Lookahead: A Far-sighted Alternative of Magnitude-based Pruning

Speaker: Sejun Park

Joint work with Jaeho Lee, Sangwoo Mo, and Jinwoo Shin
Korea Advanced Institute of Science and Technology (KAIST)

ICLR 2020
Modern neural networks are severely over-parametrized

- For \( N \) training data, \( O(N) \) parameters network can achieve zero training error [Yun et al.’19]
- e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

### Number of parameters and ImageNet classification accuracy [Zoph et al.’18]

<table>
<thead>
<tr>
<th>Model</th>
<th>image size</th>
<th># parameters</th>
<th>Mult-Adds</th>
<th>Top 1 Acc. (%)</th>
<th>Top 5 Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception V2 [29]</td>
<td>224×224</td>
<td>11.2 M</td>
<td>1.94B</td>
<td>74.8</td>
<td>92.2</td>
</tr>
<tr>
<td>NASNet-A (5 @ 1538)</td>
<td>299×299</td>
<td>10.9 M</td>
<td>2.35B</td>
<td>78.6</td>
<td>94.2</td>
</tr>
<tr>
<td>Inception V3 [60]</td>
<td>299×299</td>
<td>23.8 M</td>
<td>5.72B</td>
<td>78.8</td>
<td>94.4</td>
</tr>
<tr>
<td>Xception [9]</td>
<td>299×299</td>
<td>22.8 M</td>
<td>8.38B</td>
<td>79.0</td>
<td>94.5</td>
</tr>
<tr>
<td>Inception ResNet V2 [58]</td>
<td>299×299</td>
<td>55.8 M</td>
<td>13.2B</td>
<td>80.1</td>
<td>95.1</td>
</tr>
<tr>
<td>NASNet-A (7 @ 1920)</td>
<td>299×299</td>
<td>22.6 M</td>
<td>4.93B</td>
<td>80.8</td>
<td>95.3</td>
</tr>
<tr>
<td>ResNeXt-101 (64 x 4d) [68]</td>
<td>320×320</td>
<td>83.6 M</td>
<td>31.5B</td>
<td>80.9</td>
<td>95.6</td>
</tr>
<tr>
<td>PolyNet [69]</td>
<td>331×331</td>
<td>92 M</td>
<td>34.7B</td>
<td>81.3</td>
<td>95.8</td>
</tr>
<tr>
<td>DPN-131 [8]</td>
<td>320×320</td>
<td>79.5 M</td>
<td>32.0B</td>
<td>81.5</td>
<td>95.8</td>
</tr>
<tr>
<td>SENet [25]</td>
<td>320×320</td>
<td>145.8 M</td>
<td>42.3B</td>
<td>82.7</td>
<td>96.2</td>
</tr>
<tr>
<td>NASNet-A (6 @ 4032)</td>
<td>331×331</td>
<td>88.9 M</td>
<td>23.8B</td>
<td>82.7</td>
<td>96.2</td>
</tr>
</tbody>
</table>
Modern neural networks are severely over-parametrized

- For $N$ training data, $O(N)$ parameters network can achieve zero training error [Yun et al.’19]
- e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

More parameters
- Better generalization
- Better training accuracy
- Better optimization landscape
- Better convergence speed
- ...
Motivation: Over-parametrization in Modern Deep Learning

Modern neural networks are severely over-parametrized

• For $N$ training data, $O(N)$ parameters network can achieve zero training error [Yun et al.’19]
  • e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

More parameters
• Better generalization
• Better training accuracy
• Better optimization landscape
• Better convergence speed
• …

More parameters
• More memory
• More inference time
• More power consumption
• More CO$_2$
• …
Modern neural networks are severely over-parametrized

- For $N$ training data, $O(N)$ parameters network can achieve zero training error [Yun et al.’19]
  - e.g., 16M parameters are enough for fitting ImageNet dataset perfectly

**Motivation: Over-parametrization in Modern Deep Learning**

- More parameters
  - Better generalization
  - Better training accuracy
  - Better optimization landscape
  - Better convergence speed
  - ...

- Less parameters
  - Less memory
  - Less inference time
  - Less power consumption
  - Less CO$_2$
  - ...

**Pruning over-parametrized network**
Magnitude-based pruning (MP) is a popular pruning algorithm, removing small weight edges. Despite its simplicity, MP has been showing remarkable performance in practice:

Magnitude-based pruning (MP) is a popular pruning algorithm, removing small weight edges. Despite its simplicity, MP has been showing remarkable performance in practice. However, large weight edges may not be important as much as their weights.

Motivation: Magnitude-based Pruning

What if there exists large weight edges connected only to small weight edges?
We propose a new pruning algorithm by
1. Interpreting MP as layerwise approximation
2. Extending it to block approximation
Intuition: Magnitude-based Pruning = Layerwise Approximation

For each layer, MP minimizes Frobenius norm of difference of weight tensors before and after pruning

\[
\|W_\ell x - \tilde{W}_\ell x\|_2 \leq \|W_\ell - \tilde{W}_\ell\|_2 \cdot \|x\|_2 \\
\leq \|W_\ell - \tilde{W}_\ell\|_F \cdot \|x\|_2
\]

\(x\) : Input of the layer
\(W_\ell\) : Weight before pruning
\(\tilde{W}_\ell\) : Weight after pruning
For each layer, MP minimizes Frobenius norm of difference of weight tensors before and after pruning:

\[ \| W_\ell x - \widetilde{W}_\ell x \|_2 \leq \| W_\ell - \widetilde{W}_\ell \|_2 \cdot \| x \|_2 \]

\[ \leq \| W_\ell - \widetilde{W}_\ell \|_F \cdot \| x \|_2 \]

\( x \): Input of the layer
\( W_\ell \): Weight before pruning
\( \widetilde{W}_\ell \): Weight after pruning
Intuition: Magnitude-based Pruning = Layerwise Approximation

For each layer, MP minimizes Frobenius norm of difference of weight tensors before and after pruning

$$\text{score}_{MP}(i, j; \ell) = |W_{\ell}[i, j]|$$

Pruning edge with smallest MP score minimizes

$$\|W_{\ell} - \tilde{W}_{\ell}\|_F$$
We propose lookahead pruning (LAP) extending layerwise approximation of MP to block of layers:

$$\|W_{\ell+1}W_\ell W_{\ell-1}x - W_{\ell+1} \tilde{W}_\ell W_{\ell-1}x\|_2 \leq \|W_{\ell+1}(W_\ell - \tilde{W}_\ell)W_{\ell-1}\|_2 \cdot \|x\|_2 \leq \|W_{\ell+1}(W_\ell - \tilde{W}_\ell)W_{\ell-1}\|_F \cdot \|x\|_2$$

Assume linear activation for now.
We propose lookahead pruning (LAP) extending layerwise approximation of MP to block of layers

\[
\|W_{\ell+1}W_{\ell}W_{\ell-1}x - W_{\ell+1}\tilde{W}_{\ell}W_{\ell-1}x\|_2 \leq \|W_{\ell+1}(W_{\ell} - \tilde{W}_{\ell})W_{\ell-1}\|_2 \cdot \|x\|_2
\]

\[
\leq \|W_{\ell+1}(W_{\ell} - \tilde{W}_{\ell})W_{\ell-1}\|_F \cdot \|x\|_2
\]
We propose lookahead pruning (LAP) extending layerwise approximation of MP to block of layers.

$$\text{score}_{\text{LAP}}(i, j; \ell) = |W_\ell[i, j]| \cdot \|W_{\ell+1}[; i]\|_F \cdot \|W_{\ell-1}[j, :]\|_F$$

Pruning edge with smallest LAP score minimizes

$$\|W_{\ell+1}(W_\ell - \tilde{W}_\ell)W_{\ell-1}\|_F$$
Contribution: Lookahead Pruning for ReLU

LAP for ReLU activation under i.i.d. activation probability

\[
\mathbb{E}\left[\|W_{\ell+1}X_\ell(W_\ell - \tilde{W}_\ell)X_{\ell-1}W_{\ell-1}x\|_2^2\right]^{\frac{1}{2}} \leq \mathbb{E}\left[\|W_{\ell+1}X_\ell(W_\ell - \tilde{W}_\ell)X_{\ell-1}\|_F^2\right]^{\frac{1}{2}} \cdot \|x\|_2
\]

0-1 random diagonal matrix indicating activated neurons

![Diagram of neural network with highlighted connections and activations]
LAP for ReLU activation under i.i.d. activation probability

$$\mathbb{E} \left[ ||W_{\ell+1} X_\ell (W_\ell - \tilde{W}_\ell) X_{\ell-1} W_{\ell-1} x||_2^2 \right]^{\frac{1}{2}} \leq \mathbb{E} \left[ ||W_{\ell+1} X_\ell (W_\ell - \tilde{W}_\ell) X_{\ell-1} W_{\ell-1}||_F^2 \right]^{\frac{1}{2}} \cdot ||x||_2$$
LAP for ReLU activation under i.i.d. activation probability

$$\text{score}_{\text{LAP}}(i, j; \ell) = |W_\ell[i, j]| \cdot \|W_{\ell+1}[; i]\|_F \cdot \|W_{\ell-1}[j, :]\|_F$$

Pruning edge with smallest LAP score minimizes

$$\mathbb{E}\left[\|W_{\ell+1}X_\ell(W_\ell - \tilde{W}_\ell)X_{\ell-1}W_{\ell-1}\|_F^2\right]^{\frac{1}{2}}$$
Empirical evaluation of LAP and MP for MNIST classification task under non-linear activation functions

Experiments: Lookahead Pruning for Other Activations

ReLU

sigmoid

hyperbolic tangent
Empirical evaluation of LAP and MP for CIFAR-10 and Tiny-ImageNet classification tasks

<table>
<thead>
<tr>
<th>Test error rates of VGG-19 on CIFAR-10</th>
<th>Unpruned models have 9.02% error rate.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.09%</td>
</tr>
<tr>
<td>MP</td>
<td>8.99±0.12</td>
</tr>
<tr>
<td>LAP</td>
<td>8.89±0.14</td>
</tr>
<tr>
<td>(-1.07%)</td>
<td>(-3.96%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test error rates of ResNet-18 on CIFAR-10</th>
<th>Unpruned models have 8.68% error rate.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.30%</td>
</tr>
<tr>
<td>MP</td>
<td>8.18±0.33</td>
</tr>
<tr>
<td>LAP</td>
<td>8.09±0.10</td>
</tr>
<tr>
<td>(-1.08%)</td>
<td>(+2.59%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Top-5 test error rates of VGG-19 on Tiny-ImageNet</th>
<th>Unpruned models have 36.89% error rate.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.16%</td>
</tr>
<tr>
<td>MP</td>
<td>36.40±1.31</td>
</tr>
<tr>
<td>LAP</td>
<td>36.01±1.31</td>
</tr>
<tr>
<td>(-1.07%)</td>
<td>(-0.90%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Top-5 test error rates of ResNet-50 on Tiny-ImageNet</th>
<th>Unpruned models have 23.19% error rate.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.52%</td>
</tr>
<tr>
<td>MP</td>
<td>23.88±1.27</td>
</tr>
<tr>
<td>LAP</td>
<td>23.64±1.00</td>
</tr>
<tr>
<td>(-1.00%)</td>
<td>(-0.34%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Top-5 test error rates of WRN-16-8 on Tiny-ImageNet</th>
<th>Unpruned models have 25.77% error rate.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.22%</td>
</tr>
<tr>
<td>MP</td>
<td>25.27±0.73</td>
</tr>
<tr>
<td>LAP</td>
<td>24.99±1.85</td>
</tr>
<tr>
<td>(-1.12%)</td>
<td>(-0.87%)</td>
</tr>
</tbody>
</table>

LAP outperforms MP especially in high-sparsity regime!
We propose lookahead pruning by extending layerwise approximation of MP to block approximation.
We propose lookahead pruning by extending layerwise approximation of MP to block approximation.

In our paper, there are

- More empirical evaluations
- Variants and sequential version of LAP
- LAP for various types of layers
- LAP utilizing real activation probability
- ...

Codes are available at https://github.com/alinlab/lookahead_pruning