Computer Vision Project - Semi-supervised Image Classification

Mohammed Maqsood Shaik
University of saarland
mosh00003@stud.uni-saarland.de

GowthamKrishna Addluri
University of saarland
goad00002@stud.uni-saarland.de

Abstract

Deep neural networks have become the important model for computer vision applications. Deep networks often achieve their strong performance through supervised learning, which requires a large labeled dataset at a cost of human labour. Semi-supervised learning (SSL) provides an effective means of leveraging unlabeled data to improve a model's performance. In this report, we summarize our work within the Computer Vision project of the Neural Networks: Theory and Implementation (WiSem2021/22) course at Saarland University, which focuses on understanding and implementing Pseudo-Labelling, Virtual adversarial training and slight variation of fixmatch with similarity metric (semi-supervised algorithms) on cifar10 and cifar100 datasets with varying amount of labeled data.

1 Introduction

Unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. It is also important to consider that in our experiments we made sure that the labeled data available is balanced among output classes.

In task1, we implemented a naive algorithm of Pseudo-Labeling[Lee, 2013], which is a simple and efficient method of semi-supervised learning for deep neural networks. Using the algorithm the neural network is trained in a supervised fashion with labeled and unlabeled data simultaneously. To get the Pseudo labels for unlabeled data the classes whose predicted probability is greater than a certain threshold are considered as true labels.

In task2, we implemented Virtual adversarial training algorithm[Miyato et al., 2018] to tackle the unlabeled data. Virtual adversarial training is an effective technique for local distribution smoothness. Pairs of data points are taken which are very close in the input space, but are very far in the model output space. Then the model is trained to make their outputs close to each other. Virtual adversarial training is mainly done on unlabeled dataset to make the model robust and the same model is used in the training of labeled dataset, with this the whole model gets generalized to have less evaluation error on test dataset even with less amount of labeled data.

In task3, we tried to improve upon fixmatch algorithm, which mainly combines self-supervised learning [Chen et al., 2020] with supervised learning to achieve semi-supervised learning. The self supervised learning mainly tries to learn the features of the dataset by making the output labels of weakly augmented and strongly augmented data same without using any labels and supervised learning learns the features on labeled dataset. These learnings happen in a parallel way on the same model, this makes it robust. To improve upon this we tried considering even weakly perturbed images(VAT)[Miyato et al., 2018] along with weakly augmented images in self supervised learning. As the strongly augmented data has no defined boundary it may cross the manifold of target class.

This can be prevented by assigning the same pseudo label from weakly augmented data to the strongly augmented data points having less L1 difference between them. This is the another improvement we added on fixmatch.

The report is organized as follows: in Section 2, we describe our implementation of all the three algorithms. In Section 3, we describe the experimental setup and analyse the obtained results, in Section 4, we explain our conclusions after working in this project.

2 Implementation

2.1 Pseudo-Labeling Algorithm

In this algorithm the neural network is trained in a supervised fashion with labeled and unlabeled data simultaneously. We implemented a check to make sure total number of unlabeled data when summed up with labeled data equals to the total number of training data samples. This is to prevent ambiguity in the training data. During initial process of training, model may not predict correct labels because of insufficient training for the unlabeled data, which we consider as pseudo labels when they are greater than a certain threshold and having the same sample with correct label leads to ambiguity in further training process. To get the Pseudo labels for unlabeled data, the classes whose predicted probability is greater than a certain threshold are considered as true labels and accumulated for the whole epoch and added to the labeled dataset for the next epoch onwards.

We tried different setups for this algorithm i.e training only with labeled data for initial set of epochs(eg. 10), then considering the pseudo labeled data as training data. But this does not have much effect on the output error rate. Our intuition for the failure of this setup is that restriction of adding Pseudo labels in the training data during initial epochs makes model to overfit on labeled data, making it hard to generate correct pseudo labels when compared to the original algorithm.

We did grid search on different hyper-parameters like learning rate(0.1,0.3), threshold(0.95,0.75,0.6), training batch size(32,64), model depth(28,34), iteration-per-epoch(1600,800) for CIFAR-10 and CIFAR-100 datasets with (250,4000) and (2500,10000) amounts of labeled data respectively on validation dataset(2% of training unlabeled dataset). We have used cosine learning rate scheduler [Loshchilov and Hutter, 2016]. We are additionally saving best model which gives highest accuracy for validation data during training, by this we can get a model which is having less variance because it is not overfitting.

2.2 Virtual Adversarial Training

In this algorithm there are two training losses involved, first one is related to labeled data and second one is related to unlabeled data. For labeled data we used cross-entropy loss between predicted label and target label, whereas for unlabeled data we use virtual adversarial training loss.

VAT loss is calculated using adversarial samples. First, a random tensor of sample data size is generated from a normal distribution of one standard deviation, which can act as random perturbation when added to input sample. But as we need an adversarial sample with the small variation in input domain and a large variation in output domain, we calculated gradient with respect to the perturbation added by using the KL-divergence between clean sample and perturbed sample. We iterated the process of calculating the gradient to get a sample which maximizes the KL- divergence and minimizes the L2norm distance between clean sample and perturbed sample. Now as we found the adversarial perturbation needed we generated a loss such that KL-divergence between the clean sample and perturbed sample is minimized.

By doing this we achieved smoothness in output distribution of the model, with respect to the input and the model generalizes better because it gives similar outputs for unseen data-points which are close to data-points in the training set.

We have tried to do VAT for both unlabeled and labeled data instead of only unlabeled data, but it gave worse results. Our intuition for its failure is, since we are calculating VAT loss for both labeled and unlabeled data over-generalization is happening which is nothing but increase in the bias. One important thing to note is while calculating the VAT loss the batch normalization should be set to evaluation model inorder to prevent continuing calculation of batch statistics during VAT loss calculation

We did grid search on different hyper-parameters like learning rate(0.001, 0.005), training batch size(32,64), model depth(28,34), iteration-per-epoch(1600,800), epsilon(2.0,5.0,8.0) for CIFAR-10 and CIFAR-100 datasets with (250,4000) and (2500,10000) amounts of labeled data respectively on validation dataset(2% of training unlabeled dataset). We have tried using cosine learning rate scheduler [Loshchilov and Hutter, 2016] with warmup for first 30 epochs of the total 50 epochs. Empirically we didn't find any advantage of adding this scheduler. We are additionally saving best model which gives highest accuracy for validation data during training by this we can get a model which is having less variance because it is not over-fitting. We have considered only '1' VAT power iteration to generate VAT loss and also the alpha regularization as '1' as suggested in the paper. Also we considered vat-xi as 0.5 throughout our experiment.

2.3 Fix-Match

In the Fix-Match algorithm the idea of generating loss for labeled data remains same as other tasks. But for unlabeled data we generate two differently augmented data from the same sample, one weakly augmented and another strongly augmented. By taking these two samples a cross-entropy loss is generated between Pseudo-label(generated from weak augmentation when its output is greater than a threshold) and output generated for strongly augmented data. This can be considered as consistency regularization. This is then combined with the supervised loss while training. Important point to note here is that model remains same for both labeled and unlabeled losses. We have used cosine learning rate scheduler [Loshchilov and Hutter, 2016]. We have not considered applying weight decay for batch-normalization because it causes periodic fluctuations due to variance fluctuating between very high and very low values [Lobacheva et al., 2021]. In the same way we did not consider for bias because they don't contribute to the curvature of the model, so there is usually little point in regularising them as well. We used EMA in our implementation as suggested in the paper as it is proved [Tarvainen and Valpola, 2017] that it helps in semi-supervised learning.

2.3.1 Changes considered for improvement

i) We tried adversarial perturbed image instead of strong augmented image as suggested by authors of Fixmatch [see Sohn et al., 2020, extension section] but the output error rate worsened. Our intuition is, in principle the strong augmented image is not a equivalent choice to a adversarial perturbed image as strong augmented image has a large variation in both input and output domain when compared to weak augmented image. Whereas, the adversarial perturbed image has small variation in input and large variation in output domain. This lacks the contrast required in the input domain between the perturbed images to learn the different features of unlabeled data. It is also proved that strong augmentation [Cubuk et al., 2019] is necessary for self-supervised learning.

ii) We considered changing loss function for unlabeled data from cross-entropy loss to KL-divergence between outputs of weak and strongly augmented samples, but we skipped this by the intuition that representations of the penultimate layer in a unsupervised learning network tries to align its features towards the loss and regularization used [Kornblith et al., 2020]. In our case, since our output is classification task it is better to stick with cross-entropy loss between Pseudo labels(weak augmented) and outputs(strong augmented), which is similar to any network with classification task as output.

iii) We considered changing the way of obtaining threshold instead of constant which is used in generating Pseudo-Label from the outputs of the weak augmented data as explained in Flexmatch paper [Zhang et al., 2021]. Since, it is already implemented in Flexmatch, we skipped this.

2.3.2 Actual changes done for improvement

An adversarial perturbed sample has similar properties with weak augmented perturbation in the input domain. Because of this we used Virtual adversarial training perturbed image along with weak augmented image to generate two Pseudo-labels for same image. In our case for each image we have one weak augmented image , one VAT perturbed image and one strong augmented image. Because of this we have two cross-entropy losses between strong augmented image output and two different Pseudo-labels. This additional loss can help to learn the features better as the adversarial perturbed image will have same label as clean sample from the initial point of training because we are getting the VAT perturbed image by maximizing KL-divergence between clean sample and adversarial perturbed sample but still having the same label, which is not true for weak augmented image3. This helps in reducing the number of epochs for training as training loss for unlabeled data is high even in initial epochs. But this effect diminishes as the number of epochs increases, the weak augmentation also have correct labels4. Note that We have used same threshold level for both VAT perturbed image and weak augmented image to generate Pseudo-labels.

We added an another improvement to prevent test error to go down because of strong augmented image. Our intuition behind this is that since we are applying strong augmentations to an image without any consideration of output class boundaries it may happen that strongly augmented image output can jump the output class manifold making our goal of having same label to a weakly augmented image and to a strongly augmented image wrong1. To prevent this we tested the L1 distance between outputs(strong augmented) of multiple class images in a batch. We made the Pseudo-Labels same for the outputs(strong augmented) whose L1 distance is less than or equal to 0.3. In choosing which label to assign between them we simply chose the hard label(chosen using threshold) whose predicted class output probability is highest2.

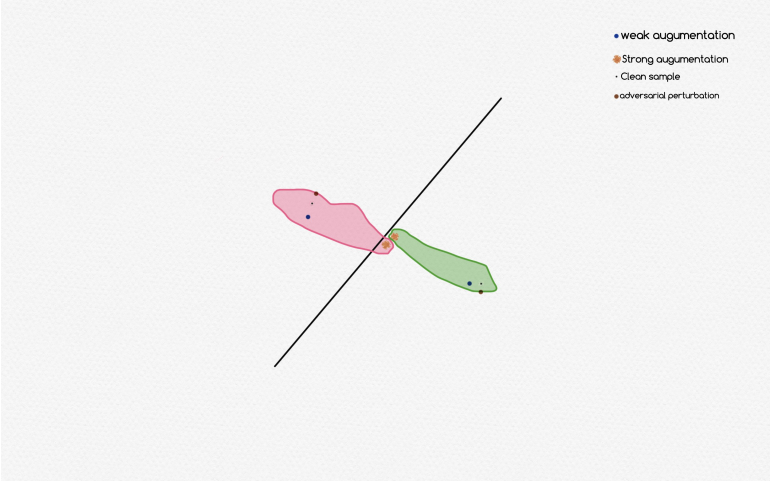

Figure 1: In the figure left side of the boundary line given belongs to one class and the other side belongs to another class. As there is no limit for strong augmentation it can jump into manifold of different class in output domain as shown in the above image and can cause ambiguity.

3 Results and Discussion

We implemented different algorithms for semi-supervised image classification with varying amounts of labeled data on CIFAR10/100. The results are reported in the table 3.

3.1 Pseudo-Labeling

As shown in the table 3 we have reported the error rates on two different datasets for three set of different threshold values. Generally the principle of choosing threshold values is to balance quality

Dataset	CIFAR-10		CIFAR-100	
Label Amount	250	4000	2500	10000
Pseudo-labeling with 0.95(Threshold)	63.7	**24.5**	**71.1**	**52.2**
Pseudo-labeling with 0.75(Threshold)	**60**	29	73.4	54.8
Pseudo-labeling with 0.6(Threshold)	60.1	30.4	76.9	57.9
Virtual Adversarial Training	**56.7**	**25.21**	**68.7**	**50.8**
Improvement on Fixmatch	**17.7**	**6.7**	**47.4**	**34.7**

Table 1: Error rates for CIFAR-10, CIFAR-100 datsets on different algorithms(Pseudo-labeling, Virtual Adversarial Training, Improvement on Fixmatch)

of label(high threshold value) suggesting that Pseudo-label is very sure to have a correct label and quantity of label(low threshold value) suggesting that Pseudo-label is not very sure of correct label, but can have many Pseudo-labels compared to the high threshold case. Empirically we found out that having a threshold of 0.95(high quality) works well for almost all cases showing that quality of a Pseudo-label plays a key role when compared to quantity, except for CIFAR-10 250 labels. Our intuition of this behaviour is because CIFAR-10 with only 250 labels which are very limited to do any kind of learning can take an advantage of having many Pseudo-labels(threshold of 0.75) even though those might be incorrect initially.

From the results, we can clearly see that for both the datasets if we have more labeled data for training we get less error rate, but the difference is not huge in CIFAR-100 dataset because we have provided less number of labeled data per class(only 100 per class where as for CIFAR-10 it is 400). Another interesting observation is that though we have provided same number of labels per class(25 per class) for both CIFAR-10 and CIFAR-100 we get a difference in error rate. Our intuition for this difference is because since CIFAR-100 has many output classes when compared to CIFAR-10 there might be lot of overlap of features between different classes. And this requires a lot of examples in the form of labeled data to create distinction between output classes.

The main caveat of this algorithm is it does not use any self-supervised training losses during its training process, which are often proved to help learn unlabeled data features better. A complete list of hyperparameters is reported in appendix 2 3.

3.2 Virtual Adversarial Training

The main idea of Virtual adversarial training algorithm is robustness of the model on unlabeled data using adversarial samples. Empirically we see that more number of labeled data is good for getting low error rate. This algorithm performs better compared to Pseudo-labeling algorithm when the number of labeled data available is less. Our intuition for this behaviour is because this algorithm considers smoothing in output distribution, with respect to the input for the unlabeled data. Whereas, Pseudo-labeling algorithm does not do any learning on the unlabeled data, it merely uses model predictions to generate Pseudo-labels which is mainly characterized by models ability(using available labeled data) to produce good results.

When considering loss functions used for learning in this algorithm there are totally two different things being done. One with cross-entropy loss using labeled data trying to train the model for classification task directly. Whereas, with KL-divergence the aim is different just making the output probabilities same for clean and adversarial unlabeled data. With this differentiation the last layers of the model struggle to have robust features suitable for classification task. This intuition was formed on the basis of the idea that the last layers of the model optimizes themselves to the loss functions and regularizations used during training [Kornblith et al., 2020].

We can observe in the figure 5 that we are taking two samples of same image, one clean(left) and another one with adversarial perturbation(any one image from right) and making the model to predict

same output probabilities for both of those samples. Few other examples of adversarial images generated can be found at 6 7 8. A complete list of hyperparameters are reported in appendix 4.

3.3 Improvement on Fixmatch

In Fixmatch Algorithm there are two important types of learning involved. The important point to note here is they are done in a parallel manner on the data . One is supervised learning and another one is self-supervised learning. In supervised learning we directly use labeled data and generate cross entropy loss. In self-supervised learning we try to create two representations of same sample and calculate cross-entropy loss between one of the outputs and Pseudo-labels generated from other. The characteristics of representations is very important in this process. One of the representation must be strong augmentation. If it is not done properly for example, color is not considered in transformations the model can easily learn the underlying histogram representation, which defeats the purpose of strong augmentation. One of the key transformations in the fixmatch algorithm to be considered is RandAugment [Cubuk et al., 2019]. This is the reason we chose to keep the strong augmentation in our improvement. We also evaluated by removing strong augmentation replacing it with VAT perturbation which leads to increase in error rate(from 6.7 to 16.22 for CIFAR10 4000 labels).

The other important augmentation is weak augmentation which is close to clean sample in the input domain. In the same way VAT gives nearest sample to the clean sample which has same class sample but crossing that boundary changes the class, like optimal direction. Because of this reason we added this in the fixmatch setup to generate additional Pseudo-labels. In our results 9 we can see that the similarity check we are doing is only working for initial epochs and helping the model learn without ambiguity. But this effect does not persist after some epochs. By this we can say that the intuition we developed about similarity does not hold. The initial peak in the similarity check maybe only due to model's natural behaviour during initial steps of training , because of this we can even work without similarity check. This behaviour is more profound(similarity check does not contribute) on CIFAR-100 dataset mainly because of many output classes the probability of having same distribution as output decreases rapidly. This behaviour can be further evaluated on different threshold levels for L1 distance(in our experiment we only evaluated on 0.3 threshold for L1 distance). The overall unlabeled loss will peak at starting 30 epochs then starts to diminish as features from unlabeled data is already learnt. A complete list of hyperparameters is reported in appendix 5. As it is taking more time to train we stopped at 100 epochs for every model even though there is still a trend of increase in accuracy.

4 Conclusion

In this project we learnt mainly about scenarios where labeled data is limited and methods to overcome it. It involves semi-supervised learning. We started with naive algorithm of Pseudo-labeling and got decent results then we tried using VAT loss to make the model robust against new data. We found that using adversarial training on labeled data makes the model robust against adversarial samples but doing the same on unlabeled data generalizes the model to get less error rate on unseen data. Finally, from the last task we learnt how impactful is the augmentation of data in unlabeled data training and experimented with different settings which we thought can improve upon fixmatch algorithm.

References

T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. URL `https://arxiv.org/abs/2002.05709`.

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719, 2019. URL `http://arxiv.org/abs/1909.13719`.

S. Kornblith, H. Lee, T. Chen, and M. Norouzi. What's in a loss function for image classification? *CoRR*, abs/2010.16402, 2020. URL `https://arxiv.org/abs/2010.16402`.

D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

E. Lobacheva, M. Kodryan, N. Chirkova, A. Malinin, and D. P. Vetrov. On the periodic behavior of neural network training with batch normalization and weight decay. *CoRR*, abs/2106.15739, 2021. URL `https://arxiv.org/abs/2106.15739`.

I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. URL `http://arxiv.org/abs/1608.03983`.

T. Miyato, S. ichi Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning, 2018.

K. Sohn, D. Berthelot, C. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *CoRR*, abs/2001.07685, 2020. URL `https://arxiv.org/abs/2001.07685`.

A. Tarvainen and H. Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780, 2017. URL `http://arxiv.org/abs/1703.01780`.

B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *CoRR*, abs/2110.08263, 2021. URL `https://arxiv.org/abs/2110.08263`.

A Miscellaneous and Results

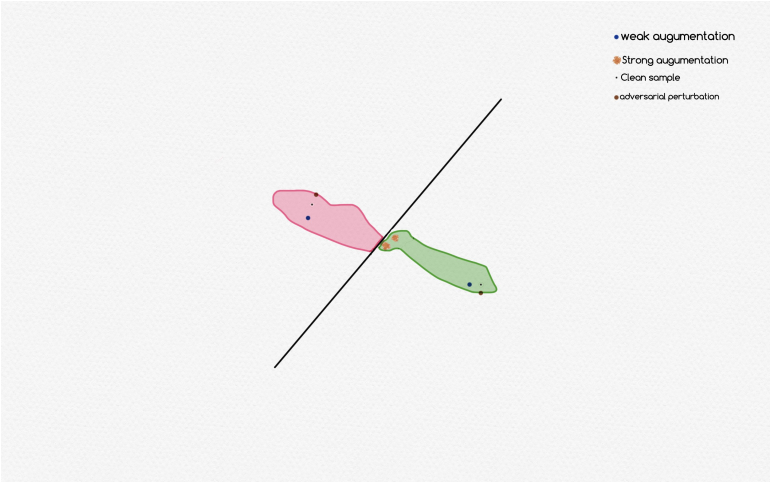

Figure 2: In the figure left side of the boundary line given belongs to one class and the other side belongs to another class.To overcome the jump in the manifold for the strong augmented image we calculated similarity(L1) distance between strong augmentations of different classes and make the strong augmentations behave similarly when the L1 distance calculated is less(0.3 in our case).

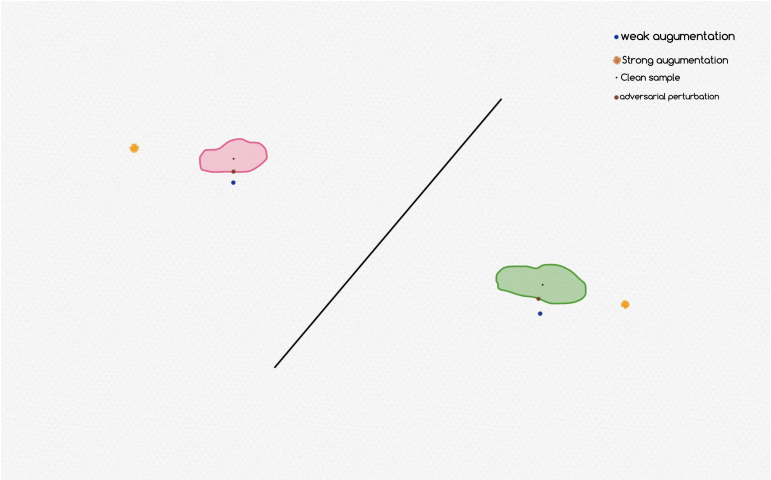

Figure 3: In the figure left side of the boundary line given belongs to one class and the other side belongs to another class.For initial epochs as we can see in the above figure weak augmented image would not give correct Pseudo-labels where as the adversarial image has correct label even in initial epochs, by applying loss between strong and adversarial image, in this case learning can be done even for initial epochs for unlabeled data

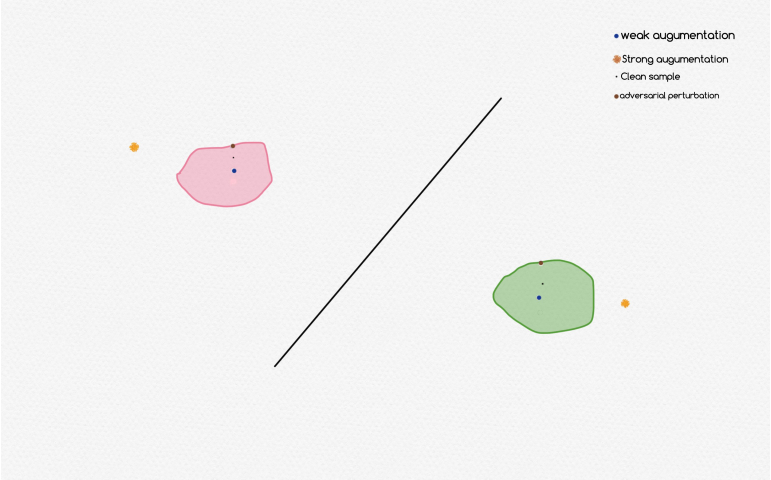

Figure 4: In the figure left side of the boundary line given belongs to one class and the other side belongs to another class.For initial epochs as we can see in the above figure weak augmented image would not give correct Pseudo-labels where as the adversarial image has correct label even in initial epochs, by applying loss between strong and adversarial image, in this case learning can be done even for initial epochs for unlabeled data

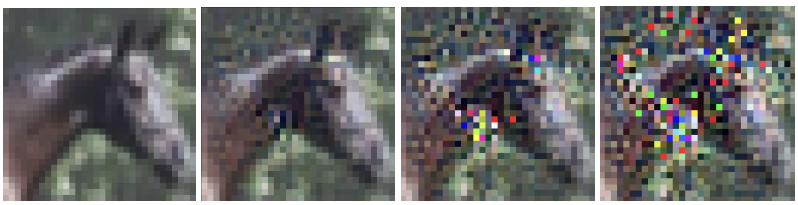

Figure 5: Original image(To the left) and VAT generated images with increasing epsilon values of 2.0, 5.0 and 8.0 respectively(To the right) for CIFAR-10 with 4000 labels

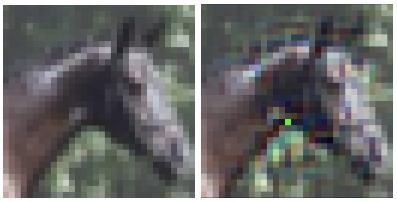

Figure 6: Original image(To the left) and VAT generated image with epsilon value of 2.0 (To the right) for CIFAR-10 with 250 labels

Figure 7: Original image(To the left) and VAT generated image with epsilon value of 5.0 (To the right) for CIFAR-100 with 2500 labels

Figure 8: Original image(To the left) and VAT generated image with epsilon value of 5.0 (To the right) for CIFAR-100 with 10000 labels

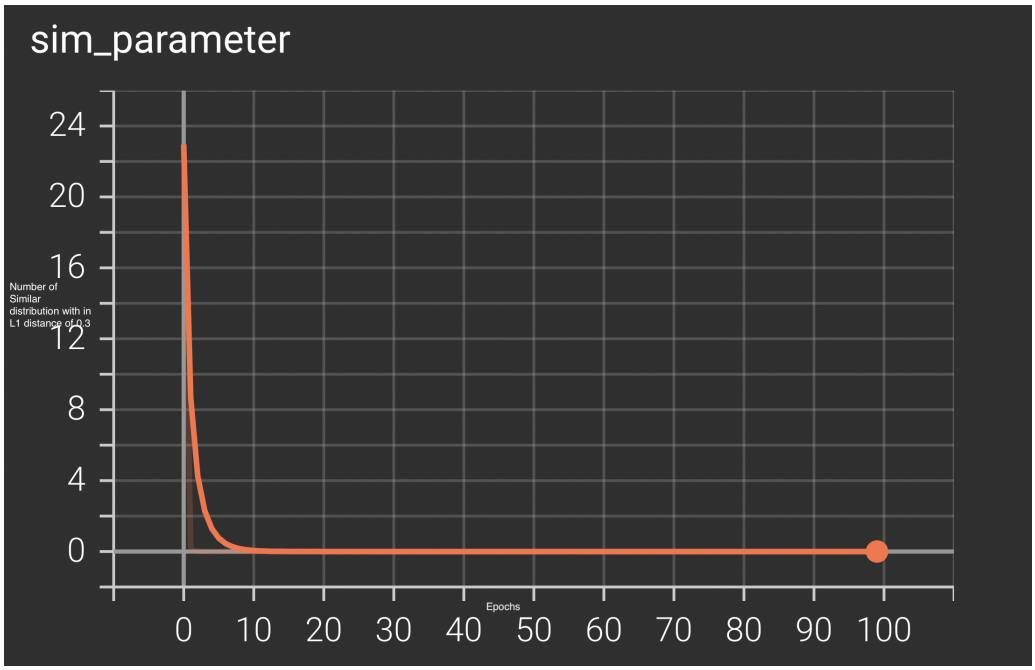

Figure 9: Number of similar distributions with L1 less than or equal to 0.3 tends to decrease as epochs increase

Dataset Size	Threshold	Learning rate	Model depth	Epochs
250	0.95	0.3	28	45
250	0.75	0.1	28	45
250	0.6	0.1	34	45
4000	0.95	0.1	34	30
4000	0.75	0.1	34	30
4000	0.6	0.1	34	30

Table 2: Hyperparameter value which got best results for CIFAR-10 dataset uisng Pseudo-labeling algorithm (32 training batch size suits for all the mentioned combinations instead of 64, this is because smaller batch size can act as regularization due to noise present in every batch. It is necessary because of size of the training data set to prevent overfitting.)

Dataset Size	Threshold	Learning rate	Model depth	Epochs
2500	0.95	0.3	28	45
2500	0.75	0.3	28	45
2500	0.6	0.3	28	45
10000	0.95	0.1	34	45
10000	0.75	0.1	28	45
10000	0.6	0.1	34	45

Table 3: Hyperparameter value which got best results for CIFAR-100 dataset using Pseudo-labeling algorithm (32 training batch size suits for all the mentioned combinations instead of 64, this is because smaller batch size can act as regularization due to noise present in every batch. It is necessary because of size of the training data set to prevent overfitting.)

Dataset	Dataset Size	Epsilon	Learning rate	Model depth
CIFAR-10	250	2.0	0.001	34
CIFAR-10	4000	2.0	0.001	28
CIFAR-100	2500	5.0	0.001	34
CIFAR-100	10000	5.0	0.001	28

Table 4: Hyperparameter value which got best results for CIFAR-100 and CIFAR-10 dataset using Virtual Adversarial Training algorithm (32 training batch size and 50 epochs each epoch with 1600 iterations suits for all the mentioned combinations)

Epsilon	Learning rate	Model depth	Model width	Batch Size	Epochs	Pseudo-label threshold
2.0	0.03	28	2	64	100	0.95

Table 5: Hyperparameter value which got best results for CIFAR-100 and CIFAR-10 dataset(2500, 10000 and 250, 4000 labels respetively) using Improvement in fixmatch algorithm (We considered Vat iteration as 1 , vat xi parameter as 0.5 and coefficient of unlabeled loss(lambda) as 1.)