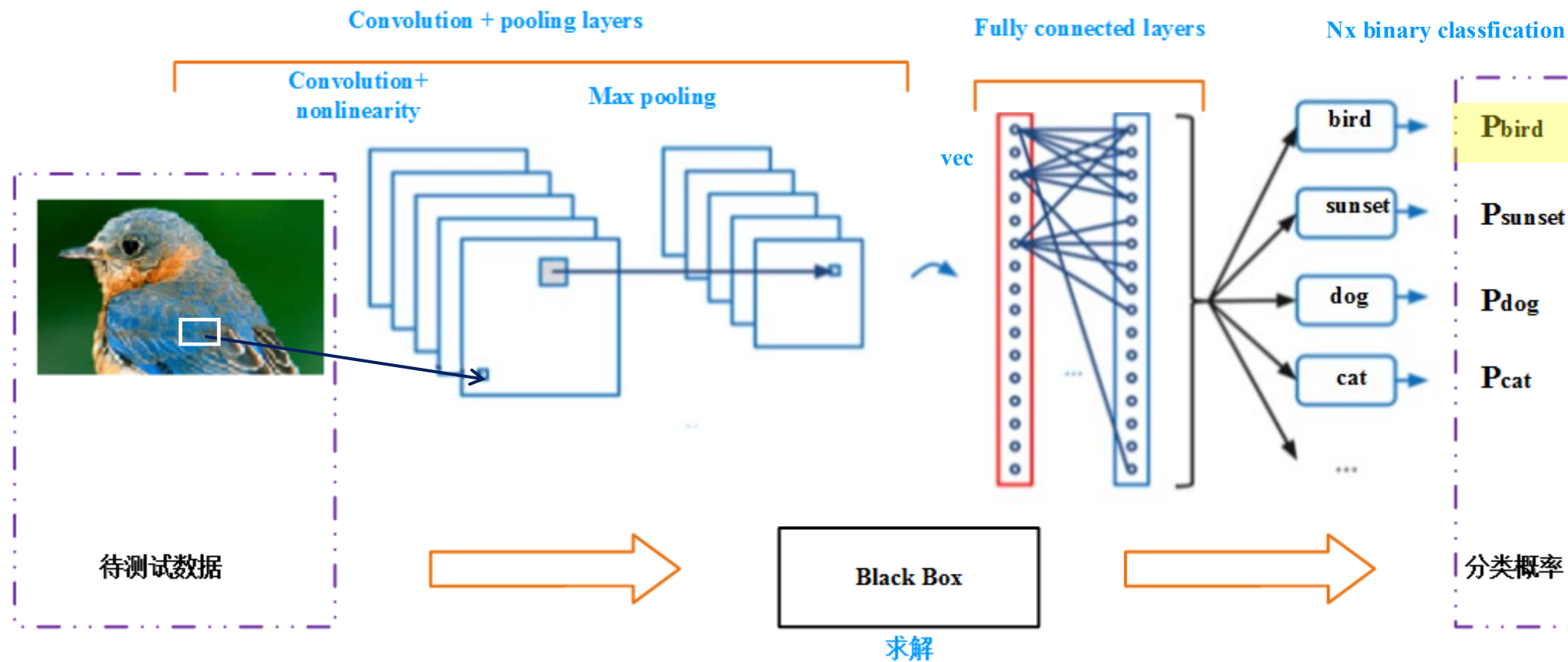


2. 卷积神经网络

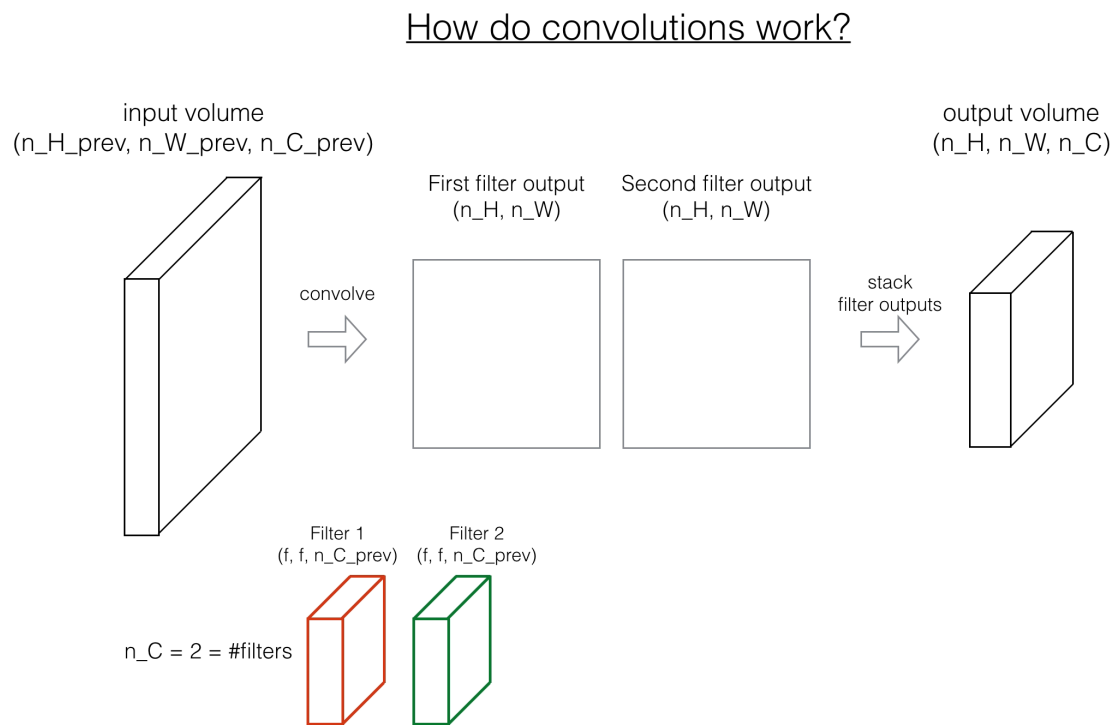
卷积神经网络

- 卷积神经网络主要包括卷积层、池化层、全连接层。



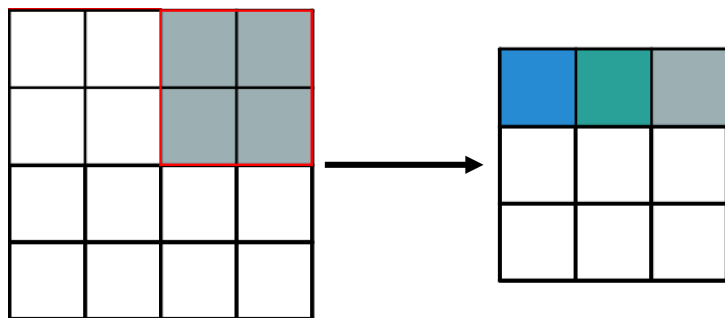
2.1 卷积层

- **卷积核**：决定了卷积的视野。
在2D卷积中，常见卷积核尺寸为3，即 3×3 像素
- **stride**：决定卷积核遍历图像时的步幅大小
- **padding**：决定处理样本时的边界。

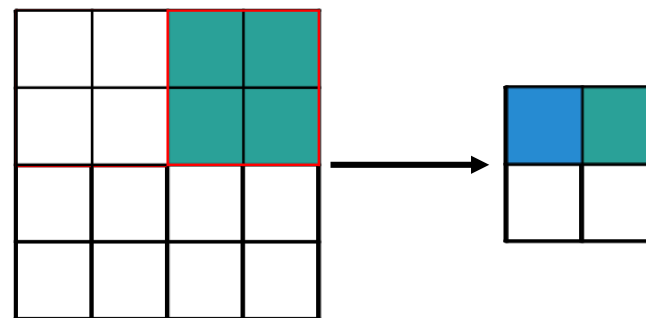


步幅 (stride)

- 表示卷积核在原图像中水平方向和垂直方向每次的步进长度。



kernel为2 Stride = 1

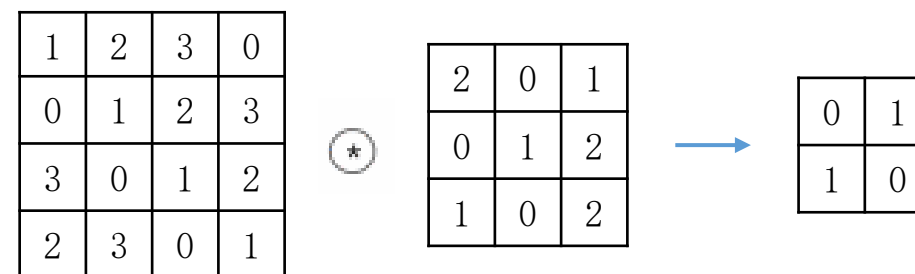


kernel为2 Stride = 2

填充 (padding)

- 问题:

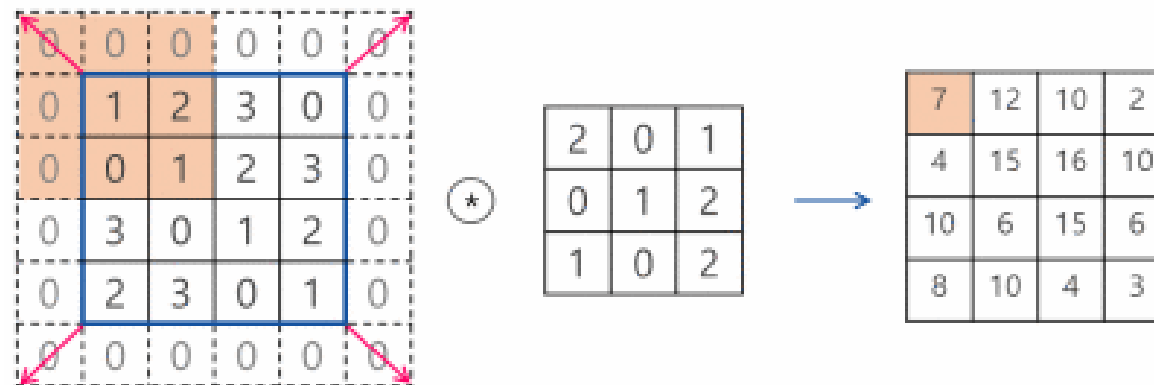
1. 卷积运算后，输出图像尺寸缩小；
2. 结果可能会丢失边缘信息。



尺寸缩小：4变为2

- 解决方案:

引入padding，扩充图像。



尺寸不变

卷积运算

- 单通道单个卷积核

1	0	1
0	1	0
1	0	1

Filter (weights)

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Image

4	3	4
2	4	3
2	3	4

Convolved
Feature

4	3	4
2	4	3
2	3	4

+

3

→

7	6	7
5	7	6
5	6	7

加偏置

stride为 1, 无 padding。

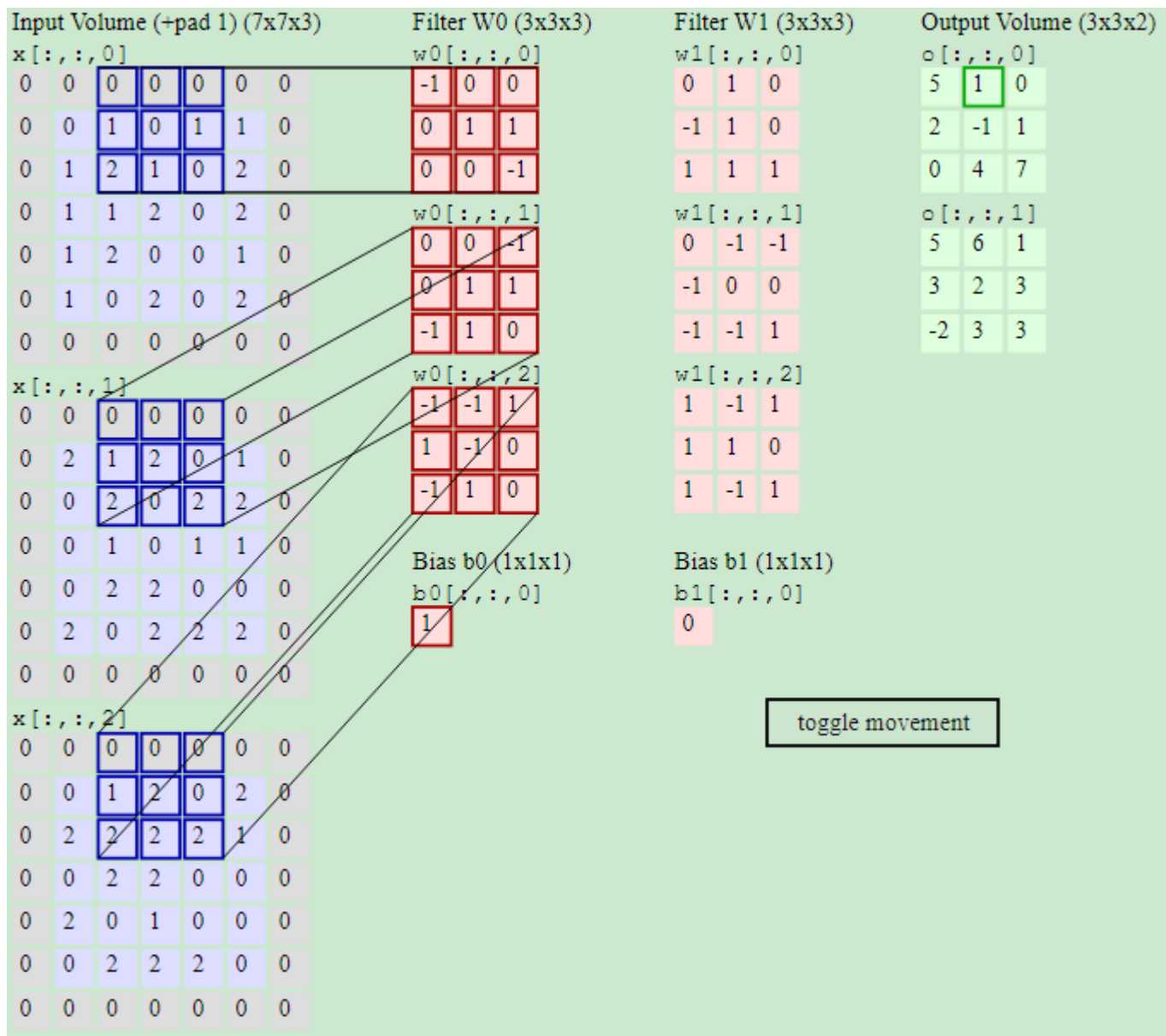
卷积运算

- 多通道多个卷积核

单通道特征图（二维）
作为输入时，使用二维
卷积核。

多通道特征图（三维）
作为输入时，使用三维
卷积核。此时卷积核第
三维和输入特征图的第
三维相等。

stride为 2, padding为 1。

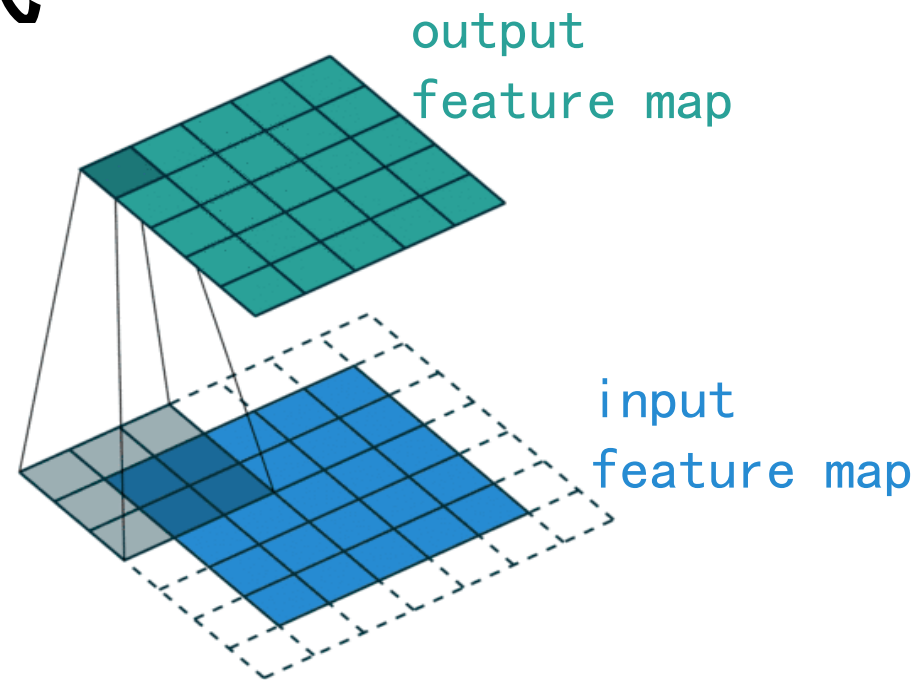


卷积输出的尺寸计算公式

- 输入大小 $n \times n$
- 卷积核大小 $k \times k$
- 步幅 (stride) s
- 填充 (padding) p

- 输出大小为 $m \times m$

$$m = \frac{(n - k + 2p)}{s} + 1$$



$$n = 5, k = 3, s = 1, p = 1$$

$m \quad ? \quad ?$

$$m = \frac{(n - k + 2p)}{s} + 1 = 5$$

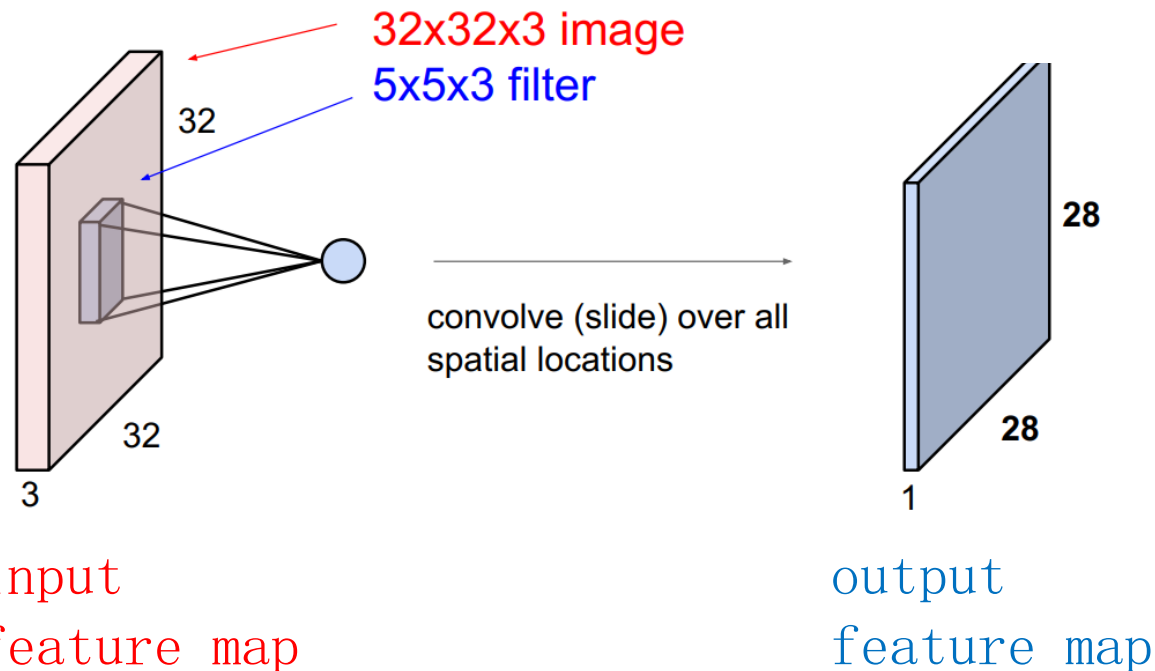
卷积参数计算公式

- 输入大小 $n \times n \times c$
- 卷积核大小 $k \times k \times c$
- 卷积核数目(输出通道数) m

- 参数数目 n'

$$n' = (k \times k \times c + 1) \times m$$

1为bias



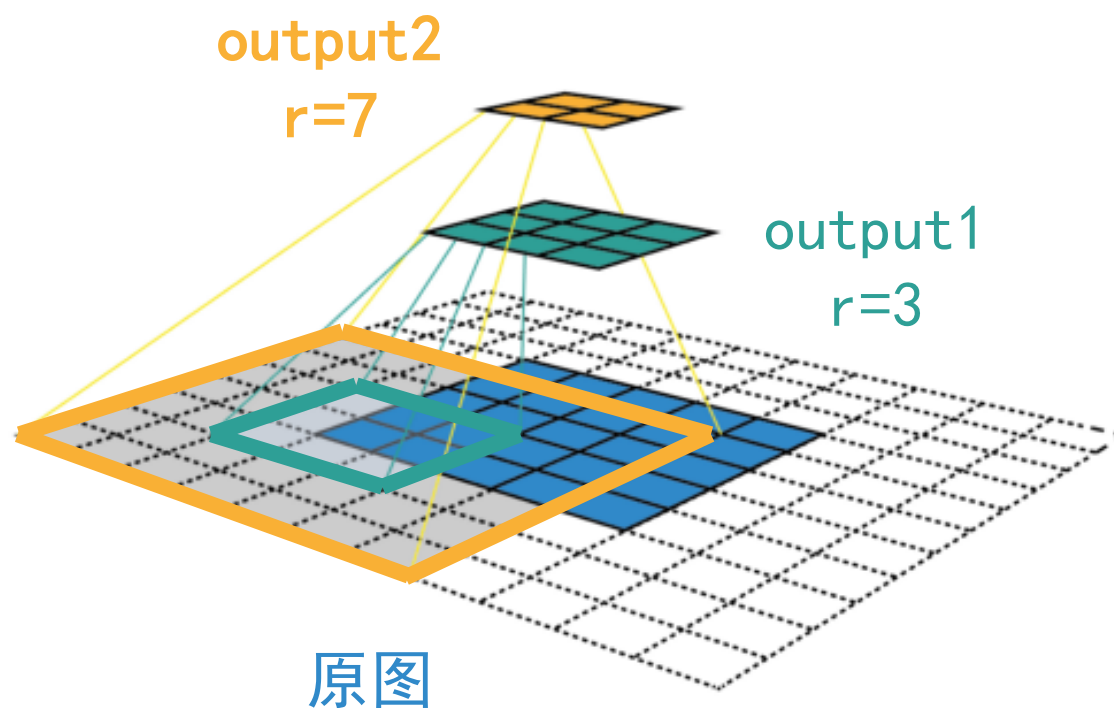
$$n = 32, c = 3, k = 5, m = 1$$

$n' \quad ? \quad ?$

$$n' = (k \times k \times c + 1) \times m = 76$$

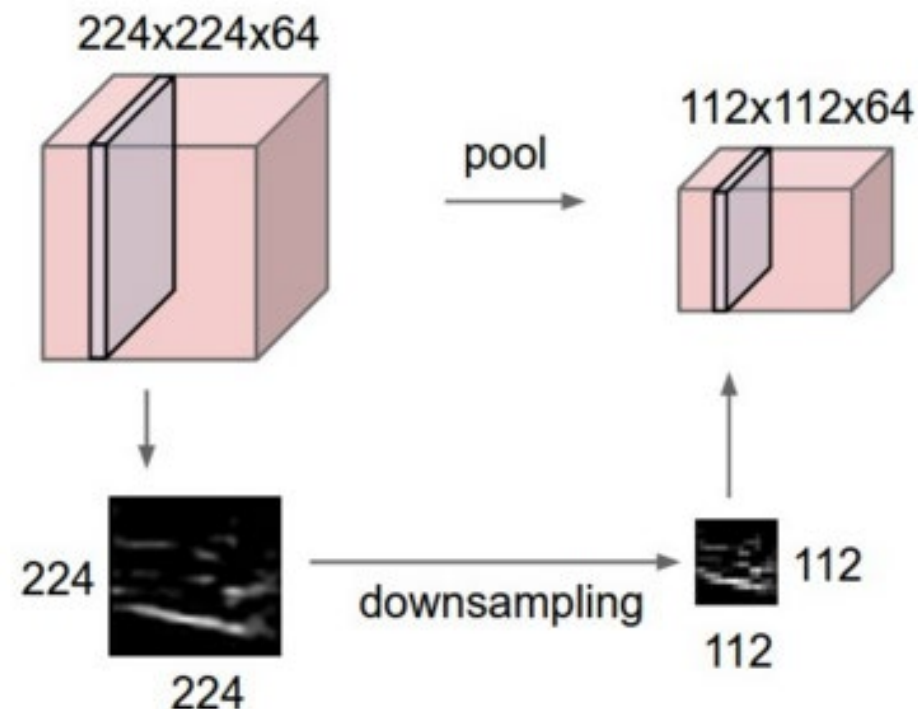
感受野 (receptive field)

- 在卷积神经网络中，决定某一层输出结果中一个元素所映射的原图的区域大小，被称作感受野。



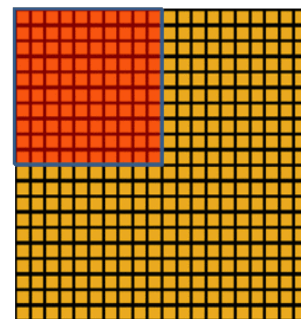
2.2 池化层 (Pooling)

- 对输入的特征图进行下采样
 1. 提供了很强的鲁棒性；
 2. 使特征图变小，简化网络计算复杂度。

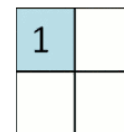


池化

- 池化简单来说就是将输入图像切块，一般选择不重叠的区域，假如池化的分割块大小为 $h \times h$ ，分割的步长为 j ，那么一般 $h=j$ ，如图，如果需要重叠，只需要 $h > j$ 即可。
- 对完整图像切分，再取切分区域中所有值的**均值或最大值**作为代表该区域的新值，即一般最常见的平均池化和最大池化两种。

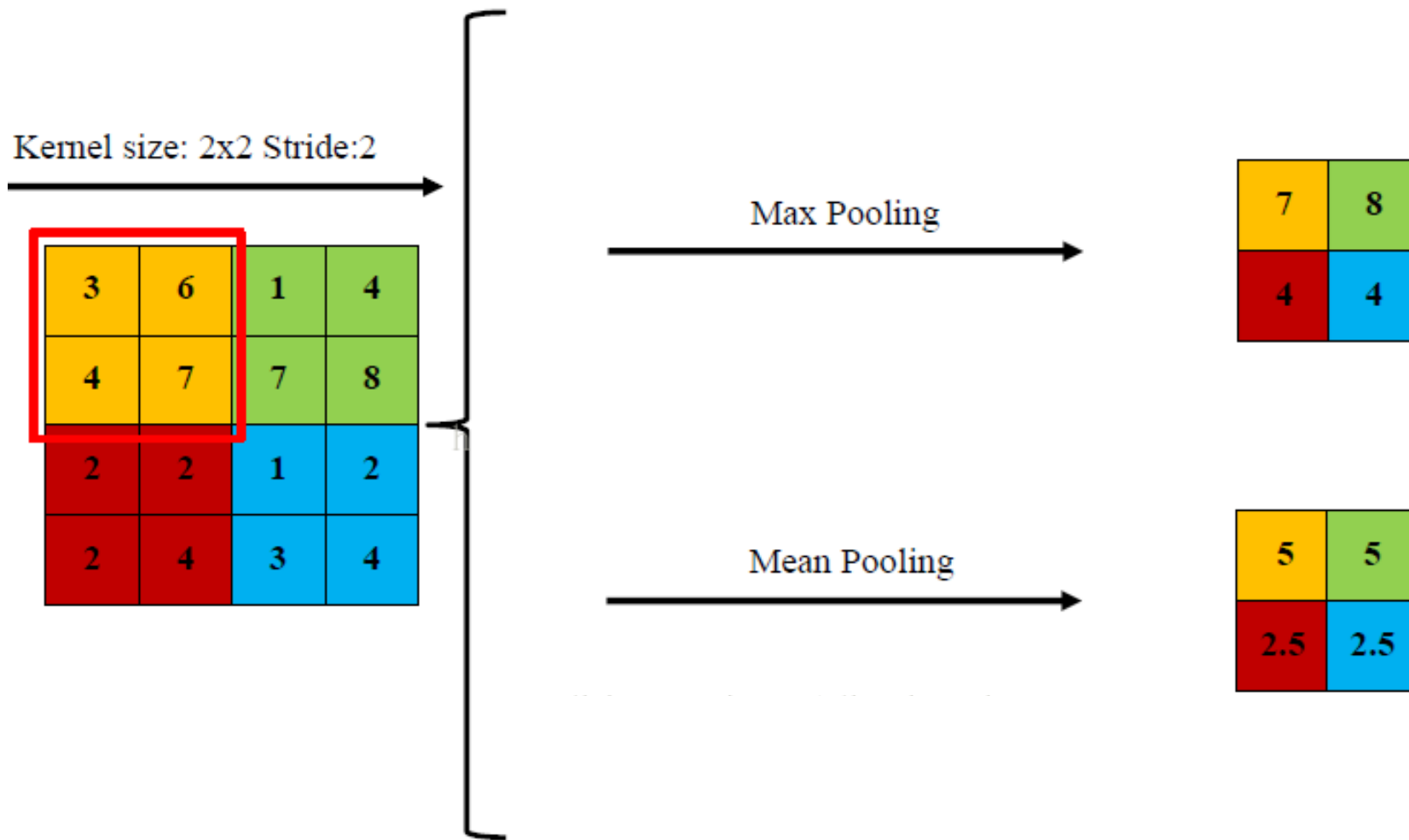


Convolved
feature



Pooled
feature

池化

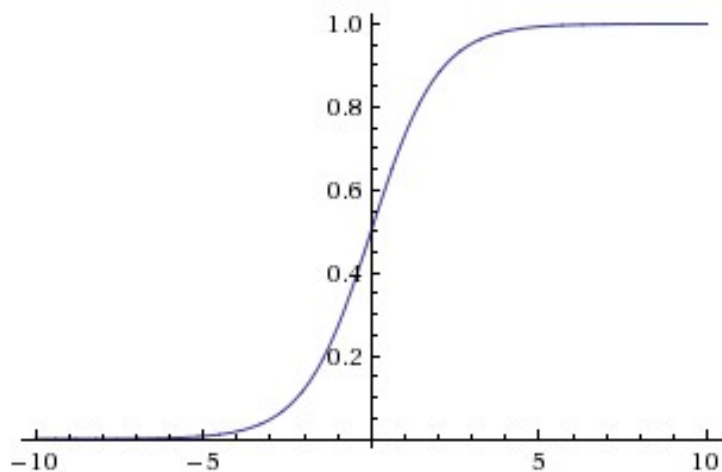


2.3 激活函数

卷积神经网络中通常会在卷积层/池化层之后跟一个**激活函数**，其主要作用是提供网络的**非线性建模**能力，可以更好的拟合实际环境中的非线性分布的数据，其一般具有以下性质：

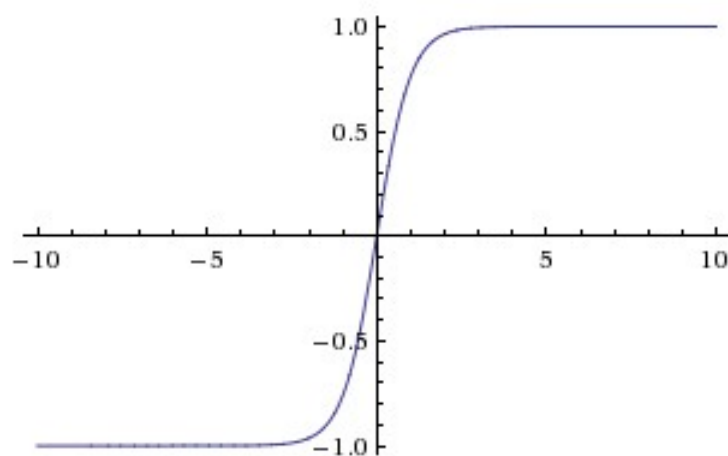
- ✓**非线性**——是多层神经网络的基础，保证多层网络不退化成单层线性网络；
- ✓**几乎处处可微**——可微性保证了在优化中梯度的可计算性；
- ✓**计算简单**——函数本身简单；
- ✓**非饱和性**——不会出现梯度消失、使得参数无法继续更新的问题；
- ✓**输出范围有限**——特定场合，如分类网络概率输出。

激活函数



sigmoid函数

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

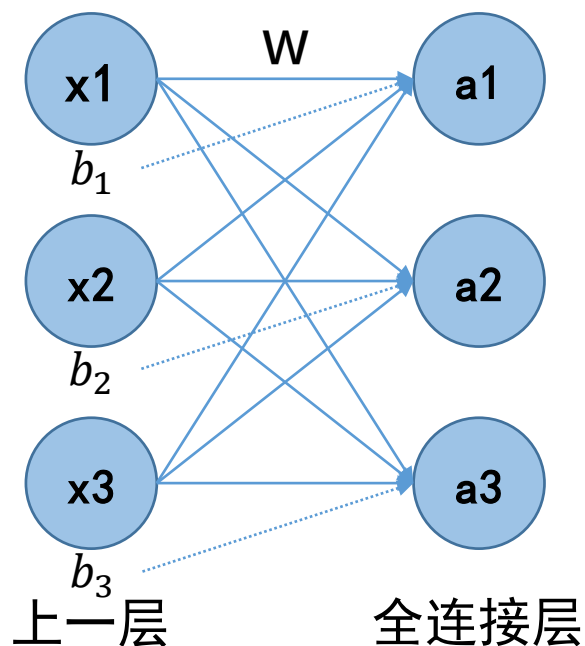


tanh函数

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

2.4 全连接层

- 全连接层的每一个结点都与上一层的所有结点相连，目的是综合前边提取到的所有特征。



全连接层的前向传播：

$$a1 = w_{11} * x1 + w_{21} * x2 + w_{31} * x3 + b_1$$

$$a2 = w_{12} * x1 + w_{22} * x2 + w_{32} * x3 + b_2$$

$$a3 = w_{13} * x1 + w_{23} * x2 + w_{33} * x3 + b_3$$

x 为输入， a 为输出， w 为权重参数， b 为偏置参数

w_{ij} : i 表示上一层神经元下标； j 表示全连接层神经元下标

全连接层实现

