Inter-class Angular Margin Loss for Face Recognition

Wenming Yang[a,*], Jingna Sun[a], Riqiang Gao[a], Jing-Hao Xue[b], Qingmin Liao[a]

[a] *Shenzhen Key Lab. of Info. Sci&Tech/Shenzhen Engineering Lab. of IS&DCP, Department of Electronic Engineering/Graduate School at Shenzhen, Tsinghua University, China*

[b] *Department of Statistical Science, University College London, UK*

Abstract

Increasing inter-class variance and shrinking intra-class distance are two main concerns and efforts in face recognition. In this paper, we propose a new loss function termed inter-class angular margin (IAM) loss aiming to enlarge the inter-class variance. Instead of restricting the inter-class margin to be a constant in existing methods, our IAM loss adaptively penalizes smaller inter-class angles more heavily and successfully makes the angular margin between classes larger, which can significantly enhance the discrimination of facial features. The IAM loss can be readily introduced as a regularization term for the widely-used Softmax loss and its recent variants to further improve their performances. We also analyze and verify the appropriate range of the regularization hyper-parameter from the perspective of backpropagation. For illustrative purposes, our model is trained on CASIA-WebFace and tested on the LFW, CFP, YTF and MegaFace datasets; the experimental results show that the IAM loss is quite effective to improve state-of-the-art algorithms.

Keywords: Face recognition; IAM loss; inter-class variance; intra-class distance; Softmax loss.

*Corresponding author

Email addresses: yang.wenming@sz.tsinghua.edu.cn (Wenming Yang), sunjn17@mails.tsinghua.edu.cn (Jingna Sun), rqgao15@gmail.com (Riqiang Gao), jinghao.xue@ucl.ac.uk (Jing-Hao Xue), liaoqm@sz.tsinghua.edu.cn (Qingmin Liao)

1. Introduction

Convolutional neural networks (CNNs) are widely used for face recognition [1–15], in which recent researches have been focused on increasing the inter-class variance and reducing the intra-class distance. A typical pipeline of using a network for training WebFace can be found in Fig. 1, in which the network is trained with the loss function in the last layer, and the representation in the penultimate layer is used as the feature of human faces.

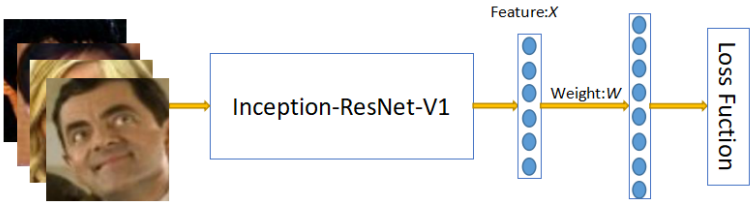


Figure 1: A typical pipeline for training WebFace

Hence the recent efforts and achievements in increasing the inter-class variance and reducing the intra-class distance can be summarized into two categories.

First, to optimize the Euclidean distance between facial features, mainly through regularization. For example, the Triplet loss [6] makes the intra-class Euclidean distance of features shorter than the inter-class distance. Wen et al. [16] reduce the intra-class Euclidean distance by adding an extra penalty. The Marginal loss of [17] and our past work [18] limit both intra-class and inter-class Euclidean distances to improve recognition accuracy. The Range loss [19] overcomes the problem of long-tailed data by equalizing intra-class Euclidean distance and increasing inter-class Euclidean distance. Except for the Triplet loss, all above methods add a regularization term on the basis of the Softmax loss, which is generally adjusted via a regularization hyper-parameter.

Second, to optimize the angle between different classes. Optimization of angles has recently become an attractive way to improve the loss function. The ℓ_2-constrained Softmax loss [20] finds that the ℓ_2-norm of the feature is related

to the quality of images and it restricts the features to lie on a hypersphere of a fixed radius. The Ring loss [21] normalizes the weights and constrains the ℓ_2-norm of features to approximate a constant. The L-Softmax loss [22] transforms the inner product of weights and features, which remains a cosine value, and multiplies a hyper-parameter to the angle. Based on the L-Softmax loss, the SphereFace loss [23] performs the ℓ_2-normalization of weights. The AM-Softmax loss [9] and the NormFace loss [24] introduce a scale factor s to the cosine value after normalizing the weights and features to make the whole network easier converge and train. The work of [9, 11] subtracts a constant margin m from the cosine value, while the ArcFace loss [10] directly adds a margin to the angle itself. In this second category, [9–11] increase the inter-class angle and reduce the intra-class angle simultaneously through introducing a constant angular margin, and they do not use an additional penalty term as with the Euclidean distance optimization (see the first category).

Inspired by these recent loss functions and considering the issue, that these state-of-the-art loss functions only introduce constant margins regardless of the distinct angles between classes, we propose in this paper a new loss function termed inter-class angular margin (IAM) loss. The IAM loss is designed to act as a regularization term for these state-of-the-art angular losses, aiming to adaptively enlarge inter-class variance by penalizing more heavily smaller inter-class angles. Moreover, we also provide an analysis of the proper range of the regularization hyper-parameter from the perspective of backpropagation. Experiments on LFW [25], CFP [26], YTF [27] and MegaFace [28] demonstrate that our proposed IAM loss is quite effective. It can substantially improve the performance of the Softmax loss and its variants for face recognition.

The rest of this paper is organized as follows. The related work is introduced in Section 2. Section 3 proposes our new IAM loss. Section 4 analyzes the range of regularization hyper-parameter β and demonstrates the differences between the IAM loss and other state-of-the-art loss functions. In Section 5, we verify the effectiveness of the IAM loss through experiments. Finally, Section 6 makes a summary of this paper.

2. Related Work

We begin by introducing some state-of-the-art loss functions that have been proposed to enhance the Softmax loss in the last layer of CNNs.

The Softmax loss is a powerful loss function expressed in terms of the conditional probability of a sample $\boldsymbol{x}_i$ belonging to a class y_i, as defined below:

$$L_{soft} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\boldsymbol{w}_{y_i}^T\boldsymbol{x}_i}}{\sum_{j=1}^{C}e^{\boldsymbol{w}_j^T\boldsymbol{x}_i}}\ , \tag{1}$$

where N is the size of a batch, C is the number of classes, $\boldsymbol{x}_i$ is the feature vector in the penultimate layer of the network, and $\boldsymbol{w}_{y_i}$ denotes the weight of the class that $\boldsymbol{x}_i$ belongs to (and $\boldsymbol{w}_j$ represents the weight of the jth class). Both $\boldsymbol{x}_i$ and $\boldsymbol{w}_j$ are vectors and their inner product is called target logit [29], hence Eq.(1) can be rewritten by using the angle $\theta_{i,j}$ between the two vectors:

$$L_{soft} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\|\boldsymbol{w}_{y_i}\|\|\boldsymbol{x}_i\|\cos\theta_{i,y_i}}}{\sum_{j=1}^{C}e^{\|\boldsymbol{w}_j\|\|\boldsymbol{x}_i\|\cos\theta_{i,j}}}\ . \tag{2}$$

As with [24, 30, 31], we can first normalize $\boldsymbol{x}_i$ and $\boldsymbol{w}_j$, and then multiply the cosine value of angle between $\boldsymbol{x}_i$ and $\boldsymbol{w}_j$ by a scale factor s to make the network easier converge, which leads to a scaled loss as

$$L_{scale} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\theta_{i,y_i}}}{\sum_{j=1}^{C}e^{s\cos\theta_{i,j}}}\ . \tag{3}$$

The scaled Softmax loss cannot tighten the intra-class distance or enlarge the inter-class variance directly. To tackle this issue, recently several enhancement methods for Eq.(3) have been proposed. As displayed in Eq.(4), SphereFace [23] multiplies angle θ_{i,y_i} by a hyper-parameter m:

$$L_{sphere} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{i,y_i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{i,y_i})}+\sum_{j=1,j\neq y_i}^{C}e^{\|\boldsymbol{x}_i\|\cos\theta_{i,j}}}\ . \tag{4}$$

In Eq.(4), the boundary for binary classification ($C = 2$) is $m\theta_{i,1} = \theta_{i,2}$ (when $y_i = 1$). Compared with the original boundary $\theta_{i,1} = \theta_{i,2}$ in Eq.(3), SphereFace produces an angular margin, which can increase inter-class variance and reduce intra-class distance. However, Eq.(4) cannot converge independently and is usually optimized with the Softmax loss.

Alternatively, [9] and [11] subtract a margin m from the cosine value of angle θ_{i,y_i} in Eq.(3):

$$L_{cos} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos\theta_{i,y_i}-m)}}{e^{s(\cos\theta_{i,y_i}-m)} + \sum\limits_{j=1,j\neq y_i}^{C} e^{s\cos\theta_{i,j}}} \ . \tag{5}$$

Then the binary classification boundary becomes $\cos\theta_{i,1} - m = \cos\theta_{i,2}$ (when $y_i = 1$). Unlike Eq.(4), Eq.(5) can converge without using the Softmax loss.

ArcFace [10] has a similar effect to Eq.(5) of [9, 11] except that ArcFace directly adds a margin to the angle itself:

$$L_{arc} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(\theta_{i,y_i}+m)}}{e^{s\cos(\theta_{i,y_i}+m)} + \sum\limits_{j=1,j\neq y_i}^{C} e^{s\cos\theta_{i,j}}} \tag{6}$$

The decision boundary under binary classification is now $\theta_{i,1} + m = \theta_{i,2}$ (when $y_i = 1$).

Eq.(5) and Eq.(6) of [9–11] all aim to optimize the angle between classes, by adding a constant margin in order to enlarge inter-class variance and shrink intra-class distance. However, an issue with these state-of-the-art losses is that the margins introduced are constant regardless of the distinct angles between classes. Hence we propose the IAM loss as a regularization term to these losses, to adaptively increase inter-class variance by penalizing more heavily smaller inter-class angles.

3. The Proposed Method

We will first propose the IAM loss as an adaptive penalty term for the state-of-the-art methods in this section, and then analyze the range of the hyper-

parameter β from the perspective of back propagation in Section 4. The rationality of β and the effectiveness of our IAM loss will be verified in Section 5.

To penalize the samples with small inter-class angles, the IAM loss is defined as

$$L_{IAM} = \frac{1}{N}\sum_{i=1}^{N}\log\frac{\frac{1}{C-1}\sum\limits_{j=1,j\neq y_i}^{C} e^{s\cos\theta_{i,j}}}{\sum\limits_{j=1}^{C} e^{s\cos\theta_{i,j}}} \ . \tag{7}$$

Regarding the proposed IAM loss, we would make the following remarks.

Firstly, the numerator in Eq.(7) approximately describes the average similarity between feature $\boldsymbol{x}_i$ and weight $\boldsymbol{w}_j$ where $j \neq y_i$. Intuitively, if the angle between feature $\boldsymbol{x}_i$ and $\boldsymbol{w}_j$ for $j \neq y_i$ is smaller, the numerator will be larger, which also will result in a larger IAM loss. That is, the IAM loss will adaptively penalize smaller inter-class angle more heavily in the training process, instead of restricting the inter-class margin to be a constant like m. Because of the independence of IAM loss as a regularization term, it can be added to many state-of-the-art methods and further promote the recognition accuracy.

Secondly, we note that for each summand in Eq.(7), which can be rewritten in the form of $\log(1-p_i)$ (see Eq.(12) below in Section 4), there exists a nonlinear monotonic transformation to obtain it from a corresponding summand in Eq.(3), which can be rewritten in the form of $-\log p_i$. Hence when there is only one observation (i.e. $N = 1$), optimizing these two losses yields the same solution. However, as $N > 1$ in practice, the summation and the nonlinearity of the transformation render a solution to the optimization in terms of the IAM loss in Eq.(7) different from the solution in terms of the scaled loss in Eq.(3).

Finally, we propose to add the IAM loss as a penalty term to the existing Softmax loss and its variants like the scaled loss in Eq.(3) with a non-negative regularization parameter β. The total loss function can be written as

$$L = L_{base} + \beta L_{IAM} \ , \tag{8}$$

where L_{base} represents a current established loss such as Eq.(3), Center loss [16], AM-loss [9], ArcFace [10], etc. In this way, we further leverage the strength of

Algorithm 1 Learning with the IAM loss

Input: Training data $\{x_i\}$; training labels $\{y_i\}$; parameters ϕ of Inception-ResNet-V1; weights of classifier $\widetilde{\boldsymbol{w}}$; learning rate r; hyper-parameter β; and maximum number of iterations t_m

Output: Parameters ϕ of Inception-ResNet-V1

$t \leftarrow 1$

repeat

  Randomly select a mini-batch of size N from the training set

  Normalize the features and weights

  Compute the total loss $L = L_{base} + \beta L_{IAM}$

  Update ϕ by $\phi \leftarrow \phi - r\frac{1}{N}\sum_{i=1}^{N}[\frac{\partial L_{base}}{\partial \widetilde{\boldsymbol{x}}_i} + \beta\frac{\partial L_{IAM}}{\partial \widetilde{\boldsymbol{x}}_i}]\frac{\partial \widetilde{\boldsymbol{x}}_i}{\partial \phi}$.

  Update $\widetilde{\boldsymbol{w}}_{y_i}$ by $\widetilde{\boldsymbol{w}}_{y_i} \leftarrow \widetilde{\boldsymbol{w}}_{y_i} - r\frac{1}{N}\sum_{i=1}^{N}[\frac{\partial L_{base}}{\partial \widetilde{\boldsymbol{w}}_{y_i}} + \beta\frac{\partial L_{IAM}}{\partial \widetilde{\boldsymbol{w}}_{y_i}}]$

  Update $\widetilde{\boldsymbol{w}}_j$ by $\widetilde{\boldsymbol{w}}_j \leftarrow \widetilde{\boldsymbol{w}}_j - r\frac{1}{N}\sum_{i=1}^{N}[\frac{\partial L_{base}}{\partial \widetilde{\boldsymbol{w}}_j} + \beta\frac{\partial L_{IAM}}{\partial \widetilde{\boldsymbol{w}}_j}]$

  $t \leftarrow t + 1$

until $t > t_m$

both the IAM loss and an established loss to reach a better loss function by weighting them adaptively via β, as illustrated in Fig. 2 in Section 5.

In the training process, it is easy to optimize the IAM loss by using many current optimizers such as RMSPROP. We show the gradients in Eq.(9), Eq.(10) and Eq.(11), and based on these equations we summarize the optimization process in Algorithm 1:

$$\frac{\partial L_{IAM}}{\partial \widetilde{\boldsymbol{x}}_i} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{1-p_i}\frac{\sum_{j\neq \widetilde{\boldsymbol{w}}_{y_i}}^{C} s(\widetilde{\boldsymbol{w}}_{y_i} - \widetilde{\boldsymbol{w}_j})e^{s\widetilde{\boldsymbol{w}}_{y_i}^T\widetilde{\boldsymbol{x}}_i}e^{s\widetilde{\boldsymbol{w}_j}^T\widetilde{\boldsymbol{x}}_i}}{(\sum_{j=1}^{C}e^{s\widetilde{\boldsymbol{w}_j}^T\widetilde{\boldsymbol{x}}_i})^2}, \tag{9}$$

where

$$p_i = \frac{e^{s\widetilde{\boldsymbol{w}}_{y_i}^T\widetilde{\boldsymbol{x}}_i}}{\sum_{j=1}^{C}e^{s\widetilde{\boldsymbol{w}_j}^T\widetilde{\boldsymbol{x}}_i}},\ \widetilde{\boldsymbol{w}}_j = \frac{\boldsymbol{w}_j}{\|\boldsymbol{w}_j\|},\ \widetilde{\boldsymbol{x}}_i = \frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}\ ;$$

$$\frac{\partial L_{IAM}}{\partial \widetilde{\boldsymbol{w}}_{y_i}} = -\frac{1}{N}\sum_{i=1}^{N}s\widetilde{\boldsymbol{x}}_i p_i\ ; \tag{10}$$

$$\frac{\partial L_{IAM}}{\partial \widetilde{\boldsymbol{w}}_j} = \frac{1}{N}\sum_{i=1}^{N} s\widetilde{\boldsymbol{x}}_i(p_{ij2} - p_{ij1}) \ , \tag{11}$$

where

$$p_{ij1} = \frac{e^{s\widetilde{\boldsymbol{w}}_j^T\widetilde{\boldsymbol{x}}_i}}{\sum_{j=1}^{C} e^{s\widetilde{\boldsymbol{w}}_j^T\widetilde{\boldsymbol{x}}_i}} \ , \ p_{ij2} = \frac{e^{s\widetilde{\boldsymbol{w}}_j^T\widetilde{\boldsymbol{x}}_i}}{\sum_{j=1,j\neq y_i}^{C} e^{s\widetilde{\boldsymbol{w}}_j^T\widetilde{\boldsymbol{x}}_i}} \ .$$

4. Discussion

4.1. The Range of Hyper-parameter β

In this section, we will discuss the range of hyper-parameter β in Eq.(8). We take Eq.(3) as the base loss function and rewrite the equation as

$$\begin{aligned} L &= L_{scale} + \beta L_{IAM} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log p_i + \beta\frac{1}{N}\sum_{i=1}^{N}\log\frac{1-p_i}{C-1} \ . \end{aligned} \tag{12}$$

Then, from the perspective of back propagation, we have

$$\begin{aligned} \frac{\partial L}{\partial \widetilde{\boldsymbol{w}}_{y_i}} &= \frac{1}{N}\sum_{i=1}^{N}(-s\widetilde{\boldsymbol{x}}_i(1-p_i) - \beta s\widetilde{\boldsymbol{x}}_i p_i) \\ &= -\frac{1}{N}\sum_{i=1}^{N} s\widetilde{\boldsymbol{x}}_i(1 - p_i + \beta p_i) \ , \end{aligned} \tag{13}$$

$$\frac{\partial L}{\partial \widetilde{\boldsymbol{w}}_j} = \frac{1}{N}\sum_{i=1}^{N} s\widetilde{\boldsymbol{x}}_i(p_{ij1} + \beta p_{ij2} - \beta p_{ij1}) \ . \tag{14}$$

We discuss the value of β as follows.

1) $\beta = 0$. In Eq.(13), the absolute value of the gradient increases as probability p_i decreases. It implies that if the intra-class angle between $\widetilde{\boldsymbol{x}}_i$ and $\widetilde{\boldsymbol{w}}_{y_i}$ is larger, the change of $\widetilde{\boldsymbol{w}}_{y_i}$ will be greater in the process of back propagation, which is conductive to reduce intra-class distance. In Eq.(14), the absolute value of the derivative for $\widetilde{\boldsymbol{w}}_j$ is positively correlated with probability p_{ij1}. It indicates that when the inter-class angle between $\widetilde{\boldsymbol{x}}_i$ and $\widetilde{\boldsymbol{w}}_j$ is smaller, the change of $\widetilde{\boldsymbol{w}}_j$ is greater, which will enlarge inter-class variance.

2) $\beta = 1$. As Eq.(13) shows, the derivative depends only on the value of $\widetilde{\boldsymbol{x}}_i$ which belongs to the y_ith class. Based on this, $\widetilde{\boldsymbol{w}}_{y_i}$ now can be regarded as the center of the y_ith class. Eq.(14) only relates to $\widetilde{\boldsymbol{x}}_i$ and p_{ij2}. Actually, p_{ij2} reflects the proportion of the inner product of $\widetilde{\boldsymbol{x}}_i$ and weight of the jth class. When the proportion is larger (i.e. the angle between $\widetilde{\boldsymbol{w}}_j$ and $\widetilde{\boldsymbol{x}}_i$ is smaller), the inter-class margin will be increased.

3) $\beta < 1$. In Eq.(13), the gradient adds one item $-\beta s\widetilde{\boldsymbol{x}}_i p$ as the IAM loss is applied, which increases the absolute value of the gradient. That is, the proposed IAM loss can optimize better by using a larger gradient to reduce intra-class distance. As presented in Eq.(14), the gradient is also larger than the traditional Softmax loss, which will increase inter-class variance significantly.

4) $\beta > 1$. The absolute value of the gradient of Eq.(13) increases as probability p_i increases. It indicates that the change of $\widetilde{\boldsymbol{w}}_{y_i}$ will be larger with smaller intra-class angle, which is against the purpose of reducing intra-class distance. In Eq.(14), with smaller inter-class angle, the gradient is lower, which is also against the purpose of enlarging inter-class variance.

In summary, β should be chosen less than 1, and our IAM loss helps to improve the optimization of scaled Softmax loss.

4.2. Differences from Other State-of-the-art Methods

In this section, we will discuss the differences between our IAM loss and other state-of-the-art loss functions [9, 11, 23, 24, 32, 33].

The AM loss [9] and CosFace [11] add a constant inter-class margin, and both can be seen as a variant of the Softmax loss. In contrast, our IAM loss optimizes the inter-class margin adaptively in the training process. A constant inter-class margin is limited due to the discrepancy between different classes, while the IAM loss can further adjust the inter-class margin. Moreover, the IAM loss is a regularization term that can be added to many state-of-the-art losses for further improvement in the classification accuracy.

The SphereFace [23] introduce an angle margin and needs to train with the Softmax loss. In the experiments on SphereFace, we find that the influence of

the Softmax loss is dominant in the most training steps, which is not conducive to optimize the inter-class margin. The IAM loss acts on the whole training process to further promote the recognition accuracy, as a regularization term, which is more flexible and universal. In Table. 6, the accuracy of "Eq.(3) + IAM" is higher than "SphereFace [23]", which also shows the priority of the IAM loss.

The NormFace [24] proposes to normalize the features and weights, and adds a scale 's' to make the network easy to converge. It optimizes the cosine similarity and does not involve the concept of enhancing the inter-class margin, as indicated in Eq.(3). From Table. 5, we can observe that our IAM loss can improve its accuracy remarkably.

The work in [33] reduces the intra-class distance and increases the inter-class margin based on the class centers and proposes an ACD loss. Its main idea is to compact the samples predicted correctly to the corresponding class center and keep the misclassified samples away from the predicted class center. In [33], only samples predicted wrongly can be used to increase the inter-class margin and only the corresponding class centers are considering. In the training process, the effect of the increased inter-class margin decreases as the wrongly predicted samples decrease; in contrast, the IAM loss has effects in the whole training process and measures the average inter-class margin among all classes, which can better ensure a large inter-class margin. Additionally, the IAM loss can be added to many state-of-the-art loss functions, which is more flexible than that of [33]. The method of [32] is similar to the center loss [16], with the difference being that [32] weights the distance between correctly predicted samples and the corresponding class centers differently from the distance between wrongly predicted samples and the relevant class centers. In [32], it does not increase the inter-class margin specifically, while the IAM loss serves for increasing the inter-class margin.

In short, our IAM loss is more flexible, can adaptively optimize the inter-class margin, and can be applied to regularize many state-of-the-art loss functions.

5. Experiments

5.1. A Toy Example

Table 1: The network for MNIST

Layer	Kernel-size	Outputs	Number
conv	[3,3]	32	2
Max-pool	[3,3]	-	1
conv	[3,3]	64	2
Max-pool	[3,3]	-	1
conv	[3,3]	128	2
Max-pool	[3,3]	-	1

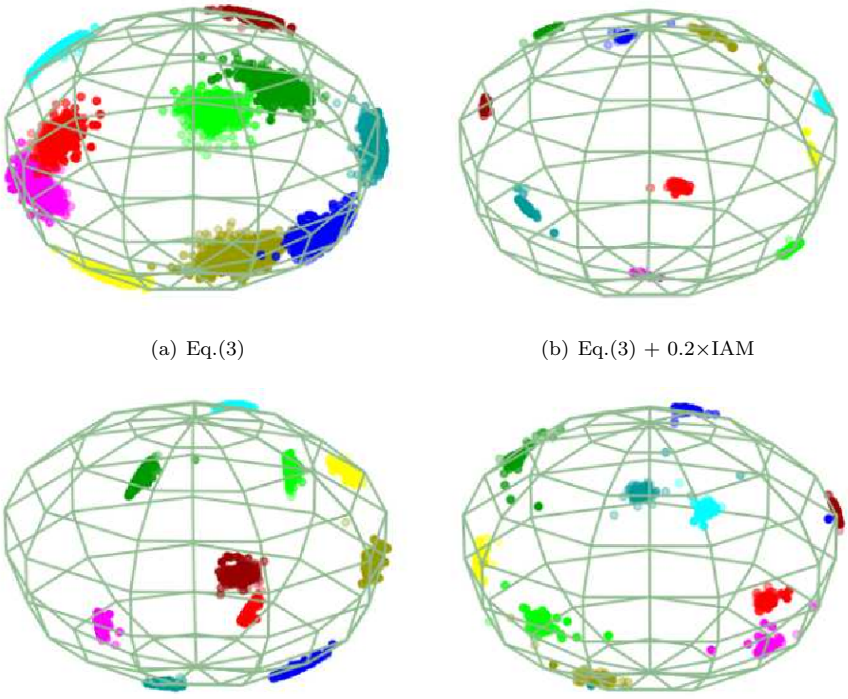
We implement experiments on MNIST dataset [34] to visualize the feature distribution so that the effect of IAM loss can be displayed intuitively. The network structure trained for the MNIST dataset [34] is shown in Table 1. The "Number" in Table 1 is the quantity of the corresponding layer. We draw 10,000 training samples. The output dimension of the penultimate layer is 3 and the 3D maps for different loss functions are shown in Fig. 2. To present the effectiveness of our IAM loss quantitatively, we also list corresponding recognition accuracies in Table 2.

Fig. 2(a) represents the 3D visualization of Eq.(3) on MNIST; in contrast, Fig. 2(b) is for the combination of Eq.(3) and the IAM loss. Similarly for the loss functions of [9, 11] in Fig. 2(c) and its regularization by the IAM loss in Fig. 2(d).

The samples are more separable after adding our IAM loss. For example, compared with Fig. 2(a), Fig. 2(b) enlarges the inter-class variance and reduces the intra-class distance significantly. The samples are clustered to their class

Table 2: Recognition accuracy on MNIST

Eq.(3)	Eq.(3) + 0.2×IAM	[9][11], $m = 0.1$	[9][11] + 0.2×IAM, $m = 0.1$
99.08%	99.42%	99.24%	99.42%



(a) Eq.(3)

(b) Eq.(3) + 0.2×IAM

(c) [9][11], $m = 0.1$

(d) [9][11] + 0.2×IAM, $m = 0.1$

Figure 2: Three-dimensional maps for different loss functions

centers, and these centers are distributed more separately on the whole sphere surface. A similar pattern can be observed by comparing Fig. 2(d) with Fig. 2(c). The accuracies showed in Table 2 also confirm the effectiveness of our IAM loss.

5.2. Experiments on Face Datasets

The face detection and alignment algorithm applied in our experiments are MTCNN [35]. The face images are pre-whitened before being fed to the network. For illustrative purposes, we adopt Inception-ResNet-V1 [6, 36, 37] as our training CNN architecture. The training dataset CASIA-WebFace [38] contains about 0.49 million face images from 10,575 subjects. During training, we set the weight decay parameter and batch size to 5e-4 and 90, respectively. The learning rate begins with 0.1 and is divided by ten at the 60K and 120K iterations. The datasets LFW [25], CFP [26], YTF [27] and MegaFace [28] are tested for the evaluation of compared methods.

LFW [25] contains about 13,000 images and has a list of 6000 pairs to verify. CFP [26] contains frontal vs frontal and frontal vs profile images. In the experiment, we test the frontal vs. profile protocol, which contains 7,000 pairs with 3,500 same pairs and 3,500 not-same pairs for 500 different subjects. YTF [27] contains 3,425 videos of 1,595 different people, and we test 5,000 video pairs by using the average features. MegaFace [28] is an extremely challenging dataset, in which a probe image should match the correct image from a gallery of about 1 million images.

We use two regularized loss functions to verify the effectiveness of our IAM loss in experiments: 1) use the IAM loss to regularize the scaled Softmax loss in Eq.(3); that is, use Eq.(12) as the loss function; the test accuracies with different values of regularization parameter β shall be discussed; and 2) use the IAM loss to regularize Eq.(5). This will not change our analysis of the derivative of $\widetilde{\boldsymbol{w}}_{y_i}$ and $\widetilde{\boldsymbol{w}}_j$, although the denominators of p_i and p_{ij1} are slightly different.

The evaluation of Eq.(12) is shown in Table 3, and we set the parameter $s = 30$. We list the validation accuracies of LFW [25], YTF [27] and CFP [26] over different values of β for verifying our analysis in Section 4. As Table 3 shows,

Table 3: Accuracy for Eq.(12) (i.e. the IAM-regularized Eq.(3))

β	LFW	YTF	CFP
2	98,900%	94.960%	91.471%
1	99.033%	94.640%	92.700%
0.9	99.017%	94.640%	**93.200**%
0.8	**99.117**%	**95.280**%	92.714%
0.7	99.083%	95.06%	92.757%
0.6	99.033%	94.920%	93.017%
0.5	99.100%	94.940%	92.400%
0.4	99.067%	94.720%	92.314%
0.3	99.050%	94.68%	92.600%
0.2	99.067%	94.520%	92.243%
0.1	98.850%	93.920%	91.114%
0	98.567%	93.740%	90.140%

our IAM loss improves the scaled Softmax loss greatly (e.g. from 98.567% to 99.117% when $\beta = 0.8$ on LFW), which is even better than the work of [9, 11] (e.g. 99.117% vs. 99.067% on LFW). Hence Table 3 indicates the effectiveness of our method. We note that, when $\beta = 2$, the test accuracy is better than the scaled Softmax loss ($\beta = 0$), and we conjecture that the loose intra-class distance caused by the Softmax loss is the reason.

Then we add the IAM loss as a regularization term to Eq.(5) and the experimental results are shown in Table 4. The s in the experiment is 30 and $m = 0.4$. As presented in Table 4, the accuracy when $\beta > 1$ is worse than that when $\beta = 0$, which is consistent with our analysis of β. Table 4 also confirms that our proposed IAM loss can improve the face verification accuracy of the recent angular loss.

Furthermore, in Table 5, we evaluate the universal effectiveness of the IAM loss by applying it to regularize state-of-the-art methods [9–11, 16, 20, 21, 23]. The results show that our IAM loss not only can largely improve the performance

Table 4: Accuracy for the IAM-regularized Eq.(5)

β	LFW	YTF	CFP
2	99.017%	94.660%	91.586%
1	98.967%	95.100%	92.271%
0.1	99.133%	95.280%	92.843%
0.09	99.150%	95.140%	92.757%
0.08	**99.200**%	95.220%	93.300%
0.07	98.950%	95.020%	93.071%
0.06	99.150%	95.420%	**93.343**%
0.05	**99.200**%	**95.520**%	93.014%
0.04	99.183%	95.260%	93.271%
0	99.067%	95.300%	93.029%

of the scaled Softmax loss in Eq.(3), but also can further improve the accuracies of the state-of-the-art methods. To further verify the effectiveness of our IAM loss on larger test datasets, we show the rank-1 accuracy on MegaFace [28] in Table 6. The probe image needs to search one million images to find the correct matching image, which is one of the most challenging datasets. From the Table. 6, we can observe that the IAM loss still can improve the state-of-the-art methods. In short, these experimental results verify the effectiveness of our IAM loss.

6. Conclusions

In this paper, we propose a new loss function termed the IAM loss to increase inter-class variance adaptively. The IAM loss can be used as a regularization term for the scaled Softmax loss and its variations, and we also analyze the range of regularization hyper-parameter and its effects. The validity of the IAM loss has also been demonstrated by combining it with state-of-the-art losses [9–11, 16, 20, 21]. On the network Inception-ResNet-V1 and the training set WebFace, our method achieves clearly promising improvements.

Table 5: Improving state-of-the-art methods

Method	LFW	YTF	CFP
Eq.(3)	98.567%	93.740%	90.140%
Eq.(3) + 0.8×IAM	**99.117**%	**95.280**%	92.714%
Eq.(3) + 0.9×IAM	99.017%	94.640%	**93.200**%
Center loss [16]	98.833%	95.120%	92.500%
[16] + 0.05×IAM	98.950%	**95.300**%	**93.129**%
[16] + 0.1×IAM	**99.083**%	**95.300**%	92.957%
ℓ_2-constrained loss [20]	98.983%	94.880%	91.786%
[20] + 0.05×IAM	**99.033**%	95.060%	92.286%
[20] + 0.1×IAM	**99.033**%	**95.540**%	**93.329**%
Ring loss [21]	**99.117**%	94.480%	91.786%
[21] + 0.05×IAM	98.967%	94.560%	**92.100**%
[21] + 0.1×IAM	99.033%	**94.920**%	91.829%
SphereFace [23]	99.100%	95.120%	92.671%
[23] + 0.05×IAM	99.067%	95.160%	**92.857**%
[23] + 0.1×IAM	**99.133**%	**95.400**%	92.657%
ArcFace [10]	99.067%	94.800%	93.014%
[10] + 0.05×IAM	**99.100**%	94.680%	**93.486**%
[10] + 0.1×IAM	98.983%	**94.980**%	93.314%
AM-loss [9][11]	99.067%	95.300%	93.029%
[9][11] + 0.05×IAM	**99.200**%	**95.520**%	93.014%
[9][11] + 0.06×IAM	99.150%	95.420%	**93.343**%
[9][11] + 0.07×IAM	98.950%	95.020%	93.017%
[9][11] + 0.08×IAM	**99.200**%	95.220%	93.300%

Table 6: Face identification on MF1. Rank 1 refers to rank-1 face identification accuracy.

Method	MageFace
Eq.(3)	48.779%
Eq.(3) + 0.8×IAM	75.230%
Eq.(3) + 0.9×IAM	**76.825**%
Center loss [16]	64.522%
[16] + 0.05×IAM	67.249%
[16] + 0.1×IAM	**69.843**%
ℓ_2-constrained loss [20]	64.767%
[20] + 0.05×IAM	67.619%
[20] + 0.1×IAM	**71.225**%
Ring loss [21]	63.215%
[21] + 0.05×IAM	63.141%
[21] + 0.1×IAM	64.795%
SphereFace [23]	63.747%
[23] + 0.05×IAM	64.560%
[23] + 0.1×IAM	**65.277**%
ArcFace [10]	72.975%
[10] + 0.05×IAM	73.071%
[10] + 0.1×IAM	73.911%
AM-loss [9][11]	77.263%
[9][11] + 0.05×IAM	77.560%
[9][11] + 0.06×IAM	**78.451**%
[9][11] + 0.07×IAM	77.671%
[9][11] + 0.08×IAM	77.973%

Acknowledgment

We thank the anonymous reviewers for their constructive comments which help improve the quality of our manuscript. This work was partly supported by the National Natural Science Foundation of China (No.61471216 and No.61771276), the National Key Research and Development Program of China (No.2016YFB0101001), and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (No.JCYJ20170307153940960 and No.JCYJ20170817161845824)

References

[1] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[2] X. Wang, X. Tang, A unified framework for subspace face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1222–1228.

[3] Z. Li, W. Liu, D. Lin, X. Tang, Nonparametric subspace analysis for face recognition, CVPR 2 (2005) 961–966.

[4] Z. Li, D. Lin, X. Tang, Nonparametric discriminant analysis for face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (4) (2009) 755–761.

[5] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: CVPR, 2014, pp. 1891–1898.

[6] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: A unified embedding for face recognition and clustering, in: CVPR, 2015, pp. 815–823.

[7] B. Leng, K. Yu, Q. Jingyan, Data augmentation for unbalanced face recognition training sets, Neurocomputing 235 (2017) 10–14.

[8] J.-J. Lv, X.-H. Shao, J.-S. Huang, X.-D. Zhou, X. Zhou, Data augmentation for face recognition, Neurocomputing 230 (2017) 184–196.

[9] F. Wang, W. Liu, H. Liu, J. Cheng, Additive margin softmax for face verification, arXiv preprint arXiv:1801.05599.

[10] J. Deng, J. Guo, S. Zafeiriou, ArcFace: Additive angular margin loss for deep face recognition, arXiv preprint arXiv:1801.07698.

[11] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, W. Liu, CosFace: Large margin cosine loss for deep face recognition, arXiv preprint arXiv:1801.09414.

[12] D. Chen, C. Xu, J. Yang, J. Qian, Y. Zheng, L. Shen, Joint bayesian guided metric learning for end-to-end face verification, Neurocomputing 275 (2018) 560–567.

[13] B. Wu, Z. Chen, J. Wang, H. Wu, Exponential discriminative metric embedding in deep learning, Neurocomputing 290 (2018) 108–120.

[14] M. Wang, W. Deng, Deep face recognition: A survey, arXiv preprint arXiv:1804.06655.

[15] M. Wang, W. Deng, Deep visual domain adaptation: A survey, Neurocomputing.

[16] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: ECCV, Springer, 2016, pp. 499–515.

[17] J. Deng, Y. Zhou, S. Zafeiriou, Marginal loss for deep face recognition, in: CVPR Workshops, 2017, pp. 60–68.

[18] R. Gao, F. Yang, W. Yang, Q. Liao, Margin loss: Making faces more separable, IEEE Signal Processing Letters 25 (2) (2018) 308–312.

[19] X. Zhang, Z. Fang, Y. Wen, Z. Li, Y. Qiao, Range loss for deep face recognition with long-tail, arXiv preprint arXiv:1611.08976.

[20] R. Ranjan, C. D. Castillo, R. Chellappa, L2-constrained softmax loss for discriminative face verification, arXiv preprint arXiv:1703.09507.

[21] Y. Zheng, D. K. Pal, M. Savvides, Ring loss: Convex feature normalization for face recognition, in: CVPR, 2018, pp. 5089–5097.

[22] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks., in: ICML, 2016, pp. 507–516.

[23] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, SphereFace: Deep hypersphere embedding for face recognition, in: CVPR, 2017, pp. 212–220.

[24] F. Wang, X. Xiang, J. Cheng, A. L. Yuille, NormFace: L_2 hypersphere embedding for face verification, arXiv preprint arXiv:1704.06369.

[25] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. rep., Technical Report 07-49, University of Massachusetts, Amherst (2007).

[26] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, D. W. Jacobs, Frontal to profile face verification in the wild, in: IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–9.

[27] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: CVPR, IEEE, 2011, pp. 529–534.

[28] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, E. Brossard, The megaface benchmark: 1 million faces for recognition at scale, in: CVPR, 2016, pp. 4873–4882.

[29] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, G. Hinton, Regularizing neural networks by penalizing confident output distributions, arXiv preprint arXiv:1701.06548.

[30] W. Liu, Y.-M. Zhang, X. Li, Z. Yu, B. Dai, T. Zhao, L. Song, Deep hyperspherical learning, in: Advances in Neural Information Processing Systems, 2017, pp. 3953–3963.

[31] Y. Liu, H. Li, X. Wang, Rethinking feature discrimination and polymerization for large-scale recognition, arXiv preprint arXiv:1710.00870.

[32] M. M. Zhang, K. Shang, H. Wu, Learning deep discriminative face features by customized weighted constraint, Neurocomputing 332 (2019) 71–79.

[33] M. M. Zhang, K. Shang, H. Wu, Deep compact discriminative representation for unconstrained face recognition, Signal Processing: Image Communication 75 (2019) 118–127.

[34] Y. LeCun, The MNIST database of handwritten digits, http://yann. lecun. com/exdb/mnist/.

[35] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Processing Letters 23 (10) (2016) 1499–1503.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, IEEE, 2015, pp. 1–9.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: CVPR, 2016, pp. 2818–2826.

[38] D. Yi, Z. Lei, S. Liao, S. Z. Li, Learning face representation from scratch, arXiv preprint arXiv:1411.7923.