Hindawi
Complexity
Volume 2019, Article ID 9206053, 10 pages
https://doi.org/10.1155/2019/9206053

Research Article

GO Loss: A Gaussian Distribution-Based Orthogonal Decomposition Loss for Classification

Mengxin Liu,[1] **Wenyuan Tao**,[1] **Xiao Zhang**,[2] **Yi Chen**,[3] **Jie Li**,[1] **and Chung-Ming Own**[1]

[1]*College of Intelligence and Computing, Tianjin University, Tianjin 300072, China*
[2]*Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong SAR 999077, China*
[3]*Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China*

Correspondence should be addressed to Chung-Ming Own; chungming.own@tju.edu.cn

Received 17 June 2019; Revised 31 October 2019; Accepted 20 November 2019; Published 12 December 2019

Academic Editor: Mahardhika Pratama

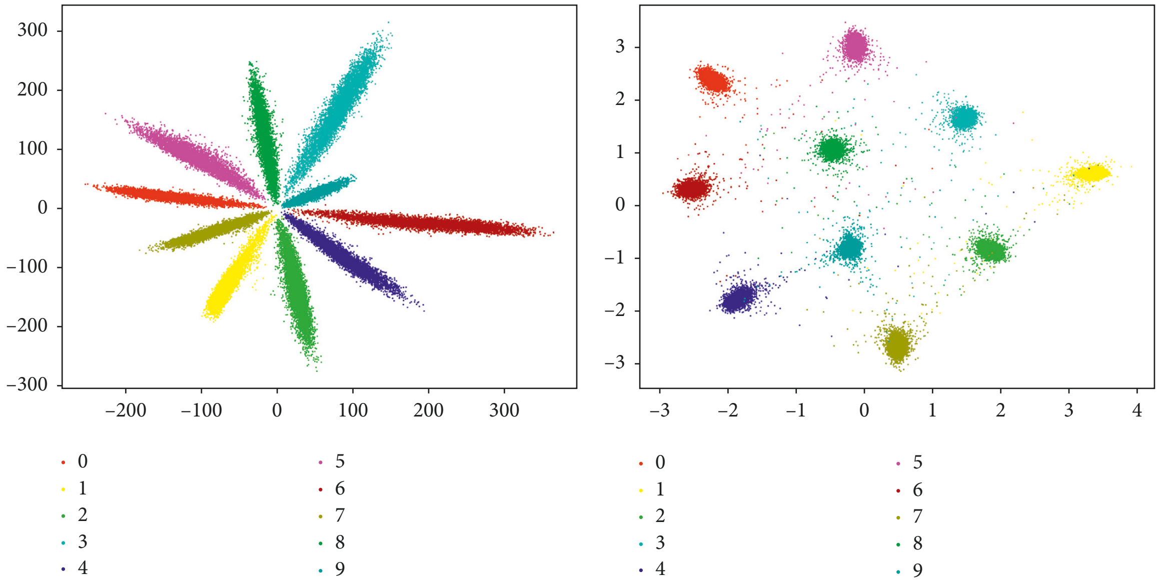
We present a novel loss function, namely, GO loss, for classification. Most of the existing methods, such as center loss and contrastive loss, dynamically determine the convergence direction of the sample features during the training process. By contrast, GO loss decomposes the convergence direction into two mutually orthogonal components, namely, tangential and radial directions, and conducts optimization on them separately. The two components theoretically affect the interclass separation and the intraclass compactness of the distribution of the sample features, respectively. Thus, separately minimizing losses on them can avoid the effects of their optimization. Accordingly, a stable convergence center can be obtained for each of them. Moreover, we assume that the two components follow Gaussian distribution, which is proved as an effective way to accurately model training features for improving the classification effects. Experiments on multiple classification benchmarks, such as MNIST, CIFAR, and ImageNet, demonstrate the effectiveness of GO loss.

1. Introduction

In recent years, deep neural networks have achieved great success [1, 2], and classification tasks have been widely used in various fields [3–6]. Loss function is an indispensable part of deep learning; various kinds of loss functions, such as MSE and BCE, are available for different tasks, including image-based object recognition [7–9], face recognition [10–12], and speech recognition [13, 14]. The performance of loss functions has been widely studied [15, 16]. A good loss function should theoretically make the distribution of features of different classes separated while ensuring the features of the same class as compact as possible.

Among the existing loss functions, soft-max cross-entropy is the most common [9, 17–19]. However, soft-max only ensures the separability of the features of different classes while lacking the ability to compress distances among features within the same class. As a result, the distances between features of different classes are less than those of the same class, as shown in Figure 1(a).

Several variants have been proposed to improve the intraclass compactness of soft-max. Some metric learning methods are used to promote the classification effectively [20–22]. These studies attempt to resolve this problem through feature normalization [23, 24] or adding an extra regularization item to construct a joint supervision [25–28]. In these studies, the stochastic gradient descent algorithm has been widely used. This algorithm can determine a convergence direction in each iteration on the fly, depending on the network parameters and training samples at the time. The feature as a vector can be decoupled into two components, namely, direction and norm. Theoretically, the two components determine the interclass separability and intraclass compactness of the distribution of the sample features, respectively. Therefore, if we treat the feature as a whole, as what the existing works do, then the optimizations of the two components will be intertwined. Therefore, the computation of the convergence center has to simultaneously consider the two components, which will interfere with each other and thus affect the final classification effects.



(a)

(b)

Figure 1: Distributions of the features trained using (a) soft-max and (b) GO loss on MNIST. Each color represents a class. The GO loss has better intraclass compactness and interclass separation than traditional soft-max loss. Best viewed in color.

In this paper, we propose an orthogonal decomposition-based loss function called GO loss, which decomposes the convergence direction into two mutually orthogonal components. Moreover, we assume that the two components follow Gaussian distribution. Specifically, the norm of the feature in the radial direction follows Gaussian distribution, while the angles (cosine value) between the features and the corresponding center vector (class weight vector) of the class in tangential direction also follow Gaussian distribution. This assumption enables the use of Bayes' rule in loss computation, which is an effective manner to model training features. We can therefore (1) model the classification loss as the cross entropy between the posterior probability of features and the corresponding class labels in tangential direction, called tangential loss, and (2) compute the difference between the norm of feature distribution and the assumed distribution in radial direction using the negative log likelihood, called radial loss. The two losses can be used to form a joint supervision for balancing interclass separability and intraclass compactness on the learned training feature space; thus, a high classification accuracy can be ensured, as shown in Figure 1.

In summary, the main contribution of the paper is a novel loss function for classification, namely, GO loss, which integrates the following:

(i) A strategy to optimize loss function through decomposing the convergence direction into two mutually orthogonal components and conducting optimization on them, respectively. This approach is different from most traditional methods that mainly rely on feature normalization and adding regularization item. The rationale is to avoid the mutual influence of the optimizations on the two components for obtaining a stable convergence center.

(ii) A solution that implements the optimization. This solution decouples the feature into direction and norm associated with the interclass separability and intraclass compactness, respectively, and conduct optimizations on the two components with the assumption that they follow Gaussian distribution.

2. Related Works

For various classification tasks, the loss function directly affects the classification effect [29–31]. In the existing methods, metric learning is widely used in the loss function to improve the classification effect [32–34]. The idea of GO loss is based on existing loss functions. We highlight the most related aspects below.

Soft-max is one of the most common loss functions in classification. It uses the inner product metric to implement the classification function. However, loose intraclass feature distribution brings difficulty in handling complex classification problems. Many other metrics, such as Euclidean and cosine distances, have been used to resolve the aforementioned problem. Thus, many variants of soft-max are available.

Contrastive loss [25] uses a predefined margin to train a Siamese network for face recognition. It minimizes the Euclidean distances between positive pairs and enlarges the Euclidean distances between negative face image pairs. However, the combinatorial explosion problem of image pairs will greatly increase the number of iterations.

Triplet loss [26] applies Euclidean distance regularization for loss optimization. The regularization is conducted on image triplets rather than the image pairs of contrastive loss to achieve a high accuracy of face recognition. However, it has the same problem as contrastive loss in terms of computational complexity.

Center loss [27] minimizes the Euclidean distance between each feature vector and its class center. However, the extra regularization item generates two convergence directions, which not only increases computation complexity but also makes the convergence center unstable to some extent.

Gaussian mixture loss [28] is an effective alternative to soft-max. Center loss is a special case of the likelihood regularization in the GM loss. The problem of Gaussian mixture is the same as that of center loss. Thus, the former generates increased computation overhead.

Ring loss [23] utilizes a different optimization mechanism, which normalizes all features through a convex augmentation of the primary loss function. In that case, all the features are put around a ring. As a result, all features have the same norm and thus cannot be used for optimization.

Large-margin soft-max loss [24] uses the cosine distance metrics to solve the inconsistency problem of distance measurements. It introduces an angular margin in the soft-max through a well-designed angular distance function. It mainly focuses on angular variation while ignoring important influence of norm on the classification effects.

The abovementioned methods optimize the loss function from the perspective of the feature distribution. Regularizing the extracted features or adding regularization terms makes the features of the same class compact and the features of different classes separated. Based on this, several loss functions for classification have been studied from the perspective of redesigning clusters [35, 36], such as GCPL loss [37] and Structure-aware loss [38].

L2T-DLF [39], which means “learning to teach with dynamic loss functions,” is a novel model to train the loss function. Through the training process, the model adjusts and changes the loss function. The trained loss function is best suited to the datasets. As a result, the best classification results are obtained.

Noise-robust loss [40] uses the joint supervision of categorical cross-entropy loss and mean absolute error to optimize the loss function from the perspective of noise-robust. When the label has a wide range of noises, this loss function can exert a better classification effect than other loss functions, which normalize the features.

SL [41], which means “symmetric cross-entropy learning,” is also proposed to solve the noise-robust problem. It boosts cross-entropy symmetrically with a noise-robust counterpart called reverse cross-entropy. SL overcomes the overfitting and under learning problem of cross-entropy when the label has the noise.

Recent research on loss function focuses on the application scenario of loss function. The methods study the loss function for the characteristics of the datasets, such as the presence of noisy labels.

As same as the existing works, we also improve the classification effect from the perspective of intraclass compactness and interclass separation of feature distribution. The aforementioned methods regard direction and norm as a whole for optimizing the loss. On the contrary, GO loss performs optimization on the two characteristics separately. This approach has not been tried before to the best of our knowledge. An unknown sufficiently large sample can be approximated as obeying Gaussian distribution. Considering the characteristics of the datasets, we reasonably assume that features obey Gaussian distribution. We use the Gaussian distribution to guide the optimization process.

3. Problem Statement

3.1. General Consideration. Several aspects should be further explained before introducing the approach.

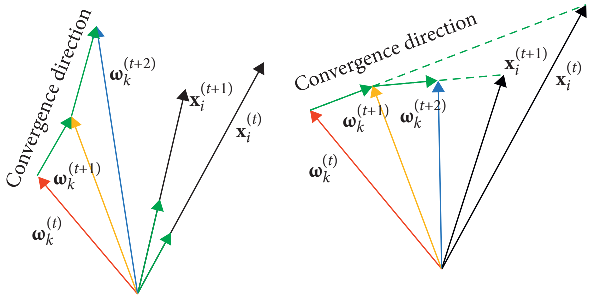
The first aspect is to determine the change in the convergence direction in existing loss functions during the iteration and the impacts of the indeterminate direction on classification results. In loss function, the affinity score (logit) is usually calculated by different metrics, such as inner product and Euclidean distance metrics. These metrics are usually used directly to calculate affinity scores or as part of the process of calculating affinity scores if they are in the form of extra regularization item. This way makes convergence direction depend on the network parameters and training samples, which are changing over each iteration. The indeterminate convergence direction causes difficulty in obtaining a stable convergence center, which indirectly leads to increased errors in the established model.

Here, we use soft-max as an example to illustrate this effect. For a K-classification task, we assume that $\mathbf{x}_i$ and $\boldsymbol{\omega}_k$ are the extracted deep feature vector and the class weight vector for class k, respectively. For inner product metric, the convergence direction is the same as the direction of $\mathbf{x}_i$. For Euclidean distance, the convergence direction, which is reflected as the vector from $\mathbf{x}_i$ to $\boldsymbol{\omega}_k$, is determined by the direction and norm of feature, as shown in Figure 2.

The second aspect is the decoupling of the feature into direction and norm. The feature vector is determined by two characteristics, namely, direction and norm, which are naturally coupled. It is therefore as incomplete as cosine metric when only one of the characteristics is considered during the optimization process. Existing metrics always treat the two characteristics as a whole. Thus, the optimization inevitably involves both of them. This condition may lead to the interference of the two characteristics with each other, which affects the final classification effects.

3.2. Approach Overview. As discussed above, we first need to decompose the convergence direction into two mutually orthogonal directions. To facilitate implementation, we make the feature as the subject of decoupling and decouple it into two components, namely, direction and norm, which correspond to tangential and radial directions, respectively. This step has two advantages. First, we can separately optimize the two components to prevent them from interacting with each other. Second, the relationship between the two components can be explicitly determined. The decomposition makes it convenient to obtain the convergence center because only one component (direction or norm) is taken into the calculation process at a time.

We assume that the two components follow Gaussian distribution to improve the accuracy of the model further. We


Convergence direction
$\boldsymbol{\omega}_k^{(t+2)}$
$\mathbf{x}_i^{(t+1)}$
$\mathbf{x}_i^{(t)}$
$\boldsymbol{\omega}_k^{(t+1)}$
$\boldsymbol{\omega}_k^{(t)}$
Convergence direction
$\mathbf{x}_i^{(t+1)}$
$\mathbf{x}_i^{(t)}$
$\boldsymbol{\omega}_k^{(t+1)}$
$\boldsymbol{\omega}_k^{(t+2)}$
$\boldsymbol{\omega}_k^{(t)}$

(a) (b)

Figure 2: Illustration of convergence direction change in the cases of (a) inner product metric and (b) Euclidean distance metric.

believe that this assumption is reasonable, especially when the overall distribution is unknown and the sample size is sufficiently large.

Figure 3 shows the optimizing process in a classification task, in which $\mathbf{x}_i$ is the extracted deep feature vector for an input sample and $\boldsymbol{\omega}_k$ is the class weight vector for the class k, from which $\mathbf{x}_i$ belongs to. As observed, the convergence direction is static during each iteration.

We design the loss function in the tangential direction to conduct classification. Given that the core purpose of classification is to separate the different classes from each other, the loss function in the tangential direction is mainly responsible for the interclass separation. We adopt the popular method called cross entropy to implement the classification function.

The ability to classify is achieved in tangential loss. Thus, the ability to classify in radial loss need not be obtained. We achieve interclass separation of feature distributions. We design the radial loss to be primarily responsible for intraclass compactness to improve the classification effect further. We achieve the intraclass compactness by reducing the difference between the actual distribution of features and the ideal Gaussian distribution of features. We use a popular method called likelihood function to measure the difference in distribution.

4. GO Loss

In this section, we first introduce the optimization of the tangential and radial components and then give the method for merging the two parts as the GO loss for implementing the joint supervision.

4.1. Optimization on Tangential Direction. In tangential direction, we first provide the formal definition of the Gaussian distribution. Then, we use Bayes' rule to calculate the posterior probability distribution. Finally, we use cross entropy to calculate the classification loss.

4.1.1. Gaussian Distribution. Let $\widehat{x}_i$ be the feature following the Gaussian distribution, as shown in equation (1), where $\widehat{x}_i = \mathbf{x}_i/|\mathbf{x}_i|$ and $\widehat{\omega}_k = \boldsymbol{\omega}_k/|\boldsymbol{\omega}_k|$. $\widehat{\omega}_k$ is the class weight from which $\widehat{x}_i$ corresponds to, and σ_k represents the covariance of class k in the feature space. For unknown K-classification task, we assume that the probability of each class is equal, whose purpose is to ensure that the prior probability is constant. The prior probability of class k is $p(k) = 1/K$. The hyperparameter α is used to control the difficulty in the training process.

$$\begin{aligned} p(\widehat{x}_i) &= \sum_{k=1}^{K} \mathcal{N}(\widehat{x}_i; \widehat{\omega}_k, \sigma_k)\, p(k) \\ &= \sum_{k=1}^{K} \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{\alpha\cdot(\widehat{x}_i-\widehat{\omega}_k)^2}{2\sigma_k^2}\right) p(k). \end{aligned} \tag{1}$$

Our ideal idea is to guarantee that the angle between the feature and its corresponding class weight obeys Gaussian distribution. However, Gaussian distribution of angles is too complicated to calculate. We use the normalized feature and its corresponding class weight vector instead of the cosine of the angle between them to avoid complex angle calculations. According to the cosine theorem, $(\widehat{x}_i - \widehat{\omega}_k)^2$ can be replaced by the cosine of the angle between the feature and its corresponding class center vector. Thus, equation (1) can be understood as a similar Gaussian distribution associated with the angular cosine. It proves the feasibility of the replacement.

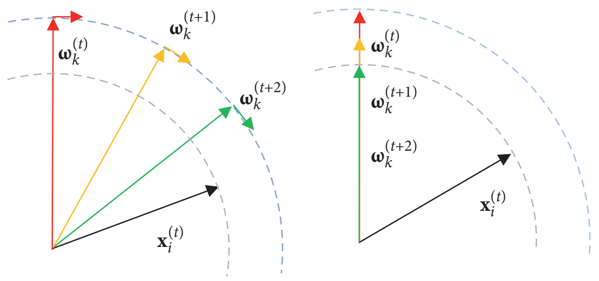
4.1.2. Bayes' Rule. Assume $\widehat{x}_i$ is a normalized feature with the label $z_i \in [1, K]$. Under Gaussian distribution assumption, its conditional probability distribution can be written as

$$p\left(\widehat{x}_i \mid z_i\right) = \mathcal{N}\left(\widehat{x}_i; \widehat{\omega}_{z_i}, \sigma_{z_i}\right). \tag{2}$$

According to Bayes' rule, its posterior probability distribution is

$$p\left(z_i \mid \widehat{x}_i\right) = \frac{\mathcal{N}\left(\widehat{x}_i; \widehat{\omega}_{z_i}, \sigma_{z_i}\right) p(z_i)}{\sum_{k=1}^{K} \mathcal{N}(\widehat{x}_i; \widehat{\omega}_k, \sigma_k)\, p(k)}. \tag{3}$$

4.1.3. Cross-Entropy Loss. We finally use the cross entropy between the posterior probability distribution and the class label to calculate the loss of the tangential direction, which is written as $\mathscr{L}_{\mathrm{T}}$ and defined as


$\boldsymbol{\omega}_k^{(t)}$
$\boldsymbol{\omega}_k^{(t+1)}$
$\boldsymbol{\omega}_k^{(t+2)}$
$\mathbf{x}_i^{(t)}$
$\boldsymbol{\omega}_k^{(t)}$
$\boldsymbol{\omega}_k^{(t+1)}$
$\boldsymbol{\omega}_k^{(t+2)}$
$\mathbf{x}_i^{(t)}$

(a)

(b)

Figure 3: Convergence direction of GO loss, which is fixed in (a) the tangential and (b) radial directions during the optimization process.

$$\begin{aligned}\mathscr{L}_{\mathrm{T}} &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{1}\left(z_i = k\right)\log p\left(k \mid \widehat{x}_i\right)\\ &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\mathcal{N}\left(\widehat{x}_i; \widehat{\omega}_{z_i}, \sigma_{z_i}\right)p\left(z_i\right)}{\sum_{k=1}^{K}\mathcal{N}\left(\widehat{x}_i; \widehat{\omega}_k, \sigma_k\right)p\left(k\right)}.\end{aligned} \tag{4}$$

$\mathbb{1}\,(\,)$ is an indicator function, which defined as

$$\mathbb{1}\left(z_i = k\right) = \begin{cases}1, & z_i = k,\\ 0, & \text{otherwise}.\end{cases} \tag{5}$$

4.2. Optimization on Radial Direction. In radial direction, we first give the formal definition of the Gaussian distribution. Then, we use Bayes' rule to calculate the posterior probability distribution. Finally, we use the likelihood to calculate the loss.

4.2.1. Gaussian Distribution. Similar to that in the tangential direction, we assume that l_2-norm of the feature $\|\mathbf{x}_i\|_2$ on the radial direction also follows the Gaussian distribution, which is defined as

$$p\left(\left\|\mathbf{x}_i\right\|_2\right) = \sum_{k=1}^{K}\mathcal{N}\left(\left\|\mathbf{x}_i\right\|_2; \left\|\boldsymbol{\omega}_k\right\|_2, \Sigma_k\right)p\left(k\right), \tag{6}$$

where $\|\boldsymbol{\omega}_k\|_2$, Σ_k, and $p(k)$ are the l_2-norm values of the class weight vector, the covariance, and the prior probability of class k, respectively. Similar to the Gaussian distribution assumption in the tangential direction, the prior probability is constant. As a result, the prior probability of class k is $p(k) = 1/K$.

4.2.2. Bayes' Rule. Assume $\|\mathbf{x}_i\|_2$ is l_2-norm of feature with the label $z_i \in [1, K]$. Under the Gaussian distribution assumption, its conditional probability distribution can be written as

$$p\left(\left\|\mathbf{x}_i\right\|_2 \middle| z_i\right) = \mathcal{N}\left(\left\|\mathbf{x}_i\right\|_2; \left\|\boldsymbol{\omega}_{z_i}\right\|_2, \Sigma_{z_i}\right). \tag{7}$$

According to Bayes' rule, its posterior probability distribution is

$$p\left(z_i \mid \left\|\mathbf{x}_i\right\|_2\right) = \frac{\mathcal{N}\left(\left\|\mathbf{x}_i\right\|_2; \left\|\boldsymbol{\omega}_{z_i}\right\|_2, \Sigma_{z_i}\right)p\left(z_i\right)}{\sum_{k=1}^{K}\mathcal{N}\left(\left\|\mathbf{x}_i\right\|_2; \left\|\boldsymbol{\omega}_k\right\|_2, \Sigma_k\right)p\left(k\right)}. \tag{8}$$

4.2.3. Likelihood Loss. For a complete dataset $\{X, Z\}$, the likelihood can be expressed as

$$\begin{aligned}&p\left(X, Z \mid \|\boldsymbol{\omega}\|_2, \Sigma\right)\\ &\quad = \prod_{i=1}^{N}\prod_{k=1}^{K} 1\left(z_i = k\right)\mathcal{N}\left(\left\|\mathbf{x}_i\right\|_2; \left\|\boldsymbol{\omega}_{z_i}\right\|_2, \Sigma_{z_i}\right)p\left(z_i\right).\end{aligned} \tag{9}$$

The negative log likelihood can be expressed as

$$\begin{aligned}&\log p\left(X, Z \mid \|\boldsymbol{\omega}\|_2, \Sigma\right)\\ &\quad = -\sum_{i=1}^{N}\left(\log\mathcal{N}\left(\left\|\mathbf{x}_i\right\|_2; \left\|\boldsymbol{\omega}_{z_i}\right\|_2, \Sigma_{z_i}\right) + \log p\left(z_i\right)\right).\end{aligned} \tag{10}$$

According to Gaussian distribution assumption, the prior probability $p(\mathbf{x}_i)$ is a constant and is equal to $1/K$ for K-classification problem. Therefore, the loss on the radial direction, which is written as $\mathscr{L}_{\mathrm{R}}$, can be simplified as

$$\mathscr{L}_{\mathrm{R}} = -\sum_{i=1}^{N}\log\mathcal{N}\left(\left\|\mathbf{x}_i\right\|_2; \left\|\boldsymbol{\omega}_{z_i}\right\|_2, \Sigma_{z_i}\right). \tag{11}$$

4.3. Joint Supervision. We have already obtained the loss functions on tangential and radial directions, namely, $\mathscr{L}_{\mathrm{T}}$ and $\mathscr{L}_{\mathrm{R}}$. In this section, we continue to introduce the merging of the two loss functions to construct the final GO loss.

Assume $\mathscr{L}_{\mathrm{O}}$ is the GO loss, which can be composed of $\mathscr{L}_{\mathrm{T}}$ and $\mathscr{L}_{\mathrm{R}}$, as shown in equation (12). Naturally, $\mathscr{L}_{\mathrm{T}}$ is only related to the cosine of the angle between the feature vector and its corresponding class weight vector, while $\mathscr{L}_{\mathrm{R}}$ is only related to the norm of the feature vector. Without loss of generality, $\mathscr{L}_{\mathrm{T}}$ and $\mathscr{L}_{\mathrm{R}}$ share all the parameters:

$$\begin{aligned}\mathscr{L}_{\mathrm{O}} &= \mathscr{L}_{\mathrm{T}} + \lambda\mathscr{L}_{\mathrm{R}} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\mathscr{N}(\widehat{x}_i;\widehat{\omega}_{zi},\sigma_{zi})p(z_i)}{\sum_{k=1}^{K}\mathscr{N}(\widehat{x}_i;\widehat{\omega}_k,\sigma_k)p(k)} \\ &\quad -\lambda\sum_{i=1}^{N}\log\mathscr{N}\left(\left\|\mathbf{x}_i\right\|_2;\left\|\boldsymbol{\omega}_{z_i}\right\|_2,\Sigma_{zi}\right).\end{aligned} \tag{12}$$

A hyperparameter α is used to control the difficulty in the training process in $\mathscr{L}_{\mathrm{T}}$. A nonnegative weighting coefficient λ is used to balance the two loss functions. If λ is set to 0, then only $\mathscr{L}_{\mathrm{T}}$ is used in the optimization, while $\mathscr{L}_{\mathrm{T}}$ and $\mathscr{L}_{\mathrm{R}}$ will have the same importance when λ is set to 1. The influence of the hyperparameter is investigated in the subsequent experiments.

5. Experiments

5.1. MNIST Datasets. In the first experiment, we compare GO loss with soft-max loss though the MNIST Handwritten Digit dataset [42]. The classification results, which are in the form of high-dimensional vectors, are projected onto a 2D plane, as shown in Figure 1. As observed, features distribute in 300 units of measure by using traditional soft-max loss and in 3 units of measure by using our GO loss. Our GO loss has better intraclass compactness and interclass separability than soft-max loss.

We train the network with different loss functions, namely, soft-max loss, center loss [27], ring loss with soft-max [23], LGM loss [28], GCPL loss [37], and SL [41]. In the aforementioned methods, center loss, ring loss, LGM loss, and GCPL loss optimize the loss function from the perspective of intraclass compactness and interclass separation of features. These methods are consistent with the goal of our GO loss. But SL is a popular method for datasets where the label has noise. We also compare from new optimization perspective. We use SampleNet, which has five convolution layers, each with 32 dimensions, and a fully connected layer with a two-dimensional output. For the existing loss function, we attempt to adjust the hyperparameters and select the best results for recording. The networks are trained with a batch size of 128 for 50 epochs, and the learning rate is set to 0.1 and then divided by 2 for every 20 epochs. The hyperparameter α is set to 20. The classification accuracy in different methods is shown in Table 1. As observed, GO loss has a better performance than other loss functions on MNIST.

Table 1: Classification accuracy on MNIST dataset.

Methods	Remark	Acc. (%)
Soft-max	—	99.28
Center loss [27]	$\lambda = 0.1$	99.62
Ring loss [23]	$\lambda = 0.1$	99.58
LGM loss [28]	$\alpha = 1$	99.36
GCPL loss [37]	$\lambda = 0.1$	99.41
SL [41]	$\eta = 0.0$	99.32
GO loss	$\boldsymbol{\lambda} = \mathbf{0.1}$	**99.66 ± 0.03**

5.2. Parameter Analysis. We also conduct experiments to investigate the influence of the hyperparameter α and λ on the performance. We set α to 10, 20, 30, and 40, each with λ of 0.1 and 0.01. Table 2 shows that the accuracy of GO loss is the highest when α is 20 and λ is 0.1. We therefore use this setting for other experiments.

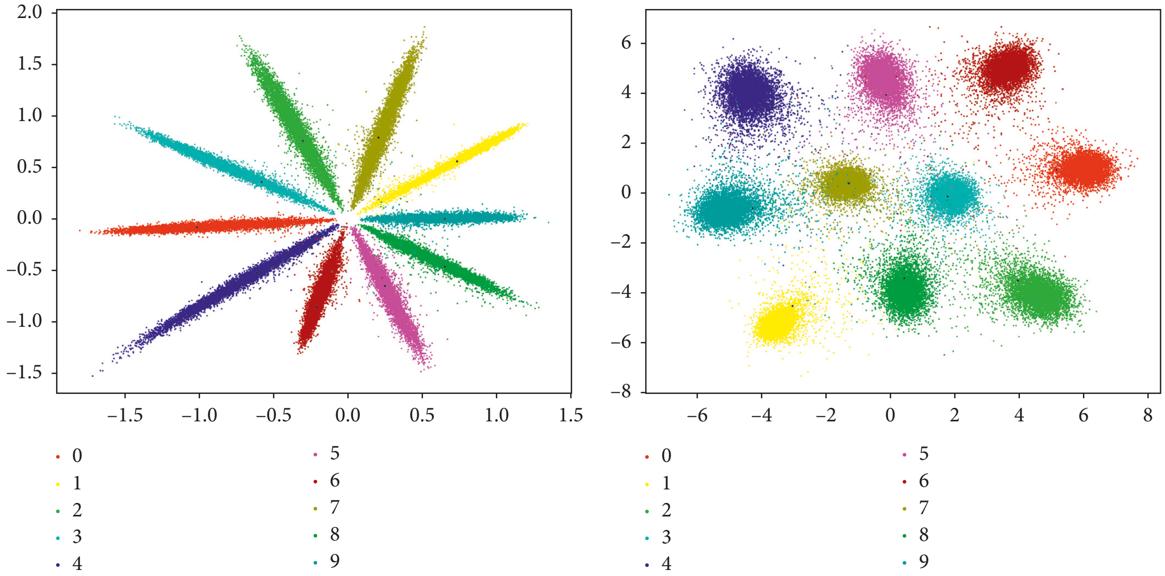
Table 2: Classification accuracy of different hyperparameter on MNIST.

α	λ	Acc. (%)
$\alpha = 10$	$\lambda = 0.1$	99.34
$\alpha = 10$	$\lambda = 0.01$	99.27
$\boldsymbol{\alpha} = \mathbf{20}$	$\boldsymbol{\lambda} = \mathbf{0.1}$	**99.69**
$\alpha = 20$	$\lambda = 0.01$	99.53
$\alpha = 30$	$\lambda = 0.1$	99.17
$\alpha = 30$	$\lambda = 0.01$	99.08
$\alpha = 40$	$\lambda = 0.1$	98.58
$\alpha = 40$	$\lambda = 0.01$	98.23

We determine the effects of tangential and radial losses on the overall GO loss. We set λ to 0, which indicates that only the tangential loss is used in GO loss. Only the radial loss cannot achieve classification. Thus, we set λ to 1, which implies that the radial loss has more evident contribution than the general experimental situation. The classification results, which are in the form of high-dimensional vectors, are projected onto a 2D plane, as shown in Figure 4. The experimental results show that the distance between the features of the same class becomes significantly larger when only the tangential loss is used as the loss function. This result shows that radial loss can effectively control intraclass compactness. When the proportion of radial loss is too large, the different classes of features will be intertwined. This condition results in poor interclass separation. This indicates that tangential loss plays a decisive role in the performance of separation between classes.

Features distribute in 300 units of measure by using traditional soft-max loss in Figure 1(a) and in 2 units of measure by using our tangential loss in Figure 4(a). The shape of their distribution may be similar. Most of the existing loss functions have similar feature distributions in two-dimensional space with soft-max. However, the reason has never been discussed to the best of our knowledge. We analyze the traditional soft-max using the inner product space metric, which is essentially a linear constraint. As a result, feature distribution is linearly separable. Although our tangential loss is calculated by the normalized feature, it is also related to the cosine according to the cosine theorem. The cosine is the inner product of the normalized vector, which is also a linear constraint. Thus, they are similar in the shape of the distribution. The Euclidean distance and our radial loss are quadratic or bilinear constraints. Thus, the features are different, as shown in Figure 4(b).

5.3. Fashion-MNIST Datasets. We conduct another experiment on the Fashion-MNIST dataset [43], which contains 70,000 grayscale images with the pixel resolution of 28 × 28.



(a)

(b)

Figure 4: The distributions of the features trained using GO loss (a) $\lambda = 0$ and (b) $\lambda = 1$ on MNIST. Different colors represent different class. The contribution of radial loss and tangential loss to the overall GO loss is shown. Best viewed in color.

The dataset contains 10 categories of fashion products and is divided into 60, 000 training samples and 10,000 testing samples. We adopt the same network and training parameters with MNIST. The classification accuracy is shown in Table 3. As observed, GO loss also has the best performance on this dataset.

WRN-28 [18] is proven to have the best results on the Fashion-MNIST datasets. We try our GO loss on this network structure. The classification accuracy is shown in Table 4. The experimental results prove that our GO loss also has excellent performance in the advanced network structure.

Table 3: Classification accuracy with SampleNet on Fashion-MNIST dataset.

Methods	Remark	Acc. (%)
Soft-max	—	91.56
Center loss	$\lambda = 0.1$	93.26
Ring loss	$\lambda = 0.1$	93.08
LGM loss	$\alpha = 1$	92.33
GCPL loss	$\lambda = 0.1$	92.65
GO loss	$\boldsymbol{\lambda} = 0.1$	**93.40 ± 0.02**

Table 4: Classification accuracy with WRN-28 on Fashion-MNIST dataset.

Methods	Remark	Acc. (%)
Soft-max	—	94.04
GO loss	$\eta = 0.0$	**94.25 ± 0.02**

5.4. CIFAR-10 and CIFAR-100 Datasets. We use GO loss to implement three more complex networks on CIFAR-10 and CIFAR-100 datasets [44]. Each dataset contains 60,000 colored images, which are divided into 50,000 training images and 10,000 testing images with the pixel resolution of 32×32. The dataset adopts standard data augmentation scheme, which includes mirroring and 32×32 random cropping after 4-pixel zero-paddings on each side [9, 24].

For CIFAR-10, we use ResNet [9] of a depth of 20 as the network structure. The batch size is set to 128 and epoch is 300. We set the learning rate to 0.1, which will become half of the original one for every 60 epochs. The hyperparameter α is set to 20. We use a weight decay of 5×10^{-4} and SGD optimization algorithm with a momentum of 0.9. The method introduced in [45] is used to initialize the network weights. The main purpose of the experiment is to compare the classification accuracy on soft-max loss and GO loss. Moreover, we compare the classification accuracy under different values of balance parameter λ (0.1 and 0.01), which describes the degree of contribution of the loss functions of radial and tangential directions to the final GO loss (Section 4.3). The experimental results are shown in Table 5. As expected, GO loss can achieve better results than traditional soft-max loss.

We use another network, namely DenseNet-BC [1] with 12 feature maps, to observe the performance of GO loss on it for eliminating the deviation in the experimental results caused by the network structure. The experiment is also conducted on the CIFAR-10 dataset. The experimental results shown in Table 6 indicate that GO loss also has a better performance than the others under this experiment condition.

For CIFAR-100, we use ResNet [9] of a depth of 50 as the network structure. The batch size is set to 128 and epoch is 300. We set the learning rate to 0.1, which will be divided by 10 for every 120 epochs. The hyperparameter α is set to 20. We use a weight decay of 5×10^{-4} and SGD optimization algorithm with a momentum of 0.9. The method introduced

Table 5: Classification accuracy with ResNet-20 on CIFAR-10 dataset.

Methods	Remark	Acc. (%)
Soft-max	—	91.35
GO loss	$\lambda = 0.01$	91.59 ± 0.02
GO loss	$\lambda = 0.1$	**91.92 ± 0.03**

Table 6: Classification accuracy with DenseNet-BC on CIFAR-10 dataset.

Methods	Remark	Acc. (%)
Soft-max	—	95.31
GO loss	$\lambda = 0.1$	**95.55 ± 0.13**

Table 7: Classification accuracy with ResNet-50 on CIFAR-100 dataset.

Methods	Remark	Acc. (%)
Soft-max	—	74.35
GO loss	$\lambda = 0.01$	74.75 ± 0.03
GO loss	$\lambda = 0.1$	**75.03 ± 0.06**

in [45] is used to initialize the network weights. The main purpose of the experiment is to compare the classification accuracy on soft-max loss and GO loss. Moreover, we compare the classification accuracy under different values of balance parameter λ (0.1 and 0.01), which describes the degree of contribution of the loss functions of radial and tangential directions to the final GO loss (Section 4.3). The experimental results shown in Table 7 indicate that the GO loss has the best effects when $\lambda = 0.1$. Therefore, tangential direction, which is more related to interclass separability than the other direction, has a greater impact on the classification accuracy.

We use another network, namely, DenseNet-BC with 12 feature maps, to observe the performance of GO loss on it for eliminating the deviation in the experimental results caused by the network structure. The experiment is also performed on the CIFAR-100 dataset. The experimental results in Table 8 indicate that GO loss also has a better performance than the others under this experiment condition.

5.5. ImageNet Dataset. We use ImageNet dataset [46] with a larger size to observe the performance of GO loss on it for verifying the scalability of GO loss. A more complex network, namely, ResNet-101 [9], is used. Soft-max is selected as the reference to compare its classification accuracy with GO loss. We use 8 Titan GPUs to train all the models. The batch size and epoch are set to 128 and 120, respectively. Meanwhile, the learning rate is initialized as 0.01 with a decay rate of 50% every 40 epochs. We also investigate the influence of varying balance parameter λ on the accuracy. The results shown in Table 9 indicate that GO loss is also effective on the large-scale datasets and will achieve a better performance with a larger λ.

Table 8: Classification accuracy with DenseNet-BC on CIFAR-100 dataset.

Methods	Remark	Acc. (%)
Soft-max	—	77.81
GO loss	$\lambda = 0.1$	**78.24 ± 0.02**

Table 9: Classification accuracy on ImageNet dataset.

Methods	Remark	Acc. (%)
Soft-max	—	77.29
GO loss	$\lambda = 0.01$	77.45
GO loss	$\lambda = 0.1$	**77.89**

6. Conclusions

In this paper, we present an orthogonal decomposition-based loss for classification.

Our approach can be summarized as follows:

(1) We propose a new optimization perspective. Specifically, we consider the optimization problem from the perspective of convergence direction.

(2) We decompose the convergence direction into two mutually orthogonal components, namely, tangential and radial directions, and conduct optimization on them separately.

(3) We decouple the direction and norm of feature to avoid their interference with each other during the optimization process.

(4) We use the direction and norm of feature to associate with the interclass separability and intraclass compactness, respectively.

(5) We use Gaussian distribution to guide the optimization processes on direction and norm of feature.

We train six networks on five datasets with different sizes to evaluate the proposed GO loss. The results demonstrate the effectiveness of GO loss. In our future work, we plan to make two improvements. First, we plan to apply GO loss to other datasets for a thorough evaluation of its performance under different application scenarios. Second, we will propose a method to quantitatively determine the value of the hyperparameters, such as by visual analytics [6] or adaptive scaling [47].

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was partly supported by the National Key RD Program of China (2018YFC0831700), the National Natural

Science Foundation of China (61972278, 61602340, and 61572348), the Beijing Intelligent Logistics System Collaborative Innovation Center (BILSCIC-2018KF-04), and the Natural Science Foundation of Hunan Province, China (2016JJ1024).

References

[1] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.

[2] X. Wang, Z. Cui, L. Jiang, W. Lu, and J. Li, “Wordlenet: a visualization approach for relationship exploration in document collection,” *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 384–400, 2020.

[3] V. A. Maksimenko, S. A. Kurkin, E. N. Pitsik et al., “Artificial neural network classification of motor-related EEG: an increase in classification accuracy by reducing signal complexity,” *Complexity*, vol. 2018, Article ID 9385947, 10 pages, 2018.

[4] Z. Cui, X. Zheng, X. Shao, and L. Cui, “Automatic sleep stage classification based on convolutional neural network and fine-grained segments,” *Complexity*, vol. 2018, Article ID 9248410, 13 pages, 2018.

[5] Z. H. Kilimci and S. Akyokus, “Deep learning-and word embedding-based heterogeneous classifier ensembles for text classification,” *Complexity*, vol. 2018, Article ID 7130146, 10 pages, 2018.

[6] J. Li, S. Chen, W. Chen, G. Andrienko, and N. Andrienko, “Semantics-space-time cube. a conceptual framework for systematic analysis of texts in space and time,” *IEEE Transactions on Visualization and Computer Graphics*, p. 1, 2018.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, pp. 1097–1105, Curran Associates, Inc., Red Hook, NY, USA, 2012.

[8] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” 2015, http://arxiv.org/abs/1502.03167.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[10] H. Qi, Z. Wu, S. Deng et al., “Research on face recognition method by autoassociative memory based on rnns,” *Complexity*, vol. 2018, Article ID 8524825, 12 pages, 2018.

[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” 2016, http://arxiv.org/abs/1611.03530.

[12] H. Qiang, C. Dong, and F. Zhang, “A novel approach to face verification based on second-order face-pair representation,” *Complexity*, vol. 2018, Article ID 2861695, 10 pages, 2018.

[13] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[14] G. Hinton, L. Deng, D. Yu et al., “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012.

[15] C. Cui, Z. Feng, and C. Tan, “Credibilistic loss aversion nash equilibrium for bimatrix games with triangular fuzzy payoffs,” *Complexity*, vol. 2018, Article ID 7143586, 16 pages, 2018.

[16] Y. Wen, Z. Li, and Y. Qiao, “Latent factor guided convolutional neural networks for age-invariant face recognition,” in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[17] C. Szegedy, V. Vincent, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[18] S. Zagoruyko and N. Komodakis, “Wide residual networks,” 2016, http://arxiv.org/abs/1605.07146.

[19] G. Larsson, M. Maire, and G. Shakhnarovich, “Fractalnet: ultra-deep neural networks without residuals,” 2016, http://arxiv.org/abs/1605.07648.

[20] H. V. Nguyen and L. Bai, “Cosine similarity metric learning for face verification,” in *Computer Vision–ACCV 2010*, Springer, Berlin, Germany, 2010.

[21] S. Ying, Z. Wen, J. Shi, Y. Peng, J. Peng, and H. Qiao, “Manifold preserving: an intrinsic approach for semi-supervised distance metric learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 2731–2742, 2017.

[22] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[23] Y. Zheng, D. K. Pal, and M. Savvides, “Ring loss: convex feature normalization for face recognition,” 2018, http://arxiv.org/abs/1803.00130.

[24] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” in *Proceedings of the International Conference on Machine Learning*, pp. 507–516, Newyork, NY, USA, 2016.

[25] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., vol. 27, pp. 1988–1996, Curran Associates, Inc., Red Hook, NY, USA, 2014.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: a unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[27] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Computer Vision–ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 499–515, Springer International Publishing, Cham, Switzerland, 2016.

[28] W. Wan, Y. Zhong, T. Li, and J. Chen, “Rethinking feature distribution for loss functions in image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.

[29] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia et al., “Multiclass classification procedure for detecting attacks on MQTT-IoT protocol,” *Complexity*, vol. 2019, Article ID 6516253, 11 pages, 2019.

[30] I. U. Haq, U. Amin, M. Khan, M. Y. Lee, and S. W. Baik, “Personalized movie summarization using deep cnn-assisted facial expression recognition,” *Complexity*, vol. 2019, Article ID 3581419, 10 pages, 2019.

[31] H. Cai, J. Han, Y. Chen et al., “A pervasive approach to EEG-based depression detection,” *Complexity*, vol. 2018, Article ID 5238028, 13 pages, 2018.

[32] M. Guillaumin, J. Verbeek, and C. Schmid, “Is that you? metric learning approaches for face identification,” in *Proceedings of the 2009 IEEE 12th International Conference on*

Computer Vision, pp. 498–505, IEEE, Kyoto, Japan, September 2009.

[33] W. Li, R. Zhao, and X. Wang, “Human reidentification with transferred metric learning,” in *Computer Vision–ACCV 2012*, pp. 31–44, Springer, Berlin, Germany, 2012.

[34] X. Yu, Z. Yu, W. Pang, M. Li, and L. Wu, “An improved emd-based dissimilarity metric for unsupervised linear subspace learning,” *Complexity*, vol. 2018, Article ID 8917393, 24 pages, 2018.

[35] J. Li, S. Chen, K. Zhang, G. Andrienko, and N. Andrienko, “Cope: interactive exploration of co-occurrence patterns in spatial time series,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2554–2567, 2019.

[36] S. Chen, J. Li, G. Andrienko et al., “Supporting story synthesis: bridging the gap between visual analytics and storytelling,” *IEEE Transactions on Visualization and Computer Graphics*, p. 1, 2018.

[37] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, “Robust classification with convolutional prototype learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3474–3482, Salt Lake City, UT, USA, June 2018.

[38] Z. Liang, M. Yang, and C. Wang, “3D graph embedding learning with a structure-aware loss function for point cloud semantic instance segmentation,” 2019, http://arxiv.org/abs/1902.05247.

[39] L. Wu, F. Tian, Y. Xia et al., “Learning to teach with dynamic loss functions,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, pp. 6466–6477, Curran Associates, Inc., Red Hook, NY, USA, 2018.

[40] Z. Zhang and M. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, pp. 8778–8788, Curran Associates, Inc., Red Hook, NY, USA, 2018.

[41] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, “Symmetric cross entropy for robust learning with noisy labels,” 2019, http://arxiv.org/abs/1908.06112.

[42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] X. Han, K. Rasul, and V. Roland, “Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms,” 2017, http://arxiv.org/abs/1708.07747.

[44] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Technical Report TR-2009, University of Toronto, Toronto, Canada, 2009.

[45] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, Santiago, Chile, December 2015.

[46] D. Jia, R. Socher, L. Fei-Fei, W. Dong, K. Li, and Li-J. Li, “Imagenet: a large-scale hierarchical image database,” in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami, FL, USA, June 2009.

[47] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, “Adacos: adaptively scaling cosine logits for effectively learning deep face representations,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10823–10832, Salt Lake City, UT, USA, 2019.

Advances in
Operations Research
Advances in
Decision Sciences
Journal of
Applied Mathematics
The Scientific
World Journal
Journal of
Probability and Statistics
International
Journal of
Mathematics and
Mathematical
Sciences
Journal of
Optimization
International Journal of
Engineering
Mathematics
International Journal of
Analysis
Journal of
Complex Analysis
Advances in
Numerical Analysis
Mathematical Problems
in Engineering
International Journal of
Differential Equations
Discrete Dynamics in
Nature and Society
International Journal of
Stochastic Analysis
Journal of
Mathematics
Journal of
Function Spaces
Abstract and
Applied Analysis
Advances in
Mathematical Physics