TETRIS: Scalable and Efficient Neural Network Acceleration with 3D Memory

Paper Introduction.

The paper describes TETRIS - a neural network accelerator that uses 3D memory. Current HW neural network (NN) accelerators use on-chip SRAM buffers and off-chip DRAM channels. However, scaling these memory systems is challenging. The paper proposes the use of a 3D memory architecture using Micron's Hybrid Memory Cube (HMC) instead of 2D SRAM buffer + DRAM channels for NN acceleration.

The paper proposes and describes the hardware architecture as well as the scheduling and partitioning techniques for TETRIS. Performance benchmarks show 4.8x better performance and 1.48x less energy over 4 LPDDR3 channels (a version of eyeriss). These improvements are attributed to the usage of 3D memory which provides high throughput with less energy usage.

Paper Outline

What are deep neural networks?

A sequence of layers -- used for image processing, other tasks, etc.
Convolutional layer - filters applied to feature maps to generate feature maps (actually a dot product)
Max pooling layer selects a maximum value from given window.
Fully connected layer - flattens data from previous layer and applies transformation.

NN acceleration

Large number of weights and feature maps => use HW acceleration.
Accelerators consist of large numbers of PEs to perform multiply-accumulate operations.

Memory challenges

Can use large on-chip SRAM, but very expensive.

Can't use large buffers.
Power consumption increases dramatically for Eyeris.

3D Memory

Vertically integrated DRAM dies on top of logic layer
TSVs (vias) used to connect vertically
Parallel performance higher
Uses HMC (Micron) with architecture as described in paper.

Partition and Scheduling

Two problems to solve - mapping, ordering
Mapping - Eyeriss solution
Ordering - Bypass solution

Evaluation

Using ZSim (a tool that lets simulate any architecture)
4.1x better performance
1.48x lesser energy usage
Td analysis

Comments

- Custom ordering via ofmap, ifmap, filter done to minimze buffer reuse or for some other reason? Why only ifmap and filter? (Prof. Bahar)
    - Analytical solution (Semanti)
    - Limited size of global buffer + don't need calculation itself, only result in last layer for ofmap (Jiwon)

- Complexity of OW bypass ordering pseudocode is O(N^2) or something else? Pseucode hard to parse in general, not much detail available. (Qishen)

- Why use ZSim instead of a different simulator? Is it because ZSim is NN specific? Why not a different simulator? (Prof. Bahar)
    - ZSim lets simulate any architecture (not just NN), GPU simulator on the other hand can only simulate GPU architectures (Jiwon)
    - ZSim is x86 generic architecture simulator? (Yash)

- Is HMC (Micron) in production or is it just a proof of concept? Is it widely available or only a prototype? (Yash)
    - At time of writing paper was a concept, probably not physically made but viable. HVM (2.5D solution) is available however. (Jiwon)

- Why use old 45nm technology. Relatively old. Why use 45nm architecture? Is it because easier to get all the specs for it? (Prof. Bahar)
    - Maybe baseline architectures have 45nm so for comparison they did it (Semanti)

- Is this particular application well suited to DRAM because it is computationally hot or is it because DRAM has some other detail? (Sam)

- Seems only benefit of 3D DRAM for this paper is random access? Doesn't seem 2D and 3D sequential applications would have a benefit? Why take advantage of random access feature of DRAM? (Sam)
    - Inherent in the way that neural networks work. (Prof. Bahar)

- Comparing 256 kB of SRAM with 16Gb LPDDR3 in the evaluation part of the paper. No 3D SRAM. (Prof. Bahar, Jiwon, Mark)

- All the references in the paper used for comparing have different numbers of PEs but exact same energy usage. Odd, probably not possible. PE Dynamic cannot be exact same in Figure 8 in paper. (Jiwon)

- Power differences between 1 vault - 16 vault. Power/vault, total power usage. No normalized power results in paper - only energy usage. Authors should show normalized power results in the paper.