Bringing Physical Dimensions to Neuromorphic Computing

Farinaz Koushanfar[1] and Tinoosh Mohsenin[2]

[1] Professor of ECE, University of California San Diego (UCSD)

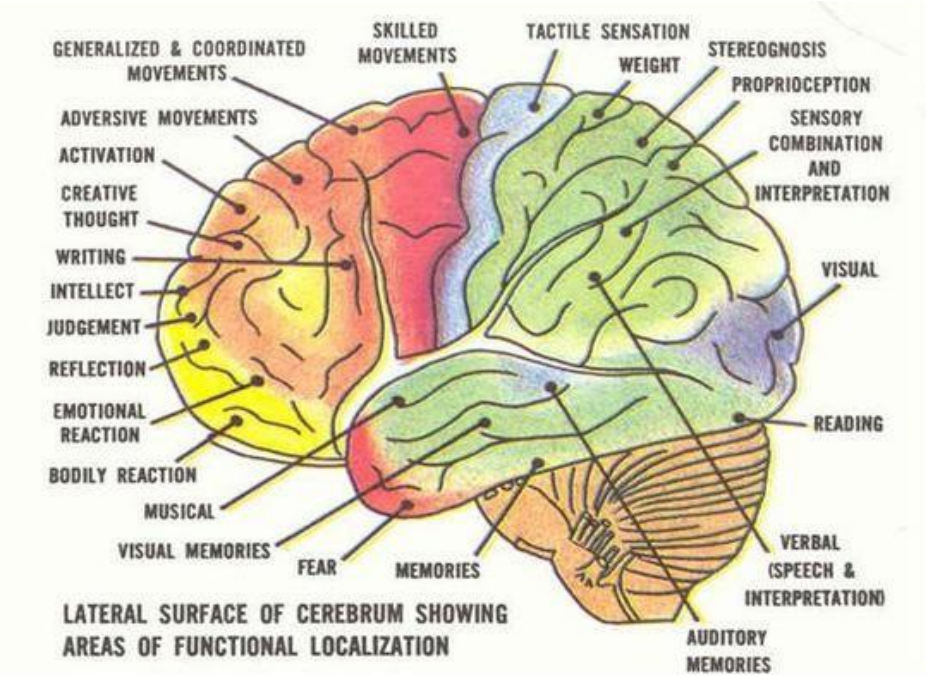
[2]Assistant Professor of CSEE, University of Maryland Baltimore County (UMBC)

Outline

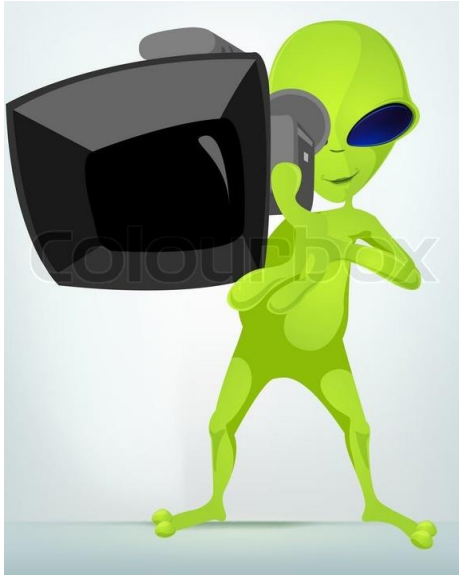
- Motivation – brain
- Brain-inspired computing
- Suggested architecture/algorithm research thrusts
  - Holistic performance-driven dimensionality reduction
  - System and network topology
  - Access architecture and order
  - Automation

Human brain?

- The biggest marvel of all!
- How can we learn from and mimic the human brain?



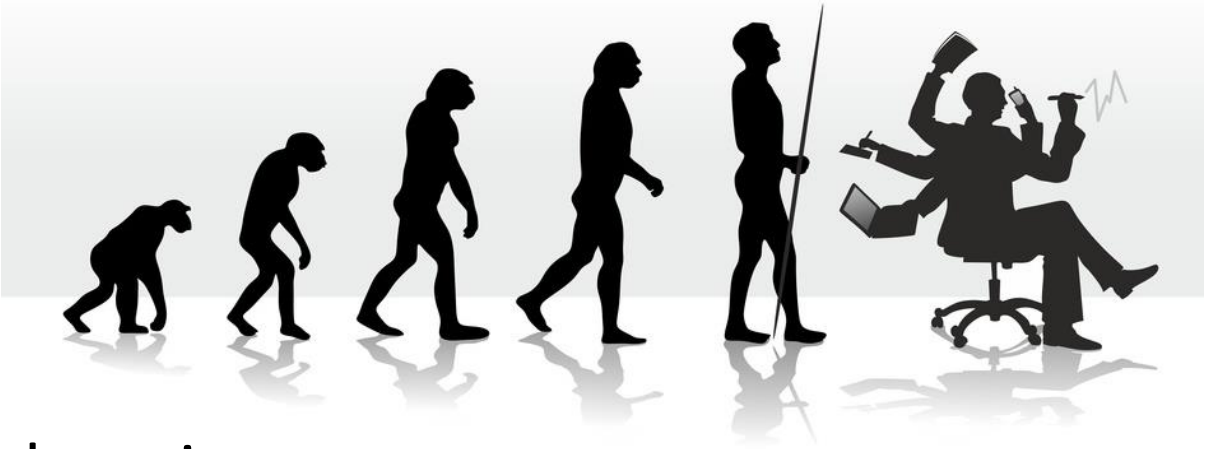
Can an alien experimentally learn and build a CPU?



- Scenario:
  - The computer looks cool to the alien!
  - Dissects the box to find the intricate CPU
  - Brings in alien technology to image, delayer, experiment, and learn physics of the silicon, transistor, gates, wires, etc.

- Can she teach other aliens to build a working computer now?
  - Probably not!
  - Unless she figures out the modular structure, functionality, architecture, and software

What we know about the human brain?



- In the learning process…
- What we know:
  - The underlying neurons and synapses are similar to other primates in terms of material and connectivity speed
  - However, there are visible differences in terms of connectivity and focused centers for computing
- The human brain follows an adaptive, data-driven, and domain-specific architecture

Can we reach the computing efficiency of the human brain?

BRAIN-INSPIRED COMPUTING

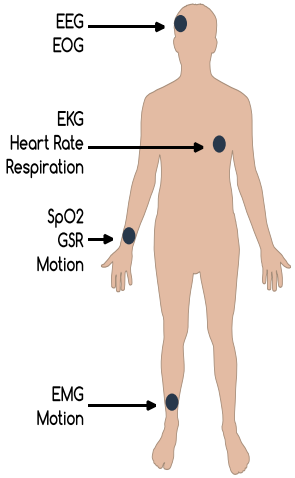
Brain-inspired computing

- Spectrum of brain models with various underlying device mechanisms, from very nonlinear complex to more simplified structures
- A surge of emerging computing fabrics and architectures
- The biggest challenge (and opportunity)
  - Alien problem: End-to-end holistic view of the system
- Simple example: deep neural networks

Deep learning revolution

Computer Vision

Cyber-Physical Systems

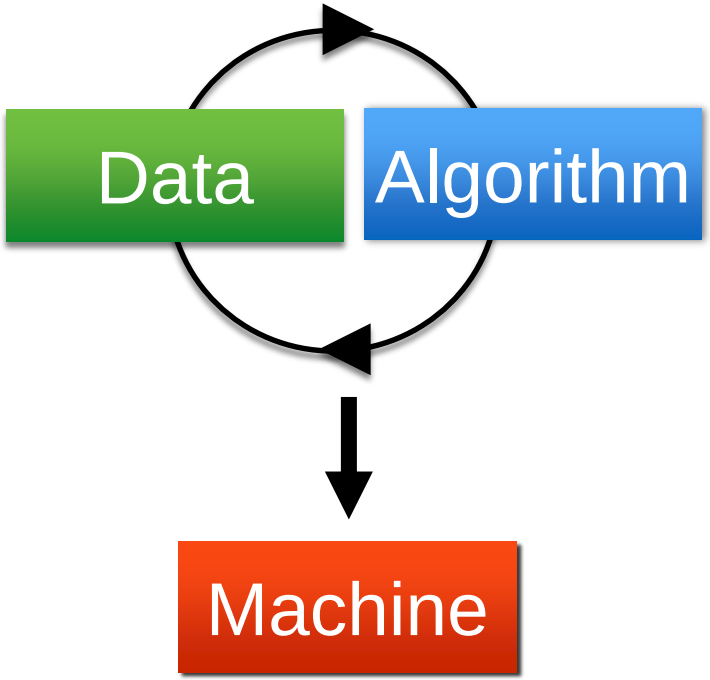


Speech Recognition

Search

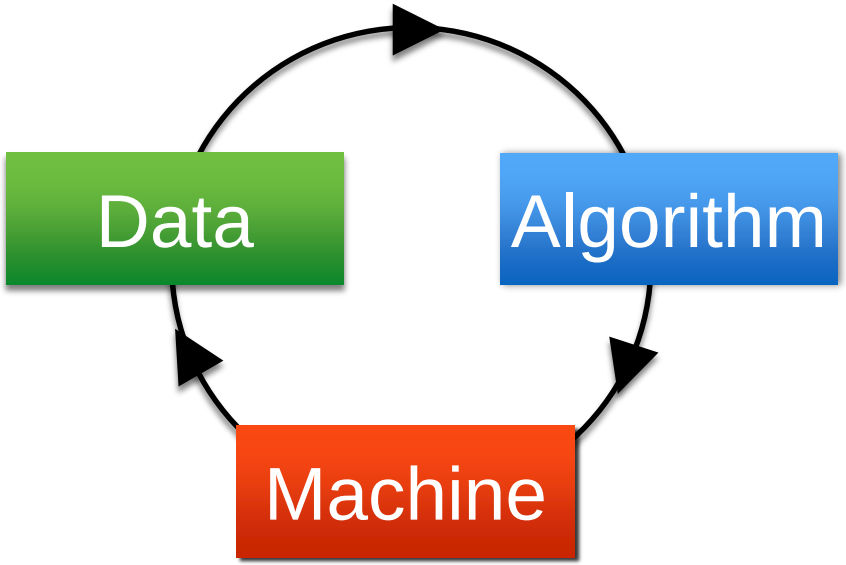
Holistic dimensionality reduction

- Contemporary practice in system design for data-driven problems



Holistic dimensionality reduction

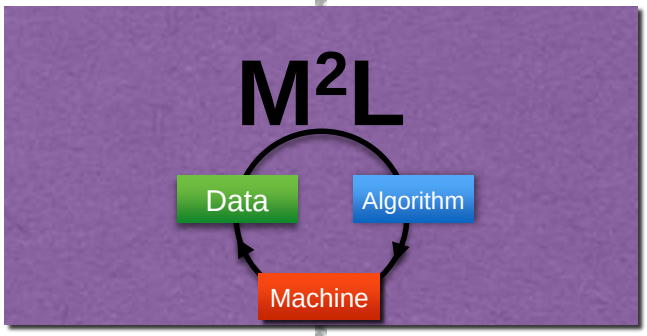
- Our suggested methodology



Automation

Data

Algorithm

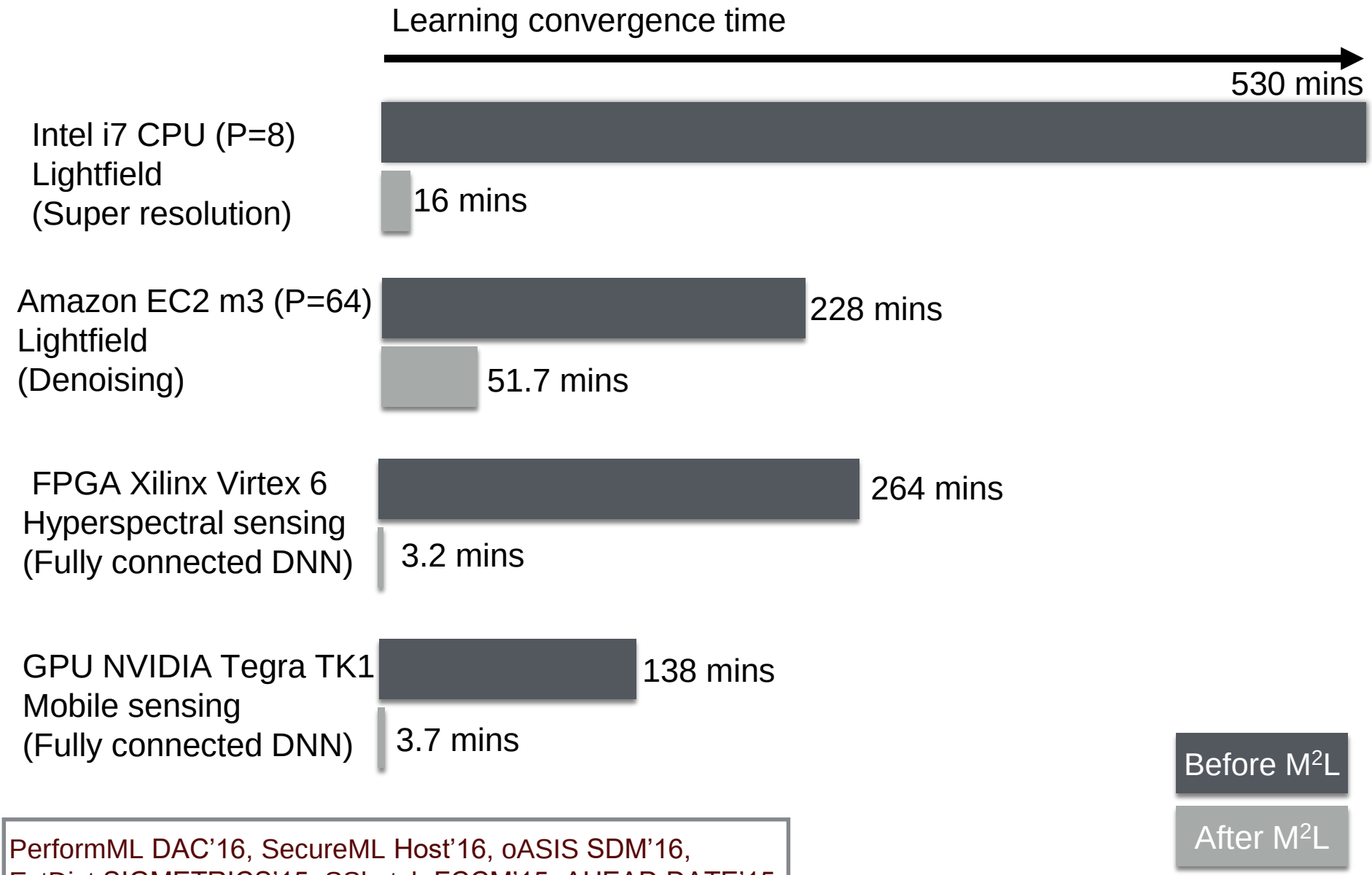


Machine

amazon web services™

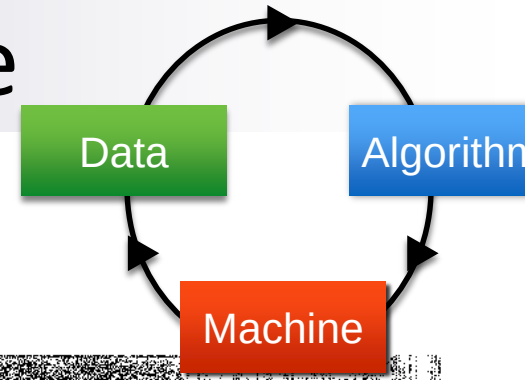
11

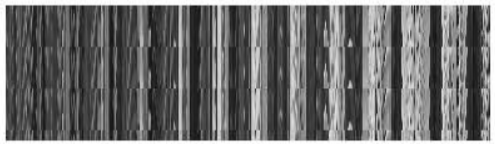
Some of M^{2}L runtime improvement results



PerformML DAC'16, SecureML Host'16, oASIS SDM'16,
ExtDict SIGMETRICS'15, SSketch FCCM'15, AHEAD DATE'15

New bounds on performance



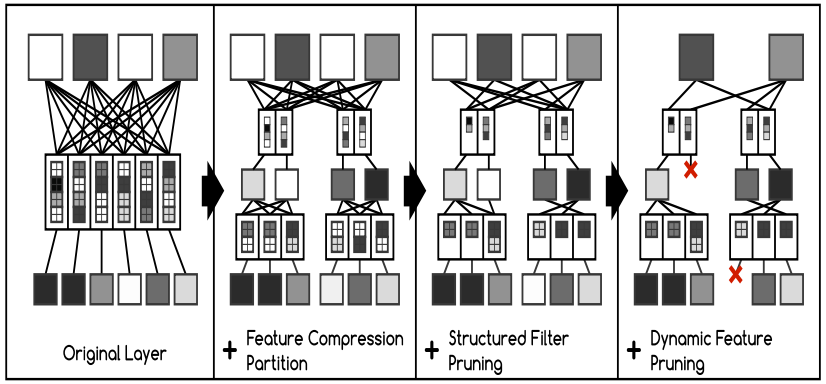


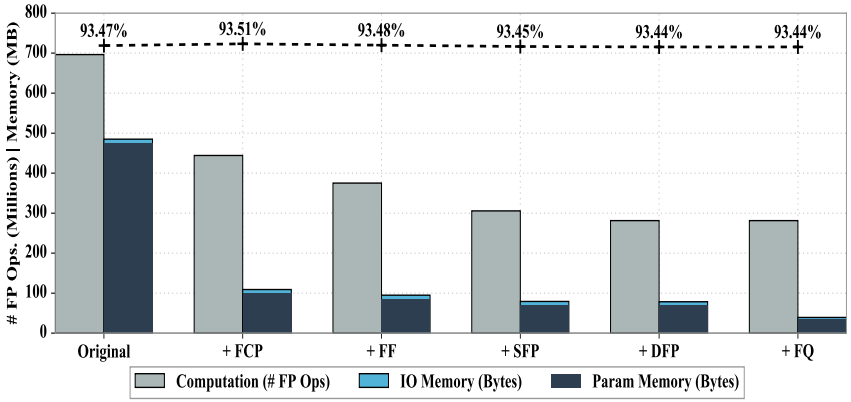
	Prior bounds [1]	M^2L [2]
Memory footprint (Byte)	$\frac{MN}{P}$	$LM + \frac{\alpha(L, A, \delta_f)N}{P}$
Computation (FLOP)	$\frac{MN}{P}$	$LM + \frac{\alpha(L, A, \delta_f)N}{P}$
Communication (Byte)	MP	$[L, LP]$

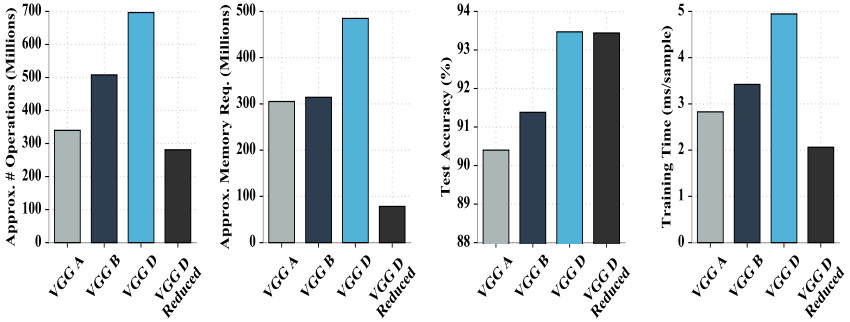
[1] Demmel et al., IPDPS'13
[2] Mirhoseini et al., DAC16

Network Reduction Techniques

- Look to reduce complexity at architectural level by removing dense connectivity
- 3 Sparsification Techniques:
  - Feature compression partition
  - Structured filter pruning
  - Dynamic feature pruning
- 2 Approximation Techniques:
  - Filter factorization
  - Filter quantization
- Demonstrated to reduce computation and memory by **60%** and **93%** w/ **<0.03%** impact on accuracy compared to baseline VGGNet on CIFAR dataset

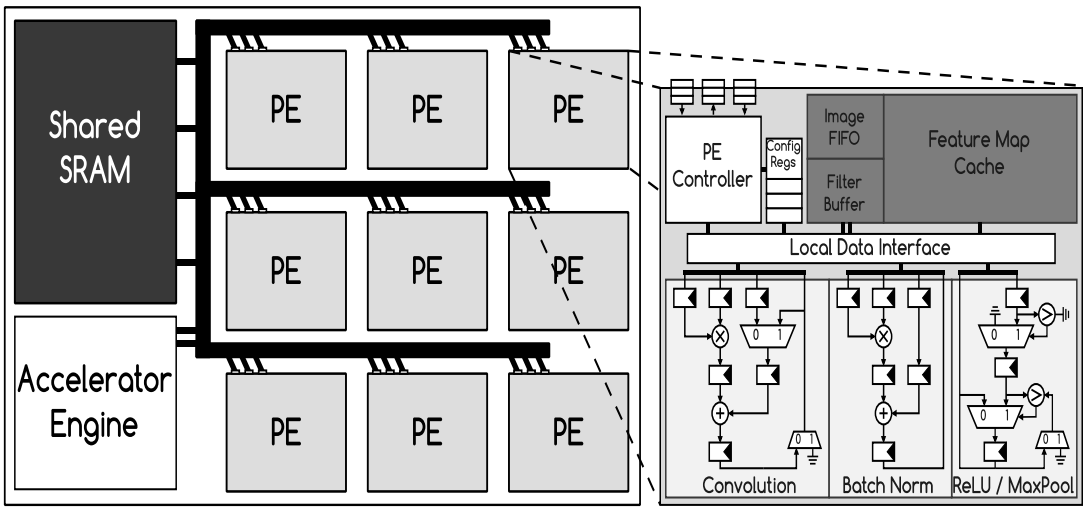


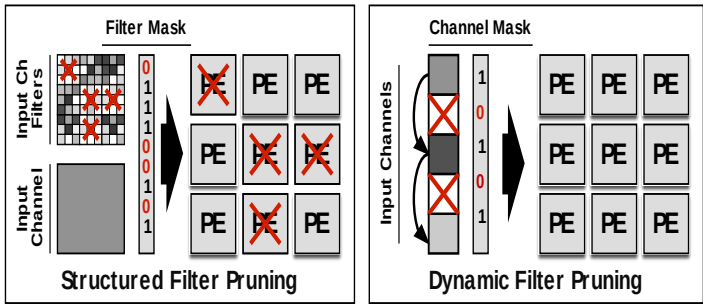




Page et al. JETC'16

SPARCNet: *SPAR*se Convolutional *NET*work Accelerator



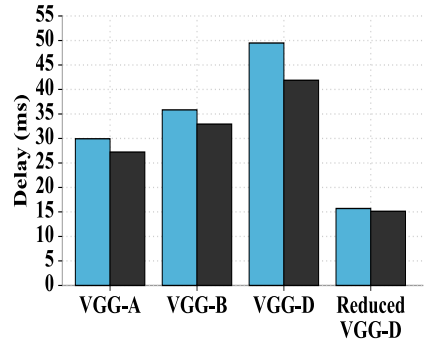


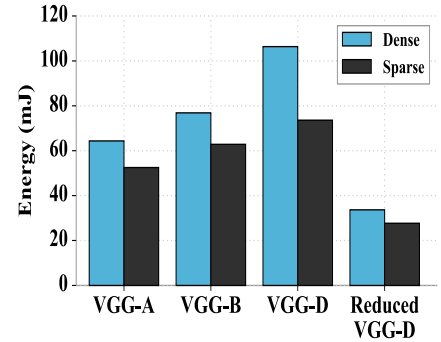
- SPARCNet is an accelerator for efficient deployment of convolutional neural networks in embedded, real-time systems.
- Processing engines perform concurrent layer operations such as 1D/2D convolutional, max-pooling, batch normalization, and ReLU.
- Operations are done using 16-bit floating-point.
- Router consists of 2 buses that are unidirectional with each having 4 16-bit channels. Channels support direct and broadcast communication patterns.
- Built-in support for three sparsification techniques.

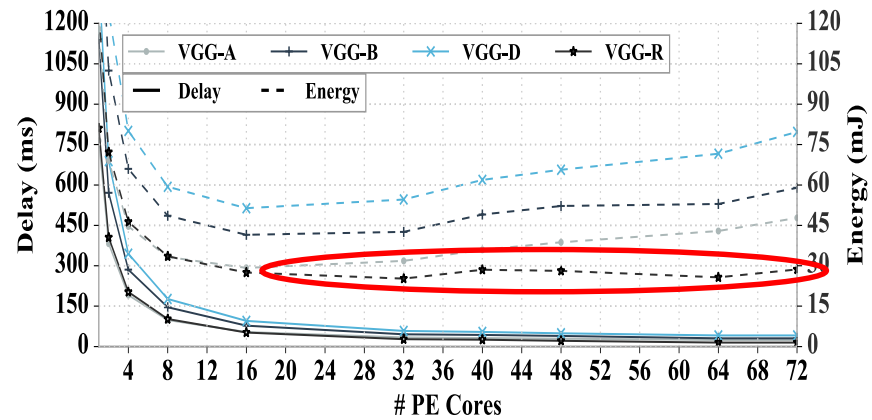
Page et al. JETC'16

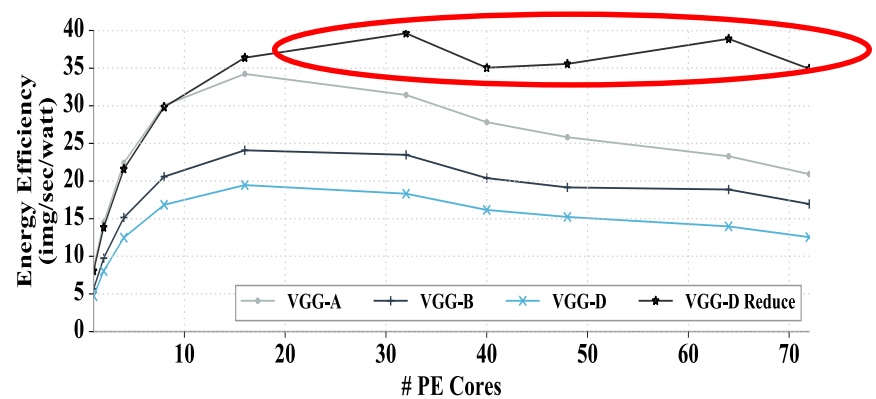
SPARCNet: Implementation & Comparison

- Evaluated on 4 convolutional neural network topologies with varying depths.
- Classification accuracy evaluated on MNIS, CIFAR, and SVHN dataset.
- Explored impact on throughput and energy using varying # PEs.
- Depending on network topology, there exists optimal # PEs that provide optimal efficiency.
- ***Reduced VGG-D*** is able to obtain much higher efficiency and benefit from increased PEs.



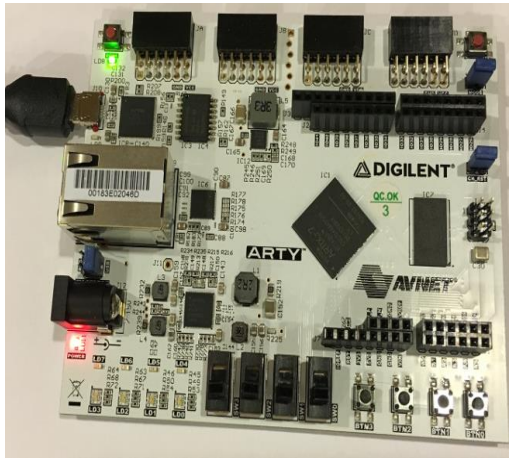






FPGA, CPU, GPU Comparison

- **Arty Artix-7 FPGA Board**
  - Artix-35T FPGA
    (5200 slices, 1800 Kb BRAM, 90 DSPs)
  - 256 MB DDR3 memory
  - Embedded Microblaze bare metal
- **NVIDIA Jetson TK1 Board**
  - 4+1 Quad-Core ARM Cortex-A15
  - K1 GPU with 192 CUDA Cores
  - 2 GB DDR3 memory
  - 16 GB eMMC
  - Running custom Ubuntu Linux
- **NVIDIA Jetson TX1 Board**
  - 64-bit A57 CPUs
  - X1 GPU with 256 CUDA Cores
  - 4 GB DDR4 memory
  - 16 GB eMMC
  - Running custom Ubuntu Linux



FPGA, CPU and GPU Comparison

- Compared on CIFAR dataset using VGGNet network with 16 computational layers.
- FPGA w/ SPARCNet accelerator improves energy efficiency by:
  - **50x** compared to base TK1 CPU only
  - **11.7x** compared to TK1 GPU implementation
  - **7.5x** compared to TX1 GPU implementation

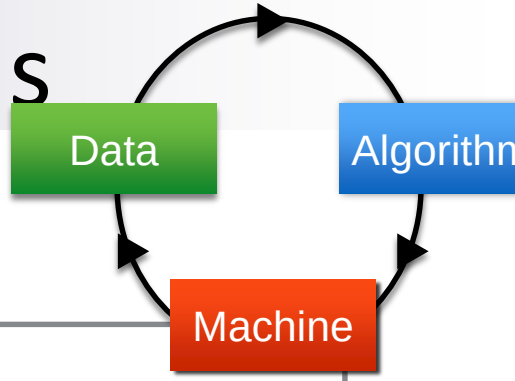
Platform	Throughput (img/s)	Power (W)	Energy (J)	Execution Time (s)	Energy Efficiency (img/s/w)	Improvement Over Base[1]
NVIDIA TK1 (base) CPU	1.01	12.5	14309	4943.82	0.08	–
NVIDIA TK1 w/ GPU	42.86	12.5	730	58.33	3.43	43x
NVIDIA TX1 w/ GPU	51.18	9.6	472	48.85	5.33	66x
SPARCNet-64 Artix-7	72.6	1.80	83.58	34.43	**40.33**	500x

FPGA Accelerator Comparison

- Compared to a number of existing accelerators using AlexNet network topology for vision tasks.
- SPARCNet has power consumption of 1.82 W & energy efficiency of 29.96 GOP/J.
- SPARCNet outperforms the next best accelerators [Zhang et al. 2015a] and [Qiu et al. 2016] by factors of 9X and 2X in efficiency.

Metrics	[Chakradhar et al. 2010]	[Gokhale et al. 2014]	[Zhang et al. 2015a]	[Qiu et al. 2016]	**This work**
Platform	Virtex-5 (SX240T)	Zynq (XC7Z045)	Virtex-7 (VX485T)	Zynq (XC7Z045)	**Artix-7 (XC7A200T)**
Precision	48-bit Fixed	16-bit Fixed	32-bit Float	16-bit Fixed	**16-bit Float**
Clock (MHz)	120	150	100	150	**100**
Network Complexity (GOP)	0.52	0.552	1.33	30.76[1]	**1.39**
Performance (GOP/s)	16	23.18	61.62	136.97	**54.52**
Total Power (W)	14	8	18.61	9.63	**1.82**
Energy Efficiency (GOP/J)	1.14	2.90	3.31	14.22	**29.96**

Evaluated domains and APIs



Datasets

A. Video
B. Hyperspectral
C. Light field camera
D. Cancer-tomography
E. Mobile sensing

Platforms

1. GP cluster: Amazon EC2, IBM iDataplex
2. GPGPU: NVIDIA GeForce, Jetson TK1
3. FPGA single acc: Xilinx Virtex 6
4. FPGA cluster: 24x Xilinx Virtex 6

ML application

(i) Visual classification
(ii) Super resolution
(iii) Denoising
(iv) Sensory recognition

Algorithms and APIs

- Streaming dictionary formation
- Iterative regression, e.g., LASSO, SVM
- Power iteration, e.g., PCA
- Distributed fully connected DNN

Improvement over state-of-the-art

- Data C, Platform 1, App (ii), 10.9x runtime
- Data E, Platform 2, App (iv), 4.8x power and 43.7 runtime
- Data E, Platform 4, App (i), 18.5x FLOPs and 76.3x runtime

Summary and Open Questions

- Develop holistic systems to bridge computation and physiology
  - Better understanding of brain functions for domain specification
  - Characterize the system in terms of learning capability, response time, energy, for multiple sets of problems
- For each new architecture/material provide programming and benchmarking sets for evaluation