EIE: Efficient Inference Engine on Compressed Deep Neural Network

Song Han*, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark Horowitz, Bill Dally

Stanford University
June 20, 2016
Deep Learning on Mobile

Phones

Drones

Robots

Glasses

Self Driving Cars

Battery Constrained!
Accurate Prediction => Large Model => More Memory Reference => High Power

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy [pJ]</th>
<th>Relative Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 bit int ADD</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>32 bit float ADD</td>
<td>0.9</td>
<td>9</td>
</tr>
<tr>
<td>32 bit Register File</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>32 bit int MULT</td>
<td>3.1</td>
<td>31</td>
</tr>
<tr>
<td>32 bit float MULT</td>
<td>3.7</td>
<td>37</td>
</tr>
<tr>
<td>32 bit SRAM Cache</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>32 bit DRAM Memory</td>
<td>640</td>
<td>6400</td>
</tr>
</tbody>
</table>

Relative Energy Cost

1 10 100 1000 10000

Deeper Learning on Mobile: Difficulty?

Model Size!
Problem 1: DNN Model Size too Large
Solution 1: Deep Compression

Our Past Work: Deep Compression

Smaller Size
90% zeros in weights
4-bit weight

Accuracy
No loss of accuracy / Improved accuracy

On-chip
State-of-the-art DNN fit on-chip SRAM
Our Past Work: Deep Compression

• **Network Pruning [1]:**
  10x fewer weights

  60M weights

  6M weights

• **Weight Sharing [2]:**
  only 4-bits per remaining weight

[1]. Han et al. NIPS 2015
[2]. Han et al. ICLR 2016, best paper award
## Deep Compression Results

<table>
<thead>
<tr>
<th>Network</th>
<th>Original Size</th>
<th>Compressed Size</th>
<th>Compression Ratio</th>
<th>Original Accuracy</th>
<th>Compressed Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>240MB</td>
<td>6.9MB</td>
<td>35x</td>
<td>80.27%</td>
<td>80.30%</td>
</tr>
<tr>
<td>VGGNet</td>
<td>550MB</td>
<td>11.3MB</td>
<td>49x</td>
<td>88.68%</td>
<td>89.09%</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>28MB</td>
<td>2.8MB</td>
<td>10x</td>
<td>88.90%</td>
<td>88.92%</td>
</tr>
<tr>
<td>SqueezeNet</td>
<td>4.8MB</td>
<td>0.47MB</td>
<td>10x</td>
<td>80.32%</td>
<td>80.35%</td>
</tr>
</tbody>
</table>

- No loss of accuracy on ImageNet dataset.
- Weights fits on-chip SRAM, taking 120x less energy than DRAM.
EIE: First Accelerator for Compressed Sparse Neural Network

Problem 2: Irregular Computation Pattern
Solution 2: EIE accelerator

Sparse Matrix
- 90% static sparsity in the weights,
- 10x less computation,
- 5x less memory footprint

Sparse Vector
- 70% dynamic sparsity in the activation
- 3x less computation

Weight Sharing
- 4bits weights
- 8x less memory footprint

Fully fits in SRAM
- 120x less energy than DRAM

Savings are multiplicative: 5x3x8x120=14,400 theoretical energy improvement.
Distributed Storage and Processing

\[
\tilde{a} \begin{pmatrix} 0 & a_1 & 0 & a_3 \end{pmatrix} \times
\begin{pmatrix}
\begin{array}{cccc}
\text{PE0} & w_{0,0} & w_{0,1} & 0 & w_{0,3} \\
\text{PE1} & 0 & 0 & w_{1,2} & 0 \\
\text{PE2} & 0 & w_{2,1} & 0 & w_{2,3} \\
\text{PE3} & 0 & 0 & 0 & 0 \\
\end{array}
\end{pmatrix}
= \begin{pmatrix}
\begin{array}{c}
\text{ReLU} \Rightarrow \\
\text{b} \\
\end{array}
\end{pmatrix}
\begin{pmatrix}
\begin{array}{c}
b_0 \\
b_1 \\
b_2 \\
b_3 \\
b_4 \\
b_5 \\
b_6 \\
b_7 \\
\end{array}
\end{pmatrix}
\]

logically

physically

<table>
<thead>
<tr>
<th>Virtual Weight</th>
<th>$W_{0,0}$</th>
<th>$W_{0,1}$</th>
<th>$W_{4,2}$</th>
<th>$W_{0,3}$</th>
<th>$W_{4,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Index</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Column Pointer</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
PE Architecture
### Benchmark

- CPU: Intel Core-i7 5930k
- GPU: NVIDIA TitanX
- Mobile GPU: NVIDIA Jetson TK1

<table>
<thead>
<tr>
<th>Layer</th>
<th>Size</th>
<th>Weight Density</th>
<th>Activation Density</th>
<th>FLOP %</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet-6</td>
<td>4096 × 9216</td>
<td>9%</td>
<td>35.1%</td>
<td>3%</td>
<td>AlexNet for image classification</td>
</tr>
<tr>
<td>AlexNet-7</td>
<td>4096 × 4096</td>
<td>9%</td>
<td>35.3%</td>
<td>3%</td>
<td></td>
</tr>
<tr>
<td>AlexNet-8</td>
<td>1000 × 4096</td>
<td>25%</td>
<td>37.5%</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>VGG-6</td>
<td>4096 × 25088</td>
<td>4%</td>
<td>18.3%</td>
<td>1%</td>
<td>VGG-16 for image classification</td>
</tr>
<tr>
<td>VGG-7</td>
<td>4096 × 4096</td>
<td>4%</td>
<td>37.5%</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>VGG-8</td>
<td>1000 × 4096</td>
<td>23%</td>
<td>41.1%</td>
<td>9%</td>
<td></td>
</tr>
<tr>
<td>NeuralTalk-We</td>
<td>600 × 4096</td>
<td>10%</td>
<td>100%</td>
<td>10%</td>
<td>RNN and LSTM for image caption</td>
</tr>
<tr>
<td>NeuralTalk-Wd</td>
<td>8791 × 600</td>
<td>11%</td>
<td>100%</td>
<td>11%</td>
<td></td>
</tr>
<tr>
<td>NeuralTalk-LSTM</td>
<td>2400 × 1201</td>
<td>10%</td>
<td>100%</td>
<td>11%</td>
<td></td>
</tr>
</tbody>
</table>
Scalability

#PEs ~ Speedup
- 64PEs: 64x
- 128PEs: 124x
- 256PEs: 210x
• Imbalanced non-zeros among PEs degrades system utilization.
• This load imbalance could be solved by FIFO.
• With FIFO depth=16, ALU utilization is > 90%.
Result of EIE

<table>
<thead>
<tr>
<th>Technology</th>
<th>45 nm</th>
</tr>
</thead>
<tbody>
<tr>
<td># PEs</td>
<td>64</td>
</tr>
<tr>
<td>on-chip SRAM</td>
<td>8 MB</td>
</tr>
<tr>
<td>Max Model Size</td>
<td>84 Million</td>
</tr>
<tr>
<td>Static Sparsity</td>
<td>10x</td>
</tr>
<tr>
<td>Dynamic Sparsity</td>
<td>3x</td>
</tr>
<tr>
<td>Quantization</td>
<td>4-bit</td>
</tr>
<tr>
<td>ALU Width</td>
<td>16-bit</td>
</tr>
<tr>
<td>Area</td>
<td>40.8 mm^2</td>
</tr>
<tr>
<td>MxV Throughput</td>
<td>81,967 layers/s</td>
</tr>
<tr>
<td>Power</td>
<td>586 mW</td>
</tr>
</tbody>
</table>

1. Post layout result
2. Throughput measured on AlexNet FC-7
Energy Breakdown

- memory: 59%
- clock network: 20%
- combinational: 11%
- register: 9%

- Act_queue: 20%
- SpmatRead: 13%
- ArithmUnit: 12%
- ActRW: 1%
Prediction Accuracy

Mixed Precision:
• 4 bit index (virtual weight)
• 16 bit real weight, 16 bit fixed point ALU
FC Layer: Speedup on EIE

Compared to CPU and GPU:
189x and 13x faster

Baseline:
- Intel Core i7 5930K: MKL CBLAS GEMV, MKL SPBLAS CSRMV
- NVIDIA GeForce GTX Titan X: cuBLAS GEMV, cuSPARSE CSRMV
- NVIDIA Tegra K1: cuBLAS GEMV, cuSPARSE CSRMV
FC Layer: Energy Efficiency on EIE

Compared to CPU and GPU:
24,000x and 3,400x more energy efficient

Baseline:
- Intel Core i7 5930K: reported by pcm-power utility
- NVIDIA GeForce GTX Titan X: reported by nvidia-smi utility
- NVIDIA Tegra K1: measured with power-meter, 60% AP+DRAM power
Comparison: Throughput

- Core-i7 5930k (22nm CPU)
- TitanX (28nm GPU)
- Tegra K1 (28nm mGPU)
- A-Eye (28nm FPGA)
- Da-DianNao (28nm ASIC)
- True-North (28nm ASIC)
- EIE (45nm ASIC)
- EIE (28nm ASIC)
Comparison: Area Efficiency

Area Efficiency (Layers/s/mm^2)

- Core-i7 5930k (22nm CPU)
- TitanX (28nm GPU)
- Tegra K1 (28nm mGPU)
- A-Eye (28nm FPGA)
- Da-DianNao (28nm ASIC)
- True-North (28nm ASIC)
- EIE (45nm ASIC) 64PEs
- EIE (28nm ASIC) 256PEs
Comparison: Energy Efficiency

Energy Efficiency (Layers/J)

- Core-i7 5930k (22nm CPU)
- TitanX (28nm GPU)
- Tegra K1 (28nm mGPU)
- A-Eye (28nm FPGA)
- Da-DianNao (28nm ASIC)
- True-North (28nm ASIC)
- EIE (45nm ASIC) 64PEs
- EIE (28nm ASIC) 256PEs
Where are the savings from?

• Four factors for energy saving:

• 10× *static* weight sparsity;
  less work to do; less bricks to carry.

• 3× *dynamic* activation sparsity;
  carry only good bricks; ignore broken bricks.

• Weight sharing with only 4-bits per weight;
  lighter bricks to carry.

• DRAM => SRAM, no need to go off-chip;
  carry bricks from San Francisco to Seoul => Incheon to Seoul.
Conclusion

- EIE: first accelerator for compressed, sparse neural network.

- Compression => Acceleration, no loss accuracy.

- Distributed storage/computation to parallelize/load balance across PEs.

- 13x faster and 3,400x more energy efficient than GPU. 2.9x faster and 19x more energy efficient than past ASIC.
Beyond EIE: a Multi-Dimension Sparse Recipe for Deep Learning

Faster Speed: EIE accelerator

Smaller Size: Deep Compression, SqueezeNet++

Higher Accuracy: DSD regularization

[1]. Han et al. “Learning both Weights and Connections for Efficient Neural Networks”, NIPS 2015
[5]. Iandola, Han, et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size”, arXiv’16
[6]. Yao, Han, et.al, “Hardware-friendly convolutional neural network with even-number filter size”, ICLR workshop 2016
Backup Slides
Sparsity: Pruning AlexNet & VGGNet

Han et al. “Learning both Weights and Connections for Efficient Neural Networks”, NIPS 2015
Retrain to Fully Recover Accuracy

- L2 regularization w/o retrain
- L1 regularization w/ retrain
- L2 regularization w/ iterative prune and retrain

Han et al. “Learning both Weights and Connections for Efficient Neural Networks”, NIPS 2015
## Weight Sharing: Accuracy with # Bits

<table>
<thead>
<tr>
<th>#CONV bits / #FC bits</th>
<th>Top-1 Error</th>
<th>Top-5 Error</th>
<th>Top-1 Error Increase</th>
<th>Top-5 Error Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>32bits / 32bits</td>
<td>42.78%</td>
<td>19.73%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8 bits / 5 bits</td>
<td>42.78%</td>
<td>19.70%</td>
<td>0.00%</td>
<td>-0.03%</td>
</tr>
<tr>
<td>8 bits / 4 bits</td>
<td>42.79%</td>
<td>19.73%</td>
<td>0.01%</td>
<td>0.00%</td>
</tr>
<tr>
<td>4 bits / 2 bits</td>
<td>44.77%</td>
<td>22.33%</td>
<td>1.99%</td>
<td>2.60%</td>
</tr>
</tbody>
</table>

Han et al. “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding” ICLR 2016
Deep Compression Result on Major Convnets

<table>
<thead>
<tr>
<th>Network</th>
<th>Top-1 Error</th>
<th>Top-5 Error</th>
<th>Parameters</th>
<th>Compress Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-300-100 Ref</td>
<td>1.64%</td>
<td>-</td>
<td>1070 KB</td>
<td></td>
</tr>
<tr>
<td>LeNet-300-100 Compressed</td>
<td>1.58%</td>
<td>-</td>
<td>27 KB</td>
<td>40×</td>
</tr>
<tr>
<td>LeNet-5 Ref</td>
<td>0.80%</td>
<td>-</td>
<td>1720 KB</td>
<td></td>
</tr>
<tr>
<td>LeNet-5 Compressed</td>
<td>0.74%</td>
<td>-</td>
<td>44 KB</td>
<td>39×</td>
</tr>
<tr>
<td>AlexNet Ref</td>
<td>42.78%</td>
<td>19.73%</td>
<td>240 MB</td>
<td></td>
</tr>
<tr>
<td>AlexNet Compressed</td>
<td>42.78%</td>
<td>19.70%</td>
<td>6.9 MB</td>
<td>35×</td>
</tr>
<tr>
<td>VGG-16 Ref</td>
<td>31.50%</td>
<td>11.32%</td>
<td>552 MB</td>
<td></td>
</tr>
<tr>
<td>VGG-16 Compressed</td>
<td>31.17%</td>
<td>10.91%</td>
<td>11.3 MB</td>
<td>49×</td>
</tr>
<tr>
<td>SqueezeNet Ref</td>
<td>42.5%</td>
<td>19.7%</td>
<td>4.8 MB</td>
<td></td>
</tr>
<tr>
<td>SqueezeNet Compressed</td>
<td>42.5%</td>
<td>19.7%</td>
<td>0.47 MB</td>
<td>10×</td>
</tr>
<tr>
<td>GoogLeNet Ref</td>
<td>31.30%</td>
<td>11.10%</td>
<td>28 MB</td>
<td></td>
</tr>
<tr>
<td>GoogLeNet Compressed</td>
<td>31.26%</td>
<td>11.08%</td>
<td>2.8 MB</td>
<td>10×</td>
</tr>
</tbody>
</table>

- SqueezeNet and GoogleNet: just Pruning and Quantization gives 10x compression.
- Inception Model is really efficient for classification.
- But it can still achieve an order of magnitude smaller with Deep Compression.
- Fits in SRAM cache.

Han et al. “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding” ICLR 2016
Pruning NeuralTalk and LSTM

- **Original**: a basketball player in a white uniform is playing with a ball
- **Pruned 90%**: a basketball player in a white uniform is playing with a basketball

- **Original**: a brown dog is running through a grassy field
- **Pruned 90%**: a brown dog is running through a grassy area

- **Original**: a man is riding a surfboard on a wave
- **Pruned 90%**: a man in a wetsuit is riding a wave on a beach

- **Original**: a soccer player in red is running in the field
- **Pruned 95%**: a man in a red shirt and black and white black shirt is running through a field

Han et al. “Learning both Weights and Connections for Efficient Neural Networks”, NIPS 2015 poster
With Sparsity Constraint, DSD Training Improves Accuracy (Baseline: NeuralTalk)

Baseline: a boy is swimming in a pool.
Sparse: a small black dog is jumping into a pool.
DSD: a black and white dog is swimming in a pool.

Baseline: a group of people are standing in front of a building.
Sparse: a group of people are standing in front of a building.
DSD: a group of people are walking in a park.

Baseline: two girls in bathing suits are playing in the water.
Sparse: two children are playing in the sand.
DSD: two children are playing in the sand.

Baseline: a man in a red shirt and jeans is riding a bicycle down a street.
Sparse: a man in a red shirt and a woman in a wheelchair.
DSD: a man and a woman are riding on a street.

Baseline: a group of people sit on a bench in front of a building.
Sparse: a group of people are standing in front of a building.
DSD: a group of people are standing in a fountain.

Baseline: a man in a black jacket and a black jacket is smiling.
Sparse: a man and a woman are standing in front of a mountain.
DSD: a man in a black jacket is standing next to a man in a black shirt.

Baseline: a group of football players in red uniforms.
Sparse: a group of football players in a field.
DSD: a group of football players in red and white uniforms.

Baseline: a dog runs through the grass.
Sparse: a dog runs through the grass.
DSD: a white and brown dog is running through the grass.

Han et al. “DSD: Regularizing Deep Neural Networks with Dense-Sparse-Dense Training Flow”, arXiv 2016