Low Complexity Multiply-Accumulate Units for Convolutional Neural Networks with Weight-Sharing

James Garland & David Gregg

The University of Dublin Trinity College
School of Computer Science and Statistics

jgarland@tcd.ie, dgregg@tcd.ie

Trinity College Dublin
The University of Dublin

Abstract

- Convolutional neural networks (CNNs) are very successful deep machine learning technologies.
- CNNs require large amounts of processing capacity and memory bandwidth.
- Proposed hardware accelerators typically contain large numbers of multiply-accumulate (MAC) units.
- One CNN accelerator approach is "weight sharing":
  - Full range of trained CNN weight values are stored in bins;
  - Index to bin is used instead of the original weight value, thus reducing data sizes and memory traffic.
- We propose here a novel multiply-accumulate (MAC) circuit that exploits binning in weight-sharing CNNs.
- Rather than computing the MAC directly we:
  - Count the frequency of each weight and place the count in a bin.
  - Compute the accumulated value in a subsequent multiply phase.
- Proposal allows hardware multipliers in the MAC circuit to be replaced with adders and selection logic.
- Results in fewer gates, smaller logic, and reduced power with a slight latency increase in application specific integrated circuit (ASIC).
- Results in fewer cells, reduced power when implemented in resource-constrained field programmable gate arrays (FPGAs).

Motivation

- CNNs have lots (100,000s or more) multiply and accumulates.
- Hardware multipliers are large / power hungry in ASICs, Fig. 1

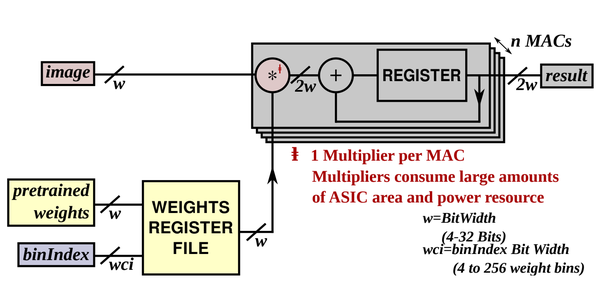

Figure 1: Weight Shared MAC

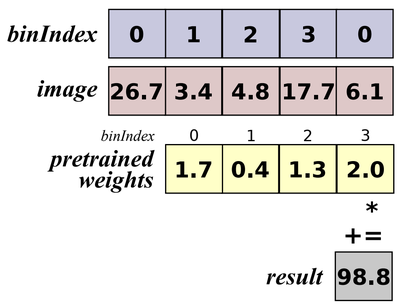

Figure 2: Weight Shared Mac Phase

- An existing approach shows relatively few weight multiplicand values needed in a CNN (weight-sharing).
- We redesign the multiply-accumulate of a weight-shared CNN into a series of accumulators followed by a multiply-accumulate phase.
- Lower power and smaller area in ASIC and FPGA hardware with slightly increased latency.

Background

- Hardware accelerators for CNN use 8-, 16-, 24- or 32-bit fixed point arithmetic [1].
- A combinatorial w-bit multiplier requires $O(w^2)$ logic gates to implement (a large part of the MAC unit).
- Han et al. [2, 3] propose an architecture for accelerating CNNs with their weight-sharing scheme, Fig 1.
- Simple MAC - pre-trained weight values are stored in a weights register file.
- Weights are indexed and retrieved by ***binIndex*** and multiplied by the corresponding image value, Fig 2.

Approach

- We propose parallel accumulate shared MAC (PASM) Fig. 3, which is multiple parallel accumulate and store (PAS) units followed by one shared MAC.
- PAS accumulates image w-bits wide, indexed into image accumulation register file with ***binIndex*** $b = 2^{wci}$.
- Post-pass MAC phase multiplies the weights with clustered image values indexed by ***binIndex***.

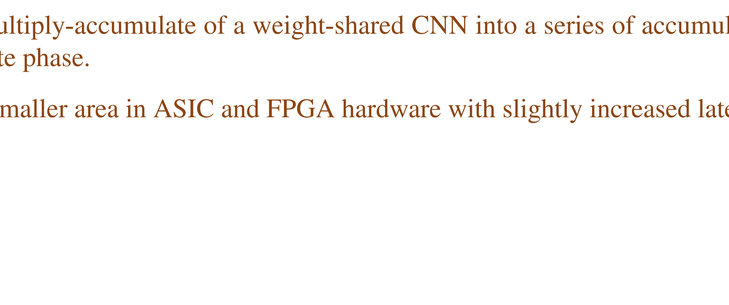

Figure 3: Multiple PAS-Shared-MAC (PASM)

- PASM has two phases:
  - PAS phase — accumulate image values into corresponding bins addressed by ***binIndex***, Fig. 4.
  - Post-pass MAC phase — multiply image bin values with weights at address ***binIndex***, Fig. 5.

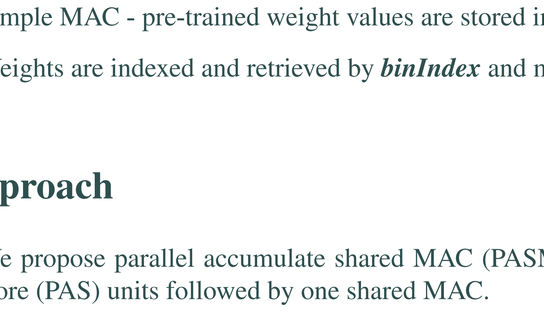

Figure 4: PAS Phase

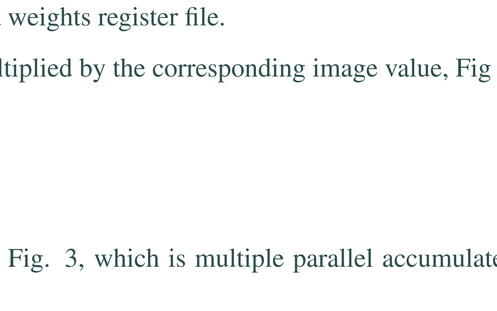

Figure 5: Post-pass MAC Phase

- PASM implemented in a CNN layer, Fig. 6.
  - M PAS, single MAC -fewer multipliers therefore reduced area and power!

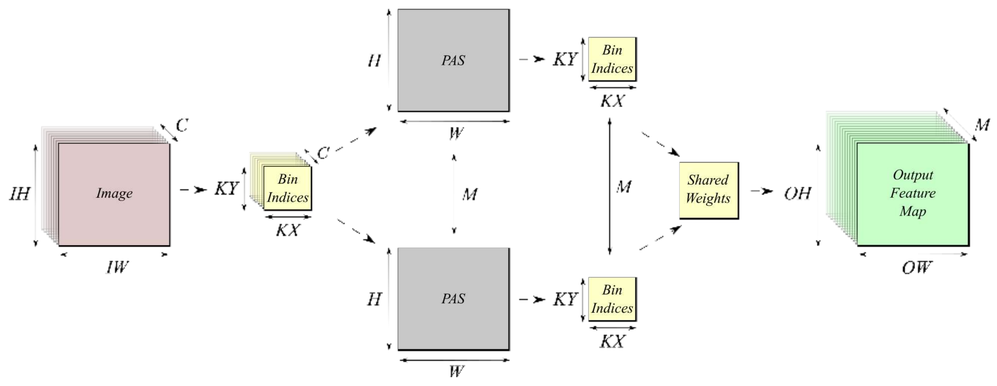

Figure 6: Pasm Implemented CNN

Evaluation

- Comparison of utilization results shows 48% fewer logic gates for a 4-bin PASM, Fig. 7.

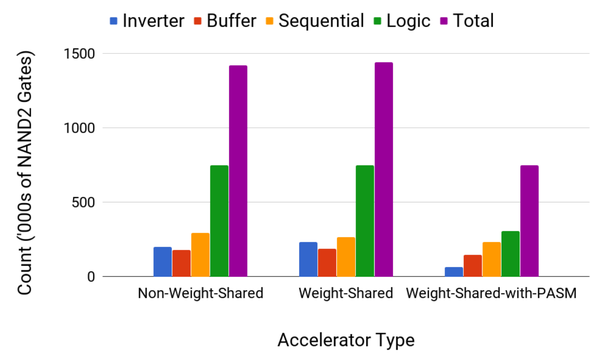

Figure 7: Utilization Comparison of PASM

- Comparison of power consumption results shows a 4-bin PASM consumes 53% less total power, Fig. 8.

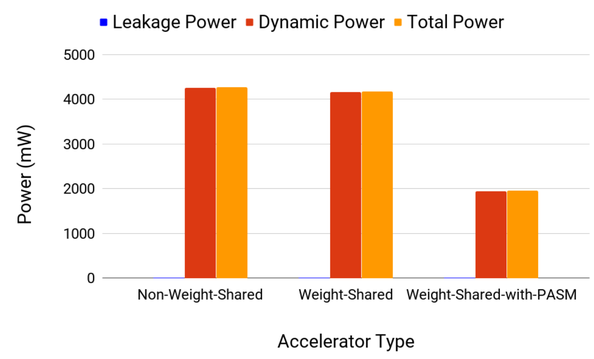

Figure 8: Power Comparison of PASM

- Latency is dependent upon number of bins. 4-bins has lowest latency increase of 8.55%, Fig. 9.

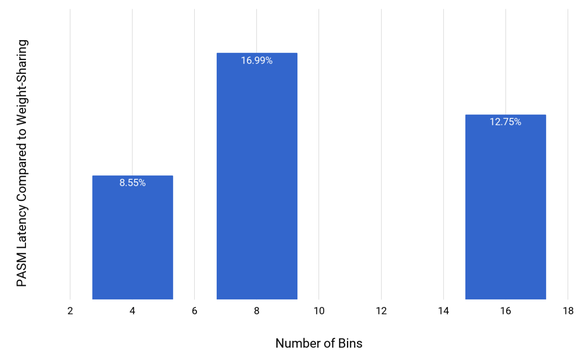

Figure 9: Latency of differing number of Bins of PASM

Publications

- IEEE CAL DOI: 10.1109/LCA.2017.2656880 - 3 citations at present.
- ACM TACO DOI: 10.1145/3233300 - 1 citation at present.

Conclusions

- MAC re-engineered into a series of accumulators followed by a multiply-accumulate phase (PASM).
- Lower power and smaller area in hardware for varying b bins and varying w bit widths with slight increase in latency.

References

[1] Y. H. Chen, J. Emer, and V. Sze. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *ACM/IEEE 43rd Annual Int. Symp. Comp. Archit.*, pages 367–379, 2016.

[2] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz, and William J. Dally. EIE: Efficient inference engine on compressed deep neural network. In *Proc. 43rd Int. Symp. Comp. Archit.*, pages 243–254, 2016.

[3] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR*, abs/1510.00149, 2015.

Acknowledgements

This research is supported by Science Foundation Ireland, Project 12/IA/1381. We are very grateful to the Institute of Technology Carlow, Ireland for their support.

Science Foundation Ireland sfi For what's next