

1 设计一个程序，实现矩阵乘法计算: $C = AB$

要求:

- (1) 程序接收矩阵大小(M, N, K)和三个矩阵在内存中的存放地址(A, B, C)作为输入，将矩阵 A, B 相乘，运算结果写入矩阵 C 。
- (2) 矩阵 A, B, C 的大小分别为 $M \times K, K \times N$ 和 $M \times N$ 。
- (3) 所有矩阵在内存中均按主行顺序排列，即对于矩阵:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

其在内存中的排列顺序为: $a_{11}, a_{12}, a_{13} \cdots a_{mn}$ 。

2 设计一个程序，实现简单的卷积计算: $O = I * W$

要求:

- (1) 程序接收与卷积运算相关的参数(卷积核大小 $kernel_size$, 输入图像边缘扩充大小 pad , 卷积计算步长 $stride$, 输入图像通道数 ic , 输入图像大小 ih 和 iw , 输出图像通道数 oc)和卷积核及输入、输出图像在内存中的地址($weight, input, output$)作为输入。计算卷积核 $weight$ 和卷积输入 $input$ 的卷积，将结果写入 $output$ 所在的内存地址中。

(2) 卷积核是一个四维矩阵，其维度为 $oc \times ic \times kernel_size \times kernel_size$ 。对于一个维度为 $n \times c \times h \times w$ 的四维矩阵 K ，其在内存中的排布顺序为 $k_{1111}, k_{1112}, k_{1113} \cdots k_{nchw}$ 。卷积输入和输出均为三维矩阵，输入图像的维度为 $ic \times ih \times iw$ ，输出图像的维度由卷积核和输入图像的大小决定。一个维度为 $c \times h \times w$ 的三维矩阵也可被认为是维度为 $1 \times c \times h \times w$ 的四维矩阵，内存排布方式同上。

- (3) 关于神经网络中卷积计算及其优化实现的相关参考资料:

- [An Introduction to Convolution Neural Networks](#)
- [Caffe中卷积是如何实现的](#)

3 完成方式

在[这里\(提取码:9ips\)](#)下载LightBLAS源码后，在 `src/lightblas/custom_mathfunctions.cpp` 下相应位置填充自己的实现代码，实现完成后再LightBLAS根目录下键入 `make` 指令编译源码，编译通过后，在LightBLAS根目录下执行 `build/tools/test_sgemm` (或 `build/tools/test_sconv`) 指令测试矩阵乘(卷积)算法实现的正确性和运行时间。

4 注意事项

- 必须在Linux系统下使用C++完成。
- 可以互相讨论，但代码严禁抄袭。
- 不允许调用任何外部矩阵计算加速库。