A Vision for Edge AI: Robust Binarized Neural Networks on Emerging Resource-Constrained Hardware

Mikail Yayla
TU Dortmund University
Lamarr Institute for ML and AI
mikail.yayla@tu-dortmund.de

Advisor: Prof. Dr. Jian-Jia Chen
TU Dortmund University
Lamarr Institute for ML and AI
jian-jia.chen@tu-dortmund.de

I. Introduction

To achieve high accuracy, Neural Networks (NNs) use a massive number of parameters and an immensely large amount of MAC operations need to be computed. Therefore, two main bottlenecks can be identified for realizing system that execute NNs: The memory subsystem and the processing units.

On the NN model side, Binary NNs (BNNs) offer high efficiency in both memory and computations at small accuracy cost compared to conventional integer or floating point based NNs. The binary nature of the weights and inputs brings three major benefits: (1) Reduction of data movements between memories and computing units, (2) replacing the multiplications with bitwise XNOR and accumulations with popcount operations, and (3) high error tolerance, i.e. robustness to perturbations in the weights, inputs, and intermediate data.

On the hardware side, the principles of approximate computing are applied on the memory and the processing units, where efficiency is traded for quality of the operation's result. In approximate memory, the supply voltage and access latency parameters of the memory are configured, to achieve lower energy consumption and faster access at the cost of reliability in form of bit errors. When NNs are executed using approximate computing units, configurable approximate circuits can be finely tuned to trade resource usage for computing errors. *BNNs synergize excellently with approximate memory and computing due to their high robustness and efficiency*.

To acquire BNN models with high accuracy, they have to be trained with large resource cost. The training of BNNs is by orders of magnitude more resource-intensive than inference. An efficient approach is to train BNNs on dedicated low-power accelerators in an on-chip setting, such as FPGAs or ASICs. In addition to energy-efficiency of on-chip training, privacy issues and data transfer overheads are eliminated, since the data does not need to be transferred to the cloud for training. For these reasons, resource-efficient models, such as BNNs, should not only be executed but also be trained on the edge.

Vision of this thesis: This thesis proposes a vision for highly resource-efficient future intelligent systems that are comprised of robust BNNs operating with approximate memory and approximate computing units, while being able to be trained on the edge. The studies conducted within the scope of the thesis are summarized in the following sections.

II. Optimizing Robustness of BNNs by Margin-Maximization

The classical approach to achieve robustness in NNs is to inject bit flips during training; an approach with significant disadvantages. First, injecting bit flips during training can significantly degrade accuracy. The higher the bit error rate (BER) during training, the more significant the accuracy degradation. Another disadvantage is the additional overhead. For every bit of the error-prone data, a decision has to be made whether to inject a bit flip, which adds numerous additional steps in the NN training. Achieving robustness in NNs without bit flip injection, and thus, conquering the above disadvantages, would be of great benefit for the robustness optimization of NNs.

The simpler structure of BNNs allows better exploration of robustness metrics based on margins. In [1], we provide formal proofs to quantify the maximum number of bit flips that can be tolerated in the weights and activations. With the proposed margin-based metrics and the well-known hinge loss for maximum margin classification in support vector machines (SVMs), we are able to construct a modified hinge loss (MHL) to train BNNs for robustness without any bit flip injections. In our experiments, we assume a general error model which the BNN uses to perform its operations. Our results indicate that the MHL enables BNNs to tolerate higher bit error rates than with bit flip training and, therefore, allows to further lower the requirements on approximate memories and computing units used for BNNs.

III. BNN Using Approximate FeFET Memory

In [2] we focus on BNNs using approximate Ferroelectric FET (FeFET) memory as on-chip memory. FeFET is a promising emerging non-volatile memory (NVM) technology, especially for BNN inference on the low-power edge. The reliability of FeFET memory, however, inherently depends on temperature. Hence, changes in temperature during runtime manifest themselves as changes in bit error rates. We reveal the temperature-dependent bit error model of FeFET memories, and show that BNN accuracy drops to unacceptable levels under the errors. We explore two countermeasures: (1) Training BNNs for bit error tolerance by injecting bit flips, and (2) applying a bit error rate assignment algorithm (BERA) which operates in a layer-wise manner and does not inject bit flips during training. In the experiments, the BNNs effectively tolerate temperature-dependent bit errors for the entire range of operating temperature for both methods.

In [5], we focus on BNNs that employ approximate FeFET-based XNOR gates for logic-in-memory (LiM), where the XNOR gates constitute both the storage for the weights and perform the XNOR operations. We investigate the probability of error in FeFET-based XNOR LiM, demonstrating the trade-off between speed and reliability. Using our reliability model, we show how BNNs can be proactively trained in the presence of XNOR-induced errors towards obtaining robust BNNs at design time. Furthermore, we provide a runtime adaptation technique, that selectively trades off errors and XNOR speed for every BNN layer. Our results demonstrate that when a small loss (e.g., 1%) in inference accuracy is acceptable, our design-time and run-time techniques provide error-resilient BNNs. Specifically, our techniques achieve an XNOR speedup of 75% and 50%, respectively, for FashionMNIST, and 38% and 24% for CIFAR10.

IV. Exploiting the Robustness of BNNs for Efficient Hardware

In [4], we exploit the robustness of BNNs for efficient hardware design. BNNs can be accelerated with analog computing based crossbar accelerators that utilize XNOR gates along with additional interface circuits. Such accelerators demand a large amount of analog-to-digital converters (ADCs) and registers, resulting in expensive designs. To increase the inference efficiency, the state of the art divides the interface circuit into an Analog Path (AP), utilizing cheap analog comparators, and a Digital Path (DP), utilizing expensive ADCs and registers. During BNN execution, a certain path is selectively triggered. Ideally, as inference via AP is more efficient, it should be triggered as often as possible. However, unless the number of weights is very small, the AP is rarely triggered in the state of the art. To overcome this, we propose a novel BNN inference scheme, called Local Thresholding Approximation (LTA). It approximates the global thresholdings in BNNs by local ones. This enables the use of the AP through most of the execution, which significantly increases the interface circuit efficiency. In our evaluations with two BNN architectures, using LTA reduces the area by 42× and 54×, the energy by 2.7× and 4.2×, and the latency by 3.8x and 1.15x, compared to the state-of-the-art crossbar-based BNN accelerators.

In [6], we optimize another analog computing scheme for BNNs, i.e. Integrate-and-Fire (IF) Spiking Neural Networks (SNNs). To achieve high inference accuracy in IF-SNNs, the analog hardware needs to represent current-based MAC levels as spike times, for which a large membrane capacitor is required. This results in high energy use, considerable area cost, and long latency, constituting one of the major bottlenecks in analog IF-SNN implementations. To alleviate this, we propose a HW/SW Codesign method, called CapMin, for capacitor size minimization in analog computing IF-SNNs. CapMin minimizes the capacitor size by reducing the number of spike times needed for accurate operation of the HW, based on the absolute frequency of MAC level occurrences in the SW. To increase the computation's tolerance to process variation, we propose CapMin-V, which trades capacitor size for protection based on the reduced capacitor size found in CapMin. CapMin achieves around 14× reduction in capacitor size over the state of the art, while CapMin-V achieves increased variation tolerance, requiring only a small increase in capacitor size.

V. Resource-Efficient Training of BNNs

Training methods for BNNs use a large number of Floating Point (FP) values during training and employ the information stored in them to obtain a binarized model, thereof suffering from high memory usage. When one of the most memory-efficient training procedures for BNNs, the Binary optimizer (Bop), is used, one momentum value encoded as FP per binary weight is stored. Bop suffers from high memory usage as well, posing a challenge for the design of on-chip BNN training accelerators with limited on-chip memory and energy budgets. The common approach is to encode the momentum values in Bop with standard FP formats for the BNN training. However, these encodings may use an excessive number of bits to encode the momentum values.

In [3], we propose methods to enable the memory-efficient training of BNNs on the edge. We first investigate the impact of arbitrary FP encodings. When the FP format is not properly chosen, we prove that updates of the momentum values can be lost and the quality of training is therefore dropped. With this insights, we formulate a metric to determine the number of unchanged momentum values in a training iteration due to the FP encoding. Based on the metric, we develop an algorithm to find FP encodings that are more memory-efficient than the standard FP encodings. In our experiments, the memory usage in BNN training is decreased by factors of 2.47×, 2.43×, 2.04×, depending on the BNN model, with minimal accuracy cost (smaller than 1%) compared to using 32-bit FP encoding.

VI. Conclusion

This thesis explores a vision for highly resource-efficient future intelligent systems comprised of robust BNNs operating with approximate memory and approximate computing units, while being able to be trained on the edge. Some results are published in peer-reviewed international conferences and journals [1, 3, 2, 5], while others are under review [4, 6].

References

[1] Buschjäger, Sebastian and Chen, Jian-Jia and Chen, Kuan-Hsun and Günzel, Mario and Hakert, Christian and Morik, Katharina and Novkin, Rodion and Pfahler, Lukas and Yayla, Mikail. "Margin-Maximization in Binarized Neural Networks for Optimizing Bit Error Tolerance". In: *DATE*. 2021.

[2] M. Yayla, S. Buschjäger, A. Gupta, J.-J. Chen, J. Henkel, K. Morik, et al. "FeFET-Based Binarized Neural Networks Under Temperature-Dependent Bit Errors". In: *IEEE Transactions on Computers* (2022).

[3] M. Yayla and J.-J. Chen. "Memory-Efficient Training of Binarized Neural Networks on the Edge". In: *DAC*. 2022.

[4] M. Yayla, F. Frustaci, F. Spagnolo, J.-J. Chen, and H. Amrouch. "Global by Local Thresholding in Binarized Neural Networks for Efficient Crossbar Accelerator Design." In: *(single blind peer review)* (2023).

[5] M. Yayla, S. Thomann, S. Buschjäger, K. Morik, J.-J. Chen, and H. Amrouch. "Reliable Binarized Neural Networks on Unreliable Beyond Von-Neumann Architecture". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* (2022).

[6] M. Yayla, S. Thomann, M.-L. Wei, C.-L. Yang, J.-J. Chen, and H. Amrouch. "HW/SW Codesign for Robust and Efficient Binarized SNNs by Capacitor Minimization". In: *(single blind peer review)* (2023).