

《深度学习中不同类型卷积的全面介绍》总结+心得

张新鹏

October 28, 2023

1 概述:

这篇文章的内容是介绍深度学习的不同类型的卷积的工作原理（例如：2D/3D/1×1/转置/扩张(Atrous)/空间可分离/深度可分离/扁平化/分组/混洗分组卷积）

2 关键词

卷积与互相关，深度学习中的卷积（单通道版本、多通道版本），3D 卷积，1 x 1 卷积，卷积运算，转置卷积（反卷积、棋盘伪影），扩张卷积（空洞卷积），可分离卷积（空间可分离卷积、深度卷积），扁平化卷积，分组卷积，混洗分组卷积，逐点分组卷积

3 卷积和互相关

深度学习中的卷积本质上是信号/图像处理中的互相关。

卷积被定义为两个函数的乘积在一个函数反转并移位后的积分。

卷积定义为： $(f * g)(t) = \int_{-\infty}^{\infty} f(\zeta)g(t - \zeta)d\zeta$

过滤器g反转，然后沿水平轴滑动。对于每个位置，计算f 和反转的g 之间的交集面积。交集面积就是该特定位置的卷积值。

互相关被称为两个函数的滑动点积或滑动内积。相关过滤器不被反转，直接滑过函数f。f和g之间的交集面积就是互相关。

在深度学习中，卷积中的过滤器不是颠倒的。严格来说，这是互相关。本质上是执行逐元素乘法和加法。但在深度学习中，习惯上将其称为卷积。因为过滤器的权重是在训练期间学习的，如果深度学习的函数g 反转了，那训练完看起来像反转函数g。所以深度学习中不需要反转。

4 深度学习中的卷积

进行卷积的目的是从输入中提取有用的特征。

在图像处理中，可以选择多种不同的过滤器进行卷积。每种类型的过滤器有助于从输入图像中提取不同的方面或特征，例如水平/垂直/对角线边缘。

在卷积神经网络中，使用过滤器通过卷积提取不同的特征，过滤器的权重在训练期间自动学习。然后将这些提取的所有特征进行“组合”以做出决策。

进行卷积有一些优点，例如权重共享和平移不变性。卷积还考虑了像素的空间关系。这些可能非常有用，尤其是在许多计算机视觉任务中，因为这些任务通常涉及某些组件与其他组件具有一定空间关系的对象识别（例如，狗的身体通常与头、四条腿和尾巴相连）。

4.1 卷积：单通道版本

在深度学习中，卷积是逐元素乘法和加法。过滤器滑动通过输入矩阵。在每个位置，它都会进行逐元素乘法和加法。每个滑动位置都以一个数字结束。

4.2 卷积：多通道版本

在许多应用中，处理的是具有多个通道的图像（最常见的RGB图像，红绿蓝三色对应三通道）。

卷积网络层通常由多个通道（通常是数百个通道）组成。每个通道描述前一层的不同方面。

“内核”是指二维数组。

“过滤器”是指堆叠在一起的多个内核的3D 结构。对于2D 过滤器，过滤器与内核相同。但对于3D过滤器和深度学习中的大多数卷积来说，过滤器是内核的集合。

每个内核都应用于前一层的输入通道以生成一个输出通道。然后将这些通道中的每一个相加，形成一个输出通道。

将此过程视为在输入层中滑动3D 过滤矩阵。注意，输入层和过滤器具有相同的深度（通道数=内核数）。

3D 过滤器仅在图像的高度和宽度2 个方向上移动（这就是为什么这种操作被称为2D 卷积，尽管3D过滤器用于处理3D 体积数据）。在每个滑动位置，执行逐元素乘法和加法，结果是一个数字。总的来说，得到了一个输出通道。

假设输入层有 D_{in} 个通道，希望输出层有 D_{out} 个通道。需要将 D_{out} 个过滤器应用到输入层。每个过滤器都有 D_{in} 个内核。

5 3D卷积

上一节在对3D 数据执行卷积，仍然将该操作称为深度学习中的二维卷积。是3D 数据的2D 卷积。过滤器深度与输入层深度相同。过滤器深度与输入层深度相同。3D 过滤器仅在2 个方向（图像的高度和宽度）移动。

在3D 卷积中，过滤器深度小于输入层深度（内核大小_i 通道大小）。3D过滤器可以在3个方向（图像的高度、宽度、通道）上移动。在每个位置，逐元素乘法和加法提供一个数字。由于过滤器在3D 空间中滑动，因此输出数字也排列在3D 空间中。输出的是3D 数据。3D 卷积可以描述3D 空间中对象的空间关系。

6 1 × 1 卷积

将一个数字与输入层中的每个数字相乘。

H x W x D 的输入与1 x 1 x D 的过滤器1 x 1 卷积之后，输出H x W x 1的输出。

如果应用N 个这样的1 x 1 卷积，然后将结果连接在一起，可以得到一个尺寸为H x W x N 的输出层。

1 x 1 卷积的一些优点是：

降维以实现高效计算（降低了维度深度），

高效的低维嵌入或特征池（假设原始输入有200 个通道，1 x 1 卷积会将这些通道（特征）嵌入到单个通道中。）。

卷积后再次应用非线性（1 x 1 卷积之后，可以添加非线性激活（例如ReLU）。非线性允许网络学习更复杂的函数。）。

在卷积网络中，不存在“全连接层”这样的东西。只有具有1x1 卷积核和完整连接表的卷积层。

7 卷积运算

上面是处理卷积中的深度。继续讨论如何处理其他两个方向（高度和宽度）的卷积，以及重要的卷积算法。

步长：定义了内核滑动图像时的步长。步长为1 意味着内核逐像素地滑过图像。步幅为2 意味着内核通过每步移动2 个像素（即跳过1 个像素）来滑动图像。我们可以使用步幅($i=2$) 对图像进行下采样。

对于大小为*i*、内核大小为*k*、填充为*p* 和步长为*s* 的输入图像，卷积的输出图像大小为*o*：

$$O = \text{lower_bound}(\frac{i+2p-k}{s}) + 1$$

8 转置卷积（反卷积）

对于许多应用程序和许多网络架构，经常希望进行与正常卷积相反方向的转换，将低维特征图映射到高维空间。例如在自动编码器或语义分割中。（在后面的例中，语义分割首先在编码器中提取特征图，然后在解码器中恢复原始图像大小，以便能够对原始图像中的每个像素进行分类。）

传统上，可以通过应用插值方案或手动创建规则来实现上采样。神经网络等现代架构倾向于让网络本身自动学习适当的转换，无需人工干预。为了实现这一点，可以使用转置卷积。

转置卷积在文献中也称为反卷积或分数跨步卷积。值得注意的是，“反卷积”这个名称不太合适，因为转置卷积并不是信号/图像处理中定义的真正的反卷积。从技术上讲，信号处理中的反卷积反转了卷积运算。

可以用直接卷积来实现转置卷积。在 2×2 输入张量上应用具有 3×3 内核的转置卷积，使用单位步长并填充了 2×2 的零边框。输出的大小为 4×4 。通过应用奇特的填充和步幅，可以将相同的 2×2 输入图像映射到不同的图像尺寸。

计算机中用矩阵乘法实现，将展开后的输入矩阵转置后与一个小矩阵相乘可以获得更大的矩阵（原blog的图片好生动，不好描述）。

总之使用的是 $C \times \text{Large} = \text{Small} \cdot C^T \times \text{Small} = \text{Large}$ 。

8.1 棋盘伪影

棋盘伪影是由转置卷积的“不均匀重叠”造成的。输出上的每个像素与其相邻像素相比接收到的信息量不同。当过滤器大小不能被步幅整除时，转置卷积具有不均匀的重叠。这种“不均匀的重叠”在某些地方映射了比其他地方更多的颜色，从而产生了棋盘伪影。

在应用转置卷积时，可以做两件事来减少此类伪影。首先，确保使用的过滤器大小能被步幅整除，避免重叠问题。其次，可以使用步幅= 1 的转置卷积，这有助于减少棋盘效应。

9 扩张卷积（Atrous Convolution）

标准的离散卷积：

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t)$$

扩张卷积如下：

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t)$$

当 $l = 1$ 时，扩张卷积变为标准卷积。

直观上，扩张卷积通过在内核元素之间插入空格来“膨胀”内核。这个附加参数 l （扩张率）表示我们想要扩大内核的程度。

扩张卷积用于在不增加内核大小的情况下廉价地增加输出单元的感受区域，这在多个扩张卷积依次堆叠时尤其有效。

《通过扩张卷积进行多尺度上下文聚合》一文的作者用多层扩张卷积构建了一个网络，其中扩张率 k 在每一层呈指数增长。结果，由感受野呈指数增长，而参数数量仅随层数线性增长！

10 可分离卷积

10.1 空间可分离卷积

空间可分离卷积对图像的二维空间维度（即高度和宽度）进行操作。从概念上讲，空间可分离卷积将卷积分解为两个单独的操作。

例如，Sobel 内核是 3×3 内核，分为 3×1 和 1×3 内核。在卷积中， 3×3 内核直接与图像进行卷积。在空间可分离卷积中， 3×1 内核首先与图像进行卷积。然后应用 1×3 内核。在执行相同操作时，这将需要6 个而不是9 个参数。

此外，空间可分离卷积比卷积需要更少的矩阵乘法。

假设我们现在使用 $m \times m$ 内核对 $N \times N$ 图像应用卷积，步长=1，填充=0。传统卷积需要 $(N-2) \times (N-2) \times m \times m$ 次乘法。空间可分离卷积需要 $N \times (N-2) \times m + (N-2) \times (N-2) \times m = (2N-2) \times (N-2) \times m$ 次乘法。

空间可分离卷积与标准卷积的计算成本之比为：

$$\frac{2}{m} + \frac{2}{m(N-2)}$$

对于图像尺寸 N 远大于滤波器尺寸 $(N \gg m)$ 的层，该比率变为 $2 / m$ 。这意味着在这种情况下 $(N \gg m)$ 下，用 3×3 的过滤器，空间可分离卷积的计算成本是标准卷积的 $2/3$ 。

但在深度学习中很少使用。主要原因之一是并非所有内核都可以分为两个更小的内核。如果我们用空间可分离卷积替换所有传统卷积，我们就会限制自己在训练期间搜索所有可能的内核。训练结果可能不是最优的。

10.2 深度可分离卷积

深度可分离卷积在深度学习中更常用。深度可分离卷积由两个步骤组成：深度卷积和 1×1 卷积。

没有在2D 卷积中使用的单个大小为 $3 \times 3 \times 3$ 的过滤器，而是分别使用3 个内核。每个过滤器的大小为 $3 \times 3 \times 1$ 。每个内核与输入层的1 个通道进行卷积（仅1 个通道，不是所有通道！）。每个这样的卷积都提供一个大小为 $5 \times 5 \times 1$ 的图。然后将这些图堆叠在一起以创建一个 $5 \times 5 \times 3$ 的图像。将 $5 \times 5 \times 3$ 输入图像与每个 $1 \times 1 \times 3$ 内核进行卷积，提供大小为 $5 \times 5 \times 1$ 的映射。在应用128 个 1×1 卷积之后，可以得到一个大小为 $5 \times 5 \times 128$ 的层。

优势是效率高，需要操作的次数少，但是减少了卷积中的参数数量。模型容量可能会显著降低，模型可能变得次优。如果使用得当，深度可分离卷积可以在不显著损害模型性能的情况下提高效率。

11 扁平化卷积

没有应用一个标准卷积过滤器将输入层映射到输出层，而是将该标准过滤器分成3个1D过滤器。这种思想与上述空间可分离卷积中的思想类似。

$C \times Y \times X$ 的过滤器被分成 $C \times 1 \times 1$ ， $1 \times Y \times 1$ ， $1 \times 1 \times X$ 。但是，实际中的过滤器不好拆分。

12 分组卷积

用于多GPU并行化训练。

假设分两组，共 D_{out} 个通道，每个过滤器组包含 $D_{out}/2$ 个过滤器。两个组创建 $2 \times D_{out}/2 = D_{out}$ 通道。然后，将这些通道与 D_{out} 通道堆叠在输出层中

12.1 分组卷积与深度卷积

分组卷积是将过滤器按个数分组，深度卷积是把单个过滤器按通道分开，分组卷积输出层的通道数取决于过滤器数量，深度卷积的输出层通道数取决于 1×1 卷积。

分组卷积的优点：

第一，高效，允许以并行方式在多个GPU上进行模型训练，模型并行化被认为比数据并行化更好。数据并行化将数据集分成批次，然后对每个批次进行训练。然而，当批量大小变得太小时，本质上是在进行随机而不是批量梯度下降。这会导致收敛速度变慢，有时甚至较差。

分组卷积对于训练非常深的神经网络非常重要。

第二，模型更加高效，模型参数随着滤波器组数量的增加而减少。

第三个，分组卷积可以提供比标称二维卷积更好的模型。

13 混洗分组卷积

混洗分组卷积涉及分组卷积和通道混洗。分组卷积计算效率高。但问题是每个过滤器组只处理从前几层的固定部分传递下来的信息。每个过滤器组仅限于学习一些特定特征。此属性会阻止通道组之间的信息流并削弱训练期间的表示。为了解决这个问题，应用了通道洗牌。

通道混洗的想法是混合来自不同过滤器组的信息。在将此特征图输入到第二个分组卷积之前，先将每个组中的通道分为几个子组，将这些子组混合在一起。经过这样的改组后，像往常一样继续执行第二个分组卷积。但由于混洗层中的信息已经混合，本质上是每个组提供特征映射层（或输入层）中的不同子组。因此，组之间的信息流动加强。

14 逐点分组卷积

对于分组卷积，分组操作是在 3×3 空间卷积上执行的，而不是在 1×1 卷积上执行的。逐点分组卷积，顾名思义，对 1×1 卷积执行分组运算。该操作与分组卷积相同，只有一项修改，在 1×1 过滤器而不是 $N \times N$ 过滤器（ $N \geq 1$ ）上执行。不太具体，没懂。

15 心得

通过这篇博客我对卷积操作更加了解，除了之前了解的常规卷积，还了解了3D卷积/ 1×1 卷积/转置卷积等等。

首先是深度学习中的卷积和信号/图像中的卷积的区别，深度学习中，卷积中的过滤器不是颠倒的，对深度学习转置过滤器没什么意义，所以不转置。

然后是3D卷积，虽然之前就了解了多通道版本的卷积，但这只是输出张量为3D用于处理3D数据，过滤器仅在图像的高度和宽度3个方向上移动，所以还是2D卷积。3D卷积中，过滤器深度小于输入层深度，3D过滤器可以在三个方向（图像的高度、宽度、通道）上移动，一个3D过滤就可以得到3D输出。

1×1 卷积。对深度学习也十分重要，使用 $1 \times 1 \times D$ 的过滤器，卷积过后长宽不变，通道数变成1，多个通道的数据被嵌入一个通道，降低了维度深度。

转置卷积（反卷积）。之前提到卷积会压缩张量的大小，没考虑过需要扩大的情况，实际中会有需要将低维特征图映射到高维空间的情况（比如提取特征后对原始图像中每个像素进行分类）。运用的是 $C \times \text{Large} = \text{Small}$ 。CT \times Small = Large。但选择的过滤器大小不当会导致“不均匀重叠”，即棋盘伪影

扩张卷积。通过在内核元素之间插入空格来“膨胀”内核。用于在不增加内核大小的情况下廉价地增加输出单元的感受区域。

可分离卷积。分空间和深度分离卷积。例如，Sobel 内核是 3×3 内核，分为 3×1 和 1×3 内核。这样卷积执行的操作数量就更少，但并非所有内核都可以分为两个更小的内核，没有普适性。深度可分离卷积由两个步骤组成：深度卷积和 1×1 卷积。需要操作的次数更少，但减少了卷积中的参数数量。模型容量可能会显著降低，模型可能变得次优。

扁平化卷积。 $C \times Y \times X$ 的过滤器被分成 $C \times 1 \times 1$ ， $1 \times Y \times 1$ ， $1 \times 1 \times X$ 。但是，实际中的过滤器不好拆分。

分组卷积。用于多GPU并行化训练，分组卷积是将过滤器按个数分组，深

度卷积是把单个过滤器按通道分开，分组卷积输出层的通道数取决于过滤器数量，深度卷积的输出层通道数取决于 1×1 卷积。

混洗分组卷积。分组卷积每个过滤器组只处理从前几层的固定部分传递下来的信息。通道混洗的想法是混合来自不同过滤器组的信息。在将此特征图输入到第二个分组卷积之前，先将每个组中的通道分为几个子组，将这些子组混合在一起。经过这样的改组后，像往常一样继续执行第二个分组卷积，组之间的信息流动加强。