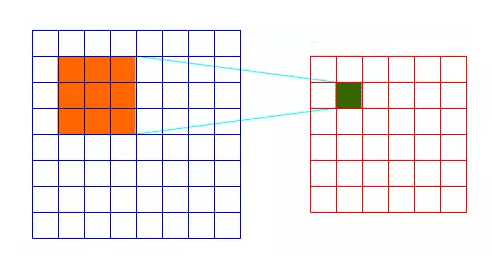
并行程序设计与算法实验

11-CUDA卷积

1. 滑窗法实现CUDA并行卷积

使用CUDA实现二维图像的直接卷积（滑窗法）。在信号处理、图像处理和其他工程/科学领域，卷积是一种使用广泛的技术。在深度学习领域，卷积神经网络(CNN)这种模型架构就得名于这种技术。在本实验中，我们将在GPU上实现卷积操作，注意这里的卷积是指神经网络中的卷积操作，与信号处理领域中的卷积操作不同，它不需要对Filter进行翻转，不考虑bias。

下图展示了滑窗法实现的CUDA卷积，其中蓝色网格表示输入图像，红色网格表示输出图像，橙色网格展示了一个­的卷积核，卷积核中每个元素为对应位置像素的权重，该卷积核的输出值为像素值的加权和，输出位置位于橙色网格中心，即红色网格中的绿色元素。滑窗法移动该卷积核的中心，从而产生红色网格中的所有元素。



**输入：**一张二维图像（）与一个卷积核（）。

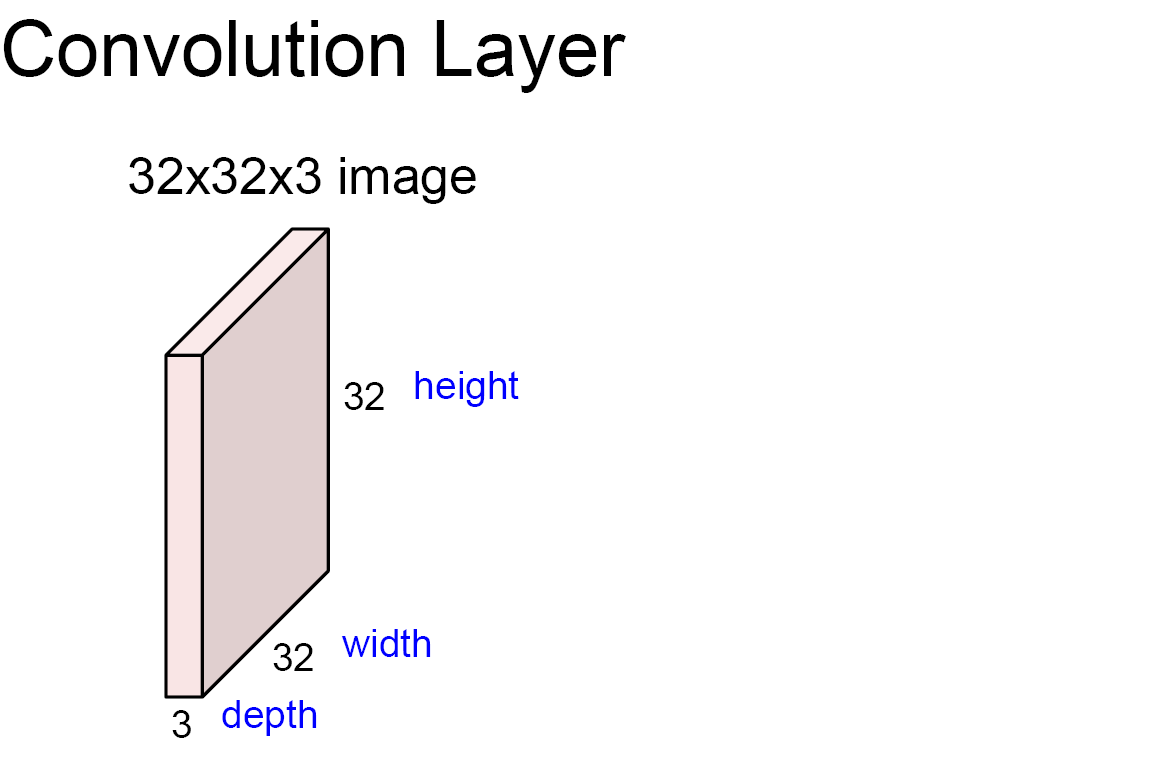
**问题描述：**用直接卷积的方式对输入二维图像进行卷积，通道数量（channel, depth）设置为3，卷积核个数为3，步幅（stride）分别设置为1/2/3，可能需要通过填充（padding）配合步幅（stride）完成卷积操作。注：实验的卷积操作不需要考虑bias (b)，bias设置为0。

**输出**：卷积结果图像（）及计算时间。

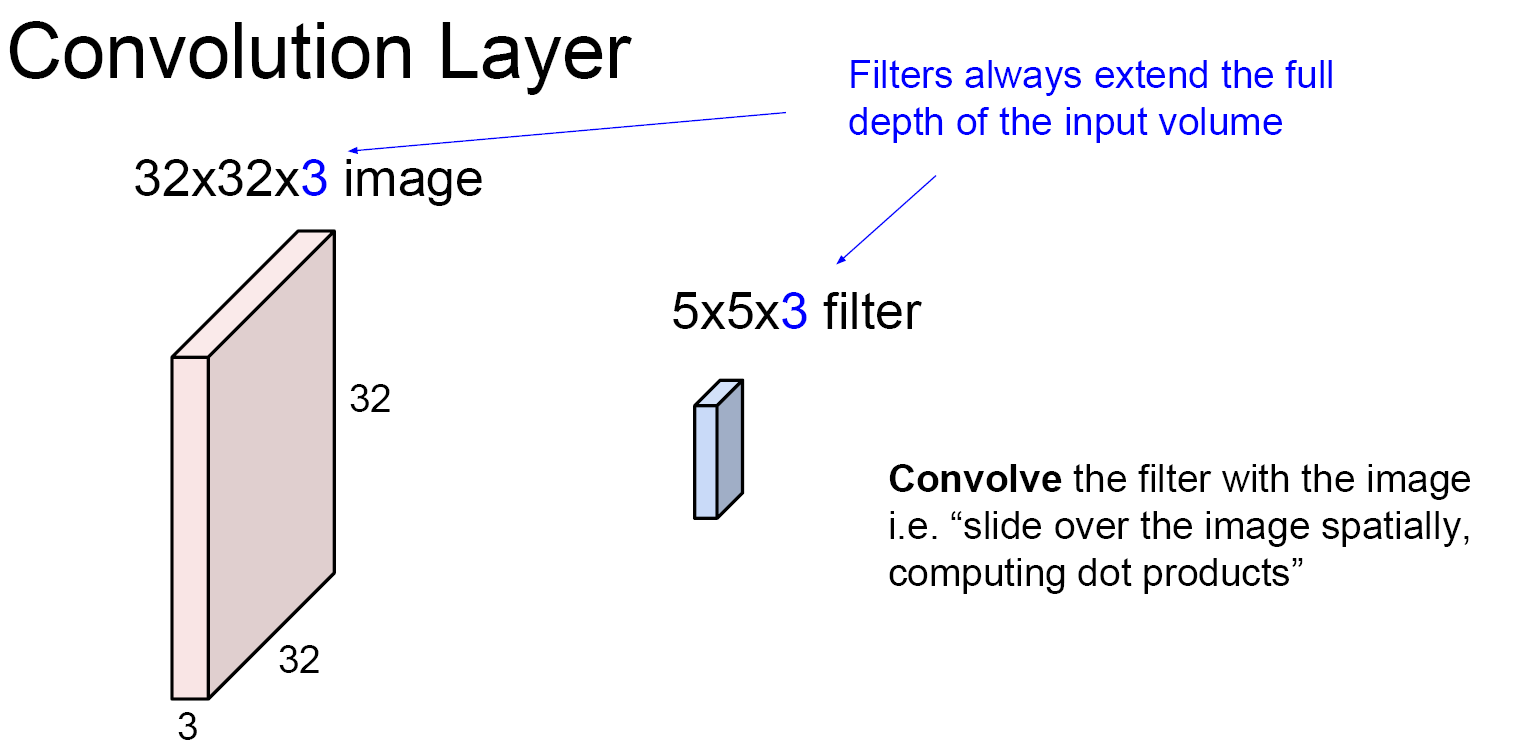
**要求：**使用CUDA实现并行图像卷积，分析不同图像大小、访存方式、任务/数据划分方式、线程块大小等因素对程序性能的影响。

**参考：**

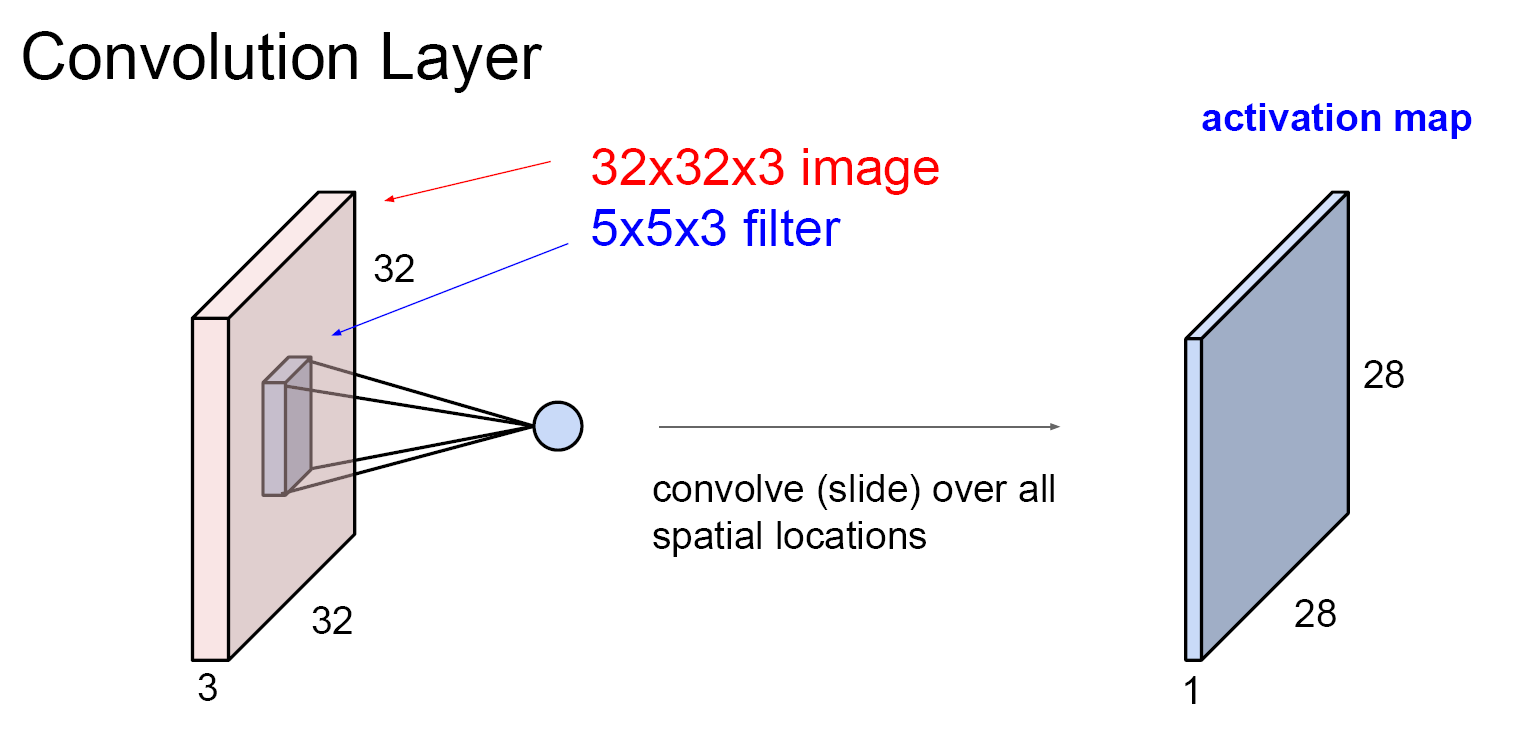
1）输入图像举例

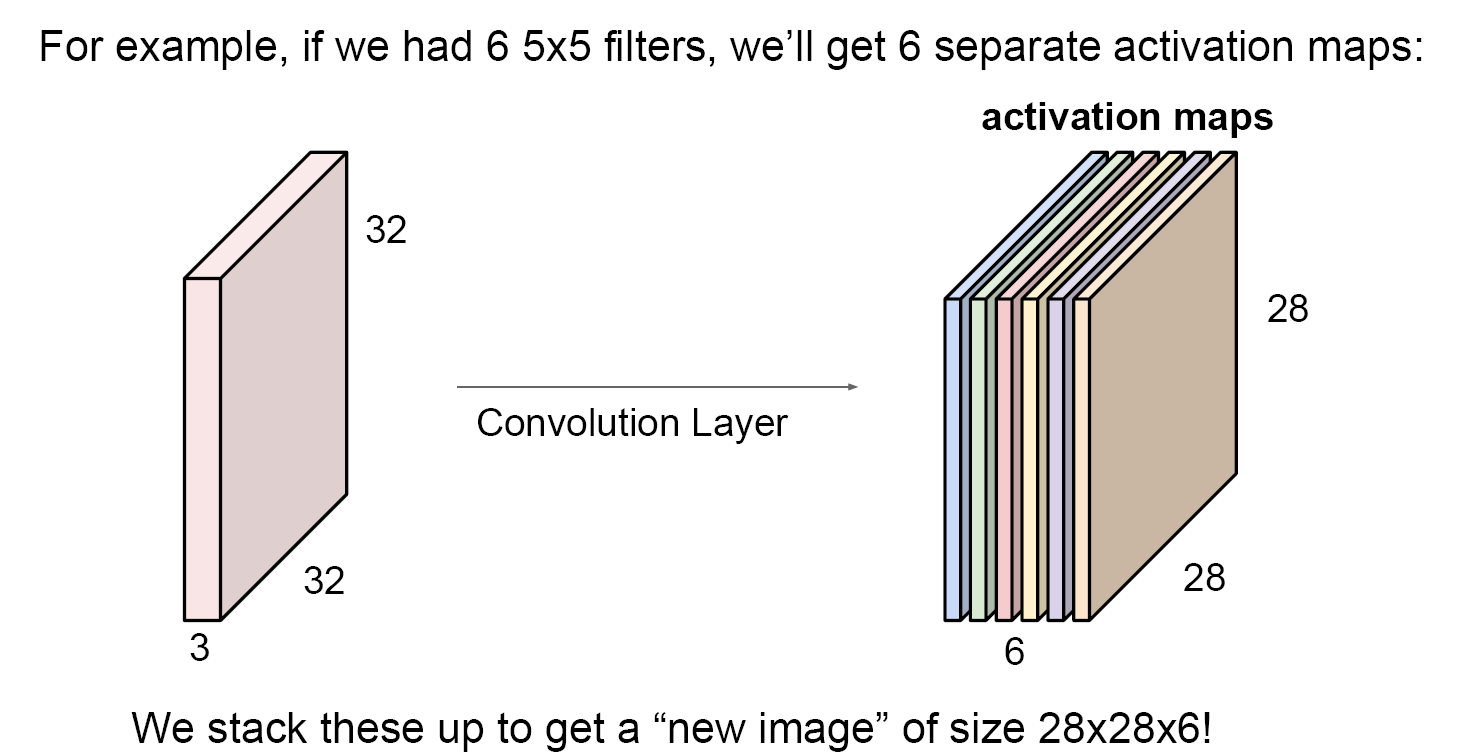


2）输入图像与卷积核（3通道）



3) 卷积计算过程





4) 卷积操作的步幅（stride）与填充（padding）

图示

描述已自动生成

图示

描述已自动生成

文本

描述已自动生成

2. 使用im2col方法实现CUDA并行卷积

滑窗法使用的卷积核对窗口内的图像像素求加权和，此过程可以写做矩阵乘法形式，其中为的权重矩阵，为的像素值矩阵。将图像中每个需要进行卷积的窗口平铺为的矩阵（列向量）并进行拼接，可将卷积计算变为矩阵乘法，从而利用此前实现的并行矩阵乘法模块实现并行卷积。具体拼接方式见下图：

图片包含 牌子, 窗户, 明亮, 钟表

描述已自动生成

**问题描述：**用im2col方法对输入二维图像进行卷积。其他设置与任务1（滑窗法并行卷积）相同。

3. 使用cuDNN方法实现CUDA并行卷积

NVIDIA cuDNN是用于深度神经网络的GPU加速库。它强调性能、易用性和低内存开销。

**要求：**使用cuDNN提供的卷积方法进行卷积操作，记录其相应Input的卷积时间，与自己实现的卷积操作进行比较。如果性能不如cuDNN，用文字描述可能的改进方法。

**CNN参考资料，见实验发布网站**

斯坦福人工智能课件Convolutional Neural Networks，by Fei-Fei Li & Andrej Karpathy & Justin Johnson

**其他参考资料 （搜索以下关键词）**

[1]如何理解卷积神经网络（CNN）中的卷积和池化

[2] Convolutional Neural Networks (CNNs / ConvNets) https://cs231n.github.io/convolutional-networks/

[3]im2col的原理和实现

[4]cuDNN安装教程

[5] convolutional-neural-networks

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks