Semi-supervised Image Classification

Kartik Teotia & Kevin Raj
7010830, 7015419
Universität des Saarlandes
kate00001, kepe00001@stud.uni-saarland.de

Abstract

Semi supervised learning takes a middle ground between supervised learning, which relies on labeled data and unsupervised learning, where we only have unlabeled data. Semi supervised learning thus uses a combination of supervised and unsupervised learning technique. In this technical report, we will discuss our approach for implementing a multitude of semi-supervised learning techniques and analyse their results. We also propose a style transfer based augmentation technique for semi supervised recognition tasks.

1 Introduction

Labeled samples in machine learning tasks are often either difficult, expensive, or time-consuming to obtain. The requirement of a human expert for obtaining labeled data is a major bottleneck for a wide variety of machine learning tasks. In this technical report, we gloss over broadly three different Semi-Supervised Learning techniques, which are pseudo-labeling [3], adversarial-example [2] based training, and a hybrid method [1] which happens to be our proposal at the end. We have explained the these techniques in respective sections.

2 Task1: Pseudo-Labeling

With Pseudo-Labeling, the idea is to train a model simultaneously on a batch of both labeled and unlabeled images. The model is trained on labeled images in a supervised manner with a cross-entropy loss,and the same model is used to get predictions for a batch of unlabeled images. We take the maximum probability class as the pseudo-label, and, then, a cross-entropy loss is calculated by comparing model predictions and the pseudo-label for the unlabeled images.

The total loss is a weighted sum of the labeled and unlabeled loss terms.

$$L = L_{labeled} + \alpha_t \times L_{unlabeled}$$

where, α_t is the weight factor for the loss from unlabeled data.

Implementation details: We used wide-resnet model without any dropout trained. Cifar10, the model width is set to 2 and trained for 50 epochs and the model width for cifar-100 is set to 8 and trained for 20 epochs. We used a batch size of 64 for both labeled and unlabeled data. The modeled is optimized using stochastic gradient descent with initial learning rate of 0.03 with nesterov momentum. In addition, we have used 1cycle learning rate policy for scheduling the learning rate during the training resulting in faster convergence. Comparison of the error-rates are given in Table.1.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Table 1: Error rates for CIFAR-10 and CIFAR-100 using different labeled samples using Pseudo-labeling technique.

	CIFAR-10						**CIFAR-100**					
# Labeled samples **Method**	250			4000			2500			10000		
Pseudo-Labeling	49.78±0.43			16.09±0.28			57.38±0.46			36.21±0.19		
Threshold	0.6	0.75	0.95	0.6	0.75	0.95	0.6	0.75	0.95	0.6	0.75	0.95
Ours	62.41	64.87	65.22	28.33	25.61	28.84	80.1	79.94	80.58	59.87	57.89	58.46

3 Task2: Variational Adversarial Training

The main idea with VAT is to generate an adversarial transformation of an image that will change the model prediction. To obtain this adversarial transformation, first, an image is taken and an adversarial variant of it is created in such a way that the KL-divergence between the model output for the original image and the adversarial image is maximized.

In our implementation, we take a labeled/unlabeled image and take its adversarial transformation. Then, the same model is used to predict label distributions for both the labeled/unlabeled and adversarial transformed image, and calculate the KL-divergence loss term for the two predictions. For labeled images, we calculate the cross-entropy loss. The final loss is a weighted sum of these two loss terms. A weight α_t is applied to decide how much the consistency loss contributes in the overall loss.

Table 2: Error rates for CIFAR-10 and CIFAR-100 using different labeled samples using VAT technique.

	CIFAR-10		**CIFAR-100**	
Method	250	4000	2500	10000
VAT	-	11.36 (±0.34)	-	-
Ours	66.57	32.8	80.09	59.24

Figure 1: Generated adversarial samples.

Implementation details: We used wide-resnet model without any dropout trained for 20 epochs. We used a batch size of 64 for both labeled and unlabeled data. The modeled is optimized using stochastic gradient descent with initial learning rate of 0.03 with nesterov momentum. In addition, we have used 1cycle learning rate policy for scheduling the learning rate during the training resulting in faster convergence. Comparison of the error-rates are given in Table.2.

4 Task3: Fixmatch combined with style transfer

This method was proposed by Sohn et al. [1] and combines pseudo-labeling and consistency regularization while vastly simplifying the overall method. It got state of the art results on a wide range of benchmarks.

As seen, we train a supervised model on our labeled images with cross-entropy loss. For each unlabeled image, weak augmentation and strong augmentations are applied to get two images. The weakly augmented image is passed to our model and we get prediction over classes. The probability for the most confident class is compared to a threshold. If it is above the threshold, then we take that class as the ground label i.e. pseudo-label. Then, the strongly augmented image is passed through our model to get a prediction over classes. This prediction is compared to ground truth pseudo-label using cross-entropy loss. Both the losses are combined and the model is optimized.

Our contribution: Data augmentation can destroy the image structures which can make the learning more difficult. To combat, we have introduced a image processing based style transfer technique called histogram matching. We further wanted to use neural style transfer for generating new labeled data, but due to time and resource constraint we have only implemented histogram matching. The structure of the labeled image is kept intact, but the style is influenced by the selected image of the same class from the unlabeled data if the prediction is above a certain threshold.

The problem with fixmatch is when the psuedo-label of the weak augmented image is incorrect, then the strong augmented image will also be incorrect. Which can lead to instability during training.

Whereas, in our approach even though the selected unlabeled image is of different class Fig.2, the style is transferred to a labeled sample. As we can see from the Fig.2, the image structure still remains same but the overall color of the image is different resulting in a new labeled image with color information from unlabeled data. This new labeled image is fed to the model and style loss is calculated. This results in more stable training and better accuracy.

$$L = L_{labeled} + \alpha \times L_{fixmatch} + \beta \times L_{style}$$

The total loss is a convex combination ie., α & $\beta \in (0, 1]$ of the standard cross-entropy loss between labeled data and target, fixmatch loss and style loss.

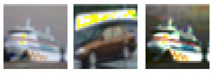

Figure 2: Generated style transfer image. 1st image is sampled from labeled data. 2nd image is selected from unlabeled data with confidence greater than threshold. 3rd image is the style transferred image.

Table 3: Error rates for CIFAR-10 and CIFAR-100 using different labeled samples using VAT technique.

	CIFAR-10		**CIFAR-100**	
Method	250	4000	2500	10000
Fixmatch	5.07±0.65	4.26±0.05	28.29±0.11	22.60±0.12
Fixmatch (our implementation)	41.09	14.61	-	-
Fixmatch + HistLoss	31.34	17.02	-	-

Implementation details: We used wide resnet model without any dropout and it is trained for 20 epochs. The original implementation of fixmatch [1] used batch size of 64 for labeled samples and 64×7 for unlabeled data. Due to gpu constraints we used a batch size of 64 for both labeled and unlabeled data. We have used expected moving average of the model similar to fixmatch. The modeled is optimized using stochastic gradient descent with initial learning rate of 0.03 with nesterov momentum. In addition, we have used 1cycle learning rate policy for scheduling the learning rate during the training resulting in faster convergence. The weights for the $L_{fixmatch}$ and L_{style} are set to one.

Results: The results for self-supervised task with style loss is compared with our implementation of fixmatch in-addition to the original fixmatch [1] Table.3. As we can see, the error-rate of the proposed technique is comparatively less than our fixmatch implementation, which highlights the importance of style loss and structure preserving data generation.

Note: For all the tasks, we restricted the training to just 20 epochs, by which the model is definitely did not converge. Plausibly, all the reported results can be improved upon further training.

References

[1] Sohn. K. et al. (2020) Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, pp. 596–608.

[2] Miyato, T., Maeda, S.I., Koyama, M. and Ishii, S. (2018) Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, pp. 1979-1993.

[3] Lee, D.H. (2013) Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning, ICML*, pp. 896.