ILMPQ : An Intra-Layer Multi-Precision Deep Neural Network Quantization framework for FPGA

Sung-En Chang*[1], Yanyu Li*[1], Mengshu Sun*[1], Yanzhi Wang[1], Xue Lin[1]
[1]*Northeastern University,*
{chang.sun, li.yanyu, sun.meng, yanz.wang, xue.lin}@northeastern.edu

***Abstract*—This work targets the commonly used FPGA (field-programmable gate array) devices as the hardware platform for DNN edge computing. We focus on DNN quantization as the main model compression technique. The novelty of this work is: We use a quantization method that supports multiple precisions along the intra-layer dimension, while the existing quantization methods apply multi-precision quantization along the inter-layer dimension. The intra-layer multi-precision method can uniform the hardware configurations for different layers to reduce computation overhead and at the same time preserve the model accuracy as the inter-layer approach. Our proposed ILMPQ DNN quantization framework achieves 70.73% Top1 accuracy in ResNet-18 on the ImageNet dataset. We also validate the proposed MSP framework on two FPGA devices i.e., Xilinx XC7Z020 and XC7Z045. We achieve $3.65\times$ speedup in end-to-end inference time on the ImageNet, comparing with the fixed-point quantization method.**

I. Introduction

Deep neural net (DNN) quantization is an crucial technique to reduce the computation, memory, and storage requirements executing on-device inference, especially for platforms with capability of customized architecture design, such as FPGA devices and ASIC chips.

Various quantization schemes have been proposed, such as binary [3], ternary [4], and Power-of-Two (PoT) [5]. These works can significantly reduce the computation but suffer from non-neglectable accuracy degradation. On the other hand, Fixed-point (Fixed) [2], [6] yields relatively small accuracy loss while still needs expensive multiplication operations during inference computation.

To address this issue, this work proposes a novel DNN quantization framework, namely ILMPQ, which is Intra-Layer Multi-precision quantization approach. Specifically, Figure 1 shows that each filter of the weight tensor or each row in the weight matrix can be assigned with a specific configuration of quantization scheme and precision. The candidates of schemes and precisions are assigned to facilitate most efficient hardware implementation, while with the capability to preserve the accuracy as the unquantized (32-bit floating-point) baseline models. This highly hardware-informative quantization strategy significantly reduces the search space of the DNN quantization problem, making our framework distinctive from existing multi-precision quantization work.

The contributions of our quantization framework are summarized as follows:

- **We propose an *Intra-Layer Flexibility*, which can be applied to all layers in a DNN model, achieving the best accuracy performance without damaging hardware efficiency.**
- **A hardware-aware solution, which can significantly reducing the problem search space.**
- **The significant inference speedup on real devices, comparing to the network-wise uniform low-bit quantization i.e., the speed upper bound of existing layer-wise multi-precision approaches.**

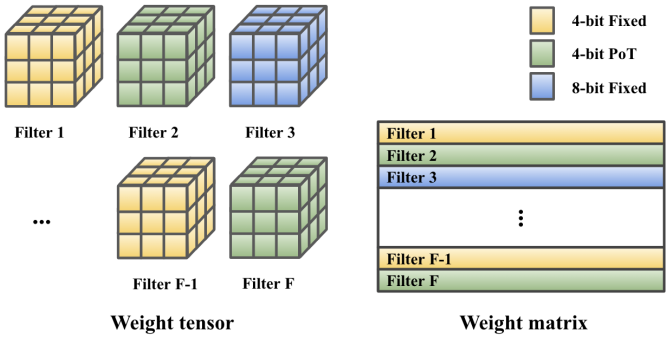

Fig. 1. **The proposed DNN quantization framework with filter-wise mixed schemes and multiple precision, which assigns quantization precision and scheme to filters of the weight tensor (or rows of the weight matrix).**

II. Proposed ILMPQ Quantization

A. Intra-Layer flexibility to preserve the accuracy

We propose *Intra-Layer Multi-precision quantization* for our quantization scheme. In each layer, we only quantize 5 percent filters of weights to 8 bit and leave the rest to 4 bit. Rather than the prior works which need to use 8 or more bits to represent the weights in the first and the last layer. Our ILMPQ can be applied in the all layers of the DNN models. This is because in *intra-layer flexibility*, the weights quantized using 8 bits can be trained to mitigate the imprecision caused by those weights quantized using fewer bits. This mitigation happens in every layer.

Besides algorithm-level advantages, the proposed intra-layer flexibility also exhibits an advantage at the FPGA hardware level. Recall that the same quantization scheme (e.g., 4-bit for 95% of weights and 8-bit for the rest of 5%) is applied to all layers of a DNN. At FPGA configuration time for a specific DNN inference task, one could allocate a portion of PEs for the low-bit portion of computation and the rest of PEs for the 8-bit portion, and this works for every layer. As for traditional

Quantization Method	PoT-4 : Fixed-4 : Fixed-8	First/Last Layer Quantization	Accuracy		Results on FPGA XC7Z020				Results on FPGA XC7Z045			
			Top-1 (%)	Top-5 (%)	Utilization LUT	Utilization DSP	Throughput (GOP/s)	Latency (ms)	Utilization LUT	Utilization DSP	Throughput (GOP/s)	Latency (ms)
(1) Fixed	0:100:0	8-bit Fixed	69.72	88.67	49%	100%	29.6	122.6	21%	100%	115.6	31.4
(2) Fixed	0:100:0	✓	68.66	87.54	45%	100%	36.5	99.3	24%	100%	142.7	25.4
(3) PoT	100:0:0	8-bit Fixed	68.20	87.14	51%	100%	62.4	58.1	40%	100%	290.5	12.5
(4) PoT	100:0:0	✓	67.11	85.93	57%	12%	72.2	50.2	44%	3%	352.6	10.3
(5) PoT + Fixed	50:50:0	8-bit Fixed	68.94	88.66	71%	100%	50.3	72.0	42%	100%	196.8	18.4
(6) PoT + Fixed	50:50:0	✓	67.98	86.75	66%	100%	75.8	47.8	38%	100%	296.3	12.2
(7) PoT + Fixed	60:40:0	8-bit Fixed	68.53	88.47	80%	100%	57.0	63.6	-	-	-	-
(8) PoT + Fixed	67:33:0	8-bit Fixed	68.46	88.22	-	-	-	-	61%	100%	245.8	14.8
ILMPQ-1	60:35:5	✓	**70.66**	**89.53**	82%	100%	**89.0**	**40.7**	-	-	-	-
ILMPQ-2	65:30:5	✓	**70.73**	**89.62**	-	-	-	-	65%	100%	**421.1**	**8.6**

TABLE I

IMPLEMENTATIONS OF DIFFERENT QUANTIZATION SCHEMES ON TWO FPGA BOARDS (XC7Z020 AND XC7Z045) FOR RESNET-18 ON IMAGENET, USING THE (EQUIVALENT) 4-BIT PRECISION.

inter-layer multi-precision scheme, it's almost impossible to perform online reconfiguration, that is, the PEs assigned to execute 8-bit first/last layers is vacant while processing the middle layers.

B. Mixed Schemes to boost the hardware efficiency

Inspired by MSQ [1], which is the state-of-the-art, FPGA specific quantization scheme. We incorporate their Mixed-Scheme quantization to our work to further boost the hardware efficiency.

As inference on FPGA is conducted layer-by-layer on GEMM core, different schemes within a layer can benefit hardware parallelism. Specifically, fixed-point convolution is done by $\texttt{GEMM}_{\texttt{Fixed}}$ and PoT is done by $\texttt{GEMM}_{\texttt{PoT}}$. We implement $\texttt{GEMM}_{\texttt{Fixed}}$ on DSP modules and $\texttt{GEMM}_{\texttt{PoT}}$ on LUT modules on FPGA, so that the heterogeneous resource can be utilized efficiently. As different FPGA device has different characteristics, the actual mixing ratio of fixed-point scheme and PoT scheme can be determined offline by examining FPGA throughput. The ideal utilization is to balance workload of the two schemes and achieve highest throughput.

C. The training process of ILMPQ framework

In the training algorithm of our ILMPQ quantization, there are two steps to determine the bitwidth and schemes within each layer. First, we compute the largest eigenvalue of Hessian matrix per filter to determine the most sensitive weights. More bits will be assigned to filters with larger eigenvalues. Then, we sort the row vectors by their variance. Rows with smaller variance are quantized to PoT, while others are quantized to fixed-point scheme. The ratio is determined offline by hardware utilization. The reason behind this assignment is that PoT scheme enjoys higher resolution around the mean area (zero) compared to fixed-point scheme, so that quantization error can be reduced if the weights to be quantized mostly fall around zero, which empirically means lower variance of weight distribution.

III. EXPERIMENTS AND EVALUATION

To present the accuracy and hardware performance with real-world applications. We compare ILMPQ with other quantization methods on the ImageNet dataset using ResNet-18, as displayed in Table I. The quantized model is trained under basic data augmentation and step learning rate on PyTorch platform. Initialized with pretrained model, the quantization-aware training takes 50 epochs. We provide hardware results on two FPGA boards, XC7Z020 and XC7Z045. Specifically, the optimal ratio on XC7Z020 is 60:35:5 (ILMPQ-1), resulting in top-1 accuracy of 70.66% and latency of 40.7ms, while the optimal ratio on XC7Z045 is 65:30:5 (ILMPQ-2), leading to top-1 accuracy of 70.73% and latency of 8.6ms. ILMPQ with the optimal ratio of quantization schemes achieves up to $3.01\times$ speedup on XC7Z020 and up to $3.65\times$ speedup on XC7Z045, with both LUTs and DSPs utilized efficiently.

IV. CONCLUSION

In this work, we propose a new dimension of DNN quantization, namely, ILMPQ, which introduce the strategy of row-wise mixed scheme and intra-layer mixed precision. We achieve state-of-art accuracy performance and up to $3.65\times$ speed up on FPGA.

ACKNOWLEDGMENT

This work is partly supported by the National Science Foundation CCF-1901378, CCF-1919117, and CCF-1937500.

REFERENCES

[1] S.-E. Chang, Y. Li, M. Sun, R. Shi, H. K.-H. So, X. Qian, Y. Wang, and X. Lin, "Mix and match: A novel fpga-centric deep neural network quantization framework," *The 27th IEEE International Symposium on High-Performance Computer Architecture (HPCA-27)*, 2020.

[2] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.

[3] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in neural information processing systems (NeurIPS)*, 2015, pp. 3123–3131.

[4] F. Li, B. Zhang, and B. Liu, "Ternary weight networks," *arXiv preprint arXiv:1605.04711*, 2016.

[5] D. Miyashita, E. H. Lee, and B. Murmann, "Convolutional neural networks using logarithmic data representation," *arXiv preprint arXiv:1603.01025*, 2016.

[6] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.