From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks

Other titles in Foundations and Trends® in Electronic Design Automation

Self-Powered Wearable IoT Devices for Health and Activity Monitoring
Ganapati Bhat, Ujjwal Gupta, Yigit Tuncel, Fatih Karabacak, Sule Ozev and Umit Y. Ogras
ISBN: 978-1-68083-748-3

On-Chip Dynamic Resource Management
Antonio Miele, Anil Kanduri, Kasra Moazzemi, Dávid Juhász, Amir R. Rahmani, Nikil Dutt, Pasi Liljeberg and Axel Jantsch
ISBN: 978-1-68083-578-6

Smart Healthcare
Hongxu Yin, Ayten Ozge Akmandor, Arsalan Mosenia and Niraj K. Jha
ISBN: 978-1-68083-440-6

Contracts for System Design
Albert Benveniste, Benoit Caillaud, Dejan Nickovic, Roberto Passerone, Jean-Baptiste Raclet, Philipp Reinkemeier, Alberto Sangiovanni-Vincentelli, Werner Damm, Thomas A. Henzinger and Kim G. Larsen
ISBN: 978-1-68083-402-4

Non-Boolean Computing with Spintronic Devices
Kawsher A. Roxy and Sanjukta Bhanja
ISBN: 978-1-68083-362-1

From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks

Leonardo Rezende Juracy
PUCRS
leonardo.juracy@edu.pucrs.br

Rafael Garibotti
PUCRS
rafael.garibotti@pucrs.br

Fernando Gehm Moraes
PUCRS
fernando.moraes@pucrs.br

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Electronic Design Automation

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

L. R. Juracy *et al.*. *From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks*. Foundations and Trends® in Electronic Design Automation, vol. 13, no. 4, pp. 270–344, 2023.

ISBN: 978-1-63828-163-4

Foundations and Trends® in Electronic Design Automation

Volume 13, Issue 4, 2023

Editorial Board

Editorial Scope

Topics

Foundations and Trends® in Electronic Design Automation publishes survey and tutorial articles in the following topics:

- System Level Design
- Behavioral Synthesis
- Logic Design
- Verification
- Test
- Physical Design
- Circuit Level Design
- Reconfigurable Systems
- Analog Design
- Embedded software and parallel programming
- Multicore, GPU, FPGA, and heterogeneous systems
- Distributed, networked embedded systems
- Real-time and cyberphysical systems

Information for Librarians

Foundations and Trends® in Electronic Design Automation, 2023, Volume 13, 4 issues. ISSN paper version 1551-3939. ISSN online version 1551-3947. Also available as a combined paper and online subscription.

Contents

1 Introduction 3

2 Basic Concepts 6

3 CNN and DNN Hardware Accelerators 11
- 3.1 Taxonomy for Hardware Accelerators 11
- 3.2 Academic Accelerators 14
- 3.3 Industrial Accelerators 34

4 Hardware Design Space Exploration Frameworks and Simulators 39
- 4.1 Hardware Design Space Exploration Frameworks 39
- 4.2 Hardware Simulators 49
- 4.3 Final Remarks Related to DSE Frameworks and Simulators 56

5 Conclusion 60

Acknowledgements 62

References 63

From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks

Leonardo Rezende Juracy, Rafael Garibotti and Fernando Gehm Moraes

School of Technology, Pontifical Catholic University of Rio Grande do Sul – PUCRS, Brazil; leonardo.juracy@edu.pucrs.br; rafael.garibotti@pucrs.br; fernando.moraes@pucrs.br

ABSTRACT

Over the past decade, a massive proliferation of machine learning algorithms has emerged, from applications for surveillance to self-driving cars. The turning point occurred with the arrival of Convolutional Neural Network (CNN) models and the incredible accuracy brought by Deep Neural Networks (DNNs) at the cost of high computational complexity. In this growing environment, graphic processing units (GPUs) have become the de facto reference platform for the training and inference phases of CNNs and DNNs due to their high processing parallelism and memory bandwidth. However, GPUs are power-hungry architectures. To enable the deployment of CNN and DNN applications on energy-constrained devices (e.g., IoT devices), industry and academic research have moved towards hardware accelerators. Following the evolution of neural networks (from CNNs

Leonardo Rezende Juracy, Rafael Garibotti and Fernando Gehm Moraes (2023), "From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks", Foundations and Trends® in Electronic Design Automation: Vol. 13, No. 4, pp 270–344. DOI: 10.1561/1000000060.

to DNNs), this survey sheds light on the impact of this architectural shift and discusses hardware accelerator trends in terms of design, exploration, simulation, and frameworks developed in both academia and industry.

1

Introduction

The past decade has witnessed the consolidation of Artificial Intelligence (AI) technology, thanks to the popularization of Machine Learning (ML) models. The technological boom of ML models started in 2012 when the world was stunned by the record-breaking classification performance achieved by combining an ML model with a high computational performance graphic processing unit (GPU) (Krizhevsky *et al.*, 2012). Since then, ML models received ever-increasing attention, being applied in different areas such as computational vision (Technologies, 2022), virtual reality (Facebook, 2022a), voice assistants (Google, 2022b), chatbots (ServiceNow, 2022), and self-driving vehicles (Tesla, 2022).

The most popular ML models are brain-inspired models such as Neural Networks (NNs), including Convolutional Neural Networks (CNNs) and, more recently, Deep Neural Networks (DNNs). They are characterized by resembling the human brain, performing data processing by mimicking synapses using thousands of interconnected neurons in a network. Synapses are composed of a data input sample plus a weight that works similarly to a filter (Goodfellow *et al.*, 2016). Then a mathematical operation is applied to the incoming synapses of a neuron (i.e., a convolution) and serves as input to an activation function. The output is then used at the synapses of the subsequent neurons (Haykin, 2009).

The popularity of CNN models was due to their accuracy in image recognition and classification (Karpathy *et al.*, 2014). CNN models have sparse connections, where all neurons of one layer are connected to all neurons of the next layer. It brings many computational benefits w.r.t. previous approaches (Arel *et al.*, 2010), such as less memory storage for synapse weights and greater reuse of weights read from memory (Goodfellow *et al.*, 2016). More recently, DNN models have emerged by outperforming conventional machine learning algorithms across a wide range of applications, e.g., image recognition, object detection, robotics, and natural language processing (Yu *et al.*, 2021). To summarize, CNN is specialized for handling grid-like data such as images and videos, and DNN is a general architecture that can handle different data formats.

Due to the success and increasing use of CNNs and DNNs everywhere, several frameworks are emerging, helping developers to build their ML models by offering the necessary mechanisms for both the training and inference phases. Examples of frameworks include Caffe (2022), PyTorch (2022), and TensorFlow (2022). These frameworks use a high-level approach to abstract the implementation of functions, such as convolution, and help in the development of ML applications. Furthermore, these frameworks abstract the training phase by providing backpropagation algorithms.

ML frameworks often rely on GPUs due to their parallelization capabilities (Chen *et al.*, 2016b; Strom, 2015), making it the de facto reference platform for the training and inference phases of CNN and DNN models. The underlying reason for the GPU wave is that the conventional central processing units (CPUs) cannot satisfy the dramatically increasing requirements for memory bandwidth and computational complexity caused by the ever-increasing model size of DNNs (Deng *et al.*, 2020). As there are not too many restrictions in the training phase of ML models, the GPU should continue to be the reference platform. However, in the inference phase, GPUs cannot be applied to all domains due to their high energy consumption, such as the Internet of Things (IoT) and wearable devices. Thus, academia and industry have been looking for power-efficient architectures (Garibotti *et al.*, 2019), making the adoption of specialized hardware a becoming trend in the inference phase of ML applications (Hsiao *et al.*, 2020; Chen *et al.*, 2019; Shivapakash *et al.*, 2020).

Hardware accelerators focus on reducing energy and power cost as well as improving data throughput (Chen *et al.*, 2016b; Andri *et al.*, 2017; Shivapakash *et al.*, 2020). These advantages make CNN and DNN hardware accelerators a suitable replacement for CPUs and GPUs for the inference phase (Dally *et al.*, 2020). With this growing interest in hardware accelerators, several works have been delving into the architectural characteristics of CNNs and DNNs to properly model their components, such as input buffers, MAC array, activation function, and output control logic (Lu *et al.*, 2017; Bai *et al.*, 2020; Chen *et al.*, 2020). In addition to these typical components, the literature presents hardware accelerators that use different approaches to access data, the most common being input stationary (IS), output stationary (OS), and weight stationary (WS) (Udupa *et al.*, 2020; Das *et al.*, 2020; Ryu *et al.*, 2022).

The weakness of this new hot topic area is the lack of information comparing hardware accelerators found in the literature. Several works on hardware accelerators use different implementations, but little or none explore the trade-offs and trends between them. One exception is Eyeriss, which proposes a comparison between different accelerators, but lacks performance assessment or area trade-offs (Chen *et al.*, 2016b). Another rare example is the work presented by Das *et al.* (2020) that compares hardware accelerators. However, the Authors consider different technology nodes, which results in an unfair analysis. In this regard, this survey aims to fill this gap by highlighting the current literature on CNN and DNN hardware accelerators and discussing hardware accelerator trends in terms of design, exploration, simulation, and frameworks developed in academia and industry.

The rest of this monograph is organized as follows. Section 2 presents concepts related to CNN and DNN models to help general readers understand the topics discussed in the following sections. Section 3 presents the state-of-the-art related to academic and industrial hardware accelerators based on CNN and DNN models. Furthermore, a taxonomy for academic hardware accelerators is proposed to classify the reviewed works. Next, Section 4 surveys the state-of-the-art on hardware simulators and DSE frameworks, bringing and discussing the advantages and disadvantages of each approach. Finally, Section 5 concludes this survey, pointing out directions for future research.

References

Agostini, N. B., S. Curzel, J. J. Zhang, A. Limaye, C. Tan, V. Amatya, M. Minutoli, V. G. Castellana, J. Manzano, D. Brooks, G.-Y. Wei, and A. Tumeo. (2022). “Bridging Python to Silicon: The SODA Toolchain”. *IEEE Micro*. 42(5): 78–88.

Ahmad, A. and M. A. Pasha. (2020). “FFConv: an FPGA-based accelerator for fast convolution layers in convolutional neural networks”. *ACM Transactions on Embedded Computing Systems*. 19(2): 1–24.

Ajayi, T., V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny, A. B. Kahng, M. Kim, J. Lee, U. Mallappa, M. Neseem, G. Pradipta, S. Reda, M. Saligane, S. S. Sapatnekar, C. Sechen, M. Shalan, W. Swartz, L. Wang, Z. Wang, M. Woo, and B. Xu. (2019). “Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project”. In: *ACM/IEEE Design Automation Conference (DAC)*. 441–444.

Alibaba. (2019). “Alibaba Hanguang 800”. URL: https://techcrunch.com/2019/09/24/alibaba-unveils-hanguang-800-an-ai-inference-chip-it-says-significantly-increases-the-speed-of-machine-learning-tasks/.

Alom, M. Z., T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari. (2018). "The history began from alexnet: A comprehensive survey on deep learning approaches". *Computing Research Repository*. abs/1803.01164(1): 1–39.

Amazon. (2018). "AWS Inferentia". URL: https://aws.amazon.com/about-aws/whats-new/2018/11/announcing-amazon-inferentia-machine-learning-inference-microchip/.

Andri, R., L. Cavigelli, D. Rossi, and L. Benini. (2017). "YodaNN: An architecture for ultralow power binary-weight CNN acceleration". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 37(1): 48–60.

Apple. (2022). "iPhone 11". URL: https://www.apple.com/iphone-11/.

Arel, I., D. C. Rose, and T. P. Karnowski. (2010). "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]". *IEEE Computational Intelligence Magazine*. 5(4): 13–18.

Asanovic, K., R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, *et al.* (2016). "The rocket chip generator". *Tech. rep.* University of California. 11p. URL: https://aspire.eecs.berkeley.edu/wp/wp-content/uploads/2016/04/Tech-Report-The-Rocket-Chip-Generator-Beamer.pdf.

Bai, L., Y. Lyu, and X. Huang. (2020). "A unified hardware architecture for convolutions and deconvolutions in CNN". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5.

Baskin, C., N. Liss, E. Schwartz, E. Zheltonozhskii, R. Giryes, A. M. Bronstein, and A. Mendelson. (2021). "Uniq: Uniform noise injection for non-uniform quantization of neural networks". *ACM Transactions on Computer Systems*. 37(1-4): 1–15.

Caffe. (2022). "Caffe". URL: https://caffe.berkeleyvision.org/.

Cao, S., W. Deng, Z. Bao, C. Xue, S. Xu, and S. Zhang. (2020). "SimuNN: A Pre-RTL Inference, Simulation and Evaluation Framework for Neural Networks". *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. 10(2): 217–230.

Cerebras. (2022). "Cerebras CS-1". URL: https://www.cerebras.net/technology/.

Chang, J., Y. Choi, T. Lee, and J. Cho. (2018). "Reducing MAC Operation in Convolutional Neural Network with Sign Prediction". In: *International Conference on Information and Communication Technology Convergence (ICTC)*. 177–182.

Chen, Y.-H., J. Emer, and V. Sze. (2016a). "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks". *ACM SIGARCH Computer Architecture News*. 44(3): 367–379.

Chen, Y.-H., T. Krishna, J. S. Emer, and V. Sze. (2016b). "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks". *IEEE Journal of Solid-state Circuits*. 52(1): 127–138.

Chen, Y.-H., T.-J. Yang, J. Emer, and V. Sze. (2019). "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices". *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. 9(2): 292–308.

Chen, T., Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam. (2014). "Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning". *ACM SIGARCH Computer Architecture News*. 42(1): 269–284.

Chen, X., Y. Han, and Y. Wang. (2020). "Communication Lower Bound in Convolution Accelerators". In: *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 529–541.

Courbariaux, M., I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio. (2016). "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1". *Computing Research Repository*. abs/1602.02830(1): 1–11.

Dally, W. J., Y. Turakhia, and S. Han. (2020). "Domain-specific hardware accelerators". *Communications of the ACM*. 63(7): 48–57.

Das, S., A. Roy, K. K. Chandrasekharan, A. Deshwal, and S. Lee. (2020). "A Systolic Dataflow Based Accelerator for CNNs". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5.

Deng, L., G. Li, S. Han, L. Shi, and Y. Xie. (2020). "Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey". *Proceedings of the IEEE*. 108(4): 485–532.

Digital, W. (2022). "Western Digital Machine Learning Accelerator". URL: https://link.westerndigital.com/welcome/mcs-bulletin/mcs-bulletin-events/machine-learning-accelerator.html?_ga=2.249821190.143199995.1570669759-1671111829.1570669759.

Domingues, A. R. P. (2020). "ORCA: A Self-Adaptive, Multiprocessor System-On-Chip Platform". *MA thesis*. PUCRS. 112p.

Du, L., Y. Du, Y. Li, J. Su, Y.-C. Kuan, C.-C. Liu, and M.-C. F. Chang. (2017). "A reconfigurable streaming deep convolutional neural network accelerator for Internet of Things". *IEEE Transactions on Circuits and Systems I: Regular Papers*. 65(1): 198–208.

Du, Z., R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam. (2015). "ShiDianNao: Shifting vision processing closer to the sensor". In: *ACM International Symposium on Computer Architecture (ISCA)*. 92–104.

Ernst, D., S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner. (2004). "Razor: circuit-level correction of timing errors for low-power operation". *IEEE Micro*. 24(6): 10–20.

Facebook. (2022a). "Facebook Horizon". URL: https://www.oculus.com/horizon-worlds/.

Facebook. (2022b). "Facebook Kings Canyon". URL: https://engineering.fb.com/data-center-engineering/accelerating-infrastructure/.

Ferianc, M., H. Fan, D. Manocha, H. Zhou, S. Liu, X. Niu, and W. Luk. (2021). "Improving Performance Estimation for Design Space Exploration for Convolutional Neural Network Accelerators". *MDPI Electronics*. 10(4): 1–14.

Ferretti, L., A. Cini, G. Zacharopoulos, C. Alippi, and L. Pozzi. (2022). "Graph Neural Networks for High-Level Synthesis Design Space Exploration". *ACM Transactions on Design Automation of Electronic Systems. preprint*: 1–19.

Foundation, L. (2022). "The LLVM Compiler Infrastructure". URL: https://llvm.org/.

Fujitsu. (2018). "Fujitsu Deep Learning Unit". URL: https://www.fujitsu.com/global/Images/deep-learning-unit.pdf.

Garibotti, R., L. Ost, A. Butko, R. Reis, A. Gamatié, and G. Sassatelli. (2019). "Exploiting memory allocations in clusterised many-core architectures". *IET Computing & Digital Techniques*. 13(4): 302–311.

Garibotti, R., B. Reagen, Y. S. Shao, G.-Y. Wei, and D. Brooks. (2017). "Using Dynamic Dependence Analysis to Improve the Quality of High-Level Synthesis Designs". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 1838–1841.

Garibotti, R., B. Reagen, Y. S. Shao, G.-Y. Wei, and D. Brooks. (2018). "Assisting High-Level Synthesis Improve SpMV Benchmark Through Dynamic Dependence Analysis". *IEEE Transactions on Circuits and Systems II: Express Briefs*. 65(10): 1440–1444.

Genc, H., S. Kim, A. Amid, A. Haj-Ali, V. Iyer, P. Prakash, J. Zhao, D. Grubb, H. Liew, H. Mao, *et al.* (2021). "Gemmini: Enabling systematic deep-learning architecture evaluation via full-stack integration". In: *ACM/IEEE Design Automation Conference (DAC)*. 769–774.

Gerogiannis, G., M. Birbas, A. Leftheriotis, E. Mylonas, N. Tzanis, and A. Birbas. (2022). "Deep Reinforcement Learning Acceleration for Real-Time Edge Computing Mixed Integer Programming Problems". *IEEE Access*. 10(1): 18526–18543.

Giri, D., K. Chiu, G. D. Guglielmo, P. Mantovani, and L. P. Carloni. (2020). "ESP4ML: Platform-Based Design of Systems-on-Chip for Embedded Machine Learning". In: *IEEE Design, Automation & Test in Europe Conference (DATE)*. 1049–1054.

Gokhale, V., J. Jin, A. Dundar, B. Martini, and E. Culurciello. (2014). "A 240 g-ops/s mobile coprocessor for deep neural networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 682–687.

Goodfellow, I., Y. Bengio, and A. Courville. (2016). *Deep Learning*. MIT Press. 781p.

Google. (2022a). "Cloud TPU". URL: https://cloud.google.com/tpu/.

Google. (2022b). "Google Assistant, your own personal Google". URL: https://assistant.google.com.

Hao, C., X. Zhang, Y. Li, S. Huang, J. Xiong, K. Rupnow, W.-m. Hwu, and D. Chen. (2019). "FPGA/DNN Co-Design: An Efficient Design Methodology for IoT Intelligence on the Edge". In: *ACM/IEEE Design Automation Conference (DAC)*. 1–6.

Haykin, S. S. (2009). *Neural networks and learning machines*. Third. Pearson Education. 906p.

Heidorn, C., F. Hannig, and J. Teich. (2020). "Design space exploration for layer-parallel execution of convolutional neural networks on CGRAs". In: *ACM SIGBED/EDAA Software and Compilers for Embedded Systems (SCOPES)*. 26–31.

Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. (2017). "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision applications". *Computing Research Repository*. abs/1704.04861: 1–9.

Hsiao, S.-F. and H.-J. Chang. (2020). "Sparsity-Aware Deep Learning Accelerator Design Supporting CNN and LSTM Operations". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–4.

Hsiao, S.-F., K.-C. Chen, C.-C. Lin, H.-J. Chang, and B.-C. Tsai. (2020). "Design of a Sparsity-Aware Reconfigurable Deep Learning Accelerator Supporting Various Types of Operations". *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. 10(3): 376–387.

Huang, B., Y. Huan, H. Chu, J. Xu, L. Liu, L. Zheng, and Z. Zou. (2021). "IECA: An In-Execution Configuration CNN Accelerator With 30.55 GOPS/mm^2 Area Efficiency". *IEEE Transactions on Circuits and Systems I: Regular Papers*. 68(11): 4672–4685.

Huawei. (2019). "Huawei Ascend 910". URL: https://www.huawei.com/en/press-events/news/2019/8/huawei-ascend-910-most-powerful-ai-processor.

IBM. (2022). "IBM Watson". URL: https://www.ibm.com/products/deep-learning-platform?mhsrc=ibmsearch_a&mhq=deep-learning-platform.

Inci, A., S. G. Virupaksha, A. Jain, V. V. Thallam, R. Ding, and D. Marculescu. (2022a). "QADAM: Quantization-Aware DNN Accelerator Modeling for Pareto-Optimality". *Computing Research Repository*. abs/2205.13045(1): 1–6.

Inci, A., S. G. Virupaksha, A. Jain, V. V. Thallam, R. Ding, and D. Marculescu. (2022b). "QAPPA: Quantization-Aware Power, Performance, and Area Modeling of DNN Accelerators". *Computing Research Repository*. abs/2205.08648(1): 1–5.

Intel. (2022). "Intel Nervana". URL: https://www.intel.com.br/content/www/br/pt/analytics/artificial-intelligence/overview.html.

Al-Jawfi, R. (2009). "Handwriting Arabic character recognition LeNet using neural network". *International Arab Journal of Information Technology*. 6(3): 304–309.

Jiao, Y., L. Han, R. Jin, Y.-J. Su, C. Ho, L. Yin, Y. Li, L. Chen, Z. Chen, L. Liu, *et al.* (2020). "7.2 A 12nm Programmable Convolution-Efficient Neural-Processing-Unit Chip Achieving 825TOPS". In: *IEEE International Solid-State Circuits Conference (ISSCC)*. 136–140.

Jouppi, N. P., A. B. Kahng, N. Muralimanohar, and V. Srinivas. (2014). "Cacti-IO: Cacti with off-chip power-area-timing models". *IEEE Transactions on Very Large Scale Integration Systems*. 23(7): 1254–1267.

Juracy, L. R., M. T. Moreira, A. M. Amory, and F. G. Moraes. (2021a). "A TensorFlow and System Simulator Integration Approach to Estimate Hardware Metrics of Convolution Accelerators". In: *IEEE Latin America Symposium on Circuits and System (LASCAS)*. 217–230.

Juracy, L. R., A. de Morais Amory, and F. G. Moraes. (2022). "A Fast, Accurate, and Comprehensive PPA Estimation of Convolutional Hardware Accelerators". *IEEE Transactions on Circuits and Systems I: Regular Papers. preprint*: 1–14.

Juracy, L. R., M. T. Moreira, A. de Morais Amory, A. F. Hampel, and F. G. Moraes. (2021b). "A High-Level Modeling Framework for Estimating Hardware Metrics of CNN Accelerators". *IEEE Transactions on Circuits and Systems I: Regular Papers*. 68(11): 4783–4795.

Karbachevsky, A., C. Baskin, E. Zheltonozhskii, Y. Yermolin, F. Gabbay, A. M. Bronstein, and A. Mendelson. (2021). "Early-stage neural network hardware performance analysis". *MDPI Sustainability*. 13(2): 1–20.

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. (2014). "Large-Scale Video Classification with Convolutional Neural Networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1725–1732.

Keras. (2022). "Layer activation functions". URL: https://keras.io/api/layers/activations/.

Kim, S., J. Wang, Y. Seo, S. Lee, Y. Park, S. Park, and C. S. Park. (2020). "Transaction-level Model Simulator for Communication-Limited Accelerators". *Computing Research Repository*. abs/2007.14897(1): 1–11.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems (NIPS)*. 1097–1105.

Kwon, H., P. Chatarasi, M. Pellauer, A. Parashar, V. Sarkar, and T. Krishna. (2019). "Understanding Reuse, Performance, and Hardware Cost of DNN Dataflow: A Data-Centric Approach". In: *IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 754–768.

Kwon, H., M. Pellauer, and T. Krishna. (2018a). "Maestro: An open-source infrastructure for modeling dataflows within deep learning accelerators". *Computing Research Repository*. abs/1805.02566(1): 1–5.

Kwon, H., A. Samajdar, and T. Krishna. (2018b). "Maeri: Enabling flexible dataflow mapping over dnn accelerators via reconfigurable interconnects". *ACM Special Interest Group on Programming Languages Notices*. 53(2): 461–475.

Lin, W. and T. Arslan. (2021). "A Column Streaming-Based Convolution Engine and Mapping Algorithm for CNN-based Edge AI accelerators". In: *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 1–6.

Liu, B., X. Chen, Y. Han, Y. Wang, J. Li, H. Xu, and X. Li. (2020a). "Search-free Accelerator for Sparse Convolutional Neural Networks". In: *ACM/IEEE Asia and South Pacific Design Automation Conference (ASPDAC)*. 524–529.

Liu, B., X. Chen, Y. Han, and H. Xu. (2020b). "Swallow: A Versatile Accelerator for Sparse Neural Networks". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 39(12): 4881–4893.

Lu, W., G. Yan, J. Li, S. Gong, Y. Han, and X. Li. (2017). "Flexflow: A flexible dataflow accelerator architecture for convolutional neural networks". In: *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 553–564.

Manasi, S. D. and S. S. Sapatnekar. (2021). "DeepOpt: Optimized scheduling of CNN workloads for ASIC-based systolic deep learning accelerators". In: *ACM/IEEE Asia and South Pacific Design Automation Conference (ASPDAC)*. 235–241.

Mediatek. (2022). "Mediatek APU". URL: https://www.mediatek.com/technology/artificial-intelligence.

Microsoft. (2022). "Project Brainwave". URL: https://www.microsoft.com/en-us/research/project/project-brainwave/.

Minutoli, M., V. G. Castellana, C. Tan, J. Manzano, V. Amatya, A. Tumeo, D. Brooks, and G.-Y. Wei. (2020). "SODA: a New Synthesis Infrastructure for Agile Hardware Design of Machine Learning Accelerators". In: *IEEE International Conference on Computer-Aided Design (ICCAD)*. 786–792.

Moolchandani, D., A. Kumar, and S. R. Sarangi. (2021). "Accelerating CNN inference on ASICs: A survey". *Journal of Systems Architecture*. 113(1): 1–26.

Munoz-Martinez, F., J. L. Abellan, M. E. Acacio, and T. Krishna. (2020). "STONNE: A Detailed Architectural Simulator for Flexible Neural Network Accelerators". *Computing Research Repository*. abs/2006.07137(1): 1–8.

NVIDIA. (2022a). "NVDLA". URL: http://nvdla.org/.

NVIDIA. (2022b). "TensorRT". URL: https://developer.nvidia.com/tensorrt.

NXP. (2022). “NXP S32V234 MPU”. URL: https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/s32v2-vision-mpus-/vision-processor-for-front-and-surround-view-camera-machine-learning-and-sensor-fusion:S32V234.

Parashar, A., P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer. (2019). “Timeloop: A Systematic Approach to DNN Accelerator Evaluation”. In: *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. 304–315.

Park, S.-S. and K.-S. Chung. (2020). “CENNA: Cost-Effective Neural Network Accelerator”. *Electronics*. 9(1): 1–19.

PyTorch. (2022). “PyTorch”. URL: https://pytorch.org/.

Qualcomm. (2019). “Qualcomm Snapdragon”. URL: https://developer.qualcomm.com/blog/accelerate-your-device-ai-qualcomm-artificial-intelligence-ai-engine-snapdragon.

Reagen, B., P. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernández-Lobato, G.-Y. Wei, and D. Brooks. (2016). “Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators”. In: *ACM International Symposium on Computer Architecture (ISCA)*. 267–278.

Renesas. (2022). “Renesas e-AI”. URL: https://www.renesas.com/jp/en/solutions/key-technology/e-ai.html.

Russo, E., M. Palesi, S. Monteleone, D. Patti, G. Ascia, and V. Catania. (2022). “MEDEA: a multi-objective evolutionary approach to DNN hardware mapping”. In: *IEEE Design, Automation & Test in Europe Conference (DATE)*. 226–231.

Ryu, S., H. Kim, W. Yi, E. Kim, Y. Kim, T. Kim, and J.-J. Kim. (2022). “BitBlade: Energy-Efficient Variable Bit-Precision Hardware Accelerator for Quantized Neural Networks”. *IEEE Journal of Solid-State Circuits*. 1(1): 1–11.

Samajdar, A., J. M. Joseph, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna. (2020). “A Systematic Methodology for Characterizing Scalability of DNN Accelerators using SCALE-Sim”. In: *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. 58–68.

Samajdar, A., Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna. (2018). “SCALE-sim: Systolic CNN accelerator”. *Computing Research Repository*. abs/1811.02883(1): 1–11.

Samsung. (2019). “Samsung Exynos”. URL: https://www.eetimes.com/document.asp?doc_id=1334340.

ServiceNow. (2022). “Enterprise Chatbot – Virtual Agent”. URL: https://assistant.google.com.

Shao, Y. S., J. Clemons, R. Venkatesan, B. Zimmer, M. Fojtik, N. Jiang, B. Keller, A. Klinefelter, N. Pinckney, P. Raina, *et al.* (2019). “Simba: Scaling deep-learning inference with multi-chip-module-based architecture”. In: *IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 14–27.

Shao, Y. S., B. Reagen, G.-Y. Wei, and D. Brooks. (2014). “Aladdin: A pre-RTL, power-performance accelerator simulator enabling large design space exploration of customized architectures”. In: *ACM International Symposium on Computer Architecture (ISCA)*. 97–108.

Shao, Y. S., S. L. Xi, V. Srinivasan, G.-Y. Wei, and D. Brooks. (2016). “Co-designing accelerators and soc interfaces using GEM5-Aladdin”. In: *IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 1–12.

Shivapakash, S., H. Jain, O. Hellwich, and F. Gerfers. (2020). “A Power Efficient Multi-Bit Accelerator for Memory Prohibitive Deep Neural Networks”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5.

Sohrabizadeh, A., Y. Bai, Y. Sun, and J. Cong. (2021). “Enabling Automated FPGA Accelerator Optimization Using Graph Neural Networks”. *Computing Research Repository*. abs/2111.08848(1): 1–12.

Spagnolo, F., S. Perri, F. Frustaci, and P. Corsonello. (2020). “Reconfigurable Convolution Architecture for Heterogeneous Systems-on-Chip”. In: *IEEE Mediterranean Conference on Embedded Computing (MECO)*. 1–5.

Strom, N. (2015). “Scalable distributed DNN training using commodity GPU cloud computing”. In: *International Speech Communication Association (ISCA)*. 1488–1492.

Tang, T. and Y. Xie. (2018). "Mlpat: A power area timing modeling framework for machine learning accelerators". In: *IEEE International Workshop on Domain Specific System Architecture (DOSSA)*. 1–3.

Tavakoli, M. R., S. M. Sayedi, and M. J. Khaleghi. (2020). "A High Throughput Hardware CNN Accelerator Using a Novel Multi-Layer Convolution Processor". In: *IEEE Iranian Conference on Electrical Engineering (ICEE)*. 1–6.

Technologies, D. (2022). "Virtualized Computer Vision for Smart Transportation". URL: https://infohub.delltechnologies.com/t/design-guide-virtualized-computer-vision-for-smart-transportation-with-genetec-1/.

TensorFlow. (2022). "TensorFlow". URL: https://www.tensorflow.org/.

Tesla. (2019). "Autopilot and Full Self-Driving Capability". URL: https://analyticsindiamag.com/under-the-hood-of-teslas-ai-chip-that-takes-the-driverless-battle-to-nvidias-home-turf/.

Tesla. (2022). "Autopilot". URL: https://www.tesla.com.

Texas. (2022). "Texas Instruments Sitara". URL: http://www.ti.com/tool/SITARA-MACHINE-LEARNING.

Toshiba. (2019). "Toshiba Visconti 5". URL: https://toshiba.semicon-storage.com/ap-en/company/news/news-topics/2019/01/automotive-20190107-1.html.

Udupa, P., G. Mahale, K. K. Chandrasekharan, and S. Lee. (2020). "Accelerating Depthwise Convolution and Pooling Operations on z-First Storage CNN Architectures". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5.

Venkatesan, R. *et al.* (2019). "MAGNet: A Modular Accelerator Generator for Neural Networks". In: *IEEE International Conference on Computer-Aided Design (ICCAD)*. 1–8.

Wu, Y. N., J. S. Emer, and V. Sze. (2019). "Accelergy: An architecture-level energy estimation methodology for accelerator designs". In: *IEEE International Conference on Computer-Aided Design (ICCAD)*. 1–8.

Xian, Z., H. Li, and Y. Li. (2020). "Weight Isolation-Based Binarized Neural Networks Accelerator". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–4.

Xiang, T., Y. Feng, X. Ye, X. Tan, W. Li, Y. Zhu, M. Wu, H. Zhang, and D. Fan. (2018). "Accelerating CNN algorithm with fine-grained dataflow architectures". In: *IEEE International Conference on Smart City (SmartCity)*. 243–251.

Xilinx. (2018). "Xilinx xDNN". URL: https://www.xilinx.com/support/documentation/white_papers/wp504-accel-dnns.pdf.

Xilinx. (2021). "Vitis AI". URL: https://www.xilinx.com/products/design-tools/vitis/vitis-ai.html.

Yang, X., M. Gao, Q. Liu, J. Setter, J. Pu, A. Nayak, S. Bell, K. Cao, H. Ha, P. Raina, *et al.* (2020). "Interstellar: Using halide's scheduling language to analyze dnn accelerators". In: *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. 369–383.

Ye, H., C. Hao, H. Jeong, J. Huang, and D. Chen. (2021). "ScaleHLS: Achieving Scalable High-Level Synthesis through MLIR". *Computing Research Repository*. abs/2107.11673(1): 1–15.

Yu, Y., Y. Li, S. Che, N. K. Jha, and W. Zhang. (2021). "Software-Defined Design Space Exploration for an Efficient DNN Accelerator Architecture". *IEEE Transactions on Computers*. 70(1): 45–56.

Zacharopoulos, G., A. Ejjeh, Y. Jing, E.-Y. Yang, T. Jia, I. Brumar, J. Intan, M. Huzaifa, S. Adve, V. Adve, *et al.* (2022). "Trireme: Exploring Hierarchical Multi-Level Parallelism for Domain Specific Hardware Acceleration". *Computing Research Repository*. abs/2201.08603(1): 1–20.

Zhang, C., P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong. (2015). "Optimizing fpga-based accelerator design for deep convolutional neural networks". In: *ACM/SIGDA International Symposium On Field-Programmable Gate Arrays (FPGA)*. 161–170.

Zhang, X., H. Ye, and D. Chen. (2021). "Being-ahead: Benchmarking and Exploring Accelerators for Hardware-Efficient AI Deployment". *Computing Research Repository*. abs/2104.02251(1): 1–12.

Zhao, Y., C. Li, Y. Wang, P. Xu, Y. Zhang, and Y. Lin. (2020). "DNN-Chip Predictor: An Analytical Performance Predictor for DNN Accelerators with Various Dataflows and Hardware Architectures". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1593–1597.