GradAug: A New Regularization Method for Deep Neural Networks

Taojiannan Yang, Sijie Zhu, Chen Chen
University of North Carolina at Charlotte

Introduction

- Deep neural networks are easily suffering from over-fitting. Popular regularization methods include data augmentation and structure regularization.
- Mixed sample data augmentation (MSDA) methods, such as Mixup and CutMix, achieve SOTA results. But they are hard to generalize to downstream tasks such as object detection and segmentation.
- Structure regularization methods, such as Dropout and StochDepth, are more generic. But they are not as effective as MSDA.

Motivation

- Our approach – GradAug – aims to regularize sub-networks with differently transformed training samples.

Key contributions:

- GradAug leverages the advantages of both data augmentation and structure regularization methods.
- GradAug is easy to implement and can be applied to various network structures and applications.
- GradAug significantly outperforms other state-of-the-art methods.

Method

Algorithm 1 Gradient Augmentation (GradAug)

> Train full-network.
> Forward pass, $output_{f} = Net(img)$
> Compute loss, $loss_{f} = criterion(output_{f}$, target $)$
> Regu-larize sub-networks.

for $i$ in range($n$) do
  Sample a sub-network, $subnet_{i} = Sample(Net,a)$
  Fix BN layer’s mean and variance, $variance, subnet_{i},track_running_stats = False$
  Forward pass with transformed images, $output_{i} = subnet_{i}(T(img))$
  Compute loss with soft labels, $loss_{i} = criterion(output_{i}$, $output_{f}$)
end for
Compute total loss, $L = loss_{f} + \sum_{i=1}^{n} loss_{i}$
Compute gradients and do backward pass

Math 1

$$L_{GA} = l(\theta, x, y) + \sum_{i=1}^{n} l(\theta_{w_{i}}, T^{i}(x), N(\theta, x))$$

$$g_{GA} = \frac{\partial l(\theta, x, y)}{\partial \theta} + \sum_{i=1}^{n} \frac{\partial l(\theta_{w_{i}}, T^{i}(x), N(\theta, x))}{\partial \theta_{w_{i}}}$$

Sub-network ($w=0.9$) Full-network

Advantages:

- GradAug leverages the advantages of both data augmentation and structure regularization methods.
- GradAug is easy to implement and can be applied to various network structures and applications.
- GradAug significantly outperforms other state-of-the-art methods.

Acknowledgement: This work is partially supported by the National Science Foundation (NSF) under Grant No. 1910844 and NSF/Intel Partnership on MLWiNS under Grant No. 2003198.

Experiments

<table>
<thead>
<tr>
<th>Model</th>
<th>FLOPs</th>
<th>Accuracy Top-1 (%)</th>
<th>Top-5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 [2]</td>
<td>4.1 G</td>
<td>76.32</td>
<td>92.95</td>
</tr>
<tr>
<td>ResNet-50 + Cutout [10]</td>
<td>4.1 G</td>
<td>77.07</td>
<td>93.34</td>
</tr>
<tr>
<td>ResNet-50 + Dropblock [18]</td>
<td>4.1 G</td>
<td>78.13</td>
<td>94.02</td>
</tr>
<tr>
<td>ResNet-50 + Mixup [12]</td>
<td>4.1 G</td>
<td>77.9</td>
<td>93.9</td>
</tr>
<tr>
<td>ResNet-50 + CutMix [13]</td>
<td>4.1 G</td>
<td>78.60</td>
<td>94.08</td>
</tr>
<tr>
<td>ResNet-50 + StochDepth [15]</td>
<td>4.1 G</td>
<td>77.53</td>
<td>93.73</td>
</tr>
<tr>
<td>ResNet-50 + Dropout [16]</td>
<td>4.1 G</td>
<td>77.10</td>
<td>93.50</td>
</tr>
<tr>
<td>ResNet-50 + ShakeDrop [22]</td>
<td>4.1 G</td>
<td>77.5</td>
<td>94.0</td>
</tr>
<tr>
<td>ResNet-50 + GradAug (Ours)</td>
<td>4.1 G</td>
<td>78.79</td>
<td>94.38</td>
</tr>
<tr>
<td>ResNet-50 + bag of tricks [28]</td>
<td>4.3 G</td>
<td>79.29</td>
<td>94.63</td>
</tr>
<tr>
<td>ResNet-50 + GradAug (Ours)</td>
<td>4.1 G</td>
<td>79.67</td>
<td>94.93</td>
</tr>
</tbody>
</table>

ImageNet Classification

- Low Data Regime
  - Cifar-10
  - STL-10

- Adversarial

- Code: https://github.com/taoyang1122/GradAug