高性能计算程序设计基础 秋季2021

**提交格式说明**

按照实验报告模板填写报告，需要提供源代码及代码描述至https://easyhpc.net/course/129。实验报告模板使用PDF格式，命名方式为高性能计算程序设计\_学号\_姓名。如果有问题，请发邮件至jiangjzh6@mail2.sysu.edu.cn，liuyh73@mail2.sysu.edu.cn 询问细节。

**任务1：**

通过CUDA实现通用矩阵乘法（Lab1）的并行版本，CUDA Thread Block size从32增加至512，矩阵规模从512增加至8192。

通用矩阵乘法（GEMM）通常定义为：

输入：M , N, K三个整数（512 ~8192）

问题描述：随机生成M\*N和N\*K的两个矩阵A,B,对这两个矩阵做乘法得到矩阵C。

输出：A,B,C三个矩阵以及矩阵计算的时间

**任务2：**

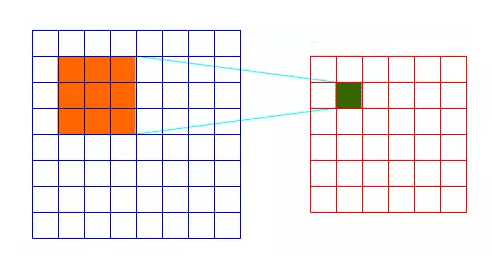
通过NVDIA的矩阵计算函数库CUBLAS计算矩阵相乘，矩阵规模从512增加至8192，并与任务1和任务2的矩阵乘法进行性能比较和分析，如果性能不如CUBLAS，思考并文字描述可能的改进方法（参考《计算机体系结构-量化研究方法》第四章）。

CUBLAS参考资料《CUBLAS\_Library.pdf》，CUBLAS矩阵乘法参考第70页内容。

CUBLAS矩阵乘法例子，参考附件《matrixMulCUBLAS》

**任务3：**

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。



任务一通过CUDA实现直接卷积（滑窗法），输入从256增加至4096或者输入从32增加至512.

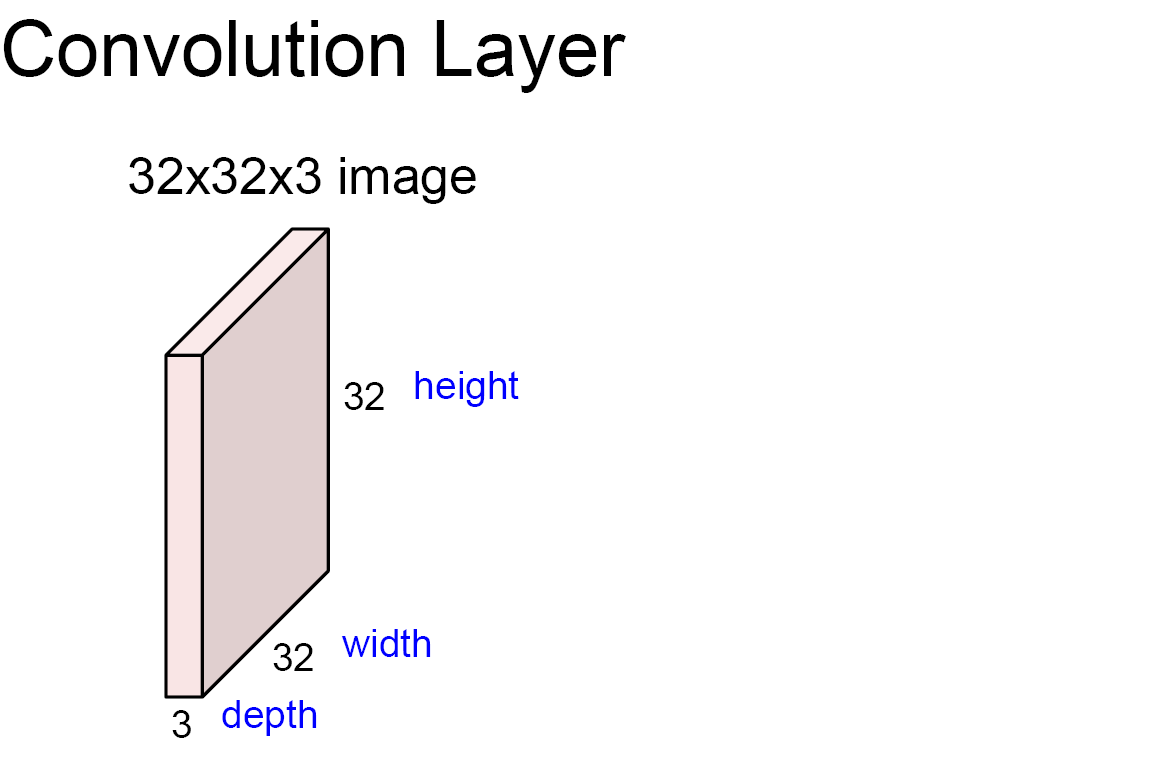
输入：Input和Kernel(3x3)

问题描述：用直接卷积的方式对Input进行卷积，这里只需要实现2D, height\*width，通道channel(depth)设置为3，Kernel (Filter)大小设置为3\*3\*3，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CNN操作。注：实验的卷积操作不需要考虑bias(b)，bias设置为0.

输出：输出卷积结果以及计算时间

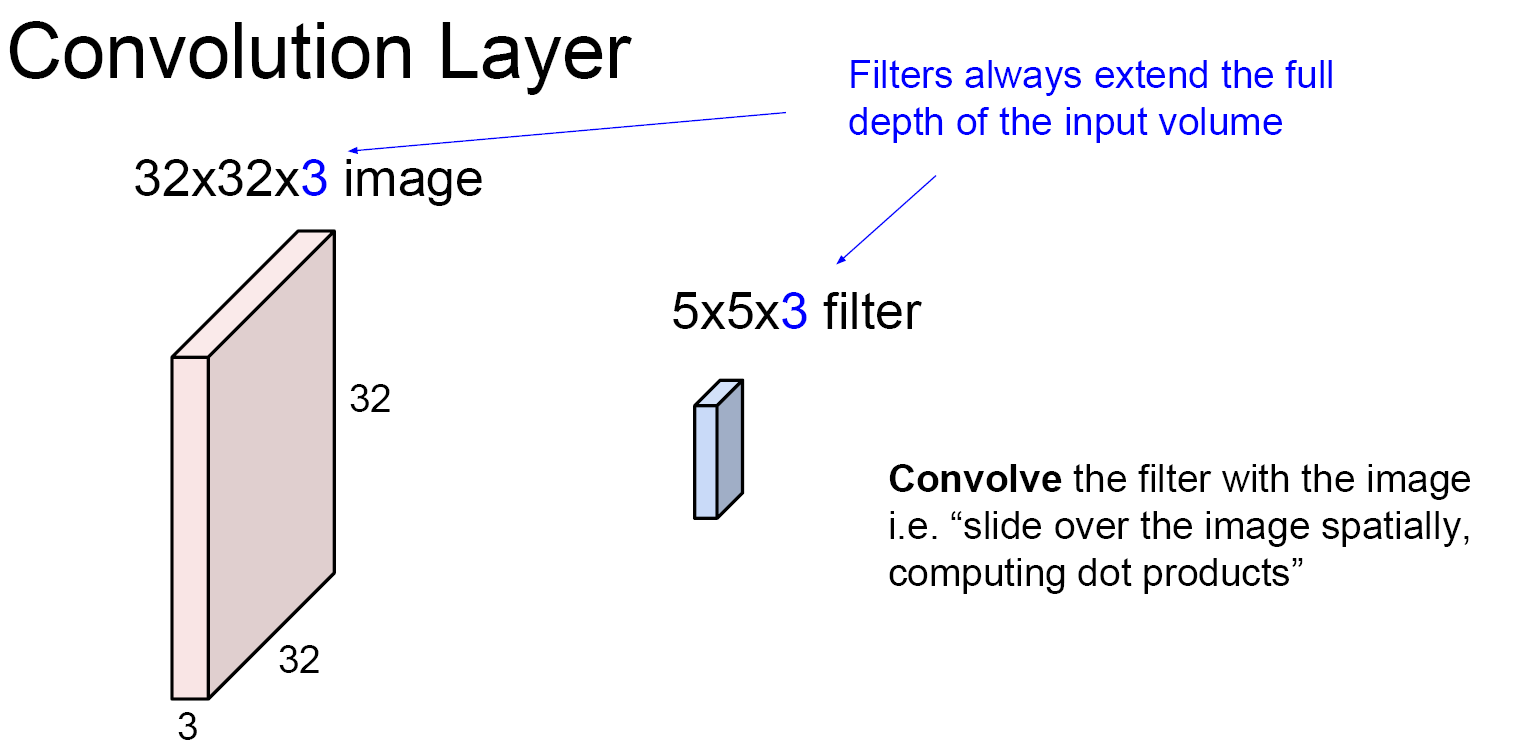
以下是部分CNN操作的解释ppt，具体参考附件中的人工智能课件。

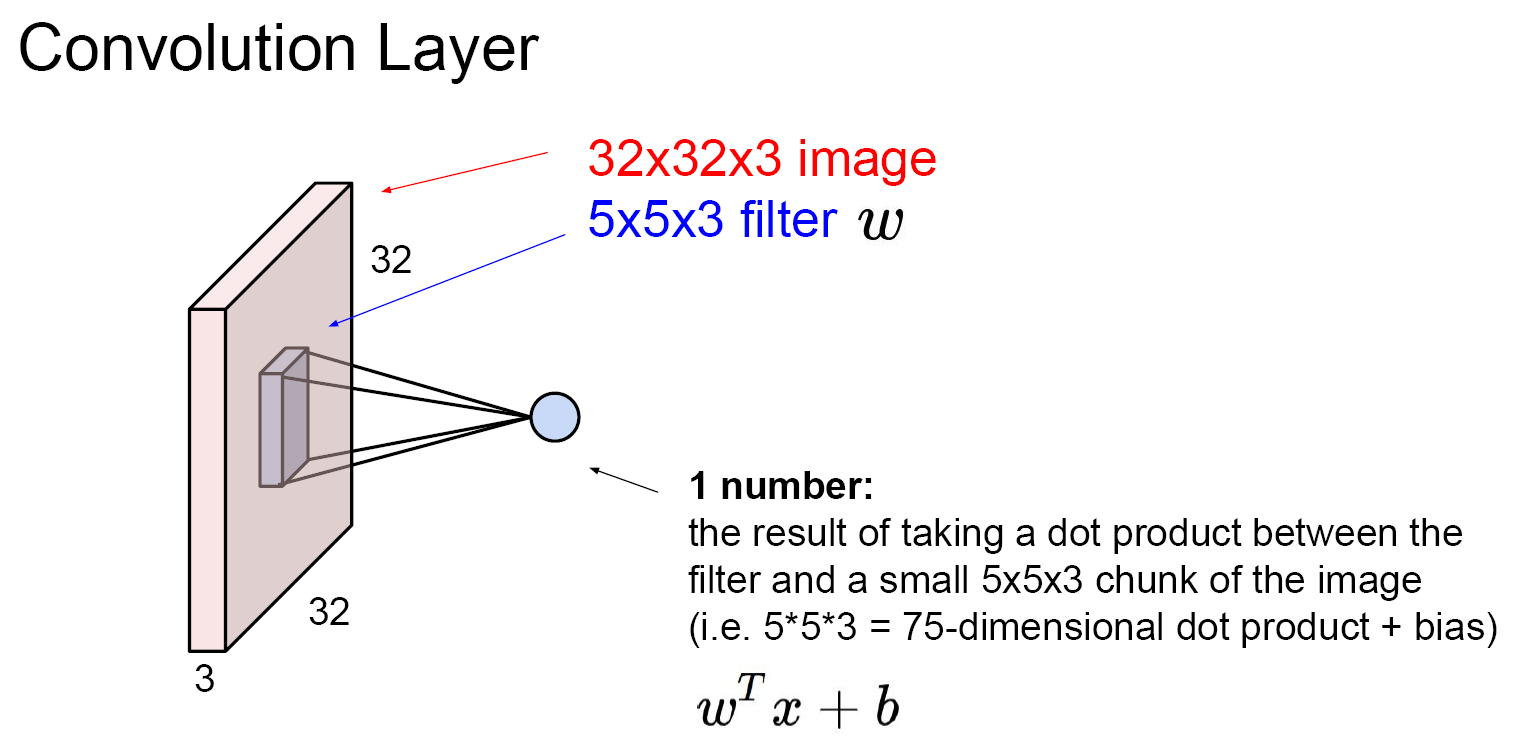
1）CNN Input Image举例



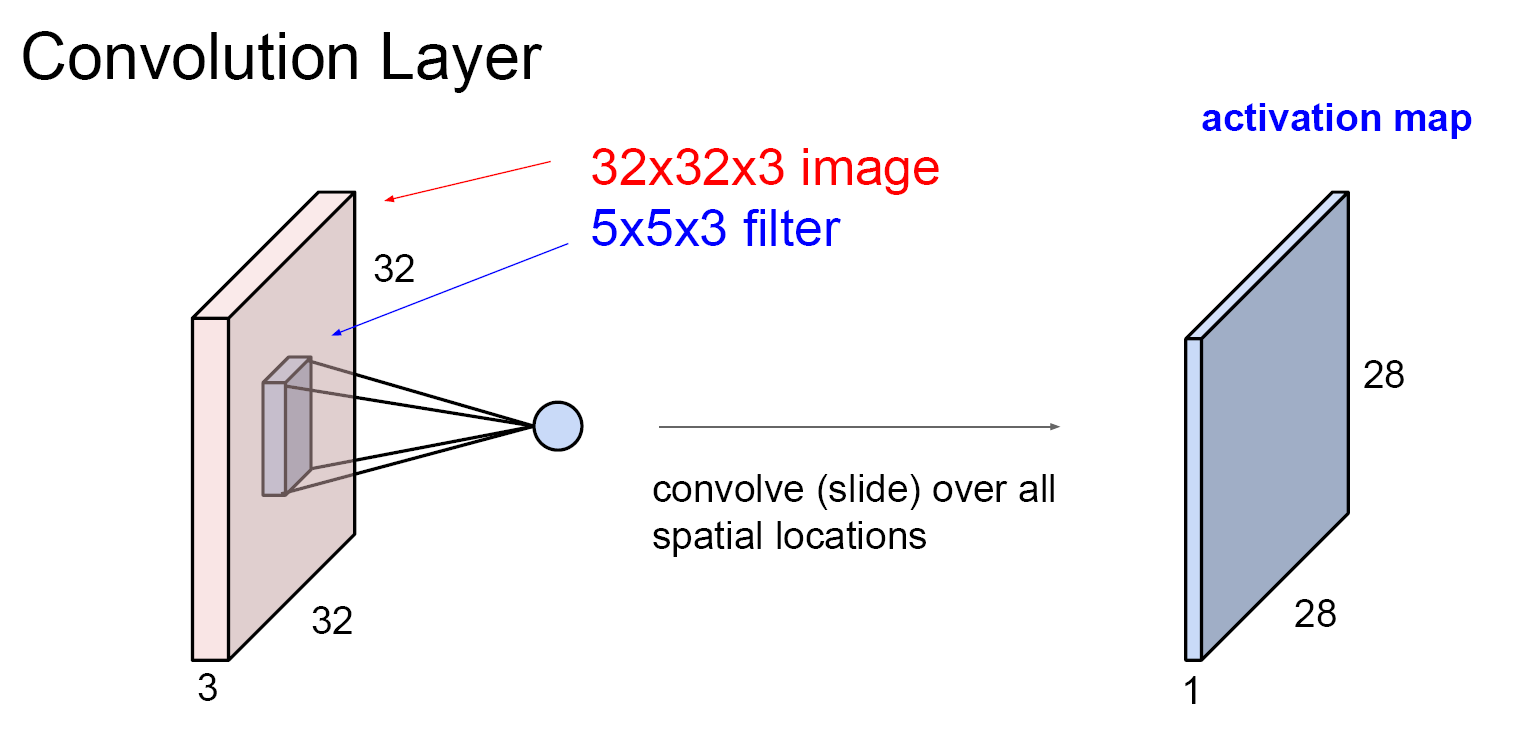
(Input)

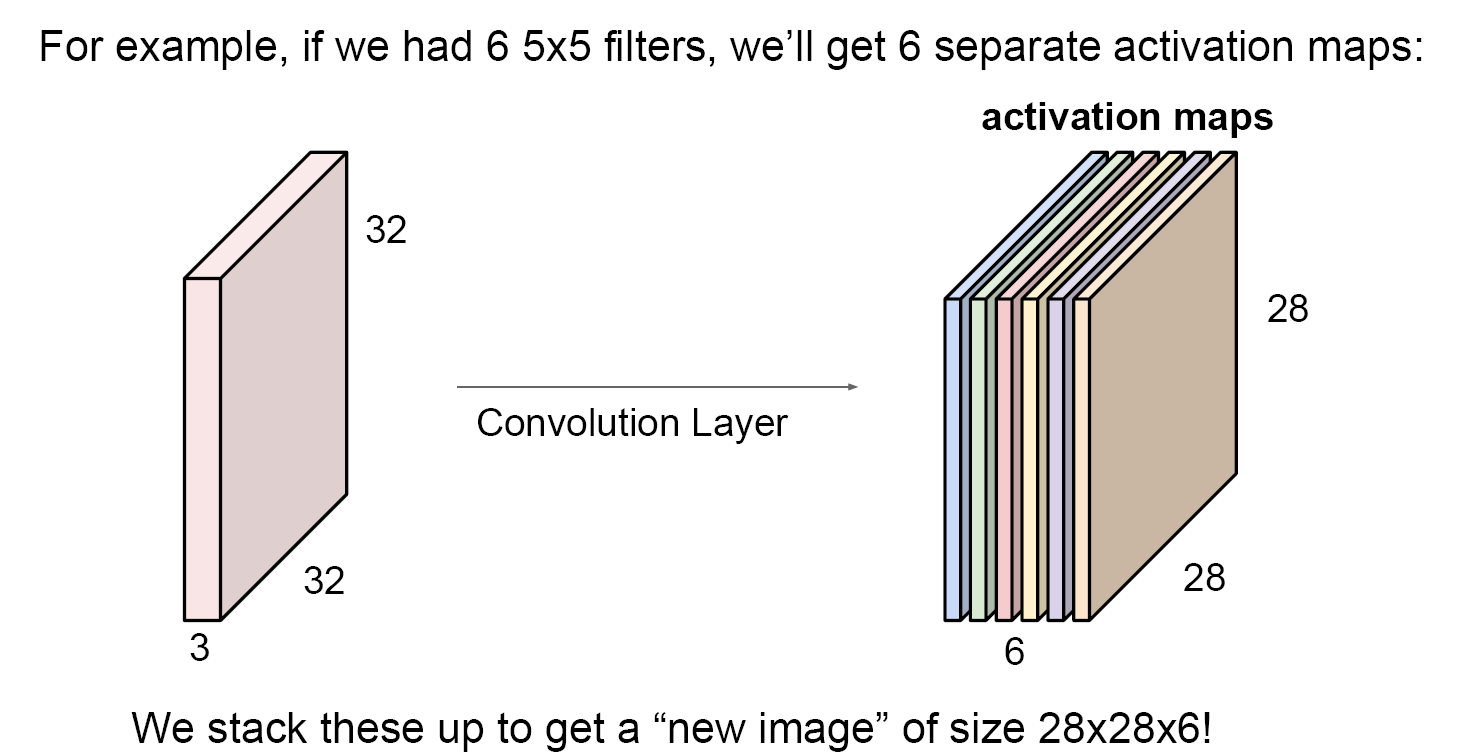
2）CNN Input Image 和 Kernel(Filter)





3)CNN 操作过程





4) CNN步幅stride和填充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

**任务4：**

任务二使用im2col方法结合上次实验实现的GEMM实现卷积操作。输入从256增加至4096或者输入从32增加至512，具体实现的过程可以参考下面的图片和参考资料。

输入：Input和Kernel (Filter)

问题描述：用im2col的方式对Input进行卷积，这里只需要实现2D, height\*width，通道channel(depth)设置为3，Kernel (Filter)大小设置为3\*3\*3，个数为3。 注：实验的卷积操作不需要考虑bias(b)，bias设置为0，步幅(stride)分别设置为1，2，3。

输出：卷积结果和时间。

图片包含 牌子, 窗户, 明亮, 钟表

描述已自动生成

**任务5：**

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。

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