COMET: A Novel Memory-Efficient Deep Learning Training Framework by Using Error-Bounded Lossy Compression

Sian Jin  
Indiana University  
Bloomington, IN, USA  
sianjin@iu.edu

Chengming Zhang  
Indiana University  
Bloomington, IN, USA  
czh5@iu.edu

Hui Guan  
University of Massachusetts  
Amherst, MA, USA  
huiguan@cs.umass.edu

Guanpeng Li  
University of Iowa  
Iowa City, IA, USA  
guanpeng-li@uiowa.edu

Xintong Jiang  
McGill University  
Montréal, QC, Canada  
xintong.jiang@mail.mcgill.ca

Shuaiwen Leon Song  
University of Sydney  
Sydney, NSW, Australia  
shuaiwen.song@sydney.edu.au

Yunhe Feng  
University of Washington  
Seattle, WA, USA  
yunhe@uw.edu

Dingwen Tao  
Indiana University  
Bloomington, IN, USA  
dita@iu.edu
Introduction

- **Challenges In Training DNN**
  - **High memory consumption**
  - Large batch size needed
  - Highly limited GPU memory space

- **Activation Data Storage For Training**
  - Must stored until used in back propagation
  - Long waiting period between generating and using the data

- **Previous Solutions**
  - Migration between CPU and GPU
    - Limited I/O throughput
  - Recomputation
    - High overhead for Conv Layer
  - Image-based compression
    - Low compression ratio

---

Memory consumption and top-1 accuracy of different state-of-the-art neural networks

Data flow in a sample iteration of training CNNs
Lossy Compression

- **Lossy Compression**
  - Compress data with little information loss in the reconstructed data
  - High compression ratio (Over 10x), compared to lossless compression (< 2x)
  - Controllable compression error

- **Lossy Compressors**
  - Transform-based lossy compression e.g., ZFP
  - Prediction-based lossy compressor e.g., SZ

- **Use Cases**
  - Reduce storage overhead
  - Improve I/O performance
  - First work to reduce memory consumption for DNN training

- **Challenges**
  - Continuous zero handling with prediction based lossy compression
  - Understand how the introduced error would propagate through the whole training process
  - Balance between compression ratio and accuracy
Overall Design

Overview of our proposed memory-efficient DNN training framework, COMET

- **Parameter Collection**: collect parameters for analysis and updating compression configurations
- **Gradient Assessment**: estimate acceptable variance in the gradient
- **Activation Assessment**: estimate acceptable error introduced for compressing activation data
- **Adaptive Compression**: deploy lossy compression
Parameter Collection
- **Offline parameters**: batch size, activation data size, corresponding output layer size
- **Simi-online parameters**: activation data sparsity, average loss, average momentum value

Gradient Assessment
- Compute $\sigma$ based on parameters and empirical experience:
  \[ \sigma = 0.01M_{Average} \]

Activation Assessment
- Error distribution estimation (uniform distribution)
- Gradient error distribution estimation (normal distribution)
- Compute error bound based on parameters and theoretical analysis:
  \[ eb = \frac{\sigma}{aL\sqrt{NR}} \]

Check out our theoretical analysis in the paper!

An example of gradient error distribution after compression
Parameter Collection

- **Offline parameters**: batch size, activation data size, corresponding output layer size
- **Simi-online parameters**: activation data sparsity, average loss, average momentum value

Gradient Assessment

- Compute $\sigma$ based on parameters and empirical experience:
  \[ \sigma = 0.01M_{Average} \]

Activation Assessment

- Compute error bound based on parameters and theoretical analysis:
  \[ eb = \frac{\sigma}{aL\sqrt{NR}} \]

Adaptive Compression

- Compression configuration update every 1000 iterations
- Modified **cusZ** for compressing sparse data
  - Zero remains zero after lossy (de)compression

Training failed without modification to the compressor

Check out our theoretical analysis in the paper!
Evaluation

➤ Evaluation Setup

• Models: AlexNet; VGG-16; ResNet-18; ResNet-50; EfficientNet
• Datasets: ImageNet-2012; Stanford Dogs
• Frameworks: TensorFlow; Caffe
• Platform: Longhorn at TACC; Bridge-2 at PSC (V100 GPUs)

➤ Error Impact Evaluation

• The accuracy loss caused by the errors added to a given convolutional layer is not noticeably amplified by its following layers

Determine the acceptable error introduced to the gradient

Accurate theoretical prediction to the gradient error distribution
### Evaluation

#### Memory Reduction Evaluation
- High compression ratio, up to 13.5x
- Little/no testing accuracy loss

<table>
<thead>
<tr>
<th>Neural Nets</th>
<th>Top-1 Accuracy</th>
<th>Peak Mem.</th>
<th>Max Batch</th>
<th>Conv. Act. Size</th>
<th>COMET</th>
<th>JPEG-ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>b. AlexNet</td>
<td>57.41%</td>
<td>2.17 GB</td>
<td>512</td>
<td>407 MB</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>c. AlexNet</td>
<td>57.42%</td>
<td>0.85 GB</td>
<td>2048</td>
<td>30 MB</td>
<td>13.5×</td>
<td>-</td>
</tr>
<tr>
<td>b. VGG-16</td>
<td>68.05%</td>
<td>17.29 GB</td>
<td>64</td>
<td>6.91 GB</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>c. VGG-16</td>
<td>68.02%</td>
<td>5.04 GB</td>
<td>256</td>
<td>0.62 GB</td>
<td>11.1×</td>
<td>-</td>
</tr>
<tr>
<td>b. ResNet-18</td>
<td>67.57%</td>
<td>5.16 GB</td>
<td>256</td>
<td>1.71 GB</td>
<td>10.7×</td>
<td>7.3×</td>
</tr>
<tr>
<td>c. ResNet-18</td>
<td>67.43%</td>
<td>1.37 GB</td>
<td>1024</td>
<td>0.16 GB</td>
<td>11.0×</td>
<td>6.0×</td>
</tr>
<tr>
<td>b. ResNet-50</td>
<td>75.55%</td>
<td>15.57 GB</td>
<td>128</td>
<td>5.14 GB</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>c. ResNet-50</td>
<td>75.51%</td>
<td>4.40 GB</td>
<td>512</td>
<td>0.46 GB</td>
<td>11.0×</td>
<td>-</td>
</tr>
</tbody>
</table>

Training accuracy curve comparison between the baseline and our proposed framework.

Comparison of accuracy and activation size between baseline training and our proposed framework.
Evaluation

➢ Performance Evaluation and Analysis

• Low compression overhead, significantly lower than data migration solution
• Raw performance improvement (sample/sec) with better GPU resource utilization
• End-end performance improvement
• High Scalability

![Validation accuracy curve of COMET under different GPU memory constraint on AlexNet](image)

![Overhead comparison between migration, recomputation](image)

Training performance on ResNet-50 with different Batch size
Conclusion and Future Work

➢ Conclusion
  • A novel memory-efficient CNN training framework via dynamically compressing the intermediate activation data through error-bounded lossy compression
  • A thorough analysis of the impact of compression error propagation during DNN training from both theoretical and empirical perspectives
  • An adaptive scheme to adaptively configure the error-bounded lossy compression based on a series of current training status data
  • Improved SZ error-bounded lossy compression to handle compressing continuous zeros
  • Reduce the memory consumption by up to $13.5 \times$ and $1.8 \times$ compared to the original training framework and the state-of-the-art method, respectively. Improve the end-to-end training performance by up to $2 \times$

➢ Future Work
  • Integrate data migration and recomputation methods to COMET
  • Explore the applicability of COMET to other types of layers and models
  • Reduce the (de)compression overhead
Thank you!

Any questions are welcome!

Contact  Dingwen Tao: ditao@iu.edu
          Sian Jin: sianjin@iu.edu