Exploring Preferential Label Smoothing for Neural Network based Classifiers

Paritosh Goyal[†◇*], Chenyang Huang[†◇], Amine Trabelsi[‡], Osmar R. Zaiane[†◇]
[†] University of Alberta
[‡] Université de Sherbrooke
[◇] Alberta Machine Intelligence Institute

Abstract

Overfitting, a common problem in Machine Learning, occurs when a predictive model learns the noise in the training data instead of the true underlying patterns and converges to perform very well with the training data but poorly with unseen data. Models that overfit cannot be deployed in practice. Regularization is a technique typically used to help a model better generalize. This is usually achieved by adding a penalty term in the loss function to discourage the model from fitting noise, making it more robust to noise in the data and, therefore more generalizable. One method of regularization is to take some of the concentration (called Smoothing Ratio (SR)) from the data sample's ground truth label and distribute it uniformly among all the other labels during training. This method is called label smoothing and is a simple yet effective method to improve generalization. In this work, we explore what happens if we distribute the SR to the non-ground truth labels based on how closely they are related to the ground truth label, instead of uniformly. We call this approach of distributing the SR based on relation between labels as Preferential Label Smoothing (PLS). PLS represents a more unified approach of performing label smoothing. Ordinary uniform label smoothing becomes pointless as the number of labels becomes large since the SR proportion distributed per label becomes negligible. PLS is inconsequential in the case of binary classification, since there are only two labels. Therefore, we investigate the effects of PLS when the number of labels in the dataset is high. We also examine the effects of uniform and preferential label smoothing, as well as the absence of label smoothing, on the training dynamics. We demonstrate our study on image classification and text classification.

Keywords: Overfitting, Label Smoothing, Regularization, Model Generalization

1. Introduction

In this work, we address the fundamental problem of overfitting. An overfitted machine learning model does not generalize well to unseen data. A useful machine learning model must be able to generalize well. There are many methods to prevent overfitting and therefore help models generalize better and eventually improve a model's performance in the task it is intended to do. Szegedy et al. [1] proposed one such mechanism for preventing overfitting and overconfidence in Neural Network (**NN**) - based models called Label Smoothing. The idea is to use soft labels instead of hard labels (one-hot encoding) while training a model, i.e. distributing some concentration from the ground truth label to all the labels uniformly rather than having full concentration on one ground truth label and no concentration on non-ground truth labels. We refer to this method as **Uniform Label Smoothing (ULS)**.

As an illustrative example, if we are learning to classify images of animals into cats, dogs, mice, chicken and fish, when presented with an image of a cat during learning, we would say it is for instance 99% a cat and distribute the remaining 1% (called smoothing ratio) uniformly among the other classes – ,i.e., 0.25% for each of dog, mouse, chicken and fish.

ULS [1] has a shortcoming that we illustrate through an example of emotion classification for text in Figure 1. Given a sentence, the goal is to determine which emotion class the sentence belongs to. In Figure 1, the ground truth vector **A** is a one-hot encoded vector

[*]paritosh@ualberta.ca, chenyangh@ualberta.ca, amine.trabelsi@usherbrooke.ca, zaiane@ualberta.ca

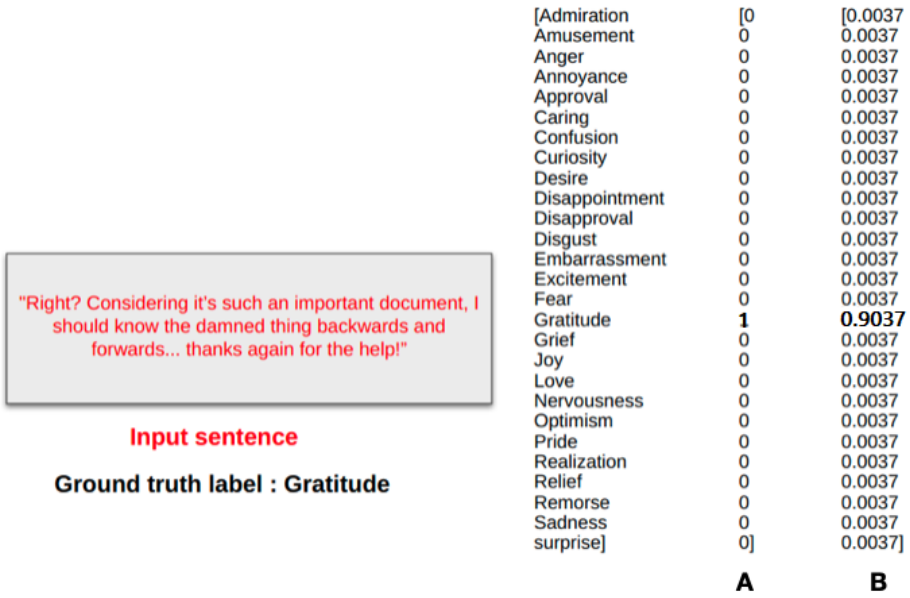


Figure 1. An example of emotion classification from a given sentence. There is an input sentence with ground truth label *Gratitude* There are a total of 27 plausible classes of emotion. **A** shows the one-hot encoded vector of the ground truth. **B** shows the ground truth vector after label smoothing as suggested by Szegedy et al. [1]

.

while the embedded vector **B** used for training is the result of the label smoothing operation proposed by Szegedy et al. [1]. Although this type of arrangement would help in preventing overfitting, it unnecessarily gives equal importance to some classes which do not have any relationship with the true label. The same is true for the example of the images above. While a cat is relatively close to a dog and therefore sharing a small part of its label is not surprising, sharing a portion of the cat label with a fish and therefore conceding resemblance could be alarming. In Figure 1, if the true label of a sentence is gratitude, then it is more likely for the sentence to express closer emotions such as joy or excitement than farther ones like disgust or remorse.

The label smoothing approach suggested by Szegedy et al. [1] appears to indiscriminately (equally) distribute the label concentration among all the non-ground truth labels. From the example in Figure 1, it is intuitive that an approach where the label concentration is distributed based on relationships between the ground truth label and the non-ground truth label might be more appropriate. We propose the idea of distributing concentration to non-ground truth labels based on how close or far in relationship the non-ground truth labels are from the ground truth label. We call this approach **Preferential Label Smoothing (PLS)**.

To the best of our knowledge, prior work does not focus on text classification. Our experiments involve both text classification and image classification. We also bridge the gap in label smoothing research, where the effects of label smoothing on the training dynamics of the NN are missing. This work seeks to answer the following questions in the context of multi-class classification (more than two classes) problems:

(1) **Does PLS help improve model performance?** ULS has helped in improving the performance of NN for image classification [1]. Hence, we verify whether PLS helps improve the performance of image and text classification.

(2) **Does ULS or PLS affect the training dynamics of the NN?** There is no prior work addressing this question. We use two approaches to study this - (*i*) effect of changing learning rate with label smoothing on the generalization error, (*ii*) length of gradients while training with different label smoothing approaches.
(3) **How does PLS affect model performance when we change the number of labels (classes) in the dataset?** The idea is to identify whether PLS (or ULS) is good for datasets which have a large number of labels or a smaller number of labels.

2. Background and Related Work

2.1. Uniform Label Smoothing

The idea behind uniform label smoothing is to modify the one-hot encoded vector of target outputs and use a smaller concentration on the true class label. The remaining (or taken) concentration from the true label is distributed to all the class labels uniformly. This proportion of the true label distributed among the other classes is called the Smoothing Ratio (SR).

Let ϵ be the SR. For a training data sample $\mathbf{x}^{(i)}$ with the ground-truth label t, let k be a label among K possible labels, the concentration distribution over the ground truth vector in the context of no label smoothing (NoLS) is $q^{NoLS}(k \mid \mathbf{x}^{(i)}) = \delta_{k,t}$, where $\delta_{k,t} = 1$ when $k = t$, 0 otherwise. Concentration distribution over the ground truth vector with ULS is noted $q^{ULS}(k \mid \mathbf{x}^{(i)})$ (see Equation 2.1) and is a mixture of the original ground-truth distribution $q(k \mid \mathbf{x}^{(i)})$ and a fixed distribution $u(k)$ with ratios $1 - \epsilon$ and ϵ, respectively. ϵ decides the level of smoothing and falls between 0 and 1, usually it is a small value of the order < 0.2. Szegedy et al. [1] proposed to use $u(k)$ as a uniform distribution on all the labels, so $u(k) = 1/K$, which gives us Equation 2.2.

$$q^{ULS}(k \mid \mathbf{x}^{(i)}) = (1 - \epsilon)\delta_{k,t} + \epsilon u(k) \tag{2.1}$$

$$q^{ULS}(k \mid \mathbf{x}^{(i)}) = (1 - \epsilon)\delta_{k,t} + \frac{\epsilon}{K} \tag{2.2}$$

Computing the cross-entropy (CE) for $q^{ULS}(k \mid \mathbf{x}^{(i)})$ with the predictions $p(k \mid \mathbf{x}^{(i)})$ gives:

$$\begin{aligned}\mathcal{L}_{CE}(q^{ULS}(k \mid \mathbf{x}^{(i)}), p(k \mid \mathbf{x}^{(i)})) = (1 - \epsilon)\mathcal{L}_{CE}(q^{NoLS}(k \mid \mathbf{x}^{(i)}), p(k \mid \mathbf{x}^{(i)})) \\ + \epsilon\mathcal{L}_{CE}(u(k), p(k \mid \mathbf{x}^{(i)}))\end{aligned} \tag{2.3}$$

It is clear that we are replacing a single CE, $\mathcal{L}_{CE}(q^{NoLS}, p)$, with a pair of losses $\mathcal{L}_{CE}(q^{NoLS}, p)$ and $\mathcal{L}_{CE}(u, p)$. The second loss punishes deviation of prediction distribution p from the prior $u(k)$, with a relative ratio of $\frac{\epsilon}{1-\epsilon}$. We refer to this change in concentration distribution over the ground truth label vector as Label Smoothing Regularization (**LSR**).

2.2. Related work

Since label smoothing helps in improving generalization of deep learning models, it has become a common practice. Many architectures use label smoothing for various tasks like classification, speech recognition and machine translation. For machine translation, highly cited architectures include the SEQ2SEQ [2] and the transformer model [3]. Label smoothing was used with a transformer [3] which helped improve its BLEU score, all of which supports usage of label smoothing.

Research in label smoothing is in its relatively initial stages. Szegedy et al. [1] introduced LSR in 2015 as a technique to improve image classification with the inception architecture. Not much attention was put into LSR until Müller et al. [4] discussed the effects of label smoothing in the domains of image classification and machine translation by studying how the representations differ between the penultimate layer of the networks with and without

label smoothing. The visualization technique to visualize penultimate layers of the output makes it clear that label smoothing encouraged representations in the penultimate layer to group labels in tight, equally distant clusters. Müller et al. [4] also show that label smoothing helps in improving model calibration for both machine translation and image classification tasks; label smoothing shows effects similar to temperature scaling [5]. Additionally, Müller et al. [4] show that LSR weakens knowledge distillation in distilled models [6]. Knowledge distillation consists of compressing a complex "teacher" neural network into a smaller and faster "student" network by retaining its knowledge. Although LSR improves the accuracy of the teacher network, teachers trained with label smoothing produce less useful student networks compared to teachers trained with hard targets. Müller et al. [4] show that less information passes from teachers (trained with label smoothing) to students by comparing mutual information between the input and output of the two models. Müller et al. [4] spurred and encouraged more work and discussions around label smoothing.

Prior related approaches other than Szegedy et al. [1] are instance-based approaches of label smoothing ([7–9]). In these approaches, even for the data points belonging to the same class label, the way label concentration is distributed changes depending on the data point. All of these previous methods can be unified under the term PLS (Preferential Label Smoothing) even though they were not referenced as such.

3. **Preferential Label Smoothing (PLS)**

The idea of PLS is to distribute the SR on non-ground truth labels based on the relationship of the non-ground truth labels with the ground truth label. This relationship can be learned from some external data or provided by an expert in the area.

We define the concentration distribution over a ground truth label vector when using PLS as in Equation 3.1. Let θ be an oracle function represented as a matrix, which contains normalized values denoting information on the relationship between the ground truth label and the non-ground truth labels. θ can be constructed based on the information given by a subject matter expert or from some external data based on relationships between the labels.

$$q^{PLS}(k \mid \mathbf{x}^{(i)}) = (1-\epsilon)\delta_{k,t} + \epsilon\theta(k) \tag{3.1}$$

Computing CE for $q^{PLS}(k \mid \mathbf{x}^{(i)})$ with the predictions $p(k \mid \mathbf{x}^{(i)})$, for $k = 1 \dots K$, gives:

$$\begin{aligned}\mathcal{L}_{CE}(q^{PLS}(k \mid \mathbf{x}^{(i)}), p(k \mid \mathbf{x}^{(i)})) = (1-\epsilon)\mathcal{L}_{CE}(q^{NoLS}(k \mid \mathbf{x}^{(i)}), p(k \mid \mathbf{x}^{(i)})) \\ +\epsilon\mathcal{L}_{CE}(\theta(k), p(k \mid \mathbf{x}^{(i)}))\end{aligned} \tag{3.2}$$

Equation (3.2) again suggests that LSR in this case is similar to replacing a single CE $\mathcal{L}_{CE}(q^{NoLS}, p)$ with a pair of losses $\mathcal{L}_{CE}(q^{NoLS}, p)$ and $\mathcal{L}_{CE}(\theta, p)$. The second loss punishes deviation of prediction distribution p from the prior relationship function $\theta(k)$, with a relative ratio of $\frac{\epsilon}{1-\epsilon}$.

4. **Objectives of the Study**

We study effects of NoLS (i.e., No Label Smoothing), ULS (i.e., Uniform Label Smoothing), and PLS (i.e., Preferencial Label Smoothing) on the classification problem for both image and text data. To the best of our knowledge classification with text data was not studied in the past.

4.1. **Label Smoothing and Gradients**

To measure an algorithm's generalization performance: we compute the absolute value of the difference between the test error and the training error, corresponding to the generalization error. Models trained with NoLS might be overconfident because the largest logit tends to become much larger than all other logits. This means theoretically, while training with NoLS the model should go through a trajectory of higher gradients so that finally, the logit attains a higher value. Using LSR stops the model from becoming overconfident, which means theoretically the model should go through a trajectory of smaller gradients.

The relationship between generalization and the length of gradients while training a NN with SGD is given by Hardt et al. [10], according to whom, for a NN trained with SGD, the generalization error is bounded by the square of the gradients and the time taken to train.

Since NoLS does not generalize well, the generalization bound should be higher for NoLS as compared to ULS and PLS. The training trajectory should go through the region of higher gradients so that the generalization bound attains a higher value. This motivates us to run a gradient-based analysis of all LSR schemes against NoLS to find out whether this intuition is correct empirically.

4.2. **Generalization and Learning Rate**

The effects of LSR include benefits to the generalization error and improved accuracy. The generalization effects of LSR are not studied with different learning rates. This is important to study to check whether the generalization effects are as prominent at smaller rates as they are at larger rates. Studying this will help us answer whether we can learn faster with NoLS, ULS or PLS.

4.3. **Effect of label smoothing on increasing and decreasing the number of classes**

We propose PLS because we expect that using PLS would help improve classification performance since we can distribute the concentration based on relationships between the labels. If there were only two labels, then using PLS is not useful because there is one ground truth label and one non-ground truth label. As we increase the number of classes in the dataset, there are more non-ground truth labels and the labels far away from the ground truth get less concentration, while the labels close to the ground truth label get more concentration. Studying how this phenomenon affects the model performance would be interesting. It is interesting because the current ULS approach [4] shows that as the number of classes increases, the effect of label smoothing on model performance keeps decreasing to the point that with a very high number of labels, there is no effect (the case of Imagenet), whereas the PLS would be more interesting because it allocates more concentration to the related labels and less concentration to unrelated labels.

5. **Label Smoothing for Image Classification**

Image classification can be studied under the use case of single-label and multi-class classification or multi-label classification. For our experiments, we study image classification for the single-label and multi-class classification problems. The goal is to study the effects of label smoothing on model performance and training dynamics.

5.1. **Datasets**

We use CIFAR-10 and CIFAR-100 image datasets [11], which are standard datasets for image classification. CIFAR-10 has $60,000$ images and 10 classes. CIFAR-10 is a balanced dataset, and each class has $6,000$ images. It has $50,000$ training images and $10,000$ test

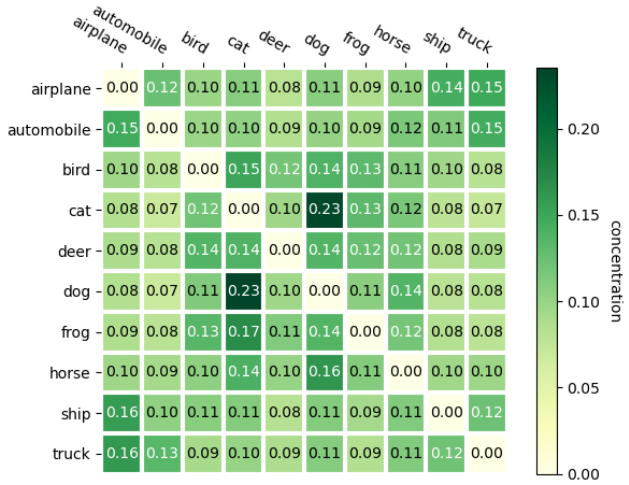


Figure 2. θ matrix for SEMLS for CIFAR-10. Diagonals are at 0 to imply that all the SR is distributed to the non-ground truth labels.

images (5000 training and 1000 test images per class). CIFAR-100 has $60,000$ images and 100 classes. CIFAR-100 is also a balanced dataset, and each class has 600 images. It has $50,000$ training images and $10,000$ test images, similar to CIFAR-10, but there are 500 training images and 100 test images per class. Classes in CIFAR-100 have groupings. The 100 classes in CIFAR-100 are grouped into 20 superclasses. Each image in the dataset has two labels - (i) a "fine" label for the class it belongs to, (ii) a "coarse" label for the superclass it belongs to. We use all the 100 fine labels.

5.2. **Approaches of PLS for image classification**

5.2.1. **Cluster Label Smoothing - CLS**

CLS is a special case of PLS where the preference is approximated by which group of superclass a label belongs to. For instance, for training CIFAR 100 dataset, if the ground truth label is 'clock' then we distribute SR among all the non-ground truth labels that come under superclass 'household electrical devices' uniformly (i.e., 'clock', 'computer keyboard', 'lamp', 'telephone', 'television'), and assign no concentration to the rest of the classes. This type of label smoothing represents a label smoothing suggested by the data expert, i.e., the curators of the CIFAR-100 dataset.

5.2.2. **Semantic Label Smoothing - SEMLS**

Semantic Label Smoothing is a special case of PLS where the preference is approximated by semantic similarity among the labels. We use the word vectors from GloVe embeddings [12] to get the vector representations of the labels in the CIFAR-10 and CIFAR-100 datasets. Using the vector representation, we compute the Euclidean distance between the labels. The distances between the labels represent how far away they are from each other. The inverse of the distances between the labels represents the similarity between the labels.

Figure 2 depicts the θ matrix (introduced in Section 3) for CIFAR-10 dataset. Each cell in the θ matrix contains the fraction of the SR concentration that should be assigned to the column label if the row is a ground truth label. These values are obtained after normalizing the similarity values between labels for each row (the diagonal cells are set to 0).

6. Experiments and Results for Image Classification

6.1. Experimental Setup

ResNet-18 and ResNet-34 models [13] are trained with CIFAR-10 and CIFAR-100 datasets. From the $50K$ images, $40K$ images (selected randomly) are used for training and the remaining $10K$ images are used for parameter tuning. We use CE loss and SGD for training. The output of the last layer of ResNet-18 and ResNet-34 is passed through a softmax layer before computing the CE loss. We use Nesterov momentum [14] $= 0.9$ as an optimizer along with SGD for optimization, and mini-batch size is kept 128. For ResNet-18 and ResNet-34 architectures, we use Kernel size $= 3 \times 3$, Stride $= 1$, average pooling is used for the pooling layer and the ReLu activation [15] function is used for the fully connected layers. We employ Batch Normalization [16] to prevent exploding gradients.

6.2. Impact of label smoothing on model performance

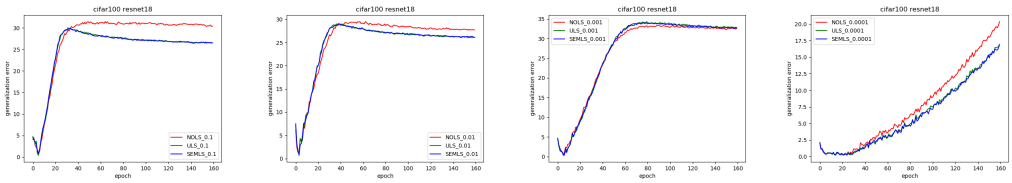
We use NoLS, ULS and PLS for the experiments here. For CIFAR-100, we use both CLS and SEMLS approaches, and just SEMLS for CIFAR-10. Each of the experiments is run five times at different random seeds. For each run of the experiment, validation loss is used as the criterion for early stopping [17]. We start training with a learning rate of 0.1 and decrease the learning rate twice - after 40 epochs and after 80 epochs, by a factor of 10 times. Weight decay used is 0.0005. We use accuracy as the performance measure following the approach of Müller et al. [4]. The results on the test set are presented in Table 1 for CIFAR-10 and CIFAR-100.

Dataset + Model	**NoLS**	**ULS**	**CLS**	**SEMLS**
CIFAR-10 + ResNet-18	93.809 ± 0.487	94.085 ± 0.273	na	**94.112 ± 0.568**
CIFAR-10 + ResNet-34	93.762 ± 0.578	93.557 ± 0.608	na	**93.793 ± 0.514**
CIFAR-100 + ResNet-18	72.789 ± 0.435	72.665 ± 0.323	72.618 ± 0.724	**72.798 ± 0.247**
CIFAR-100 + ResNet-34	71.839 ± 0.668	71.902 ± 0.787	**72.813 ± 0.689**	71.78 ± 0.912

Table 1. Top-1 classification accuracies (mean ± standard deviation for five runs) of CIFAR-10 and CIFAR-100 dataset with ResNet architectures trained with NoLS, ULS, CLS and SEMLS. The results here are in percentage.

The results in Table 1 suggest that when it comes to overall accuracy, CLS and SEMLS (both of which are special cases of PLS) are slightly better than NoLS and ULS for all of the four cases. However, we see that CLS in case of CIFAR-100 and ResNet-34 seems to do slightly better than SEMLS which means that the relationship chosen among the labels matters. Both CLS and SEMLS represent relationships from two different sources of information (CLS comes from knowledge of CIFAR-100 data curators, and SEMLS comes from the semantic similarity between the class labels). We see that there is a difference in the results when we change this relationship function, suggesting that if these relationships are chosen precisely and carefully, then model performance could be further improved. (For instance, in the case of CLS, right now we are distributing concentration uniformly among the labels that belong to the same superclass, i.e., for 'aquatic mammals' the label 'beaver' is close to the label 'dolphin', if a zoologist gives us a better relationship between these labels, then we can use that relationship for the CLS).

ULS is better than NoLS twice and worst twice. If we take into account the standard deviation, we can notice that the confidence of all label smoothings overlap with each other. The same observation is made in the results by Müller et al. [4]. In their work, there is an overlap in confidence intervals of models trained on CIFAR-10 and CIFAR-100 datasets. Although they use different models than ours, the observation is consistent with ours -



(a) LR = 0.1. *(b)* LR = 0.01. *(c)* LR = 0.001. *(d)* LR = 0.0001.

Figure 3. Generalization error for ResNet-18 trained on CIFAR-100 with different learning rates (LR).

overlapping of confidence intervals. Since the confidence intervals here are overlapping, we expect the overlap to happen in the generalization error too.

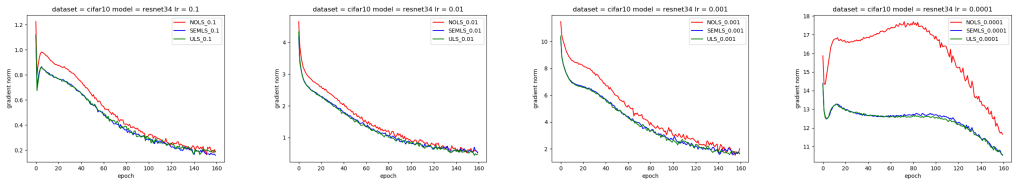
6.3. **Training Dynamics**

We use NoLS, ULS and SEMLS for the experiments here. The goal is to study how label smoothing affects the training dynamics of a classifier. Since weight decay brings extra regularization to the models, we keep our models free of weight decay so that we can isolate the effect of label smoothing. Getting the best accuracy is not the goal of these experiments rather the goal is to study the training dynamics such as - (i) interdependence of gradient norms, learning rate and the type of smoothing while training, (ii) interdependence of generalization error, learning rate and type of smoothing while training.

We keep SR = 0.1. We use four different learning rates of $[0.1, 0.01, 0.001, 0.0001]$ to train the classifiers and keep the learning rate constant throughout one training run. This helps us in analysing how changing the learning rate may affect the gradient norm and generalization error with different label smoothing approaches. We train for up to 250 epochs and observed while running experiments that the training loss and test loss changed significantly for all the learning rates (except 0.0001) within the first 100 epochs and did not change after 150 epochs, therefore we present our plots for up to 160 epochs so that the difference between the plots of ULS and SEMLS is visible. All the results depicted in the forthcoming plots are averaged over five runs.

6.3.1. **Generalization and learning rate**

Inferences from generalization plots From the plots in Figure 3, depicting the generalization error for ResNet-18 trained on CIFAR-100 with different learning rates, it is evident that the generalization error saturates after about 100-120 epochs except for the plots for the learning rate of 0.0001, indicating that 0.0001 is not reaching the saturation state and so it is not a good learning rate to train and there is no benefit of adding any label smoothing when using such a small learning rate. At learning rates of $[0.1, 0.01, 0.001]$, from the plots, it is clear that the generalization curves are overlapping for different label smoothing methods. Similar patterns are observed when using ResNet-34 or training with CIFAR-10. Plots are not reported here for lack of space. For CIFAR-100, ULS and SEMLS perform better than NoLS in terms of generalization error but they also overlap very strongly. Overlapping of generalization error is an expected behaviour since the accuracy measures are also overlapping. This suggests that both ULS and SEMLS are more advantaged with CIFAR-100 when the learning rate is high, this can be due to the large number of classes in CIFAR-100.



(a) LR = 0.1. *(b)* LR = 0.01. *(c)* LR = 0.001. *(d)* LR = 0.0001.

Figure 4. Gradient norms obtained while training ResNet-34 on Cifar-10 dataset across three smoothing at four different learning rates (LR).

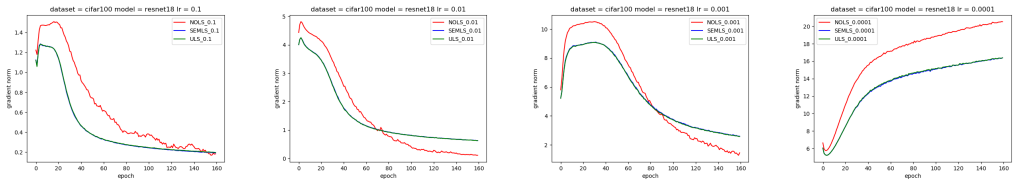
6.3.2. **Gradient Norm**

The work of [10] suggests that generalization error is directly proportional to the square of Lipschitz constant, which is proportional to the gradient norm of the weights of the network. Hence, we compare gradient norms as the training progresses for the different label smoothing approaches to assess their generalizability. The plots for gradient norm are presented in Figures 4 and 5. We report a few observations on expriments using ResNet-18 and ResNet-34 trained on both CIFAR datasets in the following.

(1) Gradient norms of models trained on CIFAR-10 (Figure 4) with ULS and SEMLS are smaller than the gradient norms of models trained with NoLS, which explains why the models trained using NoLS have slightly more tendency to overfit. Except at the end of the training phase, the norms of the models trained with NoLS are closer to the ones trained with ULS and SEMLS. The reason is - the training and test loss do not change by that point. (2) Gradient norms of models trained on CIFAR-100 (Figure 5) with ULS and SEMLS are smaller than those with NoLS for the beginning part of the training but the situation changes around 80 epochs (except for learning rate = 0.1). (3) For the learning rate of 0.1 in all cases, the gradient norm for models trained with NoLS is larger than those trained with ULS and SEMLS which might be due to the fact that with a higher learning rate, the optimizer can move away from the region of non-optimality more quickly, and so the consistency of the gradient norms is maintained through the training process. (4) Gradient norms when using ULS and SEMLS on CIFAR-100 are very close to each other, there is a very fine gap between the blue curve and green curve whereas the gap between the blue curve and green curve is larger in the case of CIFAR-10. This may be due to the fact that there are 100 labels in CIFAR-100, when distributing SR among the non-ground truth labels in CIFAR-100, each gets a smaller label concentration such that ULS and SEMLS end up assigning the same concentration to the non-ground truth labels. (5) The Gradient norm for all label smoothings, models and datasets increases as the learning rate is decreased since, by decreasing the learning rate, the optimizer moves more slowly to the region of optimality. (6) When the learning rate is 0.0001, no learning is happening and the gradients behave very differently than with other learning rates. This is because the learning rate of 0.0001 is too small and the optimizer is not able to proceed toward a region where it can actually help reduce the training and testing loss.

7. **Experiments and results for Text Classification**

We use three emotion datasets to show effects of label smoothing on text classification, the datasets that we use are single labelled and described subsequently. **TEC - Twitter Emotion Corpus** [18] has $21,051$ instances and it is an unbalanced dataset with six emotions distributed as follows - joy: 39%, sadness: 18%, surprise: 18%, fear: 13%, anger: 7%,



(a) LR = 0.1. *(b)* LR = 0.01. *(c)* LR = 0.001. *(d)* LR = 0.0001.

Figure 5. Gradient norms obtained while training ResNet-18 on Cifar-100 dataset across three smoothing at four different learning rates (LR).

disgust: 4%. **CBET - Cleaned Balanced Emotional Tweets** [19, 20] has 76,860 instances and it is a balanced dataset with nine emotions as follows - anger, surprise, joy, love, sadness, fear, disgust, guilt, thankfulness. **ISEAR - International Survey on Emotion Antecedents and Reactions** [21] has 7,666 instances and it is a balanced dataset with seven emotions as follows - joy, fear, anger, sadness, disgust, shame, guilt. TEC and CBET were formed using tweets from twitter while ISEAR was human-annotated.

For evaluating the PLS model performance, the LSTM [22] model is trained. We use 5-fold cross-validation, with the train-test split ratio of 4:1. For splitting data, a stratified split is used. The results for experiments on ISEAR, TEC and CBET with different label smoothing approaches are presented below in Table 2. The reported results are average and standard deviation. SEMLS has a slightly better performance for ISEAR (well-curated

Dataset No.	**NOLS**	**ULS**	**SEMLS**
ISEAR	58.02 ± 3.82	58.71 ± 3.13	**59.06 ± 3.51**
TEC	**52.12 ± 3.73**	51.84 ± 3.31	51.48 ± 3.41
CBET	56.01 ± 3.97	56.88 ± 3.62	**57.02 ± 3.23**

Table 2. Macro averaged F1 score (mean ± standard deviation) on emotion classification datasets trained with LSTM model with NOLS, ULS and SEMLS. The results here are in percentage.

dataset made by human annotations) and CBET (dataset which has more training samples than the other two). There can be an explanation for this - (*i*) ISEAR case - SEMLS might be capturing the relationship between emotion labels, while annotating the data it might be possible that humans are also considering the relationship of labels with the sentence they are annotating. (*ii*) CBET case - SEMLS might be working slightly better because this dataset is larger than the other two.

7.1. **Effect of Label Smoothing on Changing the Number of Classes**

For experiments related to the effect of changing the number of classes in a dataset, we take CBET dataset and vary the number of classes in the dataset, i.e., we sample sets of three datasets with 3, 5 and 7 classes from the CBET dataset. We averaged the results of three random samples for each representation of the number of classes. The results of this experiment are presented in Table 3. From the table, it is not certain which label smoothing has the best performance with the change in the number of classes in the dataset, but on average SEMLS achieves the best performance using CBET. Additional tests with a text dataset with a larger number of classes are necessary.

	NoLS	ULS	SEMLS
3 classes	57.54 ±3.23	58.00 ± 3.41	**58.14 ±3.21**
5 classes	57.14 ± 3.43	57.25 ± 3.50	**57.29 ±3.48**
7 classes	57.05 ± 3.29	**57.36 ±3.36**	57.29 ±3.28

Table 3. Macro averaged F1 score (mean ± standard deviation) on emotion classification datasets of 3 classes, 5 classes and 7 classes sampled from CBET, trained with LSTM with NOLS, ULS and SEMLS. The results here are in percentage.

8. **Conclusion**

In this work, we explored the concept of PLS (Preferential Label Smoothing) - assigning label concentration to non-ground truth labels based on their relationship with the ground truth label. We introduced two possible approaches for PLS: (1) CLS (Cluster Label Smoothing), where the preference is approximated by the group of superclass a label belongs to, such as superclasses in CIFAR-100. This type of label smoothing represents a label smoothing suggested by the expert of the field, i.e., the curators of the CIFAR-100 dataset; (2) SEMLS (Semantic Label Smoothing) which is based on the distance between a word embedding representation of the label words.

We evaluated the performance of classification models when trained with uniform (ULS) and No label smoothing (NoLS), as well as the two instances of PLS, i.e., CLS and SEMLS, on image and text. We found that SEMLS works slightly better on the CIFAR-10 image dataset and that the choice of the chosen preferential function makes a difference for the dataset with a larger set of labels, i.e., CIFAR-100. For Text Classification, we found that SEMLS is slightly better for datasets created using human knowledge (ISEAR) or containing relatively large samples (CBET).

We also studied the PLS model performance when the number of classes in the dataset is changed for text classification. We varied the number of classes in the CBET dataset and observed the macro-averaged F1-score. The results were inconclusive because the confidence intervals of the scores were overlapping.

To study training dynamics, we experimented with Image classification under minimal regularization (only label smoothing as the regularization). We examined generalization error and smoothing approaches at different learning rates. We found that at faster learning rates, the generalization error when using ULS or SEMLS is smaller or equal to when using NoLS. At slower learning rates, ULS and SEMLS have higher generalization errors than NoLS, or at very slow learning rates, training is not supported at all.

Additionally, we studied the gradient norms during the training phase of the network at different learning rates. We observed that gradient norms of SEMLS and ULS were smaller than those of NoLS when learning with a higher learning rate (0.1) and during the initial training phase. This empirically suggests that ULS and SEMLS should help reach a lower generalization bound. To the best of our knowledge, there is no prior empirical study to verify that label smoothing helps achieve a lower generalization bound.

When the number of classes in the dataset is large, gradient norm curves of ULS and SEMLS overlap significantly, and the same is observed for the generalization error, suggesting that SR gets distributed so much that SEMLS and ULS assign close to the same concentration to non-ground truth labels. This may be the case only for our approach of PLS, i.e., SEMLS, but there might exist an approach of PLS which is better than ULS. For instance, in emotion mining, a normalized co-occurence frequency of emotions in data can provide a relationship between emotion labels that can be used as a proxy for similarity. Another such example could be deriving PLS from the relationship among emotions given by an expert (e.g. a psychologist).

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the Inception Architecture for Computer Vision". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[2] I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to Sequence Learning with Neural Networks". In: *27th Intl. Conf. on Neural Information Processing Systems*. 2014.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. 2017.

[4] R. Müller, S. Kornblith, and G. Hinton. "When Does Label Smoothing Help?" In: *33rd International Conference on Neural Information Processing Systems*. 2019.

[5] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. "On Calibration of Modern Neural Networks". In: *34th International Conference on Machine Learning*. 2017.

[6] G. E. Hinton, O. Vinyals, and J. Dean. "Distilling the Knowledge in a Neural Network". In: *Deep Learning and Representation Learning Workshop, Neural Information Processing Systems* (2015).

[7] C.-B. Zhang, P.-T. Jiang, Q. Hou, Y. Wei, Q. Han, Z. Li, and M.-M. Cheng. "Delving Deep Into Label Smoothing". In: *IEEE Transactions on Image Processing* (2021).

[8] M. Maher and M. Kull. "Instance-based Label Smoothing For Better Calibrated Classification Networks". In: *20th IEEE International Conference on Machine Learning and Applications*. 2021.

[9] Z. Zhang and M. Sabuncu. "Self-Distillation as Instance-Specific Label Smoothing". In: *Advances in Neural Information Processing Systems*. 2020.

[10] M. Hardt, B. Recht, and Y. Singer. "Train Faster, Generalize Better: Stability of Stochastic Gradient Descent". In: *33rd International Conference on Machine Learning*. 2016.

[11] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. 2009.

[12] J. Pennington, R. Socher, and C. Manning. "Glove: Global vectors for word representation". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[14] G. Lan. "An Optimal Method for Stochastic Composite Optimization". In: *Mathematical Programming* (2012).

[15] K. Fukushima. "Cognitron: A self-organizing multilayered neural network". In: *Biological Cybernetics* (Sept. 1975).

[16] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *32nd International Conference on Machine Learning*. July 2015.

[17] L. Prechelt. "Early Stopping — But When?" In: *Neural Networks: Tricks of the Trade: Second Edition*. 2012.

[18] S. Mohammad. "#Emotional Tweets". In: **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, June 2012.

[19] A. G. Shahraki and O. R. Zaiane. "Lexical and Learning-based Emotion Mining from Text". In: International Conference on Computational Linguistics and Intelligent Text Processing, 2017.

[20] C. Huang. *Cleaned Balanced Emotional Tweets (CBET) Dataset*. https://github.com/chenyangh/CBET-dataset. 2019.

[21] K. R. Scherer and H. G. Wallbott. ""Evidence for universality and cultural variation of differential emotion response patterning": Correction." In: *Journal of Personality and Social Psychology* (1994).

[22] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* (Nov. 1997).