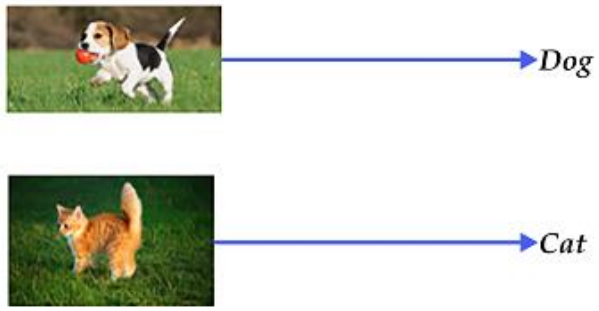
Deep Self-Learning From Noisy Labels

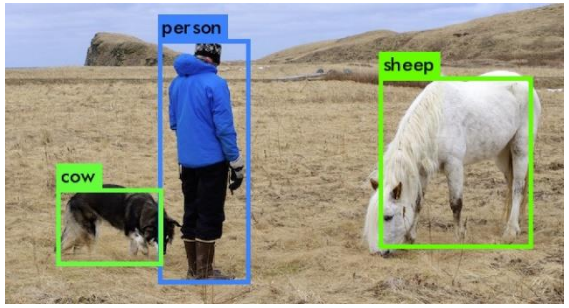
Jiangfan Han[1] Ping Luo[2] Xiaogang Wang[1]
[1]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[2]The University of Hong Kong
{jiangfanhan@link., xgwang@ee.}cuhk.edu.hk pluo@cs.hku.hk

ICCV 2019

Introduction

DNNs achieve impressive results on many computer vision tasks……





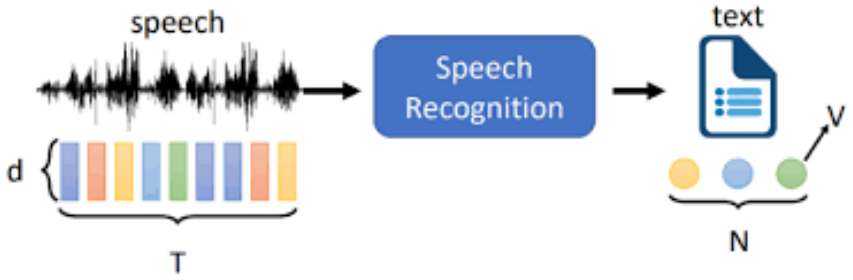


Image classification Object detection Speech recognition

But learning from noisy labels significantly degrades performances

Motivation

How to deal with learning with noise?

- Transition matrix
- Noisy tolerance loss functions
- Using additional supervision

Drawbacks

- Simple assumptions
- Infeasible for real-world noisy datasets
- Required extra clean samples

Method

Notations

- Training Set $\mathcal{D} = \{\mathbf{X}, Y\} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$
- neural network $\mathcal{F}(\theta, \mathbf{x})$
- Loss function $\mathcal{L}(Y, \mathcal{F}(\theta, \mathbf{X}))$

an optimization problem is defined as this where L represents the empirical risk.

$$\theta^* = \text{argmin}_\theta \, \mathcal{L}(Y, \mathcal{F}(\theta, \mathbf{X}))$$

when Y contains noises, the solution of the above equation would be sub-optimal.

$$\theta^* = \text{argmin}_\theta \, \mathcal{L}(Y, \hat{Y}(\mathbf{X}, \mathbf{X}_s), \mathcal{F}(\theta, \mathbf{X}))$$

Xs indicates a set of class prototypes to represent the distribution of classes.

Method

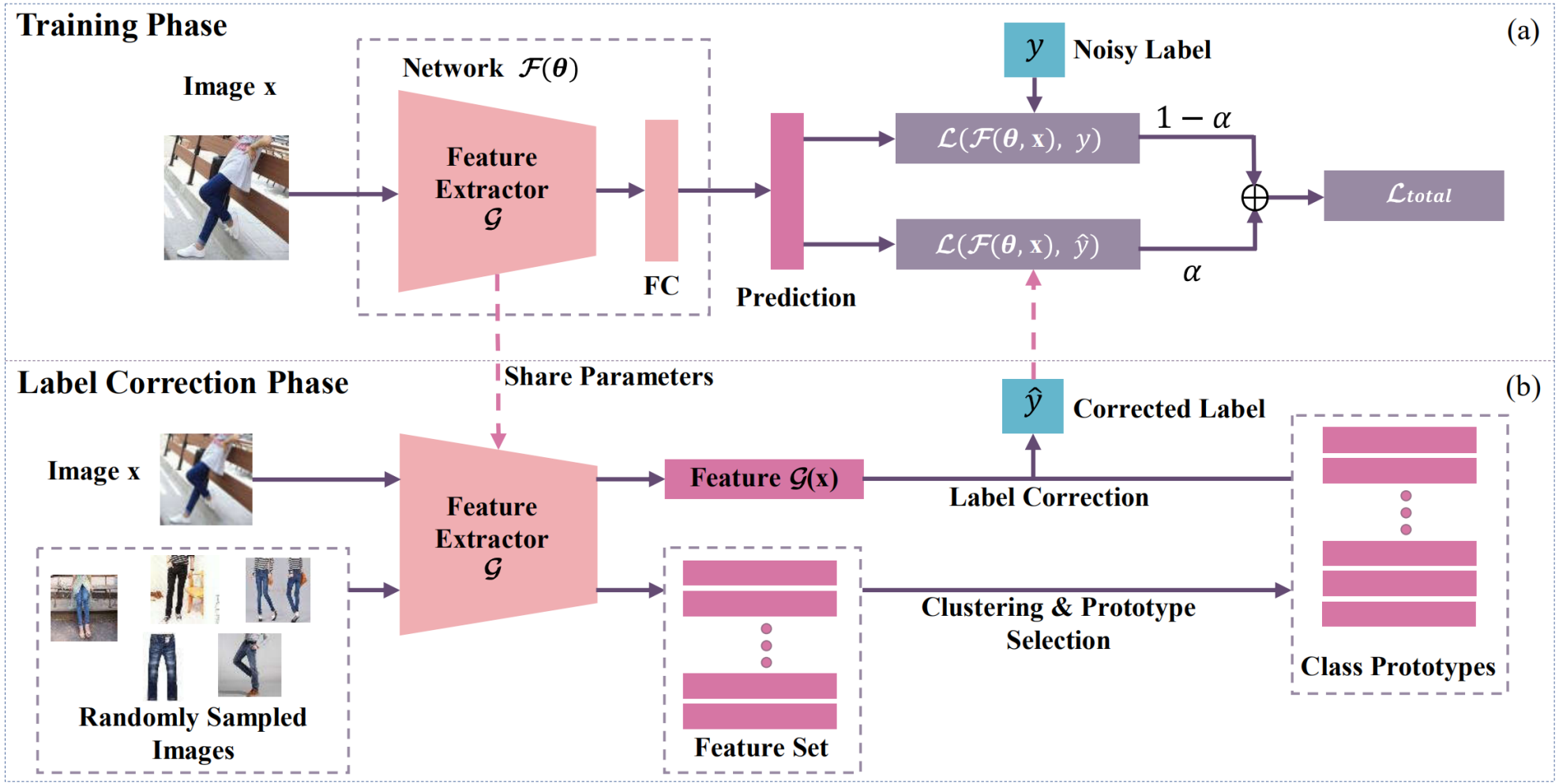


Figure 2. Illustration of the pipeline of iterative self-learning framework on the noisy dataset. (a) shows the training phase and (b) shows the label correction phase, where these two phases proceed iteratively. The deep network $\mathcal{G}$ can be shared, such that only a single model needs to be evaluated in testing.

Training Phase

The objective function is the empirical risk of cross-entropy loss

$$\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), y) = -\frac{1}{n}\sum_{i=1}^{n}\log(\mathcal{F}(\theta, \mathbf{x}_i)_{y_i})$$

The corrected label is produced by a self-training scheme in the label correction phase.

$$\mathcal{L}_{total} = (1-\alpha)\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), y) + \alpha\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), \hat{y})$$

Set α to 0 at the very beginning of training, After a preliminary network was trained, then step into the second phase and obtain the corrected label.

Label Correction Phase

Select several class prototypes for each category

1. Use the preliminary network to extract deep features of images in the training set

$$\mathcal{F}(\theta, \mathbf{x}) = f(\mathcal{G}(\mathbf{x}))$$

2. Randomly sample m images and their deep features with label c

3. Then, we calculate the cosine similarity between the deep features and construct a similarity matrix

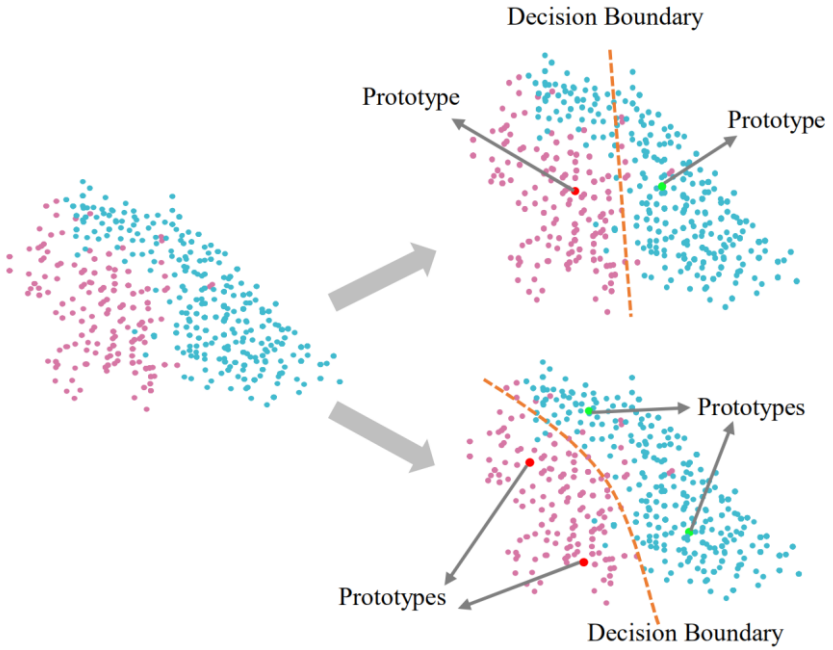
$$S_{ij} = \frac{\mathcal{G}(\mathbf{x}_i)^T \mathcal{G}(\mathbf{x}_j)}{||\mathcal{G}(\mathbf{x}_i)||_2 ||\mathcal{G}(\mathbf{x}_j)||_2}$$

Label Correction Phase

To select prototypes, we define a density for each image

$$\rho_i = \sum_{j=1}^{m} \text{sign}(S_{ij} - S_c)$$

Images with correct labels should be close to each other, while the images with noisy labels are usually isolated from others.



if the chosen prototypes are very close to each other, the representative ability of these prototypes is equivalent to using a single prototype.

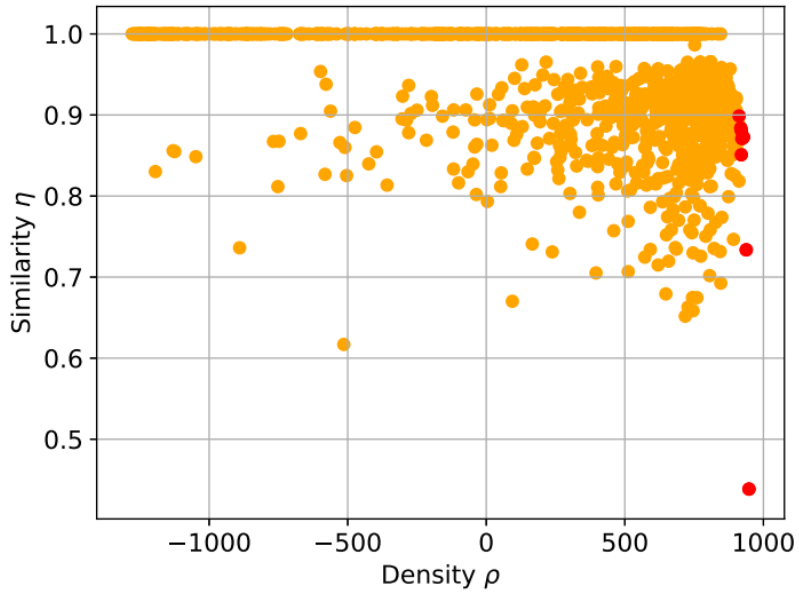
Label Correction Phase

defifine a similarity measurement ηi

$$\eta_i = \begin{cases} \max_{j,\ \rho_j > \rho_i} S_{ij}, & \rho_i < \rho_{max} \\ \min_j S_{ij}, & \rho_i = \rho_{max} \end{cases}$$

where $\rho_{max} = \max\{\rho_1, \ldots, \rho_m\}$

- high-density value ρ
  Probability of a clean label
- low similarity value η
  Far away from other clean labels



(b)

Label Correction Phase

Given an image x, the similarity score for the c-th class is calculated as

$$\sigma_c = \frac{1}{p}\sum_{l=1}^{p} \cos(\mathcal{G}(\mathbf{x}), \mathcal{G}(\mathbf{x}_{cl})), c = 1...K$$

$$\hat{y} = \text{argmax}_c \, \sigma_c, \;\; c = 1...K$$

After getting the corrected label, we treat it as complementary supervision signal to train the neural network in the training phase.

Iterative Self-Learning

The training phase and the label correction phase proceed iteratively.

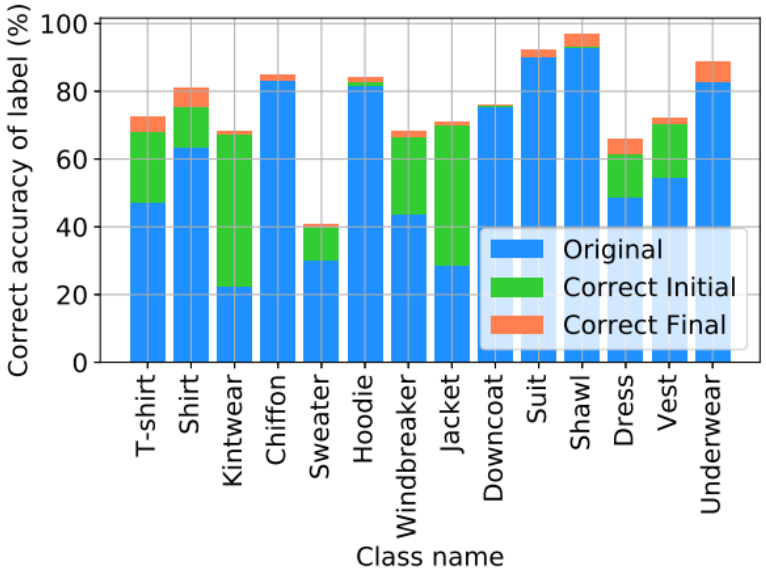
Algorithm 1 Iterative Learning

Initialize network parameter θ
for $M = 1 : \text{num\_epochs}$ **do**
  if $M < \text{start\_epoch}$ **then**
    sample $(\mathbf{X}, Y)$ from training set.
    $\theta^{(t+1)} \leftarrow \theta^{(t)} - \xi \nabla \mathcal{L}(\mathcal{F}(\theta^{(t)}, \mathbf{X}), Y)$
  else
    Sample $\{\mathbf{x}_{c1}, \ldots, \mathbf{x}_{cm}\}$ for each class label c.
    Extract the feature and calculate the similarity $\mathbf{S}$.
    Calculate the density ρ and elect the class prototypes $\mathcal{G}(\mathbf{X}_c)$ for each class c.
    Get the corrected $\hat{y}$ for each sample x_i
    sample $(\mathbf{X}, Y, \hat{Y})$ from training set.
    $\theta^{(t+1)} \leftarrow \theta^{(t)} - \xi \nabla((1-\alpha)\mathcal{L}(\mathcal{F}(\theta^{(t)}, \mathbf{X}), Y) + \alpha \mathcal{L}(\mathcal{F}(\theta^{(t)}, \mathbf{X}), \hat{Y})$
  end if
end for

Experiments

#	Method	Data	Accuracy
1	Cross Entropy	1M noisy	69.54
2	Forward [25]	1M noisy	69.84
3	Joint Optim. [35]	1M noisy	72.23
4	MLNT-Teacher [16]	1M noisy	73.47
5	Ours	1M noisy	**74.45**
6	Forward [25]	1M noisy + 25k verify	73.11
7	CleanNet w_{hard} [15]	1M noisy + 25k verify	74.15
8	CleanNet w_{soft} [15]	1M noisy + 25k verify	74.69
9	Ours	1M noisy + 25k verify	**76.44**
10	Cross Entropy	1M noisy+ 50k clean	80.27
11	Forward [25]	1M noisy + 50k clean	80.38
12	CleanNet w_{soft} [15]	1M noisy + 50k clean	79.90
13	Ours	1M noisy + 50k clean	**81.16**

Table 1. The classification accuracy (%) on Clothing1M compare with other methods.



#	Method	Accuracy
1	Cross Entropy	84.51
2	CleanNet w_{hard} [15]	83.47
3	CleanNet w_{soft} [15]	83.95
4	Ours	**85.11**

Table 5. The classification accuracy (%) on Food-101N compare with other methods.

Experiments

Number of Samples

m	320	640	1280	2560
1M noisy (Final)	74.37	74.07	**74.45**	74.27
1M noisy (Initial)	72.04	72.03	**72.09**	72.05
1M noisy + 25k verify (Final)	76.43	76.49	76.44	**76.55**
1M noisy + 25k verify (Initial)	74.09	73.97	74.17	**74.21**

Table 3. The classification accuracy (%) on Clothing1M with different number of samples used to select prototypes for each class. Final denotes the accuracy get by the model at the end of training. Initial denotes the correct accuracy by the model just step into the first label correction phase.

Prototype Selection

Method	Data	Accuracy
K-means++ [1]	1M noisy	74.08
Density peak Euc. [31]	1M noisy	74.11
Ours	1M noisy	**74.45**
K-means++ [1]	1M noisy + 25k verify	76.22
Density peak Euc. [31]	1M noisy + 25k verify	76.05
Ours	1M noisy + 25k verify	**76.44**

Table 4. The classification accuracy (%) on Clothing1M with different cluster methods used to select the prototypes.

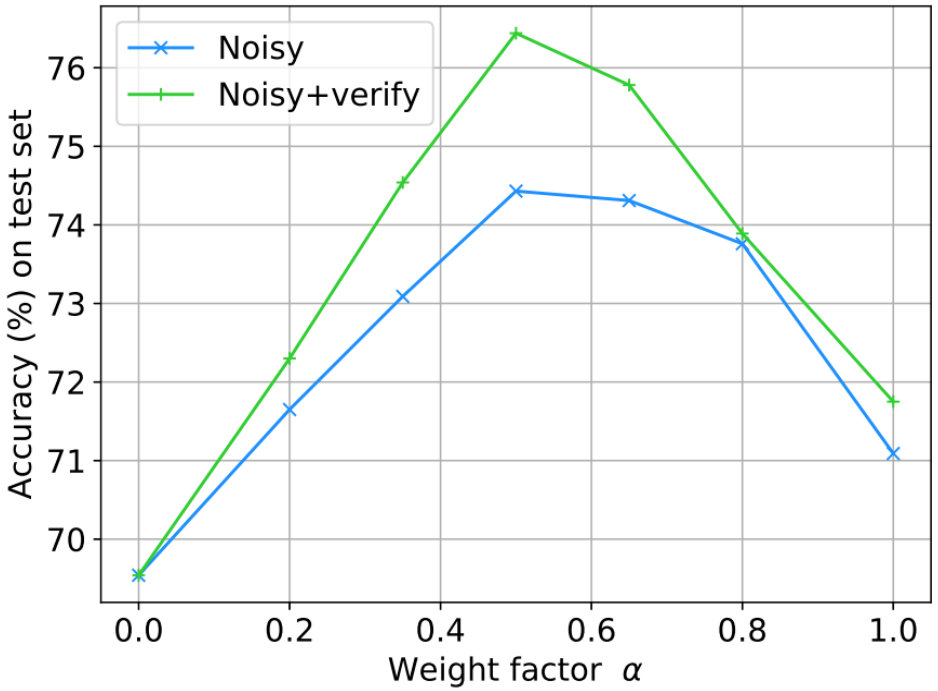
Label Correction Accuracy

	Original	Correct Initial	Correct Final
Accuracy	61.74	74.38	**77.36**

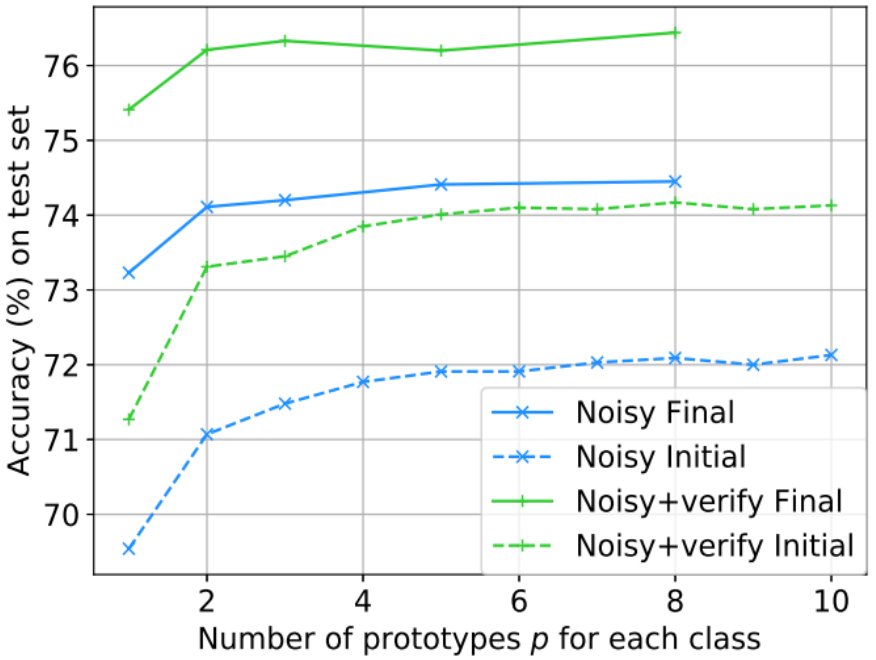
Table 2. Overall label accuracy (%) of the labels in original noisy dataset (Original), accuracy of the corrected label generated by the label correction phase in first iterative cycle (Correct Initial) and accuracy of the corrected labels generated by the final model when training ends (Correct Final).

Experiments

Weight factor α



Number of Class Prototypes



Thanks