Improving Batch Normalization with Skewness Reduction for Deep Neural Networks

Pak Lun Kevin Ding, Sarah Martin and Baoxin Li
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
Motivation

- BN layer normalizes the batch input to zero mean and unit variance
  - Smoother loss landscape
  - Faster convergence
- Making the distributions of the features in the same layer more similar would make the network perform better
  - The third moment, Skewness
    - More non-linearity
**Definition 2.** Let $\varphi_p : \mathbb{R} \to \mathbb{R}$ be a function, the skewness correction function are defined as follows:

$$
\varphi_p(x) = \begin{cases} 
  x^p & \text{if } x \geq 0 \\
  -(-x)^p & \text{if } x < 0
\end{cases}
$$

where $p > 1$. 
Batch Normalization with Skewness Reduction

**Algorithm 1:** Training stage of BNSR, applied to features $x$ over a mini-batch

**Input**: Values of $x$ over a mini-batch:  
$\mathcal{B} = \{x_{1...m}\}$

**Parameters**: Parameters to be learned: $\gamma, \beta$

**Output**: $y_i = \text{BN}_{\gamma,\beta}(x_i)$

1. $\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$
2. $\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$
3. $\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}}$
4. $\hat{x}_i \leftarrow \varphi_p(\hat{x}_i)$
5. $y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BNSR}_{\gamma,\beta}(x_i)$
Batch Normalization with Skewness Reduction

**Algorithm 2:** Testing stage of BNSR, applied to features $x$ over a mini-batch

**Input** : Values of $x$ over a mini-batch: $\mathcal{B} = \{x_1...m\}$

**Output** : $y_i = \text{BN}_{\gamma, \beta}(x_i)$

1. Calculate the population $\mu, \sigma$ by unbiased estimation or exponential moving average
2. for $i = 1...m$ do
3.     $\hat{x}_i \leftarrow \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$
4.     $\hat{x}_i \leftarrow \varphi_p(\hat{x}_i)$
5. end
6. $y_i = \gamma \hat{x}_i + \beta$
Experiments

• Determine $p$
  • VGG-19 on CIFAR-100, $p$ in $\{1.01, 1.02, 1.03, 1.04, 1.05\}$
• Impact of the similarity of the feature distributions

$\begin{align*}
  &x \leftarrow x \text{ (identity mapping)} \\
  &x \leftarrow ax + b \text{ where } a, b \sim N_m(0, 0.5) \\
  &x \leftarrow \varphi_p(x) \text{ where } p \sim \text{Unif}_{m}(1, 1.05) \\
  &x \leftarrow \varphi_p(x) \text{ where } p = 1.01
\end{align*}$

<table>
<thead>
<tr>
<th></th>
<th>BNSR</th>
<th>BN</th>
<th>Noise($\mu, \sigma$)</th>
<th>Noise($\rho$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>30.61</td>
<td>31.35</td>
<td>33.52</td>
<td>32.1</td>
</tr>
</tbody>
</table>

**TABLE I**
Comparison of error rates (%) of BNSR, BN, BN with noisy mean and variance, BN with noisy skewness on CIFAR-100. The training loss and error rate curves are in Fig. 2
Experiments

• Features in the earlier layers
  • Analyze where BNSR is more effective
    • BNSR is used for all layers
    • BNSR is used only for the earlier layers
    • BNSR is used only for the later layers

<table>
<thead>
<tr>
<th></th>
<th>100%</th>
<th>33%(uni)</th>
<th>33%(early)</th>
<th>33%(late)</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>23.49</td>
<td>23.40</td>
<td>23.74</td>
<td>25.20</td>
</tr>
</tbody>
</table>

**TABLE III**
Comparison of error rates (%) of BNSR under different percentage of usage on CIFAR-100. The training loss and testing error plots can be found in Fig. 4.
Experiments

• Comparison with other normalization schemes

<table>
<thead>
<tr>
<th></th>
<th>BNSR</th>
<th>BN</th>
<th>LN</th>
<th>IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>23.49</td>
<td>25.51</td>
<td>39.78</td>
<td>28.72</td>
</tr>
</tbody>
</table>

**TABLE II**

Comparison of error rates (%) of BNSR, BN, LN, IN on CIFAR-100. The training loss and error rate curves are in Fig. 3.

<table>
<thead>
<tr>
<th></th>
<th>BNSR</th>
<th>BN</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>22.73</td>
<td>23.17</td>
</tr>
</tbody>
</table>

**TABLE IV**

Comparison of error rates (%) of BN and BNSR on ImageNet dataset. The training loss and error rate curves are in Fig. 7.