Towards CIM-friendly and Energy-Efficient DNN Accelerator via Bit-level Sparsity

Foroozan Karimzadeh* and Arijit Raychowdhury*
*School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332
Email: fkarimzadeh6@gatech.edu, arijit.raychowdhury@ece.gatech.edu

2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC) | 978-1-6654-9005-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/VLSI-SoC54400.2022.9939651

***Abstract*—The rising popularity of deep neural network (DNN) algorithms calls for energy-efficient accelerators to enable DNNs run on edge devices. In this paper, we presented BitS-Net, a bit-level sparsity method that quantize the network to desirable numbers with more zeros in their bit representation. We demonstrated that BitS-Net can preserve the accuracy (67.73 %) with accuracy drop ¡1% compared to the original network. Moreover, it achieved up to 5x energy efficiency for ResNet-18 models on the ImageNet dataset compared to the baseline methods.**

***Index Terms*—DNN model, bit-level sparsity, DNN compression, quantization, low bit precision.**

I. Introduction

Over the past decade, a rapid progress toward machine learning and specifically DNN acceleration has been made to enable these powerful methods run on the resource constrained edge devices [1]. However, due to the inherent large size of DNN methods and the expensive cost of transferring data between external DRAM memory and SRAM in traditional CMOS, technologies such as compute-in-memory (CIM) has emerged to overcome these problems by performing the computation in the memory. As mentioned in [2], the cost of fetching data from DRAM to the chip's internal memory is three orders of magnitude higher than an add operation; which emphasize that a huge amount of energy is spend to transfer data in traditional CMOS based accelerators with limited internal memory. Therefore, CIM architecture can play an important role as the next generation of DNN accelerators.

CIM using the crossbar architecture can reduce the amount of data transfer in DNN computation by computing in the memory itself. However, due to limitation in bit-line and word-line in CIM architecture, it is hard to compute DNN models with large weight matrices. On the other hand, DNN computations are usually performed in float-32 format which leads to another issue to utilize CIM architecture. The proposed CIM for DNN accelerators can perform 1 bit per word-line which limited the bit-precision that can be used for weigh values which leads to a huge information loss as a result of ultra-low bit computation (e.g. 4 bit-width).

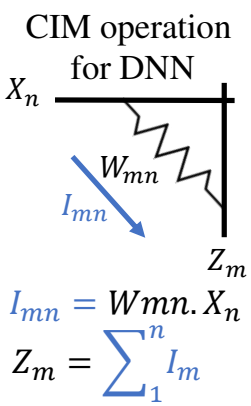
In this work, we have proposed a CIM-friendly DNN quantization framework by introducing a bit-level sparsity to quantize and sparsify the network in the bit-level during training in order to preserve the accuracy while reducing the energy requirement during the inference. In addition, we have utilized a 2-bit per cell CIM architecture proposed in [3] enabling 8-bit per cell computation to gain highe accuracy due to using higher precision values.

II. Method

In this section, the proposed bit-level sparsity method and the 2-bit/cell CIM architecture are explained.

In CIM architecture exploiting resistance of memory cells (figure 1), a memory cell itself serves as a PE and memory simultaneously [4]. In addition, CIM architectures employing emerging memory such as RRAM have achieved high energy-efficiency due to the inherent multiply-and-accumulate (MAC) functionality in BL structures. Moreover, as shown in figure 1, multiplying to a weight with zero value will result in zero current and therefore no required energy for this operation. This is the motivation of our proposed bit-sparsity method, where we quantize the network during training to increase the numbers with more zeros in their binary representation. In addition, recently a 2bit/cell CIM architecture has been developed in [3] to enable multiplication of higher bit-precision like 8 bit. Each cell can be 00, 01, 10 and 11. The energy level for multiplication to 00 is very low as opposed the energy level to 11. Based on the actual measurement, the measured energy per 2-bits for multiplying with 11, 10, 01, and 00 in the 2-bit-encoded RRAM cells in CIM architecture is 1.46 pJ/2bits, 0.73 pJ/2bits, 0.36 pJ/2bits, 79 fJ/2bits, respectively.

The goal in the proposed bit-level sparsity method is to quantize the network to the numbers with 8 bit-width precision that have more 00 and no 11 in their binary representation. We manually picked different sets of numbers as shown in Table I having this property. In another word, we want to increase the bit-level sparsity of the weigh matrices to decrease the energy required during inference. We apply the quantization during training to preserve the accuracy. The quantization scheme of the weights [5] is defined as follow where b is the bit precision, α is the scaling factor and scales the weights into $[-1, 1]$. Then the scaled W is projected by $\Pi_{(.)}$ in an element-wise manner to the defined quantization levels in Table I.



Bitwise multiplication in each memory cell

X	W	I
0	0 (R_{HRS})	0
0	0 (R_{LRS})	0
1	0 (R_{HRS})	I_{HRS}
1	1 (R_{LRS})	I_{LRS}

Fig. 1: CIM architecture using resistance of memory cells [4]

978-1-6654-9005-4/22/$31.00 ©2022 IEEE

TABLE I: The coefficient set for quantization [4] to increase the bit-level sparsity.

Set #	Desired values
1	$\pm\{0, 0.3438, 0.3750, 0.4063, 0.5, 0.6250, 0.6563, 1.0\}$
2	$\pm\{0, 0.3750, 0.5, 0.6250, 1.0\}$

$$W_q = \alpha \Pi_{Q(1,b)} \lfloor \frac{W}{\alpha}, 1 \rceil \tag{1}$$

Then each element in the original weight matrix in float32 format is quantized to a b-bit fixed-point representation. In addition, the Straight-Through Estimator (STE) is adopted in the backward path as proposed in [5] and defines as:

$$\frac{\partial W_q}{\partial \alpha} = \begin{cases} sign(W) & if\ |W| > \alpha \\ \Pi_{Q(1,b)} \frac{W}{\alpha} - \frac{W}{\alpha} & if\ |W| \leq \alpha \end{cases} \tag{2}$$

III. Result

In this section, the result of the proposed method is presented. We used ResNet-18 on Imagenet dataset to evaluate the accuracy and energy efficiency of our method. We compare the accuracy of our method (BitS-Net) with Power-of-Two (PoT) and Additive Power-of-Two (APoT) [5] and the original float-32 ResNet Model as shown in figure 2. The results show that our method can preserve the accuracy (67.73%) better which is closer to the accuracy of the original network.

In addition, we compare the energy efficiency of BitS-Net with the baseline methods. Based on the actual measurement, the measured energy per 2-bits for multiplying with 11, 10, 01, and 00 in the 2-bit-encoded RRAM cells in CIM architecture is 1.46 pJ/2bits, 0.73 pJ/2bits, 0.36 pJ/2bits, 79 fJ/2bits, respectively. We also consider the energy required for ADC in the energy computations which is 0.208 pJ/2bits. The results demonstrated the energy efficiency of BitS-Net as illustrated in figure 3.

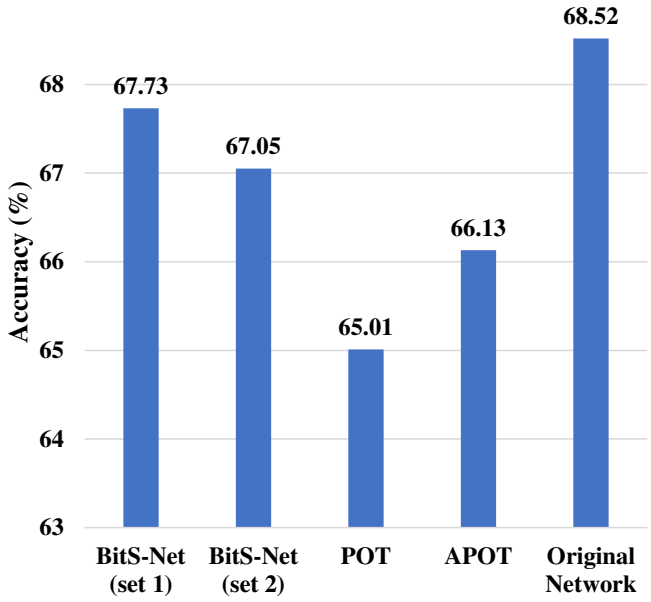


Fig. 2: Accuracy of our proposed method (BitS-Net), baseline methods (APOT and POT) and original float-32 ResNet-18 network. [4]

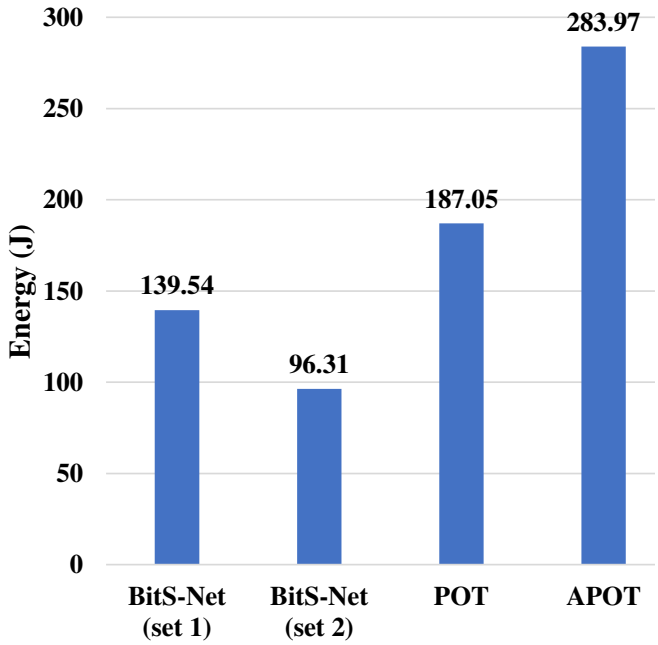


Fig. 3: The estimated energy of our proposed method (BitS-Net) and baseline methods (APOT and POT). [4]

IV. Conclusion

Using high performance DNN methods on the resource (energy and battery) constrained edge devices requires us to develop compression techniques to make efficient DNNs. In this paper, we proposed BitS-Net, a novel bit-level sparsity and CIM-friendly method by quantizing the network during training to a set of desired coefficients. We then used a 2bits/cell CIM architecture to increase the energy efficiency, as in CIM, the computation is performed in the memory. We demonstrated that BitS-net can achieve up to 5x energy efficiency while preserving the accuracy (less tha 1% accuracy reduction compared to the original float-32 network).

References

[1] F. Karimzadeh, N. Cao, B. Crafton, J. Romberg, and A. Raychowdhury, "Hardware-aware pruning of dnns using lfsr-generated pseudo-random indices," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5.

[2] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[3] J. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "29.1 a 40nm 64kb 56.67 tops/w read-disturb-tolerant compute-in-memory/digital rram macro with active-feedback-based read and in-situ write verification," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 404–406.

[4] F. Karimzadeh, J.-H. Yoon, and A. Raychowdhury, "Bits-net: Bit-sparse deep neural network for energy-efficient rram-based compute-in-memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2022.

[5] Y. Li, X. Dong, and W. Wang, "Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks," *arXiv preprint arXiv:1909.13144*, 2019.