# Semi-supervised learning using MixMatch on CIFAR10

Sai Kasyap Kamaraju
email:saikasyap@asu.edu

## Abstract

*Semi-supervised learning has demonstrated to be a ground-breaking for utilizing unlabeled data points to free the dependency on big labeled data points. In this project, a single loss is implemented that combines the present various superior techniques for semi-supervised learning to produce a new algorithm, MixMatch, to generate low entropy labels for data-augmented unlabeled points and mixing labeled and unlabeled data points using MixUp.. This study shows that MixMatch gives impressive results by a significant margin on CIFAR10 dataset with varying labeled data points. For example, on CIFAR-10 with 250 labels, the error rate is reduced by 4 times (from 60% to 15%) in comparison, with entropy minimization SSL technique. Finally, we perform an ablation study to determine elements of MixMatch that are at most significant and prove that the MixMatch is greater than sum of its individual elements [1].*

## 1. Introduction

Unsupervised learning learns from unlabeled data points only while the semi-supervised learning exploits the advantage of using both labeled and unlabeled data points. Semi-supervised learning is helpful in preventing the overfitting of the model; when the labeled data points are limited it doesn't take much time for a deep neural network to memorize the overall training data. Most semi-supervised techniques use unlabeled data points as a regularizer for training labeled data set.

SSL aims to leverage the extra information about the input data distribution to make a prediction on unlabeled data using only miniscule of labeled data. This is can be highly helpful in the medical domain where you have data labels from expensive machinery and tedious analysis from multiple human experts.

In this project, we will be discussing and implementing Semisupervised learning algorithms such as Mix match algorithm [1] and entropy minimization [2] on CIFAR10 data set. Introduced in May, 2019, by members of the Google Brain team et al. [1] as a semi-supervised learning method which is a blend and enhanced version of many methodologies that have been citied in recent years, including: Mean Teacher [3], Virtual Adversarial Training [4], and Mixup [5,6]. MixMatch is a semi-supervised learning algorithm which greatly outplayed the previous SSL approaches. Model trained on CIFAR10 dataset with 250 labeled data points using MixMatch is significantly better when compared to the traditional Entropy Minimization approach by factor of 4 on the error rate (16% vs 62.78%); as a reference the supervised model is trained on all 50,000 images has an error rate of 15.67%[1]

MixMatch generates labels on the unlabeled data points utilizing label "guesses" and it applies regularization overwhelmingly in many ways. Firstly, data augmentation is performed several times and the averages for the label "guessing" is taken. These predicted labels are then 'sharpened' to decrease the entropy. At last, MixUp is executed on the labeled and unlabeled data points.

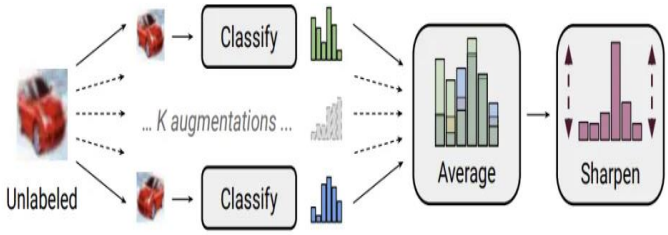


Figure 1: Diagram for generating label predictions in Mixmatch (picture taken from Fig. 1 [1])

## 2. Related Work

Though there are many SSL techniques like transductive models, generative models and graph based methods, we mainly focus on the SSL techniques from which MixMatch algorithm is derived.

In this section, we assume a generic probability distribution $P(y \mid x; \theta)$ with class labels y for an input x with parameters $\theta$[1].

### 2.1 Consistency Regularization

Consistency regularization applies data augmentation to semi-supervised learning by utilizing the concept that a classifier needs to have same class distribution for an unlabeled data after its augmentation. Consistency regularization ensures that if we augment a data point, the label should not change. Augmentation is basically creating a new data point from the existing data point.i.e. encourage same prediction on slightly perturbed inputs. Predictions shouldn't be changed dramatically if inputs only change

slightly. Hence, they use domain specific data augmentation which is a disadvantage. MixMatch utilizes a form of consistency **regularization** using standard data augmentation for images (random horizontal flips and crops)[7].

### 2.2 Entropy Minimization

Entropy in simple terms can be defined as the randomness. If we look at it from the data perspective, entropy tells us how mixed up are the classes around a point or region. More mixed up the classes, more the entropy and less sure we are about which class the region belongs to [1].

### 2.3 Traditional Regularization

This encourages the model to have a strictly linear relationship between examples so that it is harder for the model to remember the training data and make it generalized by imposing a constraint (for example: L2). This objective is achieved in Mixmatch algorithm by using Mix up that requires the output of the convex combination of inputs to be close to the convex combination of their individual outputs [7].

## 3. MixMatch

As said in section 2, Mixmatch is combination of various effective SSL techniques which are given in the following sections below.

.

### 3.1 Data Augmentation

Data augmentation is commonly used consistency regularization technique in computer vision where rotation, cropping, zooming, brightening, etc are used so that the overall content of the image is not changed. New images are generated using MixMatch while doing augmentation several times making the generated ones are stable. The findings in this paper [1] is that K =2 augmentations were enough.

### 3.2 Label Guessing

To predict labels the mix match algorithm uses a "guess" which is used in the unsupervised loss term. This is calculated as a mean of model predictions distribution across K augments as shown in Figure 1.

#### 3.2.1 Sharpening

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

Equation 1: Taken from et.al [1, equation 7]

The model's predictions are sharpened on the unlabeled data using the above equation as a form of entropy minimization. If the temperature **T** < 1, it makes sure predicted values to be true, and as **T** tends to zero the predicted values form a one-hot distribution that has least entropy (as seen in code). However, when T approaches infinity, the vector gives rise to high entropy – all values are same. In the ablation study performed, there is an accuracy reduction of over 10% when sharpening hyperparameter is removed (where **T** =1).

### 3.3 MixUp

Mixup was introduced by Zhang and others et.al [5] is a form traditional regularization that encourages the model to strictly linear relationship between the data points. In other words, it computes linear predictions between two individual labeled data points which are fed to the model. The one hot encoded labels of the images are also predicted, using the same λ coefficient as the images. That coefficient is randomly drawn from the beta distribution which has parameter α. Generally, α needs to be tuned as per dataset. At small values of α, in results small values of Mix up or no value when α = 0. As α increases, or approaches infinity, larger values bias towards Mixup making it maximum. Thus, α be controls the intensity of the Mix-up.

$$\lambda \sim Beta(\alpha, \alpha)$$
$$\lambda' = max(\lambda, 1-\lambda)$$
$$Mixup(a,b) = \lambda' * a + (1-\lambda') * b$$

Equation2: taken from [7],[1]

The paper[1] recommends a value of .75 for α. x' is the convex combination of the inputs and p' is the convex combination of the corresponding outputs in the Equation 2.

### 3.4 The MixMatch Algorithm

The full algorithm can be implemented in the following way:

1. Batches of labelled data points with their one hot labels and unlabeled data points are as considered inputs along with hyperparameters T, K and α
2. For all labelled data points in every batch augmentation is done for one time.
3.Augmentation of every unlabeled data point in every batch is done for K times to get new unlabeled data points.
4. The model is run for each data point in the K unlabeled augmented batch and their mean is taken to generate pseudo labels.
5. Sharpen the pseudo-labels to minimize the entropy.
6.Formation of sets X and U for augmented labeled data points and augmented unlabeled data points respectively.
7.Both sets X and U are combined and shuffled to form set W.
8. Labeled data points in set X are mixed up with first |X| entries of W to get X' where as |X| is the size of the labeled data in the batch.
9. Unlabeled data points in the batch are "mixed up" with rest of the entries of W to get U'.
10. The loss is calculated by subjecting both sets X' and U' to the model using the corresponding mixed-up labels [6].

### 3.5 Loss

The losses in the labeled and unlabeled are combined to get the loss function. The corresponding loss function for the labeled loss is the cross entropy loss H(a,b) while unlabeled generated predictions use L2 loss as they are less sensitive to false predictions. As the unlabeled predictions are computed by the model itself, the algorithm will not punish false predictions severely. The parameter λ keeps the balance as a regularizer.

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$$
$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta))$$
$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2$$
$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}$$

Equation3: (eqn 2,3,4,5) from [1], corresponding to the labeled loss, unlabeled loss, and total loss.

### 3.6 Hyperparameters

As per paper [1], the values sharpening temperature T and number of augments K are relatively constant at 0.5 and 2 respectively, while beta distribution parameter α and unlabeled loss weight λ need to be tuned per dataset and are equal to 75 and 0.75 in this project.

## 4. Experiment

This project is implemented on CIFAR10 data to study the effectiveness of MixMatch algorithm in comparison with Entropy Minimization and supervised learning techniques. An ablation study is also performed to access each part of MixMatch's components.

### 4.1 CNN Model

Initially, I tried using modern CNNs WideResNet (as in [1], 28 M parameters) but the computational time is very high. Then switched to Googles' Efficient B0(6M parameters) [13] which was reducing the computational time to less than half giving the same amount of accuracy for predictions. However, due to time constraint, the classic CNN model LeNet [14] is used for this project.

## 5. Results and Discussion

For reference, these tests are conducted on single GPU machine (RTX 2070) using pytorch and Fastai library. Here the Mixmatch algorithm is compared to entropy minimization [2] technique(baseline). Intially, the supervised model is trained with all 50,000 images. Next it is trained on varying labeled sizes 250,500,1000,2000,4000 with no unlabeled data. In the last leg, MixMatch is trained using the learner defined above. The Information of these computations can be seen in the code.

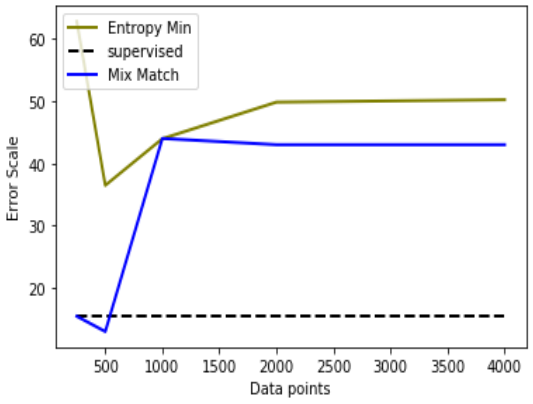


Figure 2: Comparison of error rate MixMatch, entropy minimization and supervised technique on varying data points

### 5.1 Error plot Analysis:

As we can observe from the Figure 2, the mix match algorithm outperforms the Entropy Minimization all times. The difference between supervised and MixMatch algorithm increased as data labels were increased because the number of epochs considered for those data points were quite less (2 epochs considered) when compared to initial data points where epochs were around 30 for 250 labeled data points.

### 5.2 Ablation study

The ablation study (as shown in the Table 1) calculated for Mix match algorithm are similar in results as published in the paper [1]. The only aberration was while considering 4000 data points the values calculated should be less than while considering 250. This may be due to number of epochs considered and fine tuning of the model considered. The study also confirms generation of label guesses using averaging, sharpening the distributions and MixUp significantly supports the performance of MixMatch. The unsupervised data is likely important for Mixup to perform better.

## 6. Conclusion

MixMatch clearly outperforms the Entropy Minimization technique, but on the downside the computation time it takes to get the performance. Training the supervised model is quicker relative to training MixMatch. Ineffective execution, multiple data augmentations and label "guessing" are drawbacks to its computational complexity. It took me more than 10 hours to train the wideResnet as implemented in the paper for a single calculation. A few more hours of training will improve the accuracy vastly, with last few epochs taking lot of time. This study is trained with less than 5 epochs for most of the data labeled points hence the aberration in few results. Also, model has needs be fine-tuned to get better accuracy. The augmentation and sharpening is significant, and the most important aspect in reducing the error rate, is MixUp, as it imposes linear interpolations between images and generated labels as observed in the study. The MixUp generalizes the model by minimizing memorizations of training data so does data augmentation up to certain extent. The effectiveness of Mix Match as an SSL technique in other domains such as NLP and transfer learning are yet to be evaluated. Also incorporating, other improved data augmentation techniques as UDA [16] would further improve its performance.

| Ablation | 250 labels | 4000 labels |
|---|---|---|
| MixMatch | 15.46% | 42.7% |
| Mixmatch without distribution averaging (K=1) | 31.76% | 47.4% |
| Mixmatch with K =3 | 13.58 % | 42.3 % |
| Mixmatch with K =4 | 26.3 % | 41.6% |
| Mixmatch without Temperature sharpening (T=1) | 35.4 % | 47.8% |
| MixMatch with parameter EMA | 15.23 % | 39.8% |
| MixMatch without Mixup | 27.2 % | 48.9% |
| Mixmatch with Mixup on labeled only | 14.6 % | 41.2% |
| Mixmatch with Mixup on unlabeled | 15.2 % | 42.3% |
| Mixmatch with Mixup on separate labeled and unlabeled | 16.3 % | 43.5% |
| Entropy Minimization | 62.78 % | 50.22 % |

Table 1: Ablation study of MixMatch Algorithm.

## References


[1] Berthelot, David, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. “MixMatch: A Holistic Approach to Semi-Supervised Learning.” ArXiv:1905.02249 [Cs, Stat], May 6, 2019.

[2] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In Advances in Neural Information Processing Systems, 2005..

[3] Tarvainen, Antti, and Harri Valpola. “Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results.” ArXiv:1703.01780 [Cs, Stat], March 6, 2017.

[4] Miyato, Takeru, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. “Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning.” ArXiv:1704.03976 [Cs, Stat], April 12, 2017

[5] Zhang, Hongyi, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. “Mixup: Beyond Empirical Risk Minimization.” ArXiv:1710.09412 [Cs, Stat], October 25, 2017

[6] https://mc.ai/a-fastai-pytorch-implementation-of-mixmatch/

[7] Guo, Hongyu, Yongyi Mao, and Richong Zhang. “MixUp as Locally Linear Out-Of-Manifold Regularization,” n.d., 9.

[8] Verma, Vikas, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. “Manifold Mixup: Better Representations by Interpolating Hidden States,” June 13, 2018

[9] https://mlexplained.com/2019/06/02/papers-dissected-mixmatch-a-holistic-approach-to-semi-supervised-learning-and-unsupervised-data-augmentation-explained/

[10] https://towardsdatascience.com/how-is-the-quiet-revolution-in-semi-supervised-learning-changing-the-industry-4a25f211ce1f

[11] https://github.com/noachr/MixMatch-fastai/blob/master/MixMatch%20Blog.ipynb

[12] https://paperswithcode.com/paper/mixmatch-a-holistic-approach-to-semi

[13] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv:1905.11946, 2019.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[15] https://github.com/kuangliu/pytorch-cifar

[16] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised Data Augmentation. arXiv e-prints, art. arXiv:1904.12848, Apr 2019.