Reciprocity of AI and HPC:
A Programming System Perspective

Xipeng Shen

Work sponsored by NSF & DOE @
North Carolina State University
Heterogeneous computing

Foundations of compilers, runtime, programming languages

ML, AI, & Data analytics

Real-time ML

Heterogeneous computing

Programming systems and Intelligent Computing for future
Heterogeneous computing
Foundations of compilers, runtime, programming languages
Programming systems and intelligent computing for future
ML, AI, & Data analytics
Real-time ML
High-performance ML
Heterogeneous computing
Heterogeneous computing
Foundations of compilers, runtime, programming languages
ML, AI, & Data analytics
Real-time ML
High-performance ML Accelerators

Programming systems and Intelligent Computing for future
Heterogeneous computing
Foundations of compilers, runtime, programming languages
ML, AI, & Data analytics
Real-time ML
High-performance ML
Accelerators
Emerging memory
Heterogeneous computing
Programming systems and Intelligent Computing for future
Heterogeneous computing

Foundations of compilers, runtime, programming languages

Real-time ML
High-performance ML
Accelerators
Emerging memory
Auto-programming

ML, AI, & Data analytics

Heterogeneous computing
Programming systems and Intelligent Computing for future

ML, AI, & Data analytics

Real-time ML
High-performance ML
Accelerators
Emerging memory
Auto-programming
NLP Prog. Synthesis

Heterogeneous computing

Foundations of compilers, runtime, programming languages
Heterogeneous computing
Foundations of compilers, runtime, and programming languages
ML, AI, & Data analytics
Real-time ML
High-performance ML
Accelerators
Emerging memory
Auto-programming
NLP Prog. Synthesis
Code optimizations
Heterogeneous computing
Adaptive Deep Reuse

Sparse Matrix Format Selection
Adaptive Deep Reuse

Sparse Matrix Format Selection

HPC

Prog. System.

AI/ML
ML Speed is Important

Both training & inference.

Image credit: NVIDIA & Google Image
Prior Work on Expediting DNN

➢ Aspects focused by prior work:
  • Reduce weights
  • Reduce precision
  • Leverage sparsity
Prior Work on Expediting DNN

➢ Aspects focused by prior work:
  • Reduce weights
  • Reduce precision
  • Leverage sparsity

Our focus:
Computation Reuse based on properties in convolutional layers’ *inputs*
Deep Reuse for Fast Deep Learning
Deep Reuse for Fast Deep Learning

New Technique Helps AI Learn Almost 70 Per Cent Faster

Adaptive Deep Reuse method reduces training time without sacrificing accuracy.

Matthew Greenwood | April 24, 2019 | Comment

More >>
Intuition
Intuition
Intuition
Basic on Convolution

Image

Weight Filters
Basic on Convolution

Unfold the image to a large input matrix
Deep Reuse – Forward Propagation

\[
\begin{align*}
\mathbf{x} & \quad \times \quad \mathbf{W} \\
\begin{array}{c|c}
\vec{x}_{11} & \vec{x}_{12} \\
\vec{x}_{21} & \vec{x}_{22} \\
\vec{x}_{31} & \vec{x}_{32} \\
\vec{x}_{41} & \vec{x}_{42} \\
\end{array} & \quad & \begin{array}{c|c|c|}
\vec{w}_{11} & \vec{w}_{12} & \vec{w}_{13} \\
\vec{w}_{21} & \vec{w}_{22} & \vec{w}_{23} \\
\end{array} \\
\end{align*}
\]

\[
\mathbf{y} = \begin{array}{c|c|c|}
\vec{y}_{1} & \vec{y}_{2} & \vec{y}_{3} & \vec{y}_{4} \\
\end{array}
\]
Deep Reuse – Forward Propagation

Stage 1: Clustering

\[ x^{(1)} \cdots x^{(2)} \times \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} \]
Deep Reuse – Forward Propagation

Stage 1: Clustering

Stage 2: MM between the centroid matrix and weight matrix

Remaining ratio: $r_c = \frac{|C|}{N}$
Deep Reuse – Forward Propagation

Stage 1: Clustering

Stage 2: MM between the centroid matrix and weight matrix

Stage 3: Reconstructing the output matrix
Deep Reuse – Forward Propagation

Stage 1: Clustering

Stage 2: MM between the centroid matrix and weight matrix

Stage 3: Reconstructing the output matrix
Deep Reuse – Forward Propagation

Stage 1: Clustering

Stage 2: MM between the centroid matrix and weight matrix

Stage 3: Reconstructing the output matrix

Overhead must be smaller than the time savings.
Similarity Identification: LSH

➢ Locality-Sensitive Hashing (LSH).  [ \( H: \) #hashing vectors]
➢ Fast, continuous clustering on the fly

\[
h_v(x) = \begin{cases} 
1 & \text{if } v \cdot x > 0 \\
0 & \text{if } v \cdot x \leq 0 
\end{cases}
\]

\begin{align*}
\text{v1} &= [2, -4] \\
\text{v2} &= [1, 6] \\
\text{x1} &= [1, 3] \\
\text{x2} &= [3, -2] \\
\text{x3} &= [-5, 2] \\
\text{x4} &= [3, -4] \\
\end{align*}

\begin{align*}
\text{x1} &\rightarrow 01 \\
\text{x2} &\rightarrow 10 \\
\text{x3} &\rightarrow 01 \\
\text{x4} &\rightarrow 10 \\
\end{align*}

- group 1: x1, x3
- group 2: x2, x4
Deep Reuse – Explored Factors

➢ Clustering Method: [LSH, K-Means, HyperCube]
➢ Cluster Scope [Single-input level, single-batch level, across-batch level]
➢ Clustering granularity [L: the length of the sub-vector]
➢ Similarity Metric [Angular cosine distance]
Experiments

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>DATASET</th>
<th># CONV LAYERS</th>
<th>K</th>
<th>M</th>
<th>IMAGE ORDER</th>
<th>IMAGE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CifarNet</td>
<td>Cifar10</td>
<td>2</td>
<td>75 ~ 1600</td>
<td>64</td>
<td>RANDOM</td>
<td>32 × 32</td>
</tr>
<tr>
<td>AlexNet</td>
<td>ImageNet</td>
<td>5</td>
<td>363 ~ 3456</td>
<td>64 ~ 384</td>
<td>RANDOM</td>
<td>224 × 224</td>
</tr>
<tr>
<td>VGG-19</td>
<td>ImageNet</td>
<td>16</td>
<td>27 ~ 4068</td>
<td>64 ~ 512</td>
<td>RANDOM</td>
<td>224 × 224</td>
</tr>
</tbody>
</table>

**Server:** an Intel(R) Xeon(R) CPU E5-1607 v2 and a GTX1080 GPU.

**Mobile:** Huawei SE mate mobile phone with Huawei HiSilicon KIRIN 659 processor and a 4 GB memory.
Existence of similarity

Alexnet Cov3
Inference time per batch
Inference Speed

<table>
<thead>
<tr>
<th>Network</th>
<th>Acc</th>
<th>Performance</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ΔAcc</td>
<td>S</td>
</tr>
<tr>
<td>CifarNet</td>
<td>0.7892</td>
<td>-0.0011</td>
<td>1.75X</td>
</tr>
<tr>
<td>AlexNet</td>
<td>0.5360</td>
<td>-0.0002</td>
<td>2.02X</td>
</tr>
<tr>
<td>VGG-19</td>
<td>0.7118</td>
<td>+0.0005</td>
<td>1.89X</td>
</tr>
</tbody>
</table>

On Smartphone

<table>
<thead>
<tr>
<th>Network</th>
<th>$CT_b (ms)$</th>
<th>$S$</th>
<th>$\sigma(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CifarNet</td>
<td>29.42</td>
<td>2.12X</td>
<td>0.163</td>
</tr>
<tr>
<td>AlexNet</td>
<td>326.15</td>
<td>2.55X</td>
<td>0.336</td>
</tr>
</tbody>
</table>
Speedups on Compressed Network (Inference)  
On compressed AlexNet

<table>
<thead>
<tr>
<th>Layer</th>
<th>conv1</th>
<th>conv2</th>
<th>conv3</th>
<th>conv4</th>
<th>conv5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedup</td>
<td>1.81X</td>
<td>3.29X</td>
<td>3.64X</td>
<td>3.45X</td>
<td>2.71X</td>
</tr>
</tbody>
</table>
Time Savings in Training

One-time clustering benefits both forward and backward propagations.

Three strategies:

➢ Strategy 1: Fixed \{L,H\}

➢ Strategy 2: Adaptive \{L,H\}

➢ Strategy 3: Adaptive cluster scope

See ICDE’2019 paper for details.
Short Summary

➢ Tremendous neuron vector similarities exist
➢ Online LSH clustering is efficient enough
➢ Reuse benefits both training and inference
   ➢ greater than 60% training time savings via adaptivity
   ➢ up to 2X speedups of inference
Adaptive Deep Reuse

HPC

Prog. System.

AI/ML
Adaptive Deep Reuse

HPC

Prog. System.

AI/ML
Adaptive Deep Reuse

Sparse Matrix Format Selection

HPC

Prog. System.

AI/ML
Problem

SpMV: core of many HPC applications.

Sparse Matrix

\[
\begin{pmatrix}
1.0 & 0 & 5.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.0 & 0 & 0 & 0 & 0 & 11.0 & 0 & 0 \\
0 & 0 & 0 & 9.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 7.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2.0 & 0 & 0 & 0 & 0 & 10.0 & 0 & 0 & 0 \\
0 & 0 & 0 & 8.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4.0 & 0 & 0 & 0 & 0 & 0 & 12.0 & 0 \\
\end{pmatrix}
\]

Storage formats

- CSR
- COO
- DIA
- ELL
- HYB

Multi-fold performance differences.

No one fits all.
Basic Idea

Treat matrix as an image, use *image recognition* methods for selection.
Basic Idea

Treat matrix as an image, use *image recognition* methods for selection.
Basic Idea

Treat matrix as an image, use *image recognition* methods for selection.
Special Challenge I

- Input representation: fixed size required.
- Image scaling does not work well

Solution: Binary matrix, density matrix, histogram matrix.
Special Challenge II

• DNN structure design
  • Early Merging versus Late Merging
Special Challenge III

• Architecture Sensitivity
  • Best formats differ on different machines for a matrix
Prediction Results on GPU (TitanX)

<table>
<thead>
<tr>
<th>Format</th>
<th>Ground Truth</th>
<th>DNN+Histogram</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precis.</td>
<td>Recall</td>
</tr>
<tr>
<td>CSR</td>
<td>1340</td>
<td>0.87</td>
<td>0.89</td>
</tr>
<tr>
<td>ELL</td>
<td>282</td>
<td>0.73</td>
<td>0.73</td>
</tr>
<tr>
<td>HYB</td>
<td>170</td>
<td>0.61</td>
<td>0.64</td>
</tr>
<tr>
<td>BSR</td>
<td>1806</td>
<td>0.93</td>
<td>0.90</td>
</tr>
<tr>
<td>CSR5</td>
<td>620</td>
<td>0.87</td>
<td>0.91</td>
</tr>
<tr>
<td>COO</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Overall</td>
<td>4218</td>
<td>0.90</td>
<td></td>
</tr>
</tbody>
</table>
Compared to Prior DT-based Method

Accuracy: 85% boosted to 93%. (9200 matrices; Formats: coo, csr, dia, ell)

See PPOPP’2018.
Summary

• Tremendous potential remains between AI & HPC
• Examples
  • Adaptive deep reuse (2-3X DNN speedups)
  • Sparse matrix format selection (86% cases improved)
• Other examples

Wootz: Composibility-based CNN pruning (PLDI’2019)
186X speedups of CNN pruning

Egeria (SC’2017)
NLP-based HPC advisor

Ensemble DNN training framework (SC’2018)
Automate ensemble training on Summit

TADOC (VLDB’2018, ICS’2018)
Analytics directly on compressed data
Modern Computing
Quality of Software

Security

Energy Efficiency

Reliability

Privacy

Machine learning quality

Productivity

Speed

Responsiveness

Adaptivity
Quality of Software

- Security
- Energy Efficiency
- Reliability
- Privacy
- Machine learning quality
- Productivity
- Responsiveness
- Adaptivity
- Speed
Efficiency or speed is the “cash” of modern computing

—Charles Leiserson
Focus: Maximizing computing efficiency through innovative optimizations in compilers, runtime, and algorithms.

rc-accuracy relation on conv2 of CifaNet
rc-accuracy relation on conv2 of CifaNet
Benefits from Transfer Learning

From Xeon E5-4603 to Radeon A8-7600

[Diagram showing accuracy vs. training data size with different lines indicating different methods: Train from scratch, Continuous evolvement, Top evolvement]