Learning with Biased Labels

Cognitive Computing Lab, Baidu Research

Zheng He

Outline

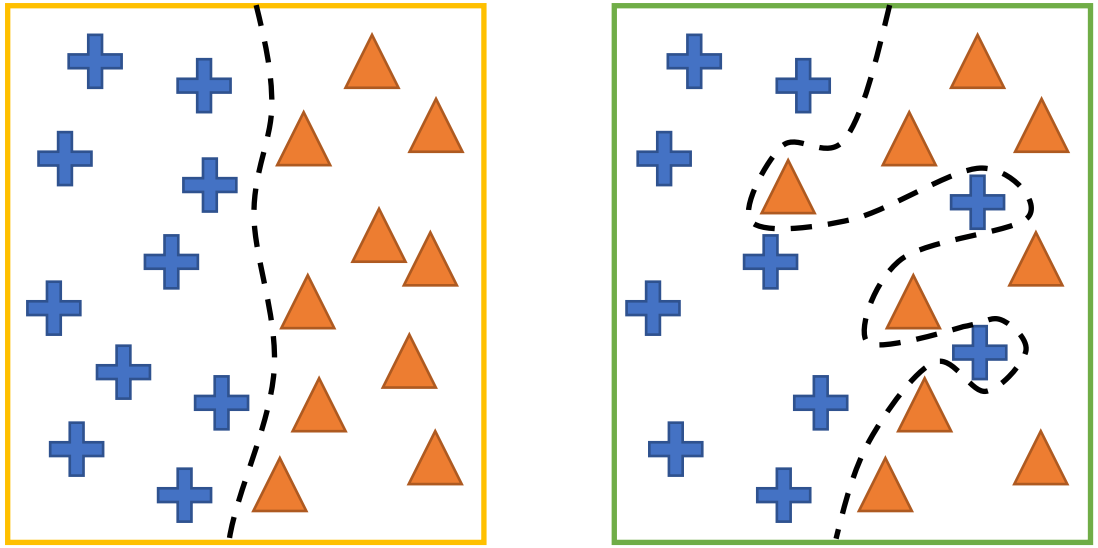
- Label noise robust learning
  - Background
  - Methods
  - New directions
- Quantized label learning
  - Motivation
  - Problem setting
  - Method
  - Experiment

Review of Label Noise Robust Learning

Problem definition

- Clean distribution: $p(x, y)$
- Corrupted distribution: $p(x, \bar{y})$, $\bar{y}$ is a noisy label which may not be true
- Given the training dataset $\bar{D} = \{x_i, \bar{y}_i\}_{i=1}^N$, the goal is to minimize the risk on clean distribution (test data):

$$R(f_\theta) = \mathbb{E}_{(x,y)\sim p(x,y)}[L(f(x), y)]$$



Standard supervised learning

Learning with label noise

Background - Label Noise Robust Learning

Label noise types

- Class-dependent noise: the true label is corrupted by a *noise transition matrix* $T \in [0,1]^{\mathrm{c} \times c}$, where c is the number of classes.

$$T_{i,j} := p(\bar{y} = j | y = i)$$

$$\begin{bmatrix} 1-\tau & \frac{\tau}{n-1} & \cdots & \frac{\tau}{n-1} \\ \frac{\tau}{n-1} & 1-\tau & & \frac{\tau}{n-1} \\ \vdots & & \ddots & \vdots \\ \frac{\tau}{n-1} & \frac{\tau}{n-1} & \cdots & 1-\tau \end{bmatrix}$$

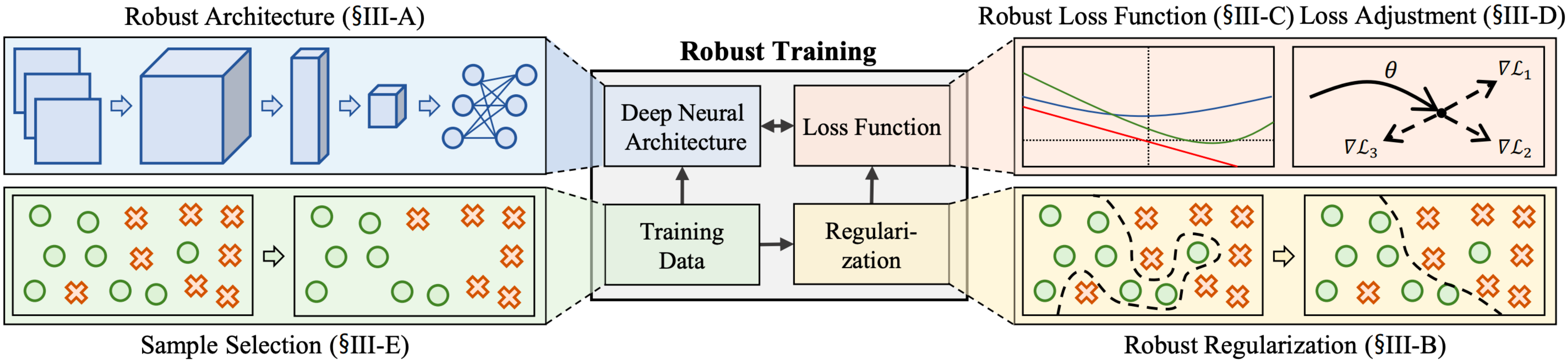
(a) Sym-flipping.

$$\begin{bmatrix} 1-\tau & \tau & 0 & 0 \\ 0 & 1-\tau & \tau & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & & \tau \\ \tau & 0 & \cdots & 1-\tau \end{bmatrix}$$

(b) Pair-flipping.

Methods - Label Noise Robust Learning

➢Deep learning approaches for dealing with label noise



Categorization of recent deep learning methods for overcomming noisy labels.

Song H, Kim M, Park D, et al. Learning from noisy labels with deep neural networks: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022.

Methods - Label Noise Robust Learning

Robust loss function

- We call a loss function L symmetric if it satisfies, for some constant K

$$\sum_{j=1}^{c} L(f(x), j) = K, \forall x \in X, \forall f$$

➢Symmetric loss is proved to be robust under class-dependent label noises.

Robust loss function

Proof. *Given symmetric loss* L, *for any* f, $R_L(f) = \mathbb{E}_{(x,y)\sim p(x,y)}[\mathrm{L}(f(x), y)]$.

For symmetric label noise ($T_{i,j} = \begin{cases} 1-\eta, i=j \\ \frac{\eta}{c-1}, i \neq j \end{cases}$), *for any* f

$$
\begin{aligned}
R_L^{\eta}(f) &= \mathbb{E}_{(x,\bar{y})\sim p(x,\bar{y})}[L(f(x), \bar{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x} \mathbb{E}_{\bar{y}|(x,y)}[L(f(x), \bar{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x} \left[(1-\eta) L(f(x), y) + \frac{\eta}{c-1} \sum_{i \neq y} L(f(x), i)\right] \\
&= \mathbb{E}_x \mathbb{E}_{y|x} \left[(1-\eta) L(f(x), y) + \frac{\eta}{c-1} \big(K - L(f(x), y)\big)\right] \\
&= \frac{K\eta}{c-1} + (1 - \frac{\eta c}{c-1}) R_L(f)
\end{aligned}
$$

Thus, if $\eta < \frac{c-1}{c}$, the minimizer of R_L is also a minimizer of R_L^{η}.

Robust loss function

- Mean Absolute Error (MAE)
  - For DNNs with a softmax output layer, MAE can be computed as:

$$L_{MAE}(f(X), Y) = \sum_{i=1}^{N} \left\| e_{y_i} - f(x_i) \right\|_1 = \sum_{i=1}^{N} 2 - 2f_{y_i}(x_i)$$

  where $e_{y_i} \in \{0,1\}^c$ refers to the one-hot encoding of y_i, and $f_{y_i}(x)$ denotes the $y_{i\text{th}}$ element of $f(x)$

- ! MAE loss is symmetric but difficult to converge

Robust loss function

- Generalized Cross Entropy (GCE)

$$\mathrm{L}_{\mathrm{GCE}}(f(X), Y) = \sum_{i=1}^{N} \frac{1 - f_{y_i}^{q}(x_i)}{q}, q \in (0,1]$$

  - if $q \to 0$, GCE is equivalent to Cross Entropy; if $q \to 1$, GCE is equivalent to MAE.
  - Robustness analysis. When using softmax output layer, $\mathrm{L}_{\mathrm{GCE}}$ is bounded as

$$\frac{c - c^{1-q}}{q} \leq \sum_{j=1}^{c} \frac{1 - f_{y_i}^{q}(x)}{q} \leq \frac{c-1}{q}$$

$$\frac{\eta(1 - c^{1-q})}{q(c - 1 - \eta c)} \leq R_{L_{GCE}}(f^*) - R_{L_{GCE}}^{\eta}(\hat{f}) \leq 0, \qquad if\ \eta < \frac{c-1}{c}$$

Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. NeurIPS, 2018.

Methods - Label Noise Robust Learning

Robust loss function

- Generalized Cross Entropy (GCE)

$$\mathrm{L}_{\mathrm{GCE}}(f(X), Y) = \sum_{i=1}^{N} \frac{1 - f_{y_i}^{q}(x_i)}{q}, q \in (0,1]$$

  - Gradient analysis:
    - GCE behaves like a weighted MAE

$$\frac{\partial L(f(x_i;\theta), y_i)}{\partial \theta} = \begin{cases} \boxed{-}\nabla_\theta f_{y_i}(x_i;\theta), & \text{MAE} \\ \boxed{-\frac{1}{f_{y_i}(x_i;\theta)}}\nabla_\theta f_{y_i}(x_i;\theta), & \text{CE} \\ \boxed{-f_{y_i}(x_i;\theta)^{q-1}}\nabla_\theta f_{y_i}(x_i;\theta), & \text{GCE} \end{cases}$$

Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. NeurIPS, 2018.

Robust loss function

- Symmetric Cross Entropy (SCE)
  - KL divergence: $KL(q||p) = H(q,p) - H(q)$
  - CE = $H(q,p)$, Reverse Cross Entropy RCE = $H(p,q)$

$$L_{RCE}(f(X), Y) = -\sum_{i=1}^{N} f^{\top}(x_i) \log e_i$$

  - Define $\log 0 = A$, RCE term is symmetric. if A=-2, RCE is exact MAE:

$$L_{RCE}(f(X), Y) = -\sum_{i=1}^{N} A\left(1 - f_{y_i}(x_i)\right)$$

  - SCE = αCE +βRCE, ensure both converge and robustness

Yisen Wang, Xingjun Ma, Zaiyi Chen, etal. Symmetric Cross Entropy for Robust Learning with Noisy Labels. ECCV 2019.

Loss adjustment

➢ Correcting loss with the estimated noise transition matrix $\hat{T}$

- Backward correction [1] : is unbiased if $\hat{T} = T$

$$L_{Backward} = \hat{T}^{-1}\langle l(f(x), 1), l(f(x), 2), \dots, l(f(x), c)\rangle^{\top}$$

- Forward correction [1]:

$$L_{Forward} = L\left(\hat{T}^{\top} f(X)^{\top}, \bar{Y}\right)$$

[1] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in Proc. NeurIPS, 2018

Robust regularization

- Reduce model complexity may prevent overfitting
  - Widely used regularization techniques: data augmentation, weight decay, dropout, label smoothing …
  - More advanced techniques:
    - Restrict the distance to initialization of parameters [1]

$$L_{\lambda}^{\text{RDI}}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - \tilde{y}_i\right)^2 + \frac{\lambda^2}{2}\left\|\boldsymbol{\theta} - \boldsymbol{\theta}(0)\right\|^2$$

    - Reduce trainable parameters [2]

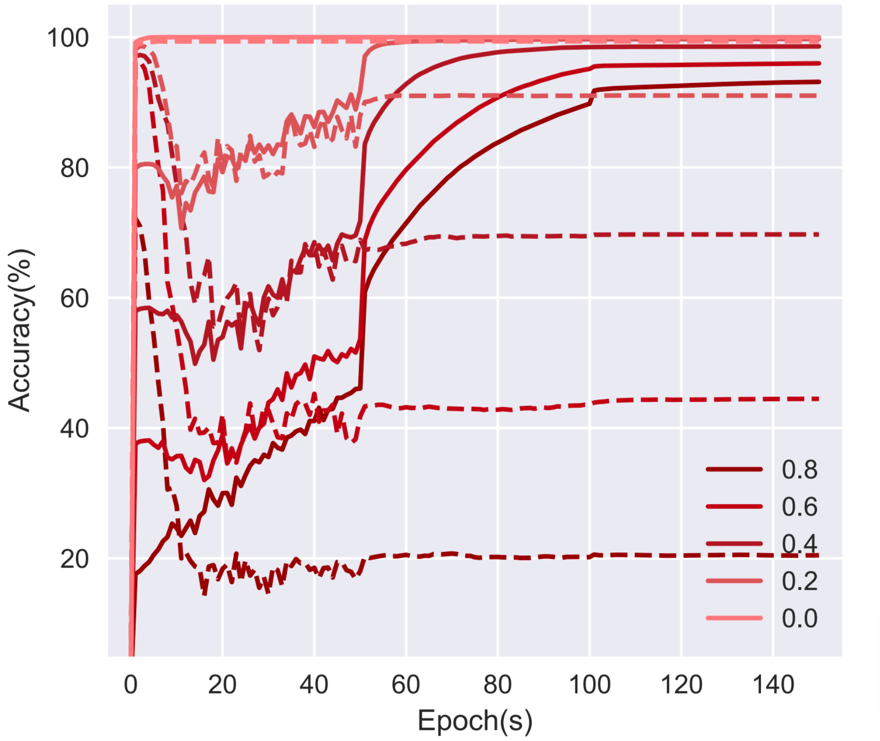
[1] Hu W, Li Z, Yu D. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. ICLR 2020.
[2] Xia X, Liu T, Han B, et al. Robust early-learning: Hindering the memorization of noisy labels. ICLR 2021.

Methods - Label Noise Robust Learning

Sample selection

➢Memorization effect

- Deep networks tend to fit easy (clean) patterns first, and gradually over-fit hard (noisy) patterns
- Memorization effect could be used to identify clean samples

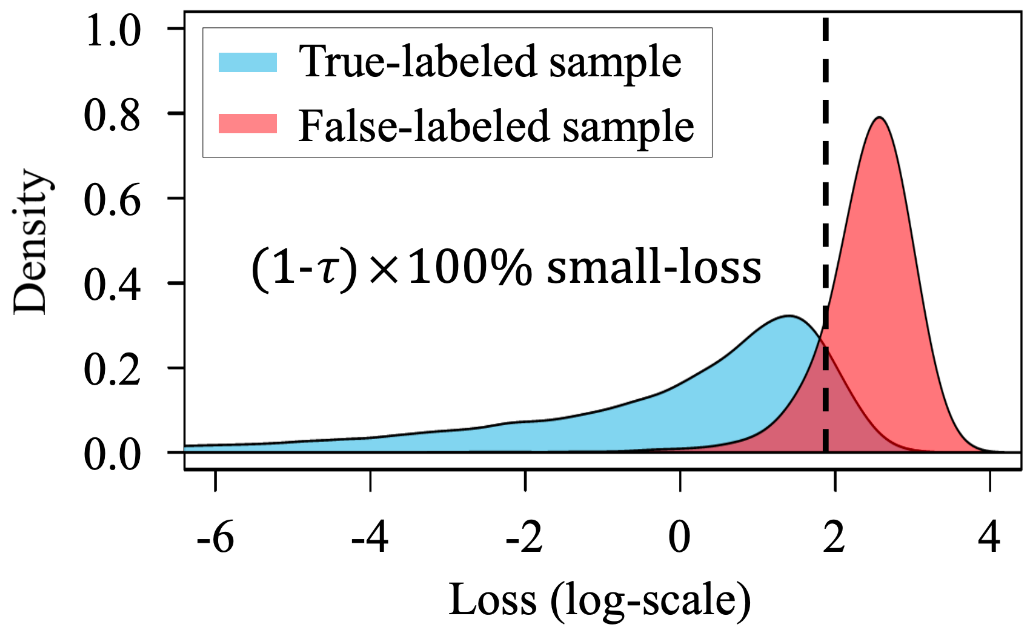


Training and test accuracy curves. Solid lines: training accuracy; dashed lines: test accuracy.

Methods - Label Noise Robust Learning

Sample selection

- ➢Select small-loss training examples as true-labeled examples.
  - Only train with small-loss examples [1];
  - Assigned more weights to small-loss examples [2].

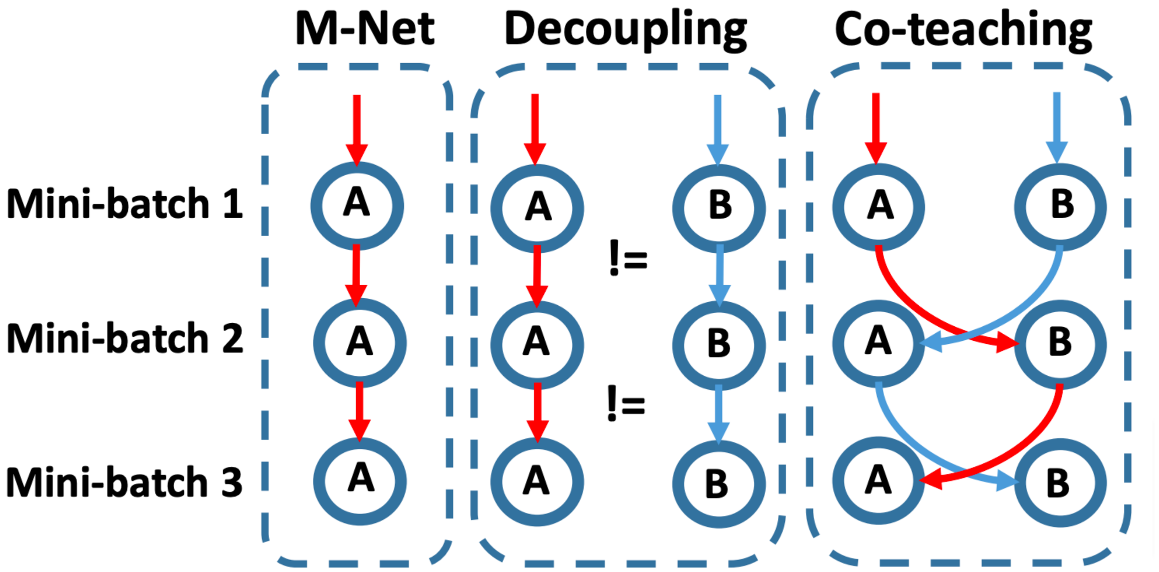


Loss distribution of training examples at the training accuracy of 50% on noisy CIFAR-100.

[1] Co-teaching: Robust training of deep neural networks with extremely noisy labels. NeurIPS 2018.
[2] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in ICML, 2018.

Sample selection

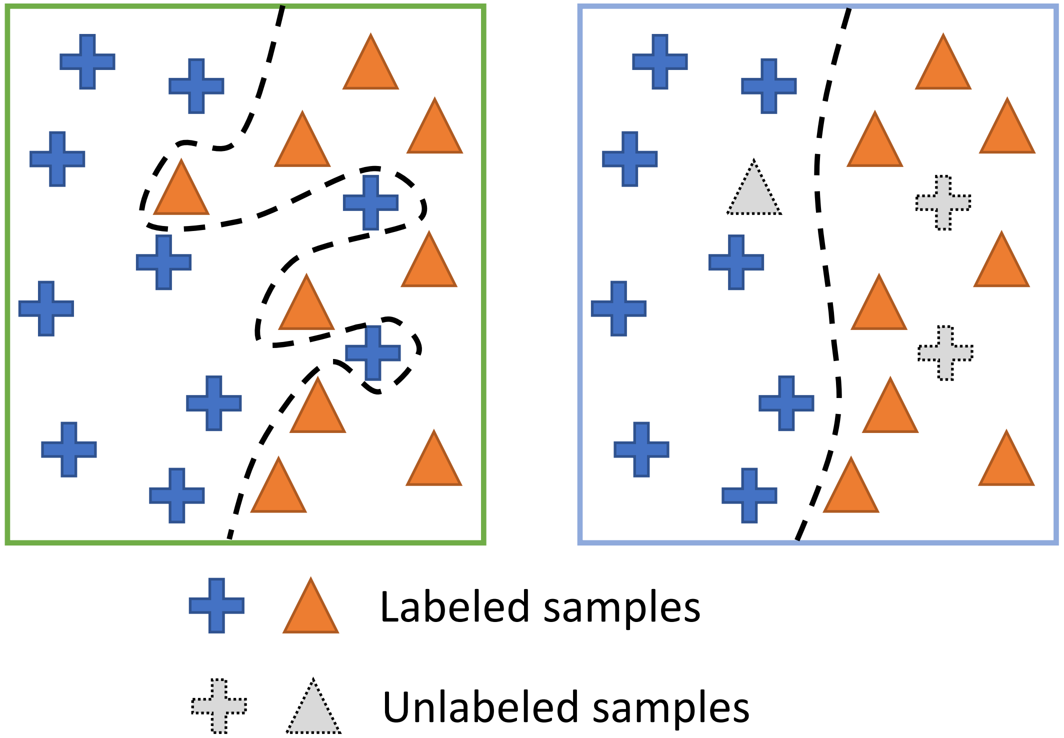
- Co-teaching
  - Train two networks simultaneously
  - In each mini-batch data, each network samples its small-loss instances, and teaches such useful instances to its peer network.
  - Two networks can attenuate each others' error.



Co-teaching: Robust training of deep neural networks with extremely noisy labels. NeurIPS 2018.

Hybrid approaches

- Combine with semi-supervised learning methods.
  - Relabeling [1]
  - Consistency regularization
    - Dividemix [2]
    - ELR [3]



[1] Zhou T, Wang S, Bilmes J. Robust curriculum learning: from clean label detection to noisy label self-correction. ICLR 2020.
[2] J. Li, R. Socher, and S. C. Hoi, “DivideMix: Learning with noisy labels as semi-supervised learning,” in Proc. ICLR, 2020
[3] Liu S, Niles-Weed J, Razavian N, et al. Early-learning regularization prevents memorization of noisy labels. NeurIPS 2020

Methods - Label Noise Robust Learning

Hybrid approaches

- Consistency regularization
  - Assumption: If two points x1, x2 reside in a high-density region are close, then so should be their corresponding outputs y1, y2
  - If a realistic perturbation was applied to the unlabeled data points, the prediction should not change significantly.

$$L_{reg} = \mathbb{E}_{x \in D_u} \left[d\left(f(x;\theta), f(\hat{x};\theta) \right) \right]$$

where $d(,)$ is a distance measure

Hybrid approaches

- Dividemix:
  - Use Gaussian Mixture Model to select clean samples
  - Apply M rounds of data augmentation

for $m = 1$ **to** M **do**
$\hat{x}_{b,m} = \text{Augment}(x_b)$
$\hat{u}_{b,m} = \text{Augment}(u_b)$
end

  - Label refinement: guided by model predictions of labeled samples x_b

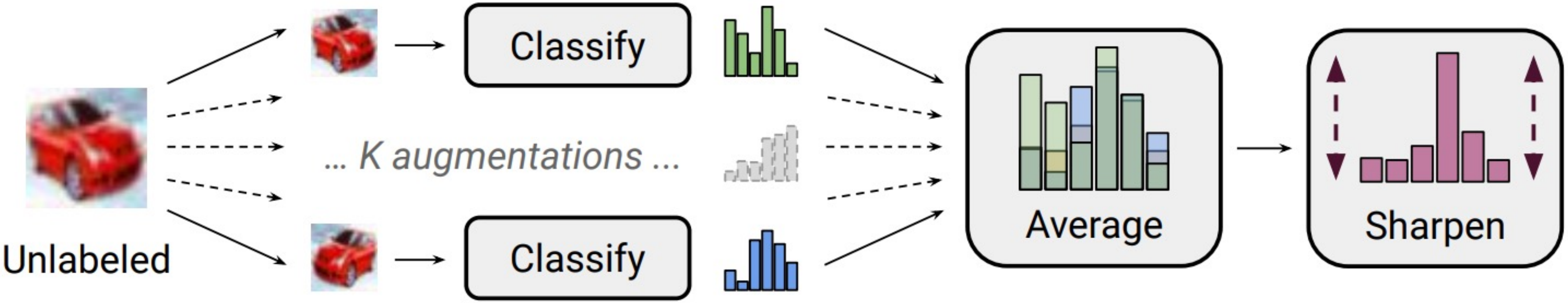
$$p_b = \frac{1}{M}\sum_m \mathrm{p}_{\text{model}}(\hat{x}_{b,m};\theta^{(k)})$$
$$\bar{y}_b = w_b y_b + (1 - w_b) p_b$$
// refine ground-truth label gui
$$\hat{y}_b = \text{Sharpen}(\bar{y}_b, T)$$

J. Li, R. Socher, and S. C. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," in Proc. ICLR, 2020

Methods - Label Noise Robust Learning

Hybrid approaches

- Dividemix:
  - Co-guessing: average the predictions from both networks of unlabeled sample u_b



$$\mathcal{L}_{\mathcal{X}} = -\frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \sum_{c} p_c \log(\mathrm{p}^{\mathrm{c}}_{\mathrm{model}}(x;\theta)), \quad \mathcal{L}_{\mathcal{U}} = \frac{1}{|\mathcal{U}'|} \sum_{x,p \in \mathcal{U}'} \|p - \mathrm{p}_{\mathrm{model}}(x;\theta)\|_2^2 .$$

J. Li, R. Socher, and S. C. Hoi, “DivideMix: Learning with noisy labels as semi-supervised learning,” in Proc. ICLR, 2020

Methods - Label Noise Robust Learning

Hybrid approaches

- ELR:
  - $L_{ELR} = L_{CE} + \lambda L_{reg}$

$$L_{reg}(f(X), Y) = \sum_{i=1}^{N} \log(1 - \langle f(x_i), t_i \rangle)$$

where t_i is moving average of model historical prediction.

  - Use early-learning stage of model prediction to hinder memorization.
  - ELR+ tricks: train two models; use weight averaging; mixup data augmentation

Liu S, Niles-Weed J, Razavian N, et al. Early-learning regularization prevents memorization of noisy labels. NeurIPS 2020.

Summary of methods

- Loss function:
  - Utilize symmetric loss functions
  - Estimate noise transition matrix
- Regularization:
  - Reduce effective model capacity
- Sample selection:
  - Design criterions to identify noisy data.
- Hybrid approaches:
  - Relabeling
  - Consistency regularization

Instance-dependent label noise

- The label corruption probability is assumed to be dependent on both the data features and class labels. The corruption probability is

$$T_{i,j}(x) \coloneqq p(\bar{y} = j | y = i, x)$$

- Difficulties:
    - How to estimate the noise transition matrix? The size of T is very huge.
    - How to identify noisy samples? The loss distribution of true-labeled and false-labeled samples may heavily overlap.

Thanks for listening!