Optimizing Binary Convolution for Compute-In-SRAM Accelerator

Author: Samantha Cobado    Advisor: Prof. Zhiru Zhang

Background

Abstract
Computationally intensive algorithms on massive amounts of data are slow to run on generic CPUs. One direction of speeding up these computations is using hardware accelerators such as the APU. Optimizing binary convolution (a complex layer in a neural network) through bit packing and minimizing tiles can help make running on the APU much faster than on a CPU.

Model Implementation

Binary Convolution

Binary convolution loops through individual bits in the inputs and weights, XNOR’ing a binary encoded bit from the input and weight, and then accumulating the results in the applicable window corresponding to a single output entry.

Because the number of entries on the APU is limited by the vector registers, tiling is necessary to break it into smaller parts. However, this requires more calls to run the APU and, thus, more data transfers.

Optimizations to Binary Convolution

Individual bits can be packed into windows. This way instead of looping over individual bits, the loops can access an entire window at once, reducing the number of iterations in a loop and memory accesses.

By pre-processing the data on the APU instead of the host, the number of tiles, number of calls to run the APU, and the number of data transfers can be significantly reduced.

Acknowledgements

Thanks to Prof. Zhang, Niansong Zhang, and Prof. Zhang’s APU team for their work in the implementation of the binary neural network and providing advice, help, and feedback throughout the project. This could not have been done without them.