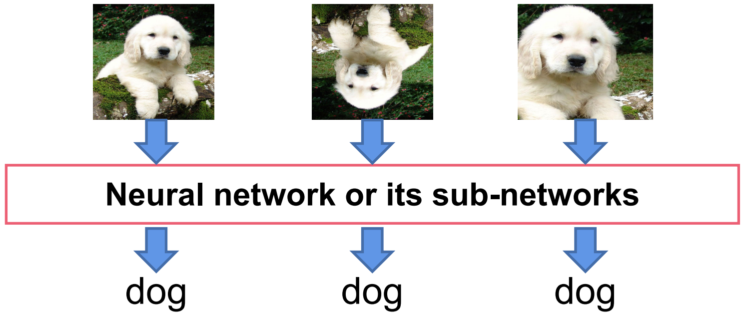
GradAug: A New Regularization Method for Deep Neural Networks

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.

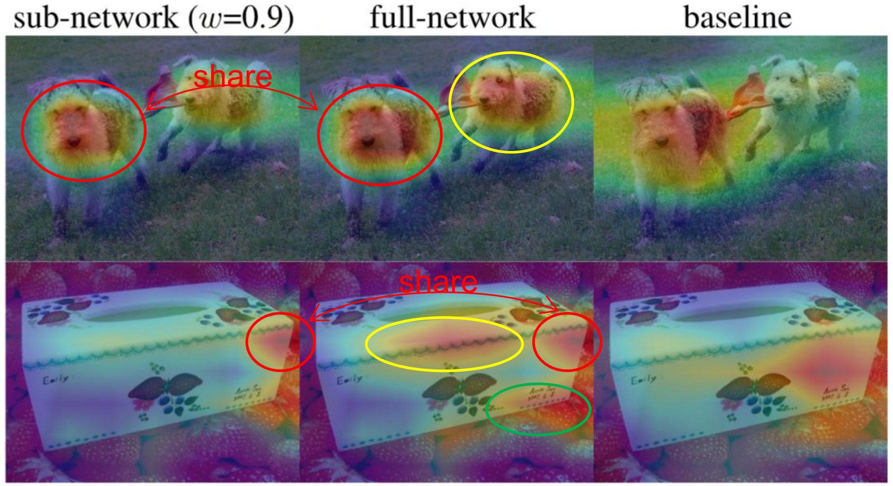
Method

Algorithm 1 Gradient Augmentation (GradAug)

Input: Network Net. Training image img. Random transformation T. Number of sub-networks n. Sub-network width lower bound α.
▷ Train full-network.
Forward pass, $output_f = Net(img)$
Compute loss, $loss_f = criterion(output, target)$
▷ Regularize sub-networks.
for i in $range(n)$ **do**
  Sample a sub-network, $subnet_i = Sample(Net, \alpha)$
  Fix BN layer's mean and variance, $subnet_i.track\_running\_stats = False$
  Forward pass with transformed images, $output_i = subnet_i(T^i(img))$
  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

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

$$g_{GA} = \underbrace{\frac{\partial l(N(\theta, x), y)}{\partial \theta}}_{\text{raw gradients}} + \underbrace{\sum_{i=1}^{n} \frac{\partial l(N(\theta_{w_i}, T^i(x)), N(\theta, x))}{\partial \theta_{w_i}}}_{\text{augmentation to raw gradients}}$$



Experiments

ImageNet Classification

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

COCO Det. and Seg.

Model	ImageNet Cls Acc (%)	Det mAP	Seg mAP
ResNet-50 (Baseline)	76.3 (+0.0)	36.5 (+0.0)	33.3 (+0.0)
Mixup-pretrained	77.9 (+1.6)	35.9 (-0.6)	32.7 (-0.6)
CutMix-pretrained	78.6 (+2.3)	36.7 (+0.2)	33.4 (+0.1)
GradAug-pretrained	**78.8 (+2.5)**	**37.7 (+1.2)**	**34.5 (+1.2)**
GradAug	**78.8 (+2.5)**	**38.2 (+1.7)**	**35.4 (+2.1)**

Low Data Regime

Model	Cifar-10			STL-10
# training labels →	250	1000	4000	1000
WideResNet-28-2	45.23±1.01	64.72±1.18	80.17±0.68	67.62±1.06
+ CutMix (p=0.5)	43.45±1.98	63.21±0.73	80.28±0.26	67.91±1.15
+ CutMix (p=0.1)	43.98±1.15	64.60±0.86	82.14±0.65	69.34±0.70
+ ShakeDrop	42.01±1.77	63.11±1.22	79.62±0.77	66.51±0.99
+ GradAug	**50.11±1.21**	**70.39±0.82**	**83.69±0.51**	**70.42±0.81**
+ GradAug-semi	**52.95±2.15**	**71.74±0.77**	84.11±0.25	**70.86±0.71**
Mean Teacher [36]	48.41±1.01	65.57±0.83	**84.13±0.28**	-

Adversarial Attack

Model	$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$	$\epsilon = 0.25$
ResNet-50	27.90	22.65	19.50	17.04	15.09
+ Cutout	27.22	21.55	17.51	14.68	12.37
+ Mixup	30.76	25.59	21.63	18.44	16.19
+ CutMix	37.73	33.42	29.69	26.29	23.26
+ GradAug	36.51	31.44	27.70	24.93	22.33
+ GradAug†	**40.26**	**35.18**	**31.36**	**28.04**	**25.12**

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.

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