Optimizing Binary Convolution for Compute-In-SRAM Accelerator

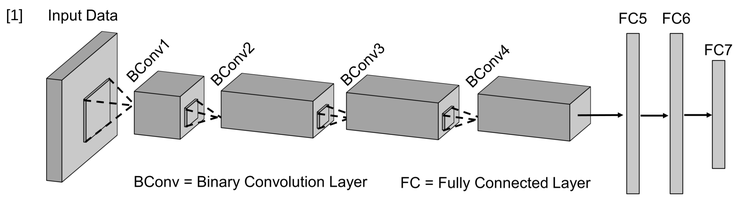
Author: Samantha Cobado Advisor: Prof. Zhiru Zhang

Background

Abstract

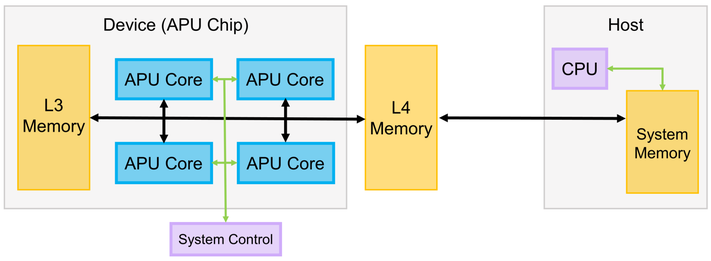
Computationally intensive algorithms on massive amounts of data are slow to run on generic CPUs. One direction of speeding up these computations is using hardware accelerators such as the APU. Optimizing binary convolution (a complex layer in a neural network) through bit packing and minimizing tiles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

Binary Neural Network

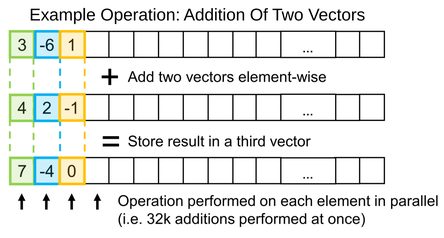


Binary neural networks were chosen for optimizing on the APU since it has potential for large amounts of speedup on in-memory computing devices due to being able to binary encode and pack data.

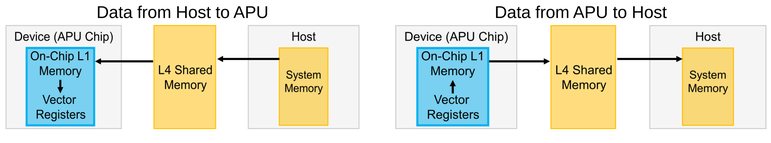
APU: Compute-In-SRAM Accelerator



Computations on the APU are extremely fast because they are performed on entries of the vector registers in parallel. The APU has a total of 16 vector registers each containing 32,000 entries.

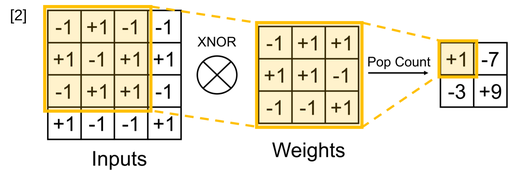


Memory transfers to and from the APU results in a significant overhead as the data needs to pass between different memory blocks.

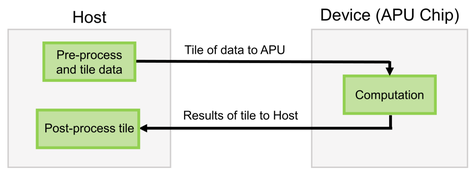


Model Implementation

Binary Convolution

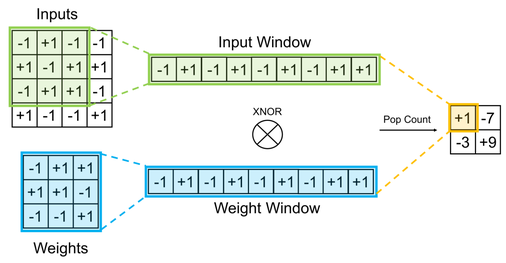


Binary convolution loops through individual bits in the inputs and weights, XNOR'ing a binary encoded bit from the input and weight, and then accumulating the results in the applicable window corresponding to a single output entry.

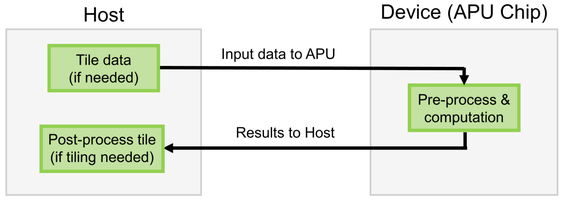


Because the number of entries on the APU is limited by the vector registers, tiling is necessary to break it into smaller parts. However, this requires more calls to run the APU and thus, more data transfers.

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 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.

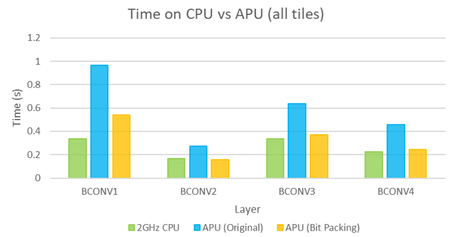
References

[1] Han, Xiaobing, Yanfei Zhong, Liqin Cao, and Liangpei Zhang. 2017. "Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification"
[2] Zhang, Yichi, Junhao Pan, Xinheng Liu, Hongzheng Chen, Deming Chen, and Zhiru Zhang. "FracBNN: Accurate and FPGA-efficient binary neural networks with fractional activations."

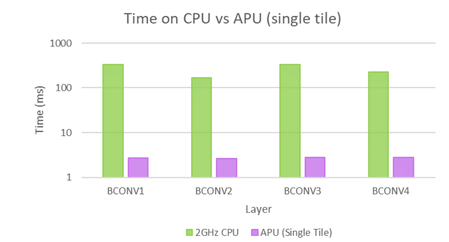
Preliminary Results

Speedup Over CPU for Binary Convolution

Time on APU includes calculations and data transfers. A binarized version of AlexNet neural network architecture was used for testing.

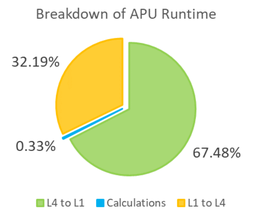


Bit packing reduced APU time to be about equal to the CPU.



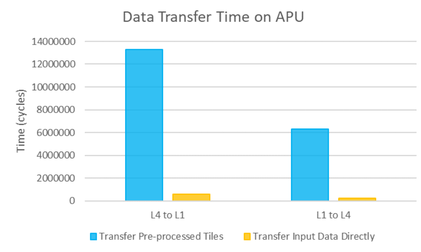
The time on the APU for a single tile is significantly less than computation on CPU, showing how important it is to minimize tiling

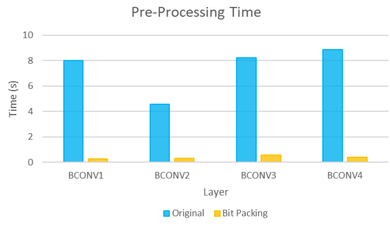
Reducing Communication Overhead



Data transfer in the original and bit packed implementations make up much of the time on APU.

Minimizing tiling by doing pre-processing on the APU can reduce data transfer time.





Bit packing can help reduce pre-processing time, minimizing additional time added to pre-process on the APU.

Combining bit packing and minimization of tiling has the potential to significantly speedup binary convolution on the APU over that on a CPU.

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.

CornellEngineering
Electrical and Computer Engineering