摘　要

本文是我在学习深度学习有关知识时的笔记 。

关键词：深度学习

Abstract

This article is my notes while learning deep learning .

Keywords: deep learning

目 录

第 1 章 卷积层 …… 1

1.1 卷积层的特点 …… 1

1.2 卷积运算 …… 1

1.2.1 一维离散卷积 …… 1

1.2.2 一维连续卷积 …… 3

1.2.3 二维离散卷积 …… 3

参考文献 …… 7

图形列表

1.1 乘法的直观表示 …… 2
1.2 Valid Convolution …… 3
1.3 二维离散 Valid 卷积 …… 4
1.4 二维离散 Full 卷积 …… 5
1.5 卷积运算中的偏置 …… 5

表格列表

符号列表

算子

Symbol	Description
$\int$	Indefinite integral
$\sum$	Sum
$*$	Convolution

缩写

CNN	Convolutional Neural Network

第 1 章　卷积层

1.1　卷积层的特点

全连接层存在什么问题呢？那就是数据的形状被“忽视”了。比如，输入数据是图像时，图像通常是高、长、通道方向上的 3 维形状。但是，向全连接层输入时，需要将 3 维数据拉平为 1 维数据。全连接层会忽视形状，将全部的输入数据作为相同的神经元（同一维度的神经元）处理，所以无法利用与形状相关的信息。

而卷积层可以保持形状不变。当输入数据是图像时，卷积层会以 3 维数据的形式接收输入数据，并同样以 3 维数据的形式输出至下一层。因此在 CNN 中，可以（有可能）正确理解图像等具有形状的数据。

CNN 中，有时将卷积层的输入输出数据称为特征图（feature map）。其中，卷积层的输入数据称为输入特征图（input feature map），输出数据称为输出特征图（output feature map）[1]。

1.2　卷积运算

卷积层进行的处理就是卷积运算。卷积运算相当于图像处理中的“滤波器运算”。

“卷积是一种特殊的加权求和”。

1.2.1　一维离散卷积

1.2.1.1　引入——多项式乘法

设多项式：

$$P(x) = a_n x^n + a_{n-1} x^{n+1} + \cdots + a_1 x + a_0 \tag{1.1}$$

$$Q(x) = b_m x^m + b_{m-1} b^{m+1} + \cdots + b_1 x + b_0 \tag{1.2}$$

那么，

$$
\begin{aligned}
R(x) &= P(x) \times Q(x) \\
&= \left(\sum_{i=0}^{n} a_i x^i\right) \cdot \left(\sum_{i=0}^{m} b_i x^i\right) \\
&= \sum_{i=0}^{m+n} c_i x^i \\
where, \quad c_i &= \sum_k a_k b_{i-k}
\end{aligned}
\tag{1.3}
$$

接下来，我们以 $P(x) = x^4 - x^3 + 2x - 4, Q(x) = x^2 - 2x + 2$ 为例。其计算结果为：

$$R(x) = x^6 - 3x^5 + 4x^4 - 8x^2 + 12x - 8$$

其乘法的直观表示如图所示：

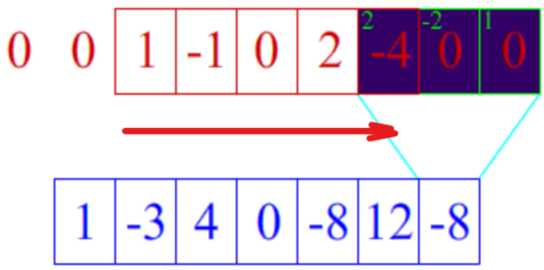

图 1.1 乘法的直观表示

在图1.1中，需要注意 $Q(x)$ 的系数是反转哒。

1.2.1.2 一维离散卷积的定义

定义 1.1. 如果 $\{a_n\}$ $\{b_m\}$ 是两个数列，那么二者的卷积为：

$$c_i = (a * b)_i = \sum_k a_k b_{i-k}$$

其中，$\{a_n\}$ 是被卷积数列，$\{b_m\}$ 是卷积核。

在上面的例子中，如果下标不合法，则用 0 来代替——这是完全补 0 的卷积，也叫 **Full 卷积**。类似的，我们也可以定义如图1.2所示的合法卷积，也叫 **Valid 卷积**。

同样的，我们也可以定义 **Same 卷积**——卷积后的数列长度和被卷积数列一样长。具体操作仅仅比 Full 卷积左右各少补一半的 0 就可以了。

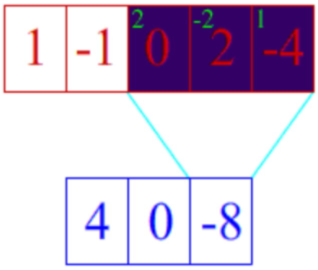

图 1.2 Valid Convolution

在此基础上，卷积核并不一定是在被卷积数列上一格一格地滑动，可以两格两格，甚至半格半格地滑动，由此派生出无数种可能[2]。

1.2.2　一维连续卷积

上文中涉及到的都是离散的数列，如果改为连续函数 $f(x)$ 与 $g(x)$ 的卷积，那么:

$$h(x) = (f * g)(x) = \int f(\tau)\, g(x-\tau)\, d\tau \tag{1.4}$$

下面我们举一个具体的例子。

假设你是一个算法工程师，过着 996 的死亡生活，每天除了调参就是调参。你每天工作效率关于时间 t 的函数为 $f(t)$（显然，如果你准时上班，正点下班，那么 $f(t)$ 在 $[0,9)\bigcup[21,24)$ 上是没有定义的）。并且工作成果的产出关于时间 t 的函数为 $g(x)$，那么你忙活一天，给团队获得的实际收益则是:

$$h(24) = \int_{9}^{21} f(\tau)\, g(24-\tau)\, d\tau$$

1.2.3　二维离散卷积

上述的卷积都是一维的卷积核在一维的被卷积张量上滑动。如果被卷积张量和卷积核是二维、三维，甚至四维及以上，我们同样也可以有相同的定义——向这些维度分别滑动，分别做卷积，最后求和即可。

1.2.3.1　二维离散卷积

Valid 卷积　如图1.3所示，将各个位置上滤波器的元素和输入的对应元素相乘，然后再求和（有时将这个计算称为乘积累加运算）。然后，将这个结果保存到输出的对应位置。将这个过程在所有位置都进行一遍，就可以得到卷积运算的输出。

Full 卷积　如图所示。

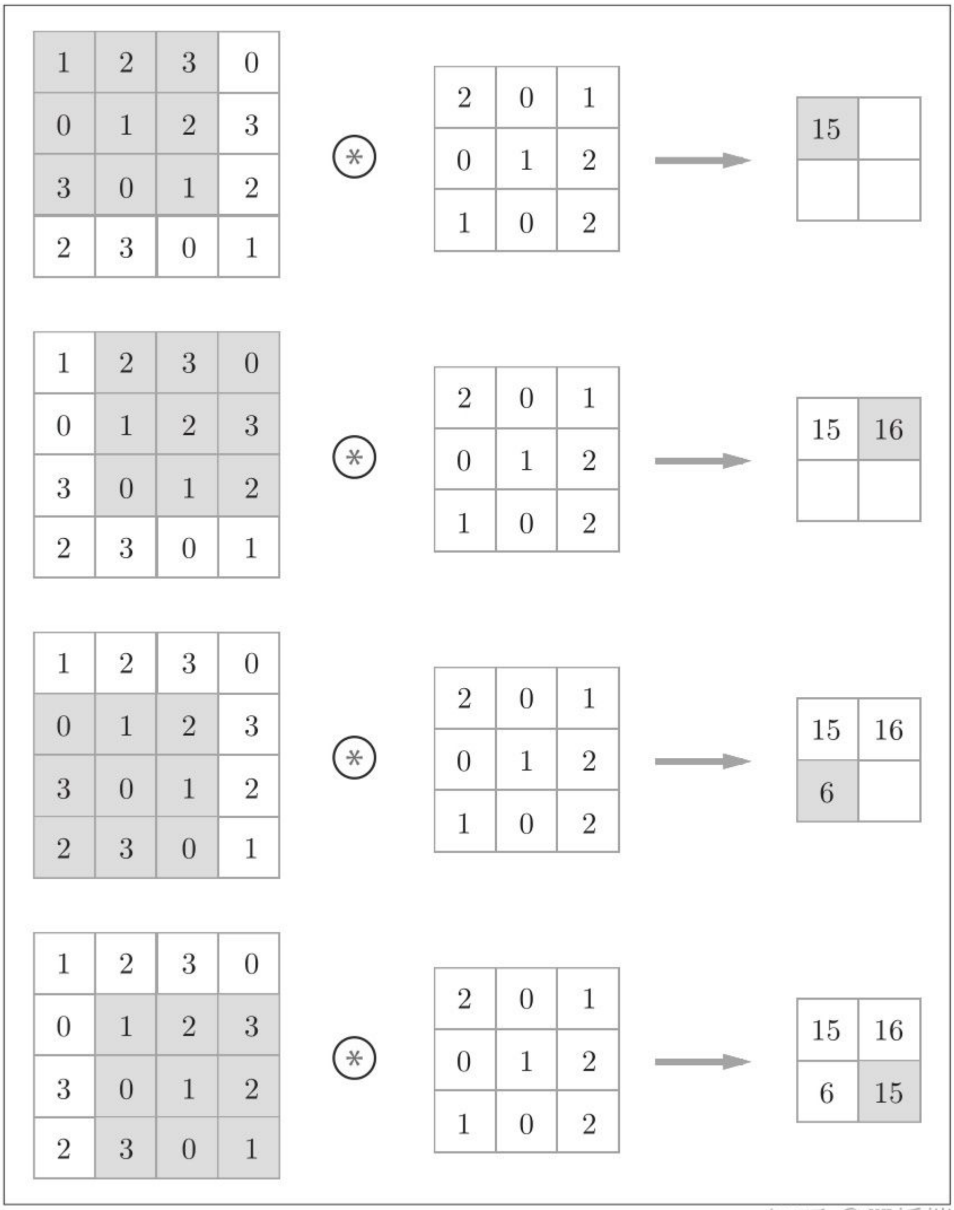

图7-4　卷积运算的计算顺序

图 1.3 二维离散 Valid 卷积

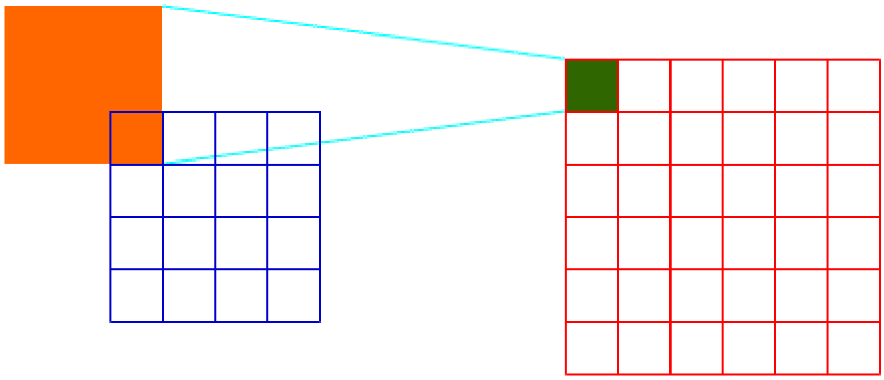

图 1.4 二维离散 Full 卷积

1.2.3.2　偏置

有些时候，在全连接的神经网络中，除了权重参数，还存在偏置。包含偏置的卷积运算的处理流如图1.5所示。

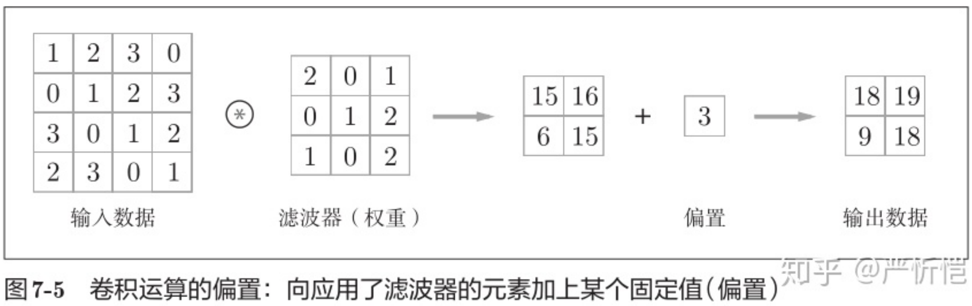

图 1.5 卷积运算中的偏置

参考文献

[1] 严忻恺. 深度学习入门-卷积神经网络（一）卷积层 [EB/OL]. 知乎, 2020. https://zhuanlan.zhihu.com/p/259751387.

[2] 刘冬煜. 卷积运算是什么？[EB/OL]. 知乎, 2020. https://www.zhihu.com/question/339496491.