MixMatch: A Holistic Approach to Semi-Supervised Learning

Presented by : AhmedElmogtaba

Authors : David Berthelot , Nicholas Carlini et.al

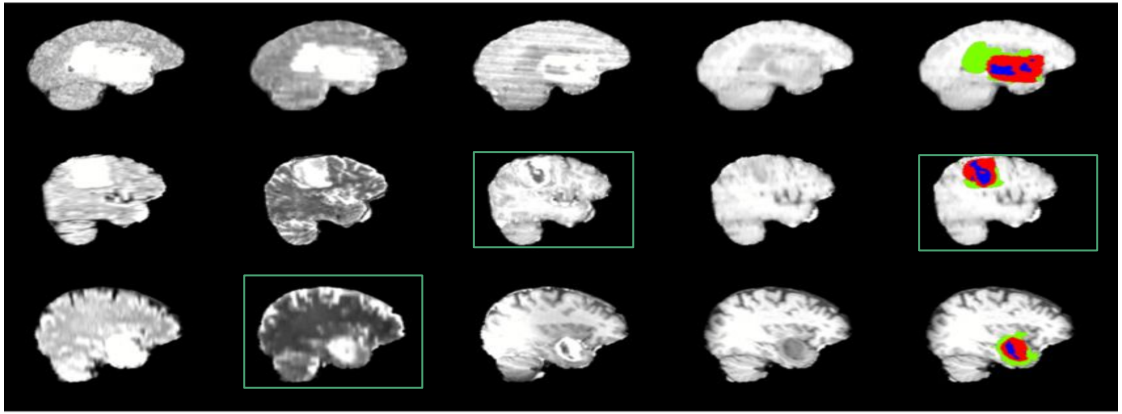
What is Semi-Supervised Learning ?

- **Semi-supervised learning** is a learning problem that involves a small number of labeled examples and a large number of unlabeled examples.

- It is a type of machine learning that sits between supervised and unsupervised learning.

Why Semi-Supervised Learning ?

- Simply because World is full of data ====> but labeling takes time and money
- So , it will be a good thing if we use the unlabeled data effectively.



Let’s go back in time a little bit before Mixmatch …….

Much Recent Approaches :

Many recent approaches for semi-supervised learning add a loss term which is computed on

unlabeled data and encourages the model to generalize better to unseen data.

This loss term falls into one of three classes :

- **entropy minimization** : which encourages the model to output confident predictions on unlabeled data.
- **consistency regularization** : which encourages the model to produce the same output distribution when its inputs are perturbed.
- **generic regularization** : which encourages the model to generalize well and avoid overfitting the training data.

So , What Mixmatch did ? . . .

They tried to use all the previous three concepts

Together ======>


Augmentation
Mix up
Sharpening

In order to tackle ======>

Semi-Supervised Learning problem

Hence the name of the paper ======> “Holistic approach”

The Three concepts & Mixmatch :

- **Consistency Regularization :** MixMatch utilizes a form of consistency regularization through the use of standard data **augmentation** for images (**random horizontal flips and crops**).

- **Entropy Minimization :** MixMatch also implicitly achieves entropy minimization through the use of a **“sharpening” function** on the target distribution for unlabeled data

- **General Regularization :** Mixmatch utilize **MixUp** as both as a regularizer (applied to labeled data points) and a semi-supervised learning method (applied to unlabeled data points).

The Algorithm :

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \ldots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \ldots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.

for $b = 1$ **to** B **do**

  $\hat{x}_b = \text{Augment}(x_b)$ *// Apply data augmentation to x_b*

  for $k = 1$ **to** K **do**

    $\hat{u}_{b,k} = \text{Augment}(u_b)$ *// Apply k^{th} round of data augmentation to u_b*

  end for

  $\bar{q}_b = \frac{1}{K} \sum_k \mathrm{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of u_b*

  $q_b = \text{Sharpen}(\bar{q}_b, T)$ *// Apply temperature sharpening to the average prediction (see eq. (7))*

end for

$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \ldots, B)\big)$ *// Augmented labeled examples and their labels*

$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \ldots, B), k \in (1, \ldots, K)\big)$ *// Augmented unlabeled examples, guessed labels*

$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ *// Combine and shuffle labeled and unlabeled data*

$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \ldots, |\hat{\mathcal{X}}|)\big)$ *// Apply MixUp to labeled data and entries from $\mathcal{W}$*

$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \ldots, |\hat{\mathcal{U}}|)\big)$ *// Apply MixUp to unlabeled data and the rest of $\mathcal{W}$*

return $\mathcal{X}', \mathcal{U}'$

Sharpening :

$$Sharpen(p, T)_i := p_i^{\frac{1}{T}} \Big/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$

- This step is inspired by the success of entropy minimization in semi-supervised learning.
- Given the average prediction over augmentations a sharpening function is applied to reduce the entropy of the label distribution.
- T (temperature) is a hyper-parameter to adjust this categorical distribution.

Mix up :

$$\lambda \sim \operatorname{Beta}(\alpha, \alpha)$$
$$\lambda' = \max(\lambda, 1 - \lambda)$$
$$x' = \lambda' x_1 + (1 - \lambda') x_2$$
$$p' = \lambda' p_1 + (1 - \lambda') p_2$$

They used MixUp for semi-supervised learning .

Unlike past work for SSL they mix both labeled examples and unlabeled examples with label guesses.

They defined a slightly modified version of MixUp , because they needed to preserve the order of the batch to compute individual loss components appropriately , they did this using the second equation

Which ensures that x' is closer to x_1 than to x_2 .

Loss Function :

$$
\begin{aligned}
\mathcal{L}_{\mathcal{X}} &= \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \mathrm{H}\left(p, \mathrm{p}_{\text{model}}\left(y \mid x; \theta\right)\right) \\
\mathcal{L}_{\mathcal{U}} &= \frac{1}{L\,|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \left\| q - \mathrm{p}_{\text{model}}\left(y \mid u; \theta\right) \right\|_2^2 \\
\mathcal{L} &= \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}
\end{aligned}
$$

- They used the standard semi-supervised loss .
- combined the typical cross-entropy loss with the squared L2 loss.
- unlike the cross-entropy, it is bounded and less sensitive to incorrect predictions.

Experiments

They did the Experiments on :

1- CIFAR10 dataset

2 - CIFAR100 dataset

3- SVHN

4- STL10

Implementation details and Results :

- They used the “Wide ResNet-28” model .instead of decaying the learning rate, they evaluated models using an exponential moving average of their parameters with a decay rate of 0.999. Second, they applied a weight decay of 0.0004 at each update for the Wide ResNet-28 model.

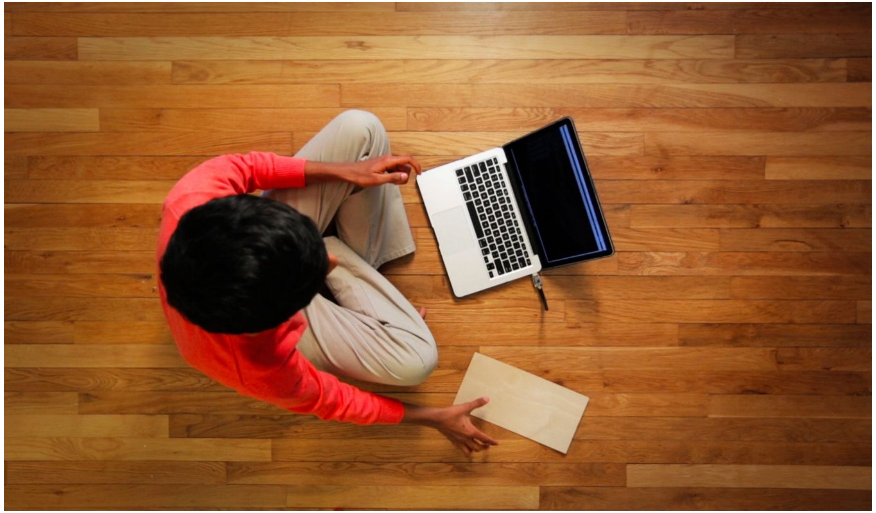
- on CIFAR-10 with 250 labels, they reduced the error rate by a factor of 4 (from 38% to 11%)

- On CIFAR-10 with 4000 labels the accuracy was 93.76 .

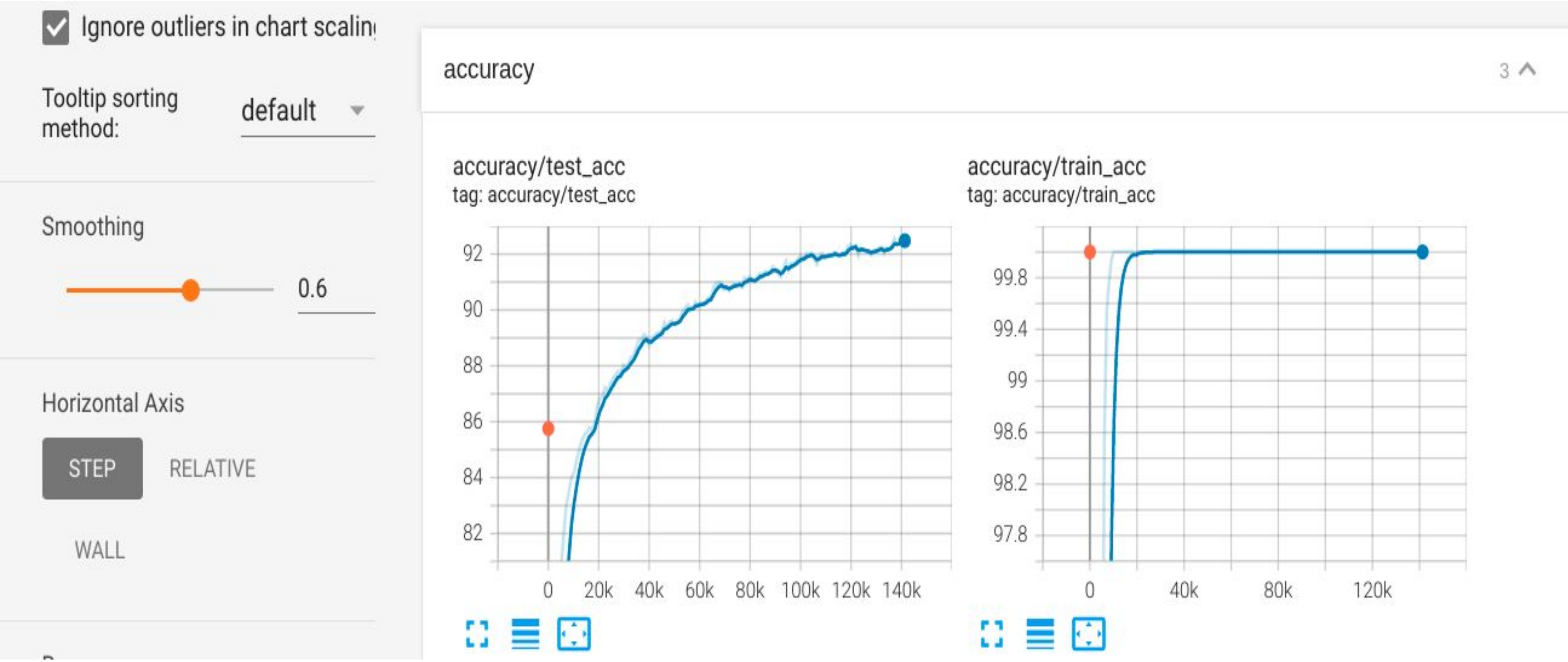
My Experiments

I've implemented Mixmatch and applied it on CIFAR 10 dataset with :

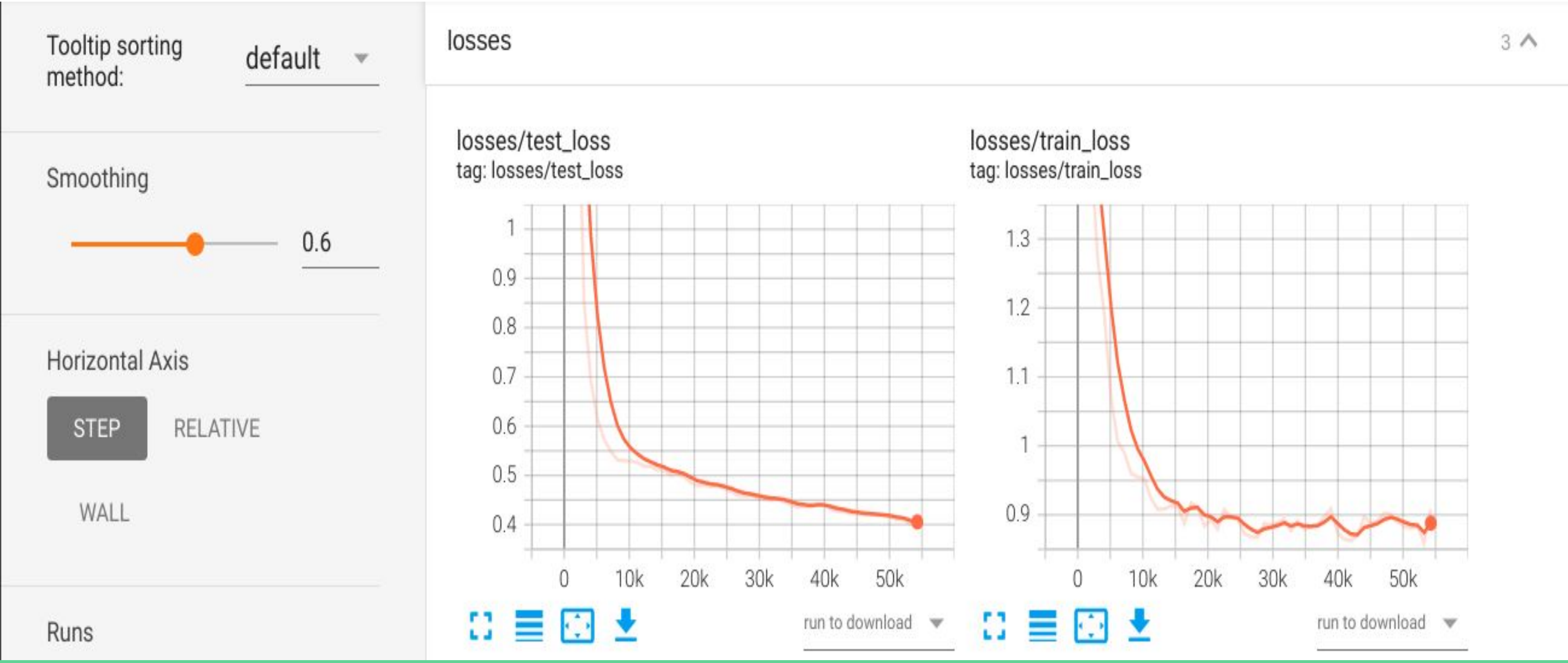
1. 250 labeled data
2. 1000 labeled data
3. 4000 labeled data



Graphs :


Ignore outliers in chart scaling
Tooltip sorting method:
default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
accuracy
3
accuracy/test_acc
tag: accuracy/test_acc
92
90
88
86
84
82
0
20k
40k
60k
80k
100k
120k
140k
accuracy/train_acc
tag: accuracy/train_acc
99.8
99.4
99
98.6
98.2
97.8
0
40k
80k
120k

Graphs :


Tooltip sorting method:
default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
losses
3
losses/test_loss
tag: losses/test_loss
1
0.9
0.8
0.7
0.6
0.5
0.4
0
10k
20k
30k
40k
50k
run to download
losses/train_loss
tag: losses/train_loss
1.3
1.2
1.1
1
0.9
0
10k
20k
30k
40k
50k
run to download

Comparison :

Number of Labels	250	1000	4000
Paper	88.92	92.25	93.76
This Code	86.7	90.2	92.5

Thanks For Your Attention !