Field-Configurable Multi-resolution Inference: Rethinking Quantization

Sai Qian Zhang[1], Bradley McDanel[2], H.T. Kung[1], and Xin Dong[1]

[1]Harvard University
[2]Franklin and Marshall College

1. Motivation

Quantization has emerged as one of the most important techniques for making Deep Neural Network (DNN) inference more efficient. A recent research focus has been on low-resolution uniform quantization (e.g., 4-bit bit-width) for weight and data values. The hardware assumed for this type of quantization is typically straightforward bit-parallel multiplier-accumulator (MAC) designs. Therefore, the motivation for reducing the precision comes down to less data movement and more efficient compute engines (e.g., 4-bit MACs instead of 8-bit MACs). The compute engine is static in that lower-resolution (e.g., 3-bit) values will not see any significant computational benefit when implemented on a 4-bit MAC.

In this work, we design a MAC that can inherently support multiple resolutions. To this end, we use a relatively new form of quantization called term quantization [7], which operations on a budget of nonzero bits (or digits, for signed-digit representations) in a group of values as opposed to simply truncating the same lowest-order bits of individual values as in conventional uniform quantization. Via term quantization, we build a multi-resolution multiplier-accumulator (mMAC) which can share terms in **efficient** support of a wide range of term budgets, corresponding to different levels of quantization.

To the best of our knowledge, no prior work has explored the design of a MAC that supports multiple resolutions. This problem is important, as many DNN inference scenarios can have a large variation in the computation requirements of the system. Our mMAC system enables a single meta multi-resolution DNN to efficiently support a wide range of configurations while achieving a good performance/cost trade-off.

2. Limitations of the State of the Art

As this paper covers multiple areas, we will describe each area separately in order to discuss their respective limitations.

DNN Training Supporting Performance/Cost Trade-off: In recent years, there has been a trend towards designing neural network that achieve an on-demand performance/cost trade-off. The most similar work to our approach is Once-For-All [3], which allows for multiple sub-models to be trained jointly using a teacher-student training paradigm. While their work derives sub-models that share **weight values**, our work proposes an approach that shares **weight terms**. Term sharing allows for additional flexibility in weight representations and therefore additional performance/cost trade-off benefits.

Quantization: Quantization has been studied extensively for reducing the associated storage, I/O, and computation costs of DNNs. The paper which proposed term quantization [7] used it only for post-training quantization. Our paper shows that term-quantization-aware model training can substantially improve the performance over just post-training term quantization. Additionally, we show how to efficiently support field-configurable multi-resolution term quantization.

DNN Hardware Exploiting Bit-level Sparsity: Some recent works [1, 4, 10, 7] have observed a high degree of bit-level sparsity in DNNs which can be exploited in hardware design to reduce the computational cost of inference. However, these works assume pre-trained networks and do not consider multi-resolution scenarios. In this work, we demonstrate that term-quantization aware training is critical for the performance of multi-resolution DNNs as noted above.

3. Key Insights

- We can train a **single meta DNN capable of spawning sub-models of varying precision** during inference.
- We can **share term across multiple sub-models** and still achieve good performance (Figure 15). This is enabled by the proposed multi-resolution training approach.
- A **single mMAC system** supports efficient implementation for multiple sub-models of varying resolutions (Figure 20).

4. Main Artifacts

4.1. Meta Multi-resolution DNN Training

Description: To support field-configurable multi-resolution inference, we have developed a DNN training approach that jointly optimizes multiple sub-models of varying resolution. The result is a single meta multi-resolution model capable of supporting multiple resolutions at runtime, with two novel properties: **storage sharing** across the sub-models, as the same non-zero terms are shared across sub-models, and **computation sharing** as all sub-models can use the same mMAC computation engine. To implement different quantization resolutions, we simply adjust the number of leading non-zero terms across groups of weights.

Evaluation: We evaluate the performance of the multi-resolution training approach in Section 6 on the following:

- (Section 6.1) How much performance is lost by enforcing term sharing instead of training each sub-model separately? The multi-resolution model is 0.25% to 1.25% worse on

ResNet-18 [5] than sub-models trained individually (Figure 15). The largest gap is for lowest-resolution sub-model.
- (Section 6.2) How does the distribution of weight values change across sub-models? We find that term quantization provides additional quantization levels which interpolate between uniform and logarithmic quantization, enabling a fine-grain performance/cost trade-off (Figure 16).
- (Section 6.4) How does uniform quantization (with varying bit-widths) compare to term quantization (with varying term budgets) under bit/term sharing? Enforcing sharing across multiple uniform quantization resolutions leads to significantly worse model performance (Figure 18).
- (Section 6.5) What is cost (*e.g.,* total runtime, memory consumption) of Meta Multi-resolution DNN training? Our proposed Meta Multi-resolution DNN training selects two sub-models to optimize for every iteration, which leads to approximately a 2× increase in runtime and memory consumption compared to training a single model (Table 3). However, for only a 2× increase, the resulting meta model can select from one of eight sub-models.

4.2. mMAC System

Description: An mMAC design that inherently supports multiple resolutions. The mMAC operates on only the non-zero power-of-two terms in a value. For example, for the value $20 = 00010100_2$, mMAC only operates on the two terms, 2^4 and 2^2, corresponding to the two nonzero bits in the value. Unlike in mMAC, in a conventional MAC, 0 bits above the least significant 1 bit require processing (e.g., the 0 in the middle of 101 bitstream for the value 20).

Via our Meta Multi-resolution training regime (Algorithm 1), the weight terms for all lower-resolution sub-models are shared with higher-resolution sub-models. This term sharing means that it is sufficient to store only the largest sub-model. Our mMAC system implements an efficient memory access by storing the term increments in the consecutive memory entries (Figure 15). Therefore, only a subset of terms are loaded from the memory when performing inference with a low-resolution sub-model.

Evaluation: We evaluate the performance of our mMAC system in Section 7 using a Xilinx VC707 FPGA evaluation board. We consider the following questions:

- (Section 7.1) How does the mMAC design compared to conventional bit-serial MAC (bMAC) and bit-parallel MAC (pMAC) on FPGA energy efficiency? mMAC achieves a 3.1× and 5.6× higher energy efficiency on average across term-pair budgets than bMAC and pMAC, respectively.
- (Section 7.2) How well does the mMAC system support a wide range of term-pair budgets in terms of latency and energy efficiency? For MobileNet-V2, the processing latency reduces by 2.7× and the energy efficiency increases by 2.5×, as the term-pair budget γ decreases from 60 to 16 (Figure 20). This shows that our mMAC system can efficiently adjust its computational cost based on γ.
- (Section 7.3) For a fixed-resolution setting, how does the mMAC system compare to other FPGA designs? On average, our system outperforms the other designs by 1.7× and 3.28× in terms of the processing latency and energy efficiency, while achieving a high classification accuracy.

5. Key Results and Contributions

Key Empirical Results:

- The multi-resolution paradigm allows a single meta model with multiple sub-model settings (up to 8 in this paper), with only moderately reduced performance compared to training them individually (Figure 15).
- Via term quantization, the multi-resolution paradigm can have the required flexibility to achieve high performance across a wide range of settings (Figure 18).
- The mMAC approach broadens the set of opportunities in trading off in cost, efficiency, and latency across a range of term-pair budgets (Table 2 and Figure 20) compared to conventional MAC designs.

Key Contributions:

- A *multi-resolution hardware system with mMAC* for supporting field-configurable multi-resolution DNN inference. The mMAC computes the dot products by processing only the non-zero terms in weight and data values.
- A *multi-resolution training paradigm* for efficient joint training of a single meta multi-resolution model capable of spawning multiple sub-models that share power-of-two terms. The method uses a teacher-student approach to train two sub-models at each iteration.
- *Sub-model configuration* at inference to meet the current resource constraints by simply adjusting the number of leading terms to use in learned weights of the meta model.

6. Why ASPLOS

ASPLOS showcases novel system architecture support on multidisciplinary research covering software design and hardware implementation. Multiple works published in ASPLOS have focused on hardware/software co-design for efficient DNN inference [8, 4, 6, 2, 9]. Our work proposes a full-stack architecture approach to support multi-resolution inference which can adapt to various deployment scenarios. As our work has both a strong architectural component (mMAC system) and model design component (multi-resolution training), we believe it fits the multidisciplinary nature of ASPLOS.

7. Citation for Most Influential Paper Award

"Field-Configurable Multi-resolution Inference" is the pioneering work on multi-resolution DNN deployment. Via term quantization, the work demonstrates that a single meta model can spawn sub-models of varying resolutions with low system overheads and minimal performance loss.

References

[1] Jorge Albericio, Alberto Delmás, Patrick Judd, Sayeh Sharify, Gerard O'Leary, Roman Genov, and Andreas Moshovos. Bit-pragmatic deep neural network computing. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 382–394. ACM, 2017.

[2] Aayush Ankit, Izzat El Hajj, Sai Rahul Chalamalasetti, Geoffrey Ndu, Martin Foltin, R Stanley Williams, Paolo Faraboschi, Wen-mei W Hwu, John Paul Strachan, Kaushik Roy, et al. Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 715–731, 2019.

[3] Han Cai, Chuang Gan, and Song Han. Once for all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[4] Alberto Delmas, Patrick Judd, Dylan Malone Stuart, Zissis Poulos, Mostafa Mahmoud, Sayeh Sharify, Milos Nikolic, and Andreas Moshovos. Bit-tactical: Exploiting ineffectual computations in convolutional neural networks: Which, why, and how. *24th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] H. T. Kung, Bradley McDanel, and Sai Qian Zhang. Packing sparse convolutional neural networks for efficient systolic array implementations: Column combining under joint optimization. *24th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019.

[7] H. T. Kung, Bradley McDanel, and Sai Qian Zhang. Term revealing: Furthering quantization at run time on quantized dnns. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020.

[8] Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna. Maeri: Enabling flexible dataflow mapping over dnn accelerators via programmable interconnects. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operation Systems (ASPLOS)*, 2018.

[9] Wei Niu, Xiaolong Ma, Sheng Lin, Shihao Wang, Xuehai Qian, Xue Lin, Yanzhi Wang, and Bin Ren. Patdnn: Achieving real-time dnn execution on mobile devices with pattern-based weight pruning. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 907–922, 2020.

[10] Sayeh Sharify, Alberto Delmas Lascorz, Mostafa Mahmoud, Milos Nikolic, Kevin Siu, Dylan Malone Stuart, Zissis Poulos, and Andreas Moshovos. Laconic deep learning inference acceleration. In *Proceedings of the 46th International Symposium on Computer Architecture*, pages 304–317. ACM, 2019.