《Reformulating the direct convolution for high-performance deep learning inference on ARM processors》论文笔记

zxp

January 27, 2024

1 论文内容

《重新设计直接卷积以在ARM处理器上实现高性能深度学习推理》这篇论文是基于《zero-memory overhead direct convolution》那篇论文。这篇论文采用和《高性能零内存卷积》一样的优化，但是这篇论文注重数据布局保持常见的NHWC。《高性能零内存卷积》对数据布局进行了改变，提出了卷积友好的数据布局，将NHWC分成适合放入向量寄存器的$N \times H_o \times W_o \times C_{o,b}$块。虽然《高性能零内存卷积》指出上一层的输出是下一层的输入，只需要在第一层和最后一层付出数据布局变换的开销，但这篇论文是针对这个问题。这篇论文提出了两个算法，在只改变核的数据布局（论文只考虑了推理没考虑反向传播，核的布局和数据在卷积中不会改变）维持input张量的数据布局为常见的NWHC。

1.1 算法A

论文中利用了ARM NEON内在函数，用一些比较底层的实现了微内核（来仔细地将数据预取到处理器向量寄存器中，可能需要使用汇编指令），用于向向量寄存器搬运数据。在没有额外内存开销的情况下维持了数据布局，但这种方法限制很大，依赖特定硬件和特定的卷积运算的参数（核的大小，input张量的大小）。

1.2 算法B

用打包的方式，打包用于计算的$W_o \times C_{i,b}$大小的块进入缓存。这样比起算法A更不依赖硬件，数据复制的成本可能会通过内部循环中足够的计算得到很好的摊销，但数据复制需要成本，而且需要额外的内存开销。

1.3 性能评估

对比glops，使用的卷积层是GoogleLeNet和ResNet-50。没有使用层融合（论文提了一句用了会更快）。数据集是ImageNet。批量设置为1。对比了im2col+GEMM和《高性能零内存卷积》中的算法和这篇论文提出的算法A，B。

1.4 结论

算法A，B优于《高性能零内存卷积》中的算法。和《高性能零内存卷积》中的算法一样部分情况优于im2col+GEMM。算法B优于算法A。内存消耗远小于im2col+GEMM

2 心得

这篇论文是基于《高性能零内存卷积》。使用了和《高性能零内存卷积》一样的优化，但保留的传统直接卷积的NHWC数据布局。证明直接卷积算法可以看作围绕微内核的循环集合。提出两种算法，完全没内存开销但依赖特定硬件和特定的卷积运算的参数的算法A，和更加友好但需要非常小的内存开销用于打包的算法B。论文指出论文的工作还是在说明直接卷积和其他卷积算法一样是可以使用的。文中的ARM是使用的硬件为ARM v8架构的CPU，实现算法A用的也是ARM特有的函数。