How to Estimate the Energy Consumption of Deep Neural Networks

Tien-Ju Yang, Yu-Hsin Chen, Joel Emer, Vivienne Sze
MIT
Problem of DNNs

Energy Estimation helps

- Understand the design trade-off
- Guide the DNN design
- Enable DNN mobile applications

Computation

- DNN
  - 15k – 300k OP/Px

- DPM
  - 0.1k – 0.5k OP/Px

Accuracy
Deep Convolutional NN Explanation

Modern Deep CNN: 5 – 152 Layers

- **Convolution**
- **Activation**
- **Norm.**
- **Pooling**

CNN Layer → Low-Level Features → CNN Layer → Mid-Level Features → CNN Layer → High-Level Features
Deep Convolutional NN Explanation

Modern **Deep CNN**: 5 – 152 Layers

- **CNN Layer** → **Low-Level Features** → **CNN Layer** → **Mid-Level Features** → **CNN Layer** → **High-Level Features**

Convolution

Takes **90% – 99% of Computation**
Convolution

Input Image (Feature Map)

Filter

\[
\begin{bmatrix}
R \\
R
\end{bmatrix} \ast \\
\begin{bmatrix}
H \\
H
\end{bmatrix}
\]
Convolution

Input Image (Feature Map)

Filter

Element-wise Multiplication
Convolution

Element-wise Multiplication

Partial Sum (psum) Accumulation

Input Image (Feature Map)       Output Image
Convolution

Filter

Input Image (Feature Map)  Output Image

Sliding Window Processing

\[ \text{Input Image (Feature Map)} \ast \text{Filter} = \text{Output Image} \]
Convolution

Input Image (Feature Map)  Output Image

Filter

Sliding Window Processing
Why Not Use # of Weights or MACs?

MAC*

filter weight
image pixel
partial sum

ALU

updated partial sum

* multiplication-and-accumulation

Reason 1:
Reason 2:
Why Not Use # of Weights or MACs?

Reason 1:  
Reason 2:  

Normalized Energy Cost*  

1× (Reference)  

200×  

* measured from a commercial 65nm process
Why Not Use # of Weights or MACs?

Memory Read | MAC | Memory Write

DRAM | ALU | DRAM

Reason 1: computation is cheap but **data movement** is expensive

Reason 2:

Normalized Energy Cost*

1× (Reference)

DRAM → ALU

200×

* measured from a commercial 65nm process
Why Not Use # of Weights or MACs?

Memory Read  |  MAC  | Memory Write

DRAM → Mem → ALU → Mem → DRAM

Extra levels of local memory hierarchy

Reason 1: computation is cheap but data movement is expensive

Reason 2:

Normalized Energy Cost*

- 1× (Reference)
- 1×
- 2×
- 6×
- 200×

* measured from a commercial 65nm process
Why Not Use # of Weights or MACs?

**Reason 1:** computation is cheap but **data movement** is expensive

**Reason 2:** where data come from/go to is important for energy

Extra levels of local memory hierarchy

<table>
<thead>
<tr>
<th>Normalized Energy Cost*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1× (Reference)</td>
</tr>
<tr>
<td>1×</td>
</tr>
<tr>
<td>2×</td>
</tr>
<tr>
<td>6×</td>
</tr>
<tr>
<td>200×</td>
</tr>
</tbody>
</table>

* measured from a commercial 65nm process
Energy Estimation Methodology
Energy Estimation Methodology

- Estimate the energy consumption of each layer separately.

For each layer, $E_{layer} = E_{comp} + E_{data}$

Computation energy only depends on the # of MACs.
Energy Estimation Methodology

- Estimate the energy consumption of each layer separately
- For each layer, $E_{layer} = E_{comp} + E_{data}$

Minimize energy consumption under the hardware constraints
Energy Estimation Methodology

- Estimate the energy consumption of each layer separately

For each layer, \( E_{\text{layer}} = E_{\text{comp}} + E_{\text{data}} \)

Data energy does NOT only depend on the # of MACs

Computation energy only depends on the # of MACs
Factor in Bitwidth

Data energy:
- Consider bitwidths in the optimization
- Scale # of bits linearly with the bitwidth
Factor in Sparsity

Apply Non-Linearity **ReLU** on Filtered Image Data

![Diagram of ReLU application on fmap and filtered data]

Pruned Network Filters

![Diagram of Prune operation on fmap and filter]

**Prune** = 6
Factor in Sparsity

Use data compression to reduce the # of bits accessed

- **Shape Configuration**
  - (# of channels and filters, etc.)

- **Weights and Input Data**
  - [0.3, 0, -0.4, 0.7, 0, 0, 0.1, ...]

- **Memory Access Optimization**

- **# of MACs Calculation**

- **Real Hardware Energy Costs of Each MAC and Memory Access**

- **Energy**
  - Network Energy Consumption
  - Layer Energy Consumption

- **Skip the MAC when at least one input is zero**

- **0 Skipped!**
Insights
Example Platform

Eyeriss [ISSCC, 2016]
A reconfigurable CNN processor

35 fps @ 278 mW*
Key Insights

Convolutional layers consume more energy than fully-connected layers

4% ← Weight → 96%

72% ← Energy → 28%

AlexNet

Input Fmap  Output Fmap  Weight  Computation
Key Insights

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

Key Insights

- Data movement is more expensive than computation
- Feature maps need to be taken into account

GoogLeNet Energy Breakdown
Application
Energy-Aware Pruning (EAP)

- Use estimated energy to guide the layer-by-layer pruning
- Start from pruning the layers that consume the most of energy

Energy-Aware Pruning (EAP)

We remove the weights having the smallest joint impact on the output instead of the small magnitude weights.

**Magnitude-based Method**

**Our Method**

$$\text{fmap} \quad \ast \quad \text{filter}$$

Error: 3

$$= 6 \rightarrow 3$$

Error: 0

$$= 6$$
Pruned Result Analysis

- EAP reduces AlexNet energy by 3.7x and outperforms the previous work by 1.7x
- Energy is more difficult to reduce than # of weights and MACs

**Normalized Energy**

<table>
<thead>
<tr>
<th></th>
<th>Ori.</th>
<th>DC</th>
<th>EAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.1x 3.7x

**# of NZ MACs**

<table>
<thead>
<tr>
<th></th>
<th>Ori.</th>
<th>DC</th>
<th>EAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NZ MACs</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.0x 6.6x

**# of NZ Weights**

<table>
<thead>
<tr>
<th></th>
<th>Ori.</th>
<th>DC</th>
<th>EAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NZ Weights</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.0x 10.6x

**DC**: S. Han et al., “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,” in ICLR, 2016.
• Energy-aware pruning achieves better trade-off

Network Comparison

Better
Summary

• We proposed an energy estimation methodology of DNNs based on the architecture, bitwidth and sparsity

• We showed that
  – # of weights and MACs are not good metrics for energy
  – data movement is more expensive than computation
  – feature maps need to be taken into account

• Better accuracy-energy trade-off can be achieved by combining the energy estimation methodology with pruning
Thank You

Learn more about energy-aware pruning at http://eyeriss.mit.edu/energy.html

Learn more about efficient neural networks at https://arxiv.org/abs/1703.09039