What is the State of Neural Network Pruning?

Davis Blalock*
Jose Javier Gonzalez*
Jonathan Frankle
John V. Guttag

*equal contribution
**Overview**

**Meta-analysis of neural network pruning**  
We aggregated results across 81 pruning papers and pruned hundreds of networks in controlled conditions  
  • Some surprising findings…

**ShrinkBench**  
Open source library to facilitate development and standardized evaluation of neural network pruning methods
Part 0: Background
• Neural networks are often accurate but large
• **Pruning**: Systematically removing parameters from a network
Typical Pruning Pipeline

Many design choices:

- **Scoring** importance of parameters
- **Schedule** of pruning, training / finetuning
- **Structure** of induced sparsity
- **Finetuning** details — optimizer, duration, hyperparameters
• **Goal**: Increase efficiency of network as much as possible with minimal drop in quality

• **Metrics**
  - Quality = Accuracy
  - Efficiency = FLOPs, compression, latency…

• Must use comparable tradeoffs
Part 1: Meta-Analysis
Overview of Meta-Analysis

• We aggregated results across 81 pruning papers
• Mostly published in top venues
• Corpus closed under experimental comparison

<table>
<thead>
<tr>
<th>Venue</th>
<th># of Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>arXiv only</td>
<td>22</td>
</tr>
<tr>
<td>NeurIPS</td>
<td>16</td>
</tr>
<tr>
<td>ICLR</td>
<td>11</td>
</tr>
<tr>
<td>CVPR</td>
<td>9</td>
</tr>
<tr>
<td>ICML</td>
<td>4</td>
</tr>
<tr>
<td>ECCV</td>
<td>4</td>
</tr>
<tr>
<td>BMVC</td>
<td>3</td>
</tr>
<tr>
<td>IEEE Access</td>
<td>2</td>
</tr>
<tr>
<td>Other</td>
<td>10</td>
</tr>
</tbody>
</table>
Robust Findings

- Pruning works
  - Almost any heuristic improves efficiency with little performance drop
  - Many methods better than random pruning
- Don’t prune all layers uniformly
- Sparse models better for fixed # of parameters
Better Pruning vs Better Architecture
Ideal Results Over Time

ResNet-50 on ImageNet

Change in Top-1 Accuracy (%)

Compression Ratio

2015
2016
2017
2018
2019

(Dataset, Architecture, X metric, Y metric, Hyperparameters) → Curve
Ideal Results Over Time

**VGG-16 on ImageNet**

- **Compression Ratio**
  - Change in Top-1 Accuracy (%)
  - Theoretical Speedup

**AlexNet on ImageNet**

- **Compression Ratio**
  - Change in Top-1 Accuracy (%)
  - Theoretical Speedup

**ResNet-50 on ImageNet**

- **Compression Ratio**
  - Change in Top-1 Accuracy (%)
  - Theoretical Speedup

Colors:
- Orange: 2015
- Red: 2016
- Purple: 2017
- Blue: 2018
- Cyan: 2019
Actual Results Over Time

**VGG-16 on ImageNet**
![Graph showing change in top-1 accuracy and compression ratio over time for VGG-16 on ImageNet.](image)

**AlexNet on ImageNet**
![Graph showing change in top-1 accuracy and compression ratio over time for AlexNet on ImageNet.](image)

**ResNet-50 on ImageNet**
![Graph showing change in top-1 accuracy and compression ratio over time for ResNet-50 on ImageNet.](image)

Theoretical Speedup

- **2015**
- **2016**
- **2017**
- **2018**
- **2019**
Quantifying the Problem

- Among 81 papers:
  - 49 datasets
  - 132 architectures
  - 195 (dataset, architecture) pairs
- Vicious cycle: extreme burden to compare to existing methods

### All (dataset, architecture) pairs used in at least 4 papers

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Architecture</th>
<th># of Papers Using Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet</td>
<td>VGG-16</td>
<td>22</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>ResNet-56</td>
<td>14</td>
</tr>
<tr>
<td>ImageNet</td>
<td>ResNet-50</td>
<td>14</td>
</tr>
<tr>
<td>ImageNet</td>
<td>CaffeNet</td>
<td>11</td>
</tr>
<tr>
<td>ImageNet</td>
<td>AlexNet</td>
<td>9</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>CIFAR-VGG</td>
<td>8</td>
</tr>
<tr>
<td>ImageNet</td>
<td>ResNet-34</td>
<td>6</td>
</tr>
<tr>
<td>ImageNet</td>
<td>ResNet-18</td>
<td>6</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>ResNet-110</td>
<td>5</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>PreResNet-164</td>
<td>4</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>ResNet-32</td>
<td>4</td>
</tr>
</tbody>
</table>
Dearth of Reported Comparisons

• **Presence of comparisons:**
  • Most papers compare to at most 1 other method
  • 40% papers have never been compared to
  • Pre-2010s methods almost completely ignored

• **Reinventing the wheel:**
  • Magnitude-based pruning: *Janowsky (1989)*
  • Gradient times magnitude: *Mozer & Smolensky (1989)*
  • “Reviving” pruned weights: *Tresp et al. (1997)*
Pop quiz!

- Alice’s network has 10 million parameters. She prunes 8 million of them. What compression ratio might she report in her paper?
  A. 80%
  B. 20%
  C. 5x
  D. No reported compression ratio
Pop quiz!

• Alice’s network has 10 million parameters. She prunes 8 million of them. What compression ratio might she report in her paper?

A. 80%
B. 20%
C. 5x
D. No reported compression ratio
Pop quiz!

• According to the literature, how many FLOPs does it take to run inference using AlexNet on ImageNet?

  A. 371 million
  B. 500 million
  C. 724 million
  D. 1.5 billion
Pop quiz!

• According to the literature, how many FLOPs does it take to run inference using AlexNet on ImageNet?

  A. 371 million  
  B. 500 million  
  C. 724 million  
  D. 1.5 billion
Part 2: ShrinkBench
Why ShrinkBench?

- Want to hold everything but pruning algorithm constant
  - Improved rigor, development time

Potential confounding factors
• Lets algorithm return arbitrary masks for weight tensors
• Standardizes all other aspects of training and evaluation
Crucial to Vary Amount of Pruning & Architecture

![Graph showing accuracy vs. compression ratio for CIFAR-VGG and ResNet-56 models with different pruning methods.](image-url)
Compression and Speedup are not Interchangeable

ResNet-18 on ImageNet

Accuracy

Compression Ratio

1 2 4 8 16

Theoretical Speedup

1 2 4 8 16 32

Method A

Method C

Method B

Method D
Using Identical Initial Weights is essential

ResNet-56 on CIFAR-10

Accuracy

Compression Ratio

Weights A

Weights B
Conclusion

• Pruning works
  • But not as well as improving architecture

• But we have no idea what methods work the best
  • Field suffers from extreme fragmentation in experimental setups

• We introduce a library/benchmark to address this
  • Faster progress in the future, interesting findings already

https://github.com/jjgo/shrinkbench
Questions?