FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling

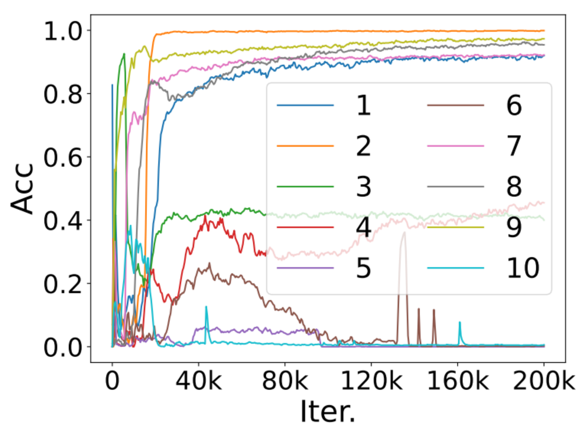
Bowen Zhang*[1], Yidong Wang*[1], Wenxin Hou[2], Hao Wu[1],
Jindong Wang†[3], Manabu Okumura†[1], Takahiro Shinozaki.†[1]

[1] Tokyo Institute of Technology, [2] Microsoft, [3] Microsoft Research Asia.
{bowen.z.ab, wang.y.ca}@m.titich.ac.jp*

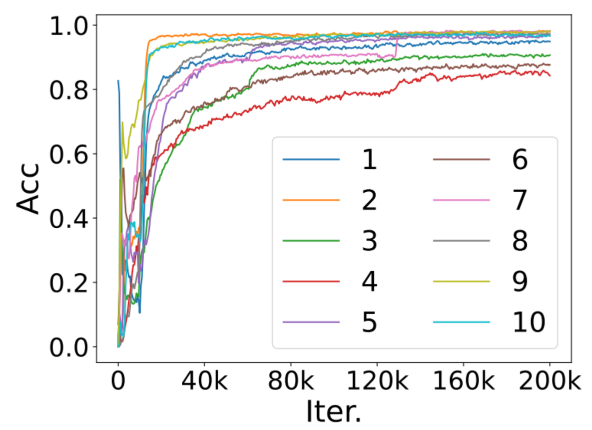
Summary

- We propose Curriculum Pseudo Labeling (CPL) to improve the convergence and accuracy of SSL. *(https://arxiv.org/abs/2110.08263)*
- FlexMatch, the integration of FixMatch and CPL, achieves state-of-the-art results.
- We open-source TorchSSL, a unified PyTorch-based semi-supervised learning codebase for the fair study of SSL algorithms. *(https://github.com/TorchSSL/TorchSSL)*

Motivation



(c) FixMatch: 56.4%



(d) FlexMatch: 94.3%

With a high fixed threshold, samples of hard-to-learn classes with low confidence are more likely to be filtered out.

With a high fixed threshold, each batch may contain more easy samples than difficult ones which is bad for the global convergence and the final accuracy of difficult classes.

With flexible thresholds, FlexMatch (d) is able to improve the accuracy of difficult classes compared to FixMatch (c) in the early stage of training.

Method

We estimate the learning effect of a class as the number of samples whose predictions fall above a high fixed threshold and into this class:

$$\sigma_t(c) = \sum_{n=1}^{N} \mathbb{1}(\max(p_{m,t}(y|u_n)) > \tau) \cdot \mathbb{1}(\arg\max(p_{m,t}(y|u_n)) = c) \quad (5)$$

We then apply the following normalization to $\sigma_t(c)$ to make its range between 0 and 1. Particularly, we further rewrite the denominator as the max of the best-learned class and unused class, which can be considered as a threshold warm-up process.

$$\beta_t(c) = \frac{\sigma_t(c)}{\max_c \sigma_t} \quad (6) \qquad \beta_t(c) = \frac{\sigma_t(c)}{\max\left\{\max_c \sigma_t, N - \sum_c \sigma_t\right\}} \quad (11)$$

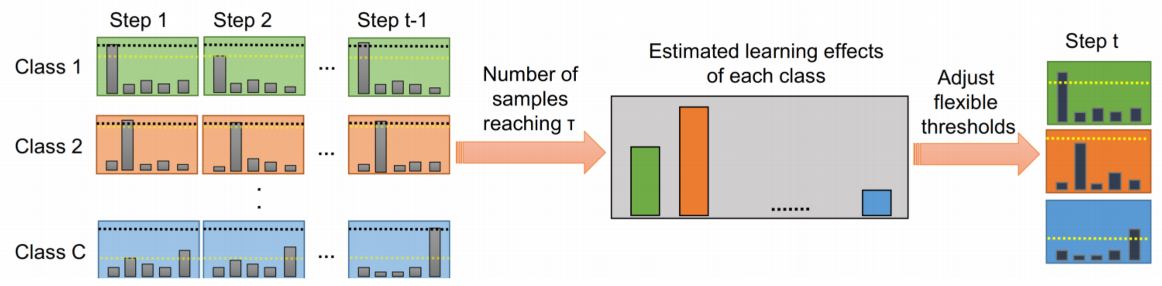
Finally, the normalized estimated learning effects are used to scale the pre-defined high threshold *for different classes at different time steps*. A customized mapping function can be further applied as described in Eq.(12):

$$\mathcal{T}_t(c) = \beta_t(c) \cdot \tau \quad (7) \qquad \mathcal{T}_t(c) = \mathcal{M}(\beta_t(c)) \cdot \tau \quad (12)$$

Putting all these together, we define the unsupervised loss, supervised loss, and the total loss as:

$$\mathcal{L}_{u,t} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) > \mathcal{T}_t(\arg\max(q_b))) H(\hat{q}_b, p_m(y|\Omega(u_b))) \quad (8) \qquad \mathcal{L}_t = \mathcal{L}_s + \lambda \mathcal{L}_{u,t} \quad (9) \qquad \mathcal{L}_s = \frac{1}{B} \sum_{b=1}^{B} H(y_b, p_m(y|\omega(x_b))) \quad (10)$$

Algorithm



Algorithm 1 FlexMatch algorithm.

Input: $\mathcal{X} = \{(x_m, y_m) : m \in (1, \ldots, M)\}$, $\mathcal{U} = \{u_n : n \in (1, \ldots, N)\}$ {M labeled data and N unlabeled data.}
$\hat{u}_n = -1 : n \in (1, \ldots, N)$ {Initialize predictions of all unlabeled data as -1 indicating unused.}
while not reach the maximum iteration **do**
  for $c = 1$ to C **do**
    $\sigma(c) = \sum_{n=1}^{N} \mathbb{1}(\hat{u}_n = c)$ {Compute estimated learning effect.}
    if $\max \sigma(c) < \sum_{n=1}^{N} \mathbb{1}(\hat{u}_n = -1)$ **then**
      Calculate $\beta(c)$ using Eq. (11) {Threshold warms up when unused data dominate.}
    else
      Calculate $\beta(c)$ using Eq. (6) {Compute normalized estimated learning effect.}
    end if
    Calculate $\mathcal{T}(c)$ using Eq. (7) {Determine the flexible threshold for class c.}
  end for
  for $b = 1$ to μB **do**
    if $p_m(y|\omega(u_b)) > \tau$ **then**
      $\hat{u}_b = \arg\max q_b$ {Update the prediction of unlabeled data u_b.}
    end if
  end for
  Compute the loss via Eq. (8), (10) and (9).
end while
Return: Model parameters.

Results

Table 1: Error rates on CIFAR-10/100, SVHN, and STL-10 datasets. The 'Flex' prefix denotes applying CPL to the algorithm, and 'PL' is an abbreviation of Pseudo-Labeling. STL-10 dataset does not have label information for unlabeled data, thus its fully-supervised result is unavailable.

Dataset	CIFAR-10			CIFAR-100			STL-10			SVHN	
Label Amount	40	250	4000	400	2500	10000	40	250	1000	40	1000
PL	69.51±4.55	41.02±3.56	13.15±1.84	86.10±1.50	58.00±0.38	36.48±0.13	74.48±1.48	55.63±5.38	31.80±0.29	60.32±2.46	9.56±0.25
Flex-PL	**65.41**±1.35	**36.37**±1.57	**10.82**±0.04	**74.85**±1.53	**44.15**±0.19	**29.13**±0.26	**69.26**±0.60	**41.28**±0.46	**24.63**±0.14	**36.90**±1.19	**8.64**±0.08
UDA	7.33±2.03	5.11±0.07	4.20±0.12	44.99±2.28	27.59±0.24	22.09±0.19	37.31±3.03	12.07±1.50	6.65±0.25	4.40±2.31	**1.93**±0.01
Flex-UDA	**5.33**±0.13	**5.05**±0.02	**4.07**±0.06	**33.64**±0.92	**24.34**±0.20	**20.07**±0.13	**12.84**±2.60	**8.05**±0.21	**5.77**±0.08	**3.78**±1.67	1.97±0.06
FixMatch	6.78±0.50	4.95±0.07	4.09±0.02	46.76±0.79	28.15±0.81	22.47±0.66	35.42±6.43	10.49±1.03	6.20±0.20	**4.36**±2.16	**1.97**±0.03
FlexMatch	**4.99**±0.16	**4.80**±0.06	**3.95**±0.03	**32.44**±1.99	**23.85**±0.23	**19.92**±0.06	**10.87**±1.15	**7.71**±0.14	**5.56**±0.22	5.36±2.38	2.86±0.91
Fully-Supervised	4.45± 0.12			19.07± 0.18			-			2.14± 0.02	

Table 2: Error rate results on ImageNet after 2^{20} iterations.

Method	Top-1	Top-5
FixMatch	43.08	19.55
FlexMatch	**35.21**	**13.96**

- CPL achieves better performance on tasks with limited labeled data.
- CPL improves the performance of existing SSL algorithms.
- CPL achieves better performance on complicated tasks.