Review on Semi-Supervised Learning

Presenter: Zhe Wang

https://qdata.github.io/deep2Read

Zhe Wang

201909

Content

Semi-Supervised Learning

Semi-Supervised Learning (SSL):
Learning from both labeled and unlabeled data.

- Supervised Learning: Data sample $\{x_i, y_i\}_{i=1}^{n}$
- Unsupervised Learning: Data sample $\{x_i\}_{i=1}^{n}$
- Semi-Supervised Learning: Data sample $\{x_i, y_i\}_{i=1}^{n} + \{x_j\}_{j=1}^{u}, u >> n$

Different methods:

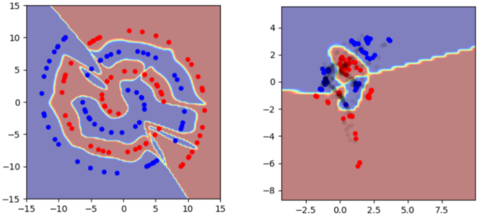
- Optimization based
- Regularization based (Entropy/Graph Reg.)
- Representation based
- Generative model based

Content

1. Definition and motivation
2. Optimization based SSL
3. Regularization based SSL
4. Generative model based SSL

MixUp

Limitations of ERM:



2D Spiral Dataset 2D Hidden Rep.

- Decision boundary is squiggly.
- Low-confident region is rather narrow.
- Decision boundary is too close to data samples
- Wired arrangements of hidden rep.
- Unseen data fall into confident region.

Solution: Data Augmentation (Vicinal Risk Minimization)

Explain: Draw samples from vicinity of existing samples to enlarge dataset.

Example:

Demonstration of sample augmentations: rotation, gaussian noise, crop, hue and saturation adjustment, elastic transform, coarse dropout

Limitation:

- Data-dependent
- Examples in the vicinity share the same class

Mixup

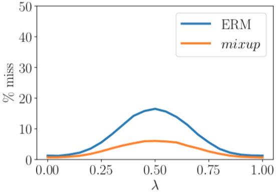
Extremely simple method: MixUp.
Generation of new data: Linear interpolation.

$$
\begin{aligned}
x &= \lambda x_1 + (1-\lambda)x_2 \\
y &= \lambda y_1 + (1-\lambda)y_2
\end{aligned}
\tag{1}
$$

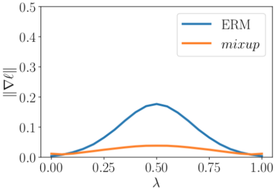
```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

(a) One epoch of *mixup* training in PyTorch.

Intuition behind:
Encourage model f to behave linearly in-between training examples. Linear behaviors reduces the amount of undesirable oscillations on OOD samples.



(a) Prediction errors in-between training data. Evaluated at $x = \lambda x_i + (1-\lambda) x_j$, a prediction is counted as a "miss" if it does not belong to $\{y_i, y_j\}$. The model trained with *mixup* has fewer misses.



(b) Norm of the gradients of the model w.r.t. input in-between training data, evaluated at $x = \lambda x_i + (1-\lambda) x_j$. The model trained with *mixup* has smaller gradient norms.

Experiments and ablation study

Different tasks on different datasets.

- classification on ImageNet, CIFAR-10 and CIFAR-100.
- speech recognition on Google Command dataset
- robustness against corrupted labels.
- robustness against adversarial examples.
- stabilization of GAN

Ablation study:

- interpolate the latent representations.
- interpolate only between the nearest neighbors.
- between inputs of the same/different class
- label smoothing

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.

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

Table 2: Results on the corrupted label experiments for the best models.

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.

Table 3: Classification errors of ERM and *mixup* models when tested on adversarial examples.

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 + label smoothing (AC + RP)	$\epsilon = 0.05$	✓	✓	5.02	5.40
	$\epsilon = 0.1$	✓	✓	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

Table 5: Results of the ablation studies on the CIFAR-10 dataset. Reported are the median test errors of the last 10 epochs. A tick (✓) means the component is different from standard ERM training, whereas a cross (✗) means it follows the standard training practice. AC: mix between all classes. SC: mix within the same class. RP: mix between random pairs. KNN: mix between k-nearest neighbors (k=200). Please refer to the text for details about the experiments and interpretations.

Manifold MixUp

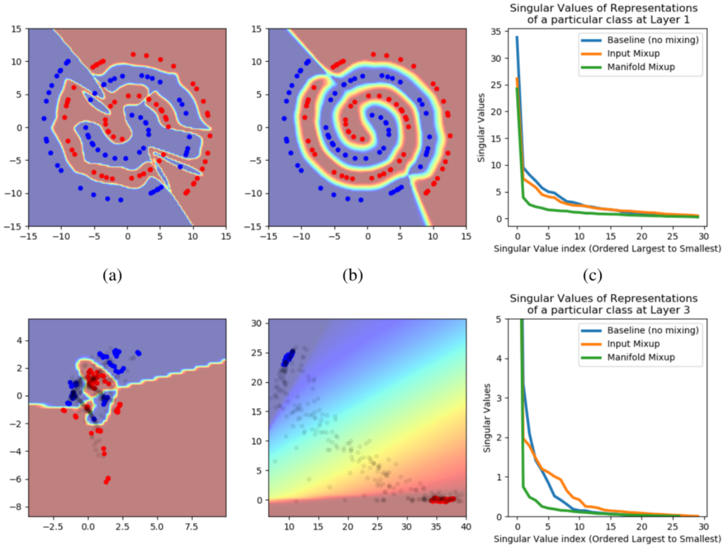
Direct extension of MixUp, play interpolation on hidden representations.

Algorithm:

For each batch, randomly sample a hidden layer k, do interpolation on representation and final labels.

$$(\hat{g}_k, \hat{y}) = (Mix_\lambda(g_k(x), g_k(x')), Mix_\lambda(y, y')) \tag{2}$$

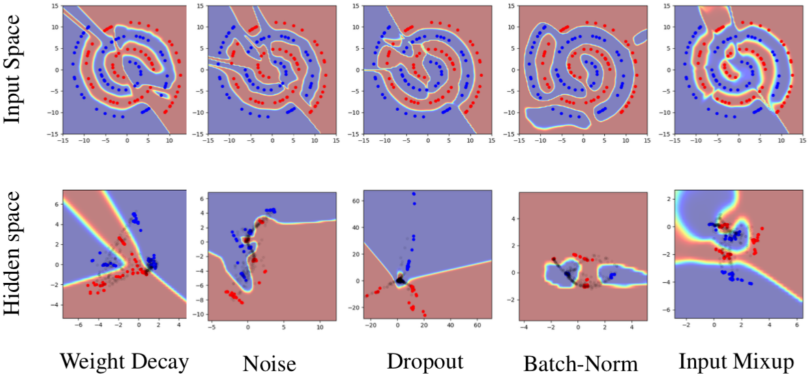
Manfold MixUp



(a) (b) (c)

(d) (e) (f)

Comparing with widely-used regularization:



- smooths decision boundaries that are further away from the training data
- flattens the class-representations
- better generalization and lower test loss
- increase performance at predicting data subject to novel deformations
- robust to adversarial attacks

Theoretical guarantee:
Given some condition, with manifold MixUp, the representations will lie on a low dimension subspace.

Experiments

Table 1: Classification errors on (a) CIFAR-10 and (b) CIFAR-100. We include results from (Zhang et al., 2018)† and (Guo et al., 2016)‡. Standard deviations over five repetitions.

PreActResNet18	Test Error (%)	Test NLL
No Mixup	4.83 ± 0.066	0.190 ± 0.003
AdaMix‡	3.52	NA
Input Mixup†	4.20	NA
Input Mixup ($\alpha = 1$)	3.82 ± 0.048	0.186 ± 0.004
Manifold Mixup ($\alpha = 2$)	$\underline{2.95 \pm 0.046}$	$\underline{0.137 \pm 0.003}$
PreActResNet34		
No Mixup	4.64 ± 0.072	0.200 ± 0.002
Input Mixup ($\alpha = 1$)	2.88 ± 0.043	0.176 ± 0.002
Manifold Mixup ($\alpha = 2$)	$\underline{2.54 \pm 0.047}$	$\underline{0.118 \pm 0.002}$
Wide-Resnet-28-10		
No Mixup	3.99 ± 0.118	0.162 ± 0.004
Input Mixup ($\alpha = 1$)	2.92 ± 0.088	0.173 ± 0.001
Manifold Mixup ($\alpha = 2$)	$\underline{2.55 \pm 0.024}$	$\underline{0.111 \pm 0.001}$

(a) CIFAR-10

PreActResNet18	Test Error (%)	Test NLL
No Mixup	24.01 ± 0.376	1.189 ± 0.002
AdaMix‡	20.97	n/a
Input Mixup†	21.10	n/a
Input Mixup ($\alpha = 1$)	22.11 ± 0.424	1.055 ± 0.006
Manifold Mixup ($\alpha = 2$)	$\underline{20.34 \pm 0.525}$	$\underline{0.912 \pm 0.002}$
PreActResNet34		
No Mixup	23.55 ± 0.399	1.189 ± 0.002
Input Mixup ($\alpha = 1$)	20.53 ± 0.330	1.039 ± 0.045
Manifold Mixup ($\alpha = 2$)	$\underline{18.35 \pm 0.360}$	$\underline{0.877 \pm 0.053}$
Wide-Resnet-28-10		
No Mixup	21.72 ± 0.117	1.023 ± 0.004
Input Mixup ($\alpha = 1$)	18.89 ± 0.111	0.927 ± 0.031
Manifold Mixup ($\alpha = 2$)	$\underline{18.04 \pm 0.171}$	$\underline{0.809 \pm 0.005}$

(b) CIFAR-100

Table 4: Test accuracy on novel deformations. All models trained on normal CIFAR-100.

Deformation	No Mixup	Input Mixup ($\alpha = 1$)	Input Mixup ($\alpha = 2$)	*Manifold Mixup* ($\alpha = 2$)
Rotation U($-20°$,$20°$)	52.96	55.55	56.48	60.08
Rotation U($-40°$,$40°$)	33.82	37.73	36.78	42.13
Shearing U($-28.6°$, $28.6°$)	55.92	58.16	60.01	62.85
Shearing U($-57.3°$, $57.3°$)	35.66	39.34	39.7	44.27
Zoom In (60% rescale)	12.68	13.75	13.12	11.49
Zoom In (80% rescale)	47.95	52.18	50.47	52.70
Zoom Out (120% rescale)	43.18	60.02	61.62	63.59
Zoom Out (140% rescale)	19.34	41.81	42.02	45.29

Table 5: Test accuracy *Manifold Mixup* for different sets of eligible layers $\mathcal{S}$ on CIFAR.

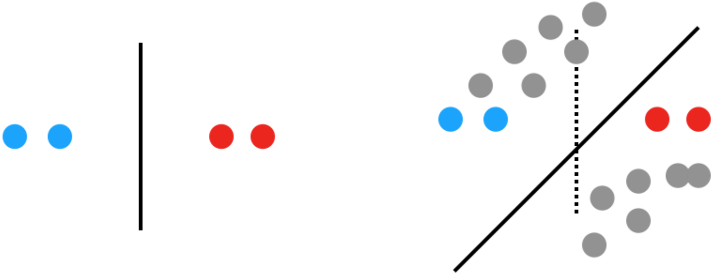
$\mathcal{S}$	CIFAR-10	CIFAR-100
$\{0, 1, 2\}$	97.23	79.60
$\{0, 1\}$	96.94	78.93
$\{0, 1, 2, 3\}$	96.92	80.18
$\{1, 2\}$	96.35	78.69
$\{0\}$	96.73	78.15
$\{1, 2, 3\}$	96.51	79.31
$\{1\}$	96.10	78.72
$\{2, 3\}$	95.32	76.46
$\{2\}$	95.19	76.50
$\{\}$	95.27	76.40

Table 6: Test accuracy (%) of Input Mixup and *Manifold Mixup* for different α on CIFAR-10.

α	Input Mixup	*Manifold Mixup*
0.5	96.68	96.76
1.0	96.75	97.00
1.2	96.72	97.03
1.5	96.84	97.10
1.8	96.80	97.15
2.0	96.73	97.23

Pseudo Label

Why unlabeled data can help?



One of the basic assumption for SSL:
In order to improve generalization performance, the decision boundary should lie in low density regions.

Minimizing the conditional entropy of class probabilities for unlabeled data.

$$H(y|x') = -\frac{1}{N}\sum_{m=1}^{N}\sum_{c=1}^{k} p(y_m^c = 1|x_m')log(y_m^c = 1|x_m') \tag{3}$$

Which means for each unlabeled data, the prediction have to be very confident (close to 1-of-k).

Main idea: Pseudo-Label are target classes for unlabeled data as if they were true labels.

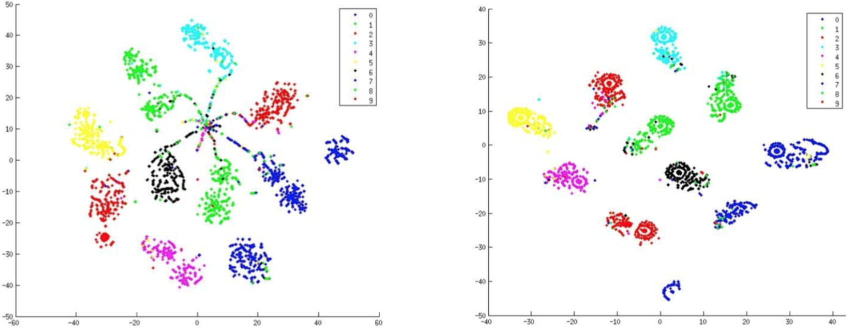
Generate pseudo labels for unlabeled data,

$$y_i' = \begin{cases} 1, & \text{if } i = \arg\max_{i'} f_{i'}(x_i'), \\ 0, & \text{otherwise}. \end{cases} \tag{4}$$

Objective function:

$$L = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i) + \alpha \frac{1}{m}\sum_{i=1}^{m} l(f(x_i'), y_i') \tag{5}$$

The trade off α will be very important, people using annealing process to graduate increase the value.



(a) without unlabeled data (dropNN)

(b) with unlabeled data and Pseudo-Label (+PL)

Meta Pseudo Label

Objective for classification:

$$\min_{\theta} L_{CE}(\theta) = -\mathbb{E}_{x \sim D}[q(y|x) log(p_{\theta}(\hat{y}|x)] \quad (6)$$

- In supervised learning, $q(y|x)$ is one hot code
- In knowledge distillation, $q(y|x) = q_{large}(y|x)$
- In SSL, for labeled data, $q(y|x)$ is one hot code, but for unlabeled data, $q(y|x) = p_{tmp}(y|x)$
    - label smoothing
    - temperature tuning

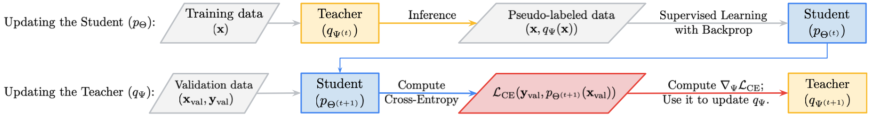
$$p(c|x) = \frac{exp(l_c(x)/\tau)}{\sum_c exp(l_c(x)/\tau)} \quad (7)$$

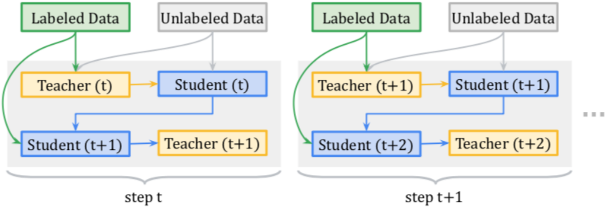
Limitations:

- The target distribution is fixed prior to model updating
- The modulation of target distribution is data agnostic.

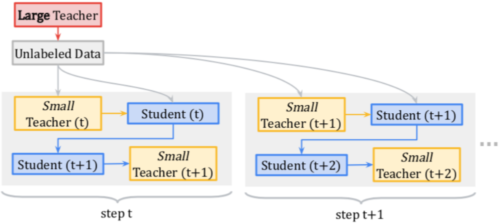
Solution: meta learning, simultaneously updating posterior and model parameters.

Teacher-Student Interaction,





Have to maintain two computational graph in the memory.



Methods	CIFAR-10 (4,000)	SVHN (1,000)
Temporal Ensemble (2017)	83.63 ± 0.63	92.81 ± 0.27
Mean Teacher (2017)	84.13 ± 0.28	94.35 ± 0.47
VAT+EntMin (2018)	86.87 ± 0.39	94.65 ± 0.19
LGA+VAT (2019)	87.94 ± 0.19	93.42 ± 0.36
ICT (2019)	92.71 ± 0.02	96.11 ± 0.04
MixMatch (2019)	93.76 ± 0.06	96.73 ± 0.31
Supervised	82.14 ± 0.25	88.17 ± 0.47
Label Smoothing	82.21 ± 0.18	89.39 ± 0.25
Supervised+MPL	$\mathbf{83.71} \pm \mathbf{0.21}$	$\mathbf{91.89} \pm \mathbf{0.14}$
RandAugment (2019)	85.53 ± 0.25	93.61 ± 0.06
RandAugment+MPL	$\mathbf{87.55} \pm \mathbf{0.14}$	$\mathbf{94.02} \pm \mathbf{0.05}$
UDA (2019a)	94.53 ± 0.18	97.11 ± 0.17
UDA+MPL	$\mathbf{96.11} \pm \mathbf{0.07}$	$\mathbf{98.01} \pm \mathbf{0.07}$

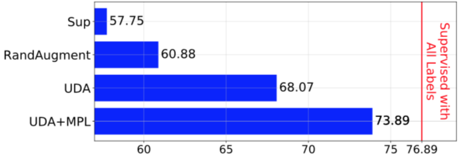


Figure 6: Top-1 accuracy of MPL and other methods on ImageNet-10%. MPL surpasses UDA by almost 6% while being only 3% below to training with all labels.

Methods	CIFAR-10	SVHN	ImageNet
Supervised	97.18 ± 0.08	98.17 ± 0.03	$84.49/97.18$
NoisyStudent	98.22 ± 0.05	$\mathbf{98.71} \pm \mathbf{0.11}$	$85.81/97.53$
ReducedMPL	$\mathbf{98.56} \pm \mathbf{0.07}$	$\mathbf{98.78} \pm \mathbf{0.07}$	$\mathbf{86.87}/\mathbf{98.11}$

Content

Entropy Regularization

Require accuracy on labeled data, and consistency on unlabeled data.

labeled data | unlabeled data

$$\arg\max_c f^c(x_i) = y_i \qquad f(aug_1(x_j)) = f(x_j)$$

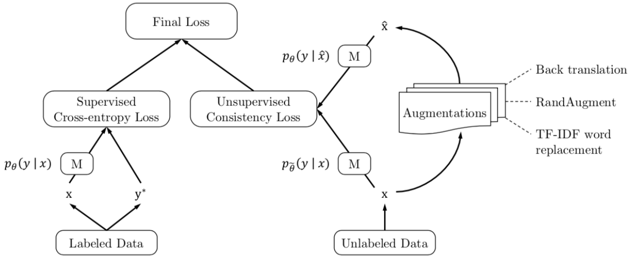


Figure 1: Training objective for UDA, where M is a model that predicts a distribution of y given x.

Loss function

$$\min_\theta E_x(-y_i \log p_\theta(\hat{y}|x_i)) + \lambda \mathbb{E}_x \mathbb{E}_{\hat{x} \sim aug(x)}[KL(p_{\hat{\theta}}(y|x)||p_\theta(y|\hat{x})] \tag{8}$$

where $\hat{\theta}$ is a copy of θ, with no gradient pass through.

- can do augmentation on either input or representations
- highly depends on quality of data augmentation
- careful choice of trade-off.
- state-of-the-art model outperforms fully supervised model.

(c) $\text{BERT}_{\text{LARGE}}$; (d) $\text{BERT}_{\text{FINETUNE}}$: $\text{BERT}_{\text{LARGE}}$ fine-tuned on in-domain unlabeled data[3]. Under each of these four initialization schemes, we compare the performances with and without UDA.

Fully supervised baseline							
Datasets (# Sup examples)		IMDb (25k)	Yelp-2 (560k)	Yelp-5 (650k)	Amazon-2 (3.6m)	Amazon-5 (3m)	DBpedia (560k)
Pre-BERT SOTA		*4.32*	2.16	29.98	3.32	34.81	0.70
$\text{BERT}_{\text{LARGE}}$		4.51	*1.89*	*29.32*	*2.63*	*34.17*	*0.64*
Semi-supervised setting							
Initialization	**UDA**	IMDb (20)	Yelp-2 (20)	Yelp-5 (2.5k)	Amazon-2 (20)	Amazon-5 (2.5k)	DBpedia (140)
Random	✗	43.27	40.25	50.80	45.39	55.70	41.14
	✓	25.23	8.33	41.35	16.16	44.19	7.24
$\text{BERT}_{\text{BASE}}$	✗	18.40	13.60	41.00	26.75	44.09	2.58
	✓	5.45	2.61	33.80	3.96	38.40	1.33
$\text{BERT}_{\text{LARGE}}$	✗	11.72	10.55	38.90	15.54	42.30	1.68
	✓	4.78	2.50	33.54	3.93	37.80	1.09
$\text{BERT}_{\text{FINETUNE}}$	✗	6.50	2.94	32.39	12.17	37.32	-
	✓	**4.20**	**2.05**	**32.08**	**3.50**	**37.12**	-

Table 4: Error rates on text classification datasets. In the fully supervised settings, the pre-BERT SOTAs include ULMFiT (Howard & Ruder, 2018) for Yelp-2 and Yelp-5, DPCNN (Johnson & Zhang, 2017) for Amazon-2 and Amazon-5, Mixed VAT (Sachan et al., 2018) for IMDb and DBPedia. All of our experiments use a sequence length of 512.

MixMatch

MixMatch = Aug + Consistency + Pseudo label + MixUp.

One-hot labeled data: $\{(x_b, p_b)\}_{b=1}^{B}$, unlabeled data $\{u_b\}_{b=1}^{B}$

Phase 1

- for labeled data x_b, generate new samples $(\hat{x}_b, p_b)$
- for unlabeled data u_b, generate new k samples $(\hat{u}_{b,k})$, send them into the model get average prediction $\bar{q}_b = \dfrac{1}{k} \sum_{i=1}^{k} P_{model}(y|\hat{u}_{b,i})$
- $q_b = sharpen(\bar{q}_b)$

Phase 2

- $\hat{\mathbb{X}} = (\hat{x}_b, p_b)_{b=1}^{B}$
- $\hat{\mathbb{U}} = (\hat{u}_{b,k}, q_b)_{b,k=1}^{B,K}$
- $\mathbb{W} = shuffle(cat(\hat{\mathbb{X}}, \hat{\mathbb{U}}))$
- $\mathbb{X}' = MixUp(\hat{\mathbb{X}}_i, \mathbb{W}_i)_{i=1}^{|\hat{\mathbb{X}}|}$
- $\mathbb{U}' = MixUp(\hat{\mathbb{U}}_i, \mathbb{W}_{i+|\hat{\mathbb{X}}|})_{i=1}^{|\hat{\mathbb{U}}|}$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \mathrm{H}(p, \mathrm{p}_{\mathrm{model}}(y \mid x; \theta))$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \mathrm{p}_{\mathrm{model}}(y \mid u; \theta)\|_2^2$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}$$

Graph Regularization

Adding graph regularization into loss function

$$L = L_0 + \lambda L_{reg}, \quad L_{reg} = \sum_{i,j} A_{ij} ||f(x_i) - f(x_j)||^2 = f(X)^T \Delta f(X), \tag{9}$$

where $\Delta = D - A$.
Nowadays, people prefer using GCN, which directly encode graph structure into nn architecture.

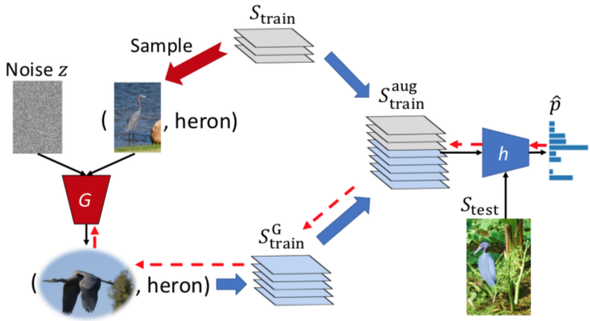
$$H^{l+1} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{1/2} H^l W^l) \tag{10}$$

Content

So far, we introduce some non parametric approach to do data augmentation. We next introduce how to generate new data via a parametric approach.

Another perspective: few shot learning. Variants of MAML require train with limited data, but hallucination based data augmentation will generate more data.

Two objectives: good classifier, "well-generated" new samples.



Inner loop: with augmented training set, find a good classifier. G is fixed.

Outer loop: G and h are trained jointly on dataset
$(x_{tr}, y), (G(z, x_{tr}), y), (x_{te}, y)$