MaxDropout: Deep Neural Network Regularization Based on Maximum Output Values

Recogna Laboratory

Claudio Filipi Gonçalves dos Santos, Mateus Roder, João Paulo Papa

December 9, 2020

Universidade Federal de São Carlos (UFSCar)
São Carlos, SP - Brasil

Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP)
Bauru, SP - Brasil
Summary

1. Introduction

2. Proposed Approach
   2.1 Algorithm
   2.2 Experiments

3. Results

4. Conclusão
Introduction
Regularization methods have been used since the very beginning of Deep Learning approaches.

LeNet-5 was trained using artificial data augmentation.

Goal: prevent neural networks to overfit.

There are different ways to perform regularization nowadays.
Introduction

Some well-known regularization:

- Data augmentation
  - AutoAugmentation [1]
  - Cutout [2]
  - RandomErasing [7]
- Change neurons value during training
  - Dropout [5]
  - Variational Dropout [4]
We performed a deeper study on Dropout

Hypothesis: could a neural network using rules on dropping values generalize better?
Proposed Approach
Presenting MaxDropout

A regularizer that works by dropping the more active (higher values) during training!
Pseudocode for MaxDropout training algorithm.

```
while training do
    for each layer do
        rate ← U(0, r)
        normTensor ← L2Normalize(Tensor)
        max ← Max(normTensor)
        Kept_Idx ← Idx.Of(normTensor, (1 − rate) * max)
        returnTensor ← Tensor * Kept_Idx
    end for
end while
```
Simulation

Figure 1: (a) Original grayscale image and its outcomes after (b) Dropout and (c) MaxDropout transformations, respectively, and (d) original colored image and its outcomes after (e) Dropout and (f) MaxDropout, respectively (dropout rate of 50%).
Three direct comparisons:

- Comparison against other regularizers
- Comparison along other regularizers
- Comparison against Dropout[5]
- Ablation

All experiments were performed on CIFAR-10 and CIFAR-100
Results
Results - Training Evolution

**Figure 2:** Convergence over CIFAR-10 test set.

**Figure 3:** Convergence over CIFAR-100 test set.
## Results - Comparison Against Other Regularizers

<table>
<thead>
<tr>
<th>Approach</th>
<th>CIFAR-100</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet18 [3, 7]</td>
<td>24.50 ± 0.19</td>
<td>5.17 ± 0.18</td>
</tr>
<tr>
<td>ResNet18+RandomErasing [7]</td>
<td>24.03 ± 0.19</td>
<td>4.31 ± 0.07</td>
</tr>
<tr>
<td>ResNet18+MaxDropout</td>
<td><strong>21.94 ± 0.23</strong></td>
<td>4.63 ± 0.11</td>
</tr>
</tbody>
</table>

**Table 1:** Results of MaxDropout and other regularizers
Table 2: Results of the MaxDropout combined with Cutout.
Results - Working Along with Other Regularizers

<table>
<thead>
<tr>
<th>Model</th>
<th>CIFAR-100</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRN [6]</td>
<td>19.25</td>
<td>4.00</td>
</tr>
<tr>
<td>WRN + Dropout [6]</td>
<td>18.85</td>
<td>3.89</td>
</tr>
<tr>
<td>WRN + MaxDropout</td>
<td><strong>18.81</strong></td>
<td><strong>3.84</strong></td>
</tr>
</tbody>
</table>

**Table 3:** Results of Dropout and MaxDropout over the WRN.
### Table 4: Ablation results concerning MaxDropout over ResNet18.

<table>
<thead>
<tr>
<th>MaxDropout Rate ($r$)</th>
<th>CIFAR-100</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>22.05 ± 0.17</td>
<td>4.76 ± 0.09</td>
</tr>
<tr>
<td>10</td>
<td>22.06 ± 0.32</td>
<td>4.71 ± 0.09</td>
</tr>
<tr>
<td>15</td>
<td>22.16 ± 0.20</td>
<td><strong>4.63 ± 0.11</strong></td>
</tr>
<tr>
<td>20</td>
<td>21.99 ± 0.21</td>
<td>4.70 ± 0.08</td>
</tr>
<tr>
<td>25</td>
<td><strong>21.94 ± 0.23</strong></td>
<td>4.70 ± 0.06</td>
</tr>
<tr>
<td>30</td>
<td>22.08 ± 0.24</td>
<td>4.67 ± 0.12</td>
</tr>
<tr>
<td>35</td>
<td>22.10 ± 0.29</td>
<td>4.71 ± 0.16</td>
</tr>
<tr>
<td>40</td>
<td>22.17 ± 0.34</td>
<td>4.79 ± 0.20</td>
</tr>
<tr>
<td>45</td>
<td>22.31 ± 0.29</td>
<td>4.71 ± 0.11</td>
</tr>
<tr>
<td>50</td>
<td>22.33 ± 0.23</td>
<td>4.75 ± 0.10</td>
</tr>
</tbody>
</table>
Conclusão
Conclusion

We introduced MaxDropout, an improved version of the original Dropout method

Experiments show that:

- it can be incorporated into existing models
- it works along with other regularizers
- it can replace the standard Dropout with some accuracy improvement
Future work

Two main tasks:

- re-implement in other frameworks (TensorFlow, MXNet)
- test in other problems (such as NLP)
https://github.com/cfsantos/MaxDropout-torch
References
Autoaugment: Learning augmentation policies from data.  

Improved regularization of convolutional neural networks with cutout.  

Identity mappings in deep residual networks.  

Variational dropout and the local reparameterization trick.  
**Dropout: a simple way to prevent neural networks from overfitting.**  

**Wide residual networks.**  

**Random erasing data augmentation.**  
Questions?

Thank you all!