Title: OVERVIEW OF TENSOR LAYOUT IN MODERN NEURAL NETWORK ACCELERATOR

JX-Ma

2024/11/30

ABSTRACT

This paper mainly analyzes two data layouts, one is NHWC and the other is N (C/x) HWx. Analyzed the impact of different L1/L2 cache configurations on cache hit/miss count, load/store count, and eviction count, in order to provide guidance for the basic hardware design of L1/L2 cache in modern neural network accelerators.

INTRODUCE

This section mainly introduces Im2col convolution. The popular convolution methods now are im2col+GEMM. This article introduces the transformation of data layout, including NHWC and N (C/x) HWx. Here, x is set to 32

Implementation of Profiling Framework

This article proposes a Block Splitting Engine engine that can analyze the most suitable partitioning scheme based on the size of the input tensor, convolution kernel, and output tensor. Subsequently, it was mentioned that the use of caching mainly focuses on the tensor layout of feature maps. So a Tensor Data Loading Engine was added later, and a hierarchical storage structure was designed by combining BSE and TDLE, including two levels of cache and one main memory: L1 cache loads data from L2 cache and stores it in L2 cache, and L2 cache loads data from main memory and stores it. TDLE first sends a tensor data request to L1. If L1 hits, L1 will directly return the tensor data to TDLE. Otherwise, L1 will go to L2 to load tensor data, and L2 will perform hit/miss tests. If it hits, L2 will return the data to L1. Otherwise, L2 will load tensor data from main memory.

SUMMARY

A performance analysis of NHWC framework BTH and N (C/x) HWx was proposed. BTH includes analysis of two common tensor layouts in modern neural networks, BSE, TDLE, and HMS, each of which can be flexibly configured. We can refer to the Block Splitting Engine engine proposed in this paper and design our own block selector, because the convolution we optimized has different block sizes in different layers, which requires different algorithms to implement different convolutional layers, and the only difference between these algorithms is the block size.