MLPAT: A Power, Area, Timing Modeling Framework for Machine Learning Accelerators

[1]Tianqi Tang, [2]Sheng Li, [1]Yuan Xie, [2]Norm Jouppi
[1]{tianqi_tang, yuanxie}@ucsb.edu,
[2]{lsheng, jouppi}@google.com
[1]University of Santa Barbara, [2]Google Inc.

Abstract

This paper introduces MLPAT, an integrated power, area, and timing modeling framework for machine learning accelerators. MLPAT has accurate modeling results, with overall area and power estimation errors below 10% when validated against TPU-v1 [1] published data. MLPAT supports comprehensive design space exploration for architecting machine learning accelerators. Using MLPAT, this paper explores the design space of precision and architecture tradeoffs for autonomous driving accelerators and demonstrates that low-precision ML accelerators can achieve more than 50% savings on chip area and power without accuracy loss.

1. Introduction

Machine learning (ML) accelerators play an important role in the era of artificial intelligence. Many emerging accelerators have been proposed to meet different design targets in a variety of use scenarios, ranging from cloud to edge devices. However, the design space of ML accelerators is very large, covering vastly different performance targets and use scenarios, which drives the need of architects for tools to navigate the design space with fast and accurate power, area, and timing (PAT) estimation in the early design stage. Although there is a surge of frameworks [2][3][4] to address these challenges, limitations exist on these frameworks. Some [3] can only provide energy estimations without chip area and timing. Others [2][4] rely on RTL and EDA tools and thus cannot provide an abstract yet white-boxed layer for agile yet accurate architecture exploration.

To bridge this gap, we introduce MLPAT, an integrated power, area, and timing modeling framework for comprehensive design space exploration of ML accelerators. Inspired by the methodology of CACTI [5] and McPAT [6], MLPAT models the emerging ML accelerators analytically at the architecture level with large design and technology options. The architecture-level analytical modeling methodology enables fast and accurate modeling.

2. MLPAT: Overview and Operations

Fig. 1 gives an overview of MLPAT. It supports modeling major components in ML accelerators, including systolic arrays, on-chip memory, activation pipeline, on-chip interconnect, and beyond. A unique feature for ML accelerators is the high t[1]olerance of low data precision because of the nature of ML algorithms. Therefore, MLPAT supports different precision, from fp32, fp16, bfloat16 [7], to integer, which opens up the door for architecting ML accelerators with precision considerations and tradeoffs.

[1] This work is done when Tianqi works as an intern in Google.

At the microarchitecture level of systolic arrays, MLPAT supports different dataflows, including weight stationary, output stationary, no local reuse, and row stationary. MLPAT models different on-chip interconnects to support not only different dataflows but also different architectures from large unified systolic arrays as in TPU-v1 [1] to tile-based PEs as in Eyeriss [2], DaDianNao [8], and MAERI [9]. At the circuit and technology level, MLPAT maps the architectural components to circuit blocks such as MAC arrays, D-Flip-Flop (DFF) arrays, memory arrays, and interconnect. These circuit blocks are then mapped to fundamental analytical RC ladder/trees and layout models to compute timing, area, and energy at different technology nodes.

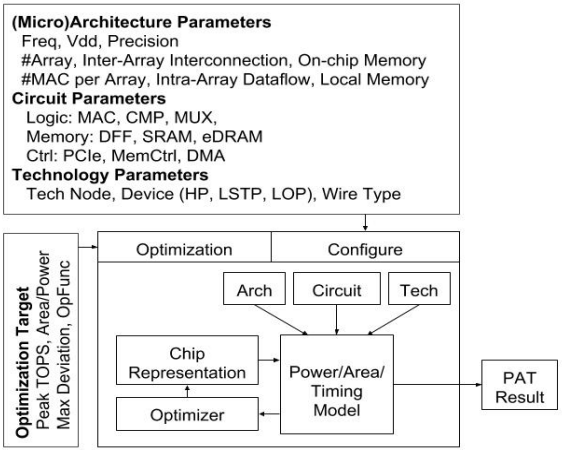

Figure 1. Block diagram of the MLPAT framework

As shown in Figure 1, users can compose ML accelerators by specifying configurations at different levels, design constraints, and optimization targets. MLPAT will generate an internal chip representation to find optimized chip architectures that satisfy the design constraints such as die area, power budget, and target performance (TeraOPs). When combined with the performance analysis/simulation tools, MLPAT can produce dynamic power results of end-to-end ML workloads running on target accelerators.

3. Validation

The primary focus of MLPAT is fast and accurate power and area modeling at the architectural level when timing (and thus the target performance, i.e. TeraOPs) is given as a main design constraint. To ensure accuracy of MLPAT, we conducted rigorous validations against TPU-v1 [1]. Figure 2 shows the validation results of power and area, with a 700MHz target clock frequency.

At the chip level, the modeling results of overall area and power, i.e. thermal design power (TDP) have <5% and <10% error respectively, compared with the published TDP (75W) and area (<331mm^2, i.e. less than half of the Haswell die size) [1]. At component level, TPU-v1 contains four major parts: (1) MAC-based systolic array for matrix multiplication; (2) Unified buffer & Weight FIFO for activation and weights; (3) Accumulator buffer for partial

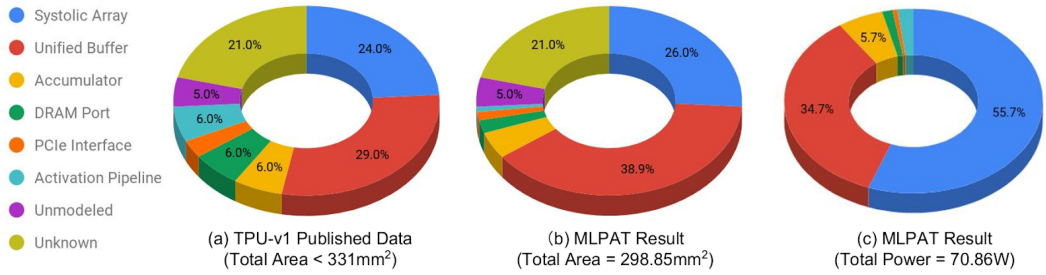

Figure 2. Area and Power Break Down of TPU-v1 Published Data [1] v.s. MLPAT Simulation Results. TPU-v1 @ 700MHz with 0.86V power supply is fabricated at 28nm. Architecture parameters used in the model are: Systolic Array Size: 256x256; Accumulator: 256 int-32 adders; Unified Buffer: 24MB, dual banks, dual ports; Accumulator Buffer: 4MB, 4k blocks per bank, dual ports; PCIe Gen3x16: 14GB/s.

sums; and (4) Activation pipeline for other operations. MLPAT produces accurate area modeling (within 2% relative error) for the systolic array and the accumulator buffer. MLPAT over-estimates the relative area of unified buffer by ~10% in our result, which may be due to the lack of knowledge on optimized placement-and-routing for the interconnect between systolic array and unified buffer in TPU-v1. We also model the peripheral interfaces including DRAM port (6.0% v.s. 2.8%), PCIe interface (3.0% v.s. 1.8%). We currently do not model host interface, controller, and misc I/O, with 5% in total. The unknown components in TPU-v1 occupy ~21% of the chip area, and we assume 20% of the top level area is overhead, consistent with many chip designs. Although there is no published power breakdown, the modeled power breakdown is shown in Figure 2(c). It shows the systolic array is the biggest power consumer, with 56% of the total chip power.

4. Case Study

Autonomous driving is one of, if not, the most important use scenarios for machine learning, especially for deep learning. It demands a daunting amount of computing power (e.g., Nvidia's Jetson Xavier [10] and Pegasus [11] delivers 30 TOPS and 320 TOPS for L3 and L5 automation, respectively). To showcase MLPAT's capability, we focus on autonomous driving inference accelerators. Low precision is an important design option for inference accelerators. Many prior ML researchers have explored the impact on accuracy of using low-precision models from fp16 to integer. However, it is not clear how the low-precision models impact the chip architecture in terms of silicon area and power. In this paper, using MLPAT we explore the relationships and tradeoffs between savings on chip area, power, and accuracy with different low-precision model/chip co-designs, including fp16, bf16, int16, and int8.

As shown in Table I, compared to fp32, the cases of fp16, bf16, and int16 all achieve more than 50% savings on power and silicon area without accuracy loss, or even a slight increase possibly due to noise introduced by the low-precision models. Accuracies are collected from previous works [12][13][14] using ResNet-50 as the model. Due to different fp32 baseline accuracy in different works, we only show the accuracy loss from the claimed fp32 baseline in each individual work. Negative accuracy loss means positive accuracy improvement, and vice versa. Though the accuracy loss of bfloat16 is not available, it is reasonable to assume bfloat16 should be at least as good as int8 due to the same bit length of mantissa. While getting more than 80% hardware benefits, int8 experiences more than 0.1% accuracy loss. The accuracy loss may vary among different neural network models and different input image datasets. Whether such accuracy loss is tolerable depends on its specific use scenario.

Table I. Power & Area Gain,
Inference Accuracy Loss v.s. Precision

Precision	Power Reduction (%)	Area Reduction (%)	Accuracy Loss ResNet-50 (%)
fp16	60.82	55.01	-0.12
bfloat16	66.27	60.96	Unavailable
int16	63.97	62.72	-0.07
int8	85.73	85.54	0.13

* FP32 is used as the baseline to normalize all other precision results. The reduction of chip power and area is obtained from MLPAT. Architecture parameters are similar to TPUv1 but scaled to make it 92TOPS in different precision.

5. Conclusions

MLPAT is the first framework to integrate power, area, and timing modeling for ML accelerators. Unlike prior tools that either only provide partial results or rely on EDA tools, MLPAT simultaneously models power, area, and timing analytically at the architecture level. MLPAT empowers architects with fast yet accurate exploration on the large and diverse design space of accelerators. Validations show reasonable agreement between MLPAT's predictions and published data for a state-of-the-art ML accelerator. With MLPAT, we have explored the tradeoffs between power, area, and accuracy of low precision with fp16, bfloat16, int16, and int8. Future work will model new architecture components and architect other state-of-the-art ML accelerators with MLPAT.

Reference

[1] Jouppi, Norman P., et al. "In-datacenter performance analysis of a tensor processing unit." *Computer Architecture (ISCA), 2017 ACM/IEEE 44th Annual International Symposium on*. IEEE, 2017.

[2] Chen, Yu-Hsin, et al. "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks." *ACM SIGARCH Computer Architecture News*. Vol. 44. No. 3. IEEE Press, 2016.

[3] Kwon, Hyoukjun, et al. "MAESTRO: An Open-source Infrastructure for Modeling Dataflows within Deep Learning Accelerators." *arXiv preprint arXiv:1805.02566* (2018).

[4] http://nvdla.org/

[5] Muralimanohar, Naveen, et al. "CACTI 6.0: A tool to model large caches." *HP laboratories* (2009): 22-31.

[6] Li, Sheng, et al. "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures." *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2009.

[7] https://en.wikipedia.org/wiki/Bfloat16_floating-point_format

[8] Chen, Yunji, et al. "Dadiannao: A machine-learning supercomputer." *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2014.

[9] Kwon, Hyoukjun, et al. "MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Programmable Interconnects." *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 2018.

[10] https://developer.nvidia.com/jetson-xavier-devkit

[11] https://www.nvidia.com/en-us/self-driving-cars/drive-platform/

[12] Micikevicius, Paulius, et al. "Mixed precision training." *arXiv preprint arXiv:1710.03740* (2017).

[13] Das, Dipankar, et al. "Mixed Precision Training of Convolutional Neural Networks using Integer Operations." *arXiv preprint arXiv:1802.00930* (2018).

[14] GTC 2018 talk “8 bit Inference with TensorRT” http://on-demand.gputechconf.com/gtc/2017/presentation/s7310-8-bit-inference-with-tensorrt.pdf