Class-Variant Margin Normalized Softmax Loss for Deep Face Recognition

Wanping Zhang, Yongru Chen, Wenming Yang*, Guijin Wang, Jing-Hao Xue, and Qingmin Liao

***Abstract*—In deep face recognition, the commonly-used softmax loss and its newly proposed variations are not yet sufficiently effective to handle the class imbalance and softmax saturation issues during the training process, while extracting discriminative features. In this brief paper, to address both issues, we propose a class-variant margin (CVM) normalized softmax loss, by introducing a true-class margin and a false-class margin into the cosine space of the angle between the feature vector and the class-weight vector. The true-class margin alleviates the class imbalance problem and the false-class margin postpones the early individual saturation of softmax. With negligible computational complexity increment during training, the new loss function is easy to implement in the common deep learning frameworks. Comprehensive experiments on the LFW, YTF and MegaFace protocols demonstrate the effectiveness of the proposed CVM loss function.**

***Index Terms*—Class-variant margin, class imbalance, early individual saturation, softmax loss.**

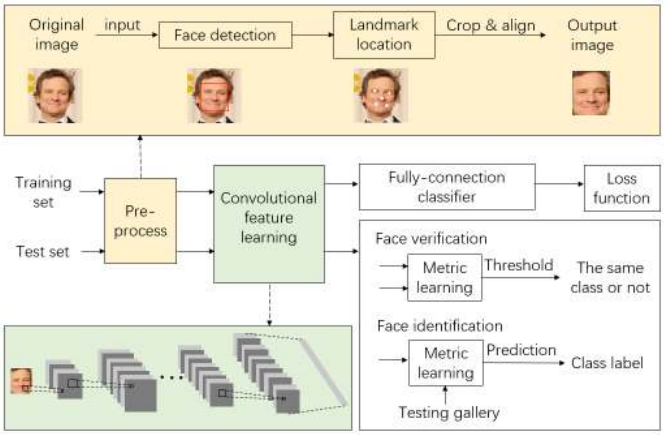


Fig. 1. A face recognition framework.

I. Introduction

CONVOLUTIONAL neural networks (CNNs) have been proved to be effective in numerous computer vision tasks, such as image classification [1], [2], object detection [3], [4], semantic segmentation [5], [6], and particularly face recognition [7], [8]. As show in Fig.1, face images need to be pre-processed firstly, i.e. by face detection, cropping and alignment, then processed images will be inputted into a CNN to extract features for recognition. For both testing settings, i.e., face verification [9] and face identification [10], the key for recognition is the discriminative feature. To construct such a powerful feature extractor, it is critical to design an appropriate loss function, which will supervise the learning of the network parameters. Hence, many studies have been focused on designing loss functions.

The existing loss functions can be largely divided into two categories: metric-based and margin-based. Metric-based loss functions are often based on metric learning to simultaneously make intra-class features compact and inter-class features remote from each other. Popular metric-based losses include contrastive loss [11], triplet loss [12] and center loss [13]. However, these losses either need a complicated sampling strategy or demand a time-consuming training [11], [12]. Margin-based loss functions generally add a margin to the classical softmax loss to make the separation more strict. Typical margin-based losses are L-softmax [8], A-softmax [14], CosFace [15] and ArcFace [16], which add margins to the angular space or cosine space, respectively.

However, there are two hard issues negatively affecting the training of CNNs for face recognition: class imbalance and softmax saturation. Class imbalance is severe in face recognition, as the number of face images per person varies greatly in most training datasets. This issue may cause the trained network to favor those categories that have more images in the training data, and bias the deep feature learning. Hence it is necessary to treat different categories differently, and some successful efforts have recently been made by the range loss [17] and the focal loss [18] to alleviate class imbalance problems. The focal loss can focus on the sparse set of hard examples through down-weighting the cross-entropy loss of well-classified examples, while the range loss aims to reduce the intra-class difference and enlarge the inter-class difference simultaneously within a mini-batch. The early softmax saturation refers to the short-lived gradients propagation that the softmax produces, which will impedes the exploration of stochastic gradient descends [19]. To mitigate this early saturation problem, noisy-softmax was proposed in [19], which injects noise into the softmax loss. Besides improvements on loss functions, there have been other work to enhance face recognition, such as separability and compactness network (SCNet) [20], semi-supervised sparse representation based

This work was partly supported by the Natural Science Foundation of Guangdong Province (No.2020A1515010711), the National Natural Science Foundation of China (No.61771276), and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (No.JCYJ20170817161845824). (Corresponding author: Wenming Yang)

W. Zhang, Y. Chen, W. Yang and Q. Liao are with Tsinghua Shenzhen International Graduate School / Department of Electronic Engineering, Tsinghua University, Shenzhen 518055, China (email: yangelwm@163.com).

G. Wang is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

J.-H. Xue is with the Department of Statistical Science, University College London, London WC1E 6BT, UK.

classification (S^3RC) [21] and specific face datasets [22].

In this brief paper, we aim to propose a simple yet effective loss function called class-variant margin (CVM) softmax loss, to address the class imbalance and softmax saturation problems for deep facial feature learning. More specifically, we introduce two margin functions into the cosine space of the softmax loss to address the two problems. For the class imbalance problem, we first introduce a reduced margin function to the cosine of the angle between the feature vector and true class weight vector, which we call a true-class margin, such that the misclassified examples can obtain a larger true-class margin to contribute more to the network optimization. For the softmax saturation problem, we introduce an additive margin function to the cosine of the angle between the feature vector and the false class weight vector, which we call a false-class margin, such that the examples near saturation can obtain a larger false-class margin to postpone the softmax saturation.

Our major contributions can be summarized as follows:

1) We propose a novel loss termed CVM loss that can simultaneously alleviate the class imbalance and softmax saturation problems in the training of CNNs.

2) The proposed CVM loss can be easily implemented under common CNN architectures and be directly optimized by the standard SGD method.

3) We train our model on the public available Casia-Webface dataset and verify its effectiveness on three popular benchmarks, LFW, YTF and MegaFace.

The rest of this paper is organized as follows. In Section II, we shall present the proposed CVM loss in details, from its motivation, intuition, formulation to discussion. In Section III, we shall empirically investigate the effectiveness of the CVM loss including the effect of its parameters and the superiority of its performance compared with other popular loss functions in face recognition. Section IV will summarize this brief paper.

II. Proposed loss function: CVM softmax loss

In this section, we will first analyze two existing problems in deep face recognition, then propose the class-variant margin (CVM) softmax loss function to address these problems, and finally present some discussions about our proposal.

A. Two existing problems in face recognition

1) Class imbalance: Class imbalance is severe in most training datasets for face recognition: e.g., for the popular Casia-Webface dataset, the curve for the number of images per person is plotted in Fig.2, where a clear long-tail distribution can be observed. In fact, it has been shown from empirical experiments and analysis that the classes with more samples will have a greater impact on the feature learning [17]. Hence, it is a critical issue to effectively handle the imbalanced data for improving feature discrimination in face recognition.

2) Softmax saturation: The softmax loss is commonly applied in classification applications. However, as rightly pointed out by [19], the softmax function suffers from an early individual saturation. For illustration, here we consider its use in a two-class classification scenario of sample x_i with the output score of class $y_i = 1$ being $P(y_i = 1 \mid x_i) = \frac{1}{1+e^{-(f_1(x_i)-f_2(x_i))}}$, in which $f_j(x_i)$ is the j-th element of the softmax input vector for sample x_i. As implied by the curve in Fig.3, the early saturated individuals (with their output scores already close to 1) actually contribute little to the gradient updating in the back-propagation process afterward. Hence, to fully exploit the information of these individuals, it is better to postpone the early individual saturation.

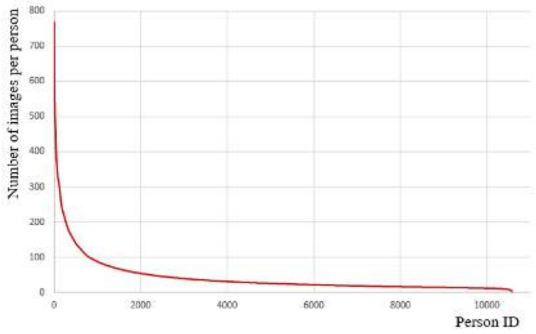


Fig. 2. The number of images per person in the Casia-Webface dataset.

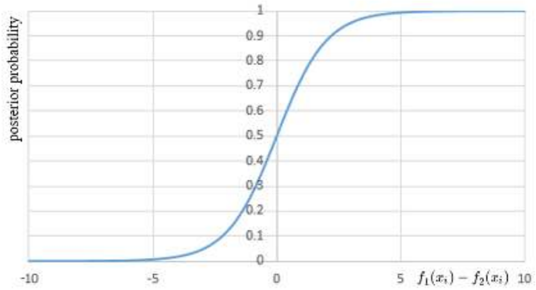


Fig. 3. A softmax function for binary classification, with the posterior probability $P(y_i = 1 \mid x_i) = \frac{1}{1+e^{-(f_1(x_i)-f_2(x_i))}}$.

B. Class-variant margin (CVM) softmax loss

In this paper, we propose a new, simple yet effective loss function termed class-variant margin (CVM) softmax loss, to address both the class imbalance and softmax saturation problems.

The original normalized softmax loss is

$$\begin{aligned} L_{ns} &= -\frac{1}{N}\sum_{i=1}^{N}\log P(y_i|x_i) = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{f_{y_i}}}{e^{f_{y_i}}+\sum_{j\neq y_i}e^{f_j}} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cdot(\cos\theta_{y_i})}}{e^{s\cdot(\cos\theta_{y_i})}+\sum_{j\neq y_i}e^{s\cdot(\cos\theta_j)}}. \end{aligned} \tag{1}$$

Two intuitions underlying our CVM loss function are:

1) Because the training was dominated by the majority classes with a large number of minority classes (persons on the long tails in Fig.2) unfortunately submerged, the new loss function is expected to strengthen the influence of the tail data during the training process.
   For the training samples from the minority classes, boundary features whose θ_{y_i} distributes around 90°

represent hard samples. Those features are key points to ensure intra-class and inter-class variations. Therefore, to enhance the impact of these points in the network training, we apply a larger margin to the cosine of the angle between the feature vector and the true class weight vector, when θ_{y_i} is around 90°. When the angle of true class is larger than 90°, a smaller margin should be applied because these training samples are likely to be outliers. Thus, we construct the true-class margin function $h(\theta)$ (see Fig.4) applied to the cosine of the angle between the feature vector and the true class weight vector.

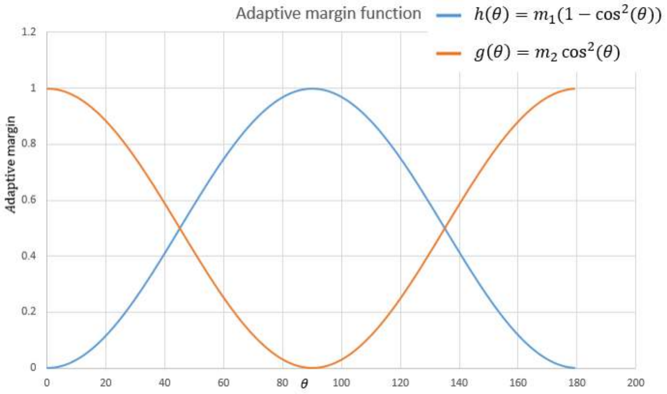


Fig. 4. true-class margin function $h(\theta)$ and false-class margin function $g(\theta)$.

2) Because the early individual saturation of the softmax loss led to short-lived gradient propagation, which is undesirable for the generalization and robust learning of the network, the new loss function should be able to postpone early saturation, for example, by enlarging the softmax input f_j when $j \neq y_i$.

   To address the softmax saturation problem, we construct the false-class margin function $g(\theta)$ to postpone the early individual saturation. When the confidence of the feature vector belonging to class j is low, i.e. the angle $\theta_{j,j\neq y_i}$ distributes near 180°, the feature vector tends to be classified correctly and approaches the softmax saturation zone. Hence, we add a larger margin (see Fig.4) to the cosine of such an angle $\theta_{j,j\neq y_i}$ to maintain a valid gradient propagation.

Combining the true-class margin function and false-class margin function, we propose the CVM loss as

$$L_{cvm} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\cdot(\cos\theta_{y_i} - h(\theta_{y_i}))}}{e^{s\cdot(\cos\theta_{y_i} - h(\theta_{y_i}))} + \sum_{j\neq y_i} e^{s\cdot(\cos\theta_j + g(\theta_j))}}, \tag{2}$$

$$\text{with} \quad h(\theta_{y_i}) = m_1(1 - \cos^2\theta_{y_i}), \tag{3}$$

$$g(\theta_j) = m_2 \cos^2\theta_j, \tag{4}$$

where subscripts j and y_i index class j and class y_i among C classes; N is the number of mini-batch; s is a scale factor; f_j (short for $f_j(x_i)$) is the j-th element of the softmax input for x_i; θ_j (short for $\theta_j(x_i)$) is the angle between the i-th feature vector x_i and the weight vector of the j-th class; $h(\theta_{y_i})$ is the margin function applied to the cosine of angle between the feature vector and the true class weight vector, named as the true-class margin; $g(\theta_j)$ is the margin function added to the cosine of angle between the feature vector and the false class weight vector, named as the false-class margin; m_1 and m_2 are two preset hyper-parameters; m_1 represents the upper bound of the true-class margin, and m_2 represents the upper bound of the false-class margin.

Plotted in Fig.4, the true-class margin $h(\theta_{y_i})$ and the false-class margin $g(\theta_j)$ are nonlinear mappings of the angles θ_{y_i} and θ_j, respectively. We design these two margin functions to alleviate the class imbalance problem and postpone early individual saturation, as elaborated below.

C. Discussion

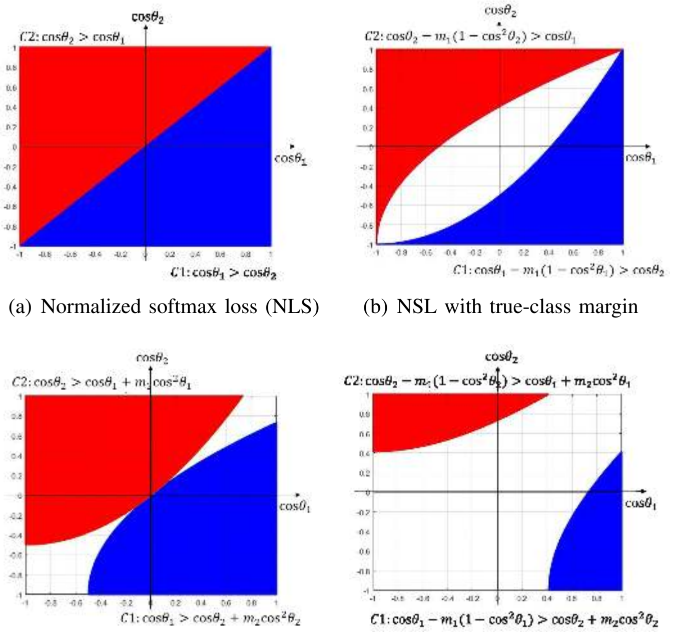
1) Margins: Here we take the two-class classification as example and list the decision boundary and margins of several popular loss functions. As shown in Table I, there have been some popular loss functions commonly used in face recognition. The softmax loss helps the convolution neural network quickly converge, but it cannot ensure the extracted features very discriminative. To improve the accuracy of face verification and face identification, CosFace and ArcFace apply a constant margin to cosine and angular spaces, respectively, to make the feature more discriminative. Although with the constant margin the inter-class distance can be enlarged, these two methods apply the same margin to all class but do not take the class discrepancy into consideration. Here, the proposed CVM normalized softmax loss applies a class-variable margin to the normalized softmax loss, through constructing true-class margin function and false-class margin function.

We also illustrate some of them in the cosine space in Fig.5, in which the blue areas represent class 1, while the red areas belong to class 2. We can observe that the decision margin of the normalized softmax loss (NLS, Fig.5(a)) is zero, making the loss function not very robust for the features around the decision boundary.

TABLE I
DECISION BOUNDARIES OF SOME POPULAR LOSS FUNCTIONS IN TWO-CLASS CLASSIFICATION CASE.

Loss Functions	Decision Boundaries
Softmax [14]	$(W_1 - W_2)x + (b_1 - b_2) = 0$
N-softmax [23]	$\cos\theta_1 - \cos\theta_2 = 0$
CosFace [15]	$\cos\theta_1 - \cos\theta_2 - m = 0$
ArcFace [16]	$\cos(\theta_1 + m) - \cos\theta_2 = 0$
CVM-softmax	$\cos\theta_1 - \cos\theta_2 - CVM(\theta_1, \theta_2) = 0$ $CVM(\theta_1, \theta_2) = m_1 \sin^2\theta_1 + m_2 \cos^2\theta_2$

To illustrate the effectiveness of the two terms (true-class margin and false-class margin), we first apply the true-class margin to the cosine of the angle between the feature vector and the true class weight vector. The samples of minority class are often distributed near the class boundary, which leads to small cosine values. Therefore, as larger margins are introduced by applying the true-class margin when both cosine values are small, we manage to make the features of those rare samples more discriminative, as shown in Fig.5(b). Then, for



(a) Normalized softmax loss (NLS) (b) NSL with true-class margin

(c) NSL with false-class margin (d) Class-variant margin softmax loss

Fig. 5. Decision margins for loss functions under binary classes case: the class 1 (C1) area is in blue and the class 2 (C2) area is in red. In (a), the C1 area is $\cos\theta_1 > \cos\theta_2$, while the C2 area is $\cos\theta_2 > \cos\theta_1$. In (b), we apply the true-class margin to the normalized softmax loss; the C1 area is $\cos\theta_1 - m_1(1-\cos^2\theta_1) > \cos\theta_2$, while the C2 area is $\cos\theta_2 - m_1(1-\cos^2\theta_2) > \cos\theta_1$. In (c), we apply the false-class margin to the normalized softmax loss; the C1 area is $\cos\theta_1 > \cos\theta_2 + m_2\cos^2\theta_2$, while the C2 area is $\cos\theta_2 > \cos\theta_1 + m_2\cos^2\theta_1$. In (d), combining the true-class margin and the false-class margin, we propose the class-variant margin softmax loss; the C1 area is $\cos\theta_1 - m_1(1-\cos^2\theta_1) > \cos\theta_2 + m_2\cos^2\theta_2$, while the C2 area is $\cos\theta_2 - m_1(1-\cos^2\theta_2) > \cos\theta_1 + m_2\cos^2\theta_1$.

the early individual saturation problem, we introduce the false-class margin to the cosine of the angle between feature vector and the false class weight vector. This approach, as shown in Fig.5(c), enables us to enlarge the margins for those samples near the saturation status, and thus to postpone the saturation process. Finally, we combine the true-class margin and the false-class margin to attain the final decision margin of our CVM loss, which addresses both the class imbalance problem and the early saturation problem, as shown in Fig.5(d).

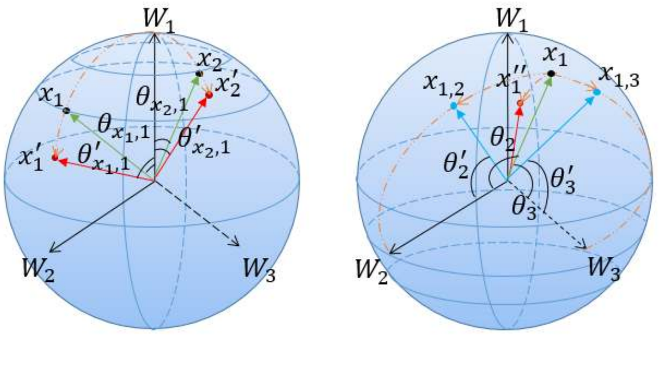


Fig. 6. Schematic diagram of adding class-variant margins to the hypersphere.

To understand the CVM loss even further in a geometric view, we also draw an illustration of adding class-variant margins in the hypersphere, as shown in Fig.6.

In the left panel of Fig.6, the original feature vectors x_1 and x_2 both belong to class 1, and the angle between x_1 and W_1 is larger than the angle between x_2 and W_1. That is, it is harder to classify the sample x_1. Therefore, we apply a larger true-class margin to the cosine of the angle $\theta_{x_1,1}$ to strengthen the impact of the sample x_1 on the training. Geometrically, through two different true-class margins, $h(\theta_{x_1,1})$ and $h(\theta_{x_2,1})$, the original feature vectors x_1 and x_2 are transformed to the new feature vectors x'_1 and x'_2, respectively; and with $h(\theta_{x_1,1}) > h(\theta_{x_2,1})$, we enhance the impact of x_1 and extract more discriminative feature.

In the right panel of Fig.6, the original feature vector x_1 belongs to class 1, and the angle θ_2 between x_1 and W_2 is larger than the angle θ_3 between x_1 and W_3, which means that x_1 to class 2 is closer to saturation than x_1 to class 3. Thus, we introduce a larger false-class margin to the cosine of the angle $\theta_{j,j\neq y_i}$ that is nearer to saturation, to obtain a smaller angle ($\theta'_2 < \theta_2$). The original feature vector x_1 is respectively transformed to two feature vectors $x_{1,2}$ and $x_{1,3}$ through false-class margins $g(\theta_2)$ and $g(\theta_3)$, respectively. With the combination of both false-class margins, a new feature vector x''_1 is actually generated to optimize the network.

2) Gradients: The CVM loss can be optimized by using standard stochastic gradient descent algorithm. To this end, we compute the back-propagation gradients for the CVM loss. The difference between the softmax loss and the CVM loss lies in f_j. Thus, it is only need to calculate f_j in forward and backward propagation is only needed. Putting Eq.(3) and Eq.(4) in Eq.(2), we can rewrite f_{y_i} and $f_{j(j\neq y_i)}$ as

$$\begin{aligned} f_{y_i} &= s\cdot\left(\cos\theta_{y_i} - m_1\cdot\left(1-\cos^2\theta_{y_i}\right)\right),\\ f_{j(j\neq y_i)} &= s\cdot\left(\cos\theta_j + m_2\cdot\cos^2\theta_j\right). \end{aligned} \tag{5}$$

For the back-propagation, we apply the chain rule to compute the partial derivative, $\frac{\partial f_{y_i}}{\partial(\cos\theta_{y_i})}$ and $\frac{\partial f_{j(j\neq y_i)}}{\partial(\cos\theta_{j(j\neq y_i)})}$ as

$$\begin{aligned} \frac{\partial f_{y_i}}{\partial(\cos\theta_{y_i})} &= s\cdot(1+2m_1\cos\theta_{y_i}),\\ \frac{\partial f_{j(j\neq y_i)}}{\partial(\cos\theta_{j(j\neq y_i)})} &= s\cdot(1+2m_2\cos\theta_{j(j\neq y_i)}). \end{aligned} \tag{6}$$

III. Experimental studies

In this section, we will introduce the experimental settings in Section A, investigate the sensitivity of parameters m_1 and m_2 in Section B, and conduct plenty of experiments in Section C on widely-used face datasets, LFW, YTF and MegaFace, to demonstrate the effectiveness of the proposed method.

A. Experimental settings

1) Training data: The Casia-Webface dataset [24] contains 0.49M images of 10,558 identities. We train the network on the cleaned version, containing 0.45M images of 10,572 identities.

2) Network settings: The CNN architecture used in our work is the SphereFace-20 network [14], which is based on 20 convolutional layers and residual units. We use PyTorch to implement the modifications on the loss function.

3) Data Pre-processing: In the data pre-processing procedure, we use MTCNN [25] (Multi-task convolutional neural network, a commonly used face detection and alignment network) to operate face detection and landmark location. First, all the training and testing images are pre-processed to extract face landmarks. Then, based on these landmarks, we make similarity transformation to align images. Finally, the original images are cropped and resized to 112×96, and each pixel is normalized by subtracting 127.5 then dividing by 128.

4) Test settings: First, we conduct experiments on the MNIST dataset [26] to investigate the effect of these two parameters m_1 and m_2. In the testing stage, features of the original image and the flipped image are concatenated together to compose the final face representation. As face features have been normalized, here we use the cosine similarity of features to measure the distance between query and gallery images in face recognition tasks. Finally, face verification and identification are conducted by thresholding and ranking the scores. We test our models on several popular public face datasets, including LFW [27], YTF [28] and MegaFace [29].

TABLE II
CNN ARCHITECTURE OF SPHEREFACE-20 NETWORK. [3X3, 64]X2, S2 MEANS 2 CASCADED CONVOLUTION LAYERS WITH 64 FILTERS OF 3X3 KERNEL SIZE, AND THE STRIDE IS 2. BESIDES, THE RESIDUAL UNIT ARE SHOWN IN DOUBLE-COLUMN BRACKETS.

Layer	SphereFace-20	params	FLOPs
Conv1.x	$[3 \times 3, 64] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 1$	0.07M	202M
Conv2.x	$[3 \times 3, 128] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	0.6M	445M
Conv3.x	$[3 \times 3, 256] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 4$	5.0M	842M
Conv4.x	$[3 \times 3, 512] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 1$	5.9M	247M
FC1	512	11M	11M

B. Effects of parameters m_1 and m_2

As shown in Eq.(3) and (4), the true-class margin function $h(\theta)$, which is aimed to address the class imbalance problem, involves a parameter m_1, while the false-class margin function $h(\theta)$, which is designed to postpone the early individual saturation, involves a parameter m_2. To investigate the sensitivity of these two parameters on the performance of the CVM loss, we conduct a series of experiments on the MNIST dataset [26].

For illustrative purposes, at different values of these two parameters from 0.1 to 0.9 in a step of 0.1, we plot in Fig.7 the average intra-class distance and inter-class distance over five runs on the MNIST dataset. From Fig.7, we can observe the followings: (i) For every m_2, a bigger m_1 decreases the intra-class distance and increases the inter-class distance. (ii) When m_1 is sufficiently small, a small m_2 will help increase the inter-class distance. When m_1 is bigger, it is also advised to have a bigger m_2 to obtain a larger inter-class distance.

C. Experiments on the LFW and YTF datasets

In this section, we evaluate the proposed CVM loss function on two face-recognition benchmark datasets: LFW and YTF [28] under the open-set protocol. The LFW dataset [27] is composed of 13,233 web-collected images from 5,749 identities, with large variations in pose, expression and illumination. The YTF dataset includes 3,425 videos of 1,595 people downloaded from Youtube, with an average of 2.15 videos per person. The duration of each video ranges from 48 to 6,070 frames, with an average of 181.3 frames per video. We follow the standard protocol of unrestricted with labeled outside data and test on 6,000 face pairs from the LFW dataset and 5,000 video pairs from the YTF dataset.

TABLE III
COMPARISON OF LOSS FUNCTIONS IN TERMS OF FACE VERIFICATION ACCURACY ON THE LFW AND YTF DATASETS.

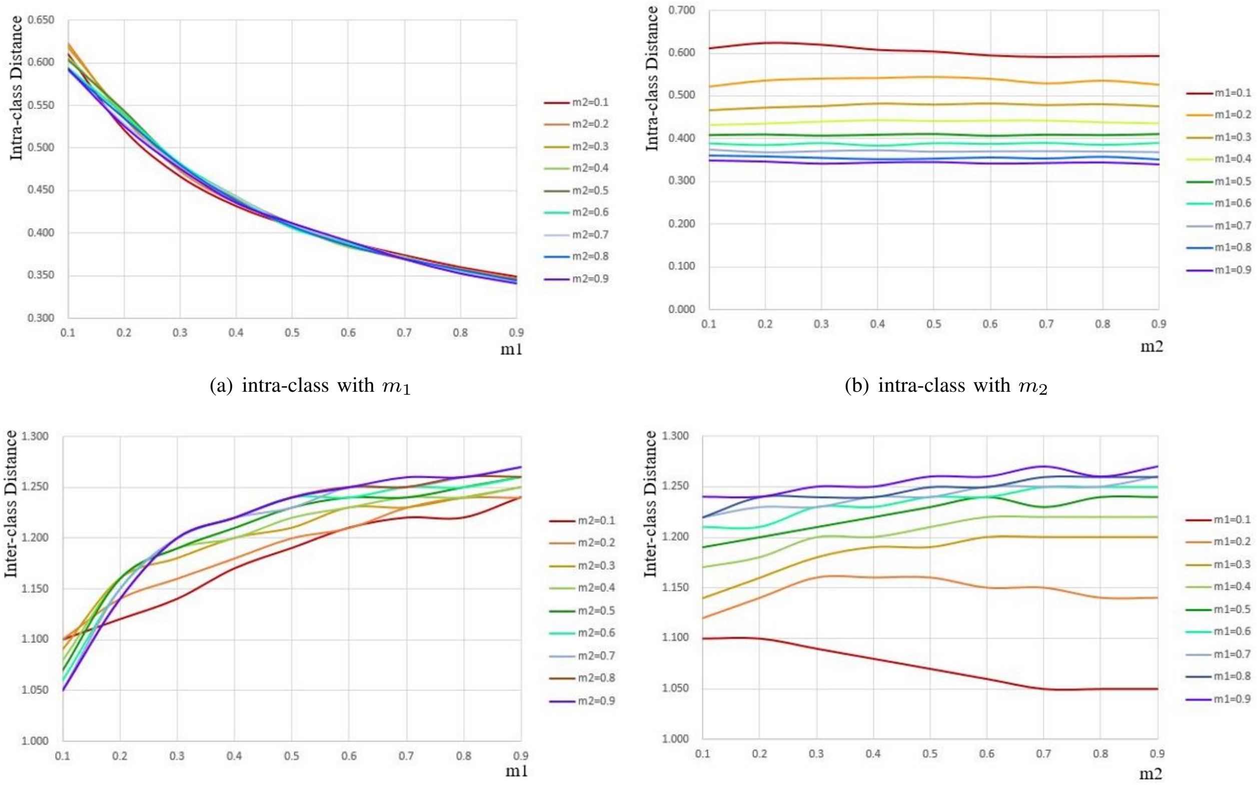
Loss function	LFW(%)	YTF(%)
Softmax loss	96.55	89.86
Normalized softmax loss	98.35	90.90
Angular softmax loss	99.15	93.56
Large margin cosine loss	99.10	92.94
Addictive angular margin Loss	99.02	92.74
CVM-softmax loss	**99.33**	**93.64**

First, we compare different loss functions (the softmax loss, the normalized softmax loss, the angular softmax loss, the large margin cosine loss, the addictive angular margin loss and our CVM softmax loss), under the same network settings and training data: we use the Casia-Webface dataset without any bells and whistles to train the SphereFace20 network. As shown in Table III, the proposed CVM loss achieves the best performance on both LFW and YTF, in particularly it outperforms the softmax loss by a significant margin, from 96.55% to 99.33% on LFW and from 89.86% to 93.64% on YTF. These results indicate that the proposed CVM loss are able to further enhance the discriminative power of the deeply learned face features.

TABLE IV
COMPARISON OF DIFFERENT FACE RECOGNITION FRAMEWORKS.

Methods	models	params	images	LFW(%)	YTF(%)
DeepID [10]	200	41.8M	-	99.47	93.20
VGG Face [30]	1	133M	2.6M	98.95	**97.30**
Deep Face [31]	3	120M	4M	98.37	91.40
Fusion [32]	5	-	500M	98.37	-
FaceNet [33]	1	140M	200M	**99.62**	95.10
Baidu [34]	1	-	1.3M	99.13	-
Range loss [17]	1	138M	1.5M	99.52	93.70
Multibatch [35]	1	1.3M	2.6M	98.80	-
Aug [36]	1	143M	0.5M	98.06	-
Center loss [13]	1	-	0.7M	99.28	94.90
Marginal loss [37]	1	-	4M	99.48	95.98
SphereFace20+CVM	1	22.5M	0.45M	99.33	93.64

Then we compare the proposed CVM loss with the state-of-the-art face recognition approaches. As can be seen in Table IV, with fewer parameters (only a single 20-layer network with 22.5M parameter size) and a smaller amount of training data (0.45M images), the CVM loss achieves competitive results to other state-of-the-art methods.



(a) intra-class with m_1

(b) intra-class with m_2

(c) inter-class with m_1

(d) inter-class with m_2

Fig. 7. Curves of intra-class distance and inter-class distance with varying m_1 and m_2.

D. Experiments on the MegaFace datasets

MegaFace is a very challenging test benchmark released for large-scale face identification and face verification. The gallery set in MegaFace consists of more than 1 million face images. The probe set contains two existing datasets: FaceScrub and FGNET. In this study, we use the FaceScrub dataset as the probe set to evaluate the performance of our proposed CVM loss. The FaceScrub dataset includes 206,863 face images from 530 celebrities. A subset composed of 3,530 images from 80 celebrities is tested for face identification and face verification. In the pre-processing procedure, we use MTCNN for face detection and alignment in the probe set and the gallery set. As faces in some images cannot be detected successfully, there are finally 961,312 face images left in the gallery set, but for the probe set, we manually crop and align those images which have failed in MTCNN.

TABLE V
MEGAFACE EXPERIMENTAL RESULTS COMPARISON.

Loss Function	MegaFace Rank1 Acc.	MegaFace Ver.
Softmax loss	43.02%	47.18%
Normalized softmax loss	48.05%	56.65%
Angular softmax loss	66.74%	73.87%
Large margin cosine loss	69.78%	75.11%
Addictive angular margin loss	64.36%	72.04%
CVM-softmax loss	**72.32%**	**79.05%**

As shown in Table V, our CVM loss not only surpasses the large margin cosine loss, which applies a fixed margin to the normalized softmax loss, but also significantly outperforms the other popular loss functions.

E. Time Complexity

The time complexity of the original softmax loss is $\mathcal{O}(n)$ given n samples, which is linear to the number of training samples. In fact, all the loss functions listed in Table III, including our CVM loss, are the modified versions based on the softmax loss and they all have $\mathcal{O}(n)$ time complexity. More specifically, compared with the normalized softmax loss, our proposed loss additionally applies dynamic margins, which only brings small burden on the training process. Moreover, the comparative experiment results listed in Table III prove the effectiveness of the proposed loss function with the same time complexity to the compared state-of-the-art methods.

IV. CONCLUSION

In this brief paper, we propose a new loss function called the class-variant margin (CVM) normalized softmax loss for deep face recognition. The proposed CVM loss introduces the true-class margin and the false-class margin to the cosine space, which can alleviate the class imbalance and softmax saturation problems in the network training. Comprehensive experiments show that the CVM loss performs better than state-of-the-art loss functions on the LFW, YTF and MegaFace datasets.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

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

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[7] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *CVPR*, 2014, pp. 1875–1882.

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

[9] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, pp. 14–003, 2014.

[10] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, 2014, pp. 1988–1996.

[11] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*. IEEE, 2005, pp. 539–546.

[12] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

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

[14] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *CVPR*, 2017.

[15] 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 *CVPR*, 2018, pp. 5265–5274.

[16] J. Deng, J. Guo, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *arXiv preprint arXiv:1801.07698*, 2018.

[17] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *CVPR*, 2017, pp. 5409–5418.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.

[19] B. Chen, W. Deng, and J. Du, "Noisy softmax: Improving the generalization ability of DCNN via postponing the early softmax saturation," in *CVPR*, 2017.

[20] L. Zhou, Z. Wang, Y. Luo, and Z. Xiong, "Separability and compactness network for image recognition and superresolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3275–3286, 2019.

[21] Y. Gao, J. Ma, and A. L. Yuille, "Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2545–2560, 2017.

[22] Z. Xiong, Z. Wang, C. Du, R. Zhu, J. Xiao, and T. Lu, "An Asian face dataset and how race influences face recognition," in *Pacific Rim Conference on Multimedia*. Springer, 2018, pp. 372–383.

[23] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.

[24] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[25] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[26] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[27] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

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

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

[30] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, 2015.

[31] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *CVPR*, 2014, pp. 1701–1708.

[32] ——, "Web-scale training for face identification," in *CVPR*, 2015, pp. 2746–2754.

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

[34] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[35] O. Tadmor, Y. Wexler, T. Rosenwein, S. Shalev-Shwartz, and A. Shashua, "Learning a metric embedding for face recognition using the multibatch method," *arXiv preprint arXiv:1605.07270*, 2016.

[36] I. Masi, A. T. Tran, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?" in *ECCV*. Springer, 2016, pp. 579–596.

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