Understanding the Limitations of Existing Energy-Efficient Design Approaches for Deep Neural Networks

Vivienne Sze

Massachusetts Institute of Technology

In collaboration with Yu-Hsin Chen, Joel Emer, Tien-Ju Yang

Based on SysML 2018 paper with the same title: Link

Contact Info

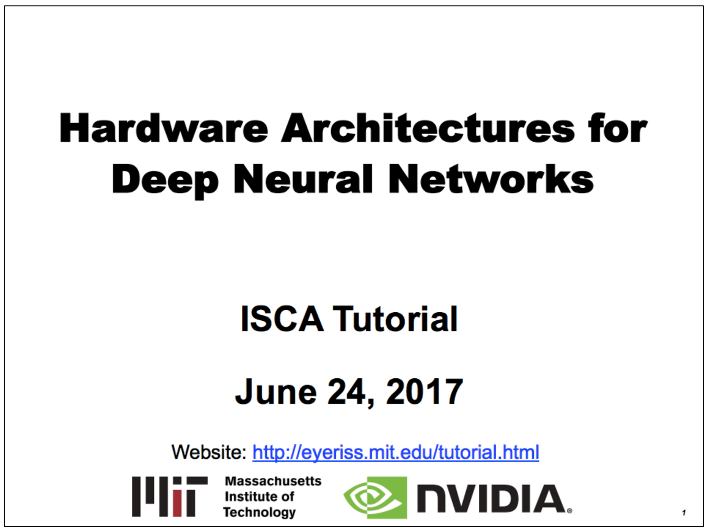
email: sze@mit.edu

website: www.rle.mit.edu/eems

Follow @eems_mit

Energy-Efficient Processing of DNNs

A significant amount of algorithm and hardware research on energy-efficient processing of DNNs



Hardware Architectures for Deep Neural Networks

ISCA Tutorial

June 24, 2017

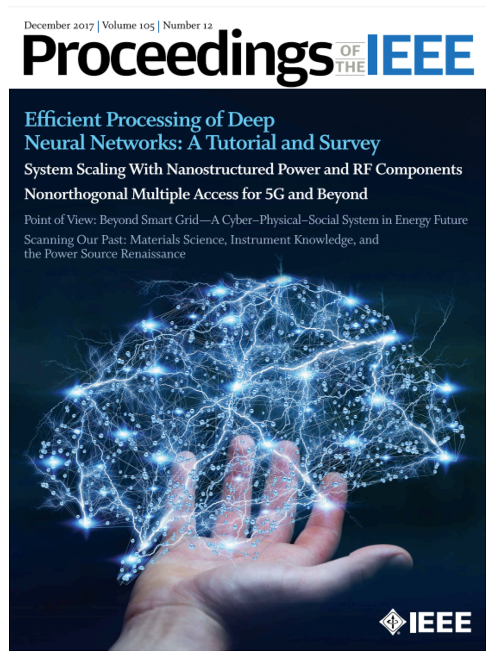
Website: http://eyeriss.mit.edu/tutorial.html

Massachusetts Institute of Technology

NVIDIA

1

http://eyeriss.mit.edu/tutorial.html



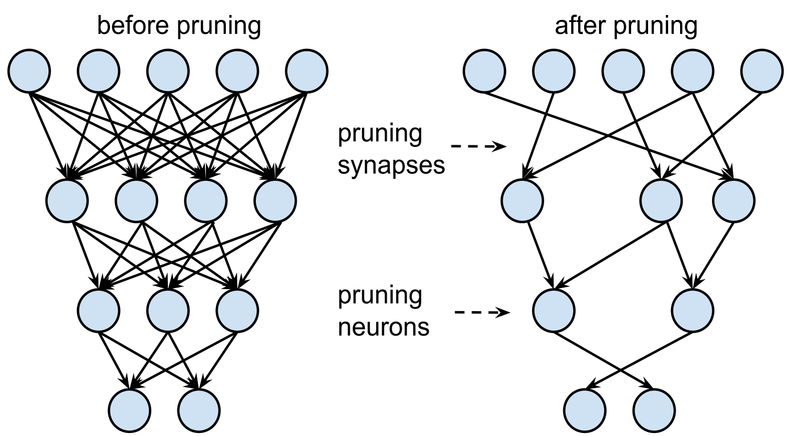
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, ***"Efficient Processing of Deep Neural Networks: A Tutorial and Survey,"*** Proceedings of the IEEE, Dec. 2017

We identified various limitations to existing approaches

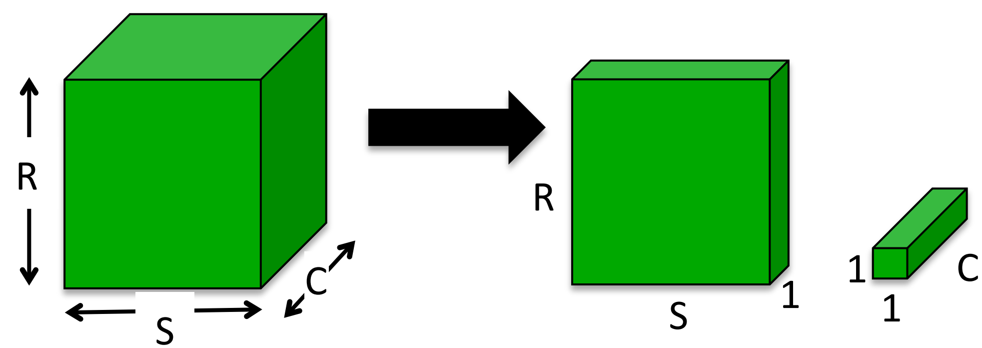
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures

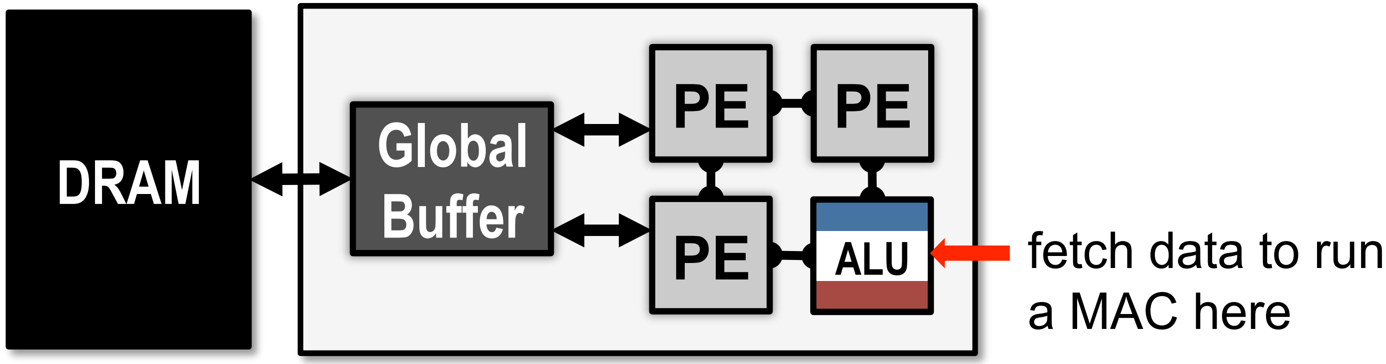


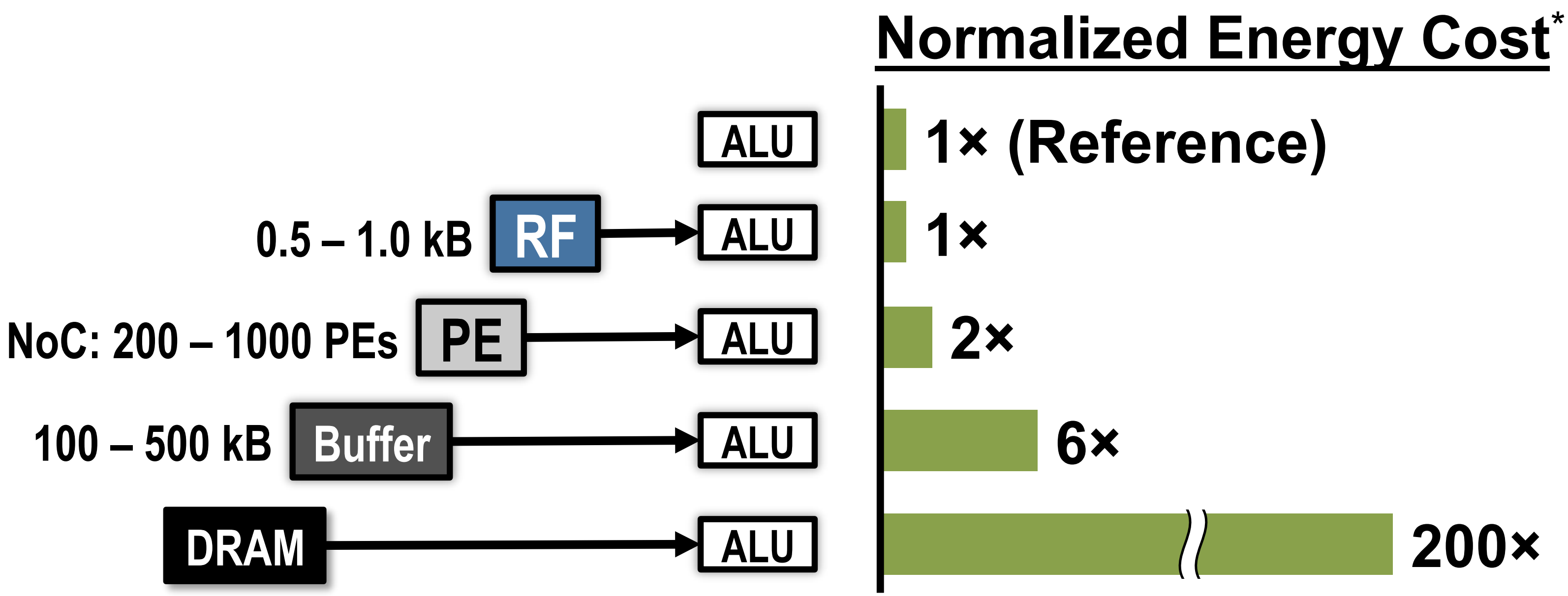
Examples: SqueezeNet, MobileNet

... also reduced precision

- Focus on reducing **number of MACs and weights**
- **Does it translate to energy savings?**

Data Movement is Expensive

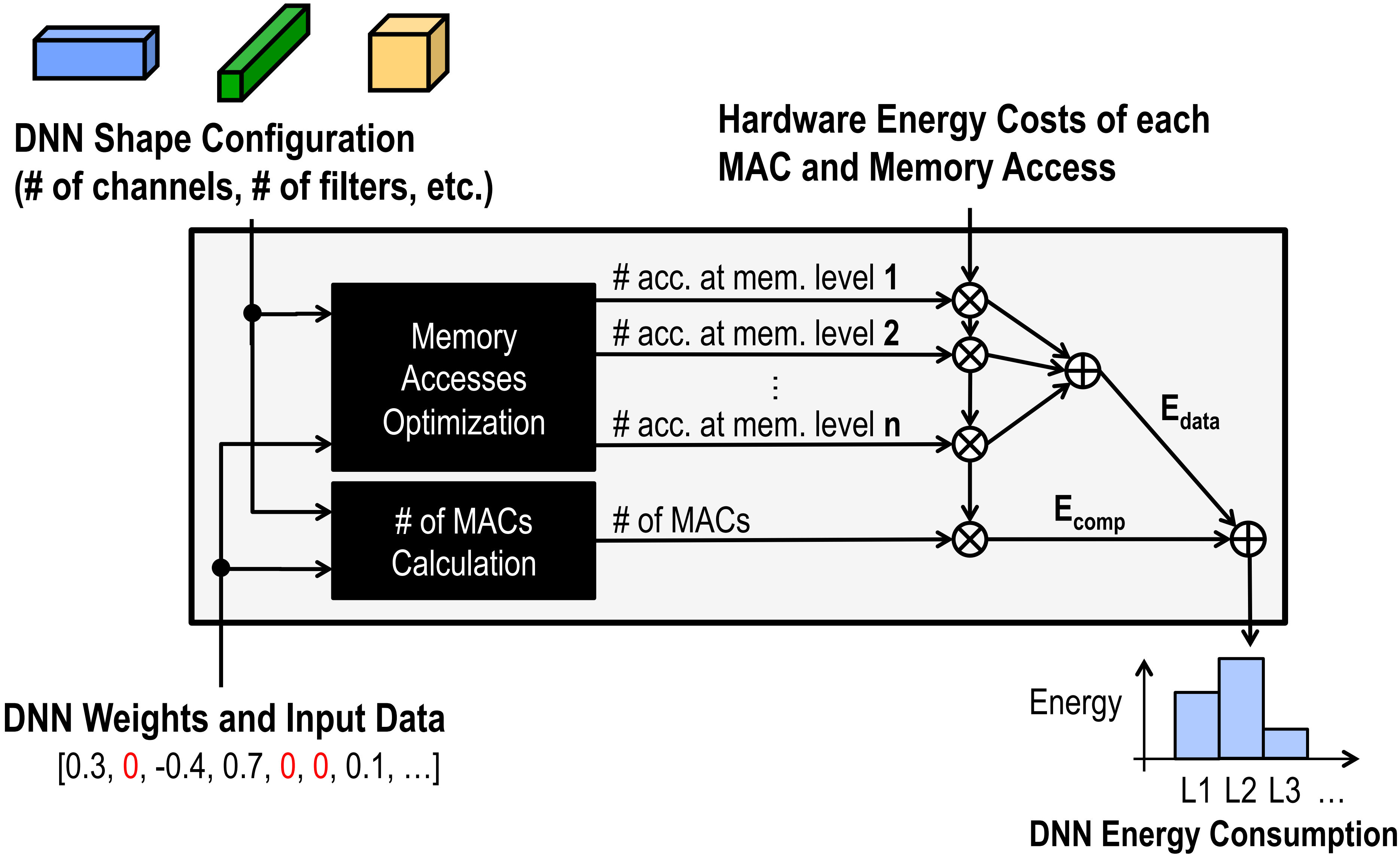




* measured from a commercial 65nm process

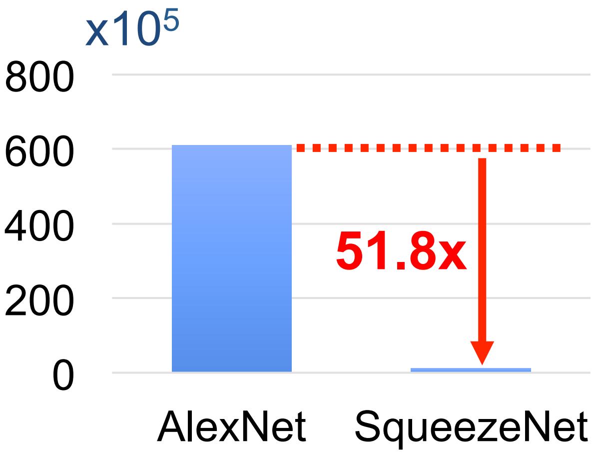
Energy of weight depends on **memory hierarchy** and **dataflow**

Energy-Evaluation Methodology

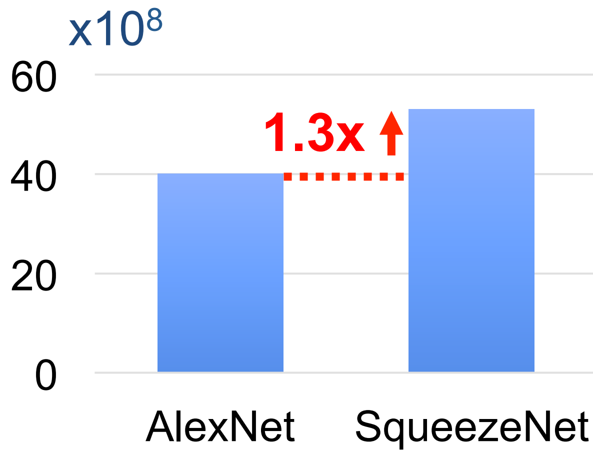


Energy estimation tool available at http://eyeriss.mit.edu

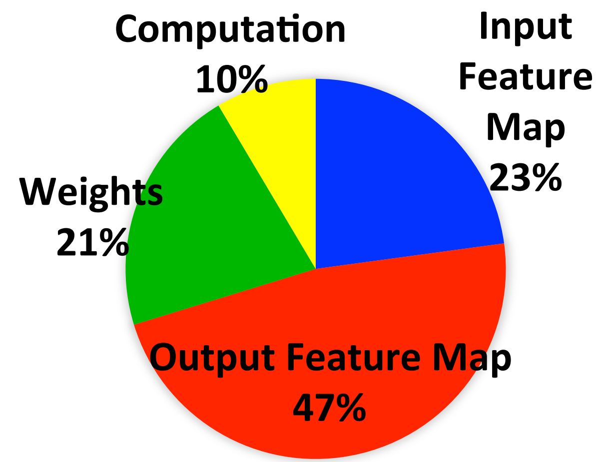
Example: AlexNet vs. SqueezeNet



# of Weights

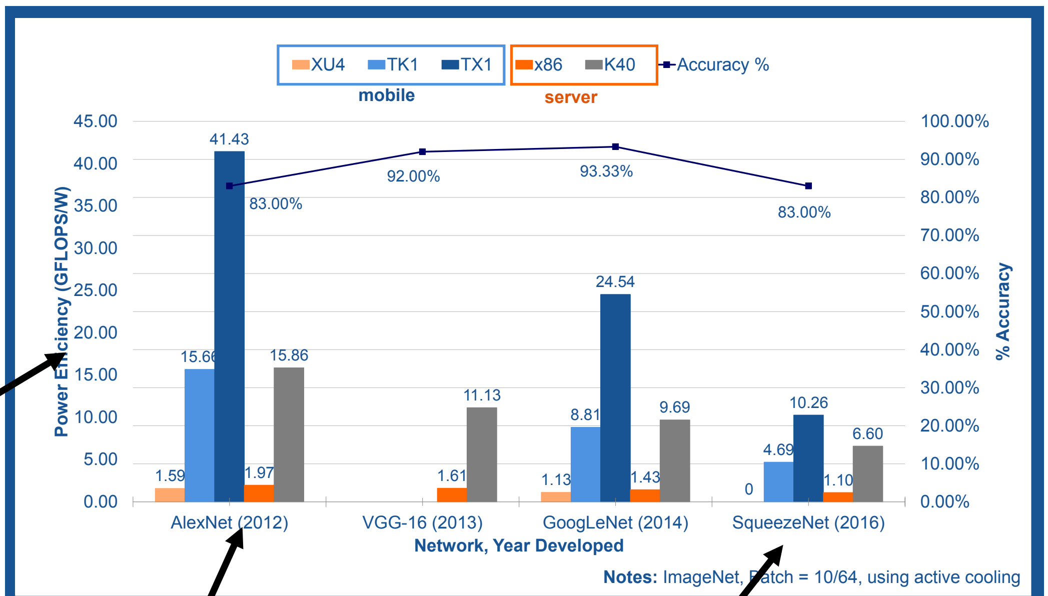


Normalized Energy



Energy Breakdown (SqueezeNet)

Results for SqueezeNet1.0 for batch size 48



Power Efficiency (GFLOPS/W)

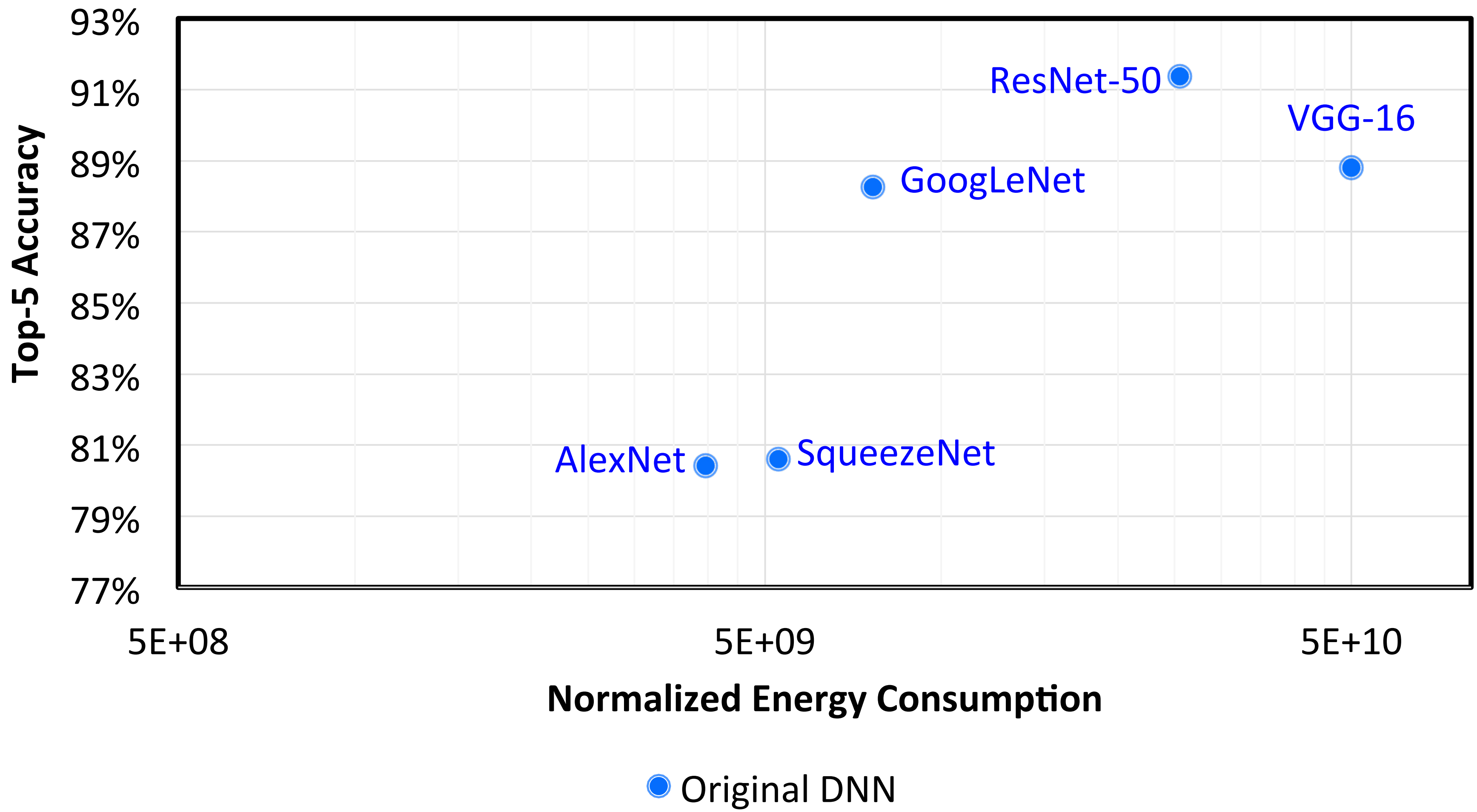
AlexNet

SqueezeNet

Number of weights *alone* is not a good metric for energy
All data types should be considered

[Movidius, Hot Chips 2016]

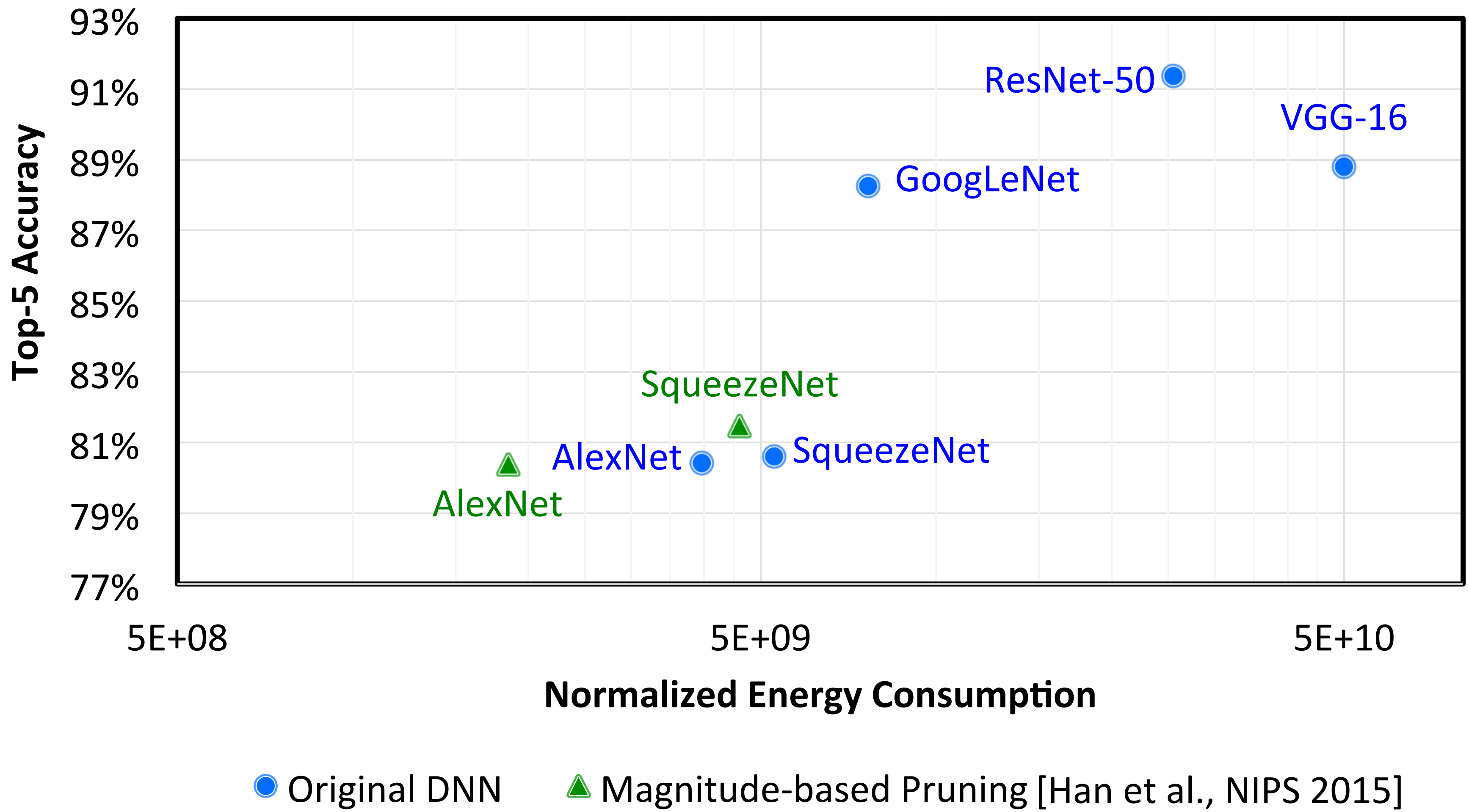
Energy Consumption of Existing DNNs



Deeper DNNs with fewer weights do not necessarily consume less energy than shallower DNNs with more weights

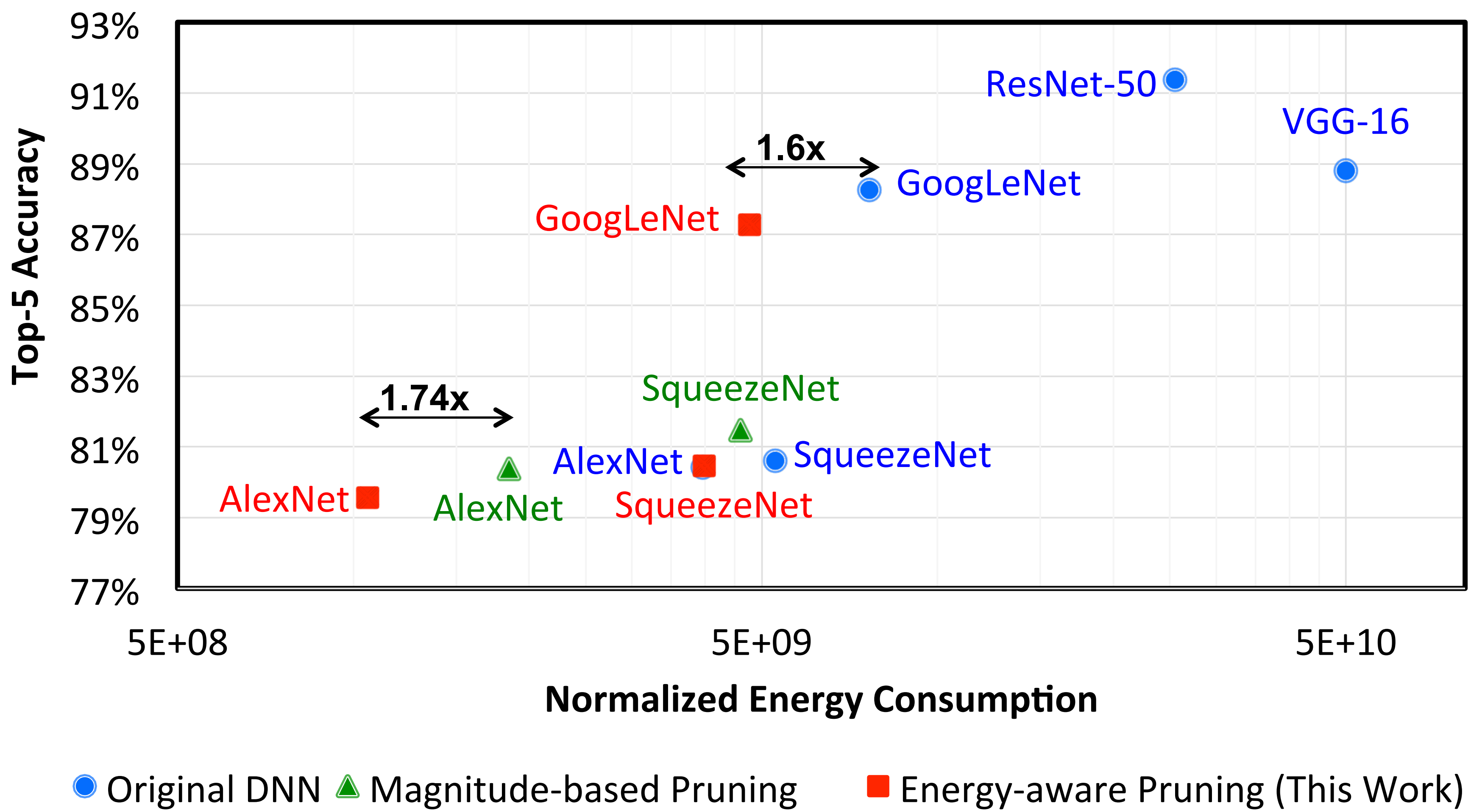
[Yang et al., CVPR 2017]

Magnitude-based Weight Pruning



Reduce number of weights by **removing small magnitude weights**

Energy-Aware Pruning

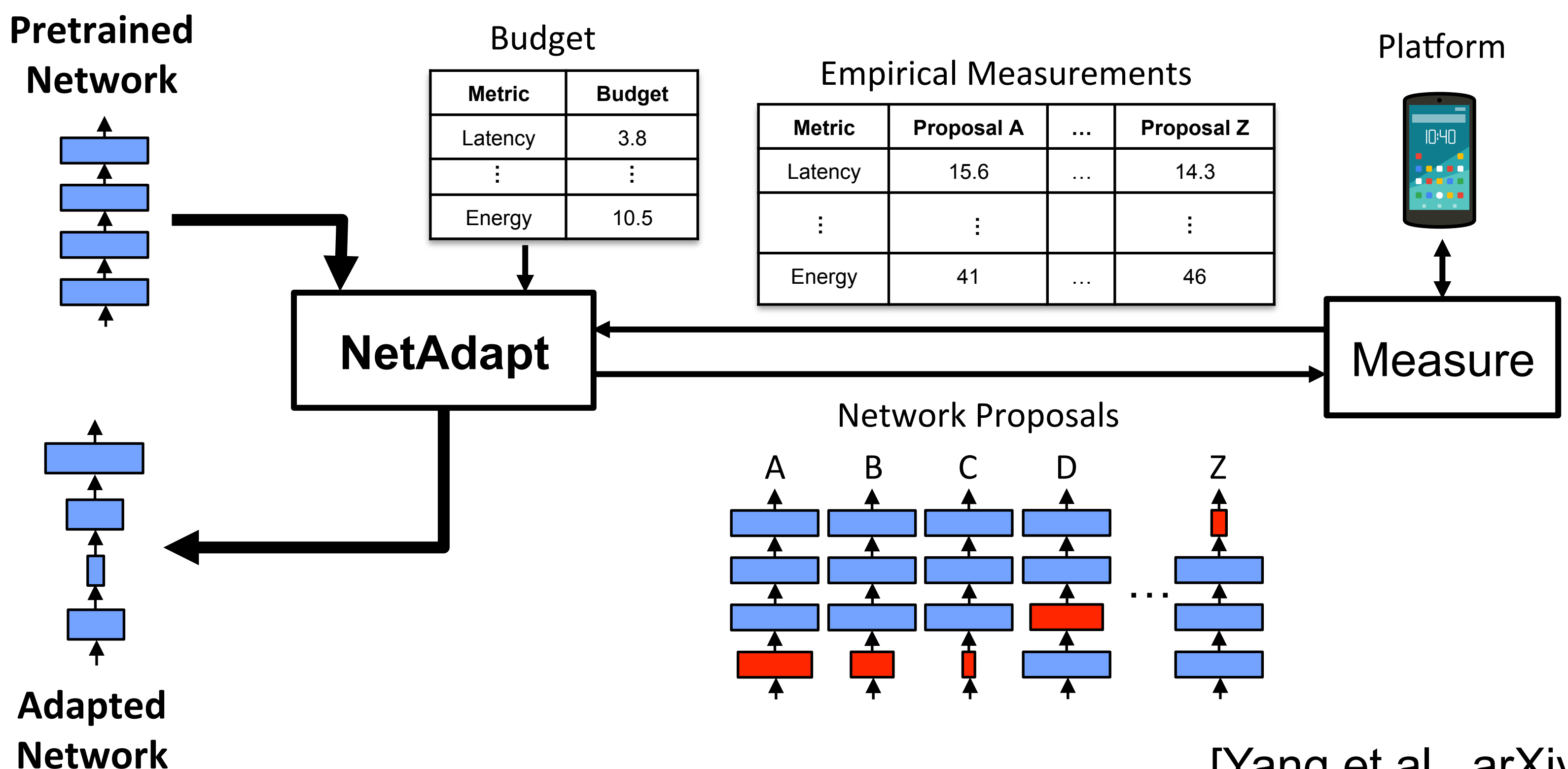


Directly target energy and incorporate it into the optimization
of DNNs to provide greater energy savings

[Yang et al., CVPR 2017]

NetAdapt: Platform-Aware DNN Adaptation

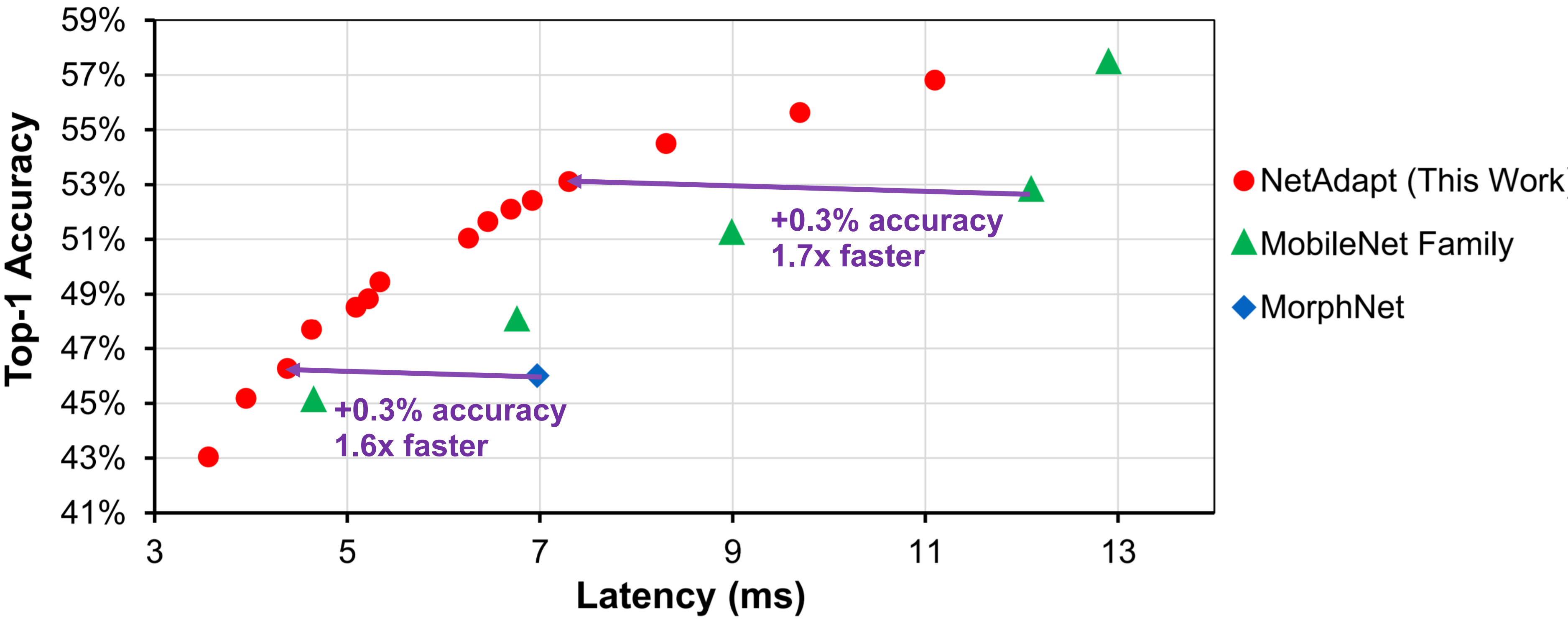
- **Automatically adapt DNN** to a mobile platform to reach a target latency or energy budget
- Use **empirical measurements** to guide optimization (avoid modeling of tool chain or platform architecture)



[Yang et al., arXiv 2018]

Improved Latency vs. Accuracy Tradeoff

- NetAdapt boosts **the real inference speed** of MobileNet by up to 1.7x with higher accuracy



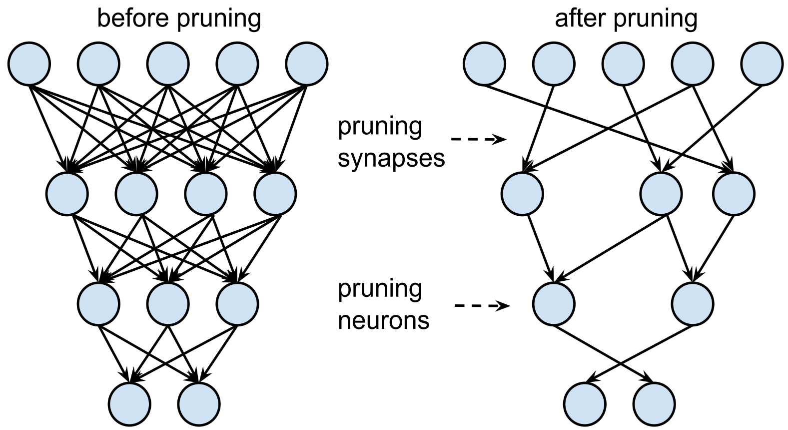
*Tested on the ImageNet dataset and a Google Pixel 1 CPU

Reference:

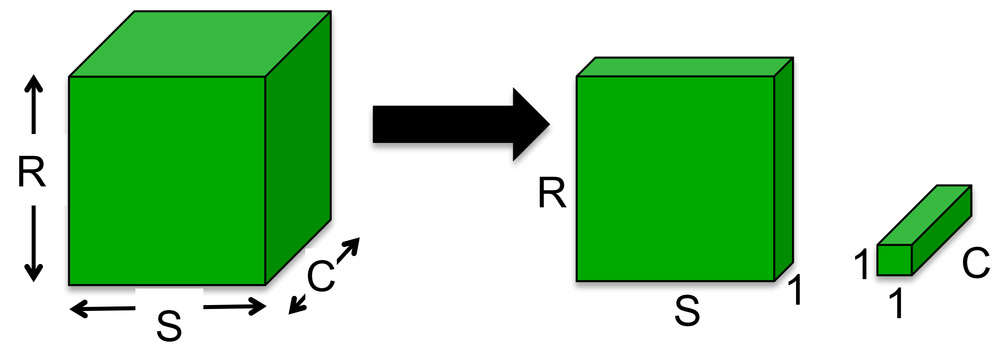
MobileNet: Howard et al, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv 2017
MorphNet: Gordon et al., "Morphnet: Fast & simple resource-constrained structure learning of deep networks", CVPR 2018

Many Efficient DNN Design Approaches

Network Pruning



Compact Network Architectures



Reduce Precision

32-bit float 10100101000000000101000000000100

8-bit fixed 01100110

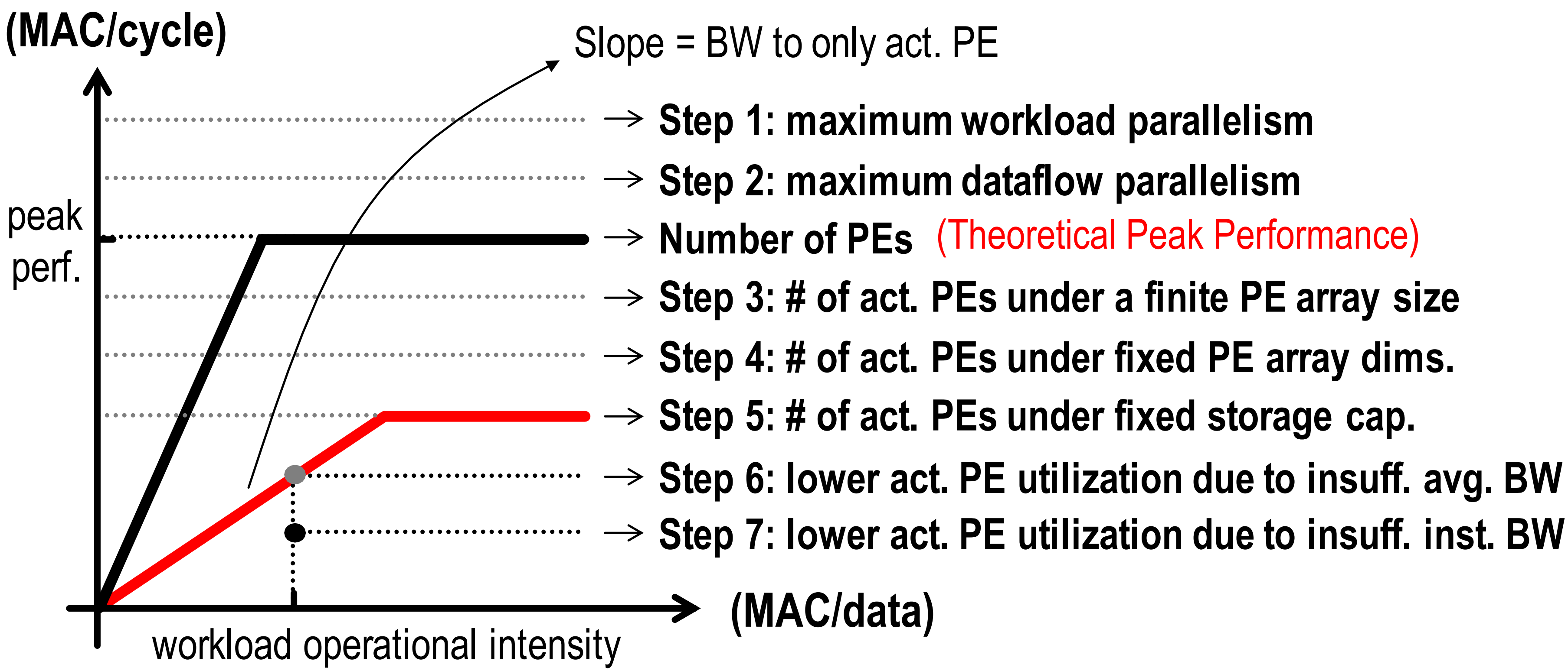
Binary 0

No guarantee that DNN algorithm designer will use a given approach.
Need flexible hardware!

Eyexam: Understanding Sources of Inefficiencies in DNN Accelerators

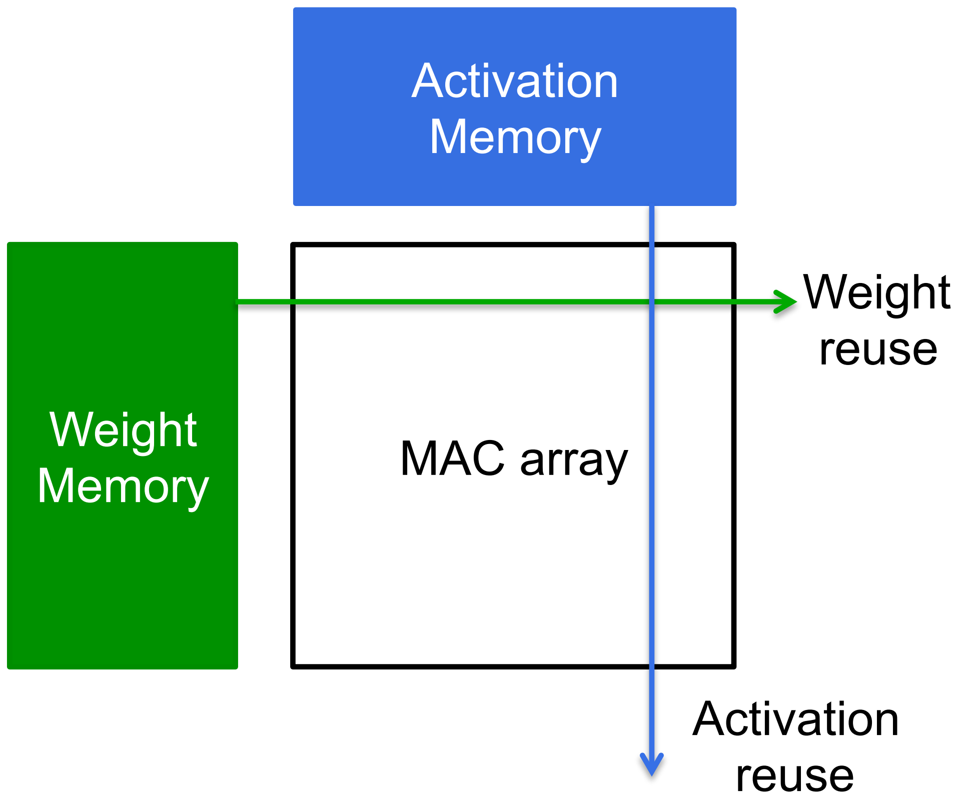
A systematic way to evaluate how each architectural decision affects performance (throughput) for a given DNN workload

Tightens the roofline model



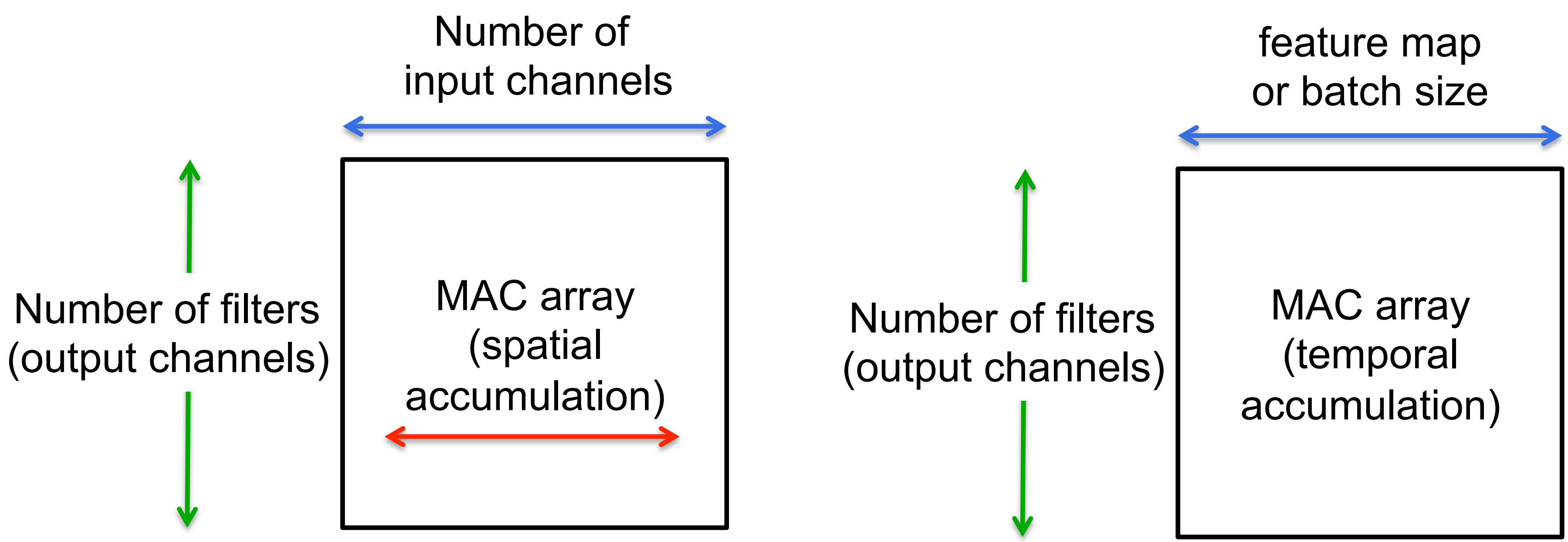
Existing DNN Architectures

- Specialized DNN hardware often rely on certain properties of DNN in order to achieve high energy-efficiency
- **Example:** Reduce memory access by amortizing across MAC array



Limitation of Existing DNN Architectures

- **Example:** reuse depends on # of channels, feature map/batch size
  – Not efficient across all network architectures (e.g., compact DNNs)
  – Can be challenging to exploit sparsity



Existing Sparse DNN Architectures

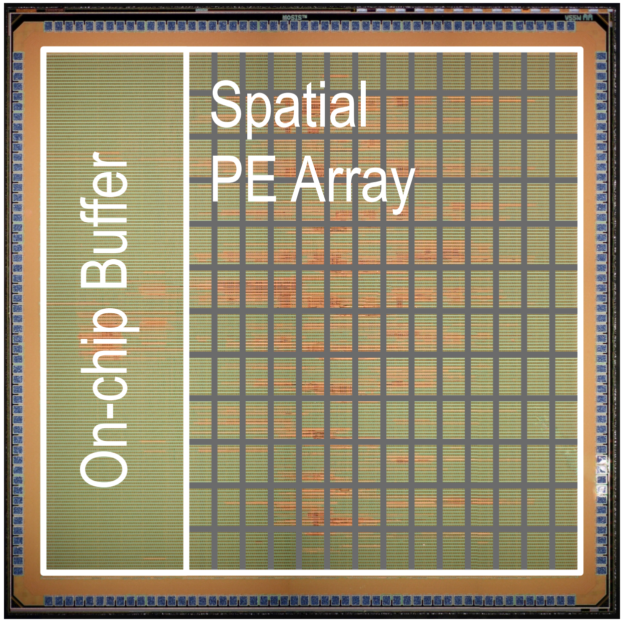
- Sparse DNN architectures translate sparsity from pruning into improved energy-efficiency and throughput
  - Perform only non-zero MACs and move data in compressed format
- Existing sparse DNN architectures optimized for either CONV or FC layer due to different BW and data reuse requirements
- Efficient for sparse DNNs, but **overhead for dense DNNs**
  - Compressed format results in **memory overhead** for dense DNNs
  - Additional control to identify location of non-zero values results in **energy overhead** for dense DNNs

Since there is **no guarantee in degree of sparsity**,
it is important to **evaluate the overhead on dense DNNs**

Goals of Eyeriss v2

To efficiently support:

- Wide range of filter shapes
  - Large **and** Compact
- Different Layers
  - e.g., CONV **and** FC
- Wide range of sparsity
  - Dense **and** Sparse



Eyeriss (v1)
[Chen et al. ISSCC 2016, ISCA 2016]
http://eyeriss.mit.edu

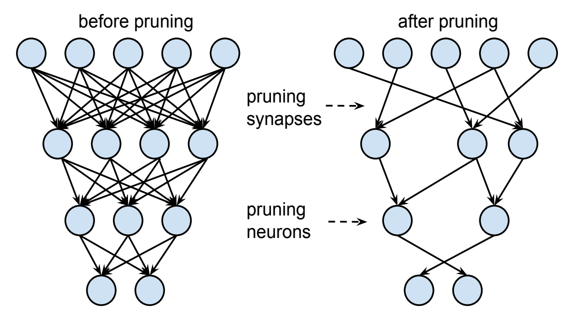
Need More Comprehensive Benchmarks

Processors should support a **diverse set of DNNs** that utilize different techniques

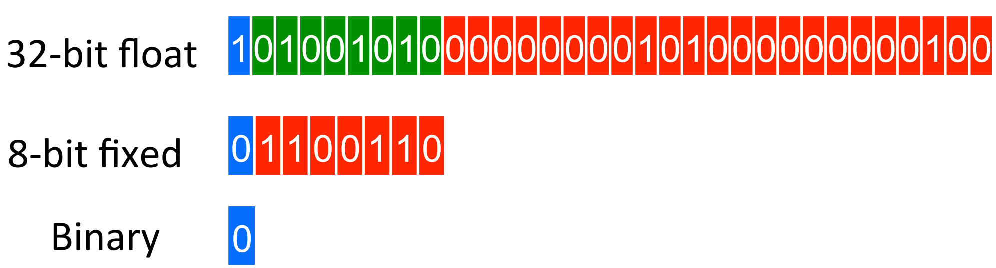
Example:

- Sparse **and** Dense
- Large **and** Compact network architectures
- Different Layers (e.g., CONV **and** FC)
- Variable Bit-width

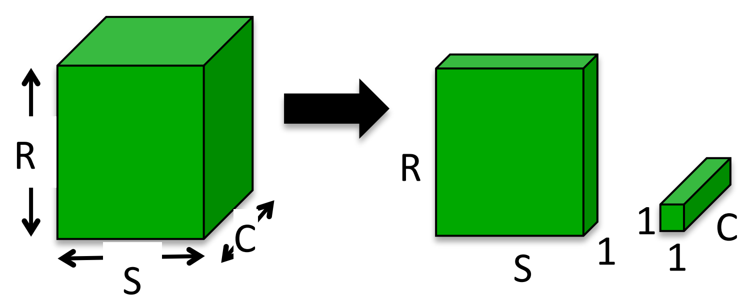
Network Pruning



Reduce Precision



Compact Network Architecture



Eyerissv2: Balancing Flexibility and Efficiency

- Flexible dataflow for high PE array utilization and data reuse for various layer shapes and sizes
- Flexible NoC that can operate in different modes for different requirements
  - Utilizes multicast to exploit spatial data reuse
  - Utilizes unicast for high BW for weights for FC and weights & activations for compact network architectures
- Processes data in both compressed and raw format to minimize data movement for both CONV and FC layers
  - Exploit sparsity in weights and activations

Benchmarking Metrics for DNN Hardware

How can we compare designs?

V. Sze, Y.-H. Chen, T-J. Yang, J. Emer,
"***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***,"
Proceedings of the IEEE, Dec. 2017

Metrics for DNN Hardware

- **Accuracy**
  - Quality of result for a given task
- **Throughput**
  - Analytics on high volume data
  - Real-time performance (e.g., video at 30 fps)
- **Latency**
  - For interactive applications (e.g., autonomous navigation)
- **Energy and Power**
  - Edge and embedded devices have limited battery capacity
  - Data centers have stringent power ceilings due to cooling costs
- **Hardware Cost**
  - $$$

Specifications to Evaluate Metrics

- **Accuracy**
  - Difficulty of dataset and/or task should be considered
- **Throughput**
  - Number of cores (include utilization along with peak performance)
  - Runtime for running specific DNN models
- **Latency**
  - Include batch size used in evaluation
- **Energy and Power**
  - Power consumption for running specific DNN models
  - Include external memory access
- **Hardware Cost**
  - On-chip storage, number of cores, chip area + process technology

Example: Metrics of Eyeriss Chip

ASIC Specs	Input
Process Technology	65nm LP TSMC (1.0V)
Total Core Area (mm^2)	12.25
Total On-Chip Memory (kB)	192
Number of Multipliers	168
Clock Frequency (MHz)	200
Core area (mm^2) / multiplier	0.073
On-Chip memory (kB) / multiplier	1.14
Measured or Simulated	Measured

Metric	Units	Input
Name of CNN Model	Text	AlexNet
Top-5 error classification on ImageNet	#	19.8
Supported Layers		All CONV
Bits per weight	#	16
Bits per input activation	#	16
Batch Size	#	4
Runtime	ms	115.3
Power	mW	278
Off-chip Access per Image Inference	MBytes	3.85
Number of Images Tested	#	100

Comprehensive Coverage

- **All metrics** should be reported for fair evaluation of design tradeoffs
- Examples of what can happen if certain metric is omitted:
  - **Without the accuracy given for a specific dataset and task**, one could run a simple DNN and claim low power, high throughput, and low cost – however, the processor might not be usable for a meaningful task
  - **Without reporting the off-chip bandwidth**, one could build a processor with only multipliers and claim low cost, high throughput, high accuracy, and low chip power – however, when evaluating system power, the off-chip memory access would be substantial
- Are results measured or simulated? On what test data?

Evaluation Process

The evaluation process for whether a DNN system is a viable solution for a given application might go as follows:

1. **Accuracy** determines if it can perform the given task
2. **Latency and throughput** determine if it can run fast enough and in real-time
3. **Energy and power consumption** will primarily dictate the form factor of the device where the processing can operate
4. **Cost**, which is primarily dictated by the chip area, determines how much one would pay for this solution

Summary

- The number of weights and MACs are not sufficient for evaluating the energy consumption and latency of DNNs
  - Designers of efficient DNN algorithms should directly target direct metrics such as energy and latency and incorporate that into their design
- Many of the existing DNN processors rely on certain properties of the DNN which cannot be guaranteed as the wide range techniques used for efficient DNN algorithm design has resulted in a more diverse set of DNNs
  - DNN hardware used to process these DNNs should be sufficiently flexible to support a wide range of techniques efficiently
- DNN hardware should be evaluated on a comprehensive set of benchmarks and metrics

For updates on Eyerissv2, Eyexam, NetAdapt, etc.

Follow @eems_mit **or join EEMS news mailing list**

Acknowledgements

Research conducted in the **MIT Energy-Efficient Multimedia Systems Group** would not be possible without the support of the following organizations:

AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
NSF
ANALOG DEVICES
BROADCOM
Google
intel
NVIDIA
DARPA
SRC
3M
QUALCOMM
SAMSUNG
TEXAS INSTRUMENTS
tsmc