《Optimizing Direct Convolutions on ARM Multi-Cores》论文笔记

zxp

January 20, 2024

1 论文内容

《在ARM 多核上优化直接卷积》这篇论文提出了nDirect，一种优化直接卷积的方式。针对移动设备和高性能计算中常见的基于arm的多核CPU。这篇论文选择直接卷积也是因为常见的im2col变换加CEMM实现卷积的方式的内存开销过大和直接卷积方法可以优于im2col。nDirect着重于在不改变数据布局的情况下优化直接卷积，主要是优化打包方式，和针对ARM平台。

1.1 介绍

除了常见im2col，论文提到了LIBXSMM库，也是在cpu实现直接卷积，并且十分先进（这篇论文是2023年11月的）。LIBXSMM支持x86和ARM平台，并且优于im2col+GEMM，不过这篇论文认为LIBXSMM的缺点是数据布局过于特殊，不兼容主流，并且基于GEMM微内核并未注意数据重用。论文提出LIBXSMM来说明这篇论文提出的nDirect的优势，数据布局符合主流。
还有调用优化的策略，im2col和XNNPACK和论文提出的nDiect是Library，LIBXSMM是JIT，Ansor是Search。

1.2 ARM平台特殊的地方

论文认为，支持ARM平台的DL框架特别少，所以数据布局要符合主流，这样才有兼容性，不用大规模重构代码或者额外内存开销。
论文认为，数据结构转换和顺序数据打包会产生大量内存负载和存储操作，多线程竞争内存带宽的时候会互相影响导致速度变慢。理想的卷积微内核应该具有高性能并且没额外的内存开销。而且论文认为现在的并行化策略太粗糙了，没考虑工作的时候的负载导致ARM多核上的卷积性能较差。
ARM cpu经常在二级缓存上同时保存数据和指令，缓存需要留出一部分给指令用。SIMD，使用的128位寄存器（放4个FP32数据）。专门定制了两个微内核，一个用来加速卷积，另一个用来填充输入张量。

1.3 nDiect实现的优化

优化还是常见的那几个。Loop Ordering方便打包。Loop tilng提高缓存局部性的关键，分块大小按缓存来分，并且因为ARM cpu会在缓存放指令所以留一部分。
nDiect特色在input张量的打包方式。打包的块来自每个通道，hight为3（这里是3应该是论文只对3x3的内核做了优化），width为通过论文中公式算出的合适放入缓存的大小。会有个内核将这些块打包到线性的缓冲区。使打包的数据在内存中连续。论文打包的这些块方便使用SIMD和FMA。
还特地关注了并行化，给模型做了线性映射。没有并行化input的channel和核的hight和width，需要写入output的同一个元素，会抢占资源，需要加锁会导致性能大幅下降。

1.4 性能评估

首先评估的是多核性能。与最佳的基准在Phytium2000+和KP920和ThunderX2（三块ARM硬件）上比吞吐量的平均值。和比能达到的CPU的峰值。nDirect能达到70-80%。使用来自VggNet的五个卷积层量化。nDirect所有单独的层优于Ansor（充分自动调优的单个卷积）的直接卷积。在三个硬件分别提高1.9倍1.82倍1.51倍。论文认为这是更好的打包和并行策略带来的。
然后评估了推理性能，和Ansor性能相当，但作为基于库的方法没Ansor昂贵的搜索开销。嵌入式平台上都优于其他替代方案，比最好的基准（单核的XNNPACK和LIBXSMM）快1.15倍和1.19倍。
之前评估关闭了硬件的超线程。测试超线程的时候每个核心运行四个线程。批处理大小匹配逻辑核的大小。1.28倍优于XNNPACK。

1.5 结论

nDirect是一种新的高效的直接卷积方案。在ARM多核cpu提供高性能高数据可重用性和深度学习框架兼容性。

2 心得

这篇论文提出的nDirect特色是数据打包方式和仔细考虑了并行化策略。论文强调数据格式的重要性，因为这涉及到兼容性问题，所以nDirect使用的是主流的数据布局，在这基础上提出新的打包方式。关于ARM cpu特殊的地方好像不是特别明显，没cpu和gpu差距大，用的优化还是那几个，论文提出的特殊地方也只有支持的框架少要多考虑兼容性，缓存中可能会放指令要留点缓存空间（前几周看的一篇x86 cpu平台上也提到要留点缓存空间）