The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

From Label Smoothing to Label Relaxation

Julian Lienen, Eyke Hüllermeier
Heinz Nixdorf Institute and Department of Computer Science
Paderborn University
33098 Paderborn, Germany
{julian.lienen,eyke}@upb.de

Abstract

Regularization of (deep) learning models can be realized at the model, loss, or data level. As a technique somewhere in-between loss and data, label smoothing turns deterministic class labels into probability distributions, for example by uniformly distributing a certain part of the probability mass over all classes. A predictive model is then trained on these distributions as targets, using cross-entropy as loss function. While this method has shown improved performance compared to non-smoothed cross-entropy, we argue that the use of a smoothed though still precise probability distribution as a target can be questioned from a theoretical perspective. As an alternative, we propose a generalized technique called label relaxation, in which the target is a set of probabilities represented in terms of an upper probability distribution. This leads to a genuine relaxation of the target instead of a distortion, thereby reducing the risk of incorporating an undesirable bias in the learning process. Methodically, label relaxation leads to the minimization of a novel type of loss function, for which we propose a suitable closed-form expression for model optimization. The effectiveness of the approach is demonstrated in an empirical study on image data.

Introduction

In standard settings of supervised learning, the result of a learning process is essentially determined by the interplay of the model class, the learning algorithm resp. the loss function this algorithm seeks to minimize in order to identify the presumably optimal model, and the training data. Of utmost practical importance, especially for flexible models such as neural networks, is a regularization of the learner, so as to prevent it from overfitting the training data. In the case where models are non-deterministic and produce probabilistic predictions, for example class probabilities in the case of classification, overfitting also manifests itself in overly confident predictors with a tendency to assign probabilities close to the extremes of 0 or 1.

While the role of the model class and the loss function in helping to regularize the learner is quite obvious, this is arguably less true for the training data. The role of the data becomes especially important when supervision is only indirect in the sense that the target of the predictor is not directly observed. Again, class probabilities constitute an important example: Even if probabilistic predictions are sought, the data will normally not provide such probabilities as training information. Instead, it typically consists of examples with a single class label attached. In such cases, the formalization of the learning problem also involves the *modeling of the training data*.

This aspect, the modeling of data, is at the core of this paper. Compared to the model class and learning algorithm, it has received rather little attention in the literature so far, and is mostly done in an implicit way. For example, when training a model by optimizing losses such as log-loss or Brier score, an observed class label is implicitly treated as a degenerate (one-point) distribution, which assigns the entire probability mass to that label. Unsurprisingly, feeding the learner with extreme distributions of that kind aggravates the problems of overfitting and over-confidence.

So-called *label smoothing* (Szegedy et al. 2016) has recently been proposed to address these issues. The idea is to remove a certain amount of probability mass from the observed class and spread it across the other classes, thereby making the distribution less extreme. While probability mass can be spread in any way, the authors suggest a uniform distribution over all classes. This modification of the data encourages the model to be less confident about the predictions, which effectively narrows the gap between the logits of the observed class and the others. This method has proved successful in various applications, such as classification of image data using deep convolutional neural networks (Szegedy et al. 2016; Müller, Kornblith, and Hinton 2019).

Although label smoothing undoubtedly provokes a regularization effect (Lukasik et al. 2020), it can be questioned from a data modeling point of view. In particular, since the smoothed probability distribution is still unlikely to match the true underlying conditional class probability, it is likely to introduce a bias that may harm the generalization performance (Li, Dasarathy, and Berisha 2020). Indeed, while label smoothing helps to calibrate the degree of confidence of a model, and improves compared to the use of the conventional cross-entropy loss with one-point probabilities, explicit calibration methods such as temperature scaling (Guo et al. 2017) turn out to calibrate models even better (Müller, Kornblith, and Hinton 2019).

In this paper, we propose *label relaxation* as an alternative

approach to data modeling. To avoid a possibly undesirable bias, the key idea is to replace a degenerate probability distribution associated with an observed class label, not by a single smoothed distribution, but by a larger *set* of candidate distributions. All distributions in this set still assign the highest probability to the observed class, but the concrete degree is not fixed, and the remaining mass can be distributed freely over the other classes. This way, the learner itself can decide on the most appropriate distribution. In other words, instead of predetermining an alleged ground-truth distribution as a target, this distribution will be determined in a data-driven way as a result of the learning process itself.

To put label relaxation into practice, we devise a suitable generalization of the Kullback-Leibler (KL) divergence loss, which is able to compare a predicted probability distribution with a class of candidate distributions. The effectiveness of learning by minimizing this generalized loss is demonstrated on commonly used image datasets. While being competitive to label smoothing and other related regularization techniques in terms of classification performance, label relaxation does indeed improve in terms of calibration, i.e., accurate estimation of probabilities, often even compared to explicit calibration techniques that require extra data.

Related Work

As already said, different actions could be taken to improve learning, including the manipulation or modification of the original training data. In this regard, one can distinguish between methods acting on the instance, feature, and label level. While there are approaches to augment the instance set, e.g., by synthetically generating additional training examples (Cireşan et al. 2010; Krizhevsky, Sutskever, and Hinton 2012), or to introduce noise in the input features (e.g., (Vincent et al. 2008; van der Maaten et al. 2013)), our focus is on the adjustment of the labels.

One of the most prominent approaches of this kind, label smoothing (Szegedy et al. 2016), was already mentioned. It turns deterministic observations of class labels into probability distributions, assigning a predefined amount of probability mass to the non-observed classes; by default, a uniform distribution is used for that purpose. The newly generated targets are then used together with a conventional cross-entropy loss. As a result, the learner no longer tries to perfectly predict the original class (with probability 1), which may lead to drastic differences among the class logits and cause numerical instabilities. Label smoothing has been applied successfully not only to the domain of image classification, but also to other domains such as machine translation. More recently, Müller, Kornblith, and Hinton (2019) observed a calibration effect produced through label smoothing. Moreover, the authors analyzed the activation patterns in penultimate layers in neural networks. Apparently, label smoothing supports a regular distribution of the classes in these layers, in which clusters of instances associated with a class are well separated and tend to be equi-distant. Additionally, label smoothing has also turned out to be effective against label noise (Lukasik et al. 2020).

While label smoothing in its original form distributes probability mass to the non-observed classes uniformly, distributions other than uniform are of course conceivable. Although the work does not directly build upon label smoothing, (Hinton, Vinyals, and Dean 2015) follows a similar approach. Here, a teacher network predicts target probabilities which are then used for training a student network to *distill* the teacher's knowledge. In a different approach, a bootstrapping technique is proposed that makes use of the model's own distribution to adjust the training labels (Reed et al. 2015). Similarly, self-distillation approaches gathering the target labels from the model itself have shown regularizing effects to improve generalization performance (Zhang et al. 2019; Yun et al. 2020).

As an alternative approach to prevent the model becoming too overconfident and closely related to label smoothing, Pereyra et al. (2017) propose the penalization of confident distributions by adding the negative entropy of the predicted distribution to the original loss. Following this principle, Dubey et al. (2018) transfer the method to fine-grained classification. Related to this, with similar effects, the so-called focal loss (Lin et al. 2020) aims to reduce the loss for "well-classified" instances, i.e., predictions close to the actual target, by dynamically scaling cross-entropy loss. With this, designed to cope with class imbalance in object detection problems, the danger of overconfidence is reduced by flattening the loss near the true target and, thereby, shrinking the gradients for confident predictions.

In addition to the approaches outlined above, further ideas to adjust the given labels in order to achieve better generalization properties can be found in the literature. For instance, the approach by Xie et al. (2016) randomly flips targets with a fixed probability, resulting in training on a dataset ensemble with shared weights. As a result, an averaging effect lowers the risk of overfitting. Motivated by (Hinton, Vinyals, and Dean 2015), Li et al. (2017) propose a related distillery approach considering noisy side information. Bagherinezhad et al. (2018) describe a model that iteratively refines label probabilities from previous model predictions in a chain of multiple networks. By refining the labels over all models, data is augmented by soft targets to prevent overfitting.

In neural network learning, the predicted probabilities should ideally match the true distribution. However, as shown empirically by Guo et al. (2017), modern neural networks tend to be calibrated very poorly. While there exists a wide range of calibration methods, including isotonic regression (Zadrozny and Elkan 2002), Bayesian binning techniques (Naeini, Cooper, and Hauskrecht 2015), or beta calibration (Kull, Filho, and Flach 2017), a simple technique called temperature scaling proved to provide strong performance compared to its competitors (Guo et al. 2017). However, most calibration methods require additional data to determine the calibration parameters, being left with less data for training the model. Typically, this comes with a loss of generalization performance. Although label smoothing reduces the calibration error compared to non-smoothed training, it is still slightly inferior to temperature scaling (Müller, Kornblith, and Hinton 2019).

Label Relaxation

In the following, we detail our idea of label relaxation as an alternative to label smoothing, i.e., the idea of modeling deterministic data, namely observed class labels, in terms of a set of probability distributions instead of a single target distribution. We also propose a generalization of an underlying loss function, which compares probabilistic predictions with a set of candidate distributions, and derive a closed-form expression for the case of the KL divergence.

Motivation

Consider a conventional setting of supervised learning, in which we are interested in learning a probabilistic classifier $\hat{p} : \mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$, where $\mathcal{X}$ is an instance space, $\mathcal{Y} = \{y_1, \ldots, y_K\}$ a set of class labels, and $\mathbb{P}(\mathcal{Y})$ the space of probability distributions over $\mathcal{Y}$. To this end, we typically proceed from training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$, i.e., observations in the form of instances labeled by one of the classes. Thus, even if we assume a ground-truth (conditional) probability distribution $p_i^* = p^*(\cdot \,|\, \boldsymbol{x}_i)$ to exist for each $\boldsymbol{x}_i \in \mathcal{X}$, this distribution will normally not be provided as training information. Instead, the training will be based on the deterministic label y_i, which is (explicitly or implicitly) treated as a degenerate (one-point) distribution $p_i \in \mathbb{P}(\mathcal{Y})$ such that $p_i(y_i \,|\, \boldsymbol{x}_i) = 1$ and $p_i(y \,|\, \boldsymbol{x}_i) = 0$ for $y \neq y_i$.

Needless to say, making the realistic assumption of a non-deterministic dependency between $\mathcal{X}$ and $\mathcal{Y}$, the true distribution p_i^* will normally be less extreme than the surrogate p_i. Therefore, providing the former as training information may suggest a level of determinism that is actually not warranted. As a consequence, the learner will be encouraged to make extreme predictions, which suggests a high degree of confidence, leading to biased probability estimates and a tendency to overfit the training data — all the more when training flexible models such as neural networks.

In label smoothing, a surrogate distribution p is replaced by a less extreme surrogate $p^s = (1 - \alpha)\, p + \alpha\, u$ as a target for the learner, where $u \in \mathbb{P}(\mathcal{Y})$ is a fixed distribution and $\alpha \in (0, 1]$ a smoothing factor. As shown by Szegedy et al. (2016), the resulting cross-entropy H, which often serves as a loss function for the learner, for a prediction $\hat{p} \in \mathbb{P}(\mathcal{Y})$ is of the form

$$\begin{aligned} H(p^s, \hat{p}) &= (1 - \alpha) H(p, \hat{p}) + \alpha H(u, \hat{p}) \\ &= (1 - \alpha) H(p, \hat{p}) + \alpha \left(D_{KL}(u||\hat{p}) + H(u) \right) \quad . \end{aligned} \tag{1}$$

Since $H(p) = 0$ for a degenerate p, the first term on the right-hand side simplifies to $H(p, \hat{p}) = D_{KL}(p||\hat{p}) + H(p) = D_{KL}(p||\hat{p})$, with D_{KL} the Kullback-Leibler divergence. Moreover, assuming u to be independent of $\hat{p}$, $H(u)$ can be treated as a constant with no influence on loss minimization. Furthermore, as pointed out by Szegedy et al. (2016), the divergence $D_{KL}(u||\hat{p})$ essentially corresponds to the negative entropy of $\hat{p}$ for the case where u is the uniform distribution. Thus, we eventually end up with a loss function of the form

$$L(p^s, \hat{p}) = (1 - \alpha)\, D_{KL}(p||\hat{p}) + \alpha\, H(\hat{p}) \quad , \tag{2}$$

i.e., a loss that augments the original cross-entropy loss by a penalty that enforces a higher entropy for the prediction $\hat{p}$, and which hence serves the purpose of regularization, as empirical results have confirmed (Pereyra et al. 2017). Training a learner with the loss (2) will obviously lead to less extreme predictions (for $\alpha = 1$, the learner will always predict the uniform distribution on $\mathcal{Y}$).

Thus, there are different ways of looking at label smoothing. According to what we just explained, it can be seen as a regularization technique, which may explain its practical usefulness. On the other side, coming back to the discussion we started with, it can also be seen as an attempt at presenting the training information in a more "faithful" way: A smoothed target probability p^s is arguably more realistic than a degenerate distribution p assigning the fully probability mass to a single class label.

However, it is still unlikely that the adjusted distribution p^s matches the ground-truth p^*. Therefore, using p^s as a more or less arbitrary target, the learner will still be biased in a possibly undesirable way. Related to this, one may wonder whether a systematic penalization of the learner for "overly correct" predictions, i.e., predictions $\hat{p}$ that are closer to the original p (the truly observed class label) than p^s, is indeed appropriate. At least in some cases, such predictions could be justified, and indeed be closer to the ground-truth.

As a presumably better but at least more faithful representation of our knowledge about the ground-truth p^*, we propose to replace the original target p by a *set* $Q \subset \mathbb{P}(\mathcal{Y})$ of candidate probabilities that are "sufficiently close" to the original target p. While the replacement of p by a single distribution p^s can be seen as a distortion of the original target, this can be considered as a *relaxation* of the target: As long as the learner predicts any distribution $\hat{p} \in Q$ inside the candidate set, it should not be penalized at all, i.e., the loss should be 0. This is to some extent comparable to the use of loss functions like the ϵ-insensitive loss in support vector regression, where the loss is 0 in the ϵ-neighborhood of the original target; essentially, this means that the original target, which is a real number, is relaxed and replaced by a set in the form of an interval.

Note that, by using set-valued targets Q_i for the training instances $\boldsymbol{x}_i$, a regularization effect can also be expected: By accepting all predictions as correct that are sufficiently close to the original target distribution, the learner is still allowed to produce extreme predictions but no longer urged to do so. Instead, the learner is more flexible and can freely choose a target $p_i^r \in Q$ that appears most appropriate. Since the p_i^r are the result of a learning process and determined in a data-driven way, one expects them to be closer to the p_i^* than the surrogates p_i^s, which are chosen arbitrarily. This is completely in line with the idea of *data disambiguation* in the context of learning from set-valued data (Hüllermeier and Cheng 2015). See Fig. 1 for an illustration of the conceptual differences between label smoothing and label relaxation.

Loss Formulation

To formalize the ideas sketched above, we leverage the theory of imprecise probabilities (Walley 1991). A convenient way to express a set of probability distributions is to provide *upper probabilities*, i.e., upper bounds on the probabilities of events. So-called *possibility distributions* $\pi : \mathcal{Y} \longrightarrow [0, 1]$

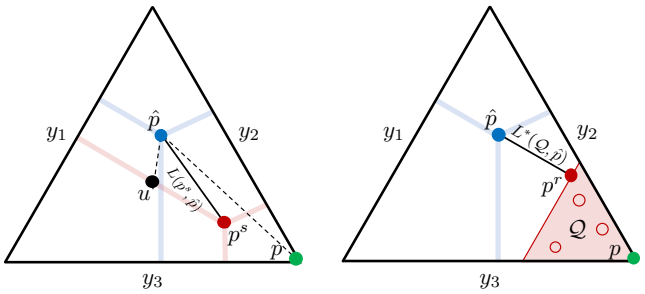

Figure 1: An illustration of label smoothing (left) and label relaxation (right), using a barycentric representation, in which points correspond to (3-class) distributions and probabilities are given by the lengths of the projections to the sides of the triangle. In the former, the original distribution p (in the lower right corner) is shifted toward the uniform distribution u, and the loss of a prediction $\hat{p}$ depends on the (KL) distance to p^s, hence on the distance to p as well as u. In label relaxation, p is replaced by a set $\mathcal{Q}$ of distributions, indicated by the shaded region. The learner is free to choose any of the distributions inside this set (non-filled circles inside the region), and the loss is determined by the minimal distance between $\hat{p}$ and any of the distributions in $\mathcal{Q}$ (filled red circle).

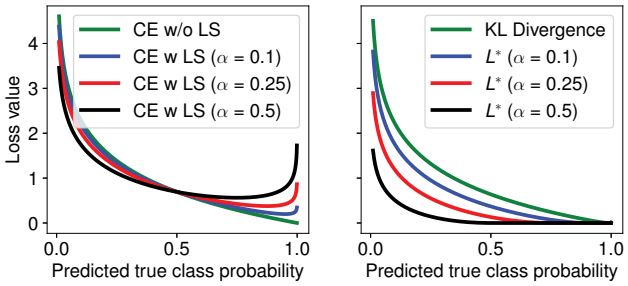

Figure 2: A comparison of the different losses discussed in this paper for binary classification. Left: Cross-entropy losses with and without label smoothing. Right: LR loss L^* based on the Kullback-Leibler divergence.

are often interpreted in this way (Dubois and Prade 2004), i.e., in the sense that $\pi(y)$ is an upper bound on $p^*(y)$. More generally, since a possibility distribution π induces a measure Π on $\mathcal{Y}$ defined by $\Pi(Y) = \max_{y \in Y} \pi(y)$ for all $Y \subseteq \mathcal{Y}$, the set of probability distributions associated with a distribution π is given by

$$Q_\pi := \left\{ p \in \mathbb{P}(\mathcal{Y}) \,|\, \forall Y \subseteq \mathcal{Y} : \sum_{y \in Y} p(y) \leq \max_{y \in Y} \pi(y) \right\} .$$

Note that a possibility distribution π is assumed to be normalized in the sense that $\pi(y) = 1$ for at least one $y \in Y$. In other words, there is at least one alternative that appears completely plausible. In our case, this alternative naturally corresponds to the class label that has actually been observed for a training instance: Potentially, this class may have a (conditional) probability of 1.

However, by assigning a certain degree $\pi(y) > 0$ of possibility also to the other classes, we can express that these classes are not completely excluded either. More specifically, consider a distribution of the following kind:

$$\pi_i(y) = \begin{cases} 1 & \text{if } y = y_i \\ \alpha & \text{if } y \neq y_i \end{cases} ,$$

where $\alpha \in [0, 1]$ is a parameter. By definition, the associated set $Q^\alpha_{\pi_i}$ is then given by the set of probability distributions p that assign a probability mass of *at most* 1 to the observed class y_i and *at most* α to the other classes:

$$Q_i^\alpha := \left\{ p \in \mathbb{P}(\mathcal{Y}) \,|\, \sum_{y_i \neq y \in \mathcal{Y}} p(y) \leq \alpha \right\} \tag{3}$$

Replacing the class labels y_i observed as training information by sets Q_i^α as new targets for the learner, we need to define a suitably generalized loss function L^*. Since the learner is still assumed to produce probabilistic predictions, the loss should be able to compare a predicted distribution $\hat{p}_i$ with a candidate set Q_i^α. According to what we said before, namely that a prediction inside Q_i^α should be considered as perfect, a natural definition is

$$L^*(Q, \hat{p}) := \min_{p \in Q} L(p, \hat{p}) , \tag{4}$$

where L is a standard loss on probability distributions, i.e., a loss $L : \mathbb{P}(\mathcal{Y})^2 \longrightarrow \mathbb{R}$. Interestingly, (4) can be seen as a special case of what has been introduced under the notion of *optimistic superset loss* in the context of superset learning (Hüllermeier and Cheng 2015), and more recently as *infimum loss* by Cabannes, Rudi, and Bach (2020). In the following, we shall refer to (4) as *label relaxation* (LR) loss.

As a theoretically convenient case, instantiating L with the Kullback-Leibler divergence, that is,

$$L(p, \hat{p}) := D_{KL}(p||\hat{p}) = \sum_{y \in \mathcal{Y}} p(y) \log \frac{p(y)}{\hat{p}(y)} ,$$

(4) simplifies as follows for sets Q_i^α of the form (3):

$$L^*(Q_i^\alpha, \hat{p}_i) = \begin{cases} 0 & \text{if } \hat{p}_i \in Q_i^\alpha \\ D_{KL}(p_i^r||\hat{p}_i) & \text{otherwise} \end{cases} , \tag{5}$$

where

$$p_i^r(y) = \begin{cases} 1 - \alpha & \text{if } y = y_i \\ \alpha \cdot \frac{\hat{p}_i(y)}{\sum_{y' \neq y_i} \hat{p}_i(y')} & \text{otherwise} \end{cases} . \tag{6}$$

We refer to the technical appendix for a formal proof of this result.

Fig. 2 shows a comparison of label smoothing (cross-entropy losses with and without smoothing, left side) with our label relaxation loss (right side). As can be seen (and is proven in the appendix), L^* based on the Kullback-Leibler divergence is convex, which makes the optimization computationally feasible. Moreover, label smoothing is not monotone and again increases for predictions close to 1, while the LR loss vanishes for values $\geq 1 - \alpha$. This cut reflects the relaxation of the problem. For multi-class problems, since the proposed label relaxation loss projects the predicted probabilities from $\hat{p}_i$ according to its own distribution to p_i^r, it is

invariant to the concretely predicted probabilities for classes not equal to the observed class.

Interestingly, the resulting losses as shown in Fig. 2 for varying α parameters seem to be very related to the focal loss as introduced in (Lin et al. 2020). However, while the focal loss deemphasizes predictions in the "well-classified" region by *almost* flat regions, our loss completely eliminates the loss in such a region for the genuine relaxation.

Evaluation

To demonstrate the effectiveness of label relaxation, an empirical evaluation on image classification datasets assessing the classification performance and calibration is conducted.

Experimental Setting

Within the empirical evaluation of our method proposal, we compare models trained by conventional cross-entropy (CE), label smoothing (LS), confidence penalizing (CP) as described by Pereyra et al. (2017), and the focal loss (FL) of Lin et al. (2020) to our label relaxation (LR) approach. To this end, we study the performances on neural networks for the task of image classification. Although the losses are completely general and not specifically tailored to any domain, this problem serves as a good representative and has been used in related studies in the past.

In addition to assessing the generalization accuracy in terms of the classification rate, we also measure the degree of calibration of the networks, i.e., the quality of the predicted class probabilities. To this end, we use the estimated expected calibration error (ECE) as done by Guo et al. (2017). This measure requires probabilities to be discretized through binning, and as suggested by Müller, Kornblith, and Hinton (2019), we fix the number of bins to 15. To compare label smoothing and our approach with explicit calibration methods, non-calibration and temperature scaling (Guo et al. 2017) serve as baselines.

Within our study, we consider MNIST (LeCun et al. 1998), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton 2009) as image datasets. While MNIST and Fashion-MNIST both have 60k training and 10k test examples, CIFAR-10 and CIFAR-100 each consist of 50k training and 10k test instances. For the first two datasets, we train our models on a simple fully connected, ReLU activated neural network structure with two hidden layers consisting of 1024 neurons each. For the latter two datasets, we train the commonly used deep architectures VGG16 (Simonyan and Zisserman 2015), ResNet56 (V2) (He et al. 2016) and DenseNet-BC-100-12 (Huang et al. 2017). While we repeated every experiment for MNIST and Fashion-MNIST with 10 different seeds, we run each of the latter experiments 5 times. The runs were conducted on 20 Nvidia RTX 2080 Ti and 10 Nvidia GTX 1080 Ti GPUs.

For a fair comparison, all hyperparameters are fixed, except the parameter α in the case of label relaxation and smoothing loss, β as degree of confidence penalization in CP, and γ as being used to adjust the focal loss. For every combination of model and dataset, we empirically determined hyperparameters (such as the learning rate schedule and additional regularization) that work reasonably well for all losses. Since all losses are quite similar to each other, this was possible without favoring some of them while putting others at a disadvantage. To diminish regularization effects by additional means, we tried to exclude other techniques (such as extensive weight decay or Dropout (Srivastava et al. 2014)) as much as possible, thereby emphasizing the effect of the different loss functions while still achieving performances close to the originally published results.

To optimize the models, SGD with a Nesterov momentum of 0.9 has been used as optimizer. In all experiments, the batch size has been fixed to 64. Depending on the model, we set the initial learning rates to 0.01 (VGG), 0.05 (simple dense), and 0.1 (ResNet and DenseNet). For each model, we optimized the learning rate schedule for generalization performance by dividing the learning rate by a constant factor (ranging from 0.1 to $\sqrt{0.1}$). We trained for either 25 (MNIST), 50 (Fashion-MNIST), 200 (CIFAR-10), or 300 (CIFAR-100) epochs. Furthermore, we used data augmentation by randomly horizontally flipping and shifting the input images in width and height. We refer to the appendix for a more comprehensive overview of the fixed hyperparameters.

Since the parameters α for LR and LS, β for CP, and γ for FL are of critical importance, they have been optimized separately on a separate hold-out validation set consisting of $1/6$ of the original training data. In the first experiments, we optimize this parameter for the highest classification rate, whereas in the second evaluation, we focus on a low ECE. In both cases, we assessed values $\alpha \in \{0.01, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4\}$, $\beta \in \{0.1, 0.3, 0.5, 1, 2, 4, 8\}$, and $\gamma \in \{0.1, 0.2, 0.5, 1, 2, 3.5, 5\}$ as suggested as reasonable parameters in the corresponding publications. The best model is then retrained on the original training data and evaluated on a separate test set. For each seed, the original training and test splits are merged and resampled to increase the variance of the experiments. This way, we achieve a better estimation of the generalization error. However, as a consequence, the presented results are not directly comparable to previously published results based on the original splits, although this special case is also covered in our experiments.

For temperature scaling, a separate hold-out validation set is used to optimize the parameter T among the values $T \in \{0.25, 0.5, 0.75, 1, 1.1, 1.2, \ldots, 2, 2.5, 3\}$. This parameter highly depends on the actually trained model and does not generalize well, i.e., a value performing well on an inner optimization run does not necessarily imply good calibration on the model finally trained. Therefore, the evaluation scenario is slightly different compared to the optimization of α: For the latter, as opposed to the case of temperature scaling, the hold-out validation set is included in the final training using the optimized parameters. This can be regarded as the price being paid for explicitly calibrating the model with temperature scaling, as opposed to the implicit calibration achieved by label smoothing and label relaxation. Here, we use 15% of the training data for calibration, which is almost comparable to the validation set used for optimizing α.

Loss	MNIST		Fashion-MNIST		Avg. Rank	
	Acc.	ECE	Acc.	ECE	Acc.	ECE
CE ($\alpha = 0$)	0.985 ± 0.002	0.010 ± 0.001	0.912 ± 0.003	0.129 ± 0.184	2	3.5
LS (α opt. for acc.)	$\mathbf{0.988} \pm 0.001$	0.106 ± 0.144	$\mathbf{0.915} \pm 0.002$	0.155 ± 0.128	**1**	5
CP (β opt. for acc.)	0.985 ± 0.002	0.012 ± 0.002	0.911 ± 0.004	0.075 ± 0.005	3	3.5
FL (γ opt. for acc.)	0.984 ± 0.002	0.009 ± 0.002	0.911 ± 0.002	0.062 ± 0.002	4.5	2
LR (α opt. for acc.)	0.985 ± 0.001	$\mathbf{0.007} \pm 0.002$	0.912 ± 0.003	$\mathbf{0.059} \pm 0.008$	2	**1**
CE ($\alpha = 0$, T opt.)	0.983 ± 0.001	$\mathbf{0.003} \pm 0.001$	0.908 ± 0.004	0.030 ± 0.003	4	2.5
LS (α opt. for ECE)	$\mathbf{0.987} \pm 0.001$	0.014 ± 0.001	$\mathbf{0.915} \pm 0.003$	0.016 ± 0.002	**1**	3.5
CP (β opt. for ECE)	0.984 ± 0.001	0.011 ± 0.001	0.911 ± 0.003	0.072 ± 0.003	2.5	4
FL (γ opt. for ECE)	0.982 ± 0.001	0.004 ± 0.001	0.907 ± 0.003	$\mathbf{0.011} \pm 0.002$	5	**1.5**
LR (α opt. for ECE)	0.985 ± 0.002	$\mathbf{0.003} \pm 0.001$	0.911 ± 0.003	0.015 ± 0.003	2	**1.5**

Table 1: Results on MNIST and Fashion-MNIST using a simple 2-layer dense architecture. Bold entries indicate the best combination with regard to the corresponding metric per dataset and optimization scheme. The resulting ranks are averaged over both datasets for the respective metric.

Results

Table 1 shows the results of all assessed loss variants with regard to their classification performance and calibration error on MNIST and Fashion-MNIST. As can be seen, with a single exception, our label relaxation approach provides the lowest calibration error on both datasets regardless of the optimization target (accuracy or ECE). Although models trained with the focal loss deliver competitive calibration results, they generalize worse than LR optimized models. Label smoothing delivers the highest classification rate, while lacking calibration abilities. By still having a competitive classification rate compared to LS, our method offers a reasonable compromise between strong generalization (in terms of classification rate) and good calibration.

Since the accuracies on MNIST and Fashion-MNIST are already quite high, a more insightful evaluation is given by the experiments on CIFAR-10 and CIFAR-100, using multiple popular deep convolutional network architectures. Table 2 summarizes the results for both datasets and the different topologies. With few exceptions, LR minimizes the calibration error in terms of ECE among the assessed losses. At the same time, in accordance with the results presented before, it provides competitive classification rates. While FL-based models also yield relatively low calibration errors, they sometimes drop significantly in terms of classification performance (e.g., VGG16 and DenseNet-BC on CIFAR-100 optimized for ECE). Also, although temperature scaling uses separate data to explicitly optimize the temperature for a low calibration error, the implicit calibration of LR outperforms temperature scaling in most of the cases. Thus, relying on losses that implicitly calibrate models seems to be a reasonable strategy for model calibration.

To get a better overview of the presented results, Table 3 shows the resulting aggregated ranks for all datasets and models per metric and parameter optimization target (accuracy or ECE). For both optimization schemes, LR clearly dominates the other losses in terms of the calibration error. At the same time, the overall classification performance is reasonably close to the best loss, especially when applying accuracy-based hyperparameter optimization. As the results demonstrate, it balances both metrics and provides a compelling alternative to the other losses, particularly for applications in which the aim is to predict probabilities matching the underlying true probabilities of the classes.

Conclusion

We proposed label relaxation as an alternative to label smoothing, an established technique for preventing overfitting and over-confidence in classifier learning: Instead of replacing the original (degenerate) distribution associated with an observed class label by another, smoother yet still precise distribution, we relax the problem by letting the learner choose from a larger set of such distributions. This kind of "imprecisiation" of training data relieves the learner from the need to reproduce unrealistically definite observations, very much like label smoothing, but also allows it to predict probabilities in a flexible way. This flexibility appears to be important, not only for accurate classification, but even more so for producing less biased and better calibrated probability estimates.

These reflections are confirmed by an empirical study in image classification. Here, the calibration of deep convolutional neural network models could be improved without a loss in classification accuracy compared to label smoothing, penalizing confident output distributions and focal loss-based optimization. Label relaxation even outperforms explicit calibration methods like temperature scaling, which, due to requiring extra data for calibration, often pay with a drop in classification performance.

The idea of modeling targets in supervised learning in terms of imprecise probabilities, combined with the minimization of generalized losses penalizing deviations from the set of associated precise distributions, is very general and could be instantiated in various ways. Here, we considered the problem of classification and generalized the KL divergence. However, motivated by the promising empirical results, we also plan to look at other problems and other combinations of "data imprecisiation" and loss functions. Even-

Model	Loss	CIFAR-10		CIFAR-100		Avg. Rank	
		Acc.	ECE	Acc.	ECE	Acc.	ECE
VGG16	CE ($\alpha = 0$)	**0.930** $\pm$ 0.002	0.041 $\pm$ 0.001	0.708 $\pm$ 0.003	0.196 $\pm$ 0.003	**1.5**	3.5
	LS (α opt. for acc.)	0.929 $\pm$ 0.001	0.148 $\pm$ 0.119	**0.711** $\pm$ 0.003	0.149 $\pm$ 0.049	**1.5**	3.5
	CP (β opt. for acc.)	0.927 $\pm$ 0.001	0.059 $\pm$ 0.002	0.703 $\pm$ 0.003	0.228 $\pm$ 0.013	3	4.5
	FL (γ opt. for acc.)	0.921 $\pm$ 0.001	0.038 $\pm$ 0.004	0.700 $\pm$ 0.005	0.190 $\pm$ 0.009	5	2.5
	LR (α opt. for acc.)	0.927 $\pm$ 0.002	**0.033** $\pm$ 0.008	0.701 $\pm$ 0.006	**0.133** $\pm$ 0.069	3.5	**1**
	CE ($\alpha = 0$, T opt.)	0.922 $\pm$ 0.001	**0.017** $\pm$ 0.003	0.689 $\pm$ 0.005	0.053 $\pm$ 0.003	3.5	2
	LS (α opt. for ECE)	**0.932** $\pm$ 0.002	0.028 $\pm$ 0.010	**0.711** $\pm$ 0.003	0.085 $\pm$ 0.005	**1**	4
	CP (β opt. for ECE)	0.922 $\pm$ 0.003	0.050 $\pm$ 0.003	0.700 $\pm$ 0.004	0.209 $\pm$ 0.004	3	5
	FL (γ opt. for ECE)	0.918 $\pm$ 0.003	0.027 $\pm$ 0.001	0.684 $\pm$ 0.007	0.048 $\pm$ 0.014	5	2.5
	LR (α opt. for ECE)	0.926 $\pm$ 0.001	0.022 $\pm$ 0.001	0.703 $\pm$ 0.006	**0.046** $\pm$ 0.005	2	**1.5**
ResNet56 (V2)	CE ($\alpha = 0$)	0.940 $\pm$ 0.002	0.041 $\pm$ 0.002	0.737 $\pm$ 0.003	0.126 $\pm$ 0.003	3	3
	LS (α opt. for acc.)	0.938 $\pm$ 0.002	0.132 $\pm$ 0.145	0.733 $\pm$ 0.004	0.110 $\pm$ 0.061	4.5	4
	CP (β opt. for acc.)	0.939 $\pm$ 0.003	0.046 $\pm$ 0.004	**0.738** $\pm$ 0.004	0.151 $\pm$ 0.007	2	4
	FL (γ opt. for acc.)	**0.941** $\pm$ 0.002	**0.036** $\pm$ 0.006	**0.738** $\pm$ 0.005	0.107 $\pm$ 0.017	**1**	**1.5**
	LR (α opt. for acc.)	0.938 $\pm$ 0.003	0.059 $\pm$ 0.090	**0.738** $\pm$ 0.003	**0.092** $\pm$ 0.030	2.5	2.5
	CE ($\alpha = 0$, T opt.)	0.933 $\pm$ 0.002	0.030 $\pm$ 0.002	0.709 $\pm$ 0.005	0.041 $\pm$ 0.006	5	3.5
	LS (α opt. for ECE)	**0.940** $\pm$ 0.002	0.017 $\pm$ 0.002	0.730 $\pm$ 0.004	0.053 $\pm$ 0.003	2	3
	CP (β opt. for ECE)	**0.940** $\pm$ 0.002	0.044 $\pm$ 0.001	**0.741** $\pm$ 0.003	0.140 $\pm$ 0.003	**1**	5
	FL (γ opt. for ECE)	0.938 $\pm$ 0.002	0.017 $\pm$ 0.002	0.738 $\pm$ 0.005	0.024 $\pm$ 0.003	3	2
	LR (α opt. for ECE)	0.939 $\pm$ 0.002	**0.016** $\pm$ 0.002	0.729 $\pm$ 0.003	**0.017** $\pm$ 0.003	3.5	**1**
DenseNet-BC (100-12)	CE ($\alpha = 0$)	**0.929** $\pm$ 0.003	0.050 $\pm$ 0.002	**0.706** $\pm$ 0.005	0.229 $\pm$ 0.005	**1**	4
	LS (α opt. for acc.)	0.927 $\pm$ 0.004	0.046 $\pm$ 0.011	0.704 $\pm$ 0.008	**0.182** $\pm$ 0.063	4	**1.5**
	CP (β opt. for acc.)	**0.929** $\pm$ 0.002	0.056 $\pm$ 0.004	0.698 $\pm$ 0.011	0.252 $\pm$ 0.018	3	5
	FL (γ opt. for acc.)	0.928 $\pm$ 0.003	0.047 $\pm$ 0.004	0.703 $\pm$ 0.001	0.223 $\pm$ 0.001	3.5	3
	LR (α opt. for acc.)	0.928 $\pm$ 0.002	**0.039** $\pm$ 0.014	**0.706** $\pm$ 0.003	0.203 $\pm$ 0.023	2	**1.5**
	CE ($\alpha = 0$, T opt.)	0.921 $\pm$ 0.003	**0.009** $\pm$ 0.004	0.687 $\pm$ 0.006	0.096 $\pm$ 0.006	4	**2**
	LS (α opt. for ECE)	**0.928** $\pm$ 0.003	0.020 $\pm$ 0.002	**0.704** $\pm$ 0.015	**0.077** $\pm$ 0.035	**1**	2.5
	CP (β opt. for ECE)	**0.928** $\pm$ 0.002	0.054 $\pm$ 0.002	**0.704** $\pm$ 0.003	0.237 $\pm$ 0.003	**1**	5
	FL (γ opt. for ECE)	0.915 $\pm$ 0.003	0.016 $\pm$ 0.001	0.681 $\pm$ 0.004	0.133 $\pm$ 0.004	5	3
	LR (α opt. for ECE)	0.922 $\pm$ 0.002	0.017 $\pm$ 0.003	0.703 $\pm$ 0.008	0.085 $\pm$ 0.006	3	2.5

Table 2: Results on CIFAR-10 and CIFAR-100 for the assessed model architectures. Here, bold entries indicate the best performances among the loss variants per dataset, model and optimization scheme. The ranks are averaged over both datasets as done before.

Loss	Acc. Opt.		ECE Opt.		Overall	
	Acc.	ECE	Acc.	ECE	Acc.	ECE
CE	**1.88**	3.5	4.13	2.5	3	3
LS	2.75	3.5	**1.25**	3.25	**2**	3.38
CP	2.75	4.25	1.88	4.75	2.31	4.5
FL	3.5	2.25	4.5	2.25	4	2.25
LR	2.5	**1.5**	2.63	**1.63**	2.56	**1.56**

Table 3: Average ranks of the losses with regard to the accuracy and ECE when a) optimizing the accuracy, b) optimizing the ECE and c) the overall ranking.

tually, a broader study of different instantiations should lead to a deeper understanding and general methodology of label relaxation.

Acknowledgments

This work was partially supported by the German Research Foundation (DFG) under Grant No. 3050231323. The authors gratefully acknowledge the funding of this project by computing time provided by the Paderborn Center for Parallel Computing (PC^2).

References

Bagherinezhad, H.; Horton, M.; Rastegari, M.; and Farhadi, A. 2018. Label Refinery: Improving ImageNet Classification through Label Progression. *CoRR* abs/1805.02641.

Cabannes, V.; Rudi, A.; and Bach, F. R. 2020. Structured Prediction with Partial Labelling through the Infimum Loss. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event, July 13-18, 2020*, volume 119, 1230–1239. PMLR.

Cireşan, D. C.; Meier, U.; Gambardella, L. M.; and Schmidhuber, J. 2010. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation* 22(12): 3207–3220.

Dubey, A.; Gupta, O.; Raskar, R.; and Naik, N. 2018. Maximum-Entropy Fine Grained Classification. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 635–645.

Dubois, D.; and Prade, H. 2004. Possibility Theory, Probability Theory and Multiple-Valued Logics: A Clarification. *Annals of Mathematics and Artificial Intelligence* 32: 35–66.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, August 6-11, 2017*, volume 70 of *Proceedings of Machine Learning Research*, 1321–1330. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity mappings in deep residual networks. In *Proceedings of the 14th European Conference on Computer Vision, ECCV 2016, Amsterdam, The Netherlands, October 11-14, 2016, Part IV*, 630–645. Springer.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2261–2269. IEEE Computer Society.

Hüllermeier, E.; and Cheng, W. 2015. Superset Learning Based on Generalized Loss Minimization. In Appice, A.; Rodrigues, P. P.; Costa, V. S.; Gama, J.; Jorge, A.; and Soares, C., eds., *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II*, volume 9285 of *Lecture Notes in Computer Science*, 260–275. Springer.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Canada.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Bartlett, P. L.; Pereira, F. C. N.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Proceedings of the 26th Annual Conference on Neural Information Processing Systems, NIPS 2012, Lake Tahoe, NV, USA, December 3-6, 2012*, 1106–1114.

Kull, M.; Filho, T. S.; and Flach, P. 2017. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Singh, A.; and Zhu, J., eds., *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, Fort Lauderdale, FL, USA, April 20-22, 2017*, volume 54 of *Proceedings of Machine Learning Research*, 623–631. PMLR.

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

Li, W.; Dasarathy, G.; and Berisha, V. 2020. Regularization via Structural Label Smoothing. In Chiappa, S.; and Calandra, R., eds., *23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, Online [Palermo, Sicily, Italy], August 26-28, 2020*, volume 108 of *Proceedings of Machine Learning Research*, 1453–1463. PMLR.

Li, Y.; Yang, J.; Song, Y.; Cao, L.; Luo, J.; and Li, L. 2017. Learning from Noisy Labels with Distillation. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 1928–1936. IEEE Computer Society.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 42(2): 318–327.

Lukasik, M.; Bhojanapalli, S.; Menon, A. K.; and Kumar, S. 2020. Does Label Smoothing Mitigate Label Noise? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event, July 13-18, 2020*, volume 119, 6448–6458. PMLR.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When Does Label Smoothing Help? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS 2019, Vancouver, BC, Canada, December 8-14, 2019*, 4696–4705.

Naeini, M. P.; Cooper, G. F.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, Texas, USA, January 25-30, 2015*, 2901–2907. AAAI Press.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, L.; and Hinton, G. E. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. *CoRR* abs/1701.06548.

Reed, S. E.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2015. Training Deep Neural Networks on Noisy Labels with Bootstrapping. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1): 1929–1958.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2818–2826. IEEE Computer Society.

van der Maaten, L.; Chen, M.; Tyree, S.; and Weinberger, K. Q. 2013. Learning with marginalized corrupted features. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, 410–418.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning, ICML 2008, Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, 1096–1103. ACM.

Walley, P. 1991. *Statistical reasoning with imprecise probabilities*. Chapman & Hall.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR* abs/1708.07747.

Xie, L.; Wang, J.; Wei, Z.; Wang, M.; and Tian, Q. 2016. DisturbLabel: Regularizing CNN on the Loss Layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 4753–4762. IEEE Computer Society.

Yun, S.; Park, J.; Lee, K.; and Shin, J. 2020. Regularizing Class-Wise Predictions via Self-Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 13873–13882. IEEE.

Zadrozny, B.; and Elkan, C. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, 694–699. ACM.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 3712–3721. IEEE.