P-DIFF: Learning Classifier with Noisy Labels based on Probability Difference Distributions

微软亚洲研究院

Wei Hu, QiHao Zhao, Fan Zhang
Dept. of Computer Science and Technology
Beijing University of Chemical Technology
Beijing, China

Yangyu Huang
Microsoft Research, Asia
Beijing, China
yangyu.huang@microsoft.com

ICPR 2020
25th INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION
Milan, Italy 10 | 15 January 2021

Introduction

We present a very simple but effective training paradigm called P-DIFF, which can train DNN classifiers but obviously alleviate the adverse impact of noisy labels. Our proposed probability difference distribution implicitly reflects the probability of a training sample to be clean, then this probability is employed to re-weight the corresponding sample during the training process.

The Proposed P-DIFF Paradigm

1) Probability Difference:

We define the **probability difference** δ of a sample, which belongs to the y-th class, as

$$\delta = p_y - p_n, \tag{4}$$

2) Global Distribution: Furthermore, only considering samples in one mini-batch [13], [15], [20] reduces the stabilization of sample selection, and a global threshold is not applied too since the loss values are rapidly changed especially in early epoches. P-DIFF adopts a selection method based on a δ histogram. We compute the histogram distribution of δ for all input samples, and this global distribution, called $DIST_{all}$, is just the **probability difference distribution**. We divide the entire range $[-1, 1]$ of the distribution into H bins. We set $H = 200$ in our implementation. Let $PDF(x)$ be the ratio of samples whose δ fall into the x-th bin as

$$PDF(x) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1 & \lceil H \cdot \frac{\delta_i+1}{2}\rceil = x \\ 0 & else\end{cases}, \tag{5}$$

where N is the number of training samples. $PDF(x)$ means the probability distribution function of $DIST_{all}$. We then define the probability cumulative function of $DIST_{all}$ as

$$PCF(x) = \sum_{i=1}^{x} PDF(i). \tag{6}$$

Moreover, given the x-th bin, we can get its value range as

$$\delta \in (2 \cdot \frac{x-1}{H} - 1, 2 \cdot \frac{x}{H} - 1]. \tag{7}$$

Algorithm 1 P-DIFF Paradigm

Input: Training Dataset D, epoch T_k and T_{max}, iteration per-epoch $Iter_{epoch}$, batch size S_{batch}, noise rate τ, batch rate M;
Output: DNN parameter $\vec{W}$;

Initialize $\vec{W}$;
for $T = 1$ **to** T_{max} **do**
  Compute the rate $R(T)$ using Equation 10;
  for $Iter = 1$ **to** $Iter_{epoch}$ **do**
    Compute the threshold $\hat{\delta}$ using Equation 9 and Equation 11;
    Get the mini-batch $\bar{D}$ from D;
    Set the gradient $G = 0$;
    for $S = 1$ **to** S_{batch} **do**
      Get the S-th sample $\bar{D}(S)$;
      Compute $\vec{P}$ of $\bar{D}(S)$ using $\vec{W}$;
      Compute the δ value using Equation 4;
      if $\delta > \hat{\delta}$ **then**
        $\omega = 1$;
      else
        $\omega = 0$;
      $G += \nabla\mathcal{L}$ (see Equation 12);
    Update $DIST_{sub}$ with the computed δ values of the last $M \times Iter_{epoch}$ mini-batches;
    Update the parameter $\vec{W} = \vec{W} - \eta \cdot G$;

$$\hat{\delta} = 2 \cdot \frac{x-1}{H} - 1. \tag{9}$$

$$R(T) = \tau \cdot \min(\frac{T}{T_k}, 1). \tag{10}$$

$$PCF(x) > R(T). \tag{11}$$

$$\mathcal{L} = -\omega \sum_{m=1}^{C} q_m log(p_m), \tag{12}$$

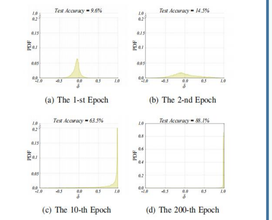

Fig. 1. $DIST_{all}$ and the corresponding performance results at different training epochs.

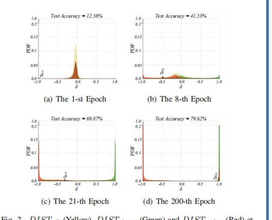

Fig. 2. $DIST_{all}$ (Yellow), $DIST_{clean}$ (Green) and $DIST_{noise}$ (Red) at different training epochs. The DNNs are trained with given noise rates. The corresponding thresholds $\hat{\delta}$ and the performance results can also be seen in the figure.

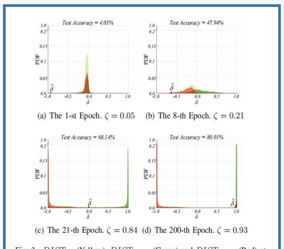

Fig. 3. $DIST_{all}$ (Yellow), $DIST_{clean}$ (Green) and $DIST_{noise}$ (Red) at different training epochs. The DNNs are trained **without** given noise rates. The corresponding thresholds $\hat{\delta}$, ζ, and the performance results are presented.

Experiment Results

TABLE V
AVERAGE TEST ACCURACY ON THREE TESTING DATASETS OVER THE LAST 10 EPOCHS. ACCURACIES OF O2U-NET ARE CITED FROM THE ORIGINAL PAPER [19], SINCE ITS AUTHORS DO NOT PROVIDE THE SOURCE CODES.

DataSet	Noise Type, Rate	Normal	Clean	Co-teaching	Co-teaching++	INCV	O2U-Net	P-DIFF
MNIST	Symmetry, 20%	94.05%	99.68%	97.25%	99.26%	97.62%	-	**99.58%**
	Symmetry, 40%	68.13%	99.51%	92.34%	98.55%	94.23%	-	**99.38%**
	Symmetry, 80%	23.61%	99.04%	81.43%	93.79%	92.66%	-	**97.26%**
	Pair, 10%	95.23%	99.84%	97.76%	99.03%	98.73%	-	**99.54%**
	Pair, 45%	56.52%	99.59%	87.63%	83.57%	88.32%	-	**99.33%**
Cifar-10	Symmetry, 20%	76.25%	89.10%	82.66%	82.84%	84.87%	85.24%	**88.61%**
	Symmetry, 40%	54.37%	87.86%	77.42%	72.32%	74.65%	79.64%	**85.31%**
	Symmetry, 80%	17.28%	80.27%	22.60%	18.45%	24.62%	34.93%	**37.02%**
	Pair, 10%	82.32%	90.87%	85.83%	85.10%	86.27%	**88.22%**	87.78%
	Pair, 45%	49.50%	87.41%	72.62%	50.46%	74.53%	-	**83.25%**
Cifar-100	Symmetry, 20%	47.55%	66.37%	53.79%	52.46%	54.87%	60.53%	**63.72%**
	Symmetry, 40%	33.32%	60.48%	46.47%	44.15%	48.21%	52.47%	**54.92%**
	Symmetry, 80%	7.65%	35.12%	12.23%	9.65%	12.94%	**20.44%**	18.57%
	Pair, 10%	52.94%	69.27%	57.53%	54.71%	58.41%	64.50%	**67.44%**
	Pair, 45%	25.99%	61.29%	34.81%	27.53%	36.79%	-	**45.36%**
Mini-ImageNet	Symmetry, 20%	37.83%	58.25%	41.47%	40.06%	43.12%	45.32%	**56.71%**
	Symmetry, 40%	26.87%	53.88%	34.81%	34.62%	35.65%	38.39%	**47.21%**
	Symmetry, 80%	4.11%	23.63%	6.65%	4.38%	6.71%	8.47%	**11.69%**
	Pair, 10%	43.19%	61.64%	45.38%	43.24%	46.34%	50.32%	**57.85%**
	Pair, 45%	19.74%	57.92%	26.76%	26.76%	28.57%	-	**37.21%**

TABLE VI
COMPARISON ON CLOTH1M

Method	ResNet-101	9-Layer CNN
Coteaching	78.52%	68.74%
Coteaching++	75.78%	69.16%
INCV	80.36%	69.89%
O2U-Net	82.38%	75.61%
P-Diff	**83.67%**	**77.38%**

TABLE VII
TRAINING TIME OF DIFFERENT APPROACHES. THE TIME OF O2U-NET IS NOT PROVIDED BECAUSE OF ITS CLOSED-SOURCE.

Approach	In Theory	Real Cost/Epoch
Normal	$1\times$	64 s
Co-teaching	$\approx 2\times$	131 s
Co-teaching++	$\approx 2\times$	143 s
INCV	$> 3\times$	217 s
O2U-Net	$> 3\times$	-
P-DIFF	$\approx 1\times$	71 s