POSTER: A Novel Memory-Efficient Deep Learning Training Framework via Error-Bounded Lossy Compression

Sian Jin  
Washington State University  
Pullman, WA, USA  
sian.jin@wsu.edu

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

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

Dingwen Tao*  
Washington State University  
Pullman, WA, USA  
dingwen.tao@wsu.edu
Why Compress DNNs’ Activation Data

Why Compress?

- High memory consumption
- Large batch size needed
- Highly limited GPU memory space

Memory consumption and top-1 accuracy of different state-of-the-art neural networks.
Deep Neural Networks

➢ Why Compress?
  • 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

Data flow in a sample iteration of training CNNs
Deep Neural Networks

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

- **Convolutional Layer?**
  - Not able to recompute without high overhead
  - Large proportion of activation data consumption

Data flow in a sample iteration of training CNNs
Deep Neural Networks

Convolutional Layer?
- Not able to recompute without high overhead
- Large proportion of activation data consumption

Why SZ Lossy Compression?
- High compression ratio compared to lossless compression
- Specifically designed for floating-point data
- GPU enabled, with high (de)compression throughput

Data flow in a sample iteration of training CNNs
Overall Design

- **Parameter Collection**: collect parameters for analysis and updating compression configurations
- **Gradient Assessment**: estimate acceptable $\sigma$ in the gradient
- **Activation Assessment**: estimate acceptable error bound for compressing activation data
- **Adaptive Compression**: deploy lossy compression

Overview of our proposed memory-efficient DNN training framework
Evaluation

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

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

<table>
<thead>
<tr>
<th>Neural Network</th>
<th>Top-1 Accuracy</th>
<th>Convolutional Act. Size</th>
<th>Compress Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>57.41%</td>
<td>407 MB</td>
<td></td>
</tr>
<tr>
<td>AlexNet compressed</td>
<td>57.42%</td>
<td>30 MB</td>
<td>13.5×</td>
</tr>
<tr>
<td>baseline</td>
<td>68.05%</td>
<td>9.30 GB</td>
<td></td>
</tr>
<tr>
<td>VGG-16 compressed</td>
<td>68.02%</td>
<td>0.83 GB</td>
<td>11.1×</td>
</tr>
<tr>
<td>baseline</td>
<td>67.57%</td>
<td>3.42 GB</td>
<td></td>
</tr>
<tr>
<td>ResNet-18 compressed</td>
<td>67.43%</td>
<td>0.32 GB</td>
<td>10.7×</td>
</tr>
<tr>
<td>baseline</td>
<td>71.49%</td>
<td>10.28 GB</td>
<td></td>
</tr>
<tr>
<td>ResNet-50 compressed</td>
<td>71.18%</td>
<td>0.93 GB</td>
<td>11.0×</td>
</tr>
</tbody>
</table>

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

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

➢ Performance Evaluation and Analysis
  • Low introduced overhead, as low as 7% of total computing time
  • High performance improvement, up to 1.27x raw performance

<table>
<thead>
<tr>
<th>Neural Network</th>
<th>Top-1 Accuracy</th>
<th>Convolutional Act. Size</th>
<th>Compress Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>57.41%</td>
<td>407 MB</td>
<td></td>
</tr>
<tr>
<td>AlexNet compressed</td>
<td>57.42%</td>
<td>30 MB</td>
<td>13.5x</td>
</tr>
<tr>
<td>baseline</td>
<td>68.05%</td>
<td>9.30 GB</td>
<td></td>
</tr>
<tr>
<td>VGG-16 compressed</td>
<td>68.02%</td>
<td>0.83 GB</td>
<td>11.1 x</td>
</tr>
<tr>
<td>baseline</td>
<td>67.57%</td>
<td>3.42 GB</td>
<td></td>
</tr>
<tr>
<td>ResNet-18 compressed</td>
<td>67.43%</td>
<td>0.32 GB</td>
<td>10.7 x</td>
</tr>
<tr>
<td>baseline</td>
<td>71.49%</td>
<td>10.28 GB</td>
<td></td>
</tr>
<tr>
<td>ResNet-50 compressed</td>
<td>71.18%</td>
<td>0.93 GB</td>
<td>11.0 x</td>
</tr>
</tbody>
</table>

Comparison of accuracy and activation size between baseline training and our proposed framework
Conclusion and Future Work

 ➢ **Our Proposed Framework**
  • A novel memory-efficient DNN training framework via dynamically compressing the intermediate activation data through error-bounded lossy compression
  • A thorough analysis on the impact of compression error propagation during CNN training from both theoretical and empirical perspectives
  • An adaptive scheme to dynamically configure the error-bounded lossy compression based on a series of current training status data
  • Evaluate our proposed framework on popular models
    • Can provide up to 13.5x compression ratio, up to 1.8x improvement compared to previous state-of-the-art method

 ➢ **Future Work**
  • Implement orthogonal methods including data migration and layer re-computation
  • Optimize (de)compression performance on activation data
Thank you!

Any questions are welcome!

Contact
Dingwen Tao: dingwen.tao@wsu.edu
Sian Jin: sian.jin@wsu.edu