Advancing Direct Convolution using Convolution Slicing Optimization and ISA Extensions

JX-Ma

2024/1/10

1 笔记

这篇文章主要提出了 sCONV, 一种基于 MLIR/LLVM 代码生成工具链的直接卷积算法. 该算法提出卷积切片分析 (CSA), 一种特定于卷积的 3D 缓存阻塞分析方法，侧重于缓存层次结构上的块重用。还有一种卷积切片优化（CSO) 这是一个代码生成通道，使用 CSA 生成平铺直接卷积宏核，基于向量的打包（VBP）这是一种针对特定架构的优化输入张量打包的解决方案。

MLIR 代表多级中间表示语言（Multi-Level Intermediate Representation），是一种用于优化编译器的开源项目。MLIR 旨在提供一种灵活的、可扩展的中间表示形式，以支持各种不同的编程模型和领域特定语言。MLIR 是 LLVM 项目的一部分，LLVM 则是一种开源编译器基础设施，提供了一套用于构建编译器的工具和库。MLIR 的目标是通过提供统一的中间表示形式，帮助编译器开发者更轻松地构建高效的编译器和优化器。

在卷积的基础知识中提到了 POWER10 与 MMA 引擎，POWER10 是 IBM 推出的一款处理器芯片，它是 IBM Power 体系结构的一部分。POWER10 处理器具有高性能、高可扩展性和高效能的特点，适用于数据中心和企业级应用。

MMA（Matrix Multiply and Accumulate）引擎是 POWER10 处理器中的一个重要特性。MMA 引擎专门用于执行矩阵乘法和累加运算，这对于深度学习和人工智能等工作负载来说非常重要，因为这些工作负载通常涉及大量的矩阵运算。MMA 引擎的加入可以提高处理器在这些领域的性能表现。

卷积切片：使用最小化重用距离的高效缓存平铺解决方案。

卷积切片分析：CSA 是一种基于缓存块的算法，它通过寻找合适的数据平铺和调度组合来减少卷积执行的时间，目标是最大化输入张量和卷积核的数据重用，一个输出张量的元素由输入张量的一个窗口和卷积核得到，不同输入张量窗口间存在着数据重叠，对输入张量进行合理的切片可以在很大程度上利用起这些重复的数据从而加快卷积的速度。

卷积切片优化：基于 CSA 计算的平铺大小和调度，对输入、过滤器、和输出进行切片，生成宏内核来计算平铺面积，它的主要思想是最大化平铺重用以利用缓存层次。

2 总结

这篇文章讲的是直接卷积算法上的优化，它通过卷积切片，选用合理的切片来让数据重用率达到最高，切片的部分实现使用了一种代码生成链的工具，将卷积的张量通过工具变换为数据重用率较高的表现

形式。想起之前我实现的卷积优化中，因为步长的存在，我使用了向量寄存器去存储不连续的元素，这样实现的数据重用率非常低，因为输入张量的窗口中存在着很多数据重叠，如果可以很好的利用这些数据重叠的部分可以更近一步的提高卷积的效率，例如输入张量的窗口大小为 11x11 步长为 4，这个时候每两个相邻的窗口直接有 7x7 的数据是重叠的，连续窗口的数量越多，重叠的数据也就更多。在以后实现卷积优化的时候，对于这些重叠数据的处理也是影响卷积性能最主要的原因之一。