mix-up : BEYOND EMPIRICAL RISK MINIMIZATION

-Summary-

Introduction

Notation & Effect

- Example : $(x, y) \sim P(X, Y)$, where $x =$ input, y = target
- Prediction : $f(x)$, where $f \in \mathcal{F}$ is expected to satisfy $f(X) = Y$
- Set of training data : $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $(x_i, y_i) \sim P$

- Expected risk : $R(f) = \int l(f(x), y) dP(x, y)$

  (note : distribution P is unknown in practical situations)
- Empirical distribution : $P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \delta(x = x_i, y = y_i)$
- Empirical risk : $R_\delta(f) = \int l(f(x), y) dP_\delta(x, y)$ = $\frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i)$

- Set of vicinity training data = $\mathcal{D}_\nu = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{m}$ (sampled m times from vicinity distribution)
- Gaussian vicinity distribution : $\nu(\tilde{x}, \tilde{y} | x_i, y_i) = \mathcal{N}\left(\tilde{x} - x_i, \sigma^2\right) \delta(\tilde{y} = y_i)$ (Chapelle, 2000)
- Mix-up vicinity distribution : $\mu(\tilde{x}, \tilde{y} | x_i, y_i) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{E}_\lambda[\delta(\tilde{x} = \lambda x_i + (1 - \lambda) x_j, \tilde{y} = \lambda y_i + (1 - \lambda) y_j)]$, where $\lambda \sim Beta(\alpha, \alpha)$, $\alpha \in (0, \infty)$
- Empirical vicinal risk : $R_\nu(f) = \frac{1}{m} \sum_{i=1}^{m} l(f(\tilde{x}_i), \tilde{y}_i)$ (usually , $m = n^2$)

Effect : Minimizing the empirical vicinal risk with mix-up improve test error (compared to ERM)

=> generalize better and robust to adversarial attack

Experimental results – Image classification

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**

Note

- Model with higher capacities or longer training runs are the ones to benefit the most from mix-up method.

  (ex : ResNet-50 (0.2%) <-> ResNeXt-101(0.5~ 0.6%), ResNet-50 at epoch 90 / 200)

Experimental results – Speech data

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**

classification error

Note

- Data preprocess :

  Data (65,000 utterances, ~1 sec, 30 classes) => Normalized spectrogram from waveforms

  => Mix-up both at waveform and spectrogram levels => Train model (CNN-based)

- Similarly, model with higher capacity (=VGG-11) benefits the most from mix-up method.

Experimental results – Memorization of corrupted labels

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

Note

- Certain portion of labels are replaced (corrupted) by random noises.
- Dropout + mix-up shows better performance on learning corrupted labels, compared to Dropout + ERM.

Experimental results – Robustness to adversarial examples

test error

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.

Note

- One of undesirable behavior of ERM is fragility to adversarial examples
- FGSM (Fast Gradient Sign Method) : $x_{adv} = x + \epsilon\ * sign(\nabla_x l(f(x), y))$ (exploit SGD process)
- I-FGSM(Iterative-FGSM) : $x_{adv}^0 = x, x_{adv}^{N+1} = X_N^{adv} + \alpha * sign(\nabla_x l(f(x), y))$
- Black box FGSM / I-FGSM implementation :

  Produce Adversarial examples (1st ERM model) ---> test robustness of 2nd /3rd model

  (Use 3 models : 1st/ 2nd = trained using ERM / 3rd = trained using mix-up)
- Mix-up method is better than ERM in terms of robustness to adversarial examples

Experimental results – Non-image data

test error

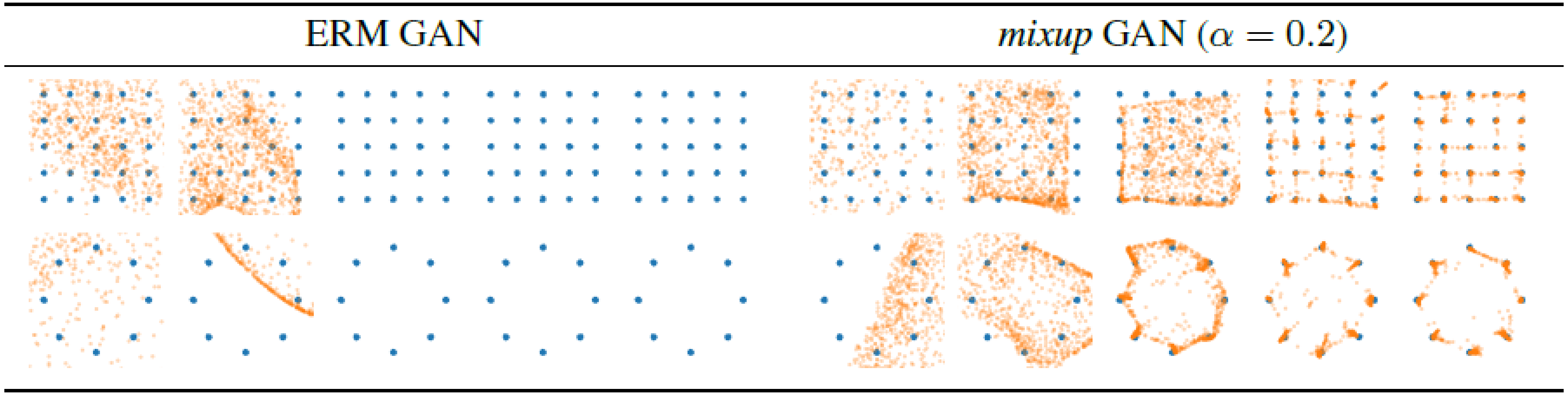
Dataset	ERM	*mixup*
Abalone	74.0	73.6
Arcene	57.6	**48.0**
Arrhythmia	56.6	**46.3**

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

Note

- Mix-up method also works well on non-image data (better than ERM, in general)

Experimental results – Stabilization of GANs



iteration : 10, 100, 1000, 10000, 20000 / blue = dataset, orange = generated samples

Note

- Original optimization of GAN : $\max_{g} \min_{d} \mathbb{E}_{x,z}\left[l(d(x),1) + l\big(d\big(g(z)\big),0\big)\right]$

  ($d(\cdot)$ = discriminator, $g(\cdot)$ = generator, l = binary cross entropy loss)

- Mix-up optimization of GAN : $\max_{g} \min_{d} \mathbb{E}_{x,z,\lambda}\left[l\big(d\big(\lambda x + (1-\lambda)g(z)\big),\lambda\big)\right]$

- Training of mix-up GANs seems promisingly robust to hyper-parameter and architectural choices.

Ablation study

AC : mix between all classes
RP : mix between random pairs
KNN : mix between k-nearest neighbors
(Here, k = 200)

median test errors of the last 10 epochs

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

Question : What is mix-up doing? (Ablation studies)

- Form of data augmentation that encourages the model f to behave linearly in-between training examples.

$\Rightarrow$ It turns out (experimentally) that mix labels + mix latent representations does not show better performance than mix labels + mix raw inputs (original mix-up)

$\Rightarrow$ As in experiments on speech data, it seems crucial to choose how to preprocess raw data and interpolate...
(can't figure out best preprocess method in practice circumstances)

Q: How to mix-up data in a good manner without background of data?

$\Rightarrow$ Mix labels + mix (trained) VAE outputs ?? (similar to mix labels + mix latent representations)