

卷积神经网络

一、卷积

1.1 定义

卷积是当把一个函数“翻转”并移位 x 时，测量 和之间的重叠。

卷积 $f * g(x)$ 可以表示为：

$$f * g(x) = \int f(z)g(x - z)dz \quad (1 - 1)$$

对象为离散的形式为：求和的形式

$$(f * g)(i) = \sum_i f(a)g(i - a) \quad (1 - 2)$$

对于二维的张量：

$$(f * g)(i, j) = \sum_a \sum_b f(a, b)g(i - a, j - b) \quad (1 - 3)$$

1.2 卷积的计算过程

通常在卷积层中使用更加直观的互相关 (cross-correlation) 运算。在二维卷积层中，一个二维输入数组和一个二维核 (kernel) 数组通过互相关运算输出一个二维数组。

为了得到卷积运算的输出，我们只需将核数组左右翻转并上下翻转，再与输入数组做互相关运算。

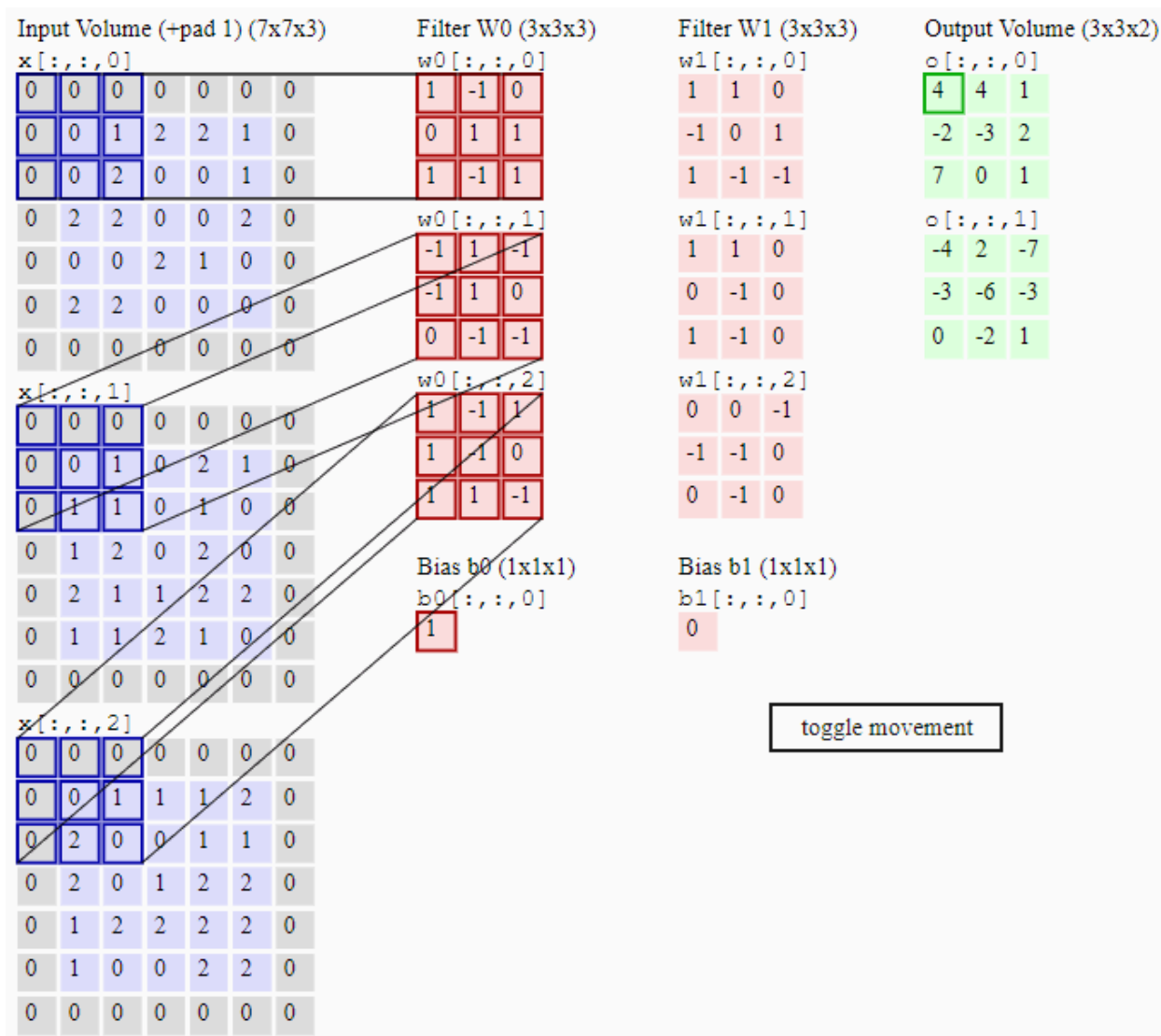


Figure 1 - 1: 卷积的计算过程

1.3 感受野

在卷积神经网络中，感受野是指在输入图像上，一个神经元（或特征图中的一个元素）可以看到或响应的区域大小。更具体地说，感受野是指输入图像的一个区域，这个区域中的像素会影响到卷积层或池化层中特定位置的输出值。感受野的计算取决于卷积层和池化层的排列方式、卷积核大小、步幅和填充方式。

感受野的计算可以表示为：

$$R_n = R_{n-1} + (k_n - 1) \cdot s_{n-1} \quad (1 - 4)$$

- R_n 是第 n 层的感受野大小
- R_{n-1} 是第 $n - 1$ 层的感受野大小
- k_n 是第 n 层的卷积核大小
- s_{n-1} 是第 $n - 1$ 层的步幅大小

1.4 输出大小

$$\begin{cases} \text{height}_{\text{out}} = \frac{\text{height}_{\text{in}} - \text{height}_{\text{kernel}} + 2 \times \text{padding}}{\text{stride}} + 1 \\ \text{width}_{\text{out}} = \frac{\text{width}_{\text{in}} - \text{width}_{\text{kernel}} + 2 \times \text{padding}}{\text{stride}} + 1 \end{cases} \quad (1 - 5)$$

全连接层的计算实例

	Batch		Height		Width		In Depth		Out Depth
Input	256	×	32	×	32	×	512		
Kernel			32	×	32	×	512	×	4096
Output	256	×	1	×	1			×	4096

Figure 1 - 2: 全连接层的计算实例

1.5 padding

padding 的计算可以使用下式：

设卷积核窗口形状是 $k_h \times k_w$

$$\begin{aligned} p_h &= k_h - 1 \\ k_w &= k_w - 1 \end{aligned} \quad (1 - 6)$$

假设这里 k_h 是奇数，我们会在高的两侧分别填充 $\frac{p_h}{2}$ 行。如果 k_h 是偶数，一种可能是在输入的顶端一侧填充 $\frac{p_h}{2}$ 行，而在底端一侧填充 $\frac{p_h}{2}$ 行。在宽的两侧填充同理。这样输入和输出具有相同的高和宽。

1.6 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，即为“平移不变性”。

1.7 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，这就是“局部性”原则。最终，可以聚合这些局部特征，以在整个图像级别进行预测。

1.8 多输入通道的情况

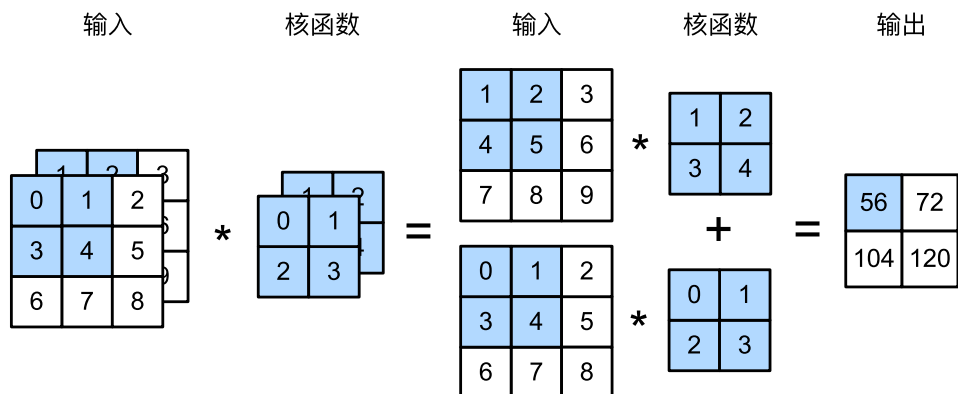


Figure 1 - 3: 多通道情况 卷积的计算过程

$$(1 \times 1 + 2 \times 2 + 4 \times 3 + 5 \times 4) + (0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3) = 56$$

$$(1 \times 2 + 2 \times 3 + 3 \times 4 + 4 \times 5) + (0 \times 1 + 1 \times 2 + 2 \times 3 + 3 \times 4) = 72$$

二、池化层（汇聚层）

降低卷积层对位置的敏感性，同时降低对空间降采样表示的敏感性。

不同于卷积层里计算输入和核的互相关性，池化层直接计算池化窗口内元素的最大值或者平均值。该运算也分别叫做最大池化或平均池化。

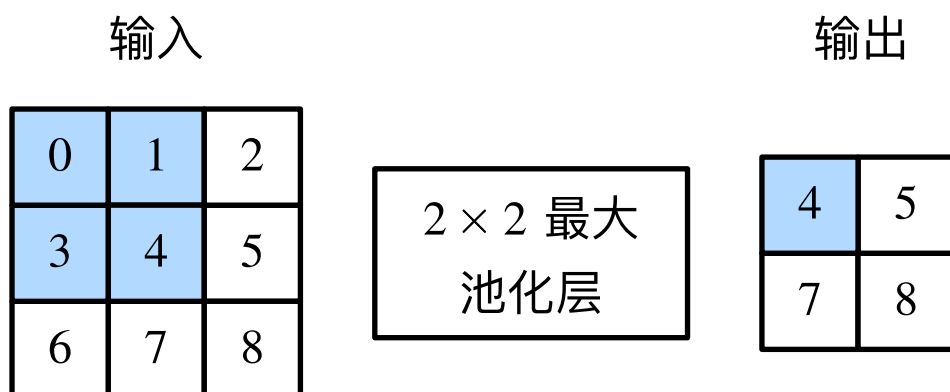


Figure 2 - 4: 池化

$$\begin{aligned} \max(0, 1, 3, 4) &= 4, \\ \max(1, 2, 4, 5) &= 5, \\ \max(3, 4, 6, 7) &= 7, \\ \max(4, 5, 7, 8) &= 8 \end{aligned} \quad (2 - 1)$$

池化层也可以在输入的高和宽两侧的填充并调整窗口的移动步幅来改变输出形状。

在处理多通道输入数据时，池化层对每个输入通道分别池化，而不是像卷积层那样将各通道的输入按通道相加。这意味着池化层的输出通道数与输入通道数相等。

三、卷积神经网络

卷积层尝试解决这两个问题。一方面，卷积层保留输入形状，使图像的像素在高和宽两个方向上的相关性均可能被有效识别；另一方面，卷积层通过滑动窗口将同一卷积核与不同位置的输入重复计算，从而避免参数尺寸过大。

3.1 LeNet 模型

LeNet 分为卷积层块和全连接层块两个部分。

卷积层块里的基本单位是卷积层后接最大池化层：卷积层用来识别图像里的空间模式，如线条和物体局部，之后的最大池化层则用来降低卷积层对位置的敏感性。卷积层块由两个这样的基本单位重复堆叠构成。在卷积层块中，每个卷积层都使用 5×5 的窗口，并在输出上使用 sigmoid 激活函数。第一个卷积层输出通道数为 6，第二个卷积层输出通道数则增加到 16。这是因为第二个卷积层比第一个卷积层的输入的高和宽要小，所以增加输出通道使两个卷积层的参数尺寸类似。卷积层块的两个最大池化层的窗口形状均为 2×2 ，且步幅为 2。由于池化窗口与步幅形状相同，池化窗口在输入上每次滑动所覆盖的区域互不重叠。

卷积层块的输出形状为(批量大小, 通道, 高, 宽)。当卷积层块的输出传入全连接层块时，全连接层块会将小批量中每个样本变平 (flatten)。

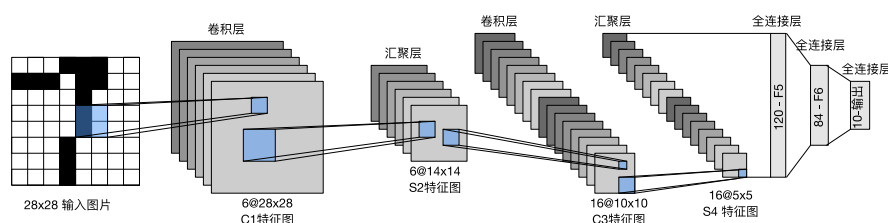


Figure 3 - 5: Lenet

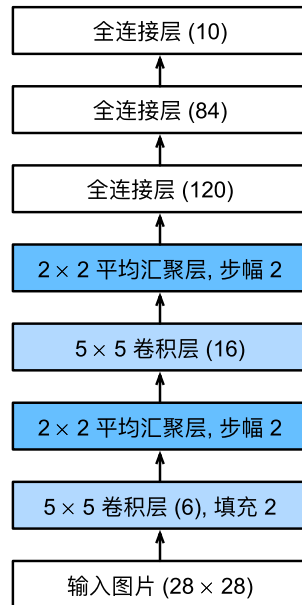


Figure 3 - 6: Lenet

四、 引用

[1] 【深度学习】一文搞懂卷积神经网络（CNN）的原理（超详细）卷积神经网络原理-CSDN 博客[EB/OL]. [2024-07-03]. https://blog.csdn.net/AI_dataloads/article/details/133250229.

[2] 【深度学习入门】——亲手实现图像卷积操作 输入是图像和使用的卷积核-CSDN 博客[EB/OL]. [2024-07-03]. <https://blog.csdn.net/rocling/article/details/103831994>.

[3] 5. 卷积神经网络 — 《动手学深度学习》 文档[EB/OL]. [2024-07-03]. https://zh-v1.d2l.ai/chapter_convolutional-neural-networks/index.html.

[4] 6. 卷积神经网络 — 动手学深度学习 2.0.0 documentation[EB/OL]. [2024-07-03]. https://zh.d2l.ai/chapter_convolutional-neural-networks/index.html.