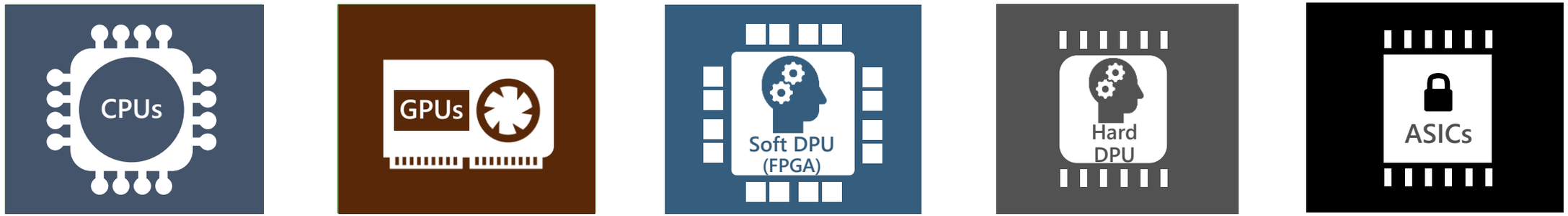
Microsoft

FPGA'18 Panel

Eric Chung
Senior Researcher
Microsoft Research

We are witnessing a Cambrian explosion of specialized architectures for Deep Learning



Thanks to the end of Moore's law and the rise of deep learning, architecture is exciting again!

The Deep Learning Wild West

- CNNs
  - 1D for text, 2D for image, 3D for video
- RNNs
  - Word vector and character embeddings
  - Bidirectional LSTM, LSTMP, GRU, etc
  - Attention layers, highway networks, etc.
- Full system considerations
  - Image/text preprocessing, image decompression, etc
  - Transfer learning
- Serving vs. Training
  - Want real-time processing (batch 1) for serving, high batch tolerable for training or offline serving
- Numerical precision

It's all just dense floating point matrix-multiply right?

CirCNN: Accelerating and Compressing Deep Neural Networks Using Block-Circulant Weight Matrices

Caiwen Ding[+,1], Siyu Liao[+,2], Yanzhi Wang[+,1], Zhe Li[1], Ning Liu[1], Youwei Zhuo[3], Chao Wang[3], Xuehai Qian[3], Yu Bai[4], Geng Yuan[1], Xiaolong Ma[1], Yipeng Zhang[1], Jian Tang[1], Qinru Qiu[1], Xue Lin[5], Bo Yuan[2]
[+]These authors contributed equally.
[1]Syracuse University, [2]City University of New York, City College, [3]University of Southern California, [4]California State University Fullerton, [5]Northeastern University
{cading,ywang393,zli89,nliu03,geyuan,xma27,yzhan139,jtang02,qiqiu}@syr.edu,sliao2@gradcenter.cuny.edu,
{youweizh,wang484,xuehai.qian}@usc.edu,ybai@exchange.fullerton.edu,xue.lin@northeastern.edu,byuan@ccny.cuny.edu

ABSTRACT

Large-scale deep neural networks (DNNs) are both compute and memory intensive. As the size of DNNs continues to grow, it is critical to improve the energy efficiency and performance while maintaining accuracy. For DNNs, the model size is an important factor affecting performance, scalability and energy efficiency. Weight pruning achieves good compression ratios but suffers from three drawbacks: *1)* the irregular network structure after pruning, which affects performance and throughput; *2)* the increased training complexity; and *3)* the lack of rigorous guarantee of compression ratio and inference accuracy.

To overcome these limitations, this paper proposes CirCNN, a principled approach to represent weights and process neural networks using *block-circulant* matrices. CirCNN utilizes the *Fast Fourier Transform (FFT)*-based fast multiplication, *simultaneously* reducing the computational complexity (both in inference and training) from $O(n^2)$ to $O(n \log n)$ and the storage complexity from $O(n^2)$ to $O(n)$, with negligible accuracy loss. Compared to other approaches, CirCNN is distinct due to its mathematical rigor: the DNNs based on CirCNN can converge to the same "effectiveness" as DNNs without compression. We propose the CirCNN architecture, a universal DNN inference engine that can be implemented in various hardware/software platforms with configurable network architecture (e.g., layer type, size, scales, etc.). In CirCNN architecture: *1)* Due to the recursive property, *FFT can be used as the key computing kernel*, which ensures universal and small-footprint implementations. *2)* The *compressed but regular* network structure avoids the pitfalls of the network pruning and facilitates high performance and throughput with highly pipelined and parallel design. To demonstrate the performance and energy efficiency, we test CirCNN in FPGA, ASIC and embedded platforms. Based on the FPGA implementation and ASIC synthesis results, CirCNN achieves 6 - 102X energy efficiency improvements compared with the best state-of-the-art results.

CCS CONCEPTS

• **Computer systems organization → Embedded hardware**;

KEYWORDS

Deep learning, block-circulant matrix, compression, acceleration, FPGA

ACM Reference format:
Caiwen Ding*,1, Siyu Liao*,2, Yanzhi Wang*,1, Zhe Li[1], Ning Liu[1], Youwei Zhuo[3], Chao Wang[3], Xuehai Qian[3], Yu Bai[4], Geng Yuan[1], Xiaolong Ma[1], Yipeng Zhang[1], Jian Tang[1], Qinru Qiu[1], Xue Lin[5], Bo Yuan[2]. 2017. CirCNN: Accelerating and Compressing Deep Neural Networks Using Block-Circulant Weight Matrices. In *Proceedings of MICRO-50, Cambridge, MA, USA, October 14–18, 2017,* 14 pages.
https://doi.org/10.1145/3123939.3124552

1 INTRODUCTION

From the end of the first decade of the 21st century, neural networks have been experiencing a phenomenal resurgence thanks to the big data and the significant advances in processing speeds. Large-scale deep neural networks (DNNs) have been able to deliver impressive results in many challenging problems. For instance, DNNs have led to breakthroughs in object recognition accuracy on the ImageNet dataset [1], even achieving human-level performance for face recognition [2]. Such promising results triggered the revolution of several traditional and emerging real-world applications, such as

EIE: Efficient Inference Engine on Compressed Deep Neural Network

Song Han* Xingyu Liu* Huizi Mao* Jing Pu* Ardavan Pedram*
Mark A. Horowitz* William J. Dally*†
*Stanford University, †NVIDIA
{songhan,xyl,huizi,jingpu,perdavan,horowitz,dally}@stanford.edu

***Abstract*—State-of-the-art deep neural networks (DNNs) have hundreds of millions of connections and are both computationally and memory intensive, making them difficult to deploy on embedded systems with limited hardware resources and power budgets. While custom hardware helps the computation, fetching weights from DRAM is two orders of magnitude more expensive than ALU operations, and dominates the required power.**

Previously proposed 'Deep Compression' makes it possible to fit large DNNs (AlexNet and VGGNet) fully in on-chip SRAM. This compression is achieved by pruning the redundant connections and having multiple connections share the same weight. We propose an energy efficient inference engine (EIE) that performs inference on this compressed network model and accelerates the resulting sparse matrix-vector multiplication with weight sharing. Going from DRAM to SRAM gives EIE 120× energy saving; Exploiting sparsity saves 10×; Weight sharing gives 8×; Skipping zero activations from ReLU saves another 3×. Evaluated on nine DNN benchmarks, EIE is 189× and 13× faster when compared to CPU and GPU implementations of the same DNN without compression. EIE has a processing power of 102 GOPS/s working directly on a compressed network, corresponding to 3 TOPS/s on an uncompressed network, and processes FC layers of AlexNet at 1.88×10^4 frames/sec with a power dissipation of only 600mW. It is 24,000× and 3,400× more energy efficient than a CPU and GPU respectively. Compared with DaDianNao, EIE has 2.9×, 19× and 3× better throughput, energy efficiency and area efficiency.

***Keywords*-Deep Learning; Model Compression; Hardware Acceleration; Algorithm-Hardware co-Design; ASIC;**

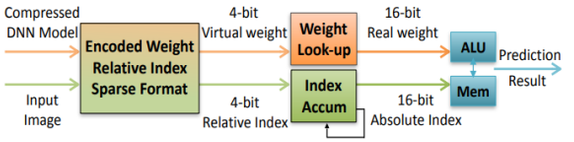


Figure 1. Efficient inference engine that works on the compressed deep neural network model for machine learning applications.

word, or speech sample. For embedded mobile applications, these resource demands become prohibitive. Table I shows the energy cost of basic arithmetic and memory operations in a 45nm CMOS process [9]. It shows that the total energy is dominated by the required memory access if there is no data reuse. The energy cost per fetch ranges from 5pJ for 32b coefficients in on-chip SRAM to 640pJ for 32b coefficients in off-chip LPDDR2 DRAM. Large networks do not fit in on-chip storage and hence require the more costly DRAM accesses. Running a 1G connection neural network, for example, at 20Hz would require $(20Hz)(1G)(640pJ) = 12.8W$ just for DRAM accesses, which is well beyond the power envelope of a typical mobile device.

Previous work has used specialized hardware to accelerate DNNs [10]–[12]. However, these efforts focus on accelerating dense, uncompressed models - limiting their utility to small models or to cases where the high energy cost

Why FPGAs for Deep Learning

- Excellent performance, energy efficiency at low precisions
- Excellent for low-batch real-time AI
- Can synthesis-specialize for different models
- Can exploit novel algorithms beyond dense FP matrix-multiply
- Easily future-proofed
- Ability to iterate rapidly in production environments
- In a diverse cloud, applicable to non-DL workloads as well

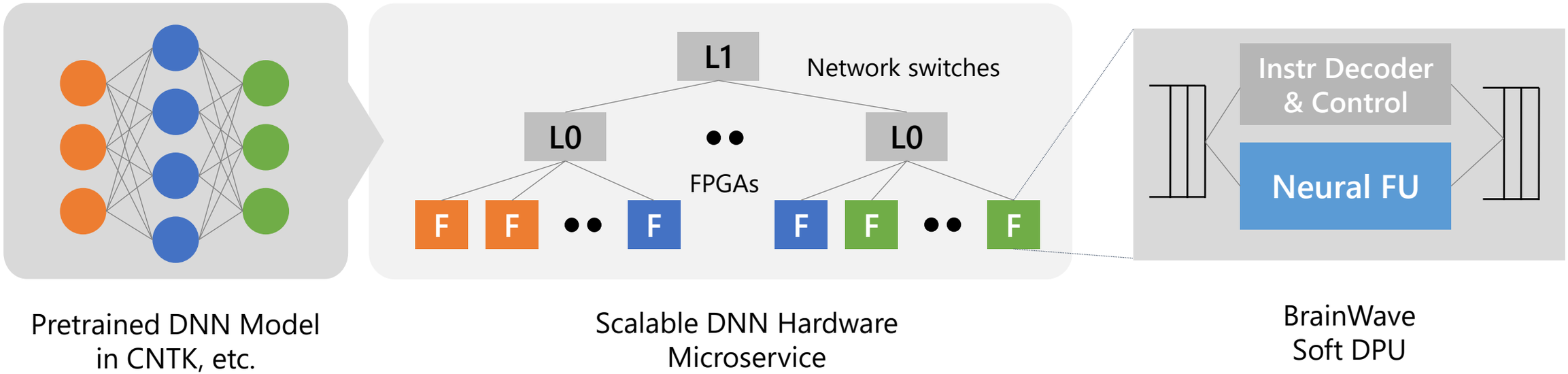
Project BrainWave

A DNN Serving Platform for Real-time AI

Fast: ultra-low latency, high-throughput serving of DNN models at low batch sizes

Flexible: adaptive numerical precision and custom operators

Friendly: turnkey deployment of CNTK/Caffe/TF/etc

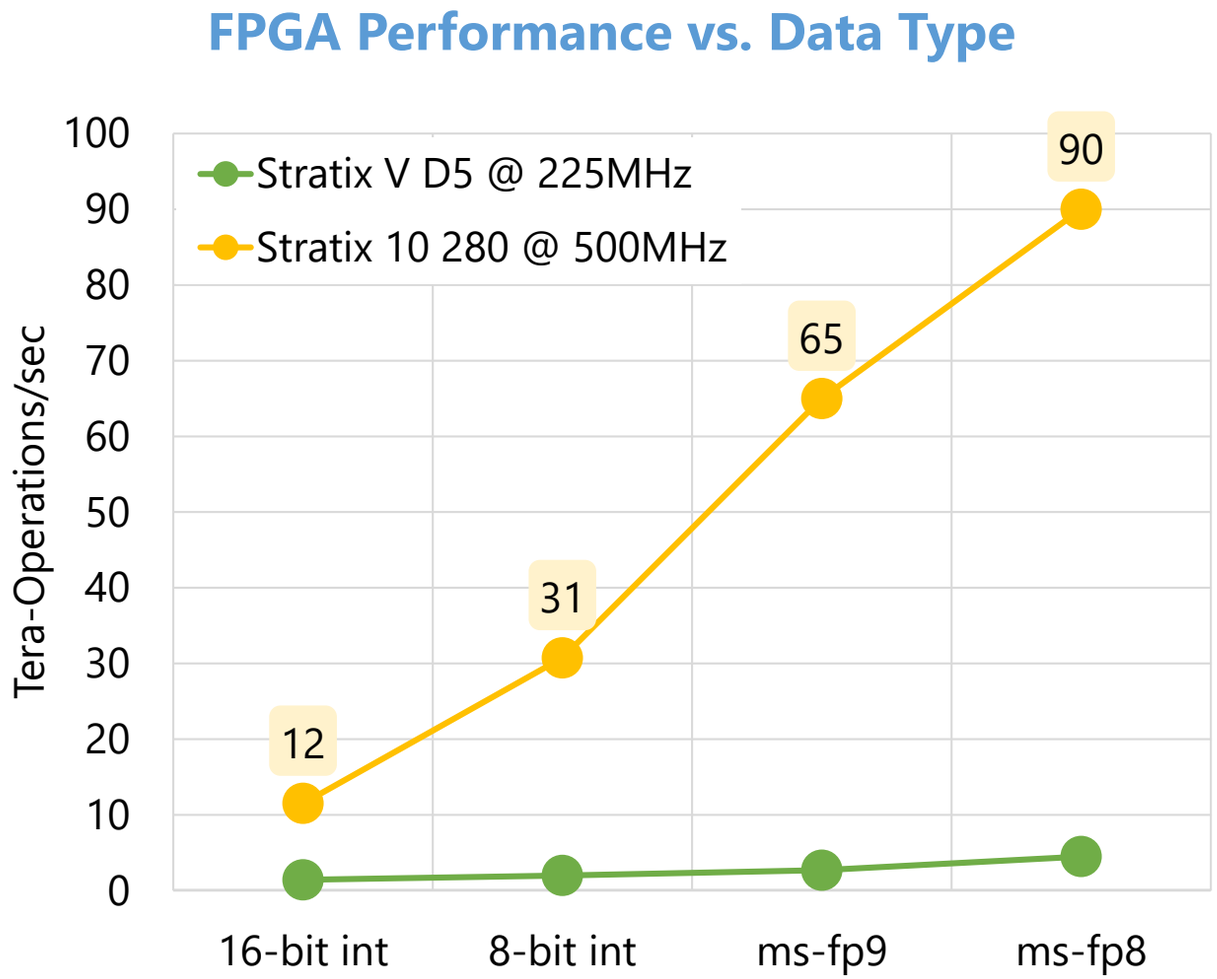


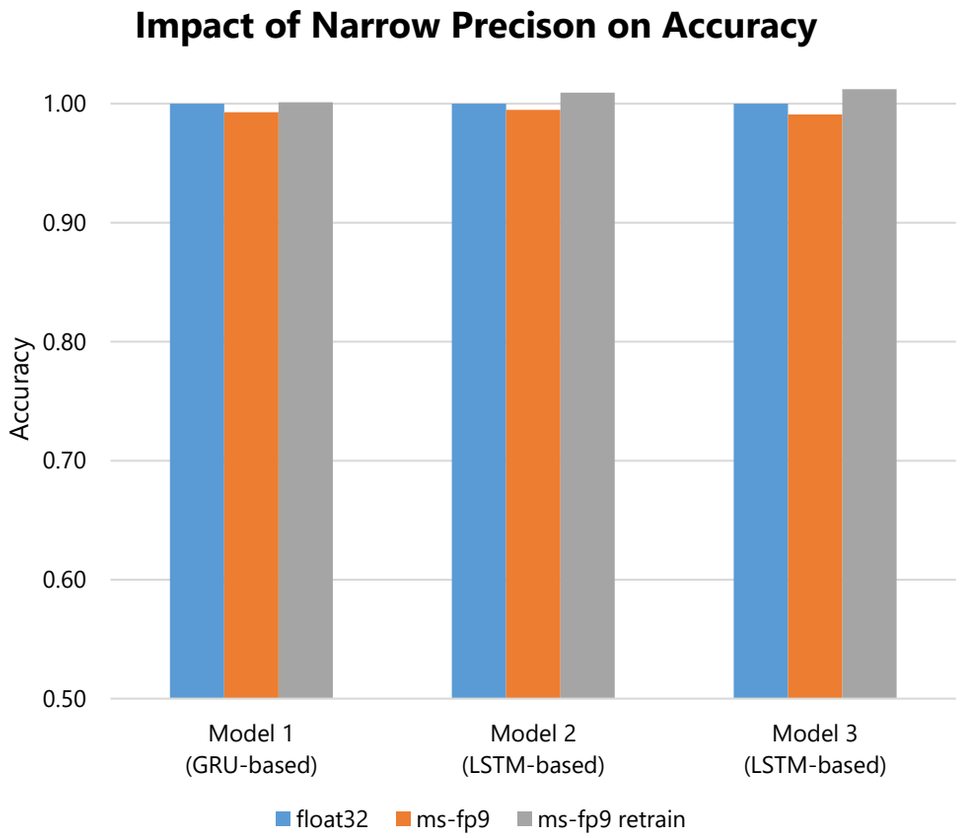
Pretrained DNN Model in CNTK, etc.

Scalable DNN Hardware Microservice

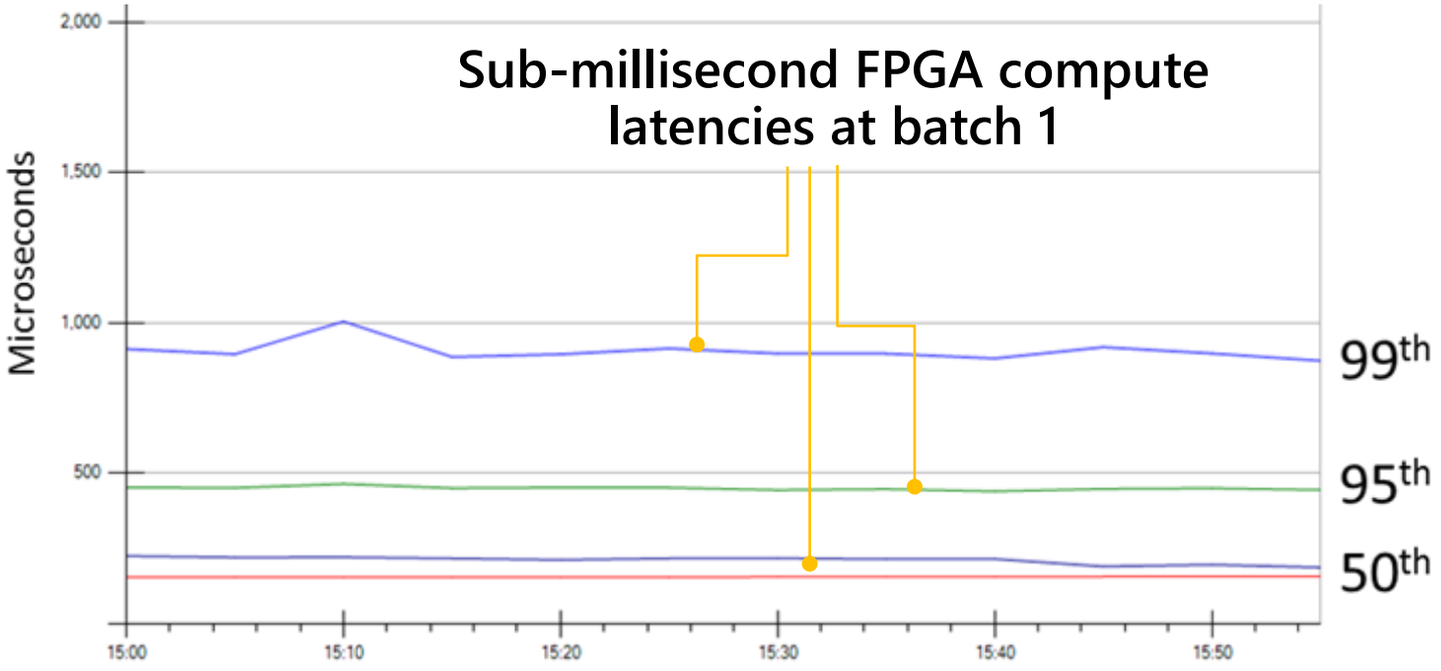
BrainWave Soft DPU

Narrow Precision Inference on FPGAs





Deployed in Production Datacenters



Deployment of LSTM-based NLP model (tens of millions of parameters)

Takes tens of milliseconds to serve on well-tuned CPU implementations

Tail latencies in BrainWave-powered DNN models appear negligible in E2E software pipelines

Parting thoughts

- Algorithmic innovation trumps brute force hardware
- The best recipe: algorithmic + hardware co-optimization
- FPGAs are well positioned for deep learning in the cloud
- Stay tuned for more announcements in 2018
- Contact: Eric Chung (erchung@microsoft.com)