A Unified Analysis of Mixed Sample Data Augmentation: A Loss Function Perspective

Chanwoo Park*[1], Sangdoo Yun*[2], and Sanghyuk Chun[2]

NeurIPS 2022

October, 2022

[1]MIT EECS
[2]NAVER AI Lab

Outline

Question

What fundamentally makes the difference between Mixup and CutMix?

- Both empirical examples (CutMix outperforms Mixup or Mixup outperforms CutMix) exist.
- Sometimes alternating Mixup and CutMix gives good results.
- Why???

Only Mixup has been analyzed theoretically.

- Zhang et al., 2021, Chidambaram et al., 2021, Carratino et al., 2021, Zhang et al., 2021

Contribution

- We show that Mixed Sample Data Augmentation (MSDA) behaves as an **input gradient and Hessian regularization** as well as a regularizer for **the first layer parameters**, MSDA improves **adversarial robustness, and generalization**.
- From our unified theoretical lens for MSDA, we can conclude that *there is no one-fit-all optimal MSDA fit to every data or model parameter*.
- New methods from the theoretical intuition, **HMix and GMix** outperforms other MSDA method in several setup.

Our Answer

Different loss function for Mixup and CutMix (especially regularization term) induces the performance difference.

- CutMix gives **a strong regularization in the product of nearby distance pixel-level** partial gradient and nearby distance Hessian of the estimated function f, while CutMix gives a weak regularization in the product of long-distance pixel-level partial gradient and long-distance Hessian of the estimated function f.
- In contrast, Mixup gives a regularization in gradient or Hessian of the estimated function f **regardless of the pixel-level distance**.

Formal Definition of MSDA

$\tilde{x}_{i,j}^{(\mathsf{MSDA})}(\lambda, 1-\lambda) = M(\lambda) \odot x_i + (1 - M(\lambda)) \odot x_j$ and
$\tilde{y}_{i,j}^{(\mathsf{MSDA})}(\lambda, 1-\lambda) = \lambda \odot y_i + (1-\lambda) \odot y_j$.

Our analysis

- $\mathbb{E}[M(\lambda)] = \lambda\vec{1}$.
- M (conditioned on λ) is determined by sample space $\mathcal{W}$.
- Formally, $M : \mathcal{W} \times \Lambda \to \mathbb{R}^s$ is a measurable function. (s is image size (i.e. 224×224)

Outline

Loss function of MSDA

Theorem

Defining MSDA Loss as

$$L_m^{MSDA}(\theta) = \mathbb{E}_{i,j\sim \mathit{Unif}([m])}\mathbb{E}_{\lambda\sim\mathcal{D}_\lambda}\mathbb{E}_M l(\theta, \tilde{z}_{i,j}^{(MSDA)}(\lambda, 1-\lambda)),$$

we can rewrite the MSDA loss ($\lim_{a\to 0}\varphi(a) = 0$*) as*

$$L_m^{MSDA}(\theta) = L_m(\theta) + \sum_{i=1}^{3}\mathcal{R}_i^{(MSDA)}(\theta) + \mathbb{E}_{\lambda\sim\tilde{\mathcal{D}}(\lambda)}\mathbb{E}_M[(1-M)^\intercal\varphi(1-M)(1-M)],$$

where $\mathcal{R}_2$ regularizes the gradient, $\mathcal{R}_3$ regularizes the Hessian.

What is different?

We want some intuition from difference between Mixup and CutMix's loss function.

$\mathcal{R}_2$ **term:**

$$\mathbb{E}_{\tilde{D}_\lambda, M}(1-M)^\intercal \mathbb{E}_{r_x \sim \mathcal{D}_X}\left(\partial f(x_i) \odot (r_x - x_i)\left(\partial f(x_i) \odot (r_x - x_i)\right)^\intercal\right)(1-M)$$

$\mathcal{R}_3$ **term:**

$$\mathbb{E}_{\tilde{D}_\lambda, M}\mathbb{E}_M(1-M)^\intercal \mathbb{E}_{r_x \sim \mathcal{D}_X}\left(\partial^2 f_\theta(x_i) \odot \left((r_x - x_i)(r_x - x_i)^\intercal\right)\right)(1-M)$$

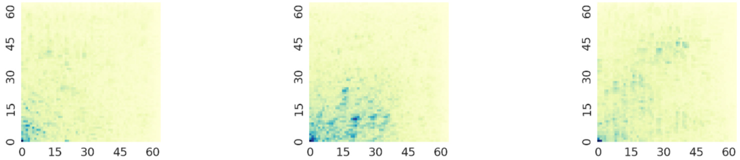
- Put $M = \lambda\vec{1}$: Mixup. Same regularization in all j, k
- Under CutMix, Strong regularization in $\partial_j f(x_i)\partial_k f(x_i)$ or $\partial^2_{j,k} f(x_i)$ if j and k are close.

Other Theoretical Results

- Loss can be interpretated with a regularization of the first layer parameters and their partial derivatives.
- We can make MSDA that having desired regularizing condition under the regularity condition.
- MSDA gives adversarial robustness and generalization property.

Outline

Comparison in terms of the regularized input gradients after MSDA training.



(a) Vanilla (no MSDA) (b) Mixup (c) CutMix

Figure 4: **Regularized input gradients by MSDA.** The normalized pixel-wise partial gradient norm product comparison among the models trained with vanilla setting (a), Mixup (b) and CutMix (c).

- As different MSDA methods regularize the input gradients $\partial_j f \partial_k f$ differently, we visualize the input gradients after training by different MSDA methods. $(\max_k \max_v |\partial_v f_\theta(x_k)\partial_{v+p} f_\theta(x_k)|)$

Understanding application cases when a specific MSDA design choice works better than others.

Scenario 1: Smaller objects by large crop size.

- randomly crop a large region of an image
- As the objects in the image become small, a close-distance relationship might be more important than a large-distance relationship.
- CutMix $>$ Mixup

Scenario 2: Larger objects by small crop size.

- randomly crop a small region of an image
- the objects in the image would become large in the cropping region and the large-distance relationship might be important.
- CutMix $<$ Mixup

New methods and Method comparison

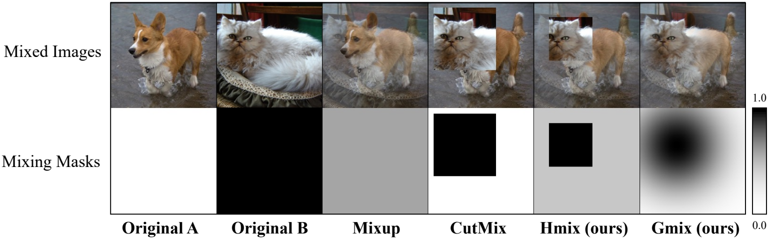


Figure: **Examples generated by different MSDAs.** From left to right, two original images to be mixed, Mixup, CutMix sample, HMix, and GMix. The first and the second rows show generated samples and their mixing masks M, respectively. We set $\lambda = 0.65$ for all images and $r = 0.5$ for HMix.

Results

Table: **Different tasks need different MSDA strategies.** Validation accuracies of Mixup and CutMix trained networks on two different scenarios on ImageNet-100. Each scenario assumes different pixel importances.

	Mixup	CutMix	Δ (CutMix - Mixup)
Scenario 1: Large crop	58.3	**64.4**	**+6.1**
Scenario 2: Small crop	**67.7**	67.0	**-0.7**

Outline

CIFAR-100 classification

Table 2: **CIFAR-100 classification.** Comparison of various MSDA methods on various network architectures. Note that PuzzleMix needs additional computations (twice than others) for computing the input saliency.

Augmentation Method	RN56	WRN28-2	PreActRN18	PreActRN34	PreActRN50
Vanilla (no MDSA)	73.23	73.50	76.73	77.68	79.07
Mixup	73.12	74.05	77.21	79.02	79.34
CutMix	74.83	74.79	78.66	80.05	81.23
PuzzleMix	-	76.51	79.38	80.89	82.46
Stochastic Mixup & CutMix	74.88	75.49	**79.25**	81.05	81.21
HMix (ours)	74.99	75.68	**79.25**	**81.07**	81.38
GMix (ours)	**75.75**	**76.15**	79.17	80.52	**81.45**

ImageNet-1K classification

Table 3: **ImageNet-1K classification.** Comparison of various MSDA methods on ResNet-50 architecture.

Augmentation Method	Top-1 accuracy
Vanilla (no MDSA)	75.68 (+0.00)
Mixup	77.78 (+2.10)
CutMix	78.04 (+2.36)
Stochastic Mixup & CutMix	78.13 (+2.45)
HMix (ours)	**78.38** (+2.70)
GMix (ours)	78.13 (+2.45)