A Network-Centric Hardware/Algorithm Co-Design to Accelerate Distributed Training of Deep Neural Networks

Youjie Li\textsuperscript{1}, Jongse Park\textsuperscript{2}, Mohammad Alian\textsuperscript{1}, Yifan Yuan\textsuperscript{1}, Zheng Qu\textsuperscript{3}, Peitian Pan\textsuperscript{4}, Ren Wang\textsuperscript{5}, Alexander Schwing\textsuperscript{1}, Hadi Esmaeilzadeh\textsuperscript{6}, Nam Sung Kim\textsuperscript{1}

\textsuperscript{1}UIUC, \textsuperscript{2}Georgia Tech, \textsuperscript{3}THU, \textsuperscript{4}SJTU, \textsuperscript{5}Intel, \textsuperscript{6}UCSD
Deep learning is growing exponentially!

Training data size (hundreds of TB)
DNN model complexity (hundreds of MB)

Training time (weeks or months)
Distributed learning is essential!
Parallelizing the learning task over multiple nodes.

```
DNN Model Replica
Data Partition

Worker
Worker
Worker
Worker

Aggregator

Gradient
Weights

TensorFlow
Caffe2
mxnet
PaddlePaddle
DL4J
```
Significant communication overhead in distributed learning!

INCEPTIONN framework
- Synchronous training equivalent to TensorFlow
- Five Nodes
- 10 Gb Ethernet
- OpenMPI 2.0
- Titan Xp GPUs
- CUDA 8.0
How to reduce communication?

Straightforward Solution: Compression
Challenges for compression

- **Challenge #1: Expensive compression overhead**

  - Original communication time in AlexNet
  - Communication with 16-bit FP truncation

- **Challenge #2: Limited compressibility of weights**

<table>
<thead>
<tr>
<th>Train AlexNet with 16-bit FP truncation</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline without truncation</td>
<td>80.2 %</td>
</tr>
<tr>
<td>Weight truncation Only</td>
<td>00.9%</td>
</tr>
<tr>
<td>Gradient truncation Only</td>
<td>79.7%</td>
</tr>
</tbody>
</table>
INCEPTIONNN

A hardware/algorithm co-design to accelerate distributed training

- Gradient-centric, decentralized training algorithm
- Hardware-friendly lossy gradient compression algorithm
- In-network accelerator for compression
**In-network acceleration**

Pushing the compression to network

Conventional practice:

- General-purpose processors
  - Heavy overhead

INCEPTIONNN:

- FPGAs
- or
- ASICs
  - Light overhead

Network Interface Card
Hardware-friendly lossy gradient compression

Requirements:
• High compression ratio
• Hardware-friendliness for acceleration
• Minimal loss in training accuracy

Solution:
• Customized lossy compression algorithm for gradients
Why gradients?

High error resilience

Limited range and skewness to zero

AlexNet gradient distribution
Key ideas for lossy gradient compression algorithm

- Remove exponents in FP representation by setting it to a constant
- Remember the diff by shifting on mantissa with a concat’ed 1
- Compress more aggressively as values are close to zeros
Compression with worker-aggregator approach

Limitations
1. Less opportunities for compression
2. Performance bottleneck at aggregators
Gradient-centric decentralized training

Approach
1. Communicate **only** gradients
2. Evenly distribute aggregation to the workers

Advantages
1. Maximize opportunities for compression
2. Balanced load for compression/decompression
Implementation
## Evaluated DNN models and system specifications

<table>
<thead>
<tr>
<th>Name</th>
<th>HDC</th>
<th>AlexNet</th>
<th>ResNet-50</th>
<th>VGG-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model size</td>
<td>5 MB</td>
<td>230 MB</td>
<td>100 MB</td>
<td>525 MB</td>
</tr>
<tr>
<td>Dataset</td>
<td>MNIST</td>
<td>ImageNet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mini-batch size</td>
<td>25</td>
<td>64</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>Training data size</td>
<td>60,000</td>
<td></td>
<td>1,281,167</td>
<td></td>
</tr>
</tbody>
</table>

### System specifications

<table>
<thead>
<tr>
<th>System specifications</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>4</td>
</tr>
<tr>
<td>System software</td>
<td>C++, CUDA 8.0, Intel MKL 2018, and OpenMPI 2.0.2</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Xeon CPU E5-2640 @2.6 GHz</td>
</tr>
<tr>
<td>GPU</td>
<td>NVIDIA Titan Xp</td>
</tr>
<tr>
<td>FPGA</td>
<td>Xilinx VC709 board</td>
</tr>
<tr>
<td>Network</td>
<td>10 Gb Ethernet</td>
</tr>
</tbody>
</table>
INC+C offers 76% lower communication time compared to the WA baseline.

INC+C offers 2.2~3.1x system-level speedup over the WA baseline.

- WA: Worker-aggregator
- INC: INCEPTIONN
- WA+C: WA with compression
- INC+C: INC with compression
Impact on final training accuracy

Only 1 or 2 more epochs are required to match the same level of accuracy
Conclusion

INCEPTIONNN
Hardware-algorithm co-designed in-network acceleration solution to reduce the communication overhead in distributed DNN training