Boosting Semi-Supervised Learning by Exploiting All Unlabeled Data

Supplementary Materials

A. Proof for the gradient of target class in EML

Here we provide the proof of the Eq.7 in Section 3.2 of the main paper. First, we can simplify y_c based on the output probability of target class p_{tc}:

$$y_c = \frac{1-p_{tc}}{C-1} \tag{1}$$

we omitted the superscript i in which denotes the sample here. The gradient of the EML loss with respect to the target class can be calculated by the chain rule:

$$\frac{\partial \mathcal{L}_{eml}}{\partial p_{tc}} = \frac{\partial \mathcal{L}_{eml}}{\partial y_c} \cdot \frac{\partial y_c}{\partial p_{tc}} \tag{2}$$

we can calculate the gradient of the first term from according to Eq.(7) of the main paper:

$$\begin{aligned}\frac{\partial \mathcal{L}_{eml}}{\partial y_c} &= -\frac{1}{BC}\sum_{c\in\ \mathbf{D}}[log(p_c) - log(1-p_c)]\\ &= \frac{1}{BC}\sum_{c\in\ \mathbf{D}}[log(1-p_c) - log(p_c)]\\ &= \frac{1}{BC}[\sum_{c\in\ \mathbf{D}} log(1-p_c) - \sum_{c\in\ \mathbf{D}} log(p_c)]\\ &= \frac{1}{BC}[log\prod_{c\in\ \mathbf{D}}(1-p_c) - log\prod_{c\in\ \mathbf{D}}(p_c)]\\ &= \frac{1}{BC}log\frac{\prod_{c\in\ \mathbf{D}}(1-p_c)}{\prod_{c\in\ \mathbf{D}}p_c}\end{aligned} \tag{3}$$

where $\mathbf{D} = \{c|c \in [1, C]\&c \neq tc\}$. we can compute the gradient of the second term from according to Eq.(2):

$$\frac{\partial y_c}{\partial p_{tc}} = -\frac{1}{C-1} \tag{4}$$

Thus, we can obtain the overall gradient according to Eq.(3) and Eq.(4):

$$\frac{\partial \mathcal{L}_{eml}}{\partial p_{tc}} == -\frac{1}{BC(C-1)}log\frac{\prod_{c\in\ \mathbf{D}}(1-p_c)}{\prod_{c\in\ \mathbf{D}}p_c} \tag{5}$$

which finishes the proof of Eq.(7) in the main paper. Furthermore, we will analysis the gradient directions of EML and cross-entropy loss are the same.

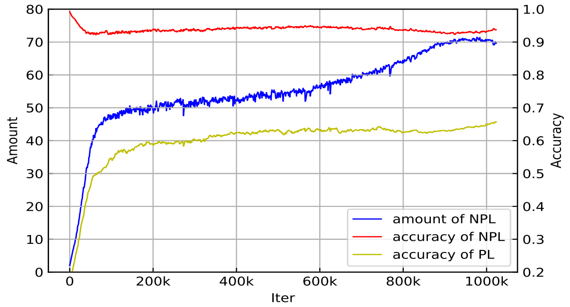

Figure 1. Visualize the experimental results on CIFAR-100 with 400 label samples.

For simplicity, we assume the first class is the target class (i.e., $tc = 1$), Eq.(5) can be written as:

$$\frac{\partial \mathcal{L}_{eml}}{\partial p_{tc}} = -\frac{1}{BC(C-1)}log\frac{(1-p_2)\cdots(1-p_C)}{p_2 \cdot p_3 \cdots p_C} \tag{6}$$

Since all confidence probabilities should larger than 0 and sum to 1 (i.e., $p_c > 0$ and $\sum_{c=1}^{C} p_c = 1$), we can obtain a series of inequalities:

$$\begin{aligned} 1-p_2 = p_1 + p_3 + \cdots + p_c > p_3, \cdots,\\ 1-p_{c-1} = p_1 + \cdots + p_{c-2} + p_c > p_c,\\ 1-p_c = p_1 + p_2 + \cdots + p_{c-1} > p_2,\end{aligned} \tag{7}$$

we multiply the above inequalities, yield:

$$\frac{(1-p_2)\cdots(1-p_C)}{p_2 \cdot p_3 \cdots p_C} > \frac{p_3 \cdots p_C \cdot p_2}{p_2 \cdot p_3 \cdots p_C} = 1 \tag{8}$$

Thus the gradient of Eq.(6) is a negative number. Meanwhile, the gradient of the cross-entropy loss with respect to the target class (i.e., $-1/p_{tc}$) is also less than zero. Therefore, our proposed EML not only constrains the non-target class to avoid them competition with the target class, but cooperating with cross-entropy to enhance the confidence of the target class so that selecting more examples with pseudo-label.

B. Analysis About ANL

B.1. Training with Limited Labeled Samples

In addition to Fig. 1(b), we further visualize the accuracy of NPL (i.e., negative pseudo label) during the training with

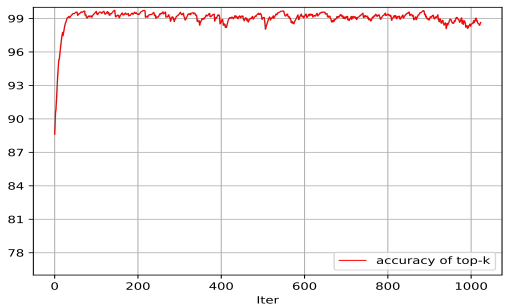

Figure 2. Visualizations the accuracy of top-k in the first 1K iterations on CIFAR-100 with 400 label samples.

limited labeled samples. As shown in Fig. 1, when operating on CIFAR-100 with 400 label samples (i.e., 4 labels per class), the accuracy of PL (i.e., pseudo label) (yellow) is just about 65%, while NPL (red) still maintains a very high accuracy (about 95%) without reducing the selected amount (i.e., k is not clearly changed). This further demonstrates the effectiveness of NPL.

B.2. Initial Noise

To illustrate that there is no noise perturbation in ANL even in the beginning of training, we visualize the accuracy of top-k in the first 1K iterations, as shown in Fig. 2. In the initial iterations, our calcauted k is close to the class number. Hence, the amount of selected NPL is very small (e.g., only assign the last-1 prediction as NPL). The possible noise is rather minor. After few iterations (no more than 20), the probability is extremely small that the positive class is the last-ranked one, so the accuracy is near to 1 without noise issue.

C. Analysis about EML

In this section, we compare some similar methods with our proposed EML, as shown in Table 1. Seeing col.2 vs col.3, it leads to a failure training when applying Label Smooth (LS) (target class label is set to 0.90) into unsupervised loss $\mathcal{L}_{us}$, since it will decrease the score of the target class thus leading to missing abundant pseudo-label samples. When using LS (0.98), the issue can be significantly alleviated. The last 3 columns show Entropy Regularization (ER) can benefit FixMatch and can be used with EML in a complementary fashion.

	FixMatch	LS(0.90)	LS(0.98)	ER	EML	EML+ER
Acc	92.48	64.59	93.02	92.95	93.47	93.72

Table 1. The comparison of different methods on CIFAR-10@40.

Dataset	CIFAR-10	CIFAR-100	SVHN	STL-10	ImageNet
Model	WRN-28-2	WRN-28-8	WRN-28-2	WRN-28-2	ResNet-50
Weight Decay	0.0005	0.001	0.0005	0.0005	0.0003
Threshold τ	0.95	0.95	0.95	0.95	0.7
Labeled Data Batch Size	64	64	64	64	128
Unlabeled Data Ratio μ	7	7	7	7	1
Learning Rate	0.03				
SGD Momentum	0.9				
EMA Momentum	0.999				

Table 2. Hyperparameter settings of FullMatch/FullFlex for CIFAR-10, CIFAR-100, SVHN and STL-10.

D. Experiment and Algorithm

D.1. FullMatch Algorithm

We present the complete algorithm for FullMatch in Algorithm 1. Please refer the main paper for all symbols and equations.

D.2. Implementation Details

For reproduction, we present the complete list of hyperparameters for FullMatch and FullFlex when operating in different benchmarks, as shown in Table 2, which are mainly consistent with TorchSSL settings.

Algorithm 1 FullMatch algorithm.

```
Input: $\mathcal{X} = \{(x_m, y_m) : m \in (1, ..., M)\}, \mathcal{U} = \{\mu_n : n \in (1, ..., N)\}$, $\tau$ is the confidence threshold. {M labeled data and N unlabeled data};
while not reach the maximum iteration do
    Generate $P^{(i)}$ and $Q^{(i)}${Predictions of strongly-augmented and weakly-augmented version, respectively};
    Calculate $k$ using Eq. (8) {Generate negative pseudo-labels for all unlabeled examples};
    Calculate ANL loss $\mathcal{L}_{anl}$ using Eq. (10) for all unlabeled examples;
    if $max\ (Q^{(i)}) \geq \tau$ then
        Calculate unsupervised loss $\mathcal{L}_{us}$ using Eq. (2) ;
        Calculate $y_c$ using Eq. (5) { Determine the label of non-target categories};
        Calculate EML $\mathcal{L}_{eml}$ using Eq. (6);
    else
        $\mathcal{L}_{eml} = 0, \mathcal{L}_{us} = 0$. {Ignore the examples without pseudo-labels};
    end if
    Calculate supervised loss $\mathcal{L}_s$ using Eq. (12);
    Update model via $\mathcal{L}_{sum} = \mathcal{L}_s + \mathcal{L}_{us} + \mathcal{L}_{anl} + \mathcal{L}_{eml}$;
end while
return Model parameters
```