Guest Editors' Introduction to the Special Issue on Machine Learning Architectures and Accelerators

Xuehai Qian, Yanzhi Wang, and Avinash Karanth

Deep learning or deep neural networks (DNNs), as one of the most powerful machine learning techniques, has achieved extraordinary performance in computer vision and surveillance, speech recognition and natural language processing, healthcare and disease diagnosis, etc. Various forms of DNNs have been proposed, including Convolutional Neural Networks, Recurrent Neural Networks, Deep Reinforcement Learning, Transformer model, etc. Deep learning exhibits an offline training phase to derive the weight parameters from an excessive training dataset, as well as an online inference phase to perform classification/prediction/perception/control tasks based on the trained model. Recently, the online learning (e.g., federated learning, transfer learning) capabilities of DNNs are also being investigated such that DNNs can adapt to new situations encountered during actual system operation and time-varying scenarios.

Deep learning models are both computation and storage-intensive since it is necessary to extract high-level features for optimization. This poses significant challenges during both the inference and training phases of the application. The inference is expected to be deployed onto mobile platforms, embedded and IoT devices with restricted power and form factor budget, but with real-time performance requirement. The training phase of the application is highly computation and data-intensive, and thus software/algorithm optimization as well as hardware acceleration re critically required.

Prior research has investigated software/algorithm optimization which includes DNN model architecture search for computation/storage reduction (e.g., depthwise-separate convolutions), model compression (weight pruning, quantization, matrix transformation, etc.), compiler-assisted optimizations, parallel computing techniques such as data parallelism and model parallelism, distributed training algorithms, federated learning, to name a few. Some of the prior research investigating hardware acceleration includes CPU/GPU-based accelerations, FPGAs, dedicated ASIC architectures, to more recently emerging in-memory computing techniques. Computer architecture research plays a key role here, both in terms of the general-purpose architectures that accommodate a wide range of hardware platforms, DNN types, and applications, as well as the specialized architectures targeting the most advanced DNN structure and specialized, critical applications.

This Special Issue of IEEE Transactions on Computers aims to find a convergence of software and hardware/architecture. It aims at DNN algorithms, parallel computing, and compiler code generation techniques that are hardware/architecture friendly, as well as computer architectures that are universal and consistently highly performant on a wide range of DNN algorithms and applications. In this co-design and co-optimization framework we can mitigate the limitation of investigating in only a single direction, shedding some light on the future of embedded, ubiquitous artificial intelligence.

Following an open call for papers, we received 56 submissions from authors in 20 different countries and on a broad range of hardware/software aspects related to deep learning/artificial intelligence acceleration. After a preliminary screening, each submission has been assigned to at least three reviewers. Eventually 12 manuscripts have been selected to form this special issue of *IEEE Transactions on Computers*. Below, we provide a brief explanation of the contribution of the paper for the special issue.

The first paper, "Crane: Mitigating accelerator under-utilization caused by sparsity irregularities in CNNs" by Y. Guan *et al.*, proposes a method of load-balancing based on a workload stealing technique, in order to mitigate the problem of computation resource under-utilization in sparse CNN accelerators. Based on this method, the authors present an accelerator, called Crane, which addresses all kinds of sparsity irregularities in CNNs. Experimental results show that Crane improves performance by 27%–88% and reduces energy consumption by 16%–48%, respectively.

The second paper, "CIMAT: A compute-in-memory architecture for on-chip training based on transpose SRAM arrays" by H. Jiang *et al.*, designs the data flow for the back-propagation process and weight update to support the on-chip training based on CIM (compute-in-memory) approach. The authors utilize the mature and advanced CMOS technology at 7nm to design the CIM architecture with 7T transpose SRAM array that supports bidirectional parallel read.

The third paper, "MViD: Sparse matrix-vector multiplication in mobile DRAM for accelerating recurrent neural networks" by B. Kim *et al.*, proposes a main-memory architecture called MViD, which performs MV-mul (matrix-

- *Xuehai Qian is with Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA. E-mail: xuehai.qian@usc.edu.*
- *Yanzhi Wang is with Electrical and Computer Engineering, Northeastern University, Boston, MA. E-mail: yanz.wang@northeastern.edu.*
- *Avinash Karanth is with School of Electrical Engineering and Computer Science, Ohio University, Athens, OH. E-mail: karanth@ohio.edu.*

Digital Object Identifier no. 10.1109/TC.2020.2997574

vector multiplication) by placing MAC units inside DRAM banks. For higher computational efficiency, the authors use a sparse matrix format and exploit quantization. Because of the limited power budget for DRAM devices, the authors architect MViD to slow down or pause MV-mul for concurrently processing memory requests from processors, while satisfying the limited power budget.

The fourth paper, "Addressing irregularity in sparse neural networks through a cooperative software/hardware approach" by Z. Xi *et al.*, proposes a cooperative software/hardware approach to address the irregularity of sparse neural networks efficiently. Initially, the local convergence is observed, namely larger weights tend to gather into small clusters during training. Based on the observation, the authors propose a software-based coarse-grained pruning technique to reduce the irregularity of sparse synapses drastically. The authors further design a multi-core hardware accelerator, Cambricon-SE, to address the remaining irregularity of sparse synapses and neurons efficiently.

The fifth paper, "Enabling efficient fast convolution algorithms on GPUs via MegaKernels" by L. Jia *et al.*, proposes a kernel fusion technique for fast convolution algorithms based on MegaKernels. Each GPU thread block is assigned with one computation stage and the authors design a parameterized task mapping algorithm to assign stages to thread blocks. A task scheduler has been built which fetches and executes the tasks following the task dependency.

The sixth paper, "Machine learning computers with fractal von Neumann architecture" by Y. Zhao *et al.*, proposes Cambricon-F, which is a series of homogeneous, sequential, multi-layer, layer-similar, machine learning computers with the same ISA. A Cambricon-F machine has a fractal von Neumann architecture to iteratively manage its components: it is with von Neumann architecture and its processing components (sub-nodes) are still Cambricon-F machines with von Neumann architecture and the same ISA.

The seventh paper, "Distributed training of support vector machine on a multiple-FPGA system" by J. Dass *et al.*, proposes and implements a first-of-its-kind system of multiple FPGAs as a distributed computing framework comprising up to eight FPGA units on Amazon F1 instances with negligible communication overhead to fully parallelize, accelerate, and scale the SVM (support vector machine) training on decentralized data.

The eighth paper, "A neural network-based on-device learning anomaly detector for edge devices" by M. Tsukada *et al.*, proposes ONLAD (on-device learning anomaly detector) and its IP core, named ONLAD Core. ONLAD is highly optimized to perform fast sequential learning to follow concept drift in less than one millisecond. ONLAD Core realizes on-device learning for edge devices at low power consumption, which realizes standalone execution where data transfers between edge and server are not required.

The ninth paper, "Pre-defined sparsity for low-complexity convolutional neural networks" by S. Kundu *et al.*, introduces convolutional layers with pre-defined sparse 2D kernels that have support sets that repeat periodically within and across filters. Due to the efficient storage of the proposed periodic sparse kernels, the parameter savings can translate into considerable improvements in energy efficiency due to reduced DRAM accesses, thus promising significant improvements in the trade-off between energy consumption and accuracy for both training and inference of DNNs.

The tenth paper, "Accelerating deep learning systems via critical set identification and model compression" by R. Han *et al.*, proposes ClipDL which accelerates the deep learning systems by simultaneously decreasing the number of model parameters and reducing the computations on critical data only. The core component of ClipDL is the estimation of critical set based on the observation that large proportions of input data have little influence on model parameter updating in many prevalent deep learning algorithms.

The eleventh paper, "WooKong: A ubiquitous accelerator for recommendation algorithms with custom instruction sets on FPGA" by C. Wang *et al.*, proposes WooKong, a ubiquitous accelerator architecture for the collaborative-filtering recommendation on FPGA. It is able to accommodate three types of CF (collaborative filtering) recommendation algorithms, including User-based CF, Item-based CF, and SlopeOne recommendations algorithms, with five different similarity analysis metrics including Jaccard, Cosine, CosineIR, Euclidean, and Pearson. To maintain flexibility for these different CF algorithms and metrics, the authors adopt custom instruction sets to manipulate the training and inferences accelerators.

The twelfth paper, "π-BA: Bundle Adjustment Hardware Accelerator based on Distribution of 3D-Point Observations" by Q. Liu *et al.*, proposes π-BA, the first hardware-software co-designed BA (bundle adjustment) hardware accelerator that exploits custom hardware to simultaneously achieve higher performance and power efficiency. Specifically, based on the key observation that not all 3D points appear on all images in a BA problem, the authors designed a Co-Observation Optimization technique to accelerate BA operations with optimized usage of memory and computation resources. In addition, a hardware-friendly differentiation method is developed, which combines the analytic and forward automatic differentiation to calculate derivatives of projection function in the BA problem.

We would like to express our sincere gratitude to all authors who submitted their work to this special issue. We also would like to thank the anonymous reviewers for their invaluable help in evaluating and judging the submissions. Further on, it is our pleasure to thank the Editor-in-Chief Ahmed Louri and Associate Editors Tao Li and James Hoe for their continuous help and support with all our organizational questions in connection with this special issue.

Xuehai Qian, *Guest Editor*
Ming Hsieh Dept of Electrical and Computer Engineering
University of Southern California, Los Angeles, CA.
Email: xuehai.qian@usc.edu

Yanzhi Wang, *Guest Editor*
Electrical and Computer Engineering
Northeastern University, Boston, MA.
Email: yanz.wang@northeastern.edu

Avinash Karanth, *Corresponding Topical Editor*
School of Electrical Engineering and Computer Science
Ohio University, Athens, OH.
Email: karanth@ohio.edu