IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 28, 2019, accepted December 11, 2019, date of publication December 16, 2019,
date of current version January 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2960065

A New Loss Function for CNN Classifier Based on Predefined Evenly-Distributed Class Centroids

QIUYU ZHU, PENGJU ZHANG, ZHENGYONG WANG, AND XIN YE
School of Communication and Information Engineering, Shanghai University, Shanghai 201900, China
Corresponding author: Qiuyu Zhu (zhuqiuyu@staff.shu.edu.cn)

ABSTRACT With the development of convolutional neural networks (CNNs) in recent years, the network structure has become more and more complex and varied, and has achieved very good results in pattern recognition, image classification, object detection and tracking. For CNNs used for image classification, in addition to the network structure, more and more researches focus on the improvement of the loss function, so as to enlarge the inter-class feature differences, and reduce the intra-class feature variations as soon as possible. Besides the traditional Softmax, typical loss functions include L-Softmax, AM-Softmax, ArcFace, and Center loss, etc. Based on the concept of predefined evenly-distributed class centroids (PEDCC) in CSAE network, this paper proposes a PEDCC-based loss function called PEDCC-Loss, which can make the inter-class distance maximal and intra-class distance small enough in latent feature space. Multiple experiments on image classification and face recognition have proved that our method achieve the best recognition accuracy, and network training is stable and easy to converge. Code is available in https://github.com/ZLeopard/PEDCC-Loss

INDEX TERMS Image classification, Softmax, PEDCC, loss function.

I. INTRODUCTION

In the past few years, convolutional neural networks (CNNs) have brought excellent performance in many areas such as image classification, object detection, and face recognition. CNNs extract features from complex datasets through kinds of convolutional layers and pooling layers, and then linear layer is performed for classification. Due to the powerful feature expression and learning ability of CNNs, we can solve a variety of visual recognition tasks.

In order to address the drawbacks currently faced by CNNs, many researchers have proposed very effective solutions, such as data augmentation, regularization, dropout, batch normalization and various activation functions. The development of the network structure is also very rapid, from the beginning of AlexNet [1] to VGGNet [2], and to the deeper ResNet [3], ResNeXt [4], DenseNet [5] and SEResNet [6], etc. The advantages of CNNs are constantly expanded.

Recent research has gradually extended to the design of loss function to obtain a more distinguishing feature distribution, which means the compactness of intra-class and the discreteness of inter-class as soon as possible. Due to the strong fitting ability of the CNNs, these methods can work well and the accuracy of classification is improved.

Due to the advantages of clear theory, easy training, and good performance, the traditional cross entropy loss function is widely used in image classification. But it is not guaranteed to obtain the optimized feature distribution mentioned above. The contrastive loss [7] and triplet loss [8] were proposed to increase the constraint on features. It can easily train large-scale data sets without being limited by display storage. But the disadvantage is that much attention is paid to local feature, leading to training difficulties and long convergence time. L-Softmax [9] introduces the margin parameter and modifies the original Softmax function decision boundary, which increases the learning difficulty by modifies $\|\boldsymbol{W}\|\,\|\boldsymbol{x}\|\,cos\theta$ to $\|\boldsymbol{W}\|\,\|\boldsymbol{x}\|\,cosm\theta$, alleviating the over-fitting problem, and producing the decision margin to make the distribution more discriminative. AM-Softmax [10] set $\|\boldsymbol{W}\| = \|\boldsymbol{x}\| = 1$, and normalize the last layer weights and output features to reduce the impact of image's resolution difference and quantily difference in data set. Then, the Euclidean feature space is converted into the cosine feature space and $cos(m\theta)$ is changed to $cos\theta - m$, which makes the backpropagation easier. For Center Loss [11],

The associate editor coordinating the review of this manuscript and approving it for publication was Mithun Mukherjee.

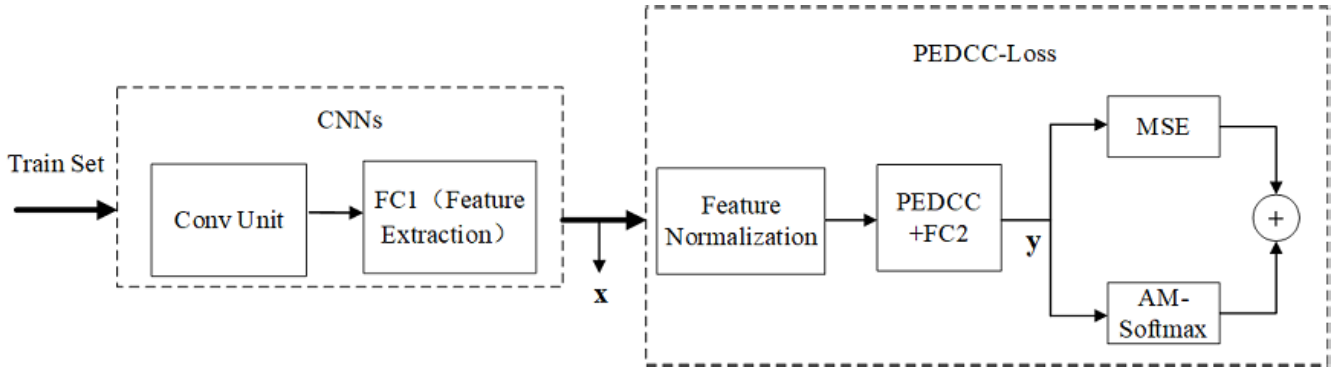

FIGURE 1. The PEDCC-Loss of CNN Classifier. x is output of FC1 layer, and y is output of FC2 layer.

in each batch, a class center is calculated, and the distance between each sample and the class center is minimized. Then, the mean square error combined with the Cross-Entropy loss is proposed, in which the class centers are also trained by stochastic gradient descent. However, the distances of familiar classes are not guaranteed to be well separated. For example, the class centers of class "0" and "6" in MNIST [12] are relatively closer.

In this paper, PEDCC proposed by us in CSAE (Zhu qiuyu *et al.*, 2019) [13] is used to generate the class centroids of the evenly distributed normalized weight, which is called PEDCC weights. We replace the weight of the classification linear layer with PEDCC weights in CNNs, and the PEDCC weights are solidified during training to maximize the inter-class distance. In the same time, we add a constrain similar to Center Loss [11] to calculate the mean square error loss (MSE loss) between the sample feature and PEDCC centroids. This can optimize the feature embedding to enforce higher similarity for intra-class samples and the biggest diversity for inter-class samples. Compared with Center loss [11], the class centroid is fixed, evenly-distributed, and is applied to AM-Softmax loss [10]. The method makes the feature distribution optimal for the compactness of intra-class and the discreteness of inter-class.

The overall system diagram is shown in Fig. 1. Details of the proposed method are given in Section 3.

Our main contributions are as follows:

1) The PEDCC proposed by us in CSAE for Auto-Encoder [13] is applied to image classification and metric learning, and used as weight parameter to solidify the classification layer in the convolutional neural network. By PEDCC the distribution of class centers is optimized, that is, inter-class distance is maximized for class balanced samples.
2) PEDCC weights are applied to AM-Softmax [10] loss, and the improved MSE loss between the feature vector and the predefined class centroid is calculated. The weighted sum of the two losses forms the PEDCC-Loss. In view of the imbalance of the sample number of different classes, an optional finetuning trick is adopted to further improve the accuracy of classification.
3) For the image recognition and face recognition tasks, multiple datasets (EMNIST [14], CIFAR100 [15], FaceScrub [16] and LFW [17]) are evaluated. Compared with the latest research work, our method achieves the optimal recognition accuracy, and network training is stable and easy to converge.

II. RELATED WORK

There are various loss functions in CNNs. Traditional loss functions include Hinge loss, Contrastive loss [7], Triplet loss [8], and the most commonly used Softmax loss function. But the Softmax loss is not good at reducing the intra-class variation. To address this problem, L-Softmax (9) introduce the margin parameter to multiply the angle between the classes in order to increase learning difficulty. However, due to the $cos(m\theta)$, the training is difficult to converge. A-Softmax [18] introduced an conceptually appealing angular margin to push the classification boundary closer to the weight vector of each class. AM-Softmax [10] and CosFace [19] are also directly adds cosine margin penalty to the target logit, which obtains better performance compared to A-Softmax [18], and easier to implement and converge. ArcFace [20] moved cosine margin to the angular margin by changing $cos\theta - m$ to $cos(\theta + m)$, and also discuss the impact of different decision boundaries. But it is found by our experiments that there is no good universality to apply to different classification tasks. Center loss [11] innovatively discussed the distance between each sample and the class center. The mean square error combined with the cross entropy was added to compress the intra-class distance. however, the center of each class is continuously optimized during the training process.

Let us review the Softmax loss, which can be expressed as follows:

$$L_{softmax} = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \tag{1}$$

where f_j is the jth element of the class output vector of the final fully connected layer, and N is the number of training samples. Since f_{y_i} is expressed as $f_{yi} = \boldsymbol{W}_{y_i}^T \boldsymbol{x}_i$, where $\boldsymbol{x}_i$ is the ith input feature, y_i is its label, and $\boldsymbol{W}_{y_i}^T$ is the corresponding network weight. The final loss function can be

written as:

$$L_i = -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i})}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_j)}}\right) \tag{2}$$

For two-classes classification, the purpose of the initial Softmax is to make $\boldsymbol{W}_1^T\boldsymbol{x} > \boldsymbol{W}_2^T\boldsymbol{x}$, that is $\|\boldsymbol{W}_1\|\|\boldsymbol{x}\|cos(\theta_1) > \|\boldsymbol{W}_2\|\|\boldsymbol{x}\|cos(\theta_2)$, which gives the correct classification result for sample $\boldsymbol{x}$ (from class 1). The motivation of L-Softmax loss [9] is to generate a decision margin by adding a positive integer variable m, which can constrain the above inequalities more strictly. As following:

$$\|\boldsymbol{W}_1\|\|\boldsymbol{x}\|\cos\theta_1 \geq \|\boldsymbol{W}_1\|\|\boldsymbol{x}\|\cos m\theta_1 > \|\boldsymbol{W}_2\|\|\boldsymbol{x}\|\cos\theta_2 \tag{3}$$

where $0 \leq \theta_1 \leq \frac{\pi}{m}$.

AM-Softmax [10] rewrites the equation of $cos(\theta)$ to: $\psi(\theta) = cos(\theta) - m$. The above formula is simpler than the $\psi(\theta)$ of L-Softmax [9] in form and calculation. In addition, based on L-Softmax [9], a constraint is added: $b = 0$, $\|\boldsymbol{W}\| = 1$. Compared with L-Softmax loss [9], the difference between the classes is only related to the angle of θ, and m is angular margin. So, after the normalization of weights and input features, the loss function is expressed as:

$$L_{AM} = -\frac{1}{N}\sum_i \log\frac{e^{s\cdot(\cos\theta_{y_i}-m)}}{e^{s\cdot(\cos\theta_{y_i}-m)} + \sum_{j=1,j\neq y_i}^{c} e^{s\cdot\cos\theta_j}} \tag{4}$$

Center loss [11] calculates the class center of several samples of each class in each batch, and then calculates the MSE loss between each sample and the class center.

$$L_C = \frac{1}{2}\sum_{i=1}^{M}\|\boldsymbol{x}_i - c_{y_i}\|^2 \tag{5}$$

where c_{y_i} represents the center calculated by the y_i class, and M is sample number. Finally, the joint loss function is $L = L_{softmax} + \lambda L_C$.

In this paper, PEDCC is used to generate an evenly distributed class centroids, replacing the center calculated in Center loss [11], and MSE loss is used to further reduce the distance between the sample and the class center. Secondly, the fixed and evenly distributed PEDCC weights are directly used as the classification layer weights, and are not updated during training. Finally, the two losses are combined and optimized simultaneously, thus achieving the theoretical optimal distribution. For subsequent comparision, the four main loss functions is drawn as block diagrams as shown in Fig. 2.

We performed the visulization of features x after FC1 for the MNIST [12] dataset, and compare various loss functions in Pytorch 1.0 [21] to show the feature distribution of two-dimensional space and three-dimensional space, respectively, with epochs of 30, as shown in Fig. 3.

It can be seen that, in the Euclidean space, the PEDCC-loss is distributed on the 2-D or 3-D spherical surface, and each cluster is evenly distributed and compact. Comparatively, Center loss [11] randomly clustered in the feature space,

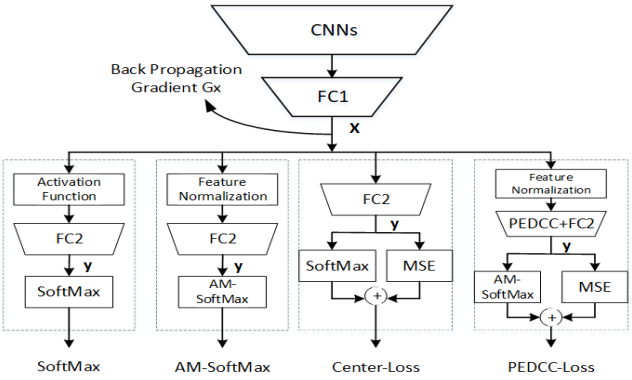

FIGURE 2. Block diagrams of network structures for four losses.

which reduces the inter-class distance, and AM-Softmax [10] has similar result. In the cosine space, PEDCC-loss can not only use margin to separate the inter-class space, but also make the clusters more evenly-distributed, and the sample is closer to the predefined class center.

III. PROPOSED METHOD

A. THE DISTANCE OF INTRA-CLASS AND INTER-CLASS

From the perspective of statistical pattern recognition and image classification, the original image is understood as high-dimensional features. Through some traditional machine learning methods, dimensionality reduction is performed on high-dimensional features. The main goal of dimensionality reduction is to generate low-dimensional expressions with higher similarity for intra-class samples and high diversity for inter-class samples, such as the classic LDA method. Finally, we usually use the Euclidean distance for image recognition, or the cosine distance for face recognition to classify the samples.

Suppose that the sample class in the data set is w_i ($i = 1, 2, 3 \ldots c$) and c is the total number of classes. $\boldsymbol{x}_i$ represents the sample feature vectors in the class w_i. So we can get the mean $\boldsymbol{\mu}_i$ of class w_i and the mean $\boldsymbol{\mu}$ of all samples as following:

The distance of inter-class is

$$D_{inter} = \sum_{i=1}^{c} P_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T(\boldsymbol{\mu}_i - \boldsymbol{\mu}) \tag{6}$$

Thus, for the distance of inter-class, if the number of samples is class balanced and the features are nomalized, the distance of inter-class is maximized only if all of the $\boldsymbol{\mu}_i$ are evenly distributed on the feature supersphere.

The distance of intra-class is

$$D_{intra} = \sum_{i=1}^{c} P_i E_i(\boldsymbol{x} - \boldsymbol{\mu}_i)^T(\boldsymbol{x} - \boldsymbol{\mu}_i) \tag{7}$$

where c is the number of classes, P_i is the prior probability of class w_i, $\boldsymbol{\mu}_i$ is the mean of samples of class w_i and $\boldsymbol{\mu}_i = E_i[\boldsymbol{x}]$. $\boldsymbol{\mu}$ is the mean of all samples and $\boldsymbol{\mu} = E[\boldsymbol{x}]$.

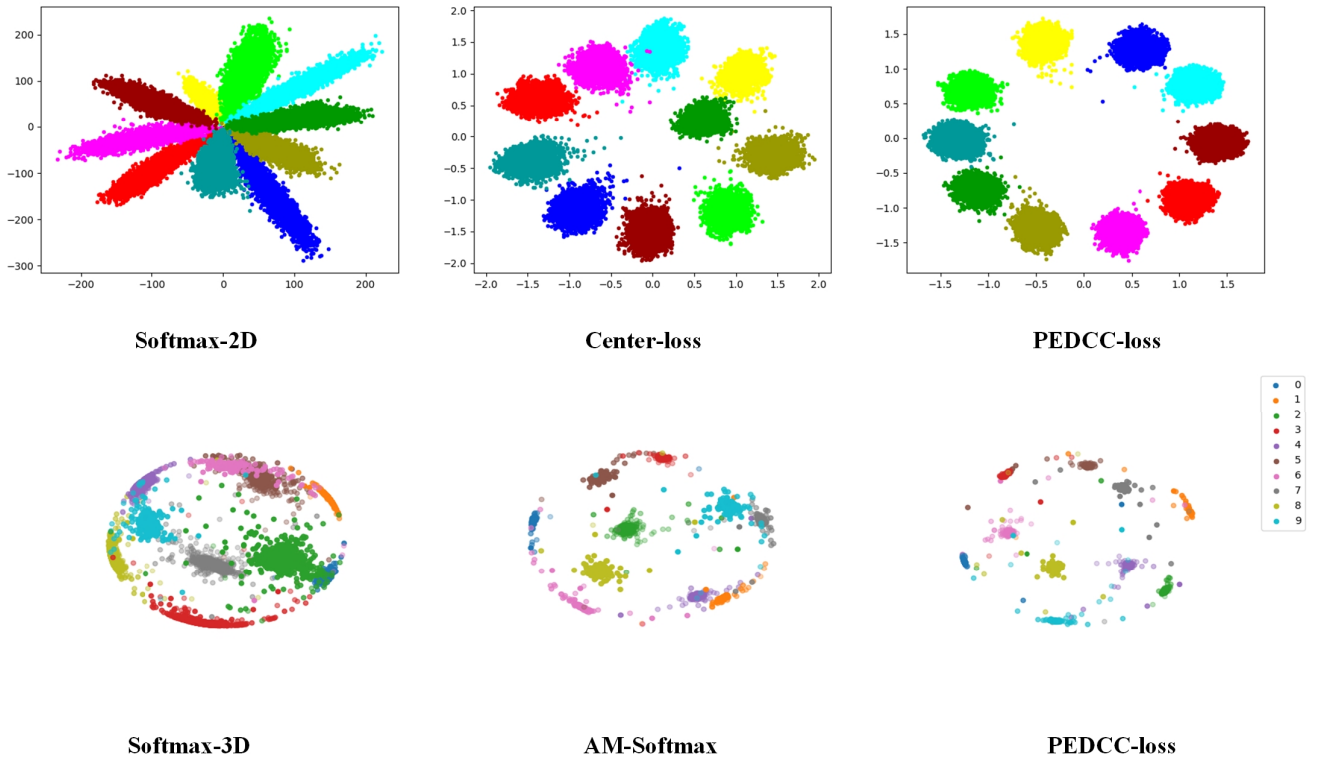

FIGURE 3. Visualization of features x after FC1 for different methods in 2-D and 3-D space.

L-Softmax [9] introduces the concept of margin to increase the difficulty of learning, which is more concerned with inter-class distances than traditional Cross Entropy loss. After adding margin, the distribution of each class becomes slender, which improves the gap between classes. In some visual tasks, such as image classification, it also improves the recognition accuracy, but the intra-class distance is not minimal.

Center Loss [11] learns a center for deep features of each class and penalizes the distances between the deep features and their corresponding class centers. Increasing the degree of aggregation within a class is not difficult for a neural network with powerful learning ability. However, Increasing the inter-class distance is a difficult problem. Different classification tasks may have different distances, and the distance between classes is relatively close. If the intra-class distance is large, there will be overlap between class samples. This leads to misclassification, and there is currently no effective way to avoid this problem.

This paper creatively makes use of predefined evenly distribution class centroids, which makes the distance of inter-class fixed and separated from each other maximally, and simultaneously forces the samples to close to the predefined center as soon as possible.

B. PEDCC

In this paper, by pre-defined the optimal clustering center, the clustering centers of the classes are artificially set, and these clustering centers are evenly distributed on the hypersphere surface of the feature space, so that the inter-class distance is maximized.

PEDCC[13] is actually a class center $\boldsymbol{\mu}_i$ in formula 6,7. Because PEDCC is optimal uniformly distributed, the mean of the class center is 0, as $\boldsymbol{\mu} = 0$. So PEDCC-Loss makes D_{intra} smaller and D_{inter} maximal by adding constraints.

In this way, we learn a mapping function through CNNs, and map different classes samples to the center of these predefined classes, then to cluster them. So that the distances between different classes can be separated maximally.

The method of generating the predefined class center is based on the physical model with the lowest isotropic charge energy on the sphere, that is, it is assumed that the n charge points on the hypersphere have repulsive force with each other, and the repulsive force decreases as the distance between the points increases. At the end of the movement, the point on the hypersphere stops moving. Due to the repulsive force, the n points will be evenly distributed on the hypersphere. When the equilibrium state is reached, the n points on the hypersphere are farthest apart. The detail of algorithm implementation is visible in [13].

In [13], the sample distribution of PEDCC is visualized in 3D space.

Since PEDCC is randomly generated, does different PEDCC affect the final results of network training? Assuming that there are two arbitrary PEDCC matrices, P_1 and P_2, whose size is MN, where M is the input dimension and N is the output dimension (class number), the relationship

between them is expressed as $P_1 = RTP_2$, where R is a rotation matrix and T is a permutation matrix. In theory, this rotation and permutation matrix are absorbed into the first linear layer $FC1$ during the network training. Therefore, consistent results are obtained for the same latent features. Our experiments also prove that different PEDCC does not affect the final network recognition results.

C. PEDCC-LOSS

The previous section gives the concept of inter-class distance and intra-class distance in pattern recognition, which is very important in traditional machine learning and deep convolutional neural networks. The essence of machine learning is to learn good feature distribution, and PEDCC gives the theoretically optimal distribution of cluster centers. Therefore, based on the above two concepts, this section will give a new loss function called PEDCC-Loss for CNNs.

The classification layer parameters in the traditional CNNs are trained with the overall network, and the weights are updated using back propagation by minimalize the loss. In the Euclidean space, the score of each sample is calculated by the formula $s_{y_i} = \|\boldsymbol{W}_{y_i}\| \|\boldsymbol{x}_i\| cos(\theta_{y_i})$. Then we convert scores to probabilities by Softmax function, and obtain the result of classification.

Because the sample numbers of each class and the quality of the image samples may be different in the dataset, the weight vector $\boldsymbol{W}$ is different too. the visualization of $\boldsymbol{W}$ is the vector from origin to the class center, and the visualization of x is the vector from origin to the point with each different color (corresponding to different classes, see Softmax 2D in Fig. 2). Then, the classification layer weight is actually the vector trained by CNNs with sufficient discriminative ability.

The PEDCC is artificially given a plurality of evenly distributed class centers, which are evenly distributed sample points on the unit hypersphere, or a plurality of evenly scattered vectors. Therefore, the global optimal solution of the objective function of the classification layer of CNNs is essentially to obtain a plurality of scattered vectors with sufficient discrimination. We replace the last linear layer's weight of the convolutional neural network with the predefined class-centered (PEDCC weight), and during the training phase, only the weights of previous layers are updated.

At the end of training phase, in order to obtain better recognition performance, depending on different dataset, a fine-tuning processing of the PEDCC weight of the last linear classification layer are adopted optionally. PEDCC-loss are given as following:

$$L_{PEDCC\text{-}AM} = -\frac{1}{N}\sum_i \log \frac{e^{s\cdot(\cos\theta_{y_i}-m)}}{e^{s\cdot(\cos\theta_{y_i}-m)}+\sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_j}} \tag{8}$$

$$L_{PEDCC-MSE} = \frac{1}{2}\sum_{i=1}^{N}\left\|\boldsymbol{x}_i - pedcc_{y_i}\right\|^2 = \frac{1}{2}\sum_{i=1}^{N}(\|\boldsymbol{x}_i\|^2 + \left\|pedcc_{y_i}\right\|^2 - 2\boldsymbol{x}_i \cdot pedcc_{y_i}) = \sum_{i=1}^{N}(1-\cos\theta_{y_i}) \tag{9}$$

$$L_{PEDCC-Loss} = L_{PEDCC\text{-}AM} + \lambda\sqrt[n]{L_{PEDCC\text{-}MSE}} \tag{10}$$

where s and m follow the setting of [20], N is sample number, λ is a weighted coefficient and $n \geq 1$ is a constrain factor of the $L_{PEDCC-MSE}$. On the unit hypersphere, the distance from the sample to the predefined class center is less than 1. A constraint factor n is added to the MSE to increase the difficulty of reducing the intra-class distance, which can increase the recognition performance on certain values for some experiments.

It is noted that the normalized weights in $L_{PEDCC\text{-}AM}$ are PEDCC weights, while the normalized weights of L_{AM} are gradually obtained by training. Although the formulas are the same, they represent different meanings. $L_{PEDCC-MSE}$ is also different from Center-Loss. The class centers in $L_{PEDCC-MSE}$ are PEDCC whose distribution is predefined and optimal, while those in Center-Loss are gradually updated in each minibatch.

D. CHARACTERISTIC ANALYSIS

In addition to the analysis from the perspective of intra-class distance and inter-class distance, in order to explain the function of PEDCC-Loss further, we explore it from the perspective of the back propagation gradient. Although theoretically analysis of AM-Softmax and the gradient of MSE is relatively simple, we can't directly compare the back propagation gradient at FC2 output due to the different network weight. In order to compare different losses, we separate feature extraction layer and loss function layer, and obtain gradient in feature extraction layer FC1 using Pytorch hook function. For any loss, the network of CNNs and FC1 layer is the same. The network structures of Softmax, AM-Softmax, Center-Loss and PEDCC-Loss are shown in Fig. 2.

Although the network performance is not completely related to the magnitude of the gradients, it can still explain the performance of the network to some extent. In the comparative experiments tested on CIFAR100 dataset, the gradients of four different loss functions, Softmax-Loss, Center-Loss with $\lambda = 1$, PEDCC-AM with $s = 15$ and $m = 0.5$ and PEDCC-AM-MSE with $s = 15$, $m = 0.5$, and $\lambda = 10$, are obtained at the output of FC1, and the curves are drawn as Fig. 4. As we see in the Fig. 4(b), the gradient of FC1 output x is much larger than that of FC2 output $\boldsymbol{y}$, which indicates that FC2 magnifies the gradient of backward propogation for Softmax.

For these functions, Softmax is used mainly to increase inter-class distance, and also can reduce intra-class distance to a centain extent. While MSE loss is used to decrease intra-class distance. Compared with simple Softmax, Center-Loss with $\lambda = 1$ greatly increase the gradient

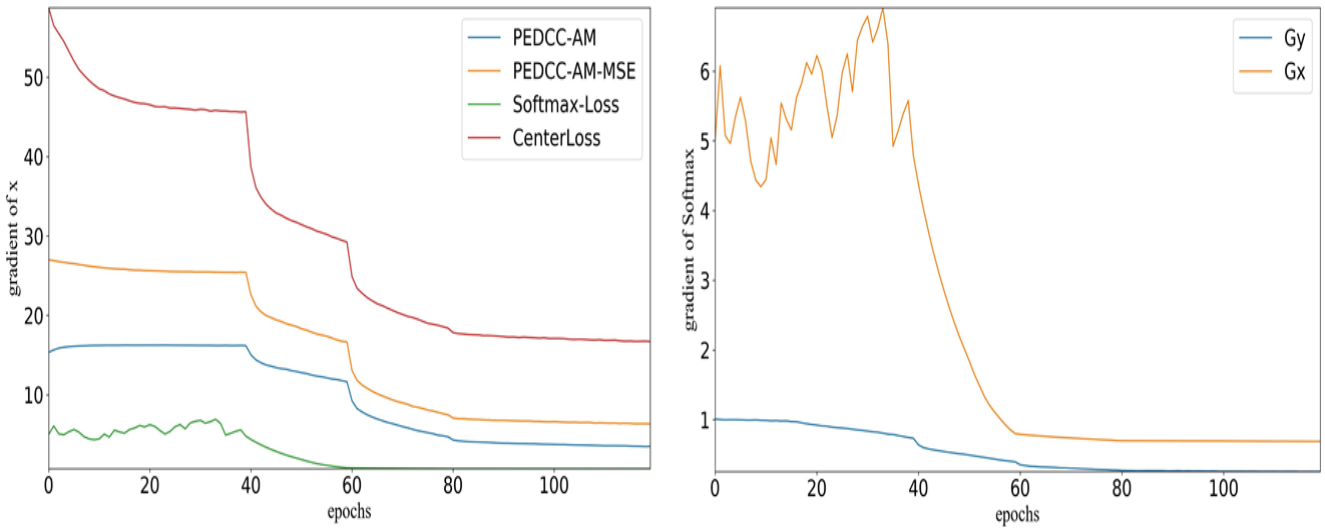

FIGURE 4. (a): The gradients curve of G_x in Softmax, Center-Loss, PEDCC-AM, PEDCC-AM-MSE. (b): The gradinets curve of G_x and G_y in Softmax.

of back propagation in Fig. 4. The curve shows that MSE accounts for the majority of the total gradient, and the performance improvement depends mainly on MSE. This is because the Loss value of MSE is proportional to the square of feature $\boldsymbol{y}$, and the norm of feature $\boldsymbol{y}$ usually is relatively large. In our experiments, it is about 6 at the beginning of network training for Center-Loss.

In order to further reduce the distance of intra-classes, AM-Softmax improves the role of Softmax through feature normalization, scaling factor s and additive margin m. s is designed to compensate for the small feature values caused by feature normalization. Therefore, the role of reducing intra-class distance is mainly realized by m. As can be seen from figure, the gradient of AM-Softmax is much larger than that of Softmax, which is a main reason for the performance improvement of AM-Softmax.

In view of the above analysis, we propose to combine PEDCC, AM-Softmax, and MSE in our PEDCC-Loss. Since PEDCC weights correspond to the optimal clustering centers of samples, we regard PEDCC weights as the weights of FC2. Therefore, we use the MSE of samples and PEDCC weights, AM-Softmax loss to further optimize the design, which greatly reduces the intra-class distance. Thus, the classification accuracy can be further improved. From the gradient point of view, when MSE component is added to PEDCC-AM loss, the gradient range changes greatly, that is to say, the ability of back propagation is enhanced.

In practical experiments, we find that the optimal weight of MSE is 10, so that AM-Softmax and MSE have similar effects on the back-propagation gradient and achieve the best recognition accuracy.

IV. EXPERIMENT RESULT

A. IMPLEMENTATION DETAILS

Our experiment is implemented using Pytorch 1.0 [21], which performs image classification and face recognition tasks respectively. Different PEDCC normalized weights are generated according to the number of dataset classes. The network structure of the image classification is the same as [9] where VGG [2] is used, and the batchsize is 256. The network structure of face recognition is the same as [20] where ResNet18 (IRBlock) [3] with 512 features is used, and the batchsize is 128. During the training phase, the initial learning rate is 0.1, the weight decay is 0.0005, and the momentum is 0.9. The SGD training algorithm is used for both models for parameters in PEDCC-Loss, $s = 15$ and $\lambda = 10$ in all experiments.

Since the number of samples in each class in different datasets may be unbalanced, and different classes may have a slightly different clustering propertied, resulting in that a fixed PEDCC weight will not reach the globally optimal state. We allow PEDCC weights to be fine-tuned within a certain range, that is, a PEDCC weight is set with a very small learning rate to fine-tune the class center after a certain training epoch. In this paper, the training epochs are 120, so we begin to finetune the PEDCC weight with learning rate 1e-3 at epoch 70 to obtain an globally optimal distribution.

B. IMAGE CLASSIFICATION TASKS

In the image classification task, the EMNIST [14] dataset is firstly used. The data set has six division methods: ByClass, ByMerge, Balanced, Letters, Digits, and MNIST. We use the Balanced data set for training. The data set has a total of 131,600 characters pictures, and are evenly divided into 47 classes, each class with 2800 characters. The experimental results are shown in Table 1.

Then, we used the more representative CIFAR100 [15] dataset for test, which has 100 natural images, 500 training sets, and 100 test sets. For this dataset, standard data augmentation [9] is performed, that is, the training set image is padding 4 pixels, and then be randomly clipped to 32×32. The 0.5 probability horizontal flip is also performed, while

TABLE 1. Accuracy with various loss function in EMNIST.

Loss Function	Accuracy
Hinge Loss	88.22
Cross Entropy Loss	88.42
L-Softmax (m=2)	88.69
L-Softmax (m=4)	88.81
A-Softmax (m=4)	88.83
Center Loss	89.21
AM-Softmax (m=0.5)	89.45
ArcFace (m=0.5)	**89.52**
PEDCC-Loss(m=0.5 n=1)	89.60
PEDCC-Loss -finetuning(m=0.5 n=1)	**89.83**
PEDCC-Loss (m=0.5 n=2)	89.47
PEDCC-Loss -finetuning(m=0.5 n=2)	89.66
PEDCC-Loss (m=0.5 n=3)	89.51
PEDCC-Loss -finetuning(m=0.5 n=3)	89.73

TABLE 2. Accuracy with various loss function in CIFAR100.

Loss Function	Accuracy
Hinge Loss	67.10
Cross Entropy Loss	69.02
L-Softmax (m=2)	70.05
L-Softmax (m=4)	70.47
A-Softmax (m=4)	70.86
Center Loss	71.01
AM-Softmax (m=0.5)	71.43
ArcFace (m=0.5)	**71.76**
PEDCC-Loss(m=0.5 n=1)	72.71
PEDCC-Loss -finetuning(m=0.5 n=1)	72.66
PEDCC-Loss (m=0.5 n=2)	73.03
PEDCC-Loss -finetuning(m=0.5 n=2)	**73.23**
PEDCC-Loss (m=0.5 n=3)	71.59
PEDCC-Loss -finetuning(m=0.5 n=3)	71.89

the test set is not processed. The test results are shown in Table 2. Experimental results show that PEDCC-Loss has the similar convergence speed as other loss functions, but the recognition accuracy is higher.

In CIFAR100 [15], our method predefines 100 classes of 512-dimensional class centers distributed on the hypersphere. After the parameters are solidified, the loss of the training set is also lower than the AM-Softmax [10] of the same parameter. This shows the effectiveness of our method, and the addition of PEDCC-MSE further compresses intra-class distance, and in terms of accuracy, PEDCC-loss also obtains the best results in the classification.

C. FACE RECOGNITION TASKS

In the test phase of face recognition, we only use the network to generate face features and calculate cosine distance between two faces. If the distance is small enough, they are belong to the same person, otherwise, the different persons. In other words, PEDCC-Loss will enable the network to learn how to identify the similarity between two faces. The distance between the sample and fixed class center is only considered in training phase.

TABLE 3. Accuracy with various loss function in LFW.

Loss Function	Accuracy
Cross Entropy Loss	91.07
L-Softmax (m=4)	91.22
A-Softmax (m=4)	91.74
Center Loss	92.21
AM-Softmax (m=0.5)	**92.85**
ArcFace (m=0.5)	92.56
PEDCC-Loss(m=0.5 n=1)	91.43
PEDCC-Loss -finetuning(m=0.5 n=1)	92.89
PEDCC-Loss (m=0.5 n=2)	92.04
PEDCC-Loss -finetuning(m=0.5 n=2)	**93.36**
PEDCC-Loss (m=0.5 n=3)	91.27
PEDCC-Loss -finetuning(m=0.5 n=3)	92.76

After L-Softamx [9], many studies have focused on the loss function of face recognition, because face recognition pays more attention to the validity of the feature vector, and the increase of the number of classes can better reflect the validity of the loss function. Here we train ResNet18 for the FaceScrub [16] dataset, which contains more than 100,000 face-aligned images for 530 people, with 265 for men and women. After training the model, the 512-dimensional feature vector extracted are used to test the LFW [17] dataset. The training picture size is 144×144, which is randomly clipped to 128×128, and flipped by the same 0.5 probability level. The number of test faces for LFW [15] is 6000 pairs.

Through the above experiments, we can know that, compared with the weight of random initialization, the PEDCC weight proposed can get a better weight distribution result and make the model more precise, and a nonlinearity factor added to the MSE also can increase the accuracy. Due to the imbalance in the number of samples of various classes, the fixed PEDCC weights are not usually optimal. So, by using the finetuning strategy, we can see that its accuracy has been effectively improved.

V. CONCLUSION

We propose a new loss function based on predefined evenly distributed class centroids for convolutional neural networks. The fixed PEDCC weights are substituted for the parameters of the classification layer in the network, and the improved cross entropy loss is combined with the mean square error of the predefined class center, where a nonlinearity factor is also added to the MSE to increase the learning difficulty. Experimental results show that PEDCC-Loss achieves the best results in image classification and face recognition tasks, and network training is stable and easy to converge.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, May 2017.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec. 2015, *arXiv:1512.03385*. [Online]. Available: https://arxiv.org/abs/1512.03385
[4] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5987–5995.
[5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.
[6] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," Sep. 2017, *arXiv:1709.01507*. [Online]. Available: https://arxiv.org/abs/1709.01507
[7] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.
[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[9] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 507–516.
[10] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.
[11] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Computer Vision–ECCV*, vol. 9911, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 499–515.
[12] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the Web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[13] Q. Zhu and R. Zhang, "A classification supervised auto-encoder based on predefined evenly-distributed class centroids," Feb. 2019, *arXiv:1902.00220*. [Online]. Available: https://arxiv.org/abs/1902.00220
[14] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: An extension of MNIST to handwritten letters," Feb. 2017, *arXiv:1702.05373*. [Online]. Available: https://arxiv.org/abs/1702.05373
[15] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep. 2009.
[16] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 343–347.
[17] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," vol. 15, 2008.
[18] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," Apr. 2017, *arXiv:1704.08063*. [Online]. Available: https://arxiv.org/abs/1704.08063
[19] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 5265–5274.
[20] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," Jan. 2018, *arXiv:1801.07698*. [Online]. Available: https://arxiv.org/abs/1801.07698
[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS*, Oct. 2017, p. 4.

QIUYU ZHU received the bachelor's degree from Fudan University, in 1985, the master's degree from the Shanghai University of Science and Technology, in 1988, and the Ph.D. degree in information and communication engineering from Shanghai University. He is currently a Professor with Shanghai University. His research interests include image processing, computer vision, machine learning, smart city, and computer application. He is a coauthor of approximately 100 academic articles, and principal investigator for more than 10 governmental-funded research projects, more than 30 industrial research projects, many of which have been widely applied.

PENGJU ZHANG received the bachelor's degree in engineering from Henan Polytechnic University (HPU), in 2018. He is currently pursuing the master's degree with the School of Communication, Shanghai University.

His research interests major in computer vision and pattern recognition. During the postgraduate study, he has participated in projects such as Face Detection and Recognition, and Small Object Detection.

ZHENGYONG WANG received the B.S. degree in communication engineering from Yangzhou University, Jiangsu, China, in 2018. He is currently pursuing the master's degree in communication engineering with Shanghai University, Shanghai, China. From 2014 to 2018, he was an undergraduate student at Yangzhou University. His research interests include the development of computer vision, machine learning, and pattern recognition. Now, as a graduate student at Shanghai University, he has been researching in the fields of image processing, pattern recognition, smart city, and computer application.

XIN YE received the bachelor's degree in engineering from the Information Engineering College, Zhejiang University of Technology, in 2012. He is currently with the School of Communication, Shanghai University.

His research directions are computer vision and pattern recognition. During the postgraduate study, he has participated in projects such as face recognition and published a number of EI conference papers. He is mainly involved in the study of incremental learning. He won the National First Prize in the electronic design competition during his undergraduate course.