mixup: BEYOND EMPIRICAL RISK MINIMIZATION

Abstract

regularizes the neural network to favor simple linear behavior
reduces the memorization of corrupt labels
increases the robustness to adversarial examples
stabilizes the training of generative adversarial networks

Introduction

these neural networks share two commonalities
1 minimize their average error over the training data
2 size of networks scales linearly with the number of training examples

convergence of ERM is guaranteed as long as the size of the learning machine does not increase with the number of training data
ERM allows large neural networks to memorize (instead of generalize)
-> predictions drastically when evaluated on examples just outside the training distribution

Vicinal Risk Minimization
human knowledge is required to describe a vicinity or neighborhood around each example
additional virtual examples can be drawn from the vicinity distribution of the training examples
ex) horizontal reflections, slight rotations, and mild scalings
-> dataset-dependent, and thus requires the use of expert knowledge
does not model the vicinity relation across examples of different classes.

Contribution

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \qquad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \qquad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

Virtual Training Examples

a new state-of-the-art performance
increases the robustness of neural networks

From Empirical Risk Minimization to Mixup

$$\mu(\tilde{x}, \tilde{y}|x_i, y_i) = \frac{1}{n}\sum_{j}^{n} \mathop{\mathbb{E}}_{\lambda} \left[\delta(\tilde{x} = \lambda \cdot x_i + (1-\lambda)\cdot x_j, \tilde{y} = \lambda \cdot y_i + (1-\lambda)\cdot y_j)\right],$$

implementation of mixup training is straightforward, and introduces a minimal computation overhead

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j,$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j,$$

1 combinations of three or more examples with weights sampled from a Dirichlet distribution does not provide further gain,

2 single data loader to obtain one minibatch,

3 interpolating only between inputs with equal label did not lead to the performance gains

What is mixup doing?

Model	Method	Epochs	Top-1 Error	Top-5 Error
ResNet-50	ERM (Goyal et al., 2017)	90	23.5	-
	mixup $\alpha = 0.2$	90	**23.3**	**6.6**
ResNet-101	ERM (Goyal et al., 2017)	90	22.1	-
	mixup $\alpha = 0.2$	90	**21.5**	**5.6**
ResNeXt-101 32*4d	ERM (Xie et al., 2016)	100	21.2	-
	ERM	90	21.2	5.6
	mixup $\alpha = 0.4$	90	**20.7**	**5.3**
ResNeXt-101 64*4d	ERM (Xie et al., 2016)	100	20.4	5.3
	mixup $\alpha = 0.4$	90	**19.8**	**4.9**
ResNet-50	ERM	200	23.6	7.0
	mixup $\alpha = 0.2$	200	**22.1**	**6.1**
ResNet-101	ERM	200	22.0	6.1
	mixup $\alpha = 0.2$	200	**20.8**	**5.4**
ResNeXt-101 32*4d	ERM	200	21.3	5.9
	mixup $\alpha = 0.4$	200	**20.1**	**5.0**

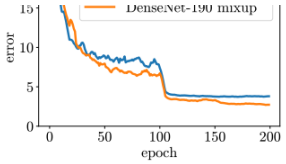
Table 1: Validation errors for ERM and *mixup* on the development set of ImageNet-2012.

model trained with mixup is more stable in terms of model predictions and gradient norms in-between training samples.

Experiments

Dataset	Model	ERM	*mixup*

CIFAR-10	PreAct ResNet-18	5.6	**4.2**
	WideResNet-28-10	3.8	**2.7**
	DenseNet-BC-190	3.7	**2.7**
CIFAR-100	PreAct ResNet-18	25.6	**21.1**
	WideResNet-28-10	19.4	**17.5**
	DenseNet-BC-190	19.0	**16.8**



Model	Method	Validation set	Test set
LeNet	ERM	**9.8**	**10.3**
	mixup ($\alpha = 0.1$)	10.1	10.8
	mixup ($\alpha = 0.2$)	10.2	11.3
VGG-11	ERM	5.0	4.6
	mixup ($\alpha = 0.1$)	4.0	3.8
	mixup ($\alpha = 0.2$)	**3.9**	**3.4**

Label corruption	Method	Test error		Training error	
		Best	Last	Real	Corrupted
20%	ERM	12.7	16.6	0.05	0.28
	ERM + dropout ($p = 0.7$)	8.8	10.4	5.26	83.55
	mixup ($\alpha = 8$)	**5.9**	6.4	2.27	86.32
	mixup + dropout ($\alpha = 4, p = 0.1$)	6.2	**6.2**	1.92	85.02
50%	ERM	18.8	44.6	0.26	0.64
	ERM + dropout ($p = 0.8$)	14.1	15.5	12.71	86.98
	mixup ($\alpha = 32$)	11.3	12.7	5.84	85.71
	mixup + dropout ($\alpha = 8, p = 0.3$)	**10.9**	**10.9**	7.56	87.90
80%	ERM	36.5	73.9	0.62	0.83
	ERM + dropout ($p = 0.8$)	30.9	35.1	29.84	86.37
	mixup ($\alpha = 32$)	25.3	30.9	18.92	85.44
	mixup + dropout ($\alpha = 8, p = 0.3$)	**24.0**	**24.8**	19.70	87.67

Metric	Method	FGSM	I-FGSM
Top-1	ERM	90.7	99.9
	mixup	**75.2**	99.6
Top-5	ERM	63.1	93.4
	mixup	**49.1**	95.8

(a) White box attacks.

Metric	Method	FGSM	I-FGSM
Top-1	ERM	57.0	57.3
	mixup	**46.0**	**40.9**
Top-5	ERM	24.8	18.1
	mixup	**17.4**	**11.8**

(b) Black box attacks.

TABULAR DATA

Dataset	ERM	*mixup*
Abalone	74.0	73.6
Arcene	57.6	**48.0**
Arrhythmia	56.6	**46.3**

Dataset	ERM	*mixup*
Htru2	2.0	2.0
Iris	21.3	**17.3**
Phishing	16.3	15.2

Table 4: ERM and *mixup* classification errors on the UCI datasets.

STABILIZATION OF GENERATIVE ADVERSARIAL NETWORKS (GANS)

$$\max_g \min_d \mathop{\mathbb{E}}_{x,z,\lambda} \ell(d(\lambda x + (1-\lambda)g(z)), \lambda).$$

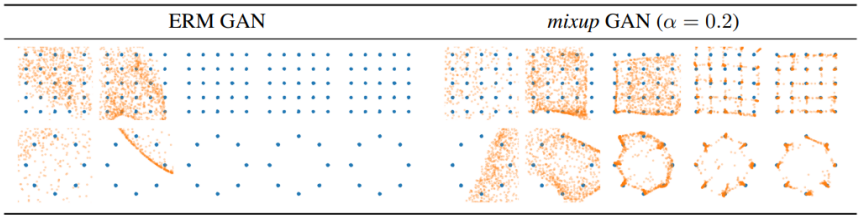


Figure 5: Effect of *mixup* on stabilizing GAN training at iterations 10, 100, 1000, 10000, and 20000.

ABLATION STUDIES

Method	Specification	Modified		Weight decay	
		Input	Target	10^{-4}	5×10^{-4}
ERM		✗	✗	5.53	5.18
mixup	AC + RP	✓	✓	**4.24**	4.68
	AC + KNN	✓	✓	4.98	5.26
mix labels and latent representations (AC + RP)	Layer 1	✓	✓	4.44	**4.51**
	Layer 2	✓	✓	4.56	4.61
	Layer 3	✓	✓	5.39	5.55
	Layer 4	✓	✓	5.95	5.43
	Layer 5	✓	✓	5.39	5.15
mix inputs only	SC + KNN (Chawla et al., 2002)	✓	✗	5.45	5.52
	AC + KNN	✓	✗	5.43	5.48
	SC + RP	✓	✗	5.23	5.55
	AC + RP	✓	✗	5.17	5.72
label smoothing (Szegedy et al., 2016)	$\epsilon = 0.05$	✗	✓	5.25	5.02
	$\epsilon = 0.1$	✗	✓	5.33	5.17
	$\epsilon = 0.2$	✗	✓	5.34	5.06
mix inputs +	$\epsilon = 0.05$	✓	✓	5.02	5.40

label smoothing (AC + RP)	$\epsilon = 0.1$	✓	✓	5.08	5.09
	$\epsilon = 0.2$	✓	✓	4.98	5.06
	$\epsilon = 0.4$	✓	✓	5.25	5.39
add Gaussian noise to inputs	$\sigma = 0.05$	✓	✗	5.53	5.04
	$\sigma = 0.1$	✓	✗	6.41	5.86
	$\sigma = 0.2$	✓	✗	7.16	7.24

mixup =
random convex combination of raw inputs
convex combination of one-hot label encodings

ablation study로서 알게 된 점들
First, mixup is the best data augmentation method we test, and is significantly better than the second best method (mix input + label smoothing)
Second, the effect of regularization can be seen by comparing the test error with a small weight decay (10−4) with a large one (5 × 10−4).
SMOTE algorithm (Chawla et al., 2002) does not lead to a noticeable gain in performance.

Related Work

Data augmentation
rotation, translation, cropping, resizing, flipping
random erasing

in speech recognition, noise injection

mixup 단점
only operate among the nearest neighbors within a certain class at the input / feature level
does not account for changes in the corresponding labels

label smoothing
penalizing high-confidence softmax distributions
두 개의 단점
-> independent from the associated feature values

mixup enjoys several desirable aspects of previous data augmentation and regularization schemes
not require significant domain knowledge

Discussion

with increasingly large α, the training error on real data increases, while the generalization gap decreases.
do not yet have a good theory for understanding the ‘sweet spot’ of this bias-variance trade-off
increasing the model capacity would make training error less sensitive to large α -> giving mixup a more significant advantage.

several possibilities for further exploration
First, is it possible to make similar ideas work on other types of supervised learning problems, such as regression and structured prediction?
Second, can similar methods prove helpful beyond supervised learning?
ex) Can we extend mixup to feature-label extrapolation to guarantee a robust model behavior far away from the training data?