Face recognition based on adaptive margin and diversity regularization constraints

Zhemin Zhang1 | Xun Gong1 | Junzhou Chen2

1 Southwest Jiaotong University School of Information Science and Technology, Chengdu, China
2 Sun Yat-sen University School of Intelligent Systems Engineering, Guangzhou, China

Correspondence
Xun Gong, Southwest Jiaotong University, School of Information Science and Technology, Chengdu, 610031, China.
Email: xgong@swjtu.edu.cn

Abstract
In recent years, a more robust facial feature can be learned by convolutional neural networks once introducing margins into loss functions. Those methods set a margin for each class manually to squeeze the intra-class variations within each class equally. However, the internal feature distributions of different persons in the real world are highly unbalanced, and the distance between different identities is not uniform either. As a result, applying the same margin on all classes might not lead to higher inter-class differences. To address this problem, this paper proposes an adaptive margin based on feature distribution to squeeze the feature interior spaces of different classes. Simultaneously, because the inter-class margin can adequately represent the distribution of different classes in the feature space, this paper proposes a novel diversity regularization method. The regularization weights of each class are dynamically set depending on their margins. This method proposed in this paper is intuitively interpretable and can be easily applied to other classification scenarios. Experiments on current existing benchmarks have demonstrated the superiority of our method over state-of-the-art competitors.

1 | INTRODUCTION

As one of the most common computer vision tasks, face recognition has made significant progress in recent years [1–3]. Face recognition is essentially a small sample size problem. Therefore, intra-class compactness and inter-class separability are two critical factors for designing a good face feature extraction algorithm. Before deep learning was widely used in face recognition, much work has been done on improving the discrimination of feature extraction or classifiers, such as Fisher linear discriminant analysis (Fisherface) [4] and support vector machine [5]. The development in this area shows that large-scale training data is significant to improve the accuracy of face recognition in the wild. However, it is difficult for traditional algorithms to handle large-scale face data.

In recent years, convolutional neural networks (CNNs) have gradually become the prioritized choice for face feature extraction, and recognition performance of face recognition has also been greatly improved. In addition to building large-scale training face datasets, such as CASIA-WebFace, VGGFace2 and MS-Celeb-1M, designing novel loss functions have drawn much attention to improving the intra-class compactness of features. As an important research topic, many loss function variants are proposed to enhance the discrimination of features, e.g. L-softmax [6] and SphereFace [7] impose an angular margin to concentrate the samples in a sphere manifold. Similar to SphereFace, CosFace [8] and ArcFace [9] also impose angular margins to the decision boundaries of original softmax loss, leading to further improvement.

These methods commonly set the same margin for each class. The intra-class variations within each class are compressed equally. These methods implicitly assume that the features of the different classes are generally uniformly distributed. However, the internal feature distribution of different classes in the
FIGURE 1 Visualization of the features of the CIFAR-10 extracted under different loss functions. (a) Two-dimensional feature distribution obtained by the softmax loss function. (b) ArcFace increases intra-class compactness. (c) AMR-Loss has a more compact distribution of features within a class, and the spacing between classes is also larger.

FIGURE 2 The decision margin of two different loss functions in binary classification. The dotted line represents the decision boundary. Note that \( \theta_i \) is the angle between \( x_i \) and \( w_{y_i} \) (as shown in Equation (2), \( x_i \) denotes the deep feature of the \( i \)-th samples, belonging to the \( y_i \)-th class, \( w_{y_i} \) denotes the weights for the \( y_i \)-th class in the last fully connected layer), \( m \) is the constant margin and \( m_1 \) is the margin of class \( C_1 \) (\( m_1 \) can be trained).

Real-world data is highly unbalanced, as illustrated in Figure 1(a), and the distance between different classes is uneven. The uniform treatment is not able to increase the separability between classes. For intra-class feature learning, based on feature distribution, we propose an adaptive margin to squeeze the feature interior spaces of different classes.

In this paper, we propose a novel loss function, i.e. adaptive margin and regularization loss (AMR-Loss), which uses an adaptive margin and diversity regularization to impact the feature distribution. For output layers, diversity regularization aims to distribute the classifier neurons (classifier neurons are the projection bases of the last layer (i.e. output layer) before input to softmax) as uniformly as possible to improve the inter-class feature separability, as shown in Figure 4(b). Based on the feature distribution, the margins of different classes are determined adaptively, and the margins of each class are variable and learnable. Formally, \( m_i \) denotes margin trained for each class (\( m_i \) can be regarded as the shortest distance between class \( C_i \) and all other classes), such that the decision boundary is given by \( r(\cos(\theta_1 + m_1) - \cos \theta_2) = 0 \). The decision boundary of ArcFace is \( r(\cos \theta_1 + m) - \cos \theta_2 = 0 \) (\( r \) is the scale factor), as shown in Figure 2. On the other hand, we find that the margin corresponding to a class can fully reflect the class’s classification performance in the training phase—small margins correspond to low classification performance, as shown in Table 1. Because a class’s corresponding margin can fully reflect its classification performance during the training process, we weight the diversity regularization by the margin, so that diversity regularization focuses on the class with poor discrimination. This weight is called dynamic diversity regularization weight. Overall, our main contributions are as follows:

1. We introduce an adaptive margin, which ensures that the model can learn a particular margin for each identity to squeeze its intra-class variations adaptively.
2. We present a regularization term based on the adaptive margin, which explicitly enlarges the distance between different identities. Feature discrimination is improved by promoting inter-class separability for face recognition.
3. The experimental results on image classification and existing accessible face datasets show that the proposed method can be applied to other variants of softmax to improve their performance further.

TABLE 1 Accuracy of each class on CIFAR-10 and the corresponding margin value

| Classes     | AeroCar | Bird | Cat | Deer | Dog | Frog | Horse | Ship | Truck |
|-------------|---------|------|-----|------|-----|------|-------|------|-------|
| Accuracy (softmax) | 91.69% | 96.21% | 88.25% | 82.36% | 92.51% | 86.29% | 93.27% | 93.63% | 95.19% |
| Accuracy (AMR-Loss) | 92.03% | 97.30% | 89.43% | 83.07% | 94.64% | 87.30% | 94.01% | 94.82% | 96.26% |
| Margin      | 0.68    | 0.91  | 0.55 | 0.51 | 0.70 | 0.54 | 0.75  | 0.77  | 0.90  |

2 | RELATED WORKS

As CNNs and face recognition are well studied, in this section, we mainly discuss two sub-fields closely related to this paper: (1) loss function; (2) diversity regularization.
2.1 Evolution of loss function

The loss function is an essential part of a deep CNNs, which points out the direction of optimization for network training. There are two types of loss function server for two purposes: verification and classification. The contrastive loss [10, 11] is a verification loss function that optimizes the Euclidean distance between paired features in the feature space. Besides, softmax loss is the essential classification loss function, and there are various evolutions [12]. The center loss proposed by Wen et al. [2] penalizes the Euclidean distance between the feature and its corresponding clustering center, to reduce intra-class variations. L2-softmax adds a constraint to the features during training such that the L2-norm remains constant, which provides similar attention to both good and bad quality faces. NormFace [13] builds a cosine layer in a standard CNNs model by normalizing the features and weights of the last inner product layer. Recent studies have found that adding a cosine angle margin between different classes has a noticeable effect on improving feature discrimination. Large-margin softmax and A-softmax [14] add a multiplicative angular margin to squeeze each class. Similar to A-softmax, CosFace and ArcFace add cosine margin through addition. Unlike ArcFace setting fixed margin, in AdaptiveFace [15], the problem of the unbalanced number of samples in different classes of face datasets is solved by the adaptive margin corresponding to the number of training samples in each class.

2.2 Diversity regularization

Although CNNs have achieved great success, over-parameterization [16] can lead to redundant and highly correlated neurons (e.g. weights of the last inner-product layer), as shown in Figure 4(a). In order to reduce the redundancy of neurons and release the generalization ability of the network, current works address the redundancy problem by enforcing relatively large diversity between pairwise projection bases via regularization. Such methods are diversity regularization [17]. Diversity regularization is widely used in sparse coding [18], ensemble learning [19], metric learning [20] etc. The early studies of sparse coding have shown that the generalization ability of codebook (CodeBook) can be improved by diversity regularization. Diversity is usually modelled using the (empirical) covariance matrix. Many recent works have been carried out to improve the diversity regularization in the neural network [21], in which the regularization is mainly realized by promoting large-angle orthogonality or reducing the covariance between base vectors. The diversity of neurons is promoted by minimizing the hyperspherical energy of neurons on the hypersphere [22, 23].

3 INTUITION AND MOTIVATIONS

This section explains our motivation for the proposed diversity regularization based on the adaptive margin.

3.1 The relationship between adaptive margin and class separability

We experiment on CIFAR-10 to analyze the relationship between margin values and feature distribution. We use a simple eight-layer CNNs and softmax Loss for training. At test time, we test the accuracy of each class, shown in Table 1; we found that the classification accuracy of bird, cat and dog in CIFAR-10 is significantly lower than in other classes.

We randomly selected 20 samples from each class in the test set and used T-SNE [24] to reduce the network's output dimension from 512-D to 2-D for visualization. As shown in Figure 1(a), there is no apparent separation between the features of birds, cats and dogs in CIFAR-10 and other classes' features, which leads to poor classification results. Then, to verify the relationship between the distribution of class features in feature space and adaptive margin, we experimented with AMR-Loss on CIFAR-10. The changes in the margin in the training process of AMR-Loss are shown in Figure 3 (see Section 5.4 for experimental settings).

Figure 3 suggests that the classes (such as a car, ship) with good separability (the feature distribution is separated from other classes) in the training process, their margin's growth rate is much higher than other classes, and finally keep a large value (0.91, 0.90), the class with bad separability (such as bird and cat), margin keeps a small value (0.55, 0.51) for a long time. Therefore, the value of the margin should be adapted to the distribution of features.

3.2 Diversity regularization

Inter-class separability and intra-class compactness are two key factors that affect the feature distinguishability. However, CNN’s methods based on softmax, such as Center Loss, SphereFace and ArcFace, mainly focus on the intra-class compactness.

Let $G$ be all the network layers of the model except the output layer, and $\theta$ be the parameters of the model. The matrix $W \in \mathbb{R}^{d \times s}$ represents the parameters of the output layer ($d$ is the length of the output features; $n$ is the number of classes in the training set), which maps the image features predicting the classes. Given the input image $I$, we get its feature representa-
softmax loss of margin with adaptive MARGIN AND DIVERSITY feature angle:

In a batch, and its class label is \( y_i \).

The points in the figure represent the clustering center for each class \((w_j)\). The class color in the figure corresponds to Figure 1. It can be seen that the clustering centers of each class obtained by AMR-Loss are more scattered

**FIGURE 4** softmax Loss and AMR-Loss. The points in the figure represent the clustering center for each class \((w_j)\). The class color in the figure corresponds to Figure 1. It can be seen that the clustering centers of each class obtained by AMR-Loss are more scattered.

**4 | MARGIN AND DIVERSITY REGULARIZATION OF ADAPTIVE FEATURE DISTRIBUTION**

In this section, we will give details of our approach, i.e. adaptive margin’s diversity regularization based on feature distribution.

**4.1 | Softmax loss of margin with adaptive feature distribution**

In order to improve the compactness within the class, a fixed margin is introduced based on Equation (4), the expression is as follows:

\[
L_{\text{softmax}} = -\frac{1}{M} \sum_{i=1}^{M} \log \frac{e^{y_i w_j^T x_i + b_j}}{\sum_{j=1}^{M} e^{w_j^T x_i + b_j}}
\]

where \( x_i \in \mathbb{R}^d \) represents the output feature of the \( i \)-th sample in a batch, and its class label is \( y_i \). \( w_j \in \mathbb{R}^d \) represents the \( j \)-th column of the weight \( \mathbf{W} \in \mathbb{R}^{d \times n} \) of the output layer, and \( b_j \in \mathbb{R}^n \) is the bias term. The number of samples in a batch is \( M \), and the number of classes in the training set is \( n \).

Applying L2 normalization on \( w_j \) and \( x_i \) to make \( \|w_j\| = 1 \), \( \|x_i\| = 1 \), the feature distance can be expressed as the following feature angle:

\[
w_j^T x_i = \|w_j\| \|x_i\| \cos \theta_j = \cos \theta_j
\]

where \( \theta_j \) is the angle between \( x_i \) and its corresponding \( w_j \).

Based on Equation (3) and fix the bias \( b = 0 \), modify Equation (2) as follows:

\[
L_{\text{angle}} = -\frac{1}{M} \sum_{i=1}^{M} \log \left( \frac{e^{y_i \cos \theta_j}}{\sum_{j=1}^{n} e^{y_j \cos \theta_j}} \right)
\]

where \( y_i \) is the label of the \( i \)-th sample.

In Section 5.4, we visualized the output layer weight \( w_j \) (clustering center of the \( j \)-th class) of the CIFAR-10 model trained by the softmax to the three-dimensional sphere, as shown in Figure 4(a), it can be seen that the distance between \( w_j \) of poorly separable classes (such as bird, cat, dog) is relatively small. Therefore, we can increase the separability between classes by focusing on expanding the distance between \( w_j \) of these poor separable classes to improve the discrimination of features potentially. Combined with these two parts, we use the feature distribution adaptive margin to adjust the weight of our diversity regularization dynamically. The training focuses on expanding the less separable classes and forces the clustering center \( w_j \) away from each other during training.

**Image 62x618 to 272x734**

**TABLE 4** softmax Loss and AMR-Loss. The points in the figure represent the clustering center for each class \((w_j)\). The class color in the figure corresponds to Figure 1. It can be seen that the clustering centers of each class obtained by AMR-Loss are more scattered.
to limit the changes in margin values during training (Dynamic margin Loss, $L_{dm}$):

$$L_{dm} = \text{sigmoid} \left( -\frac{1}{M} \sum_{m=1}^{M} m_{y_i} \right), \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

The above formula calculates the average margin of all classes in a batch. $m_{y_i}$ is the margin corresponding to class $y_i$, and $M$ is the size of the number of samples in a batch. The Adaptive margin value is between 0.5 and $\pi$ because if there is no minimum limit of 0.5, $m$ will become negative about the beginning of the training, which will affect the training instead. We also limit the maximum value to $\pi$ to ensure the stability of the training.

$$\frac{\partial L_{dm}}{\partial m_{y_i}} = \frac{\sum_{m=1}^{M} \delta(y = y_i) \cdot (-1)}{1 + \sum_{m=1}^{M} \delta(y = y_j)} \cdot \frac{\partial \text{sigmoid}(\mu)}{\partial (\mu)},$$

$$\mu = -\frac{1}{M} \sum_{m=1}^{M} m_{y_i}.$$  

(9)

We add a sigmoid function to limit the gradient of $L_{dm}$, which makes the margin change smoother so that we can use margin to change the training weights of our diversity regularization dynamically. Equation (8) ensures that the margin's value is as large as possible, provided that the training data can be fitted. Without this item, the model tends to make the margin 0 during training. The gradient of $m_{y_i}$ in Equation (8) is shown in Equation (9).

### 4.2 Diversity regularization

As mentioned in Section 3.1, the separability of different classes can be represented by an adaptive margin with feature distribution in our method. Different regularization weights are set according to different margin values, which are defined as

$$k_i = 2.0 - \text{sigmoid}(m_{y_i} - 0.5) \times \beta.$$  

(10)

In the above formula, $m_{y_i}$ is the margin corresponding to class $y_i$, and $\beta$ is the scale factor, which enlarges the difference between $k_i$. $\beta = 10$ is used in this paper. The formula represents the margin's increase by calculating the difference between $m_{y_i}$ and the initial value of 0.5 during training, and then re-scale it by $\beta$. Finally, the sigmoid function is used to limit the values between 0 and 1, and subtract from 2.0 to get the corresponding regularization weights for each class. The weight $k_i$ obtained by Equation (10) changes with the change of $m_{y_i}$, and the larger $m_{y_i}$, the smaller the weight $k_i$, and vice versa, as shown in Figure 5.

As described in Section 3.2, $w_j$ can be considered as the cluster centre for all samples labelled $y = j$. To increase the distance between feature distributions of different classes of samples, we introduce a diversity regularization that enlarges the distance between feature cluster centres and adds weight to our regularization using the weight $k_i$ derived from the feature distribution's adaptive margin. The smaller the margin, the larger the $k_i$, which makes the regularization focus on expanding the distance between cluster centres of the less separable classes. After Equation (10), the regularization term is defined as (Regularization Weights Loss, $L_{rw}$)

$$L_{rw} = \frac{1}{n} \sum_{i=1}^{n} \max_{(i \neq j)} \left( \frac{1}{\|w_i - w_j\|_2} \right) \cdot (k_i),$$  

(11)

$n$ is the number of classes. $w_i$ and $w_j$ represent the cluster centres of Class $i$ and Class $j$, and $(k_i)$, is the diversity regularization weight of Class $i$, where $\max$ stands for stop gradient; that is, the $k_i$ gradient computed by $L_{rw}$ is no longer propagated backward (tf.stop_grad$(k_i)$ command is used in Tensorflow). Because $k_i$ is the diversity regularization weight set by the adaptive margin, in Equation (11), we only need to use the adaptive margin to reflect the class's classification performance. We do not need to update the margin through Equation (11), so $(\ast)$ is used. In Equation (11), the max() function is used to only punish the distance between the weight vector of other classes that are most similar to the corresponding weight vector of each class and the weight vector of that class, to reduce the calculated amount of backpropagation. The corresponding $w_j$ gradient for Equation (11) is as follows:

$$\frac{\partial L_{rw}}{\partial w_j} = \frac{2(w_j - w_i)}{\|w_j - w_i\|_2^4} \cdot (k_i).$$  

(12)

By forcing $w_j$ to be separated from each other during training, Equation 11 makes $w_j$ more diverse and reduces its redundancy, as shown in Figure 4(b).
4.3 | Final loss design

Combining feature distribution adaptive margin with adaptive regularization, the overall loss function is (AMR-Loss, $L_{AMR}$)

$$L_{AMR} = L_{AM} + \lambda (L_{dm} + L_{rw})$$

(13)

where $\lambda$ is the balance factor between the two formulae. Because $L_{dm}$ and $L_{rw}$ are mutually reinforcing, the farther clustering centres deviate from each other, the larger the margin will be. Through experiments, it is better to use the same balance factor $\lambda$ for $L_{dm}$ and $L_{rw}$.

In Algorithm 1, we summarize the learning detail in the CNNs with AMR-Loss.

**ALGORITHM 1 The Adaptive Margin and Regularization Loss**

**Input**: Training data $\mathcal{X}$. Initialized parameters $\theta$, in convolutional layers. Parameters $\theta_f$ and $\theta_r$ in output layers, respectively. Hyperparameter $\lambda$ and learning rate $\eta$. The number of iteration $t \leftarrow 0$.

**Output**: The parameters $\theta_c$.

1: while not converge do
2: \hspace{1em} $t \leftarrow t + 1$.
3: \hspace{1em} Compute the joint loss by $L_{AMR}^t = L_{AM}^t + \lambda (L_{dm}^t + L_{rw}^t)$.
4: \hspace{1em} Compute the backpropagation error $\frac{\partial L_{AMR}^t}{\partial \theta_c}$ for each $i$ by $\frac{\partial L_{AMR}^t}{\partial \theta_c} = \frac{\partial L_{AM}^t}{\partial \theta_c} + \lambda \cdot \frac{\partial L_{dm}^t}{\partial \theta_c} + \lambda \cdot \frac{\partial L_{rw}^t}{\partial \theta_c}$.
5: \hspace{1em} Update the parameters $\theta_c$ for each $j$ by $\theta_c^{t+1} = \theta_c^t - \eta \cdot \frac{\partial L_{AMR}^t}{\partial \theta_c}$.
6: Update the parameters $\theta_f$ by $\theta_f^{t+1} = \theta_f^t - \eta \cdot \frac{\partial L_{AM}^t}{\partial \theta_f}$.
7: Update the parameters $\theta_r$ by $\theta_r^{t+1} = \theta_r^t - \eta \cdot \frac{\partial L_{dm}^t}{\partial \theta_r}$.
8: end while

5 | EXPERIMENT AND ANALYSIS

To evaluate AMR-Loss, we conducted experiments on CIFAR-10 and CASIA-WebFace. First, we report the results of AMR-Loss visualization on the CIFAR-10 dataset to validate the effectiveness of AMR-Loss. Second, we compare the performance of AMR-Loss against existing methods on all datasets. Finally, we provide visualizations and analysis to illustrate how AMR-Loss has achieved its effectiveness.

5.1 | Implementation details

5.1.1 | Experimental settings

All experiments in this paper are implemented in Tensorflow. We set the batch size to 32 and train the model on an NVIDIA GeForce GTX 1080Ti (11GB) GPU. The learning rate starts from 0.1, and the learning rate of 20000 times per iteration is divided by 10, and a total of 1.5 million times of training. We use the Momentum optimizer and set the momentum to 0.9 and weight decay to 5e-4.

5.1.2 | Network structure

The network structure of this paper uses ResNet20 architecture used in [2, 10]. The network structure is shown in Figure 6. The network accepts $112 \times 112$ RGB images as input, and four residual blocks contain a total of 20 convolutional layers, and the shape of the output feature map is $7 \times 7 \times 512$. Then, the feature map of $7 \times 7 \times 512$ is mapped to a 512-D vector by the first fully connected layer (FC1), and then the 512-D vector and the parameters of the last fully connected layer (output layer) are normalized with $L_2$. After normalization, $x$ and $w$ are dot product to get the angle between the current feature and the target weight, then add our adaptive margin. $L_{dm}$ is used for training the adaptive margin of feature distribution. $w$ is transferred into $L_{rw}$ before normalization. $L_{rw}$ uses margin to add weight to our diversity regularization so that each class of corresponding $w$ has a further distance from each other.

5.2 | Dataset

As shown in Table 2, we train our CNNs model with CASIA-WebFace, and some samples of this dataset can be found in Figure 7.

We use standard test methods to test each dataset. We use LFW and YTF sets. LFW dataset includes 13,233 face images from 5749 different identities. YTF dataset includes 3424 videos from 1595 different individuals, with an average of 2.15 videos per person. The clip durations vary from 48 frames to 6070 frames, with an average length of 181.3 frames. Both LFW and YTF contain faces with large variations in pose, expression and illumination. Considering that LFW has been well solved, we further evaluate our method on the more challenging LFW BLUR protocol [30].

We use the CFP-FP with a large change in attitude and the AgeDB-30 with a large change in age. The CFP-FP dataset includes 500 people, each with 4 profile faces and 10 front faces. AgeDB-30 data set is a face data set in a natural scene with great changes in posture, expression, occlusion and age. There are 12240 face pictures, 440 different people; most of them are celebrities, including actors, writers, scientists, etc. Each picture in the data set has class, age and gender information.

5.3 | Parameter analysis

As shown in Equation (10), the only hyperparameter in AMR-loss is the weight factor $\lambda$. Given a large $\lambda$, the $L_{dm}$ and $L_{rw}$ play a leading role in training. If $\lambda$ is small, $L_{am}$ loss will dominate the learning process. We use different $\lambda$ training methods and show the accuracy of LFW.

When $\lambda = 0$, the loss described in Equation (10) only works with $L_{AM}$. As shown in Figure 8, with the increase of $\lambda$, the performance of LFW increases rapidly and reaches the peak value when $\lambda = 8$. So in the following experiments, we use $\lambda = 8$. 
To intuitively understand the changes in the margin using CASIA-WebFace training, we randomly selected five identities to track their margin changes during training, as shown in Figure 9.

### 5.4 Test on CIFAR-10 dataset

To intuitively show AMR-Loss’s effect, we designed an experiment (network structure shown in Table 3) to demonstrate the feature distribution trained by different loss functions.

![Figure 7](image7.png) CASIA-WebFace dataset, in which the image of the same column belongs to the same label

In our experiment in CIFAR-10, we randomly selected 20 samples for each class in the test set, extract features from the network trained by different loss functions and use T-SNE to reduce features from 512-D to 2-D for visualization.

As shown in Figure 1(a), the features obtained from the model trained by softmax have certain separability, but there is no obvious boundary between the features of some classes (such as bird, cat, dog). By adding a margin, each class’s intra-class features are more compact (Figure 1(b)). We propose an adaptive margin to improve intra-class compactness and an adaptive regularization to improve inter-class separability. As shown in Figure 1(c), AMR-Loss gets the features that are more

![Figure 8](image8.png) Accuracy of LFW at different $\lambda$

![Figure 9](image9.png) The change of margin in face dataset training
TABLE 3 The network structure is used in the visualization experiment of CIFAR-10

| Layer     | CNN-8                  |
|-----------|------------------------|
| Conv1.x   | [3 × 3, 64] × 2        |
| Pool1     | 2 × 2 max pooling, stride 2 |
| Conv2.x   | [3 × 3, 128] × 2       |
| Pool2     | 2 × 2 max pooling, stride 2 |
| Conv3.x   | [3 × 3, 256] × 2       |
| Pool3     | 2 × 2 max pooling, stride 2 |
| Conv4.x   | [3 × 3, 512] × 2       |
| Pool4     | 2 × 2 max pooling, stride 2 |
| Fully Connected | 512                  |

TABLE 4 softmax, ArcFace, Lam and Lam + Lrw accuracy on CIFAR-10

| Method          | CIFAR-10 |
|-----------------|----------|
| Softmax         | 91.40    |
| ArcFace         | 91.71    |
| $L_{AM}$        | 92.23    |
| $L_{AM} + L_{rw}(L_{AMR})$ | 92.45 |

Compact within a class and have obvious boundaries between classes.

Then we also visualize the weight ($w_j$) of the network output layer. As shown in Figure 4(b), it is evident that the feature clustering centres of each class obtained by our method are more scattered.

Finally, we test the accuracy of the CIFAR-10. As shown in Table 4 on CIFAR-10, $L_{AM} + L_{rw}(L_{AMR})$ performance is significantly better than softmax and its variants, which proves the effectiveness of AMR-Loss in the image classification task.

5.5 Tests on LFW and YTF

We use a network architecture shown in Figure 6 to train softmax, its variants and our proposed AMR-Loss. The original and left and right flipped face images are input into the model, and the corresponding output is connected as the final feature. We evaluate performance on LFW and YTF datasets under the standard test method.

As shown in Table 5, our proposed AMR-Loss can significantly improve the performance of the original softmax loss. $L_{AM} + L_{rw}(L_{AMR})$ combination is better than $L_{AM}$ loss alone and has a very obvious difference. AMR-Loss also has better performance than other softmax variants (such as Sphereface, ArcFace). It is shown that under the joint supervision of adaptive margin and adaptive regularization, the model can learn more distinguishing features, proving the effectiveness of AMR-Loss.

TABLE 5 The performance comparison between LFW, YTF and LFW BLUFR datasets

| Method                  | Data             | LFW  | YTF  |
|-------------------------|------------------|------|------|
| softmax Loss            | 97.88            | 90.10|
| Center Loss             | 98.91            | 93.40|
| $L_{softmax}$           | 99.01            | 93.00|
| SphereFace              | WebFace          | 99.26| 94.10|
| ArcFace                 | 99.30            | 94.11|
| $L_{AM}$                | 99.33            | 94.09|
| $L_{AM} + L_{rw}(L_{AMR})$ | 99.38 | 94.13|

TABLE 6 The performance of the LFW BLUFR protocol

| Method                  | VR@FAR = 0.001% | VR@FAR = 0.01% |
|-------------------------|-----------------|----------------|
| softmax Loss            | 83.44           | 93.53          |
| SphereFace              | 96.91           | 98.93          |
| ArcFace                 | 98.28           | 99.31          |
| $L_{AM}$                | 98.26           | 99.32          |
| $L_{AM} + L_{rw}(L_{AMR})$ | 98.47 | 99.36 |

Considering that LFW has been well solved, we further evaluate our method on the more challenging LFW BLUFR protocol, which focuses on low FARs. We report the result in Table 6. As it is clear, our method is superior to all current state-of-the-art methods.

5.6 Tests on CFP-FP and AgeDB-30

In Table 7, we show the validation accuracy of softmax-Loss, ArcFace, $L_{AM}$, $L_{AM} + L_{rw}(L_{AMR})$ in CFP-FP and AgeDB-30 datasets. Experiments show that our method has a significant improvement compared with existing methods such as ArcFace in the more difficult test set. It shows that AMR-Loss has a better effect on various datasets than the existing methods.

TABLE 7 The performance comparison between CFP-FP and AgeDB-30 datasets

| Method                  | Data  | CFP-FP | AgeDB-30 |
|-------------------------|-------|--------|----------|
| softmax Loss            | 93.02 | 89.63  |
| ArcFace                 | WebFace | 90.63  | 89.63    |
| $L_{AM}$                | 94.01 | 90.79  |
| $L_{AM} + L_{rw}(L_{AMR})$ | 94.05 | 90.82  |

5.7 Visualizations

Furthermore, to visualize AMR-Loss’s effect, we designed a toy experiment to demonstrate the feature distributions trained by
FIGURE 10 Feature distribution visualization of several loss functions

FIGURE 11 The margin distributions

different loss functions. We select face images from six identities in CASIA-WebFace to train CNNs models (Table 3), which output three-dimensional features. We normalized the obtained three-dimensional features and plotted them on the sphere. The participated losses are softmax, ArcFace and the proposed AMR-Loss with different $\lambda$. As shown in Figure 10, we can observe that the bad aggregation and separability within softmax loss classes lead to bad decision boundaries. ArcFace reduces intra-class variations but ignores the separability between classes to some extent. Its clustering centre has not changed. AMR-Loss reduces intra-class variations and increases inter-class variations. Furthermore, by increasing $\lambda$, the features of those classes are almost clustered at one point. After training with CASIA-WebFace, we counted the distribution of adaptive margin, as shown in Figure 11.

6 | CONCLUSION

In this paper, we propose a new loss function AMR-Loss for face recognition, which consists of two parts. First, we introduce the margin of adaptive feature distribution for each class, increasing the compactness of the intra-class adaptively. The second is the diversity regularization of adaptive weights, which focuses on the classes with low separability by adaptively adjusting the regularized weights and forcing the expansion of the boundary between classes in training. Compared with the existing methods, this method increases the intra-class compactness and expands the inter-class separability. As described in the experimental part, our method provides significant improvements in accuracy on multiple face datasets.

ACKNOWLEDGEMENTS

The authors thank to Yang Zhou and Rui Chen for their contributions to this paper. This work is partially supported by the National Natural Science Foundation of China (61876158), Sichuan Science and Technology Program (2019YFS0432), National Key Research and Development Project (2016YFC0802209), the Key Research and Development Program of Guangzhou under Grant 202007050002.

REFERENCES

1. Kaiming, H., et al.: Deep residual learning for image recognition. IEEE conference on computer vision & pattern recognition. IEEE computer society, 2016, pp. 770–778
2. Wen, Y., et al.: A Discriminative Feature Learning Approach For Deep Face Recognition. In: Computer Vision – ECCV 2016, pp. 499–515, Springer International Publishing, Heidelberg (2016)
3. Xiao, Z., et al.: Range loss for deep face recognition with long-tailed training data. IEEE international conference on computer vision, pp. 5409–5418 (2017)
4. Belhumeur, P.N., et al.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 711–720 (1997)
5. Suykens, J., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. 9, 293–300 (1999)
6. Wang, F., et al.: Additive margin softmax for face verification. IEEE Signal Process. Lett. 25(7), 926–930 (2018)
7. Weiyan, L., et al.: Sphereface: Deep hypersphere embedding for face recognition. IEEE conference on computer vision and pattern recognition, pp. 212–220 (2017),
8. Hao, W., et al.: Cosface: Large margin cosine loss for deep face recognition. IEEE conference on computer vision and pattern recognition, pp. 5265–5274 (2018)
9. Jiankang, D., et al.: Arcface: Additive angular margin loss for deep face recognition. IEEE conference on computer vision and pattern recognition, pp. 4690–4699 (2019)
10. Chopra, S., et al.: Learning a similarity metric discriminatively, with application to face verification. IEEE computer society conference on computer vision and pattern recognition, vol. 1, pp. 539–546 (2005)
11. Florian, S., et al.: Facenet: a unified embedding for face recognition and clustering. IEEE conference on computer vision and pattern recognition, pp. 815–823 (2015)
12. Taigman, Y., et al.: Deepface: Closing the gap to human-level performance in face verification. IEEE conference on computer vision and pattern recognition, pp. 1701–1708 (2014)
13. Wang, F., et al.: Normface: l2 hypersphere embedding for face verification. Proceedings of the 25th ACM international conference on multimedia, pp. 1041–1049 (2017)
14. Wang, F., et al.: Additive margin softmax for face verification. IEEE Signal Process Lett. 25(7), 926–930 (2018)
15. Liu, H., et al.: AdaptiveFace: Adaptive margin and sampling for face recognition. IEEE conference on computer vision and pattern recognition, pp. 11947–11956 (2019)
16. Aruni, R., et al.: Reducing duplicate filters in deep neural networks. In NIPS: workshop on Deep Learning: Bridging Theory and Practice vol. 1, 1–7 (2017)
17. Pengtao, X., et al.: Learning latent space models with angular constraints. Proceedings of the 34th international conference on machine learning, vol. 70, pp. 3799–3810 (2017)
18. Ramirez, I., et al.: Classification and clustering via dictionary learning with structured incoherence and shared features. IEEE computer society conference on computer vision and pattern recognition, pp. 3501–3508 (2010)
19. Nan, L., et al.: Diversity regularized ensemble pruning. Joint European conference on machine learning and knowledge discovery in databases, pp. 330–345 (2012)
20. Xie, P., et al.: Orthogonality-promoting distance metric learning: Convex relaxation and theoretical analysis. In ICML, pages 5403–5412, (2018)
21. Di, X., et al.: All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthornormality and modulation. IEEE conference on computer vision and pattern recognition, pp. 6176–6185 (2017)
22. Liu, W., et al.: “Learning towards Minimum Hyperspherical Energy,” in Advances in Neural Information Processing Systems (NeurIPS), 1–16, 2018
23. Miyato, T., et al.: Spectral normalization for generative adversarial networks. International Conference on Learning Representations. pp. 1–26 (2018)
24. Laurens, V.D.M.: Accelerating t-SNE using tree-based algorithms. J. Mach. Learn. Res. 15(1), 3221–3245 (2014)
25. Dong, Y., et al.: Learning face representation from scratch. arXiv preprint (2014); arXiv:1411.7923 (2014)
26. Huang, G.B., et al.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October (2007)
27. Wolf, L., et al.: Face recognition in unconstrained videos with matched background similarity. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 529–534. IEEE (2011)
28. Sengupta, S., et al.: Frontal to profile face verification in the wild. IEEE winter conference on applications of computer vision, pp. 1–9 (2016)
29. Stylianos, M., et al.: AgeDB: The first manually collected, in-the-wild age database. IEEE conference on computer vision and pattern recognition workshops, pp. 51–59 2017
30. Liao, S., et al.: A benchmark study of large-scale unconstrained face recognition. IEEE international joint conference on biometrics pp. 1–8 (2014),

**How to cite this article**: Zhang Z, Gong X, Chen J. Face recognition based on adaptive margin and diversity regularization constraints. IET Image Process. 2021;15:1105–1114. https://doi.org/10.1049/ipr2.12089