Large Margin In Softmax Cross-Entropy Loss

Takumi Kobayashi
takumi.kobayashi@aist.go.jp

National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Japan

Abstract

Deep convolutional neural networks (CNNs) are trained mostly based on the softmax cross-entropy loss to produce promising performance on various image classification tasks. While much research effort has been made to improve the building blocks of CNNs, the classifier margin in the loss attracts less attention for optimizing CNNs in contrast to the kernel-based methods, such as SVM. In this paper, we propose a novel method to induce a large-margin CNN for improving the classification performance. By analyzing the formulation of the softmax loss, we clarify the margin embedded in the loss as well as its connection to the distribution of softmax logits. Based on this analysis, the proposed method is formulated as regularization imposed on the logits to induce a large-margin classifier in a compatible form with the softmax loss. The experimental results on image classification using various CNNs demonstrate that the proposed method favorably improves performance compared to the other large-margin losses.

1 Introduction

In recent years, convolutional neural networks (CNNs) are widely applied to various image classification tasks with great success [15]. While much research effort has been made in improving CNNs from the architectural viewpoint [10, 21, 28], there is a risk of over-fitting in the deep CNNs due to the huge number of CNN parameters. To cope with the over-fitting problem, the CNNs are trained on large-scale annotated image datasets [5, 32] with data augmentation techniques [15]. In addition to the approach toward enlarging the training data, some network layers such as BatchNormalization [12], DropOut [22] and stochastic pooling [30] effectively contribute to properly training the CNNs.

The CNNs are generally optimized based on the softmax cross-entropy loss, *softmax loss* in short. The softmax loss is arguably the most popular classification loss due to its simple formulation and probabilistic interpretation [2, 6]. On the other hand, to address the issue of over-fitting as well as promote the discriminative training, the contrastive loss is proposed [9] to deal with pair-wise samples through Siamese networks. It is further extended to the triplet loss [20] for levering triplet samples to the training. Those methods polynomially increase the number of training pairs and triplets, thus requiring an efficient sample selection scheme to cope with large-scale data. There are also *large-margin* approaches on the loss function [3, 16, 26]. The classifier margin had been attracted keen attention in the framework of kernel-based methods, such as SVM [1], for improving generalization performance [25] while mitigating the over-fitting. In the deep learning literature, the large-margin methods focus on the forms of classifier, such as inner-product [3, 16] and distance-based GMM [26], and then directly manipulate the classifier margin during training to provide large-margin CNNs. Those methods, however, demand to carefully control the margins throughout the end-to-end training since the larger margin poses the harder optimization problem; for properly training CNNs, the margin should be gradually increased as the training epoch proceeds.

In this paper, we propose a novel method to induce large-margin CNNs *implicitly* in comparison to the *explicit* large-margin methods [16, 26] mentioned above. Through analyzing the softmax loss, we reveal that the margin is embedded in the loss in a form dependent on the softmax logits; that is, even the softmax loss encourages a large-margin classifier to some extent. Based on this analysis, we formulate the proposed method as *regularization* on the logits to further enhance the large-margin effect in the softmax loss. The proposed method *indirectly* affects the classifier margin via the regularization without touching margin; the margin is adaptively controlled during the training in contrast to the large-margin methods that directly manipulate the margins. Thus, by incorporating the proposed regularization into the ordinary softmax loss, we can simply improve the performance of CNNs without changing the other components of CNNs nor training procedures.

2 Toward Large-Margin Loss

2.1 General formulation of margin based loss

In multi-class classification of C classes, a CNN provides a logit vector $\boldsymbol{f} \in \mathbb{R}^C$ for softmax; the logits are produced such as by the last fully-connected layer applying the linear classifier $\boldsymbol{f} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$ to the feature vector $\boldsymbol{x}$ of the penultimate layer. Following [4], it is natural to regard as a classification *margin* the difference between the target score (logit) f_y and the other maximum score f_{c^*}, where y indicates the ground truth class label and $c^* = \arg\max_{c \neq y} f_c$. Based on the margin $f_y - f_{c^*}$, the classification loss function is generally defined as

$$\mathtt{l}(\boldsymbol{f}, y) = \mathtt{L}(\max_{c \neq y} f_c - f_y + \rho) = \mathtt{L}(f_{c^*} - f_y + \rho), \tag{1}$$

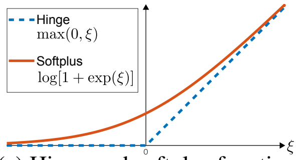
where $\mathtt{L}(\xi)$ is a monotonically increasing function (Fig. 1a) to measure the loss based on the biased margin $\xi = f_{c^*} - f_y + \rho$. Here, the bias $\rho > 0$ operates like a *lower bound*[1] of the classification margin so that $f_y - \max_{c \neq y} f_c \geq \rho$; the larger bias would enlarge the classification margin via the loss function. In [4], the multi-class SVM classifier is optimized by employing the hinge loss $\mathtt{L}(\xi) = \max(0, \xi)$ with the bias $\rho = 1$.

2.2 Softmax cross-entropy loss

We analyze the softmax cross-entropy loss (*softmax loss*) from the viewpoint of mathematical formulation. Given the logit vector $\boldsymbol{f} \in \mathbb{R}^C$ and the ground truth label $y \in \{1, \cdots, C\}$, the softmax loss is formulated as the following cross entropy between the softmax posterior and the ground truth one;

$$\mathtt{l}(\boldsymbol{f}, y) = -\log \mathrm{p}_y(\boldsymbol{f}) = -\log \frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)} = \log\left[\sum_{c=1}^{C} \exp(f_c - f_y)\right], \tag{2}$$

[1] ρ is strictly a lower bound in the hinge function producing zero loss for $\xi \leq 0$. It works in a *pseudo* manner of lower bound for the softplus function which exponentially approaches zero on $\xi \leq 0$ as shown in Fig. 1a.



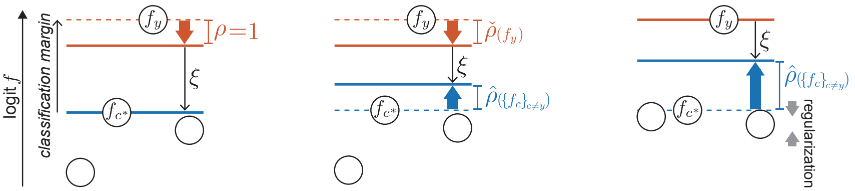
(a) Hinge and softplus functions

Margin-based Loss: $\mathtt{L}(\max_{c\neq y} f_c - f_y + \rho)$

	Hinge loss	Softmax loss
loss function $\mathtt{L}(\xi)$	$\max[0,\xi]$	$\log[1+\exp(\xi)]$
bias ρ	1	$\log[\sum_{c\neq y}\exp(f_c - f_{c^*})]$

(b) Comparison in the large-margin framework (see Sec. 2.2)

Figure 1: Comparison of hinge loss and softmax loss in the framework of margin-based loss.



(a) Hinge loss [4] (b) Large-margin losses [16, 26] (c) Ours

Figure 2: Margins in various loss methods. The circles indicate the logits of corresponding classes; f_y is the ground truth one and $f_{c^*} = \max_{c\neq y} f_c$. ξ is the biased margin to be finally assessed by the function $\mathtt{L}(\xi)$ for the classification loss (1); $\mathtt{L}(\xi) = \max(0,\xi)$ in (a), while $\mathtt{L}(\xi) = \text{softplus}(\xi)$ in (b,c). The methods are also characterized by the forms of bias ρ.

which can be rewritten by using *softplus* function [7], $\text{softplus}(\xi) = \log\{1+\exp(\xi)\}$, into

$$\mathtt{l}(\boldsymbol{f},y) = \log\left[1+\sum_{c\neq y}\exp(f_c - f_y)\right] = \text{softplus}\left[\underbrace{\log\left\{\sum_{c\neq y}\exp(f_c)\right\}}_{\text{LSE}(\{f_c\}_{c\neq y})} - f_y\right]. \tag{3}$$

This reformulation reveals that the softmax loss (2) measures the significance of the target logit f_y in comparison with the others $\{f_c\}_{c\neq y}$ by applying softplus to the loss function $\mathtt{L}$ in (3), while the hinge function is employed in SVM [4] (Sec. 2.1). It is noteworthy that the softplus and hinge functions are compared from the viewpoint of classification loss (Fig. 1a), though those functions have been discussed in the framework of non-linear activation [8]. Then, to further analyze the softmax loss from the viewpoint of the margin-based loss (1), we focus on the log-sum-exp (LSE) transformation that aggregates the logits of the non-target classes $c\neq y$ in (3), implicitly playing a key role for inducing a large-margin classifier in the softmax loss.

The log-sum-exp (LSE) function, which is applied to the logits $\{f_c\}_{c\neq y}$ in (3), holds the following relationship [18];

$$\max_{c\neq y}(f_c) = f_{c^*} < \text{LSE}(\{f_c\}_{c\neq y}) \leq f_{c^*} + \log(C-1), \tag{4}$$

since the LSE implicitly contains the maximum logit f_{c^*} in the form of

$$\text{LSE}(\{f_c\}_{c\neq y}) = f_{c^*} + \log\left[\sum_{c\neq y}\exp(f_c - f_{c^*})\right] = f_{c^*} + \log\left[1+\sum_{c\notin\{c^*,y\}}\exp(f_c - f_{c^*})\right], \tag{5}$$

where $0 < \exp(f_c - f_{c^*}) \leq 1$. The LSE function is regarded as providing the smooth maximum due to the relationship (4) in which the second equality holds only for the uniform logits $f_c = f_{c^*}, \forall c \neq y$.

By combining (3) and (5), the softmax loss (2) results in

$$\mathtt{l}(\boldsymbol{f},y) = \mathrm{softplus}\left(f_{c^*} - f_y + \log\left[\sum_{c\neq y}\exp(f_c - f_{c^*})\right]\right), \tag{6}$$

which is interpreted as the margin-based loss (1) by setting $\mathtt{L}(\xi) = \mathrm{softplus}(\xi)$ and the bias of

$$\hat{\rho}(\{f_c\}_{c\neq y}) = \log\left[\sum_{c\neq y}\exp(f_c - f_{c^*})\right]. \tag{7}$$

Thus, the softmax loss induces a large-margin classifier due to this bias $\hat{\rho}$, though it is not explicit in the original loss form (2). It should be noted that the bias (7) is a variable depending on the logits $\{f_c\}_{c\neq y}$, while the hinge loss employs a constant $\rho = 1$, as shown in Fig. 1b.

2.3 Regularization for large margin

The bias $\hat{\rho}$ (7), which is implicitly embedded in the softmax loss, depends on the distribution of the non-target logits $\{f_c\}_{c\neq y}$ through the log-sum-exp (LSE) function. Based on the characteristics of the LSE shown in (4), the bias approaches zero by isolating the maximum f_{c^*} from the others ($f_{c^*} \gg f_c$), while it becomes larger as the distribution of the logits $\{f_c\}_{c\neq y}$ is close to uniform $f_c = f_{c^*}, \forall c \neq y$. Thus, in order to derive the larger-margin classifier, we propose a regularization method to enlarge the bias $\hat{\rho}$ by increasing uniformity of those logits. For that purpose, we measure the diversity of the logits $\{f_c\}_{c\neq y}$ by means of the symmetric Kullback-Leibler divergence;

$$\tilde{\mathcal{D}}(\mathtt{q}) = \frac{1}{2}\{\mathcal{D}(\mathtt{q}\,\|\,\mathtt{u}) + \mathcal{D}(\mathtt{u}\,\|\,\mathtt{q})\} = \frac{1}{2}\sum_{c\neq y}\left\{\mathtt{q}_c - \frac{1}{C-1}\right\}\log(\mathtt{q}_c), \tag{8}$$

where

$$\text{Softmax posterior:}\left\{\mathtt{q}_c = \frac{\exp(f_c)}{\sum_{c'\neq y}\exp(f_{c'})}\right\}_{c\neq y},\ \text{Uniform probability:}\left\{\mathtt{u}_c = \frac{1}{C-1}\right\}_{c\neq y}, \tag{9}$$

$$\text{KL divergences: }\mathcal{D}(\mathtt{q}\,\|\,\mathtt{u}) = \sum_{c\neq y}\mathtt{q}_c\log\left[(C-1)\mathtt{q}_c\right],\ \mathcal{D}(\mathtt{u}\,\|\,\mathtt{q}) = \sum_{c\neq y}\frac{1}{C-1}\log\left[\frac{1}{(C-1)\mathtt{q}_c}\right]. \tag{10}$$

We leverage the symmetric KL divergence (8) to the regularization with the softmax loss (3) for encouraging uniformity of the logits $\{f_c\}_{c\neq y}$, thereby enlarging the bias $\hat{\rho}$ in (7). Thus, the proposed large-margin loss is finally described by

$$\mathtt{l}_{ours}(\boldsymbol{x},y) = -\log\frac{\exp(f_y)}{\sum_{c=1}^{C}\exp(f_c)} + \frac{\lambda}{2}\sum_{c\neq y}\left\{\frac{\exp(f_c)}{\sum_{c'\neq y}\exp(f_{c'})} - \frac{1}{C-1}\right\}\log\left\{\frac{\exp(f_c)}{\sum_{c'\neq y}\exp(f_{c'})}\right\}, \tag{11}$$

of which derivatives with respect to the logits $\boldsymbol{f}$ are given by

$$\frac{\partial\mathtt{l}_{ours}}{\partial f_c} = \begin{cases} \mathtt{p}_y - 1 & c = y \\ \mathtt{p}_c + \frac{\lambda}{2}\left[\mathtt{q}_c(x_c - \sum_{d\neq y}x_d\mathtt{q}_d) + (\mathtt{q}_c - \frac{1}{C-1})\right] & c \neq y \end{cases}, \tag{12}$$

where we use two types of softmax posteriors $\mathrm{p}_c = \frac{\exp(f_c)}{\sum_{c'=1}^{C}\exp(f_{c'})}$ and $\mathrm{q}_c = \frac{\exp(f_c)}{\sum_{c'\neq y}\exp(f_{c'})}$. The proposed loss layer (11) is stacked on the top of the network in the same manner as an ordinary loss, without changing any other components in the network nor training procedure.

As shown in Fig. 2, we compare the loss methods in terms of margin on the basis of the general margin-based loss formulation (Sec. 2.1) as follows.

– The **hinge loss [4]** imposes a constant bias $\rho = 1$.

– The **previous large-margin losses [16, 26]** for training deep CNNs introduce the extra bias $\check{\rho}$ in addition to the one $\hat{\rho}$ (7) naturally embedded in the softmax loss; the bias $\check{\rho}$ is formulated based on the target logit f_y. Directly manipulating the margin via the extra bias requires to be carefully controlled throughout the end-to-end learning since the larger margin generally poses the more difficult classification problem; it is important to gradually enhance the effect of margin as suggested by the authors [16, 26]. Practically speaking, however, it is hard to properly design the optimization schedule regarding the margin besides the overall learning parameters, such as learning rate, in a trial-and-error approach for deep neural networks on large-scale datasets.

– The **proposed method** *indirectly* enhances the bias $\hat{\rho}$ through the regularization based on the divergence of logits (8) without touching the margin itself. Thus, we can simply apply the proposed loss (11) with a constant regularization parameter, say $\lambda = 0.3$ in this study, following the standard training procedure of CNNs. Thus, in comparison to the other methods [16, 26], the proposed method is practically useful from this optimization viewpoint.

3 Discussion

Connection to Label Smoothing Regularization (LSR) [27]. The label smoothing regularization is again attracting attention for training deep CNNs [23, 31]. The method slightly degrades the ground truth by introducing uniform distribution over C classes. The softmax cross-entropy loss with so polluted ground truth label results in

$$\mathtt{l}_{LSR}(\boldsymbol{x},y) = -(1-\lambda)\log(\mathrm{p}_y) - \frac{\lambda}{C}\sum_{c=1}^{C}\log(\mathrm{p}_c) = -(1-\lambda)\log(\mathrm{p}_y) + \lambda\{\mathcal{D}(\mathrm{u}\,\|\,\mathrm{p}) + \log(C)\}, \tag{13}$$

where the regularization parameter λ is introduced to control the degree of degrading labels. The regularization term corresponds to the KL divergence $\mathcal{D}(\mathrm{u}\|\mathrm{p})$ (10) between the uniform distribution u and the softmax posterior p over C classes. Thus, from our viewpoint (Sec. 2), LSR can be interpreted as regularization to suppress the diversity of $\{f_c\}_{c\neq y}$ *to some extent* in a similar way to ours (11), which also enlarges the bias $\hat{\rho}$ thereby contributing to a large-margin classifier. It is noteworthy that the proposed framework theoretically reveals the large-margin aspect of LSR, though it has been employed rather heuristically. And, our method (11) is clearly different from LSR (13) in that our regularization (8) excludes the logit f_y of the ground truth class to straightforwardly enhance the bias $\hat{\rho}$ (7) via the symmetric KL divergence. It effectively encourages a large-margin classifier without impeding the discriminative training based on ground truth labels. On the other hand, LSR pollutes the ground truth information by multiplying $1-\lambda$ and thus requires the regularization parameter to be carefully determined; $\lambda = 0.1$ in general [23].

Connection to Center Loss [27]. The proposed method could be slightly related to the center loss [27] which minimizes the within-class variance as regularization combined with the softmax loss. They, however, differ in the following three points regarding regularization functionality. First, the center loss contributes to discriminative feature representation while our regularization works on enlarging classification margins for higher generalization performance. Second, the center loss focuses on the feature distribution across *samples* and our method considers the diversity of logits over *classes* at each sample. Third, the center loss is applied to the penultimate layer that provides a feature vector $\boldsymbol{x}$ unlike our regularization that operates at the last loss layer. Thus, the proposed method would be compatible with the center loss by jointly working as complementary regularization to train CNNs.

Geometrical meaning of the regularization. Here, we consider the classifier $\boldsymbol{f} = \boldsymbol{W}^\top \boldsymbol{x}$ that minimizes the proposed regularization term (8), *i.e.*, producing the logits $\{f_c\}_{c \neq y}$ close to uniform. On the assumption that there is no correlation among classes, such a classifier holds the *super-symmetric* form of $\boldsymbol{w}_c^\top \boldsymbol{w}_c = \nu$, $\boldsymbol{w}_c^\top \boldsymbol{w}_{c'} = -\frac{\nu}{C-1}, \forall c \neq c', \exists \nu > 0$; for the detail, refer to the supplementary material. The super-symmetric classifier is closely connected to the optimal classifier on the simplex build upon the optimal Bayesian feature representation [19]. Thus, the proposed regularization pushes a classifier toward the super-symmetric one while enlarging classifier margins as well as encouraging discriminative feature representation.

Margin by LSE. While the bias (7) is naturally derived from LSE in the softmax loss, it is possible to apply the hard (constant) bias $\rho = 1$ to reformulate the softmax loss (3) into $\text{softplus}(f_{c^*} - f_y + 1)$, analogous to the hinge loss [4] of $\max(0, f_{c^*} - f_y + 1)$. In preliminary experiments, however, we found that CNNs are not properly trained at all by both this modified loss and the hinge loss. Those losses explicitly employ the maximum logit f_{c^*} and thereby provide the updates (gradients) only for the two logits of f_y and f_{c^*} which are poorly back-propagated and too sparse to train the deep neural networks. On the other hand, the LSE-based *soft* bias (7) works well to effectively provide the dense gradients (12) compatible for the back-propagation.

4 Experimental Results

The proposed method (11) is generally applicable to training CNNs as a classification loss, and we evaluate the performance of the method on image classification tasks.

4.1 Ablation study

We analyze the proposed method from various aspects by applying it to train the 13-layer network [24] on Cifar100 dataset [14]; we train the network, whose detailed architecture is shown in the supplementary material, by applying SGD with a batch size of 128, weight decay of 0.0001, momentum of 0.9 and the learning rate monotonically decreasing in log scale from 0.1 to 0.0001 over 50 epochs. We repeat the evaluation three times with different initial random seeds to report the average and the standard deviation of error rates (%).

In Sec. 2, we considered the regularization to enhance uniformity of the logits, leading to the large-margin classifier, and for that purpose there can be three types of divergence as the regularization; the asymmetric ones $\mathcal{D}(\mathtt{q} \| \mathtt{u})$ and $\mathcal{D}(\mathtt{u} \| \mathtt{q})$ in (10), and the symmetric one $\tilde{\mathcal{D}}(\mathtt{q})$ averaging those two in (8). Those three types of regularization forms are compared in Table 1 on various regularization parameter values λ. We can see that $\mathcal{D}(\mathtt{u} \| \mathtt{q})$ and $\tilde{\mathcal{D}}(\mathtt{q})$ are slightly superior to $\mathcal{D}(\mathtt{q} \| \mathtt{u})$, since the divergence $\mathcal{D}(\mathtt{u} \| \mathtt{q})$ enhances the uniformity more strongly than $\mathcal{D}(\mathtt{q} \| \mathtt{u})$ as shown in Fig. 4. By combining those asymmetric divergences, the symmetric $\tilde{\mathcal{D}}(\mathtt{q})$ stably contributes to the performance improvement. As to the regularization parameter λ, the performance is sufficiently improved at $\lambda = 0.3$. Based on these

Table 1: Performance results on three types of regularization forms with various regularization parameter λ.

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\mathcal{D}(\mathtt{u}\,\|\,\mathtt{q})$	27.99±0.18	27.62±0.16	27.30±0.34	27.35±0.14	26.98±0.22	27.11±0.22	26.77±0.35	26.99±0.16	26.89±0.07
$\mathcal{D}(\mathtt{q}\,\|\,\mathtt{u})$	28.45±0.41	28.52±0.44	27.79±0.23	27.77±0.09	27.40±0.11	27.33±0.12	27.15±0.07	27.14±0.16	26.98±0.09
$\tilde{\mathcal{D}}(\mathtt{q})$	28.21±0.32	27.92±0.12	27.27±0.15	27.38±0.10	27.13±0.17	26.73±0.15	26.98±0.09	27.13±0.08	26.87±0.08

Table 2: Comparison to the methods discussed in Sec. 3.

Method	Error rate (%)
Softmax loss	28.62±0.24
Super-symmetry	28.67±0.46
LSR (13) ($\lambda = 0.01$)	28.80±0.23
LSR (13) ($\lambda = 0.05$)	28.51±0.22
LSR (13) ($\lambda = 0.1$)	28.48±0.13
LSR (13) ($\lambda = 0.3$)	28.65±0.25
Ours (11)	27.27±0.15

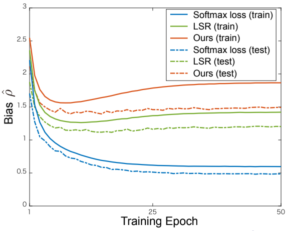


Figure 3: Bias value $\hat{\rho}$ (7) during the training.

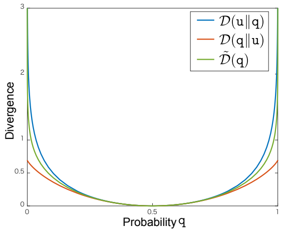


Figure 4: Divergences in case of two categories $\{\mathtt{q}, 1-\mathtt{q}\}$.

experimental results, we use the regularization of $\tilde{\mathcal{D}}(\mathtt{q})$ in (11) with $\lambda = 0.3$[2].

Then, the proposed method is compared to the ones that are intrinsically related to ours as discussed in Sec. 3; the super-symmetric classifier and the label smoothing regularization (LSR) [23]. We implement the super-symmetric classifier by imposing the constraint of super-symmetry on the classifier; the detailed form of the constraint is shown in the supplementary material. For fair comparison, the LSR (13) is equipped with various λs, including $\lambda = 0.1$ suggested in [23]. As shown in Table 2, the proposed method outperforms those comparison methods as well as the original softmax loss. The super-symmetric constraint poorly works, even deteriorating the performance. Such a constraint on the classifier is too strong to properly train the network from scratch and thus it might be necessary to gradually enhance the effect of the constraint during the end-to-end learning. And, the actual class categories would have some correlation which slightly violates the ideal super-symmetric form.

The LSR rather favorably works since both regularizations of ours and LSR induces a large-margin classifier through enlarging the bias $\hat{\rho}$ in (7). Fig. 3 shows the bias value $\hat{\rho}$ empirically measured during the training. As the training proceeds, the original softmax loss decreases it monotonically while to the regularizations of ours and LSR work to first decrease and then increase the bias. This experimental result also demonstrates that the regularization adaptively controls the biased margin ξ (Fig. 2) according to the situation of the trained network without manually designing the optimization schedule [16, 26]. And, we can empirically validate the role of LSR for the large-margin classifier (Sec. 3), though such an aspect of LSR has not been mentioned so far. Nonetheless, the LSR is inferior to the proposed method, requiring the regularization parameter λ to be carefully tuned, due to the degradation of the ground truth label. The proposed regularization form (8) excluding the ground-truth class is compatible with the primary softmax loss without impeding the discriminative training based on the ground truth. Actually, as shown in Table 1, our regularization can work with even larger λ while LSR accepts the carefully tuned small λ, usually $\lambda = 0.1$ as suggested in [23].

[2] While $\lambda > 0.5$ also works on some datasets as shown in the other experimental results, we find $\lambda = 0.3$ generally improves performance on various datasets and CNNs.

Table 3: Performance results on the degenerated training set of Cifar100.

(a) Smaller-scale training samples

# of sample ($\times 50,000$)	0.9	0.8	0.7	0.6	0.5
Softmax loss	29.66±0.21	30.94±0.06	32.71±0.07	34.11±0.14	36.87±0.26
Ours ($\lambda = 0.3$)	28.62±0.12	29.95±0.09	31.35±0.47	33.49±0.22	36.06±0.06
Ours ($\lambda = 0.5$)	28.09±0.03	29.56±0.20	30.83±0.20	33.13±0.09	35.58±0.37
Ours ($\lambda = 1$)	27.88±0.23	29.21±0.13	30.83±0.15	33.11±0.46	35.49±0.15

(a) Noisy ground truth labels

Pollution rate	0.1	0.2	0.3	0.4	0.5
Softmax loss	33.97±0.21	37.17±0.34	40.54±0.35	43.65±0.81	48.37±0.19
Ours ($\lambda = 0.3$)	31.59±0.17	34.98±0.07	38.23±0.26	42.06±0.11	46.87±0.32
Ours ($\lambda = 0.5$)	31.00±0.19	34.38±0.19	37.91±0.17	41.53±0.31	46.41±0.09
Ours ($\lambda = 1$)	30.42±0.20	33.50±0.07	36.75±0.23	40.76±0.25	46.43±0.19

4.2 Degenerated training set

The large-margin inducing regularization would improve the generalization performance of neural networks so as to effectively cope with a degenerated training set. In this experiment, we evaluate such a generalization performance on two types of degradation regarding the *number of training samples* and the *correctness of the ground truth annotation* over training samples. These two situations are frequently found on the real-world recognition tasks in the cases that collecting training samples is costly and manual annotation is poorly performed by immatured annotators.

The proposed method is applied to train the 13-layer network on the smaller-scale training set of Cifar100; we sub-sample the training samples by the ratio from 0.9 (45,000 samples) to 0.5 (25,000 samples). On the other hand, in order to simulate the noisy ground truth label, we switch the ground truth labels into the other incorrect ones only on the partial set of the training samples; the ratio of the polluted training samples is from 0.1 (*clean*) to 0.5 (*dirty*). Table 3 shows the performance results on those degenerated training sets. In both cases, the proposed method works well, outperforming the softmax loss, and we can see that the stronger regularization with the larger λ clearly improves the performance, which demonstrates the effectiveness of the larger-margin classifier on these training situations; in the case of noisy annotation, the large-margin regularization helps the training to exploit the consistently discriminative information that the correct labels exhibit.

4.3 Comparison to the other methods

Finally, the proposed method is compared with the other large-margin methods. As mentioned in Sec. 3, the multi-class hinge loss [4] in Fig. 1b does not work for training CNNs at all due to the sparse gradients (updates). To mitigate the issue, we modify the hinge loss into

$$\mathtt{l}_{hinge}(\boldsymbol{x},y) = \frac{1}{C-1}\sum_{c\neq y}\max[0, f_c - f_y + 1], \tag{14}$$

so as to provide dense gradients over all the logits $\boldsymbol{f}$ for properly optimizing the deep CNNs via back-propagation. In addition, inspired by the hard bias $\rho = 1$ in the hinge loss (Fig. 2), the softmax cross-entropy loss (3) can also be modified to

$$\mathtt{l}_{mod}(\boldsymbol{x},y) = \mathrm{softplus}\left[\log\left\{\sum_{c\neq y}\exp(x_c)\right\} - x_y + 1\right] = -\log\frac{\exp(x_y - 1)}{\exp(x_y - 1) + \sum_{c\neq y}\exp(x_c)}, \tag{15}$$

Table 4: Performance comparison on the deep CNNs.

Dataset	Cifar10 [14]	SVHN [17]	ImageNet [6]		
Network	WResNet [29]	WResNet [29]	VGG-16 [21]	VGG-16-mod [13]	MobileNet [11]
Softmax loss	$4.73_{\pm 0.07}$	$2.93_{\pm 0.03}$	27.94	25.65	29.84
Hinge loss (14)	$5.26_{\pm 0.15}$	$3.15_{\pm 0.04}$	46.58	45.17	55.79
Modified loss (15)	$4.68_{\pm 0.07}$	$2.89_{\pm 0.05}$	27.65	24.92	29.15
LGM [26]	$4.75_{\pm 0.04}$	$2.65_{\pm 0.07}$	-	-	34.12
LSM [16]	$4.49_{\pm 0.13}$	$2.80_{\pm 0.03}$	27.92	24.98	39.35
Ours ($\lambda = 0.3$)	$4.60_{\pm 0.05}$	$2.67_{\pm 0.03}$	27.27	24.53	28.94

Dataset	ImageNet [6]	Places-365 [32]
Network	ResNeXt-50 [28]	VGG-16-mod [13]
Softmax loss	22.69	45.02
Ours ($\lambda = 0.3$)	22.27	44.61

Dataset	ImageNet [6]	
Network	VGG-16-mod [13]	ResNeXt-50 [28]
Modified loss (15) + Ours ($\lambda = 0.3$)	24.37	22.00

where the bias $\check{\rho} = 1$ is added to $\hat{\rho}$ (7) in our framework (Fig. 2). For comparison, we also apply the large-margin methods of LGM [26] and LSM [16]. The large-margin effects in the losses of LSM and LGM are dynamically controlled so as to gradually increase the margin throughout the end-to-end training; the dynamic schedule is designed in the way that the authors suggest [16, 26]. Note that while LGM [26] alters the linear classifier into the distance-based one by means of Gaussian mixtures, the other methods including ours are applied to the ordinary linear classifier $\boldsymbol{f} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$.

These loss functions are applied to train the deep CNNs on image classification tasks; WideResNet28-10 (WResNet) [29] on Cifar10 [14] and SVHN [17] datasets, and VGG-16 [21], modified VGG-16 (VGG-16-mod) [13] and MobileNet [11] on ImageNet [6] dataset. We train these CNNs from scratch and report (top-1) error rate (%) on the validation set provided in the respective datasets; the detailed training procedures for these CNNs are provided in the supplementary material.

The performance results are shown in Table 4. The modified hinge loss (14) poorly works, indicating that the hinge function providing rather sparse gradients is not suitable for training CNNs. Though the LGM and LSM work relatively favorably on WResNet, they fail to improve the performance of MobileNet. This result shows that it is necessary to carefully tailor the the margin-controlling schedule in LGM and LSM for each network by considering both the datasets and the training procedures, which imposes a heavy burden. In LGM, the VGG models are not favorably trained, being collapsed, maybe due to the large dimensionality (4096-dim.) of the feature vectors $\boldsymbol{x}$ at the penultimate layer; it might be difficult to train the distance-based classifier (GMM) in such a large dimensional feature space. On the other hand, the modified softmax loss (15), which is improved in our large-margin framework, works fairly well in spite of the simple formulation, and in particular the proposed method produces favorable performance on these various CNNs by consistently improving the performance of the softmax loss. Note again that these two methods simply substitute for the ordinary softmax loss without controlling the regularization parameter λ during the end-to-end training. The proposed method is further applied to the deeper CNN of ResNeXt-50 [28] on ImageNet and to the scene classification task on Places-365 dataset [32] using the CNN of VGG-16-mod. As shown in Table 4, the method also renders the performance improvement in these cases, demonstrating the general applicability to various classification. The proposed method (8) works just as regularization so that it is applicable to the other softmax-based losses. For example, we can apply the proposed method to the modified softmax loss (15) which contains the hard bias $\check{\rho} = 1$ producing favorable performance as shown above, and find that the combination further improves performance as demonstrated in Table 4.

5 Conclusion

In this paper, we have proposed a novel regularization method to enhance the classification margin in the softmax loss. Through analyzing the softmax loss, we reveal the large-margin effect in the loss which is dependent on the distribution of logits, and then formulate the regularization on the logits by means of the symmetric KL divergence for inducing the large-margin classifier. Our analysis of the softmax loss also theoretically clarifies the large-margin aspect of the label smoothing regularization which has been applied rather heuristically. In the experiments on various image classification using deep CNNs, the proposed method produces favorable performance in comparison with the other large-margin methods.

References

[1] P. J. Bartlett, B. Schölkopf, D. Schuurmans, and A. J Smola. *Advances in Large-Margin Classifiers*. MIT Press, 2000.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] B. Chen, W. Deng, and J. Du. Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation. In *CVPR*, pages 4021–4030, 2017.

[4] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[6] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64, 2014.

[7] C. Dugas, Y. Bengio, F. Belisle, C. Nadeau, and R. Garcia. Incorporating second-order functional knowledge for better option pricing. In *NIPS*, pages 451–457, 2001.

[8] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, pages 315–323, 2011.

[9] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Journal of Machine Learning Research*, 37:448–456, 2015.

[13] T. Kobayashi. Analyzing filters toward efficient convnets. In *CVPR*, pages 5619–5628, 2018.

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

[15] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[16] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, pages 507–516, 2016.

[17] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[18] F. Nielsen and K. Sun. Guaranteed bounds on the kullback-leibler divergence of univariate mixtures. *Entropy*, 18(12):442, 2016.

[19] N.Otsu. Optimal linear and nonlinear solutions for least-square discriminant feature extraction. In *ICPR*, pages 557–560, 1982.

[20] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[24] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, pages 1195–1204, 2017.

[25] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[26] W. Wan, Y. Zhong, T. Li, and J. Chen. Rethinking feature distribution for loss functions in image classification. In *CVPR*, pages 9117–9126, 2018.

[27] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515, 2016.

[28] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017.

[29] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016.

[30] M. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. In *ICLR*, 2013.

[31] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, pages 3774–3782, 2017.

[32] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.