LEARNING SPARSE NEURAL NETWORKS THROUGH L0 REGULARIZATION

Christos Louizos, Max Welling, Diederik P. Kingma

STA 4273 Paper Presentation
Daniel Flam-Shepherd, Armaan Farhadi & Zhaoyu Guo

March 2nd, 2018
Neural Networks: the good and the bad

Neural Networks: ...

1. are flexible function approximators that scale really well
2. are overparameterized and prone to overfitting and memorization

So what can we do about this?

Model compression and sparsification!

Consider the Empirical Risk minimization problem

$$\min_{\theta} R(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(f(x_i; \theta), y_i) + \lambda ||\theta||_p$$

where

1. \( \{(x_i, y_i)\}_{i=1}^{N} \) is the iid dataset of input-output pairs
2. \( f(x; \theta) \) is the NN using parameters \( \theta \)
3. \( ||\theta||_p \) is the \( L^p \) norm
4. \( L(\cdot) \) is the loss function
Lp Norms

The $L_0$ ”norm” is just the number of nonzero parameters.

$$||\theta||_0 = \sum_{j=1}^{||\theta||} \mathbb{I}[\theta_j \neq 0]$$

This does not impose shrinkage on large $\theta_j$ rather it directly penalizes $|\theta|$. 

Figure: $L_p$ norm penalties for parameter $\theta$ from lousizos et al
If we use the $L_p$ norm $R(\theta)$ is non-differentiable at 0.

How can we relax this optimization and ensure $0 \in \theta$?

First, Reparameterize by putting binary gates $z_j$ on each $\theta_j$.

$$\theta_j = \tilde{\theta}_j z_j, \quad z_j \in \{0, 1\}, \quad \tilde{\theta}_j \neq 0, \quad \|\theta\|_0 = \sum_{j=1}^{|\theta|} z_j$$

let $z_j \sim \text{Ber}(\pi_j)$ with pmf $q(z_j | \pi_j)$ and we can formulate the problem as:

$$\min_{\tilde{\theta}, \pi} R(\tilde{\theta}, \pi) = \mathbb{E}_{q(z | \pi)} \left[ \frac{1}{N} \sum_{i=1}^N L(f(x_i; \tilde{\theta} \odot z), y_i) \right] + \lambda \sum_{j=1}^{|\theta|} \pi_j$$

we cannot optimize the first term.
Smooth the objective so we can optimize it!

Let gates $z$ be given by a hard-sigmoid rectification of $s$, as follows

$$z = g(s) = \min(1, \max(0, s)), \quad s \sim q_\phi(s)$$

The probability of a gate being active is

$$q_\phi(z \neq 0) = 1 - Q_\phi(s \leq 0)$$

Then using the reparametrization trick on $s = f(\phi, \epsilon)$ so $z = g(f(\phi, \epsilon))$

$$\min_{\tilde{\theta}, \phi} \mathbb{E}_{p(\epsilon)} \left[ \frac{1}{N} \sum_{i=1}^{N} L(f(x_i; \tilde{\theta} \odot g(f(\phi, \epsilon))), y_i) \right] + \lambda \sum_{j=1}^{\theta} 1 - Q_\phi(s_j \leq 0)$$

Okay but which distribution $q_\phi(s)$ should we use?
An appropriate smoothing distribution \( q(s) \) is the binary concrete rv \( s \):

\[
\begin{align*}
  u & \sim U(0, 1), \quad s = \text{Sigmoid} \left( \frac{\log u - \log(1 - u) + \log \alpha}{\beta} \right) \\
  \bar{s} & = s(\zeta - \gamma) + \gamma \quad \text{and} \quad z = \min(1, \max(0, \bar{s}))
\end{align*}
\]

1. \( s \) is a concrete binary distributed
2. \( \alpha \) is the location parameter, and
3. \( \beta \) is the temperature parameter
4. \( z \) is the hard concrete distribution.
5. we stretch \( s \rightarrow \bar{s} \) into the range \((\gamma, \zeta)\) where \( \zeta < 0 \) and \( \gamma > 1 \).
Figure 2: (a) The binary concrete distribution with location $\log \alpha = 0$ and temperature $\beta = 0.5$ and the hard concrete equivalent distribution obtained by stretching the concrete distribution to $(\gamma = -0.1, \zeta = 1.1)$ and then applying a hard-sigmoid. Under this specification the hard concrete distribution assigns, roughly, half of its mass to $\{0, 1\}$ and the rest to $(0, 1)$. (b) The expected value of the aforementioned concrete and hard concrete gate as a function of the location $\log \alpha$, obtained by averaging 10000 samples. We also added the value of the gates obtained by removing the noise entirely. We can see that the noise smooths the hard-sigmoid to a sigmoid on average.

Figure: Figure 2 from lousizos et al
From earlier, we had $1 - Q_\phi(s \leq 0)$ in $L_0$ complexity loss of the objective function. Now, if the random variable is hard concrete, then we can say:

$$1 - Q_\phi(s \leq 0) = \text{Sigmoid}(\log \alpha - \beta \log \frac{-\gamma}{\zeta})$$

During test time, the authors use the following for the gate:

$$\hat{z} = \min(1, \max(0, \text{Sigmoid}(\log \alpha)(\zeta - \gamma) + \gamma)) \text{ and } \theta^* = \tilde{\theta}^* \odot \hat{z}$$
## Experiments - MNIST Classification and Sparsification

<table>
<thead>
<tr>
<th>Network &amp; size</th>
<th>Method</th>
<th>Pruned architecture</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP 784-300-100</td>
<td>Sparse VD (Molchanov et al., 2017)</td>
<td>512-114-72</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>BC-GNJ (Louizos et al., 2017)</td>
<td>278-98-13</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>BC-GHS (Louizos et al., 2017)</td>
<td>311-86-14</td>
<td>1.8</td>
</tr>
<tr>
<td>( L_{0_{hc}}, \lambda = 0.1/N )</td>
<td></td>
<td>219-214-100</td>
<td>1.4</td>
</tr>
<tr>
<td>( L_{0_{hc}}, \lambda \text{ sep.} )</td>
<td></td>
<td>266-88-33</td>
<td>1.8</td>
</tr>
<tr>
<td>LeNet-5-Caffe 20-50-800-500</td>
<td>Sparse VD (Molchanov et al., 2017)</td>
<td>14-19-242-131</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>GL (Wen et al., 2016)</td>
<td>3-12-192-500</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>SBP (Neklyudov et al., 2017)</td>
<td>3-18-284-283</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>BC-GNJ (Louizos et al., 2017)</td>
<td>8-13-88-13</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>BC-GHS (Louizos et al., 2017)</td>
<td>5-10-76-16</td>
<td>1.0</td>
</tr>
<tr>
<td>( L_{0_{hc}}, \lambda = 0.1/N )</td>
<td></td>
<td>20-25-45-462</td>
<td>0.9</td>
</tr>
<tr>
<td>( L_{0_{hc}}, \lambda \text{ sep.} )</td>
<td></td>
<td>9-18-65-25</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Experiments - MNIST Classification

**Figure:** Expected FLOPs. Left is the MLP. Right is the LeNet-5
## Experiments - CIFAR Classification

<table>
<thead>
<tr>
<th>Network</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>original-ResNet-110 (He et al., 2016a)</td>
<td>6.43</td>
<td>25.16</td>
</tr>
<tr>
<td>pre-act-ResNet-110 (He et al., 2016b)</td>
<td>6.37</td>
<td>-</td>
</tr>
<tr>
<td>WRN-28-10 (Zagoruyko &amp; Komodakis, 2016)</td>
<td>4.00</td>
<td>21.18</td>
</tr>
<tr>
<td>WRN-28-10-dropout (Zagoruyko &amp; Komodakis, 2016)</td>
<td>3.89</td>
<td>18.85</td>
</tr>
<tr>
<td>WRN-28-10-$L_{0_{hc}}$, $\lambda = 0.001/N$</td>
<td>3.83</td>
<td>18.75</td>
</tr>
<tr>
<td>WRN-28-10-$L_{0_{hc}}$, $\lambda = 0.002/N$</td>
<td>3.93</td>
<td>19.04</td>
</tr>
</tbody>
</table>
Figure: Expected FLOPs of WRN at CIFAR 10 (left) & 100 (right)
Discussion & Future Work

Discussion

1. $L_0$ penalty can save memory and computation
2. $L_0$ regularization lead to competitive predictive accuracy and stability

Future Work

1. Adopt a full Bayesian treatment over the parameter $\theta$
Thank You . . .