

Implementation of a Robust and Power-Efficient Nonlinear 64-QAM Demapper using In-Memory Computing

Amro Eldebiky¹, Georg Böcherer², Maximilian Schädler², Stefano Calabrò², Bing Li¹, Ulf Schlichtmann¹

¹Electronic Design Automation chair, Technical University of Munich, 80333 Munich, Germany

²Huawei Munich Research Center, 80992 Munich, Germany

*amro.eldebiky@tum.de

Abstract: Analog in-memory computing reduces power consumption sacrificing computational accuracy. We implement multiplication-accumulation in resistive RAM accounting for non-idealities (variations, quantization, ADC noise). The floating-point performance is recovered while minimizing power consumption in offline 64-QAM experiments. © 2022 The Author(s)

1. Introduction

Nonlinear equalization and demapping are instrumental in high-speed optical communications to compensate transmission impairments. Recently, neural networks (NNs) are proposed to implement equalization and soft demapping of received symbols [5, 7, 9, 13]. NNs are suitable to implement high-speed processing in optical communications due to parallelization and the availability of large amount of training data.

However, digital implementations of NNs are power-hungry due to the huge number of multiply-accumulate (MAC) operations and memory accesses. Analog in-memory computing (IMC) based on emerging devices, e.g., resistive RAMs (RRAM), is introduced [14] to tackle such challenges. In IMC accelerators, the weights of NNs are represented by the conductances of RRAM cells. MAC operations are realized by Ohm's and Kirchhoff's laws, so that a high computation and energy efficiency are achieved. IMC is reported to achieve a 17-time higher energy efficiency than digital implementation [10]. However, IMC platforms suffer from hardware (HW) non-idealities, namely weight variations, coarse weight quantization, and ADC noise.

Previous work addresses the impact of weight deviations by training approaches. In [4], NN soft-demappers are trained under an optimized Lipschitz-constant constraint to prevent error amplification through the layers. However, such approach does not take into account the low-level RRAM architecture.

In this work, we integrate the Lipschitz-method into a HW-aware training framework. We ①combine the Lipschitz constraints with quantization aware-training to consider HW implementation (variations, weight/activation quantization, ADC noise), ②optimize weight/activation quantization by exhaustive search to minimize the bit width and study quantization effects on robustness and power, ③consider different splitting approaches for representations of weights into more than one RRAM cell.

The remainder of the paper is structured as follows: In Sec. 2 and 3, we state our system model and methodology, respectively, and in Sec. 4, we discuss the experimental setup and results.

2. System model and problem statement

Fig. 1(a) shows a crossbar architecture [11] able to represent positive and negative weights and shows HW non-idealities in an IMC accelerator. The first HW non-ideality is the deviation of conductance values written to RRAM cells. Physical parameter variations and errors of RRAM cells [12] cause the programmed conductance to deviate from the nominal value. Consequently, the feature maps at the output of the layers become erroneous. The most used model to describe weight deviations is the log-normal model [3] shown in Fig. 1(a), where $w_{nominal}$ is the nominal value of a trained weight, and θ is an independent normally distributed random variable with standard deviation σ . The second HW non-ideality is the limited number of usable conductance levels in a RRAM cell which dictates the max number of bits stored in one cell, usually to 6-7 bits at most [10]. Furthermore, the resistor noise and "kT/C" of the ADC converting the result of an analog MAC to digital domain cause a code-transition error expressed as input-referred noise [8]. The ADC noise is modeled as a Gaussian noise added to the input of an ideal ADC. The standard deviation σ_{ADC} in a properly dimensioned systems is in the range of 0.5 LSBs [6].

Besides, the quantization setting of each layer's output dictates the ADC size. ADC power scales exponentially with the resolution [2] and represents around 50% of the total power of IMC accelerators [14].

As a problem demonstration, we consider the NN soft-demapper in [13]. The output layer has m neurons where m is the number of bits per real dimension, i.e., $m = 3$ for 64-QAM. The achievable rate per real dimension is calculated [1] as in Fig. 1(b), where n is the number of training symbols. l_{ij} and b_{ij} are the j -th soft bit in the NN output and the true output for the i -th training symbol, respectively. The sign of l_{ij} defines the hard decision and the magnitude defines the confidence. Activations and weights were quantized to 4 bits, i.e. the A4W4 format is used. The model is tested under different weight variation levels represented on the x-axis, and 2 different ADC noise levels of 0.5 LSB and 1 LSB. The dataset used is obtained with the experimental setup in Fig. 2. 250 NN

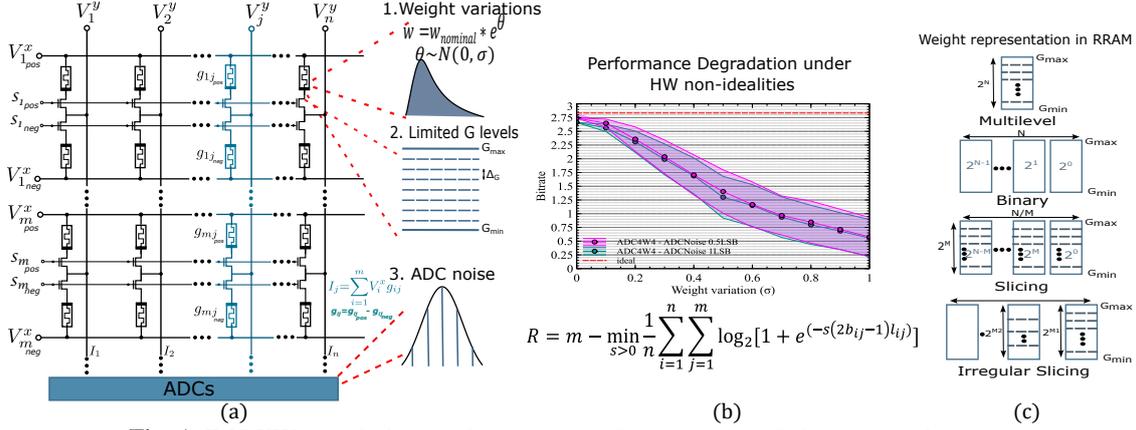


Fig. 1. IMC HW non-idealities and impact on performance of a soft-demapper architecture.

weights samples were simulated at each σ value. The solid lines represent the mean values and the ranges represent the standard deviation of the bitrate. According to Fig. 1(b), the achievable rate degrades significantly even with relatively small variations, which makes the NN demapper unusable in practice. At lower weight variations, the degradation is determined by the ADC noise, whereas, the weight variations dominate the error at higher values.

3. Methodology

3.1. Quantization-aware Lipschitz constrained training

Lipschitz-constrained training of soft demapper is presented in [4] to increase robustness against weight variations. The Lipschitz constant determines how an error at the input of a NN layer is magnified through the layer. The error magnification in a layer is suppressed by constraining the Lipschitz constant. The Lipschitz constant of each layer is upper bounded by the spectral norm of the weight matrix as follows.

$$\sup \left(\frac{|\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2)|_p}{|\mathbf{x}_1 - \mathbf{x}_2|_p} \right) = \|\mathbf{w}\|_p \leq k, \quad \phi(\mathbf{w}_i, k_i) = \frac{1}{\max(1, \frac{\|\mathbf{w}_i\|_2}{k_i})} \mathbf{w}_i \quad (1)$$

where x_1, x_2 are the nominal input to a layer, and the input affected by deviations in previous layers, respectively. Equation (1) guarantees the L^2 norm of \mathbf{w}_i does not exceed k_i . The value k_i for each layer is optimized numerically.

However, in its original formulation, the Lipschitz constraints do not consider the limited number of bits which can be written to one RRAM cell [10]. We address this by combining Lipschitz constraining with quantization aware training. During the start of the training, quantization is represented through soft tanh functions. As the training proceeds, the softness is reduced moving towards hard quantization at the end of the training. The soft quantization during training allows gradient backpropagation which is not possible in case of hard quantization.

3.2. Optimized quantization configuration

The quantization affects the robustness against HW non-idealities and the HW complexity as previously discussed. Accordingly, the quantization settings need to be optimized with the Lipschitz constraints. The search is performed by a greedy algorithm. The best Lipschitz constraint is determined for each layer by particle swarm optimization (PSO). Next, an exhaustive search is executed for the quantization settings allowing weight quantization from 1 to 8 bits and activations quantization from 1 to 4, and 8 bits. Each combination is evaluated with respect to robustness according to [4], besides, the total ADC and DAC power is evaluated based on UMC 90nm technology as in [2].

3.3. Optimized weight representation

Different weight mappings to RRAM cells are explored. As shown in Fig. 1(c), the first approach is the multilevel mapping which maps a N -bit weight to one cell. In this case, the cell has to have 2^N conductance states. The second approach is the binary representation which splits each N -bit weight to N cells. Each cell stores one bit of the weight so only 2 conductance states G_{max}, G_{min} are utilized. The weighted sum of the individual currents is obtained in the analog domain. The individual currents are amplified by gains representing the significance of each column in the weighted sum. The third mapping follows a slicing scheme in which each RRAM cell holds M bits. The number of cells needed for a weight is N/M and the cell should have 2^M conductance states. The weighted summation is obtained similarly to the binary case but with different gains depending on M . The last representation is the irregular slicing in which each RRAM cell holds a different number of bits such that the most significant bits are distributed over more cells, while the least significant bits are stacked in fewer cells.

4. Experimental results & discussion

The NN demapper architecture in [13] is trained with quantization-aware Lipschitz constraints considering HW non-idealities and tested at different non-ideality levels regarding weight variations, quantization, and ADC noise. The NN weights were sampled 250 times according to the variation model. For each sample, the achievable rate was evaluated as in Fig. 1(b). The ADC noise is modeled as described previously in [8].

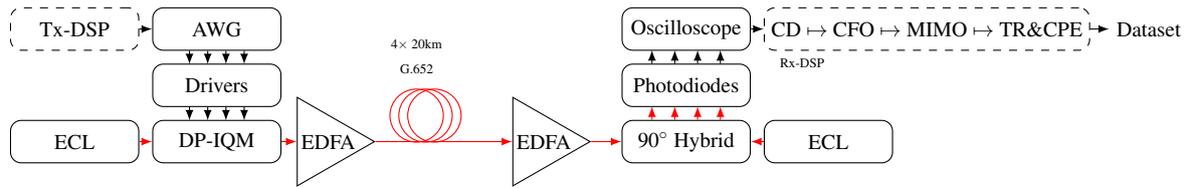


Fig. 2. An experimental coherent single carrier transmission; 80km G.652 fiber link; optimal launch power: 6.6 dBm. 80GBd DP-64QAM signal; gross data rate: 960Gb/s; FEC overhead: 15%; training sequences overhead: 3.47%; net bit rate: 800Gb/s; At the transmitter, a constant amplitude zero auto-correlation (CAZAC) training sequence for framing, carrier frequency offset and channel estimation; four 60GHz 3dB-bandwidth amplifiers for the electrical signals of the arbitrary waveform generator (AWG); two tunable 100 kHz external cavity lasers (ECLs) are used at the transmitter and receiver; a booster EDFA amplifies the optically modulated signal; The receiver: optical 90°-hybrid and four 100GHz balanced photodiodes; an oscilloscope with 256GSa/s and 110GHz 3dB-bandwidth digitizes the electrical signals.

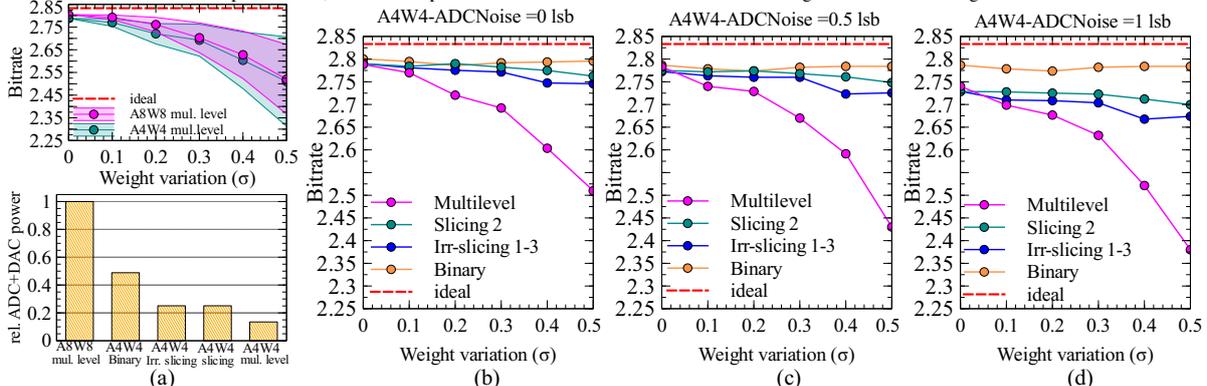


Fig. 3. Quantization search results and total ADC, DAC power for different weight representations and achievable rate at different ADC noise and weight variation level.

The start for the quantization search is 8-bit activations and 8-bit weights (A8W8). The optimized quantization setting returned from the search is A4W4. The performance comparison with the starting point A8W8 is shown in Fig. 3(a) top. As shown, A4W4 maintains the robustness of A8W8, while satisfying the RRAM cell conductance states limit and reducing the total ADC and DAC power by 86% for multilevel mapping.

The considered mappings of the 4-bit weights among cells are: binary, regular slicing with $M = 2$, and irregular slicing with one cell holding the MSB and one cell holding the 3 LSBs. The binary representation has the highest HW overhead because each weight is represented with 4 cells. Regular and irregular slicing would result in the same overhead. Fig. 3(a) bottom shows comparison of the total ADC and DAC power consumption of the different mappings of A4W4 and A8W8 multilevel. Fig. 3(b), (c), and (d) show the bitrate for different weight representations at different weight variations along the x-axis, and different ADC noise levels in the different subfigures. We tested weight variations up to $\sigma = 0.5$. This variation setting is already very large for RRAM cells [3]. The tested ADC noise levels σ_{ADC} are 0, 0.5, and 1 LSB. The line labeled as ideal denotes the performance of the software floating point double precision NN. The experiments show that, as expected, the binary representation achieves the highest robustness as each cell utilizes only two levels. However, this comes at the cost of a larger crossbar size. A good balance is shown in the performance of regular and irregular slicing as their performance is still close to the binary mapping especially with the typical ADC noise levels of 0.5 LSB [6]. Experiments demonstrate that the achieved bit rate can be recovered from as low as 1.4 to 2.78, 2.75, 2.73, or 2.43 bits per channel use depending on the used weight representation for a 64-QAM NN demapper.

References

1. G. Böcherer et al. Probabilistic shaping and forward error correction for fiber-optic communication systems. *Journal of Lightwave Technology*, 2019.
2. T. Bos et al. Architecture optimization for energy-efficient resolution-scalable 8–12-bit sar adcs. *Analog Integrated Circuits and Signal Processing*, 2018.
3. L. Chen et al. Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017.
4. A. Eldebiky et al. Power-efficient and robust nonlinear demapper for 64qam using in-memory computing. In *European Conference on Optical Communication (ECOC)*, 2022.
5. S. Fujisawa et al. Nonlinear impairment compensation using neural networks. In *Optical Fiber Communication Conference*, 2021.
6. J. Hu et al. A 9.4-bit, 50-ms/s, 1.44-mw pipelined adc using dynamic source follower residue amplification. *IEEE Journal of Solid-State Circuits*, 2009.
7. B. Karanov et al. Deep learning for communication over dispersive nonlinear channels: performance and comparison with classical digital signal processing. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 192–199. IEEE, 2019.
8. W. Kester. Adc input noise: the good, the bad, and the ugly. is no noise good noise? *Analog Dialogue*, 40(02):1–5, 2006.
9. T. Koike-Akino et al. Neural turbo equalization: Deep learning for fiber-optic nonlinearity compensation. *Journal of Lightwave Technology*, 2020.
10. C. Li et al. Analogue signal and image processing with large memristor crossbars. *Nature electronics*, 2018.
11. Q. Liu et al. 33.2 a fully integrated analog rram based 78.4 tops/w compute-in-memory chip with fully parallel mac computing. In *2020 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 500–502. IEEE, 2020.
12. D. Niu et al. Impact of process variations on emerging memristor. In *Design Automation Conference (DAC)*, 2010.
13. M. Schädlér, G. Böcherer, and S. Pachnicke. Soft-demapping for short reach optical communication: A comparison of deep neural networks and volterra series. *Journal of Lightwave Technology*, 39(10):3095–3105, 2021.
14. A. Shafiee et al. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *International Symposium on Computer Architecture (ISCA)*, 2016.