Tsinghua-Berkeley Shenzhen Institute

Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels

32nd Conference on Neural Information Processing Systems (NIPS 2018)

Outline

- Introduction
- Memorization Effect
- Method
- Experiments
- Future Work

Introduction

- A label often corresponds to the true class of the sample, but it may be subjected to a noise process before being presented to the learning algorithm. It is therefore important to ***distinguish the true class of an instance from its observed label***.

- Learning situations where label noise occurs can be called ***imperfectly supervised***, i.e. pattern recognition applications where the assumption of label correctness does not hold for all the elements of the training sample.

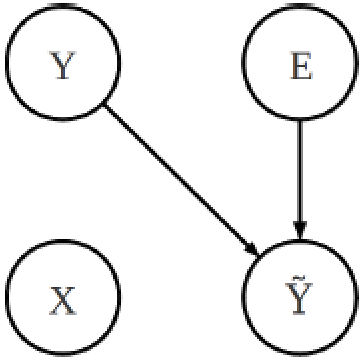
- The ***ubiquity of noise*** is an important issue for practical machine learning, e.g. in medical applications where most medical diagnosis tests are not 100 percent accurate and cannot be considered a gold standard.

Introduction

- **Sources of Label Noise:**

Sources	Examples
Insufficient information	The answers of a patient during anamnesis may be imprecise or incorrect or even may be different if the question is repeated.
Errors occur in labelling process	Using cheap, easy-to-get labels from non-expert using frameworks like the Amazon Mechanical Turk.
When the labelling task is subjective	In electrocardiogram analysis, experts seldom agree on the exact boundaries of signal patterns.
Data encoding or communication problems	In spam filtering, sources of label noise include misunderstanding the feedback mechanisms and accidental click.

Introduction



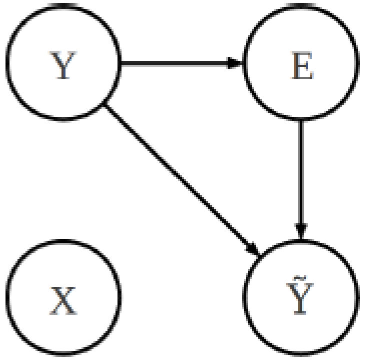
(a)

noisy completely at random (NCAR)

In the NCAR case, the observed label is different from the true class with a probability:

$$p_e = P(E = 1) = P(Y \neq \tilde{Y})$$

Introduction



noisy at random (NAR)

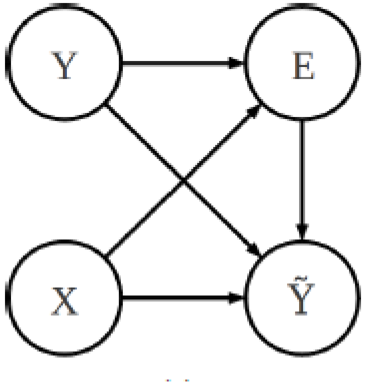
E is still independent of X

$$P(\tilde{Y}=\tilde{y}|Y=y)=\sum_{e\in\{0,1\}} P(\tilde{Y}=\tilde{y}|E=e,Y=y)P(E=e|Y=y).$$

$$\gamma=\begin{pmatrix}\gamma_{11} & \cdots & \gamma_{1n_Y}\\ \vdots & \ddots & \vdots\\ \gamma_{n_Y1} & \cdots & \gamma_{n_Yn_Y}\end{pmatrix}=\begin{pmatrix}P(\tilde{Y}=1|Y=1) & \cdots & P(\tilde{Y}=n_Y|Y=1)\\ \vdots & \ddots & \vdots\\ P(\tilde{Y}=1|Y=n_Y) & \cdots & P(\tilde{Y}=n_Y|Y=n_Y)\end{pmatrix}$$

$$\text{Symmetry flipping:}\quad Q=\begin{bmatrix}1-\epsilon & \frac{\epsilon}{n-1} & \cdots & \frac{\epsilon}{n-1} & \frac{\epsilon}{n-1}\\ \frac{\epsilon}{n-1} & 1-\epsilon & \frac{\epsilon}{n-1} & \cdots & \frac{\epsilon}{n-1}\\ \vdots & & \ddots & & \vdots\\ \frac{\epsilon}{n-1} & \cdots & \frac{\epsilon}{n-1} & 1-\epsilon & \frac{\epsilon}{n-1}\\ \frac{\epsilon}{n-1} & \frac{\epsilon}{n-1} & \cdots & \frac{\epsilon}{n-1} & 1-\epsilon\end{bmatrix}$$

Introduction



noisy not at random (NNAR)

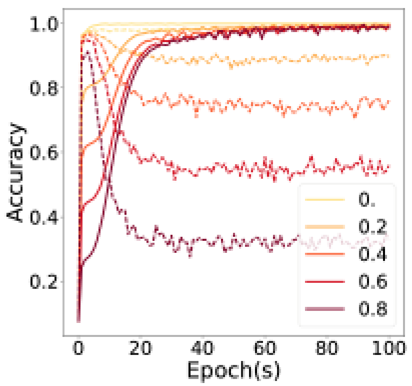
E depends on both variables X and Y , i.e. mislabeling is more probable for certain classes and in certain regions of the X space.

$$p_e = P(E=1) = \sum_{y \in \mathcal{Y}} P(Y=y) \times \int_{x \in \mathcal{X}} P(X=x|Y=y) P(E=1|X=x, Y=y) dx$$

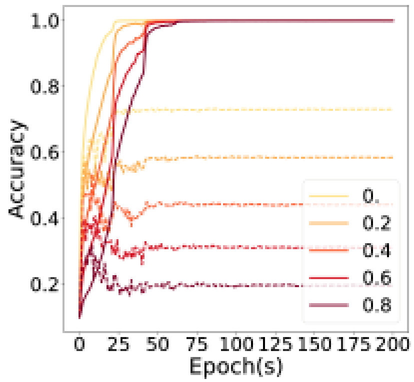
$$\text{Pair flipping:} \quad Q = \begin{bmatrix} 1-\epsilon & \epsilon & 0 & \dots & 0 \\ 0 & 1-\epsilon & \epsilon & & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & & 1-\epsilon & \epsilon \\ \epsilon & 0 & \dots & 0 & 1-\epsilon \end{bmatrix}$$

Memorization Effect

- **Definition of “Memorization”:** the behavior exhibited by DNNs trained on noise.



MNIST

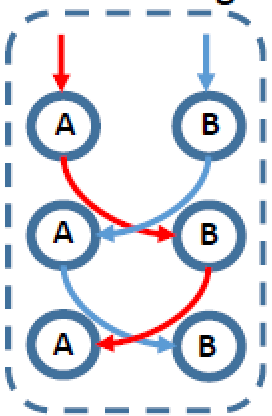


CIFAR-10

◆ The network achieves maximum accuracy on the validation set before achieving high accuracy on the training set. Thus the model first learns the simple and general patterns of the real data before fitting the noise (which results in decreasing validation accuracy).

Method: Co-teaching

Co-teaching



Algorithm 1 Co-teaching Algorithm.

```
Input w_f and w_g, learning rate η, fixed τ, epoch T_k and T_max, iteration N_max;
for T = 1, 2, ..., T_max do
    Shuffle training set D;                                         //noisy dataset
    for N = 1, ..., N_max do
        Fetch mini-batch D̄ from D;
        Obtain D̄_f = argmin_{D':|D'|≥R(T)|D̄|} ℓ(f, D');   //sample R(T)% small-loss instances
        Obtain D̄_g = argmin_{D':|D'|≥R(T)|D̄|} ℓ(g, D');   //sample R(T)% small-loss instances
        Update w_f = w_f − η∇ℓ(f, D̄_g);                              //update w_f by D̄_g;
        Update w_g = w_g − η∇ℓ(g, D̄_f);                              //update w_g by D̄_f;
    end
    Update R(T) = 1 − min{ (T/T_k) τ, τ };
end
Output w_f and w_g.
```

Where: $\bar{\mathcal{D}}_f = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \ell(f,\mathcal{D}')$, $\bar{\mathcal{D}}_g = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \ell(g,\mathcal{D}')$, $w_f = w_f - \eta\nabla\ell(f,\bar{\mathcal{D}}_g)$, $w_g = w_g - \eta\nabla\ell(g,\bar{\mathcal{D}}_f)$, $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$.

Method: Co-teaching

Q1. Why can sampling small-loss instances based on dynamic R(T) help us find clean instances?

- Small-loss instances are more likely to be the ones which are correctly labeled.
- Memorization effect.

Q2. Why do we need two networks and cross-update the parameters?

- Different classifiers can generate different decision boundaries and then have different abilities to learn.
- Peer-review.

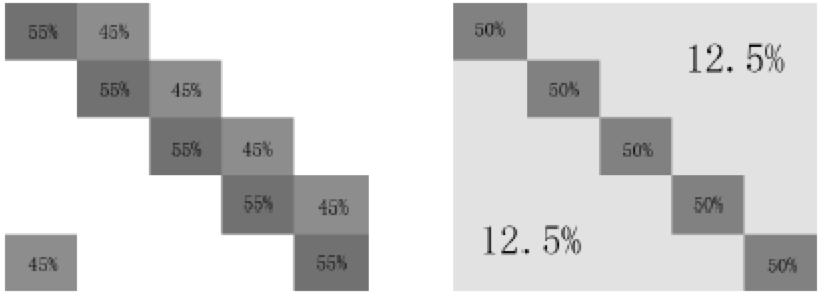
Experiments

- NN architecture:

CNN on *MNIST*	CNN on *CIFAR-10*	CNN on *CIFAR-100*
28×28 Gray Image	32×32 RGB Image	32×32 RGB Image
	3×3 conv, 128 LReLU 3×3 conv, 128 LReLU 3×3 conv, 128 LReLU	
	2×2 max-pool, stride 2 dropout, $p = 0.25$	
	3×3 conv, 256 LReLU 3×3 conv, 256 LReLU 3×3 conv, 256 LReLU	
	2×2 max-pool, stride 2 dropout, $p = 0.25$	
	3×3 conv, 512 LReLU 3×3 conv, 256 LReLU 3×3 conv, 128 LReLU	
	avg-pool	
dense 128→10	dense 128→10	dense 128→100

Experiments

- Noise level and evaluation metric
  - Test Accuracy = (# of correct predictions) / (# of test dataset)
  - Label Precision = (# of clean labels) / (# of all selected labels)



(a) Pair ($\epsilon = 45\%$).
(b) Symmetry ($\epsilon = 50\%$).

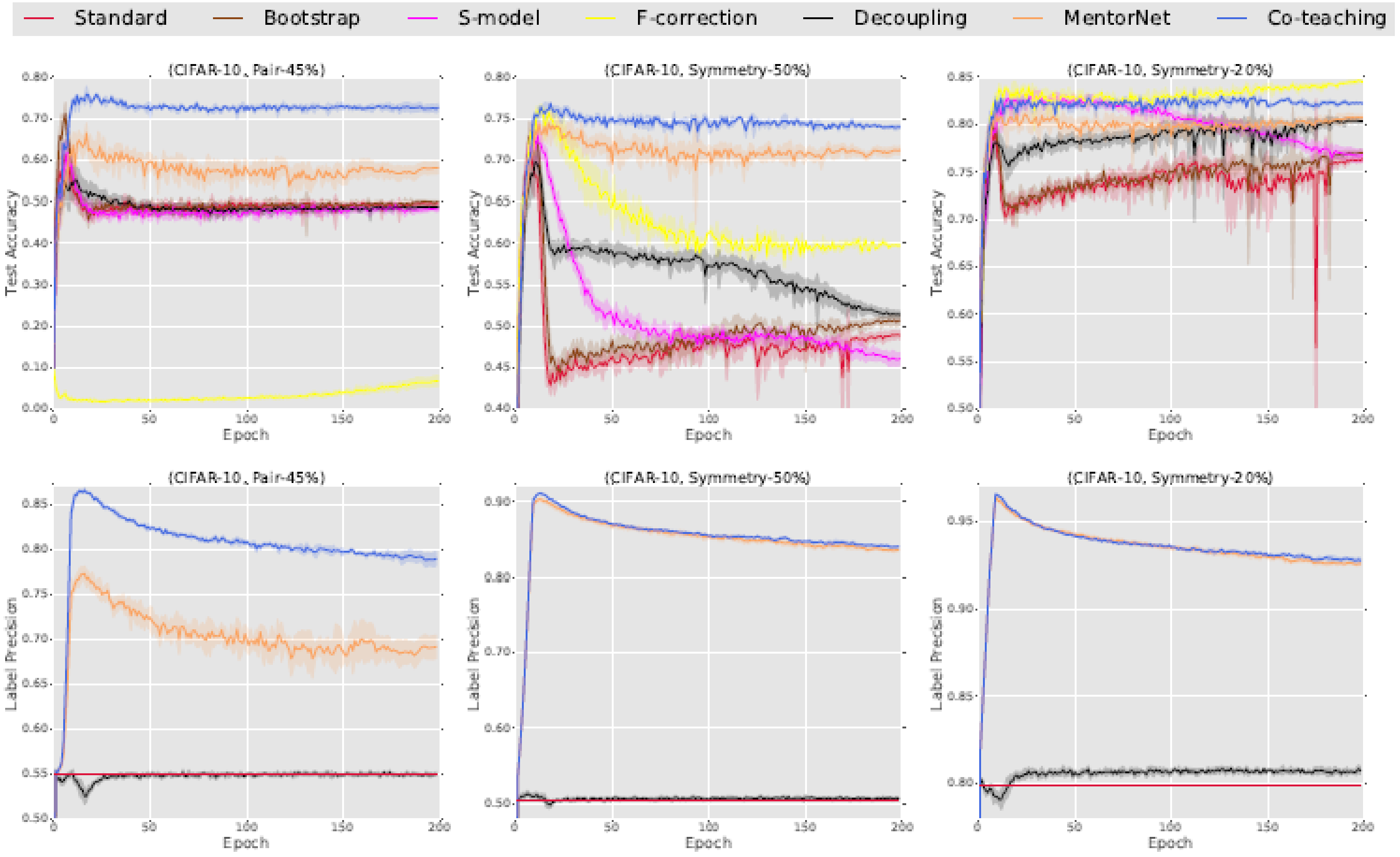
Figure 2: Transition matrices of different noise types (using 5 classes as an example).

Experiments

- Results on CIFAR-10

Table 5: Average test accuracy on *CIFAR-10* over the last ten epochs.

Flipping,Rate	Standard	Bootstrap	S-model	F-correction	Decoupling	MentorNet	Co-teaching
Pair-45%	49.50% ±0.42%	50.05% ±0.30%	48.21% ±0.55%	6.61% ±1.12%	48.80% ±0.04%	58.14% ±0.38%	**72.62%** ±0.15%
Symmetry-50%	48.87% ±0.52%	50.66% ±0.56%	46.15% ±0.76%	59.83% ±0.17%	51.49% ±0.08%	71.10% ±0.48%	**74.02%** ±0.04%
Symmetry-20%	76.25% ±0.28%	77.01% ±0.29%	76.84% ±0.66%	**84.55%** ±0.16%	80.44% ±0.05%	80.76% ±0.36%	82.32% ±0.07%



(a) Pair-45%.
(b) Symmetry-50%.
(c) Symmetry-20%.

Figure 5: Results on *CIFAR-10* dataset. Top: test accuracy vs. number of epochs; bottom: label precision vs. number of epochs.

Experiments

- Choices of hyper-parameters

$$R(T) = 1 - \tau \cdot \min\{T^c/\bar{T}_k, 1\}$$

$$\tau = \{0.5, 0.75, 1, 1.25, 1.5\}\epsilon.$$

Table 7: Average test accuracy on *MNIST* over the last ten epochs.

		$c = 0.5$	$c = 1$	$c = 2$
Pair-45%	$T_k = 5$	75.56%±0.33%	87.59%±0.26%	87.54%±0.23%
	$T_k = 10$	**88.43%±0.25%**	87.56%±0.12%	87.93%±0.21%
	$T_k = 15$	**88.37%±0.09%**	87.29%±0.15%	**88.09%±0.17%**
Symmetry-50%	$T_k = 5$	91.75%±0.13%	91.75%±0.12%	**92.20%±0.14%**
	$T_k = 10$	91.70%±0.21%	91.55%±0.08%	91.27%±0.13%
	$T_k = 15$	91.74%±0.14%	91.20%±0.11%	91.38%±0.08%
Symmetry-20%	$T_k = 5$	97.05%±0.06%	97.10%±0.06%	97.41%±0.08%
	$T_k = 10$	97.33%±0.05%	96.97%±0.07%	**97.48%±0.08%**
	$T_k = 15$	97.41%±0.06%	97.25%±0.09%	**97.51%±0.05%**

Table 8: Average test accuracy of Co-teaching with different τ on *MNIST* over the last ten epochs.

Flipping,Rate	0.5ϵ	0.75ϵ	ϵ	1.25ϵ	1.5ϵ
Pair-45%	66.74%±0.28%	77.86%±0.47%	87.63%±0.21%	**97.89%±0.06%**	69.47%±0.02%
Symmetry-50%	75.89%±0.21%	82.00%±0.28%	91.32%±0.06%	**98.62%±0.05%**	79.43%±0.02%
Symmetry-20%	94.94%±0.09%	96.25%±0.06%	97.25%±0.03%	98.90%±0.03%	**99.39%±0.02%**

Future Work

- Adapt Co-teaching paradigm to train deep models under other weak supervisions.
- Investigate the theoretical guarantees for Co-teaching.
- Analysis for generalization performance on deep learning with noisy labels.

Thanks for Listening!

Input: the labeled training set L

the unlabeled training set U

Process:

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x

Use L to train a classifier h_2 that considers only the x_2 portion of x

Allow h_1 to label p positive and n negative examples from U'

Allow h_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

Randomly choose $2p+2n$ examples from U to replenish U'

图 1　标准协同训练算法 [BlumM98]