

11 卷积和汇聚层

概要

- 卷积
 - 平移不变性和局部性
 - 卷积层
 - 填充和步幅
- 多个输入和输出通道
- 池化

全连接的问题

分类图像中的狗和猫

- RGB图像具有 36M 个元素
- 使用 100 个神经元单隐含层的 MLP 模型：
 - 有 36 亿个参数
 - 超过地球上的狗和猫的数量 (900M 狗+ 600M 猫)

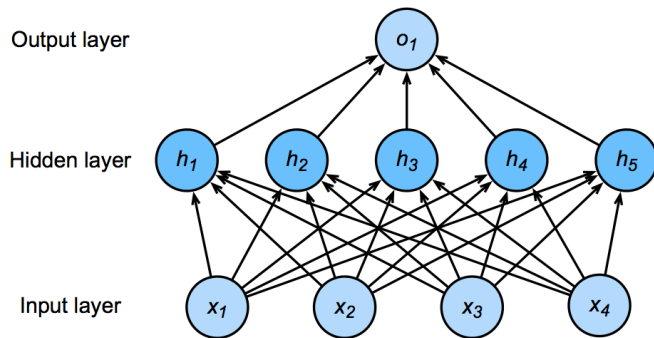


Dual
12MP
wide-angle and
telephoto cameras



单隐含层全连接网络

- 用一个标准的全连接网络（MLP）来分类
 - 36M 特征， 100个神经元
- 参数爆炸 (Parameter Explosion)
 - 3.6B 参数 = 14GB
- 空间结构丢失 (Loss of Spatial Structure)
 - “压平” 操作破坏了图像固有的二维结构。像素和其近邻在向量中可能相距甚远，网络无法先验地知道它们是相邻的



$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Waldo 在哪？

➤ 位置无关的方法



两个原则

➤ 平移不变性

- 不管检测对象出现在图像中的哪个位置，神经网络的前面几层应该对相同的图像区域具有相似的反应

➤ 局部性

- 神经网络的前面几层应该只探索输入图像中的局部区域，而不过度在意图像中相隔较远区域的关系



向生物视觉系统借鉴

➤ 神经科学研究发现，动物的视觉皮层神经元具有两个显著特征

➤ 局部感受野

- 每个神经元只对视野中的一小块局部区域（感受野）产生反应。它只关心这个小区域内的模式，如边缘、角点或纹理

➤ 层级与重复处理

- 整个视野由大量感受野相似的神经元覆盖。这些神经元执行类似的功能（如检测特定方向的边缘），只是位置不同。低级特征（如边缘）被组合成更复杂的特征（如形状），逐层递进

➤ 于是，CNN的两大设计原则

➤ 局部性: 仿照局部感受野

- 特征仅需通过局部像素计算。对应CNN的稀疏连接。一个输出神经元只与输入的一小块区域连接

➤ 平移不变性: 仿照重复处理

- 一个特征的检测方式不应因其在图像中的位置而改变，对应CNN中的参数共享
- 检测同一特征的神经元共享同一组权重，这组权重就是卷积核

卷积：提取特征的利器

冲激函数：通过冲激函数的筛选性质和卷积性质，可以从信号中提取出特定的特征或信息

1D特征探测器

➤ CV长期使用手工设计的卷积核（也称为滤波器）来提取图像的特定特征

➤ 一维示例：在信号中检测“跃升”边缘

➤ 找到信号值突然增大的位置

➤ 设计一个简单的核 $[-1, 1]$ 。用于计算相邻两个信号点的差异

➤ 卷积过程 (Convolution): 将核在信号上滑动

➤ 在平坦区域（如 $[10, 10]$ ），计算结果是 $10*(-1) + 10*1 = 0$

➤ 在跃升边缘（如 $[10, 20]$ ），计算结果是 $10*(-1) + 20*1 = 10$

➤ 输出特征: 输出信号在“边缘”位置产生一个强烈的响应（一个峰值），而在其他地方响应为零

➤ 输出信号的峰值精确标记输入信号中“跃升”特征的位置



2D特征探测器

➤ 二维示例：在图像中检测“垂直”边缘

➤ 检测图像中的垂直边缘

➤ 设计一个3x3的核。用于计算一个局部区域内左右像素的亮度差

➤ 卷积过程 (Convolution): 将3x3的核在图像上滑动

➤ 当核完全位于左侧暗区或右侧亮区时，由于区域内像素值相同，计算结果接近于0

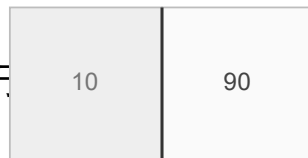
➤ 当核的中心跨越“明暗边界”时，得到一个很大的正数

➤ 输出特征图: 生成的特征图在原始图像的“垂直边缘”位置上会形成一条明亮的“竖线”，其他区域则很暗

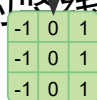
➤ 2D核成功地将图像中的“垂直边缘”特征提取了出来

二维示例：图像边缘检测

输入图像:

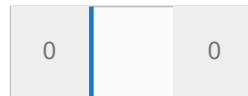


2D 探测核 (垂直边缘)



卷积

输出特征图:



亮线高亮了垂直边缘

二维卷积层

➤ $\mathbf{Y} = \mathbf{X} \star \mathbf{W} + b$

➤ $\mathbf{W}$ 和 b 是可学习的参数

➤ $\mathbf{X}$: $n_h \times n_w$ 输入矩阵

➤ $\mathbf{W}$: $k_h \times k_w$ 核矩阵

➤ b : 偏差标量

➤ $\mathbf{Y}$: $(n_h - k_h + 1) \times (n_w - k_w + 1)$ 输出矩阵

0	1	2
3	4	5
6	7	8

*

0	1
2	3

=

19	25
37	43

不同卷积核提取不同的特征



(维基百科)

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



边缘检测

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



锐化

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

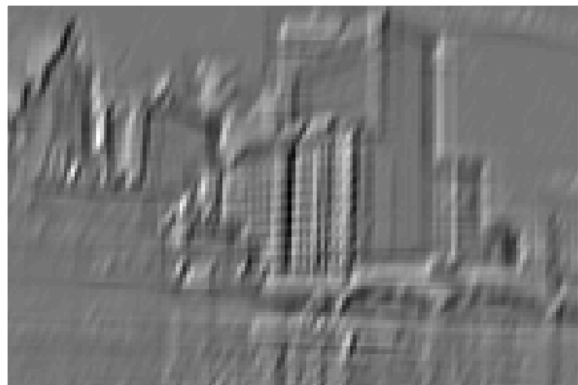
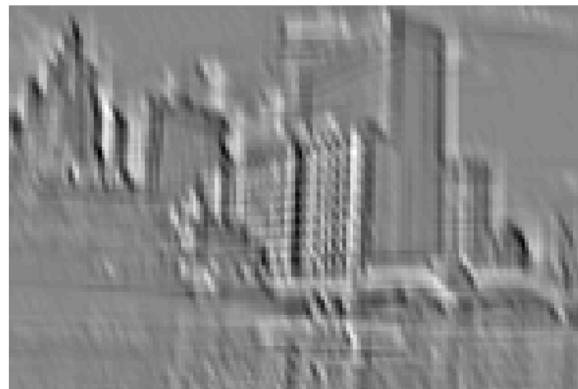


高斯模糊

例子



(Rob Fergus)



互相关（实际上）与卷积

➤ 2-D 互相关

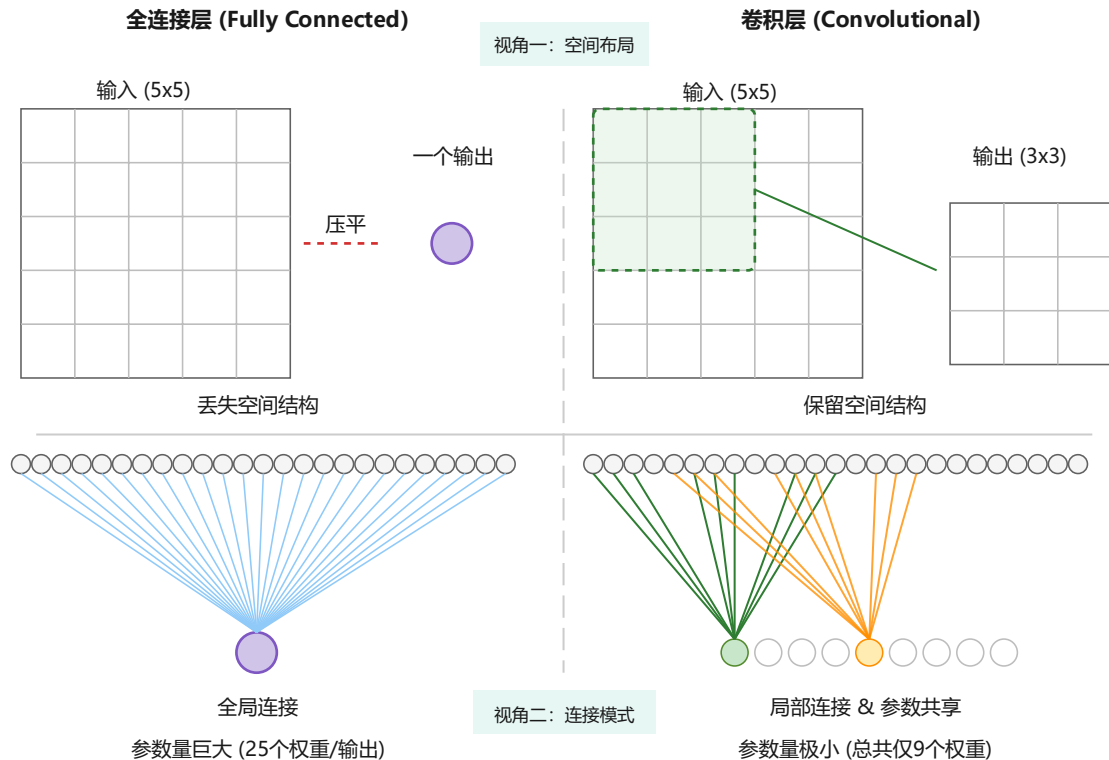
$$y_{i,j} = \sum_{a=1}^h \sum_{b=1}^w w_{a,b} x_{i+a,j+b}$$

➤ -2D 卷积

$$y_{i,j} = \sum_{a=1}^h \sum_{b=1}^w w_{-a,-b} x_{i+a,j+b}$$

➤ 在对称性方面没有差别

CNN VS FCN



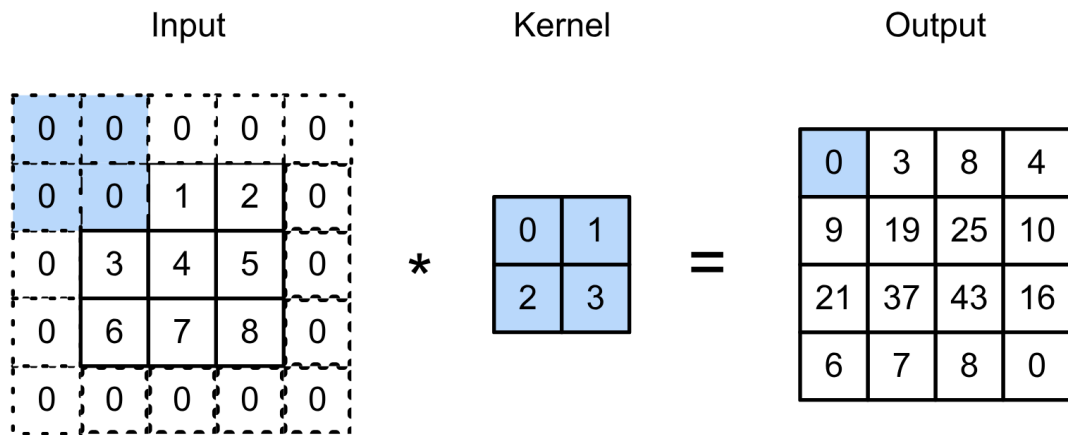
填充和步幅

填充

- 给定输入图像 (32 x 32)
- 应用 5 x 5 大小的 卷积核
 - 第1层得到输出大小 28 x 28
 - 第7层得到输出大小 4 x 4
 - 更大的卷积核可以更快地减小输出
- 形状从 $n_h \times n_w$ 减少到
 - $(n_h - k_h + 1) \times (n_w - k_w + 1)$
- 不填充, 卷积后的图像大小会发生改变

填充

填充：输入图像的边界填充元素



$$0 \times 0 + 0 \times 1 + 0 \times 2 + 0 \times 3 = 0$$

填充

- 填充 p_h 行和 p_w 列，则输出为：

$$(n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1)$$

- 通常取 $p_h = k_h - 1$, $p_w = k_w - 1$
 - 使输入和输出具有相同的高度和宽度
 - 当 k_h 为奇数：在上下两侧填充 $p_h/2$
 - 当 k_h 为偶数：在上侧填充 $\lceil p_h/2 \rceil$, 在下侧填充 $\lfloor p_h/2 \rfloor$

步幅

- 步幅主要解决计算效率问题，并充当下采样（Downsampling）的角色
- 当Stride=1时，相邻两次卷积操作的感受野（Receptive Field）高度重叠，提取到的特征也可能高度相似和冗余
 - 同时，生成与输入同样大的高分辨率特征图，会给后续层带来巨大的计算和内存压力
- 增大步幅 (Stride > 1) 通过将步幅设置为2或更大，卷积核每次“跳”着滑动
 - 降低输出尺寸（下采样）。步幅越大，输出的特征图尺寸越小。直接降低网络的计算量
 - 增大感受野。下一层一个像素所对应的原始输入区域（即感受野）会变得更大
- 与池化层（Pooling）的关系
 - 增大步幅的卷积和池化层（如Max Pooling）都可以实现下采样
 - 现代网络设计中（如ResNet），使用步幅为2的卷积来代替池化层进行下采样，下采样的同时学习到有用的空间特征

步幅

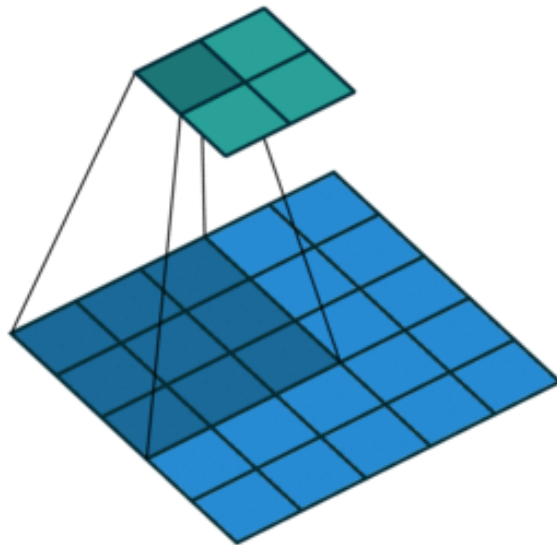
➤ 步幅是指行/列的滑动步长

➤ 例：高度3 宽度2 的步幅

Input					Kernel		Output	
0	0	0	0	0	*	=	0	8
0	0	1	2	0			6	8
0	3	4	5	0				
0	6	7	8	0				
0	0	0	0	0				

$$0 \times 0 + 0 \times 1 + 1 \times 2 + 2 \times 3 = 8$$

$$0 \times 0 + 6 \times 1 + 0 \times 2 + 0 \times 3 = 6$$



步幅

- 给出高度 s_h 和宽度 s_w 的步幅，输出形状是

$$\lfloor (n_h - k_h + p_h + s_h)/s_h \rfloor \times \lfloor (n_w - k_w + p_w + s_w)/s_w \rfloor$$

- 如果 $p_h = k_h - 1$, $p_w = k_w - 1$

$$\lfloor (n_h + s_h - 1)/s_h \rfloor \times \lfloor (n_w + s_w - 1)/s_w \rfloor$$

- 如果输入高度和宽度可以被步幅整除

$$(n_h/s_h) \times (n_w/s_w)$$

多个输入和输出通道

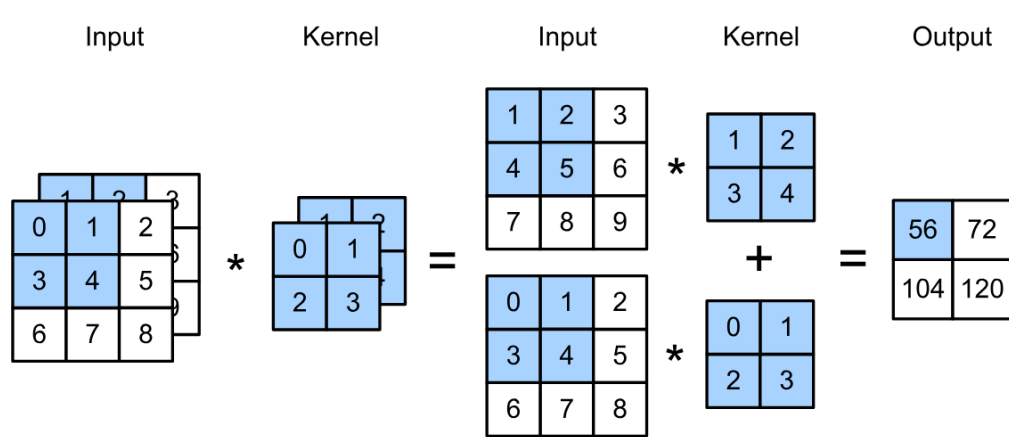
多个输入通道

- 彩色图像可能有 RGB 三个通道
- 转换为灰度会丢失信息



多个输入通道

- 每个通道都有一组卷积核，最后的输出结果是所有通道卷积结果的和
 - 卷积核个数为输入的通道数【所有通道都起作用】



$$\begin{aligned} & (1 \times 1 + 2 \times 2 + 4 \times 3 + 5 \times 4) \\ & + (0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3) \\ & = 56 \end{aligned}$$

多个输入通道

➤ $\mathbf{X}: c_i \times n_h \times n_w$, 输入

➤ c_i 个 $n_h \times n_w$ 的输入

➤ $\mathbf{W}: c_i \times k_h \times k_w$, 卷积核

➤ c_i 个 $k_h \times k_w$ 卷积核

➤ $\mathbf{Y}: m_h \times m_w$, 输出

$$\text{➤ } \mathbf{Y} = \sum_{i=0}^{c_i} \mathbf{X}_{i,:} \star \mathbf{W}_{i,:}$$

多个输出通道

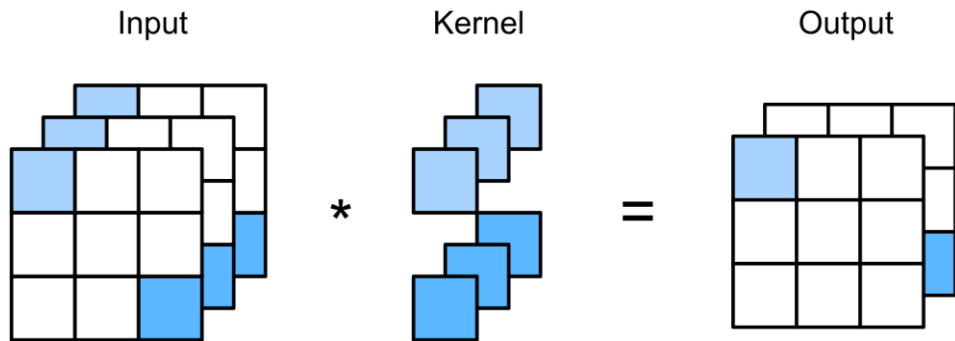
➤ 可以有多组卷积核，每组卷积核产生一个输出通道

➤ 输入 $\mathbf{X}$: $c_i \times n_h \times n_w$

➤ 内核 $\mathbf{W}$: $c_o \times c_i \times k_h \times k_w$

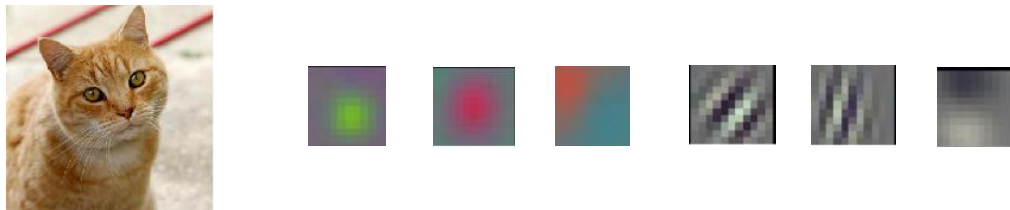
➤ 输出 $\mathbf{Y}$: $c_o \times m_h \times m_w$

➤ $\mathbf{Y}_{i,:,:} = \mathbf{X} \star \mathbf{W}_{i,:,:,:}$ for $i = 1, \dots, c_o$



多个输入和输出通道

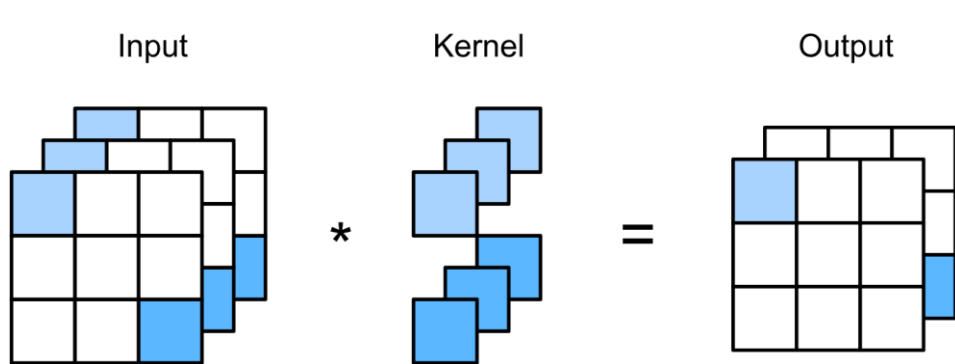
- 每个输出通道可以识别特定模式



- 输入通道内核识别并组合输入中的模式

1x1 卷积层

- $k_h = k_w = 1$ 是一个受欢迎的选择。
- 它不识别空间模式（相邻元素间没有相互作用），只融合通道
- 相当于具有输入 $c_i n_h n_w$ 和重量 $c_o \times c_i$ 的稠密层



汇聚(pooling)

为什么要汇聚？

- 卷积特征图 (Feature Map) 虽然包含了丰富的局部特征信息，但存在两大问题
 - 位置过敏感性 (Sensitivity to Feature Location)
 - 网络需要额外花费大量的参数去学习微小平移的“不变性”，效率很低
 - 计算冗余性 (Computational Redundancy)
 - 一个典型的卷积层输出的特征图尺寸依然很大。相邻区域的特征往往高度相似。高分辨率、高冗余度的特征图进入下一层，会带来巨大的计算量和参数量，且极易导致过拟合
- 因此，汇聚操作的核心目标
 - 引入不变性 (Introduce Invariance): 使模型对特征的微小位移、形变或旋转不那么敏感，提升模型的泛化能力

2-D 最大汇聚

➤ 最大汇聚 “在这个局部区域内，我们探测到的最强的特征信号是什么？”

➤ 只保留最显著的特征，并丢弃掉较弱的信号

➤ 能保留纹理、边缘等最关键的信息，同时对噪声不敏感

➤ 更好地保留特征的“激活强度”，对分类任务通常更有利

0	1	2
3	4	5
6	7	8

2 x 2 Max
Pooling

4	5
7	8

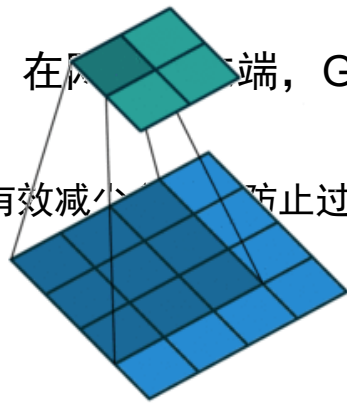
$$\max(0,1,3,4) = 4$$

➤ 平均池化 (Average Pooling): 在这个局部区域内，特征的平均响应强度是多少？”

➤ 它将区域内的所有信息都考虑进来，形成一个更平滑的表达。

➤ 在现代网络中常见的全局平均池化 (Global Average Pooling, GAP)。在网络末端，GAP将整个 $H \times W$ 的特征图直接汇聚成一个 1×1 的值

➤ 一种极强的降维和正则化手段，常用于替代传统网络末端的全连接层，能有效减少过拟合



2-D 最大汇聚

➤ 返回滑动窗口中的最大值

垂直边缘检测

```
[[1. 1. 0. 0. 0.  
 [1. 1. 0. 0. 0.  
 [1. 1. 0. 0. 0.  
 [1. 1. 0. 0. 0.]
```

卷积输出

```
[[ 0.  1.  0.  0.  0.  
 [ 0.  1.  0.  0.  0.  
 [ 0.  1.  0.  0.  0.  
 [ 0.  1.  0.  0.  0.]
```

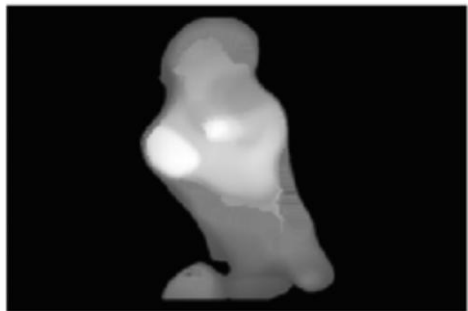
2-D 最大池化

```
[[ 1.  1.  1.  0.  
 [ 1.  1.  1.  0.  
 [ 1.  1.  1.  0.  
 [ 1.  1.  1.  0.]
```

可容1像素移位

平均汇聚层

- 最大汇聚层：每个窗口中最强的模式信号
- 平均汇聚层：
 - 将最大池化层中的“最大”操作替换为“平均”
 - 窗口中的平均信号强度



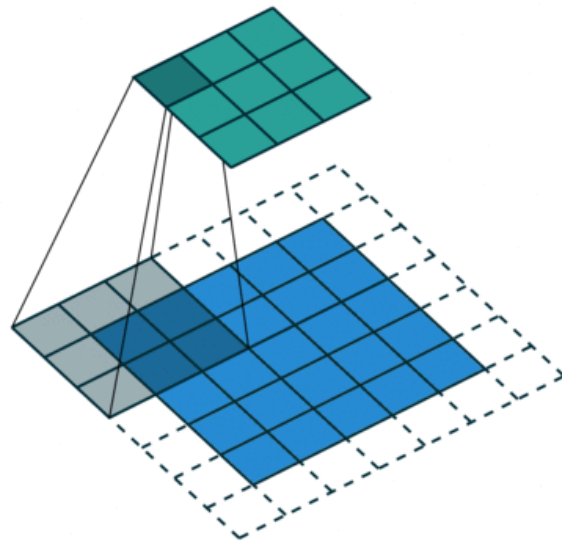
最大池化层



平均池化层

汇聚层 - 填充，步幅和多个通道

- 池化层与卷积层类似，都具有填充和步幅
- 没有可学习的参数
- 在每个输入通道应用池化层以获得相应的输出通道
- $\# \text{输出通道} = \# \text{输入通道}$



总结

➤ 卷积层

- 与稠密层相比，模型容量降低
- 有效地检测空间模式
- 计算复杂度高
- 通过填充，步幅和通道控制输出形状

➤ 最大 / 平均汇聚层

- 提供一定程度的平移不变性