卷积神经网络

徐晨鸥 2019.1.10

卷积网络指的是至少在网络的一层中使用卷积运算来替代一般矩阵乘法运算的神经网络。

卷积运算

宇宙飞船在时刻t的位置为x(t)

假设位置传感器受到了一定程序的噪声干扰。为了得到位置的低噪声估计，对测量结果进行加权平均

（对于最近的测量结果赋予更高的权重，加权函数为w(a)，a表示测量结果距离当前时刻的时间间隔）

得到一个平滑估计函数s：

$$s(t) = \int x(a)w(t-a)da$$

这种运算就称为卷积，一般用星号表示。

卷积运算

$$s(t) = \int x(a)w(t-a)da$$

参数x : 输入

参数w : 核函数

输出s : 又称作**特征映射**

一次在多个维度进行卷积运算

把一张二维的图像I作为输入，需要一个二维的核K

$$s(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

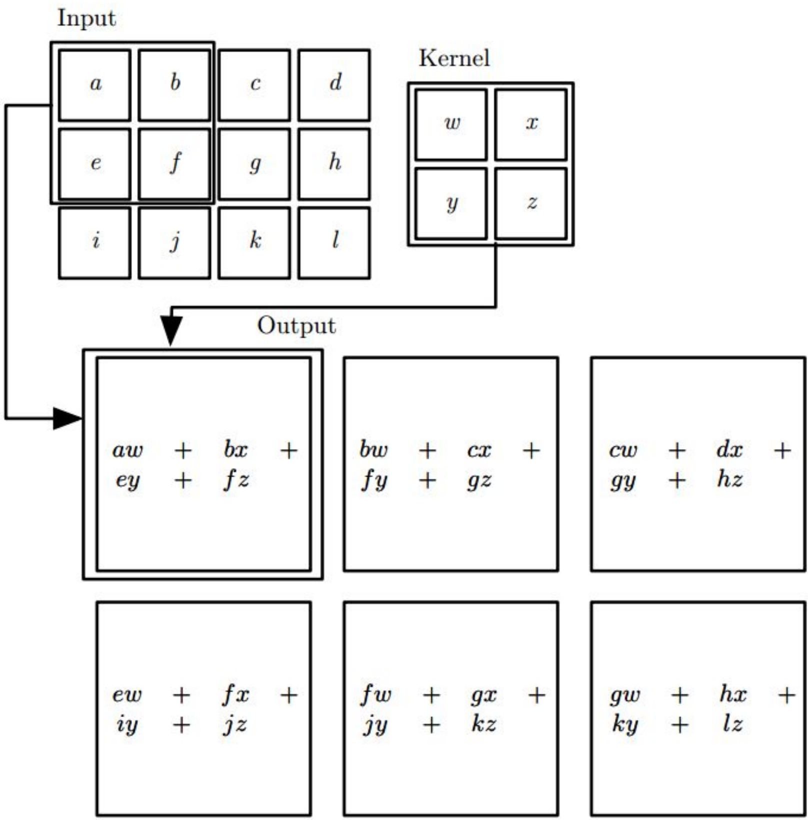


图 9.1: 一个 2 维卷积的例子（没有对核进行翻转）。我们限制只对核完全处在图像中的位置进行输出，在一些上下文中称为“有效”卷积。我们用画有箭头的盒子来说明输出张量的左上角元素是如何通过对输入张量相应的左上角区域应用核进行卷积得到的。

任何一个使用矩阵乘法但是不依赖矩阵结构的特殊性质的神经网络算法，都适用于卷积运算，并且不需要对神经网络做出大的修改。

稀疏交互、参数共享、等变表示

稀疏交互

输入的图像可能包含成千上万个像素点，但是我们可以通过只占用几十个到上百个像素点的核来检测一些小的有意义的特征。（如图像的边缘）

意味着需要存储的参数更少，减少了模型的存储需求，也提高了它的统计效率。

稀疏交互

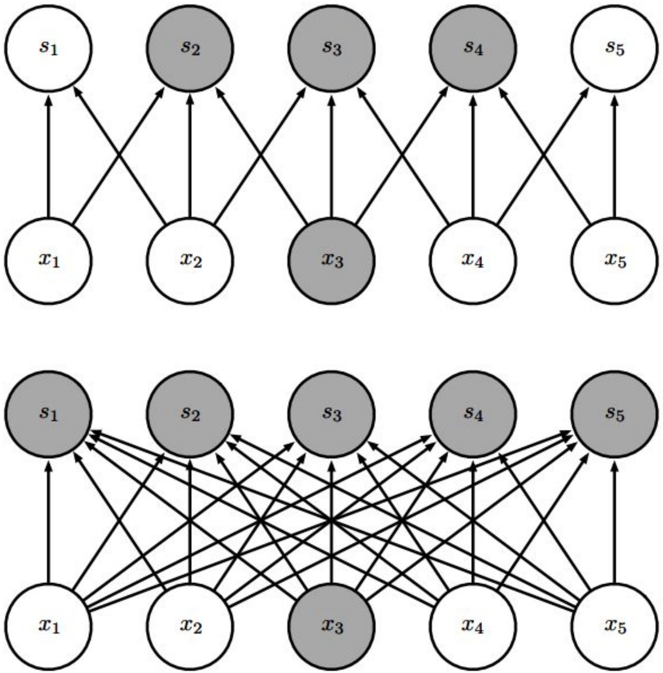


图 9.2: 稀疏连接，对每幅图从下往上看。我们强调了一个输入单元 x_3 以及在 $\boldsymbol{s}$ 中受该单元影响的输出单元。*(上)* 当 $\boldsymbol{s}$ 是由核宽度为 3 的卷积产生时，只有三个输出受到 $\boldsymbol{x}$ 的影响[2]。*(下)* 当 $\boldsymbol{s}$ 是由矩阵乘法产生时，连接不再是稀疏的，所以所有的输出都会受到 x_3 的影响。

稀疏交互

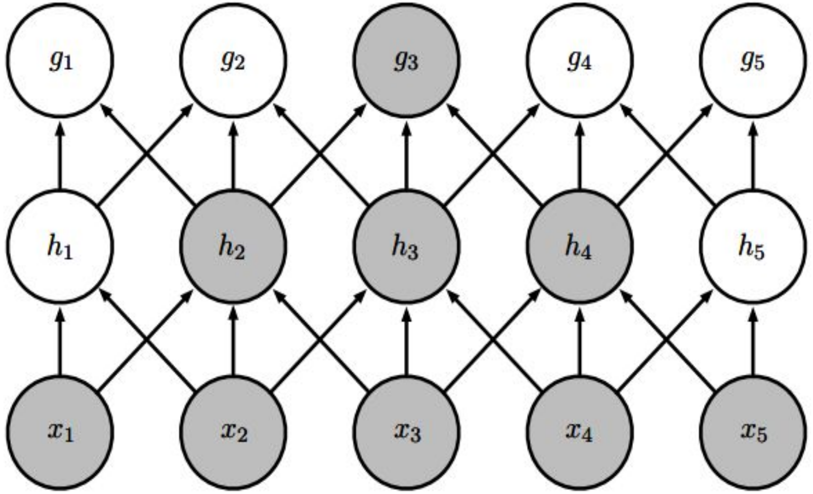


图 9.4: 处于卷积网络更深的层中的单元，它们的接受域要比处在浅层的单元的接受域更大。如果网络还包含类似步幅卷积（图 9.12）或者池化（第 9.3 节）之类的结构特征，这种效应会加强。这意味着在卷积网络中尽管直接连接都是很稀疏的，但处在更深的层中的单元可以间接地连接到全部或者大部分输入图像。

参数共享

一个模型的多个函数使用相同的参数。

传统神经网络：计算一层输出的时候，权重矩阵的每一个元素只使用一次。

卷积神经网络：核的每一个元素都作用在输入的每一位置上。

卷积运算中的参数共享保证我们只需要学习一个参数集合，而不是对于每一个位置都学习一个单独的参数集合。同样减少了模型的存储需求降低至了k个参数。

参数共享

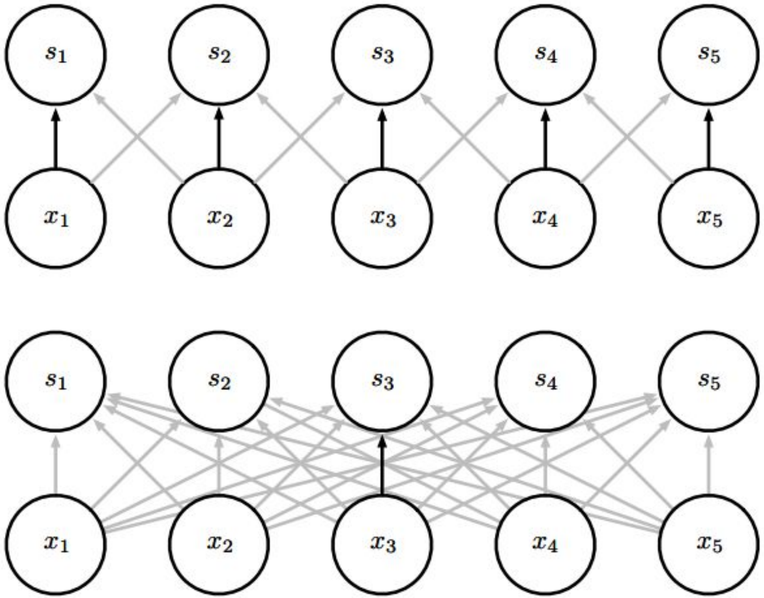


图 9.5: 参数共享。黑色箭头表示在两个不同的模型中使用了特殊参数的连接。*(上)* 黑色箭头表示在卷积模型中对 3 元素核的中间元素的使用。因为参数共享，这个单独的参数被用于所有的输入位置。*(下)* 这个单独的黑色箭头表示在全连接模型中对权重矩阵的中间元素的使用。这个模型没有使用参数共享，所以参数只使用了一次。

等变表示

如果一个函数满足输入改变，输出也以同样的方式改变这一性质，就是等变的。

特别的，函数f(x)与g(x)满足下式，f(x)对于变换g具有等变性。

$$f(g(x)) = g(f(x))$$

等变表示

例子：

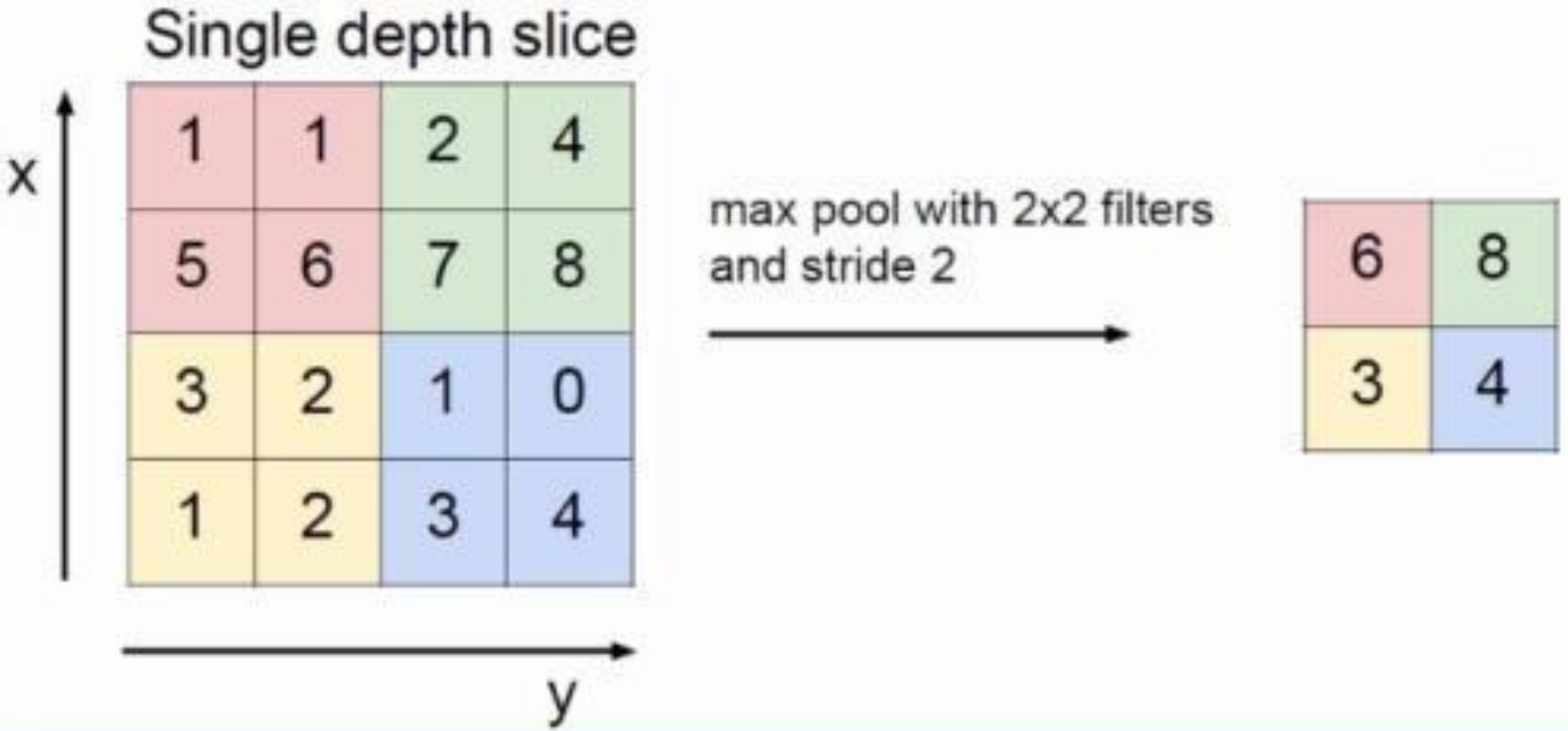
如果我们先对I进行向右移动一个单位，再进行卷积得到的结果，和先对I卷积再对输出右移一个单位得到的结果是一样的。

应用：

把输入的一个事件向后延时，在输出中仍然有完全相同的表示。

池化

MAX POOLING


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

卷积网络的典型层

第一层：并行计算多个卷积产生一组线性激活响应。
第二层：每一个线性激活响应通过一个非线性的激活函数。
第三层：使用**池化函数**调整输出

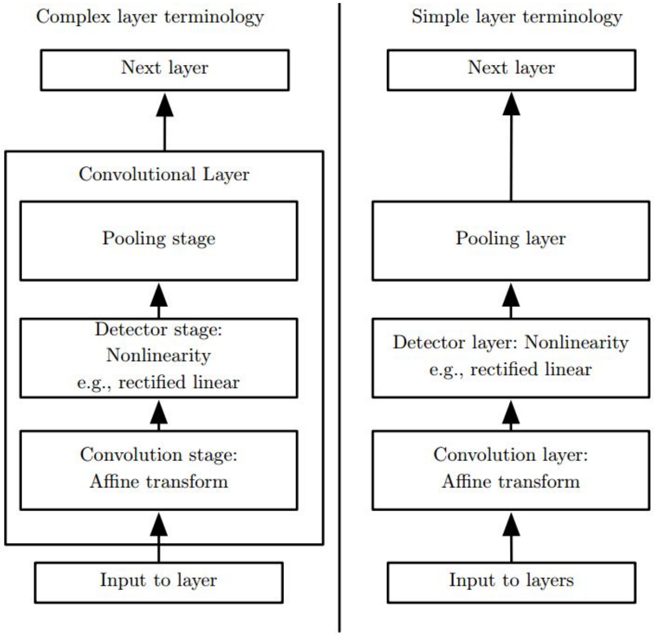


图 9.7: 一个典型卷积神经网络层的组件。有两组常用的术语用于描述这些层。*(左)* 在这组术语中，卷积网络被视为少量相对复杂的层，每层具有许多"级"。在这组术语中，核张量与网络层之间存在一一对应关系。在本书中，我们通常使用这组术语。*(右)* 在这组术语中，卷积网络被视为更多数量的简单层；每一个处理步骤都被认为是一个独立的层。这意味着不是每一"层"都有参数。

池化

当输入作出少量平移时，池化能够帮助输入的表示近似**不变**。

局部平移不变性是很有用的性质。例如：判定一张图像是否包含人眼的时候，不需要知道眼睛的精确位置。只需要保证一只眼睛在脸的左边，一只在右边。

池化综合了全部邻居的反馈

使得池化单元少于探测单元成为可能。（可以通过综合池化区域的k个像素的统计特征而不是单个像素的统计特征来实现。）

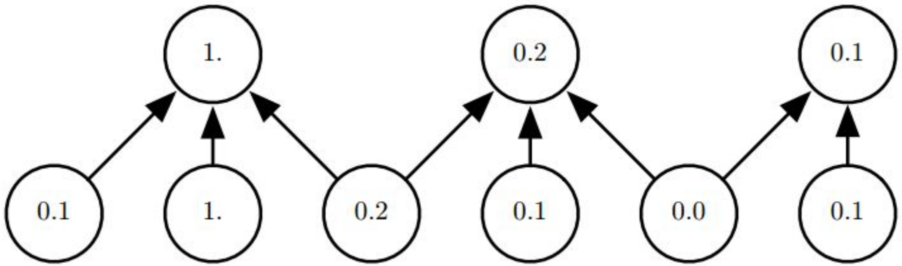


图 9.10: 带有降采样的池化。这里我们使用最大池化，池的宽度为三并且池之间的步幅为二。这使得表示的大小减少了一半，减轻了下一层的计算和统计负担。注意到最右边的池化区域尺寸较小，但如果我们不想忽略一些探测单元的话就必须包含这个区域。

基本卷积函数的变体

卷积运算通常是并行的

单个核的卷积只能提取一种类型的特征，尽管它作用在多个空间位置上。

有效卷积

无论怎么样都不进行填充的极端情况。

卷积核只能访问图像中能够完全包含整个核的位置。

输入图像宽度是m，核宽度k，则输出的宽度会变成m-k+1。

相同卷积

只进行足够的零填充来保持输出和输入具有相同的大小。

一般零填充的最优数量是“有效卷积”和“相同卷积”之间的值。

全卷积

进行足够多的零填充，使得每个像素在每个方向上恰好被访问了k次。

最终输出图像的宽度为m+k-1

零填充对网络大小的影响

右图不是用零填充，每层缩小5个像素。

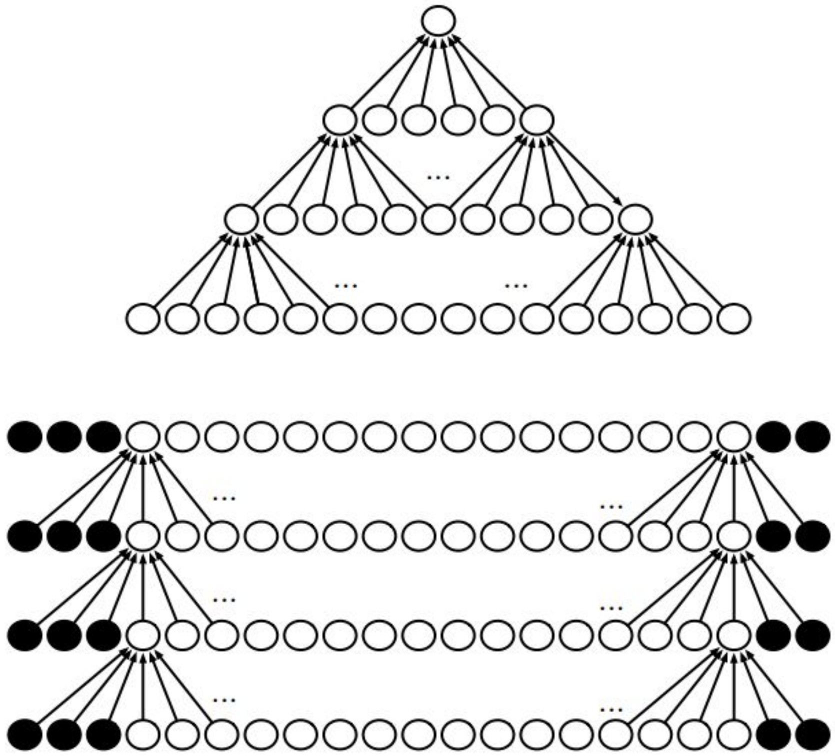


图 9.13: 零填充对网络大小的影响。考虑一个卷积网络，每层有一个宽度为六的核。在这个例子中，我们不使用任何池化，所以只有卷积操作本身缩小网络的大小。(上) 在这个卷积网络中，我们不使用任何隐含的零填充。这使得表示在每层缩小五个像素。从十六个像素的输入开始，我们只能有三个卷积层，并且最后一层不能移动核，所以可以说只有两层是真正的卷积层。可以通过使用较小的核来减缓收缩速率，但是较小的核表示能力不足，并且在这种结构中一些收缩是不可避免的。(下) 通过向每层添加五个隐含的零，我们防止了表示随深度收缩。这允许我们设计一个任意深的卷积网络。

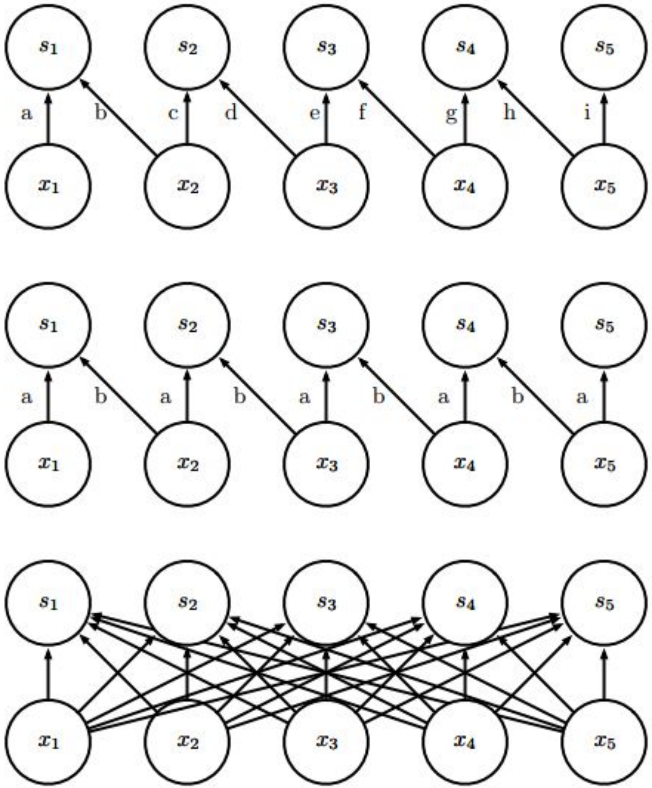


图 9.14: 局部连接，卷积和全连接的比较。*(上)* 每一小片（接受域）有两个像素的局部连接层。每条边用唯一的字母标记，来显示每条边都有自身的权重参数。*(中)* 核宽度为两个像素的卷积层。该模型与局部连接层具有完全相同的连接。区别不在于哪些单元相互交互，而在于如何共享参数。局部连接层没有参数共享。正如用于标记每条边的字母重复出现所指示的，卷积层在整个输入上重复使用相同的两个权重。*(下)* 全连接层类似于局部连接层，它的每条边都有其自身的参数（在该图中用字母明确标记的话就太多了）。然而，它不具有局部连接层的连接受限的特征。

无监督训练得到卷积核

1.简单初始随机

2.手动设计，每个核在一个特定的方向和尺度检测边缘

3.使用无监督的标准学习核(例:K均值聚类算法用于小图像块，使用每个学得的中心作为卷积核)