卷积算子优化-1 卷积计算的特点

什么是卷积?

卷积是一种数学运算，常用于信号处理和图像处理领域。在计算机视觉和深度学习中，卷积是一种重要的操作，用于提取图像或其他数据的特征。

在二维图像处理中，卷积操作可以理解为将一个滤波器（也称为卷积核）应用于输入图像的每个像素，通过对每个像素及其周围像素的加权求和来生成输出图像。卷积操作可以捕捉到图像中的局部特征，例如边缘、纹理等。

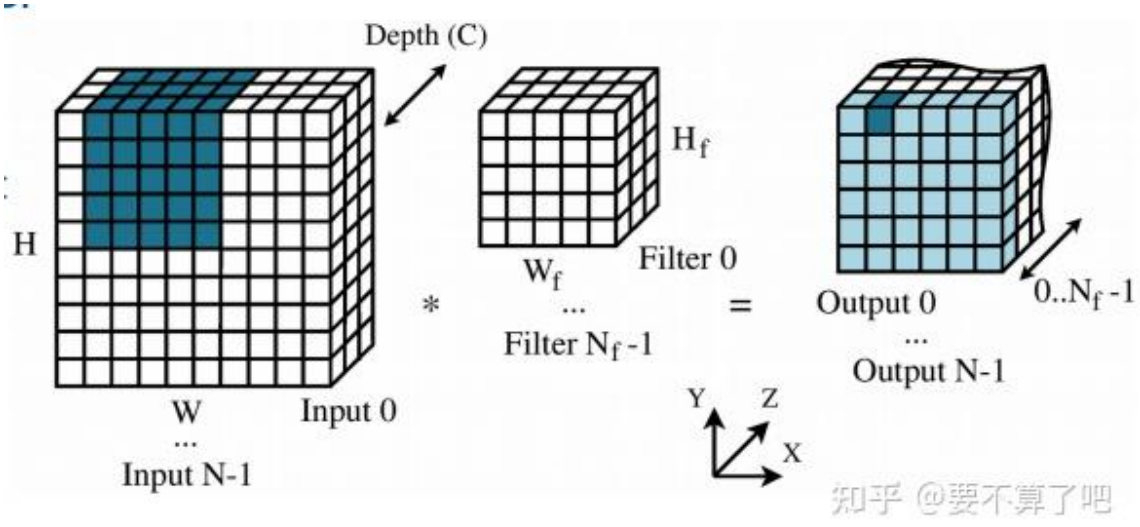
卷积操作

输出元素是一个滤波器和输入对应元素的点积，卷积操作有以下特点：

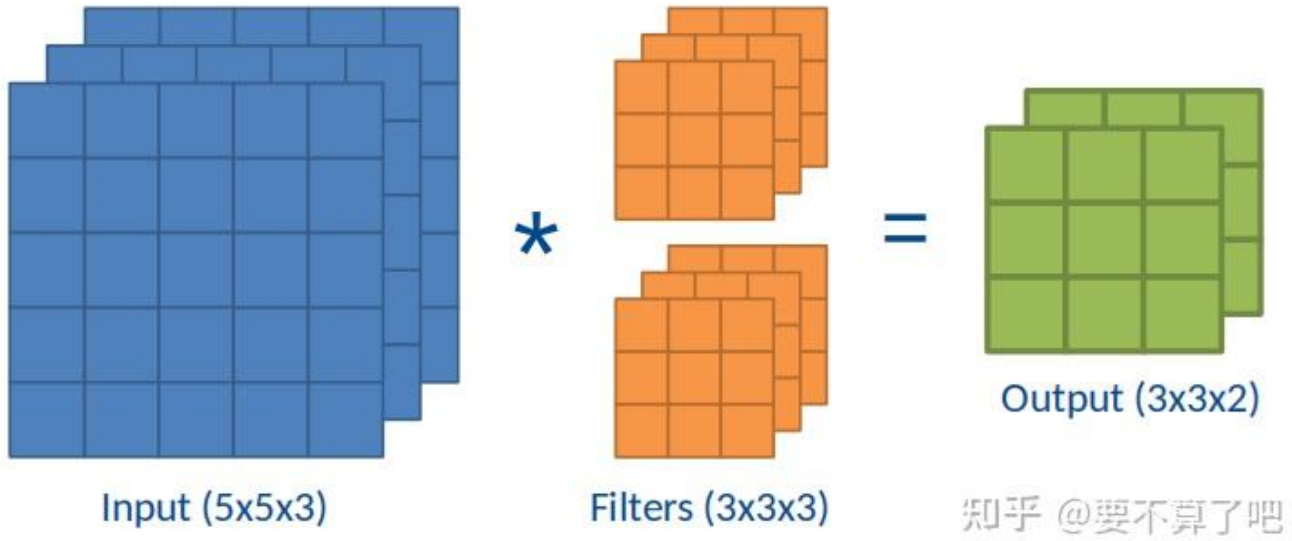
- 输入的深度和过滤器的深度（通道数）相等

- 每个输出元素对应不同输入元素（深蓝色高亮部分）

- 一个输出（共N*Nf个）的平面是其中一个输入(共N个）与其中一个滤波器（共Nf个）的卷积，所以输出深度=过滤器数



卷积运算



卷积参数

- 批次（Batch Size）：卷积操作时同时处理的样本数量，通常用N表示。

- 输入尺寸（Input Size）：输入数据的尺寸是指输入的多维数组的大小。对于图像，通常是指图像的高度、宽度和通道数。高度为H，宽度为W，通道数（深度）为C。

- 卷积核数量：在卷积操作中使用的卷积核的数量。每个卷积核都可以提取输入数据的不同特征。通常用K表示。

- 卷积核尺寸（Filter Size）：卷积核的大小。卷积核的尺寸决定了卷积操作的感受野大小和特征提取能力，高度为R，宽度为S。

- 步幅（Stride）：步幅定义了卷积核在输入数据上滑动的步长。较大的步幅可以减小输出的尺寸，而较小的步幅可以保持输出与输入的尺寸相同。

- 填充（Padding）：填充是在输入数据的边缘周围添加额外的像素或值。填充可以用于控制输出的尺寸，保持输入和输出的尺寸相同，或者在边缘处保留更多的信息。

- 输出尺寸（Output Size）：卷积操作的输出大小取决于以下几个因素输入尺寸、 卷积核尺寸、填充和步幅等。

根据这些因素，可以使用以下公式计算卷积操作的输出大小：

输出大小 = (输入尺寸 + 2 * 填充 - 卷积核尺寸) / 步幅] + 1

例如：当卷积的输入尺寸为NCHW，卷积核尺寸为KCRS时，Stride=2，Padding=1，输出大小为NKOhOw，其中

Oh= (H + 2 * Padding - R) / Stride] + 1

Ow= (W + 2 * Padding - S) / Stride] + 1

Note:如果输出大小不是整数，通常会向下取整。

卷积的实现

一个C++代码的例子，在给定输入大小（NCHW格式）、内核大小（KCRS格式）、Stride为1并且Padding为0的情况下执行卷积运算

```
int Oh = (H + 2 * Padding - R) / STRIDE + 1;
int Ow = (W + 2 * Padding - S) / STRIDE + 1;
for (int n = 0; n < N; n++) {
    for (int k = 0; k < K; k++) {
        for (int oh = 0; oh < Oh; oh++) {
            for (int ow = 0; ow < Ow; ow++) {
                for (int c = 0; c < C; c++) {
                    for (int r = 0; r < R; r++) {
                        for (int s = 0; s < S; s++) {
                            output[n][k][oh][ow] += input[n][c][oh + r][ow + s] * kernel[k][c][r][s];
                        }
                    }
                }
            }
        }
    }
}
```

优化思路

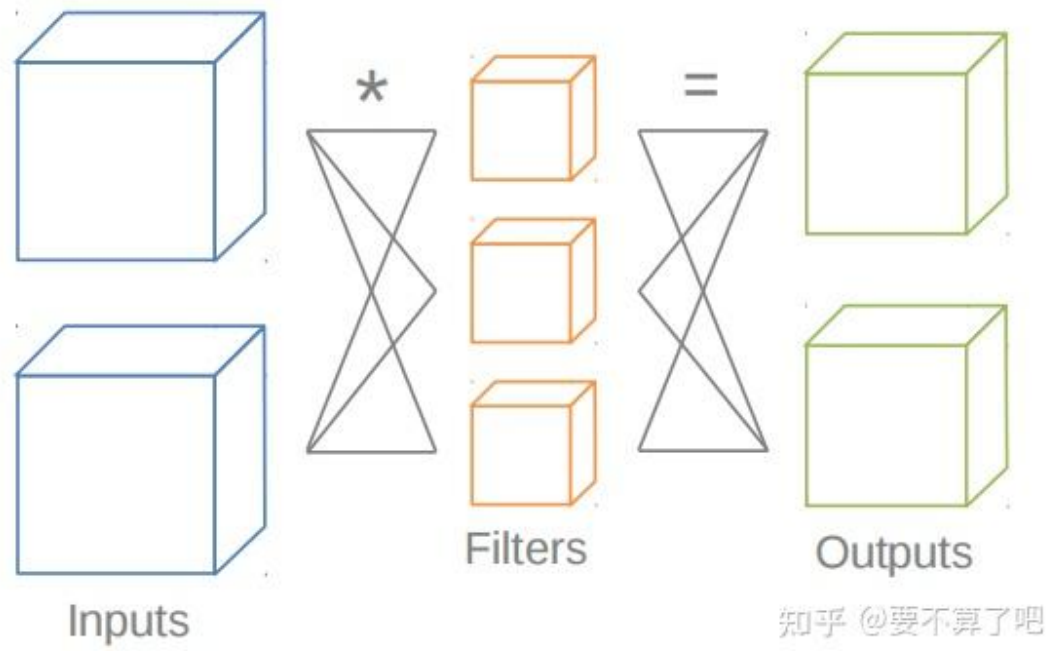
数据重用

在GPU上开发高效卷积操作的一个关键方面是最大化数据共享，这也是减少通信的一个关键因素。卷积操作存在着两级的数据重用，

在同一个卷积层，同一批次的输入会与所有数量的滤波器进行卷积，那么

- 每个滤波器会被所有输入使用

- 每个输入会被所有滤波器使用

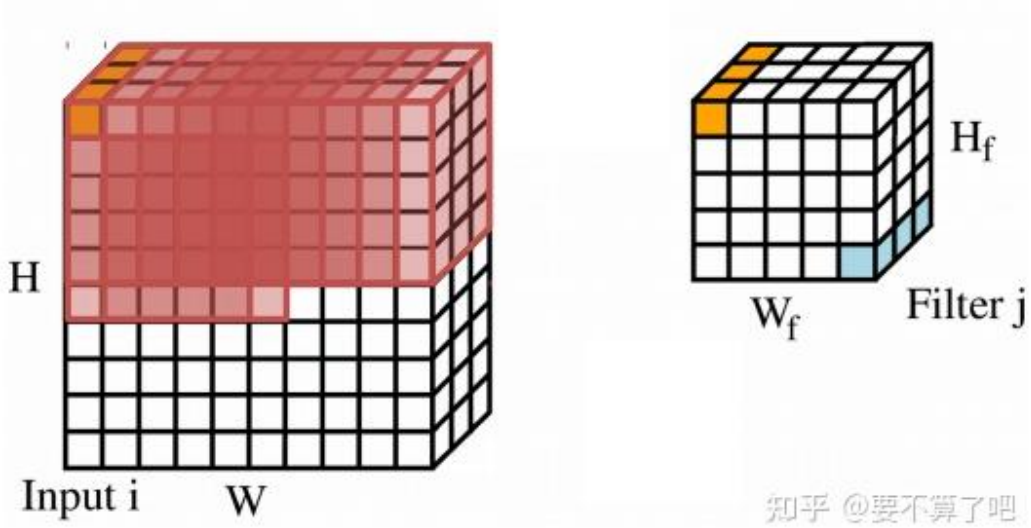


卷积层的数据重用

在同一次卷积中

- 输入元素重用，对于当前通道的同一滤波器，输入元素会重复读，越是处于输入中心的元素重复读的越多

- 滤波器元素重用，对于当前通道的同一输入，滤波器元素会重复读



卷积操作时的数据重用

重复计算

根据最小滤波器算法，进行输出尺寸为m，滤波器尺寸为r的一维卷积，用F（m，r）表示，进行的乘法次数为m+r-1。在二维卷积可以嵌套最小一维算法F（m，r）和F（n，s）来形成最小二维算法，以计算m×n个输出，用F（m×n，r×s）表示。需要的乘法次数为$(m + r - 1)(n + s - 1)$。在朴素实现中，需要的乘法为（m*n*r*s）次，在m=n=2，r=s=3时，朴素实现需要36次乘法，而最小滤波器算法给出最小乘法次数为16次，计算复杂度减少36/16=2.25倍。

下一篇将会介绍几种常用的卷积算法：

参考：

[Filling the Performance Gap in Convolution Implementations for NVIDIA GPUs]

[Fast Algorithms for Convolutional Neural Networks - arXiv.org]