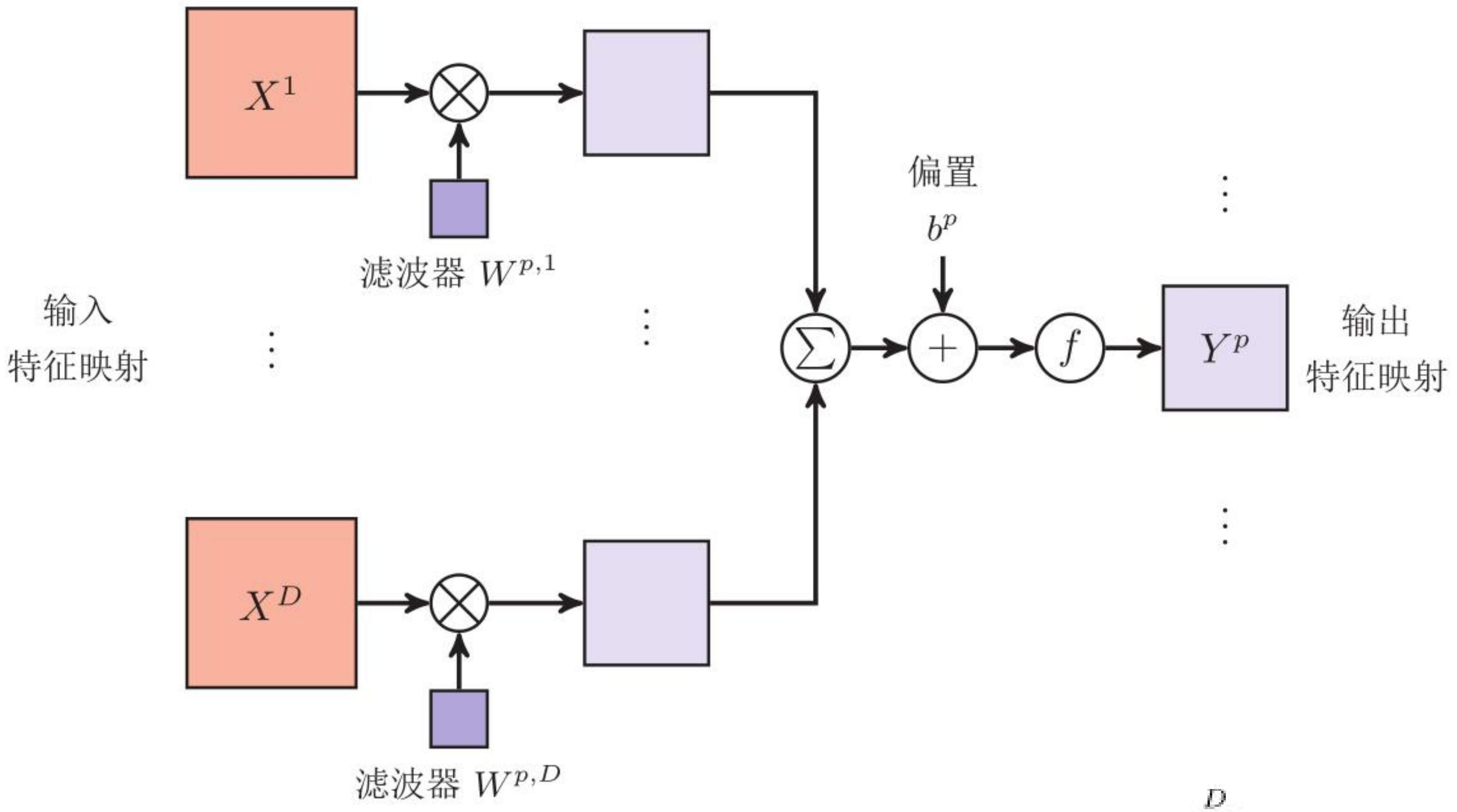
- 基本操作单元：卷积层

- *size = 3*
- *c_in = 3*
- *c_out = 2*
- *stride = 1*
- *padding = 0*

多卷积核



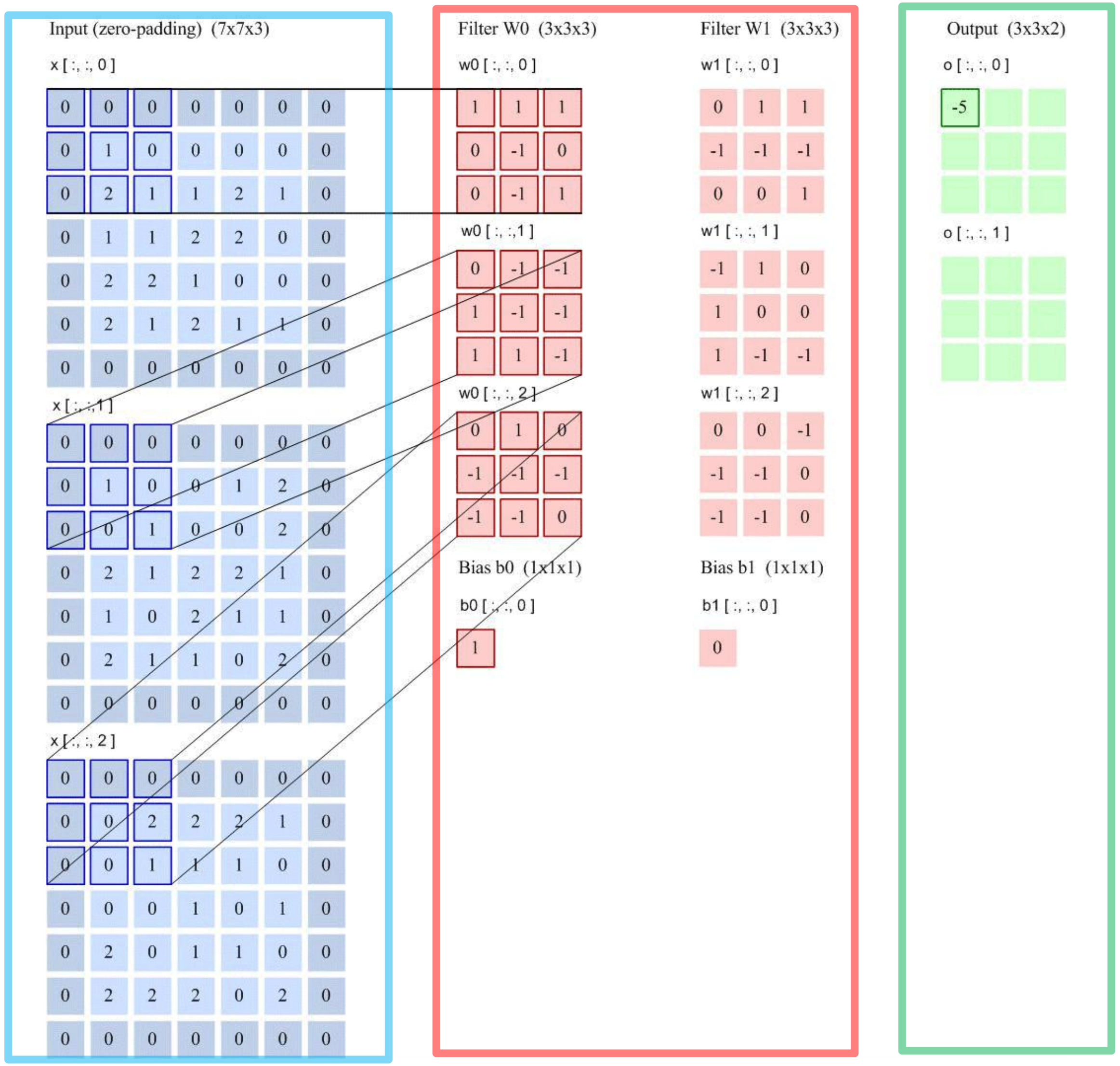
卷积层的映射关系



$$Z^p = \mathbf{W}^p \otimes \mathbf{X} + b^p = \sum_{d=1}^{D} W^{p,d} \otimes X^d + b^p,$$

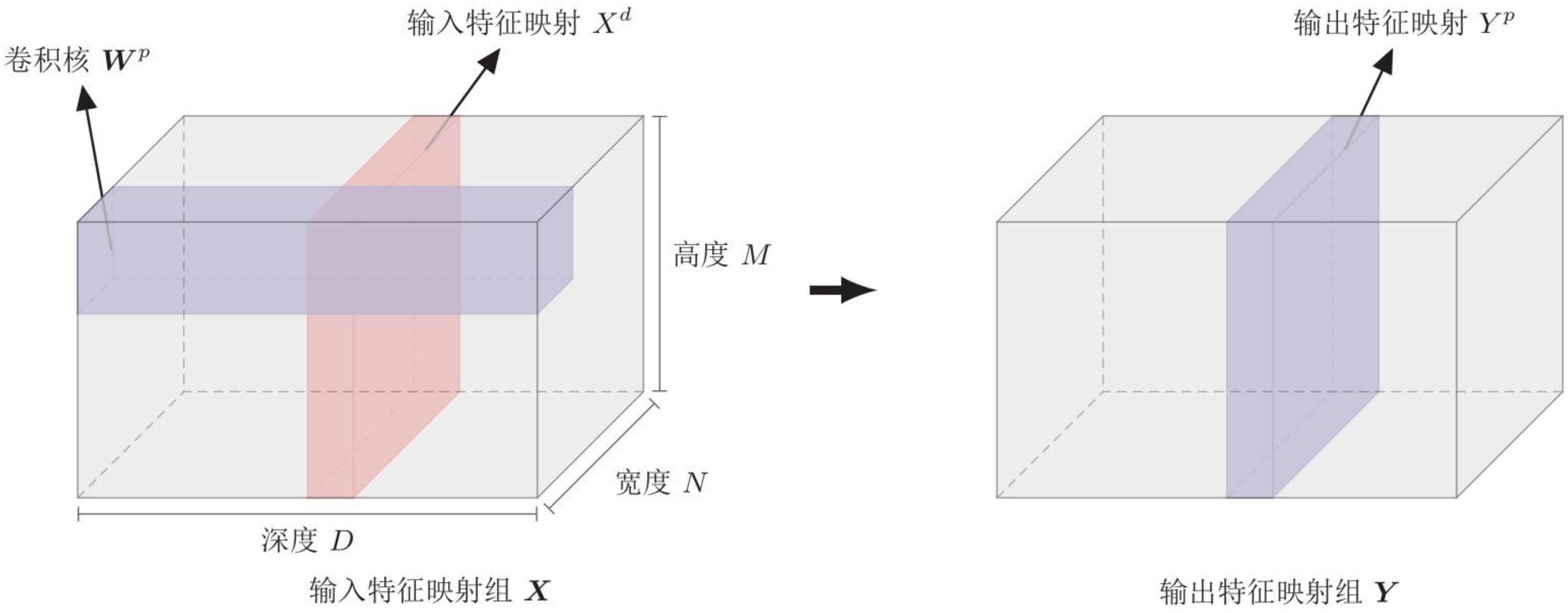
$$Y^p = f(Z^p).$$

步长2 filter个数3 3*3 填充



卷积层

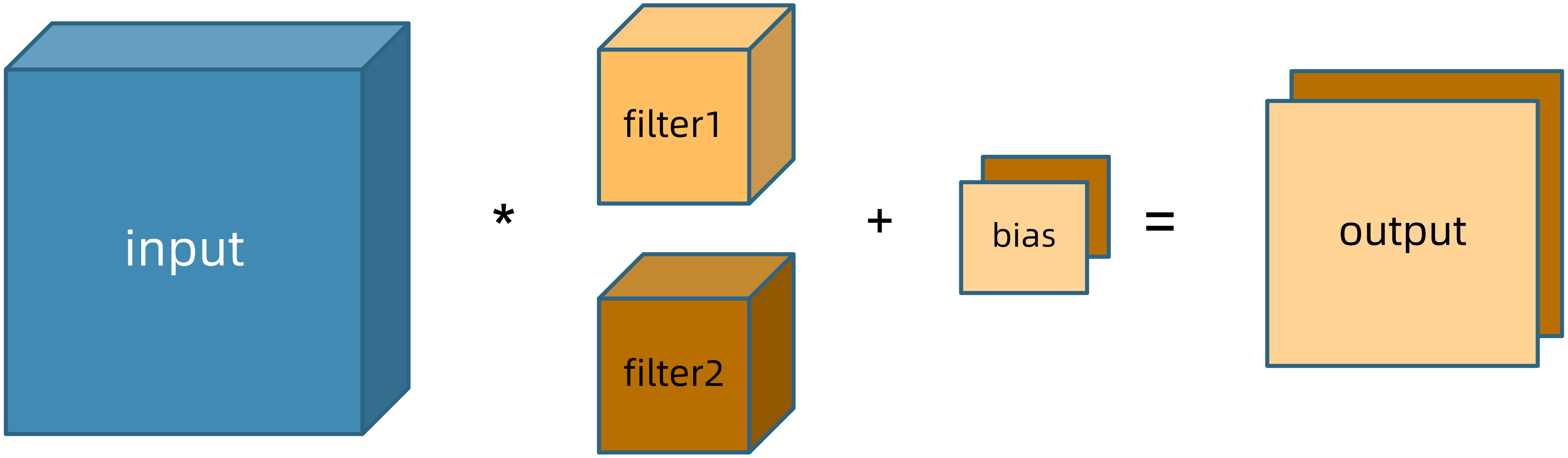
- 典型的卷积层为3维结构



$$Z^p = \mathbf{W}^p \otimes \mathbf{X} + b^p = \sum_{d=1}^{D} W^{p,d} \otimes X^d + b^p,$$

$$Y^p = f(Z^p).$$

总结：卷积层参数



$w_i \times h_i \times c_i$　　$c_o \times f \times f \times c_i$　　$1 \times 1 \times c_o$　　$w_o \times h_o \times c_o$

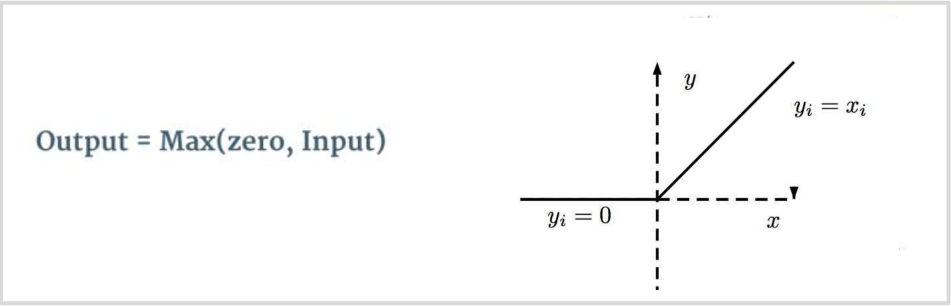
- stride
- pad
- $\text{output} = \left[\frac{w_i+2p-f}{s}+1\right] \times \left[\frac{h_i+2p-f}{s}+1\right]$

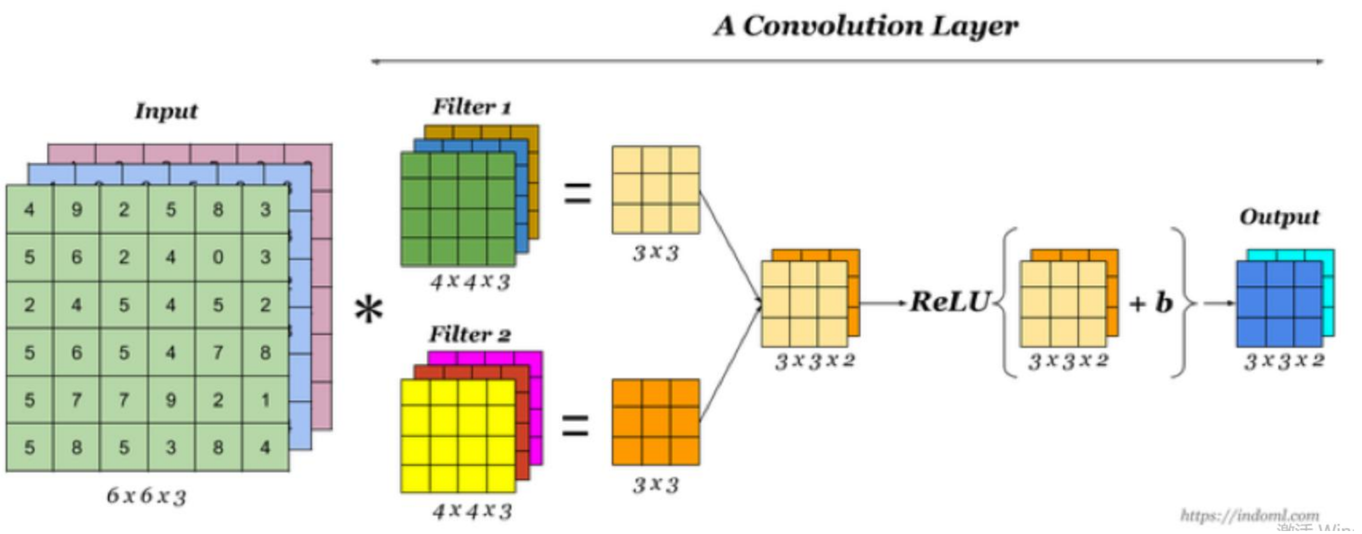
- filter：可训练
- bias：可训练，使分类器偏离激活函数原点，更灵活；
- activation

激活层

激活层

- 基本操作单元：激活层





激活层

- 基本操作单元：激活层

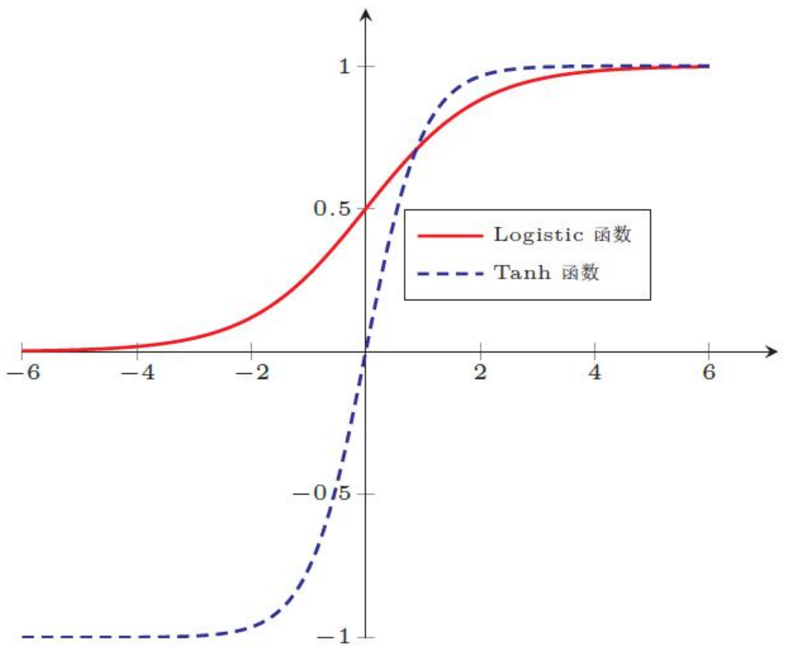


图 4.2 Logistic 函数和 Tanh 函数

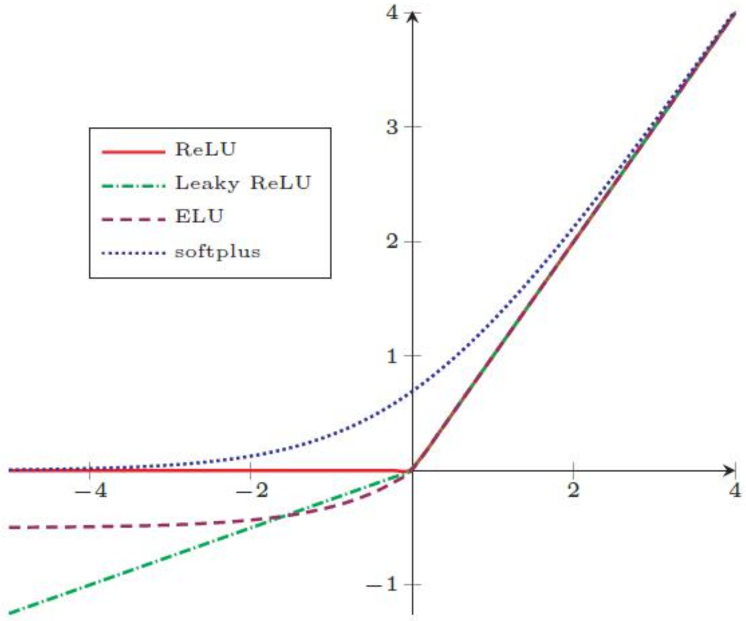
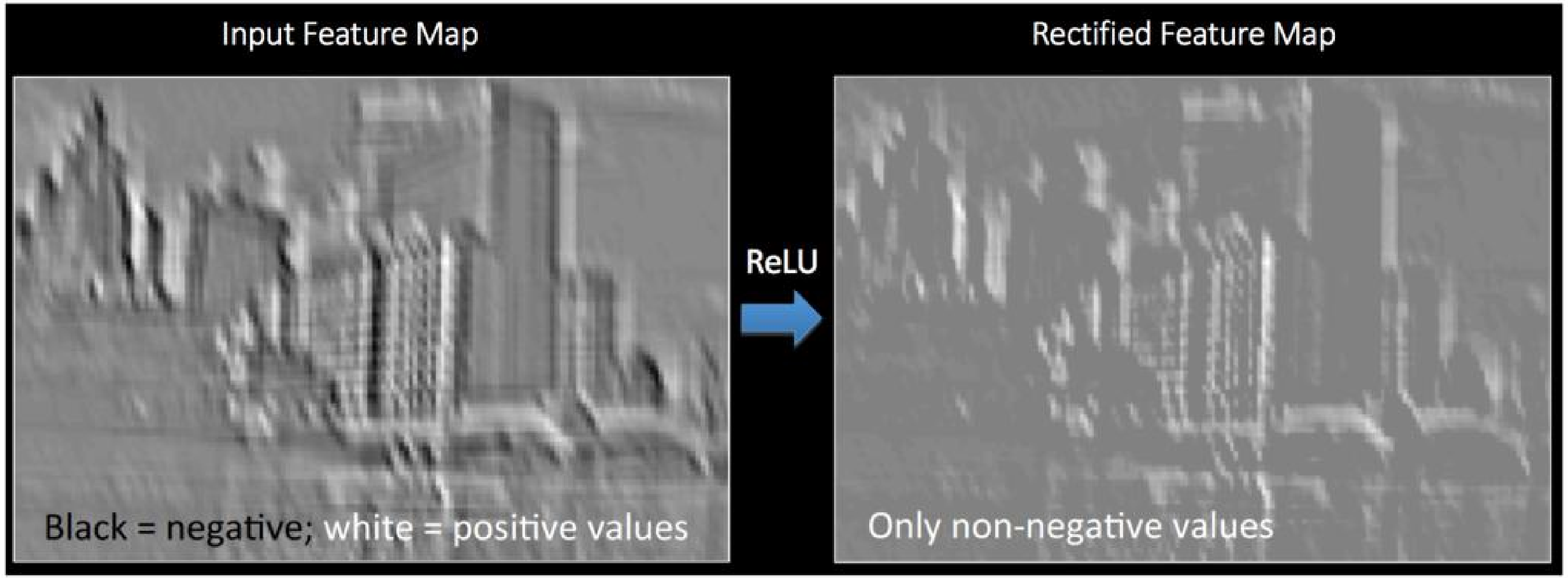


图 4.4 ReLU、Leaky ReLU、ELU 以及 Softplus 函数

激活层

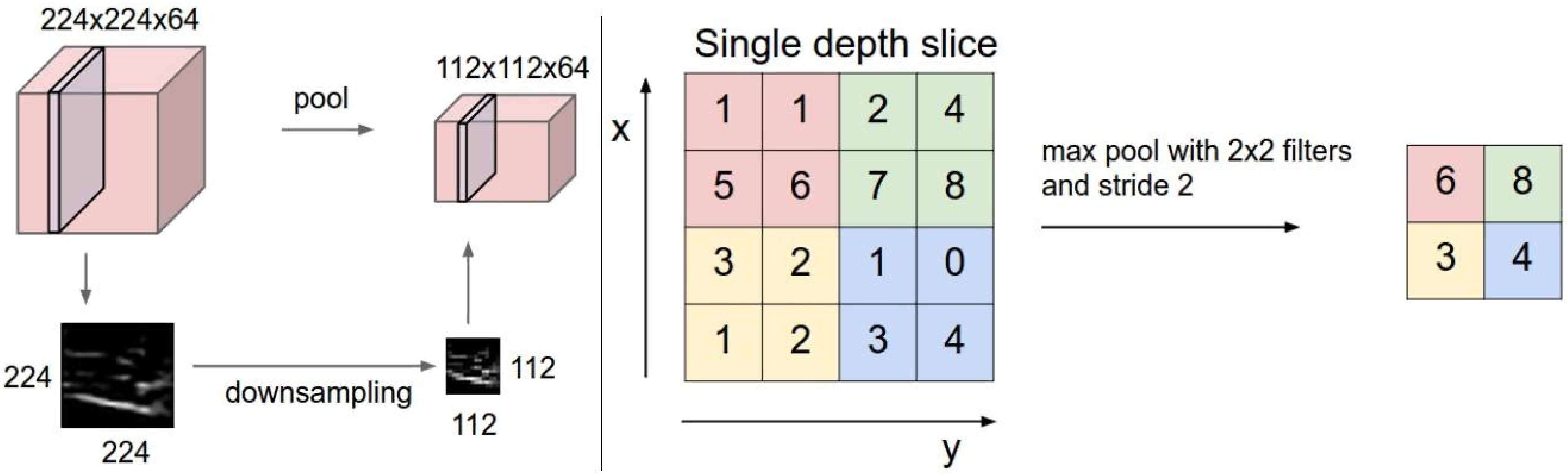
- 基本操作单元：激活层



池化层

池化层

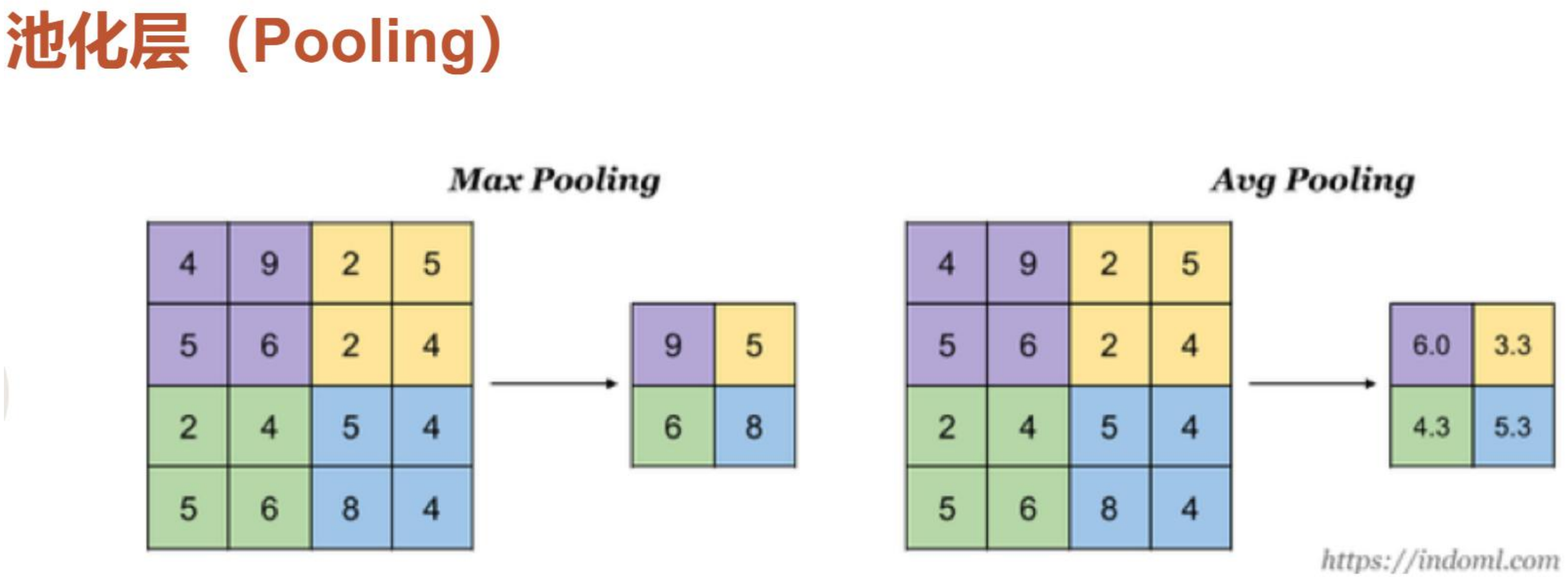
• 基本操作单元：池化Pooling / 降采样 层



Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume. **Left:** In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved. **Right:** The most common downsampling operation is max, giving rise to **max pooling**, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2x2 square).

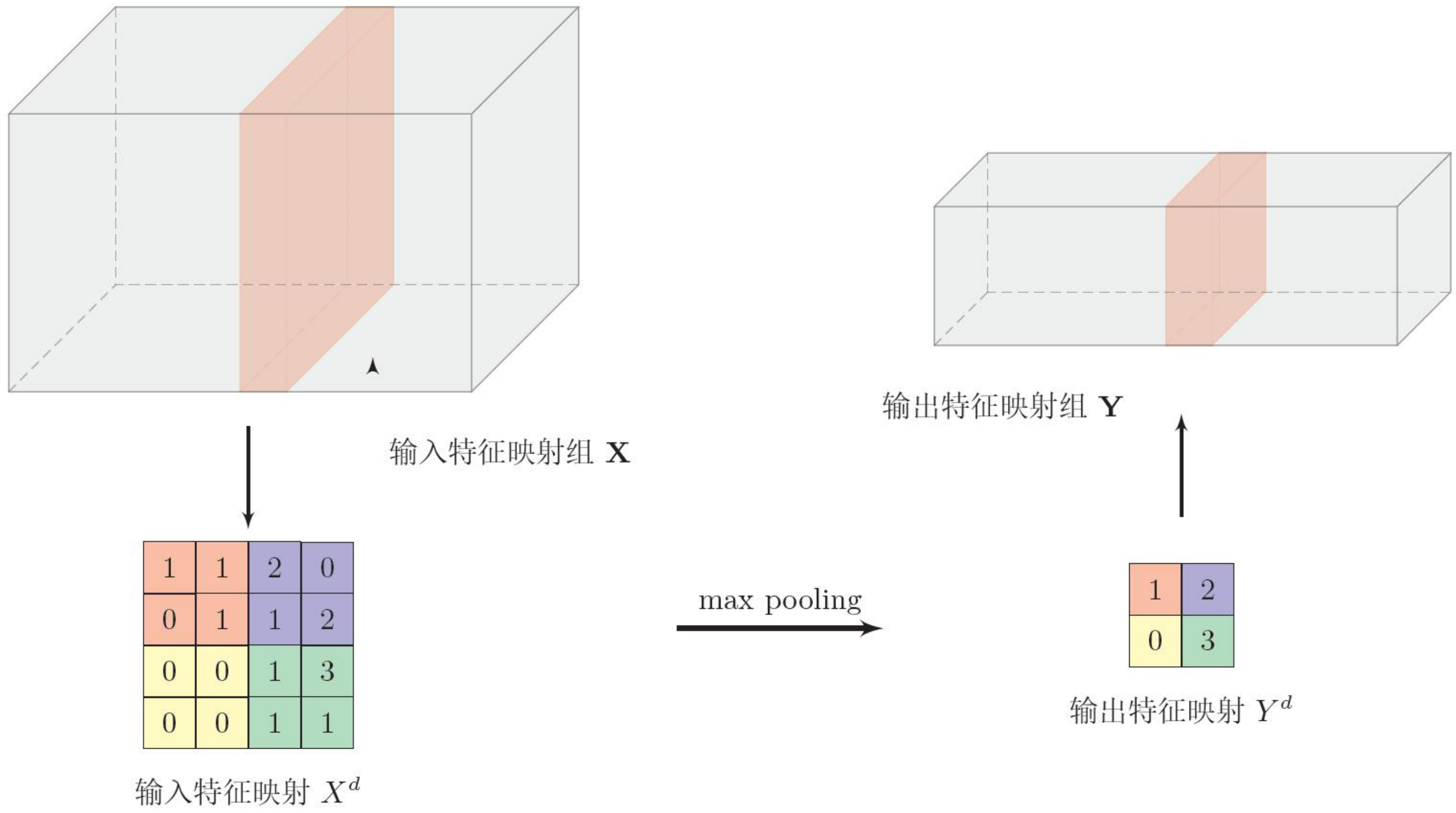
池化层

- 基本操作单元：池化Pooling / 降采样 层



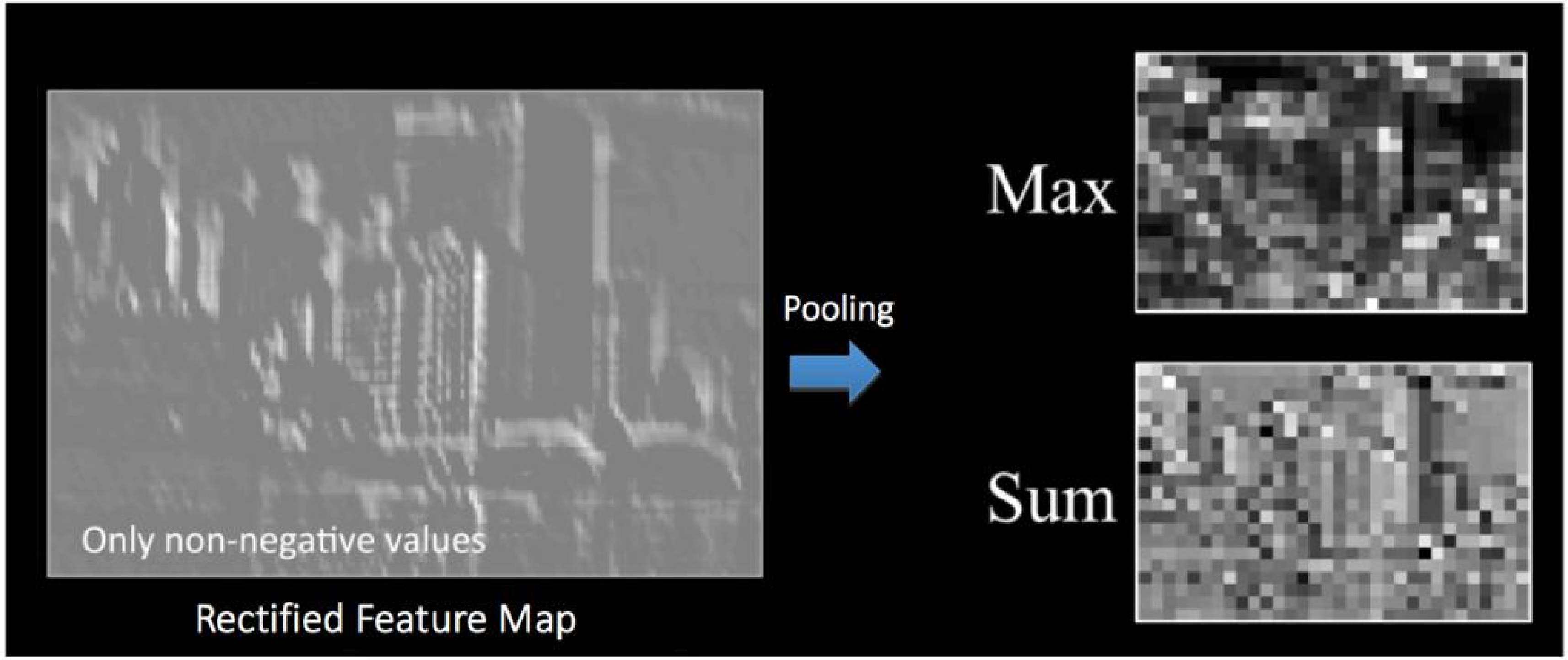
池化层

- 池化Pooling / 降采样 层



池化层

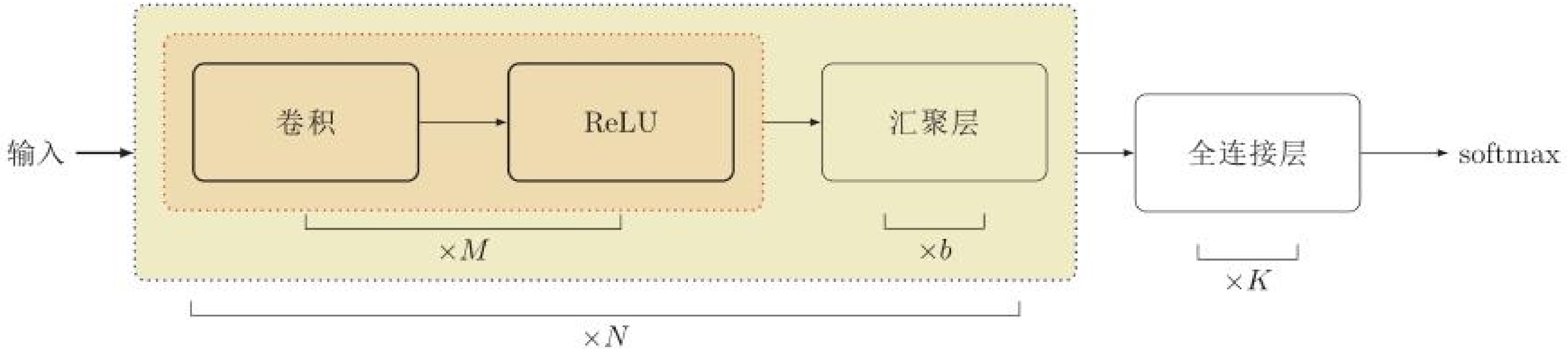
• 基本操作单元：池化Pooling / 降采样 层



CNN结构

卷积网络结构

- 卷积网络是由卷积层、汇聚层、全连接层交叉堆叠而成。
  - 趋向于小卷积、大深度
  - 趋向于全卷积
- 典型结构



  - 一个卷积块为连续M 个卷积层和b个汇聚层（M通常设置为2 ~ 5，b为0或1）。一个卷积网络中可以堆叠N 个连续的卷积块，然后在接着K 个全连接层（N 的取值区间比较大，比如1 ~ 100或者更大；K一般为0 ~ 2）。

Example: LeNet5

```python
class LeNet(nn.Module):
    def __init__(self):
        super(LeNet, self).__init__()
        self.conv1 = nn.Conv2d(1, 6, 5, padding=2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1   = nn.Linear(16*5*5, 120)
        self.fc2   = nn.Linear(120, 84)
        self.fc3   = nn.Linear(84, 10)

    def forward(self, x):
        x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))
        x = F.max_pool2d(F.relu(self.conv2(x)), (2, 2))
        x = flatten(x)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

- 深度特征学习

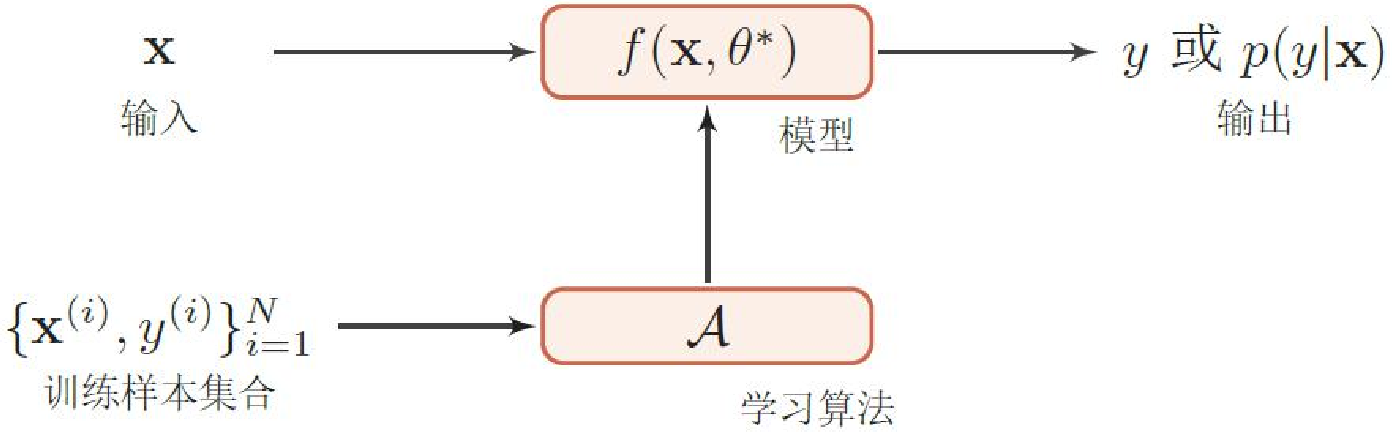
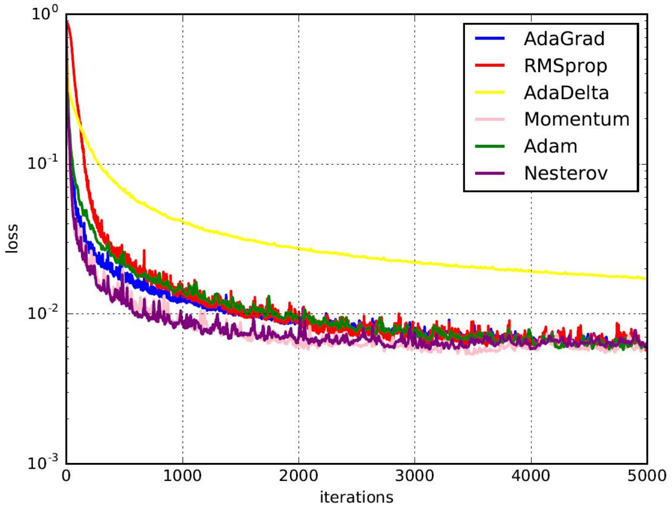


图 2.2 机器学习系统示例

$L = 1, \text{ if } y \neq f(x)$

$L = CE(y, f(x))$ 交叉熵

$L = ||y - f(x)||_1$



- 深度特征学习

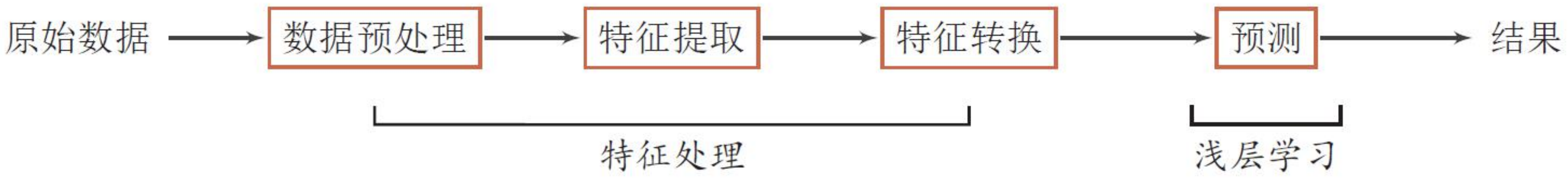


图 1.3 传统机器学习的数据处理流程

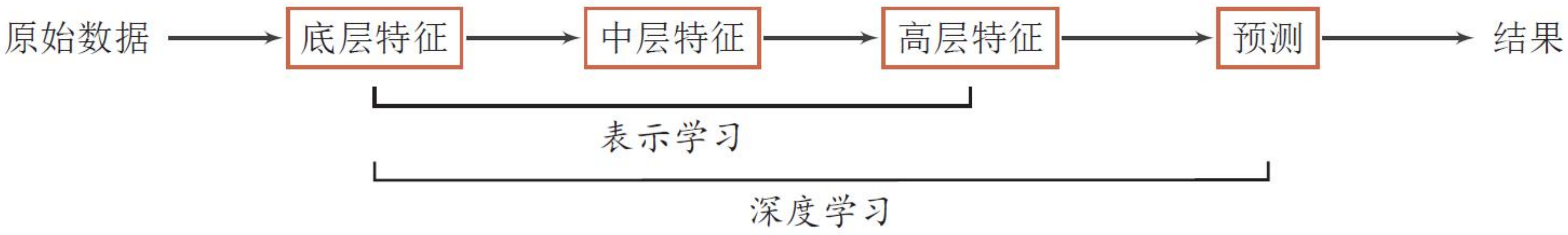
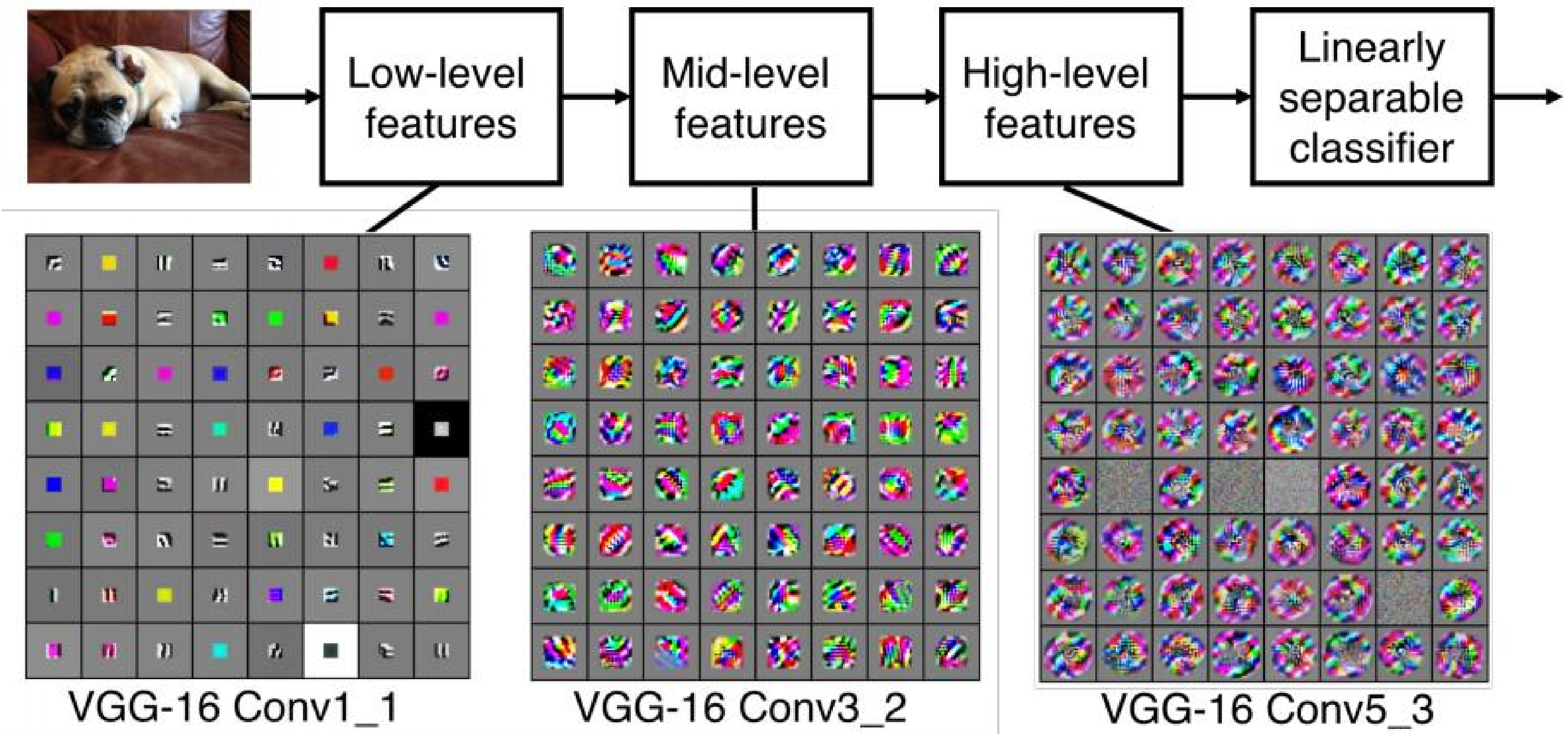


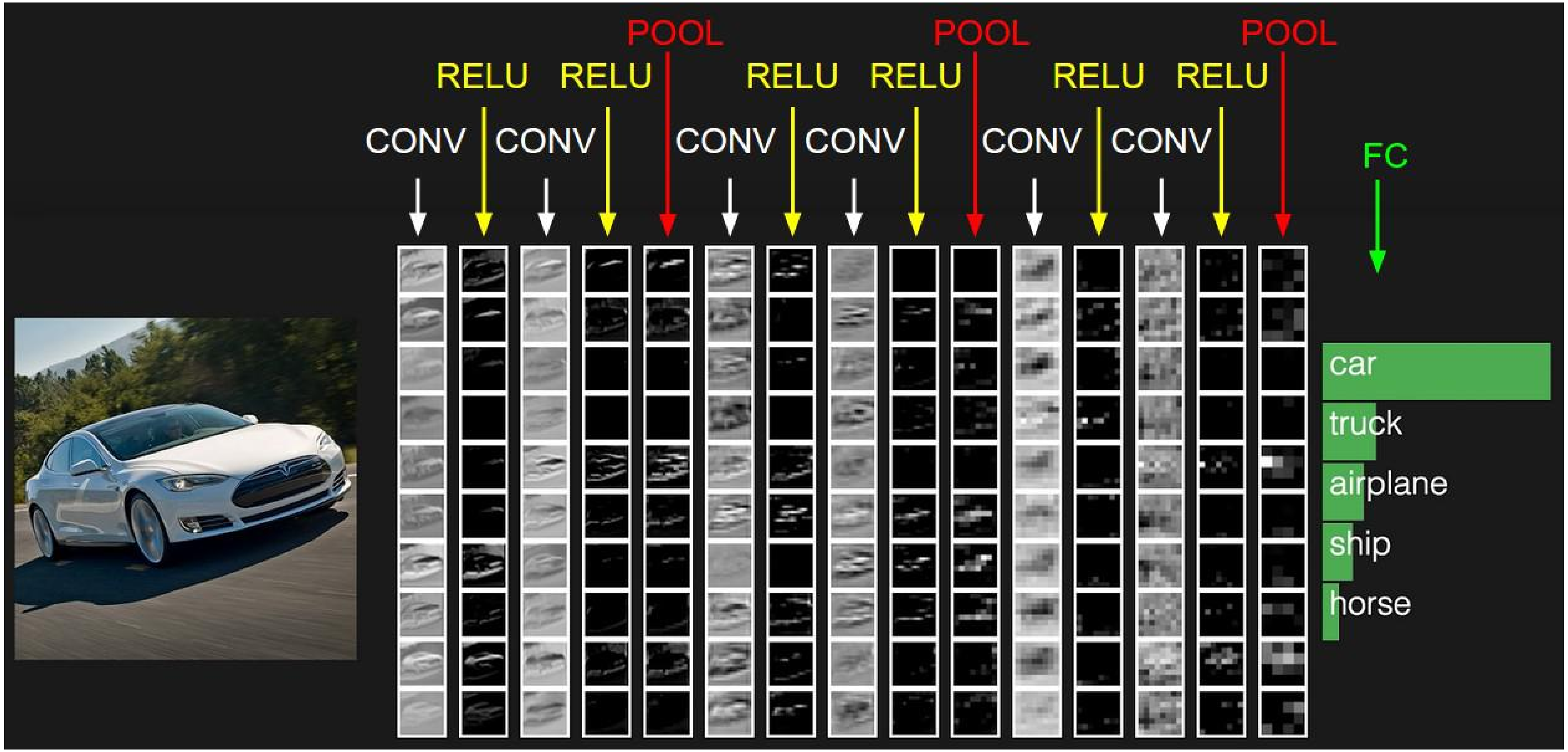
图 1.5 深度学习的数据处理流程

表示学习

- CNN以图像的原始像素作为输入，基于输出层定义的损失函数使用反向传播算法端到端(End-to-end)学习，从而自动学习得到图像底层到高层的层次化语义表达



表示学习



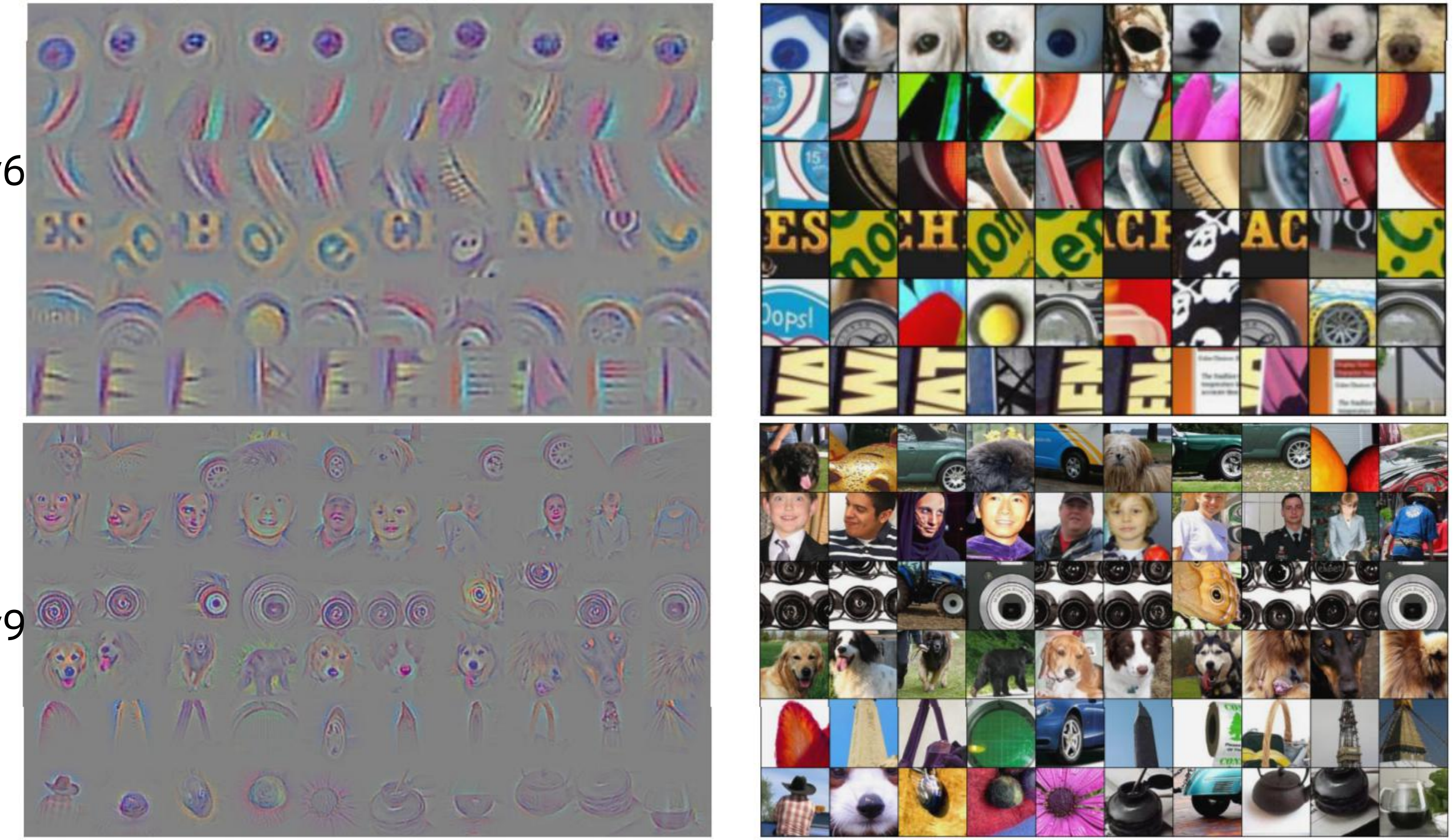
POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

浅层学习局部特征，深层学习整体特征

- 神经网络可视化：

conv6

conv9



Springenberg, J. T.; Dosovitskiy, A.; Brox, T. & Riedmiller, M. Striving for simplicity: the all convolutinal net ICML, 2015, 1-12

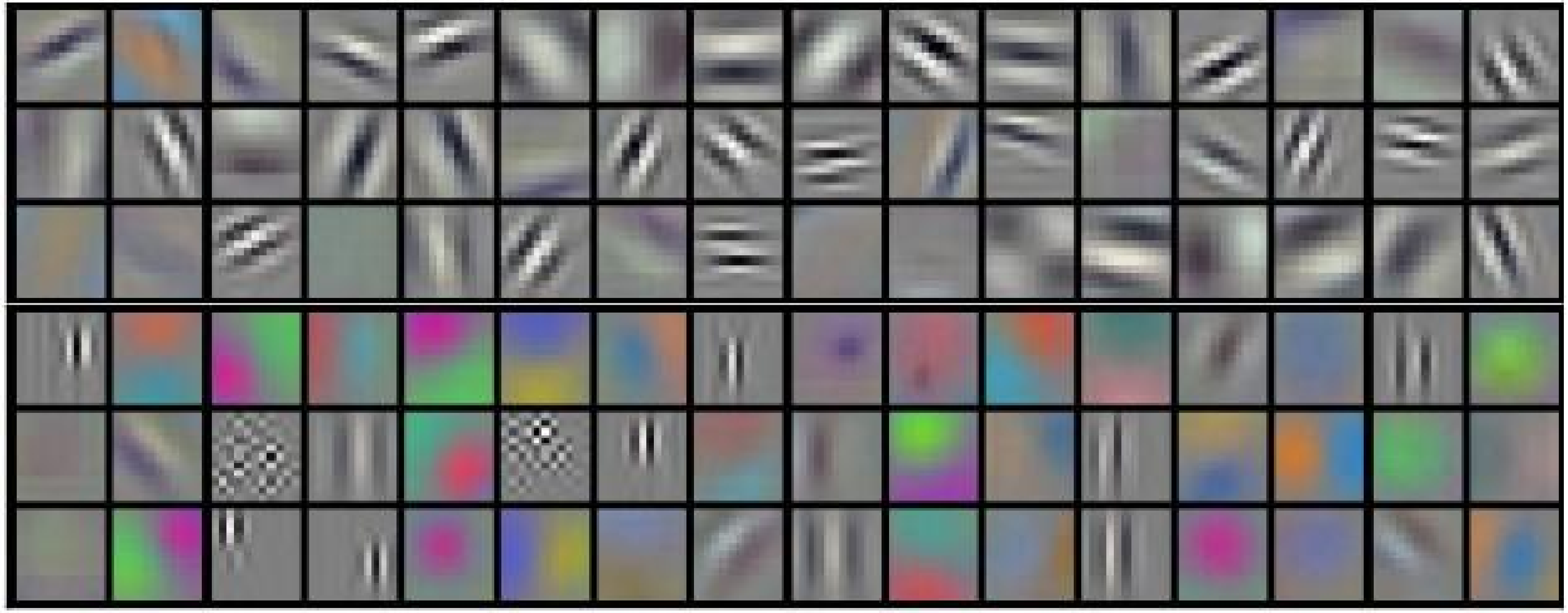
卷积神经网络结构

- 为何选择“深”而非“广”的网络结构
  - ➢ 即使只有一层隐层，只要有足够的神经元，神经网络理论上可以拟合任意连续函数。为什么还要使用深层网络结构？
- 深度网络可从局部到整体“理解图像”
  - ➢ 学习复杂特征时（例如人脸识别），浅层的卷积层感受野小，学习到局部特征，深层的卷积层感受野大，学习到整体特征。
- 深度网络可减少权重数量
  - ➢ 以宽度换深度，用多个小卷积替代一个大卷积，在获得更多样特征的同时所需权重数量也更少。

CNN可视化

CNN 可视化：滤波器

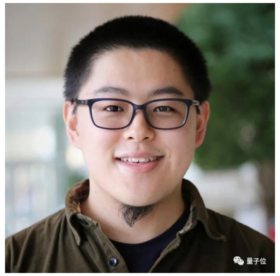
• AlexNet中的滤波器（96 filters [11x11x3]）

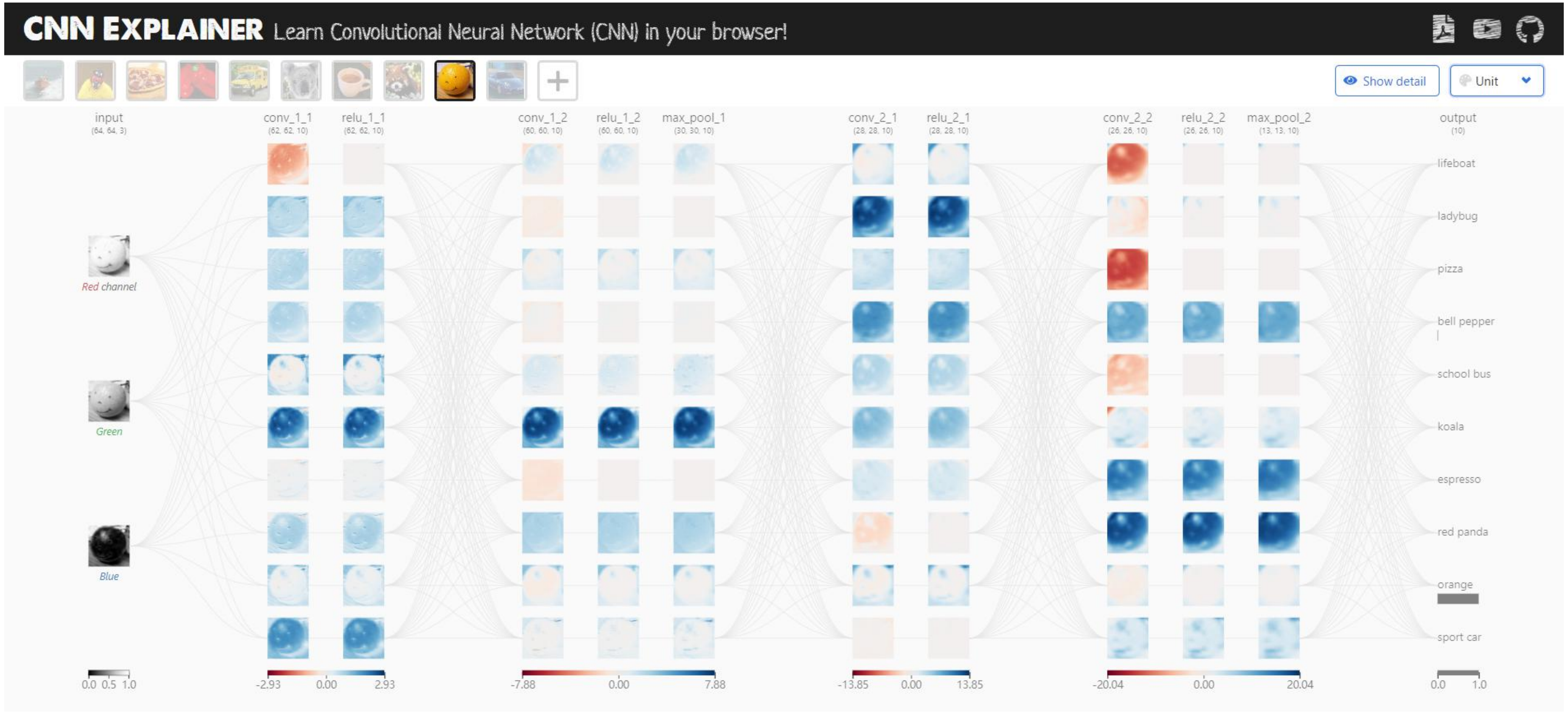


CNN 可视化：CNN Explainer 解释器

- CNN解释器：https://poloclub.github.io/cnn-explainer/
- GitHub：https://github.com/poloclub/cnn-explainer
- 论文：https://arxiv.org/abs/2004.15004

佐治亚理工 Zijie Wang



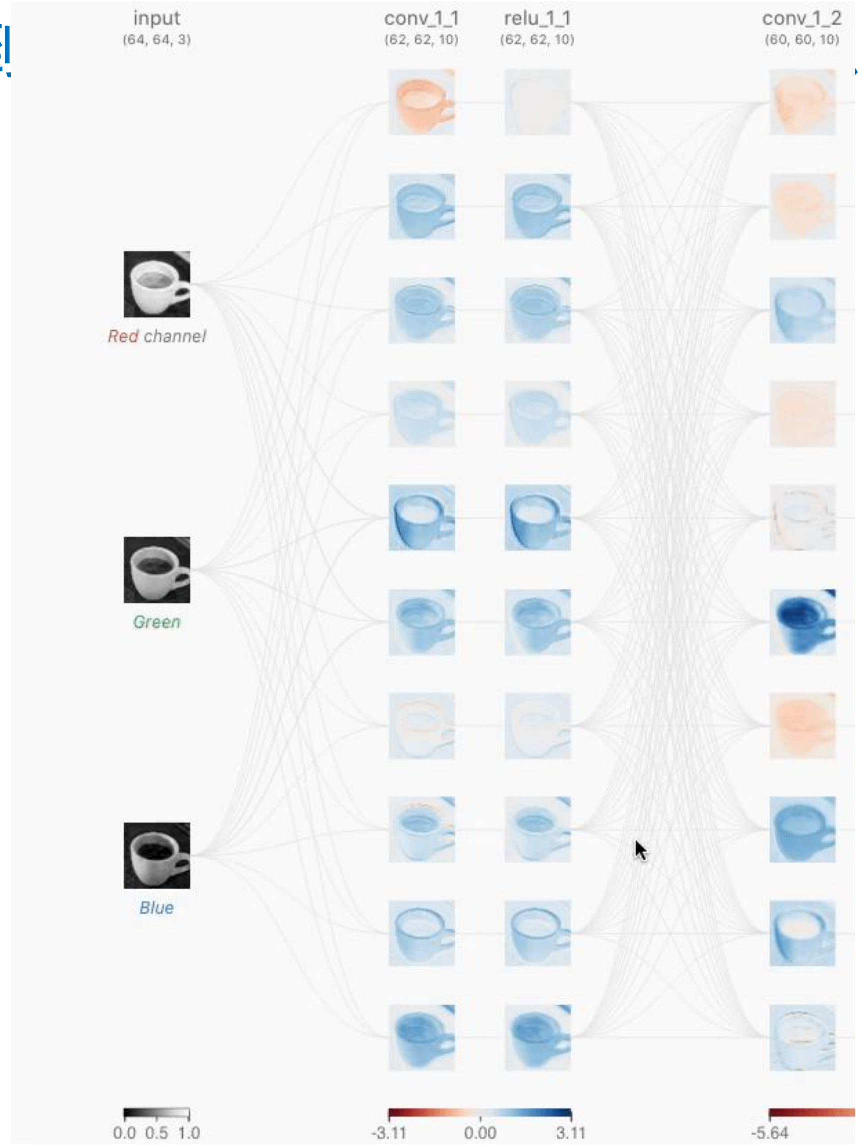


CNN 可视化：CNN Explainer 解释器

- 单击神经元，进入弹性解释视图，就可以看到卷积核滑动的过程的动画模拟：

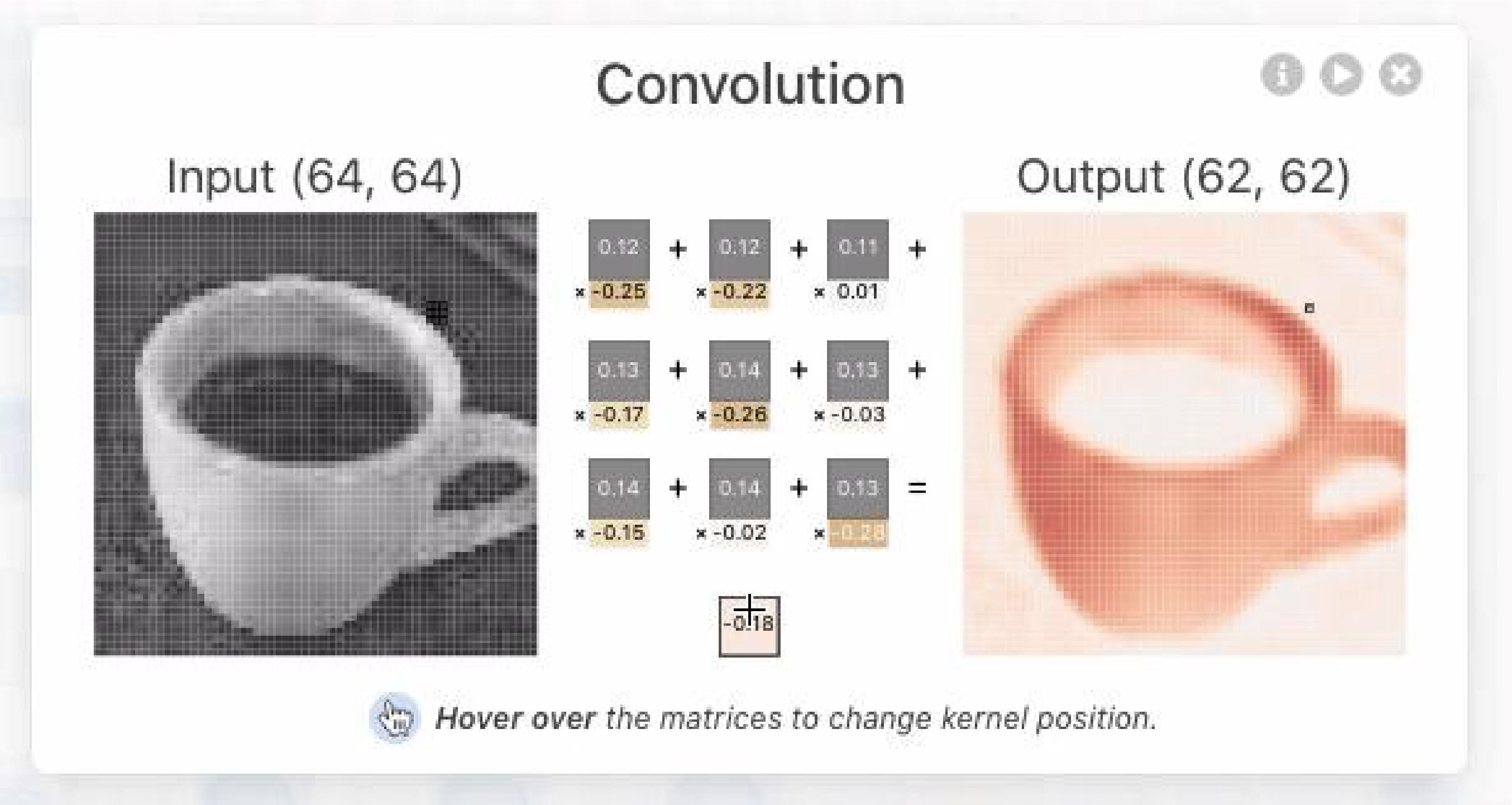
CNN 可视化：CNN Explainer 解释器

- 单击神经元，进入弹性解释视图，就可以看到:



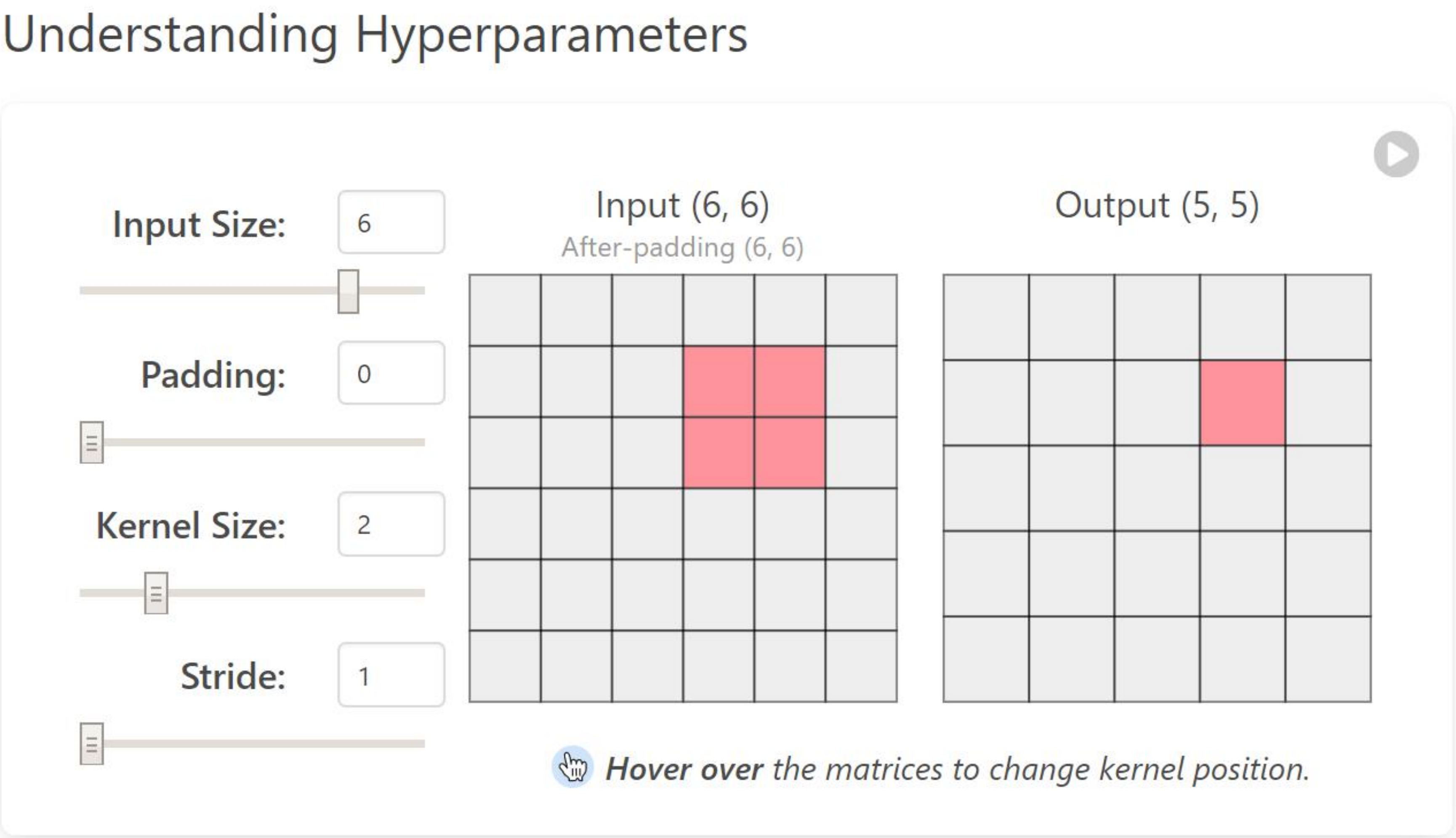
CNN 可视化：CNN Explainer 解释器

- 单击神经元，进入弹性解释视图，就可以看到卷积核滑动的过程的动画模拟：



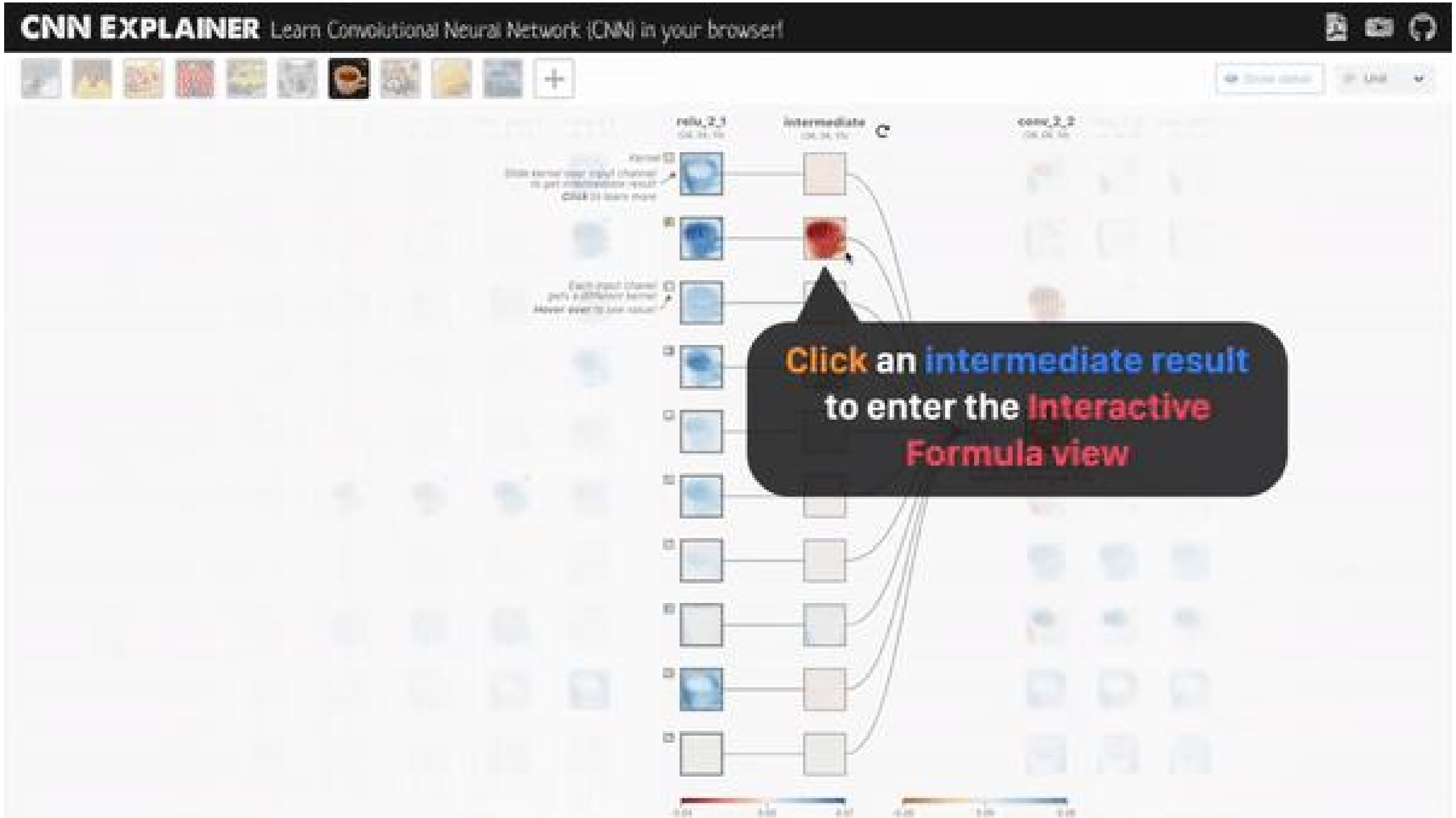
CNN 可视化：CNN Explainer 解释器

- 理解超参：



CNN 可视化：CNN Explainer 解释器

• 点击一个正在卷积的过程图，就可以看到更具体的过程：

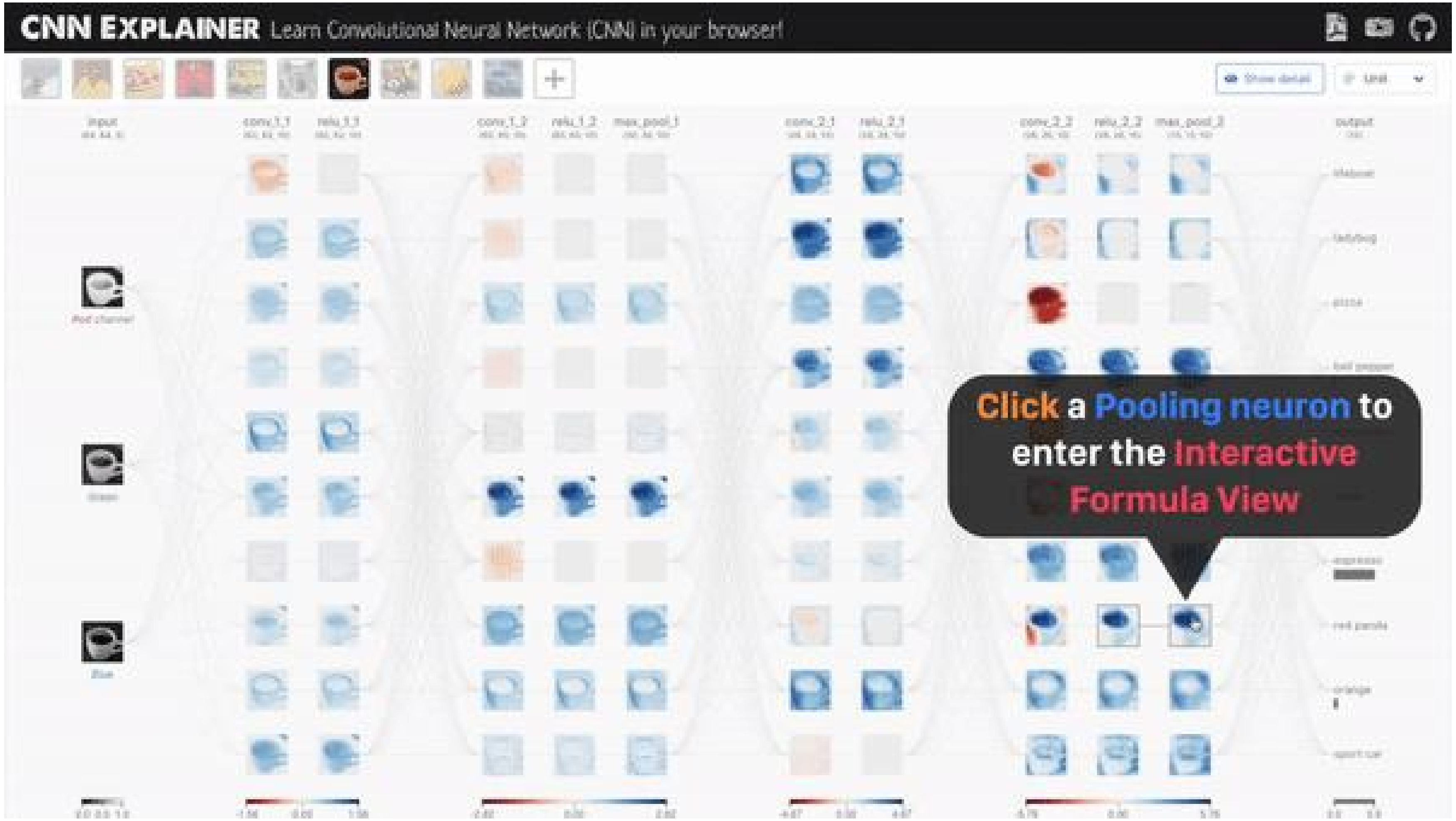


CNN 可视化：CNN Explainer 解释器

- 点击一个ReLU层的神经元，可以看具体过程，ReLU函数是这样工作的：

CNN 可视化：CNN Explainer 解释器

- 点击一个池化神经元，也可以看具体最大池化层是怎样工作的：

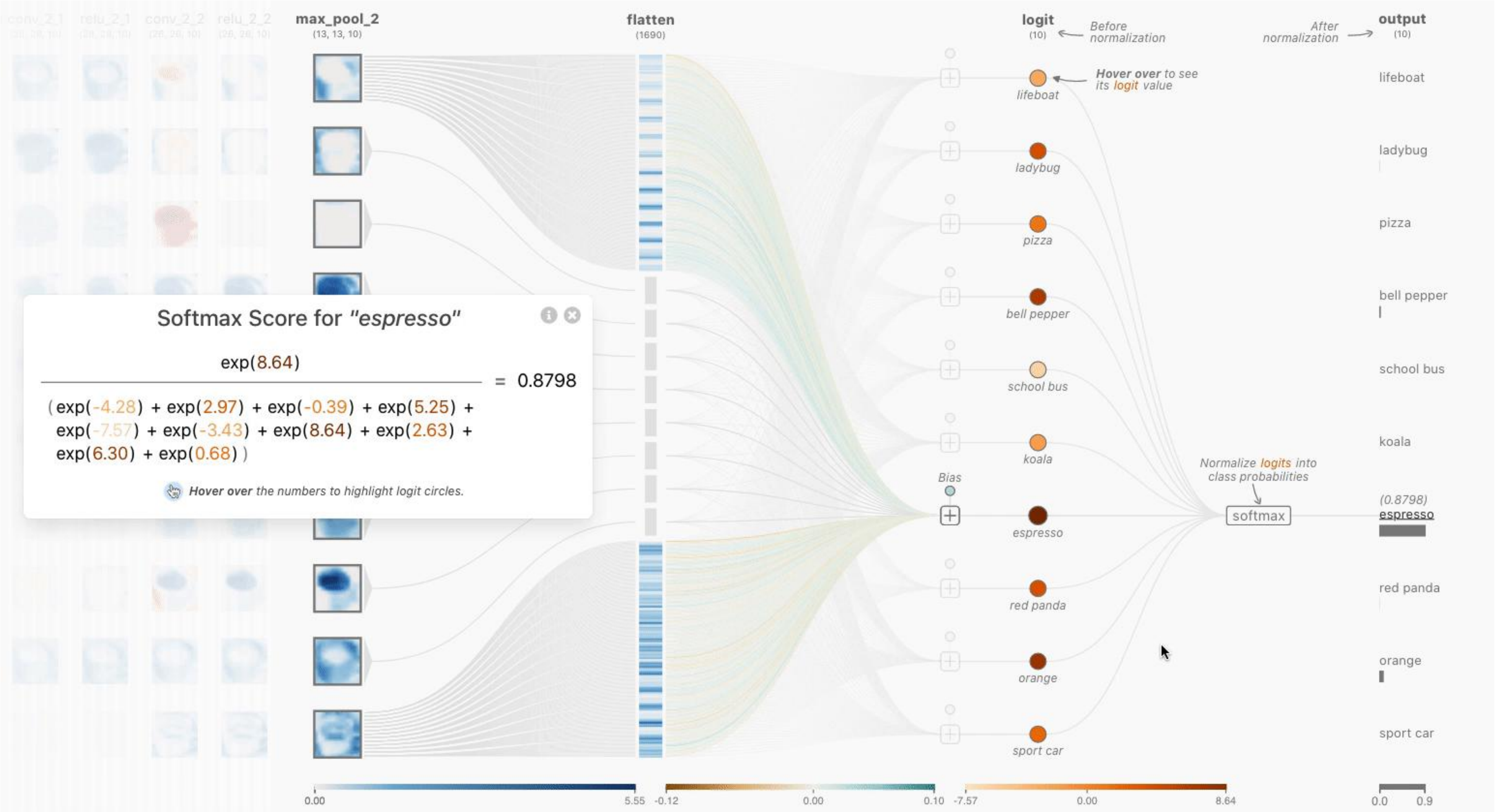


CNN 可视化：CNN Explainer 解释器

- 看清CNN是怎么输出预测的
  - 点击最右侧的输出神经元，进入弹性解释视图：

CNN 可视化：CNN Explainer 解释器

- 可以查看Softmax函数的详情



CNN 可视化：CNN Explainer 解释器

- 10层处理

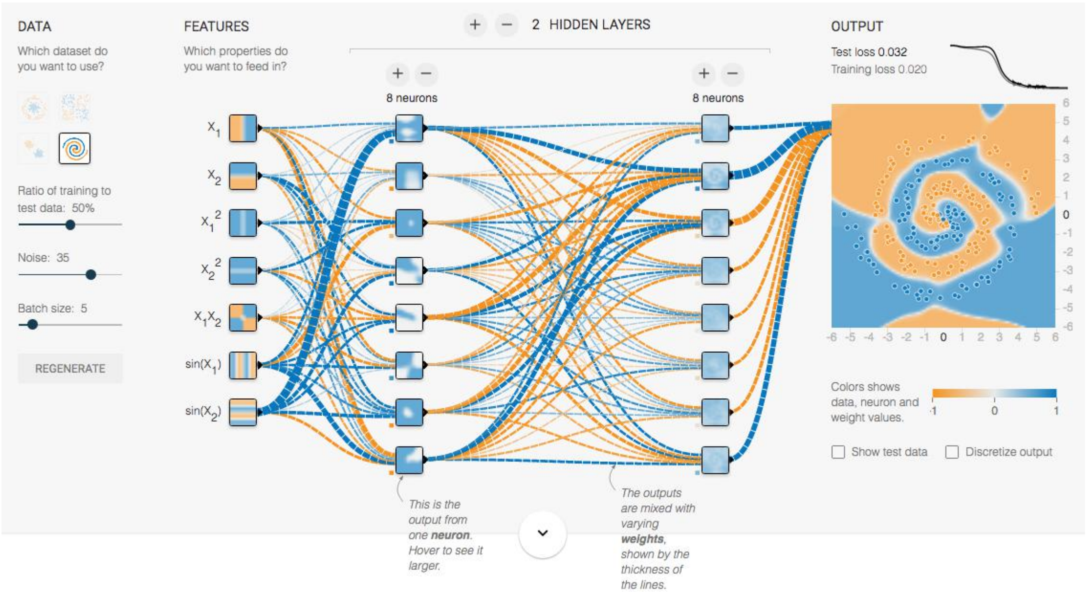
CNN 可视化：CNN Explainer 解释器

- 10层处理

总结

总结

- 神经网络
- 反向传播
- 卷积神经网络
  - 卷积层
  - 池化层
  - 激活层
  - 全连接层、线性层
- 经典网络结构
  - LeNet，AlexNet，VGG，ResNet，DenseNet
- Pytorch、TensorFlow



Thank You !

Q & A