Rectangular Packing for High-utilization Analog In-memory Compute Mappings

Lawrence Roman A. Quizon

April 2025

Contents

Abstract

The use of Artificial intelligence (AI) offers great benefits in terms of scalability, application space and viability for extreme edge devices. However, extreme edge devices are constrained to work with extremely low amounts of memory and energy. For this purpose, developments ranging from the emergence of more efficient AI algorithms all the way to the design and fabrication of more efficient application-specific ICs have been emerging in the recent years.

Analog in-memory computing shows itself as the most area and energy-efficient solution for AI inference in extreme edge devices. However, typical analog in-memory computing (AIMC) architectures require large amounts of energy for writing the weights to the memory cells. Reusing the written weights as much as possible is imperative to achieve high energy efficiency in AIMC.

Existing works on AIMC acceleration tend to ignore model utilization entirely or target a specific model in an ad-hoc manner. Since AIMC arrays may need to be designed for generality, there is a need to design a utilization mapping and tool flow that can optimize for a set of target models. We take into account a set of models and derive a minimum highest-utilization viable mapping at a specific latency constraint.

Matrices are typically mapped one-to-one to a memory array, causing very little array utilization when layers with small footprint are mapped. We propose a new AIMC architecture that can achieve high utilization by allowing multiple layers to be mapped to the same AIMC array, a restriction that many DNN compiler works self-impose. We also propose a new mapping algorithm that can achieve high utilization by packing multiple models into the same AIMC array. We show that our proposed architecture and mapping algorithm can achieve high utilization while maintaining low latency and energy consumption.

1 Introduction

The use of artificial intelligence (AI) in extreme edge devices such as wireless sensor nodes (WSNs) will greatly benefit the scalability and application space of such nodes. AI can be applied to solve problems with clustering, data routing, and most importantly it can be used to reduce the volume of data transmission via data compression or making conclusions from data within the node itself [1].

However, since devices in the extreme edge are constrained to work with extremely low amounts of memory and energy [2], even the simplest AI models are difficult to execute with typical sequential processors. WSNs have memories in the order of kB and clock speeds in the order of kHz to MHz due to energy constraints, rendering them unable to run state-of-the art AI applications.

A promising hardware approach allowing the use of AI in low-power edge devices is in-memory computing (IMC) [3]. IMC allows very high energy savings compared to other approaches by bypassing the most energy-expensive and time-consuming part of AI processing: memory accesses. Analog IMC (AIMC) with memristors has proven to be fast and efficient at multiply-and-accumulate operations (MACs) which are by far the most common operation used by AI software. Additionally, since memristors are nonvolatile memory devices, they are particularly robust to energy interruptions from ultra-low power situations in WSN.

Several problems bar the use Analog IMC in extreme edge devices. The memristive devices typically used by Analog IMC architectures typically report much higher write energy costs than read energy costs []. This poses the need to amortize the write energy cost over as many MAC operations as possible in the AIMC architecture.

Typical Digital DNN accelerators also seek to amortize the cost of writes in memory by implementing as much data reuse as possible. This is made possible by spatial arrangement of parallel processing elements (PEs) in the architecture. The spatial arrangement of PEs allows for data reuse by allowing the same data to be used by multiple PEs at the same time. DNN compilers figure out the arrangement of data and computational workflow for DNNs in digital accelerators that best take advantage of the spatial arrangement of PEs to maximize data reuse.

2 Background

3 Background

Bibliography

[1] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys and Tutorials*, 2014. DOI: 10.1109/comst.2014.2320099.

[2] D. Ma, G. Lan, M. Hassan, W. Hu, and S. K. Das, "Sensing, computing, and communications for energy harvesting iots: A survey," *IEEE Communications Surveys and Tutorials*, 2019. DOI: 10.1109/comst.2019.2962526.

[3] D. A. Patterson *et al.*, "A case for intelligent ram," *IEEE Micro*, 1997. DOI: 10.1109/40.592312.