Efficient Computing for AI and Robotics

Vivienne Sze
Massachusetts Institute of Technology

Contact Info
email: sze@mit.edu
website: www.rle.mit.edu/eems
Processing at “Edge” instead of the “Cloud”

Communication  Privacy  Latency
Camera and radar generate ~6 gigabytes of data every 30 seconds.

Self-driving car prototypes use approximately 2,500 Watts of computing power.

Generates wasted heat and some prototypes need water-cooling!
Existing Processors Consume Too Much Power

< 1 Watt

> 10 Watts
Transistors Are Not Getting More Efficient

Slowdown of Moore’s Law and Dennard Scaling

General purpose microprocessors not getting faster or more efficient

Need **specialized hardware** for significant improvements in speed and energy efficiency
Energy-Efficient AI with Cross-Layer Design

**Algorithms**
- Convolutions
- Pooling
- Convs
- Linear Classifier
- Object Categories / Positions
- F4 maps
- S2 feature maps
- C1 feature maps
- C3 feature maps

**Systems**

**Architectures**
- Link Clock
- Core Clock
- DCNN Accelerator
- 14x12 PE Array
- Off-Chip DRAM
- 64 bits

**Circuits**
## Power Dominated by Data Movement

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (pJ)</th>
<th>Relative Energy Cost</th>
<th>Area (µm²)</th>
<th>Relative Area Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>8b Add</td>
<td>0.03</td>
<td>1</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>16b Add</td>
<td>0.05</td>
<td>10</td>
<td>67</td>
<td>10</td>
</tr>
<tr>
<td>32b Add</td>
<td>0.1</td>
<td>100</td>
<td>137</td>
<td>100</td>
</tr>
<tr>
<td>16b FP Add</td>
<td>0.4</td>
<td>1000</td>
<td>1360</td>
<td>1000</td>
</tr>
<tr>
<td>32b FP Add</td>
<td>0.9</td>
<td>1,000</td>
<td>4184</td>
<td>1,000</td>
</tr>
<tr>
<td>8b Mult</td>
<td>0.2</td>
<td>10,000</td>
<td>282</td>
<td>10,000</td>
</tr>
<tr>
<td>32b Mult</td>
<td>3.1</td>
<td>100,000</td>
<td>3495</td>
<td>100,000</td>
</tr>
<tr>
<td>16b FP Mult</td>
<td>1.1</td>
<td>1,000,000</td>
<td>1640</td>
<td>1,000,000</td>
</tr>
<tr>
<td>32b FP Mult</td>
<td>3.7</td>
<td>10,000,000</td>
<td>7700</td>
<td>10,000,000</td>
</tr>
<tr>
<td>32b SRAM Read (8KB)</td>
<td>5</td>
<td>100,000,000</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>32b DRAM Read</td>
<td>640</td>
<td>100,000,000,000</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Memory access is **orders of magnitude** higher energy than compute

[Horowitz, ISSCC 2014]
Autonomous Navigation Uses a Lot of Data

Semantic Understanding

- High frame rate
- Large resolutions
- Data expansion

Geometric Understanding

- Growing map size
Visual-Inertial Localization

Determines location/orientation of robot from images and IMU

*Subset of SLAM algorithm (Simultaneous Localization And Mapping)
Localization at under 25 mW

First chip that performs complete Visual-Inertial Odometry

Front-End for camera
(Feature detection, tracking, and outlier elimination)

Front-End for IMU
(pre-integration of accelerometer and gyroscope data)

Back-End Optimization of Pose Graph

Consumes $684\times$ and $1582\times$ less energy than mobile and desktop CPUs, respectively

[Joint work with Sertac Karaman (AeroAstro)]

http://navion.mit.edu

[Zhang et al., RSS 2017], [Suleiman et al., VLSI 2018]
Key Methods to Reduce Data Size

**Navion:** Fully integrated system – no off-chip processing or storage

![Diagram of Key Methods to Reduce Data Size](image)

- **Apply Low Cost Frame Compression**
- **Exploit Sparsity in Graph and Linear Solver**

Use **compression** and **exploit sparsity** to reduce memory down to 854kB

[Suleiman et al., VLSI 2018] Best Student Paper Award
Deep Neural Networks

Deep Neural Networks (DNNs) have become a cornerstone of AI

Computer Vision

Speech Recognition

Game Play

Medical
DNNs for Understanding the Environment

Depth Estimation

State-of-the-art approaches use Deep Neural Networks which require up to several hundred millions of operations and weights to compute! >100x more complex than video compression
Properties We Can Leverage

- Operations exhibit **high parallelism** → **high throughput** possible
- Memory Access is the Bottleneck

Memory Read | MAC* | Memory Write
---|---|---
DRAM | ALU | DRAM
filter weight | partial sum | updated partial sum
image pixel | multiply-and-accumulate
partial sum

**Worst Case:** all memory R/W are **DRAM** accesses

- Example: AlexNet has **724M** MACs → **2896M** DRAM accesses required
Properties We Can Leverage

- Operations exhibit **high parallelism** → **high throughput** possible

- **Input data reuse** opportunities (**up to 500x**)
Exploit Data Reuse at Low-Cost Memories

Specialized hardware with small (<1kB) low cost memory near compute

Normalized Energy Cost*

1× (Reference)

1×

2×

6×

200×

* measured from a commercial 65nm process

Farther and larger memories consume more power
Deep Neural Networks at under 0.3 W

Exploits data reuse for \(100\times\) reduction in memory accesses from global buffer and \(1400\times\) reduction in memory accesses from off-chip DRAM

Overall \textbf{\textgreater}10\textbf{x} energy reduction compared to a mobile GPU

[Chen et al., ISSCC 2016, ISCA 2016]  Micro Top Picks Award  http://eyeriss.mit.edu
Features: Energy vs. Accuracy

![Graph showing Energy vs. Accuracy](image)

Exponential

Energy/Pixel (nJ)

- VGG16
- AlexNet
- HOG

Measured in 65nm*

- 1. DPM v5 [Girshick, 2012]

Accuracy (Average Precision)

Measured in on VOC 2007 Dataset
1. DPM v5 [Girshick, 2012]

* Only feature extraction. Does not include data, classification energy, augmentation and ensemble, etc.

[Suleiman et al., ISCAS 2017]
A significant amount of algorithm and hardware research on energy-efficient processing of DNNs

We identified various limitations to existing approaches.
**Design of Efficient DNN Algorithms**

- Popular efficient DNN algorithm approaches

**Network Pruning**

- Before pruning
- After pruning
- Pruning synapses
- Pruning neurons

**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

Energy of weight depends on memory hierarchy and dataflow

*measured from a commercial 65nm process
Energy-Evaluation Methodology

DNN Shape Configuration
(# of channels, # of filters, etc.)

DNN Weights and Input Data

[0.3, 0, -0.4, 0.7, 0, 0, 0.1, …]

Hardware Energy Costs of each MAC and Memory Access

Memory Accesses Optimization

# of MACs Calculation

# acc. at mem. level 1
# acc. at mem. level 2
⋯
# acc. at mem. level n

# of MACs

E_{comp}

E_{data}

Energy

DNN Energy Consumption

Tool available at: https://energyestimation.mit.edu/

[Yang et al., CVPR 2017]
Key Observations

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

Energy Consumption of GoogLeNet

- Output Feature Map: 43%
- Input Feature Map: 25%
- Weights: 22%
- Computation: 10%

[Yang et al., CVPR 2017]
Directly target energy and incorporate it into the optimization of DNNs to provide greater energy savings

- Sort layers based on energy and prune layers that consume most energy first
- EAP reduces AlexNet energy by **3.7x** and outperforms the previous work that uses magnitude-based pruning by **1.7x**

Pruned models available at [http://eyeriss.mit.edu/energy.html](http://eyeriss.mit.edu/energy.html)

[Yang et al., CVPR 2017]
# of Operations vs. Latency

- # of operations (MACs) does not approximate latency well

**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)

---

**Budget**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>3.8</td>
</tr>
<tr>
<td>Energy</td>
<td>10.5</td>
</tr>
</tbody>
</table>

**Empirical Measurements**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Proposal A</th>
<th>...</th>
<th>Proposal Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>15.6</td>
<td>...</td>
<td>14.3</td>
</tr>
<tr>
<td>Energy</td>
<td>41</td>
<td>...</td>
<td>46</td>
</tr>
</tbody>
</table>

**Network Proposals**

A, B, C, D, Z

---

Code available at [http://netadapt.mit.edu](http://netadapt.mit.edu)  

[Yang et al., ECCV 2018]

---

*In collaboration with Google’s Mobile Vision Team*
NetAdapt boosts **the real inference speed** of MobileNet by up to 1.7x with higher accuracy.

**Reference:**

- **MorphNet:** Gordon et al., “Morphnet: Fast & simple resource-constrained structure learning of deep networks”, CVPR 2018

*Tested on the ImageNet dataset and a Google Pixel 1 CPU*
FastDepth: Fast Monocular Depth Estimation

Depth estimation from a single RGB image desirable, due to the relatively low cost and size of monocular cameras.

RGB

Prediction

Auto Encoder DNN Architecture (Dense Output)

[Joint work with Sertac Karaman]
Apply NetAdapt, compact network design, and depth wise decomposition to decoder layer to enable depth estimation at high frame rates on an embedded platform while still maintaining accuracy.

Configuration: Batch size of one (32-bit float)

Models available at [http://fastdepth.mit.edu](http://fastdepth.mit.edu)  [Wofk*, Ma* et al., ICRA 2019]
Monitoring Neurodegenerative Disorders

Dementia affects 50 million people worldwide today (75 million in 10 years) [World Alzheimer’s Report]

Mini-Mental State Examination (MMSE)

Q1. What is the year? Season? Date?
Q2. Where are you now? State? Floor?
Q3. Could you count backward from 100 by sevens? (93, 86, ...)

Clock-drawing test


• Neuropsychological assessments are **time consuming** and require a trained specialist
• Repeat medical assessments are **sparse**, mostly **qualitative**, and suffer from **high retest variability**

[Joint work with Thomas Heldt (IMES)]
Eye movements can be used to quantitatively evaluate severity, progression or regression of neurodegenerative diseases.

Clinical measurements of saccade latency are done in constrained environments that rely on specialized, costly equipment.

High-speed camera: Phantom v25-11

Substantial head support: SR EYLEINK 1000 PLUS

Measure Eye Movements Using Phone

Develop algorithm to measure eye movement using a **consumer-grade camera** rather than high-cost research-grade camera.

*Enable low-cost in-home longitudinal measurements.*

[Saavedra Peña et al., EMBC 2018] [Lai et al., ICIP 2018]
Summary

• Energy-Efficient AI extends the reach of AI beyond the cloud by **reducing communication requirements**, **enabling privacy**, and **providing low latency** so that AI can be used in wide range of applications ranging from robotics to health care.

• **Cross-layer design with specialized hardware** enables energy-efficient AI, and will be critical to the progress of AI over the next decade.

Today’s slides available at [www.rle.mit.edu/eems](http://www.rle.mit.edu/eems)
Acknowledgements

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

Today’s slides available at [www.rle.mit.edu/eems](http://www.rle.mit.edu/eems)
Additional Resources

Overview Paper

Book Coming Soon!

More info about Tutorial on DNN Architectures
http://eyeriss.mit.edu/tutorial.html

MIT Professional Education Course on “Designing Efficient Deep Learning Systems”
http://professional-education.mit.edu/deeplearning

For updates  
Follow @eems_mit

http://mailman.mit.edu/mailman/listinfo/eems-news
References

• Energy-Efficient Hardware for Deep Neural Networks
  – Project website: http://eyeriss.mit.edu

• Limitations of Existing Efficient DNN Approaches
  – Hardware Architecture for Deep Neural Networks: http://eyeriss.mit.edu/tutorial.html
References

• Co-Design of Algorithms and Hardware for Deep Neural Networks

• Energy-Efficient Visual Inertial Localization
  – Project website: [http://navion.mit.edu](http://navion.mit.edu)
References

• Monitoring Neurodegenerative Disorders Using a Phone