南京航空航天大學

Nanjing University of Aeronautics and Astronautics

NLNL: Negative Learning for Noisy Labels

Youngdong Kim Junho Yim Juseung Yun Junmo Kim

School of Electrical Engineering, KAIST, South Korea

{ydkim1293, junho.yim, st24hour, junmo.kim}@kaist.ac.kr

ICCV 2019

Noisy Labels

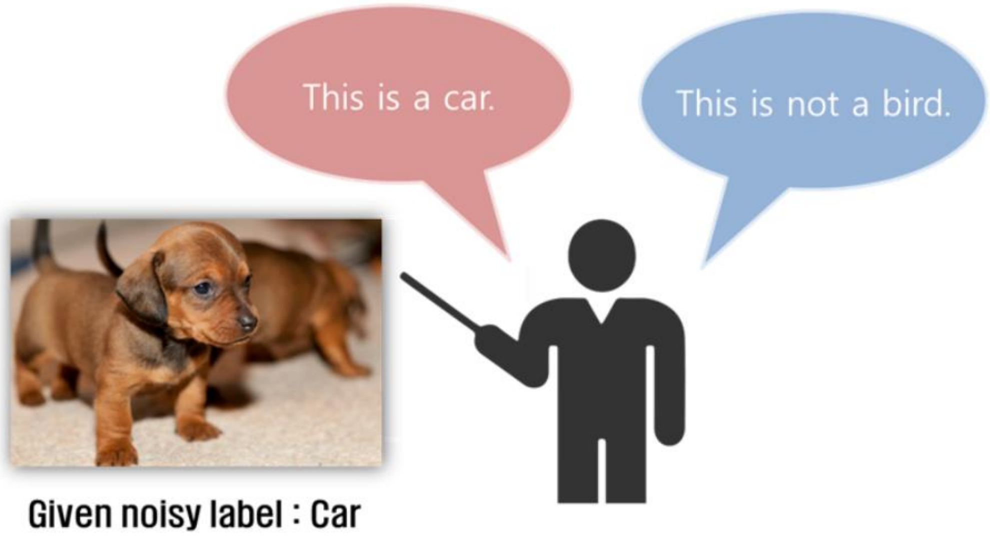
1.Robust Loss/Architecture: ***GCE, SCE, Curriculum Loss, S-Model***

2.Sample Selection: ***Decouple, Mentornet, Co-teaching, SELF, SELFIE, Co-teaching+***

3.Meta Learning: ***Meta-Weight-Net, MLNT, Learning to reweight***

4.Semi-supervised Learning: ***DivideMix, SELF***

Introduction



- Positive Learning(PL): Input image belongs to this label
- Negative Learning(NL): Input image does not belong to this label (complementary label)

Intuition

南京航空航天大学 Nanjing University of Aeronautics and Astronautics

- If inaccurate labels, or noisy labels, exist, training with PL will provide **wrong information**, thus severely degrading performance.

- Training CNN to acknowledge that this image is not a bird is in some way an act of providing CNN the **right information** because a dog is clearly not a bird.

Negative Learning

南京航空航天大学 Nanjing University of Aeronautics and Astronautics

Algorithm 1 Complementary label generation

Input: Training label $y \in \mathcal{Y}$

while iteration **do**

$\overline{y}$ = Randomly select from $\{1, \ldots, C\} \setminus \{y\}$

Output: Complementary label $\overline{y}$

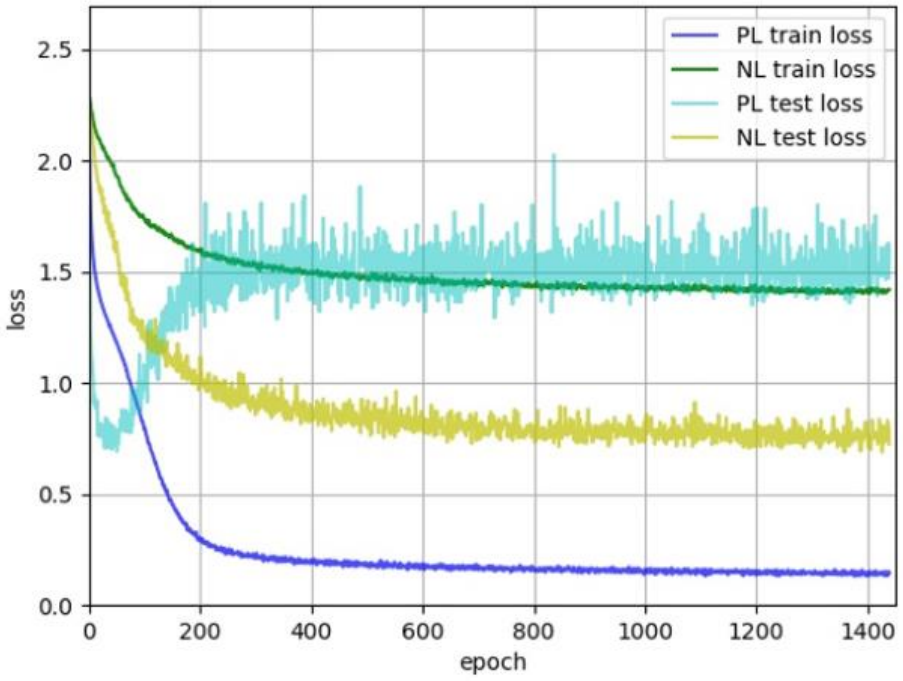
Training with PL:

$$\mathcal{L}(f, y) = -\sum_{k=1}^{c} \boldsymbol{y}_k \log \boldsymbol{p}_k \tag{1}$$

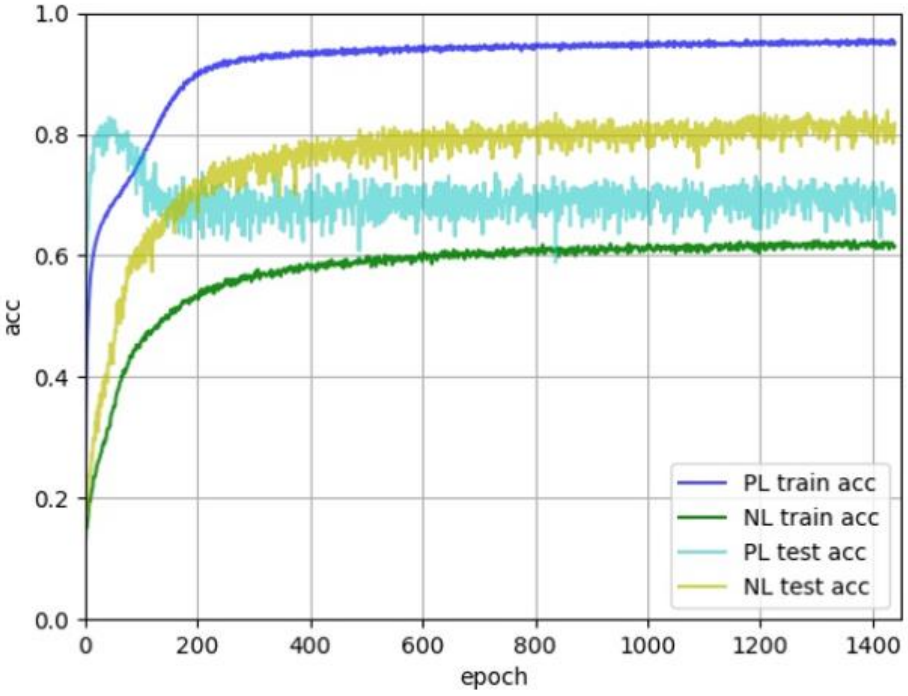
Training with NL:

$$\mathcal{L}(f, \overline{y}) = -\sum_{k=1}^{c} \overline{\boldsymbol{y}}_k \log(1 - \boldsymbol{p}_k) \tag{2}$$

Negative Learning



(a)



(b)

Figure 2: Comparison between PL and NL. (a): Loss graph of PL and NL. (b): Accuracy graph of PL and NL.

Algorithm (SelNLPL)

Algorithm 2 Overall process of SelNLPL

Input: Training data $(\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})$, network $f(\boldsymbol{x}; \boldsymbol{\theta})$, total epoch T

for $i \leftarrow 1$ to T **do** ▷ NL
    Batch $\leftarrow$ Sample $\boldsymbol{x}$
    Update f by minimizing Eq. 2
for $i \leftarrow 1$ to T **do** ▷ SelNL
    Batch $\leftarrow$ Sample $\boldsymbol{x}$ if $\boldsymbol{p}_y > 1/\mathrm{c}$
    Update f by minimizing Eq. 2
for $i \leftarrow 1$ to T **do** ▷ SelPL
    Batch $\leftarrow$ Sample $\boldsymbol{x}$ if $\boldsymbol{p}_y > \gamma$
    Update f by minimizing Eq. 1

Output: Network $f(\boldsymbol{x}; \boldsymbol{\theta})$

Method

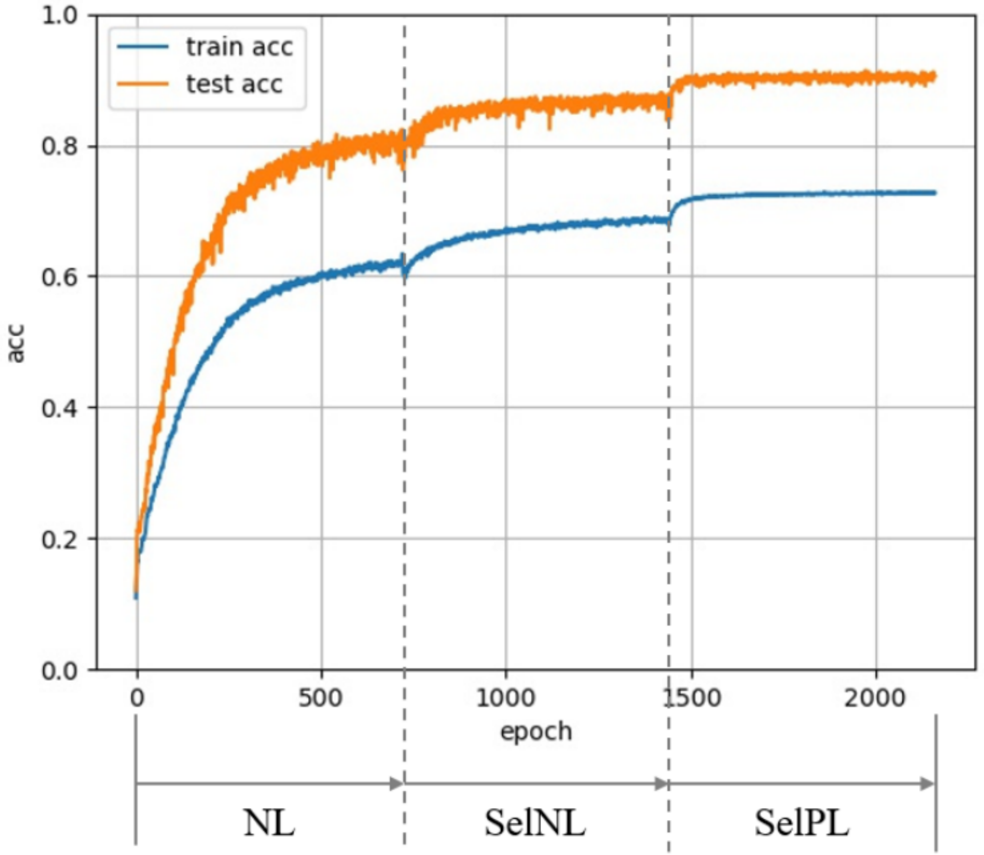


Figure 4: Accuracy graph of SelNLPL. Training is performed sequentially with NL, SelNL, and SelPL.

Negative Learning

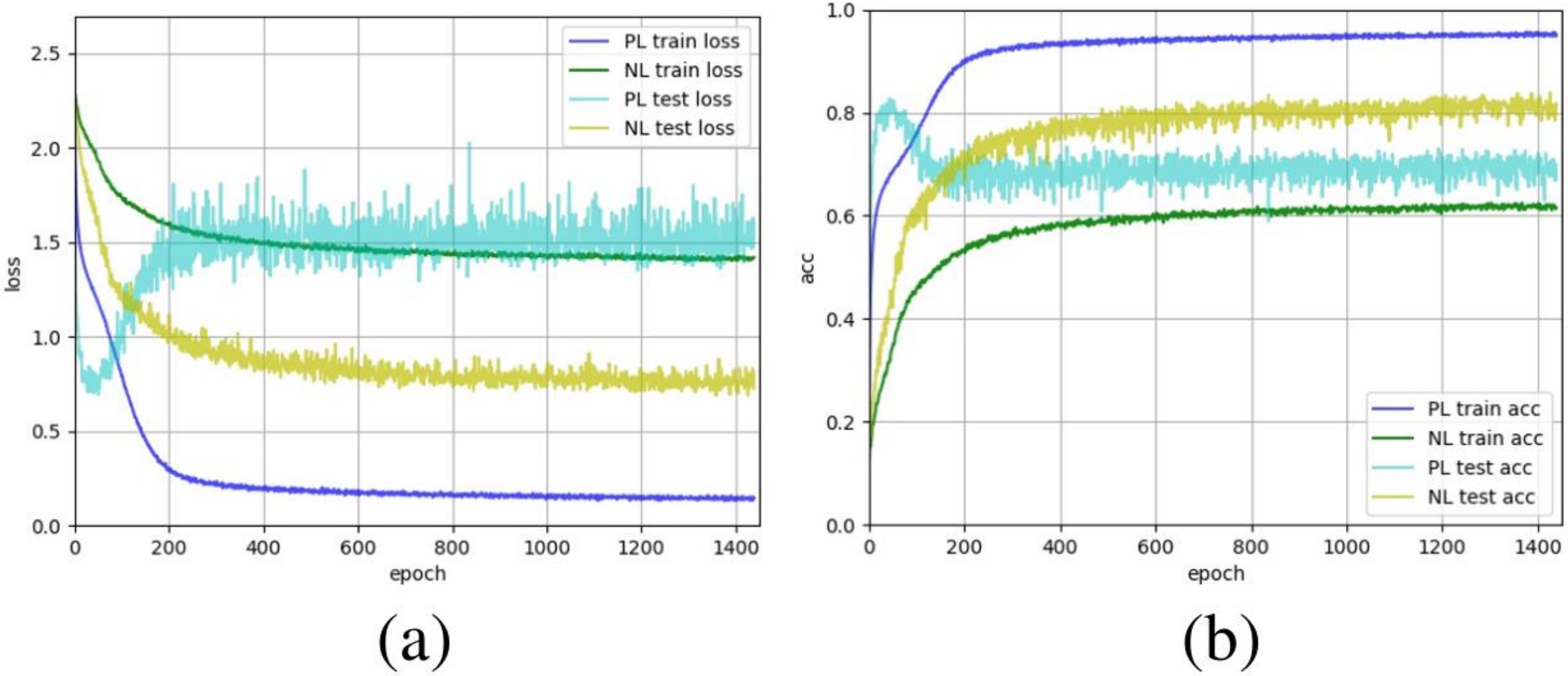
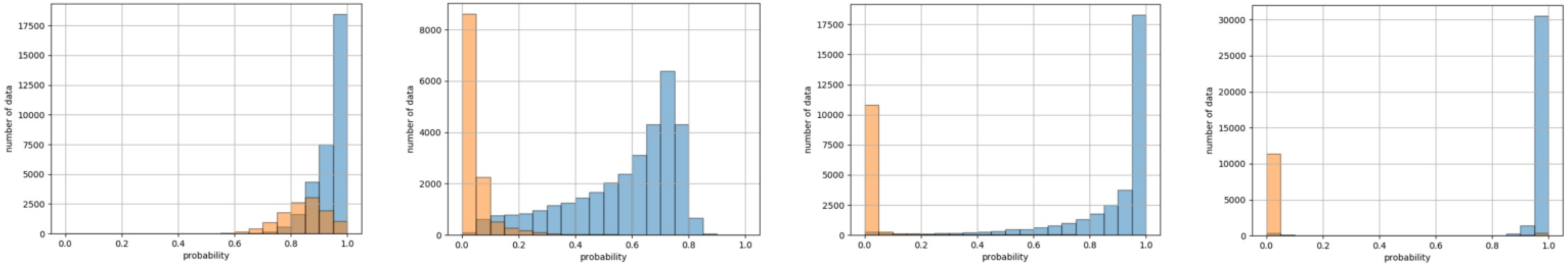


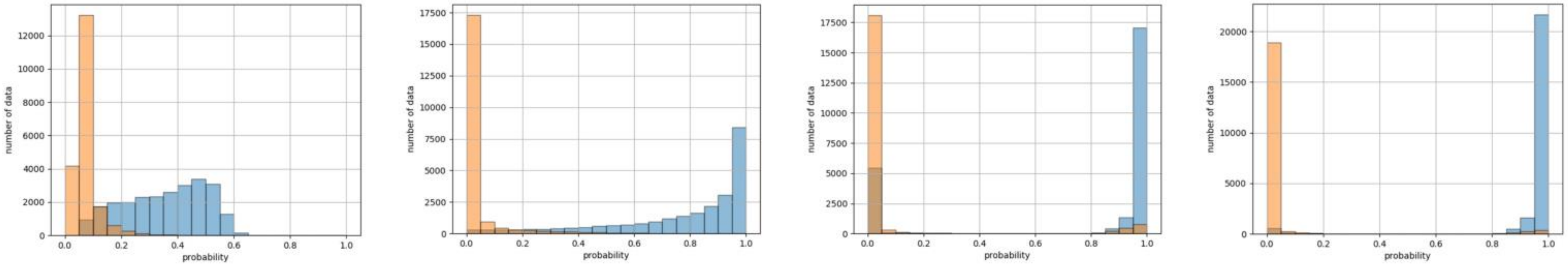
Figure 2: Comparison between PL and NL. (a): Loss graph of PL and NL. (b): Accuracy graph of PL and NL.

Experiments



(a) (b) (c) (d)

Figure 3: Histogram showing the distribution of CIFAR10 training data with 30% *symm-inc* noise, according to probability $\boldsymbol{p}_y$ (confidence). Blue indicates clean data, whereas orange indicates noisy data. (a): PL. (b): NL. (c): NL→SelNL. (d): NL→SelNL→SelPL (SelNLPL).



(a) (b) (c) (d)

Figure 7: Histogram showing the distribution of CIFAR10 training data with 50% *symm-inc* noise, according to probability (confidence). (a): NL. (b): NL→SelNL. (c): NL→SelPL. (d): NL→SelNL→SelPL (SelNLPL).

Experiments

Datasets	Model	Methods	*Symm* 20	40	60	80	*Asymm* 10	20	30	40
FashionMNIST	ResNet18	CE	93.24	92.09	90.29	86.20	94.06	93.72	92.72	89.82
		MAE [3]	80.39	79.30	82.41	74.73	74.03	63.03	58.14	56.04
		Forward T [25]	93.64	92.69	91.16	87.59	94.33	94.03	93.91	93.65
		Forward $\hat{T}$ [25]	93.26	92.24	90.54	85.57	94.09	93.66	93.52	88.53
		L_q [39]	93.35	92.58	91.30	88.01	93.51	93.24	92.21	89.53
		Truncated L_q [39]	93.21	92.60	91.56	88.33	93.53	93.36	92.76	91.62
		Ours	**94.82**	**94.16**	**92.78**	-	**95.10**	**94.88**	**94.66**	**93.96**
CIFAR10	ResNet34	CE	86.98	81.88	74.14	53.82	90.69	88.59	86.14	80.11
		MAE [3]	83.72	67.00	64.21	38.63	82.61	52.93	50.36	45.52
		Forward T [25]	88.63	85.07	79.12	64.30	91.32	90.35	89.25	88.12
		Forward $\hat{T}$ [25]	87.99	83.25	74.96	54.64	90.52	89.09	86.79	83.55
		L_q [39]	89.83	87.13	82.54	64.07	90.91	89.33	85.45	76.74
		Truncated L_q [39]	89.70	87.62	82.70	67.92	90.43	89.45	87.10	82.28
		Ours	**94.23**	**92.43**	**88.32**	-	**94.57**	**93.35**	**91.80**	**89.86**
CIFAR100	ResNet34	CE	58.72	48.20	37.41	18.10	66.54	59.20	51.40	42.74
		MAE [3]	15.80	9.03	7.74	3.76	13.38	11.50	8.91	8.20
		Forward T [25]	63.16	54.65	44.62	24.83	**71.05**	**71.08**	**70.76**	**70.82**
		Forward $\hat{T}$ [25]	39.19	31.05	19.12	8.99	45.96	42.46	38.13	34.44
		L_q [39]	66.81	61.77	53.16	29.16	68.36	66.59	61.45	47.22
		Truncated L_q [39]	67.61	62.64	54.04	29.60	68.86	66.59	61.87	47.66
		Ours	**71.52**	**66.39**	**56.51**	-	70.35	63.12	54.87	45.70

Table 3: Comparison with results reported by Zhang *et al.* [39]

Experiments

		30% *symm-inc* noise				50% *symm-inc* noise			
		Accuracy	Estimated noise (%)	Recall	Precision	Accuracy	Estimated noise (%)	Recall	Precision
#1	NL-SelNL-SelPL	93.82	27.60	96.28	94.01	91.17	45.15	95.80	95.38
#2	NL-SelNL	92.44 (-1.38)	33.76	98.47	78.62	89.36 (-1.81)	52.75	98.56	83.99
#3	NL-SelPL	93.41 (-0.41)	28.17	97.04	92.84	72.91 (-18.26)	54.35	92.12	76.19
#4	NL	87.32 (-6.5)	52.24	99.80	51.49	72.53 (-18.64)	89.21	99.99	50.38

Table 7: Analysis of measuring significance of each step of SelNLPL. #1: SelNLPL. #2: Deleting SelPL from #1. #3: Deleting SelNL from #1. #4: Deleting SelNL and SelPL from #1.

Conclusion

- Utilizing the proposed NL, we introduce a new framework, called **SelNLPL**, for filtering out noisy data from training data.

- Our method does not require any prior knowledge of the type or number of noisy data points. It does not require any tuning of hyper-parameters that depend on prior knowledge, making our method applicable in real life.

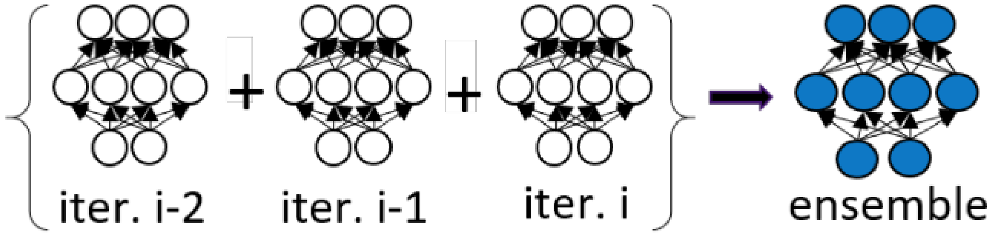
南京航空航天大學
Nanjing University of Aeronautics and Astronautics

SELF: Learning To Filter Noisy Labels With Self-Ensembling

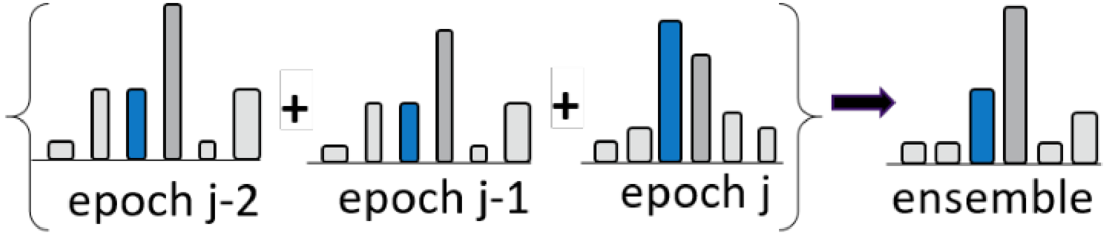
Duc Tam Nguyen [*†], Chaithanya Kumar Mummadi [*‡], Thi Phuong Nhung Ngo [‡]
Thi Hoai Phuong Nguyen [§], Laura Beggel [‡], Thomas Brox [*]

ICLR 2020

SELF-ENSEMBLE LEARNING



(a) Model ensemble (Mean teacher)



(b) Predictions ensemble

Exponential running average: $$\bar{z}_j = \alpha \bar{z}_{j-1} + (1-\alpha)\hat{z}_j$$

Algorithm

Algorithm 1 *SELF*: Self-Ensemble Label Filtering pseudocode

Require: $\mathcal{D}_{train}$ = noisy labeled training set
Require: $\mathcal{D}_{val}$ = noisy labeled validation set
Require: (x, y) = training stimuli and label
Require: α = ensembling momentum, $0 \leq \alpha \leq 1$

```
i ← 0                                      ▷ counter to track iterations
M_i ← train(D_train, D_val)                ▷ initial Mean-Teacher ensemble model training
M_best ← M_i                               ▷ set initial model as best model
z̄_i ← 0                                    ▷ initialize ensemble predictions of all samples
                                             (ignored sample index for simplicity)
while acc(M_i, D_val) ≥ acc(M_best, D_val) do   ▷ iterate until no best model is found on D_val
    M_best ← M_i                           ▷ save the best model
    D_filter ← D_train                     ▷ set filtered dataset as initial label set
    i ← i + 1
    for (x, y) in D_filter do
        ẑ_i ← M_best(x)                    ▷ evaluate model output ẑ_i
        z̄_i ← α z̄_{i-1} + (1 − α) ẑ_i       ▷ accumulate ensemble predictions z̄_i
        if y ≠ argmax(z̄_i) then            ▷ verify agreement of ensemble predictions & label
            y ← ∅ in D_filter              ▷ identify it as noisy label & remove from label set
        end if
    end for
    M_i ← train(D_filter, D_val)           ▷ train Mean-Teacher model on filtered label set
end while
return M_best
```

Experiments

NOISE RATIO	CIFAR-10 40%	CIFAR-10 60%	CIFAR-10 80%	CIFAR-100 40%	CIFAR-100 60%	CIFAR-100 80 %
USING NOISY VALIDATION SET						
REED-HARD (REED ET AL., 2014)	69.66	-	-	51.34	-	-
S-MODEL (GOLDBERGER & BEN-REUVEN, 2016)	70.64	-	-	49.10	-	-
OPEN-SET WANG ET AL. (2018)	78.15	-	-	-	-	-
RAND. WEIGHTS (REN ET AL., 2018)	86.06	-	-	58.01	-	-
BI-LEVEL-MODEL (JENNI & FAVARO, 2018)	89.00	-	20.00	61.00	-	13.00
MENTORNET (JIANG ET AL., 2017)	89.00	-	49.00	68.00	-	35.00
L_q (ZHANG & SABUNCU, 2018)	87.13	82.54	64.07	61.77	53.16	29.16
TRUNC L_q(ZHANG & SABUNCU, 2018)	87.62	82.70	67.92	62.64	54.04	29.60
FORWARD $\hat{T}$ (PATRINI ET AL., 2017)	83.25	74.96	54.64	31.05	19.12	08.90
CO-TEACHING (HAN ET AL., 2018B)	81.85	74.04	29.22	55.95	47.98	23.22
SELF (OURS)	**93.70**	**93.15**	**69.91**	**71.98**	**66.21**	**42.09**
USING CLEAN VALIDATION SET (1000 IMAGES)						
DAC (THULASIDASAN ET AL., 2019)[2]	90.93	87.58	70.80	68.20	59.44	34.06
MENTORNET (JIANG ET AL., 2017)	78.00	-	-	59.00	-	-
RAND. WEIGHTS (REN ET AL., 2018)	86.55	-	-	58.34	-	-
REN ET AL (REN ET AL., 2018)	86.92	-	-	61.31	-	-
SELF (OURS)	**95.10**	**93.77**	**79.93**	**74.76**	**68.35**	**46.43**

THANKS