Advancing Direct Convolution Using Convolution Slicing Optimization and ISA Extensions

Hao Deng

January 2024

1 Reading Reflection

Traditional convolution consists of two parts–Im2col and GEMM. One of the drawbacks is that performing Im2col on input and filter tensors generates huge matrices, which may lead to poor memory hierarchy.

This paper proposed the SConv algorithm using Convolution Slicing Optimization(CSO) and Instruction-Set Architecture (ISA) extensions. SConv mainly consists of 4 parts.

1. Convolution Slicing Analysis (CSA): CSA computes the sizes, distributions, and scheduling of tiles (tiles are tensors divided into small parts). The most important rules it follows is that input tile, filter tile, and output tile must all fit in L1 cache.
2. Convolution Slicing Optimization (CSO): CSO is responsible for packing tensors and performing convolution on tiles based on the order generated by CSA.
3. Vector-Based Packing (VBP): The fact that the first element in the next window is the second element in the current window enables the use of VBP. Using vector shift operation (shift the vector to the left) can reduce data loads.
4. Micro-Kernel: Calculate the result.

The results of SConv show noticeable improvement in direct convolution. SConv replaced im2col and the packing routine in im2col+GEMM with a single packing step. Compared with im2col+GEMM, SConv reduces data-manipulation overhead and cache misses. There are two noticeable points mentioned in the paper. The core of improving direct convolution is to reduce reuse distance.

When designing algorithms, we need to consider the trade-off between the compilation time and model development cycle time. We don't want the compilation to cost too much time, because it would not be conducive to testing/debugging models.