MsrFace: Multi-Sphere Radius Loss for Deep Face Recognition

Xi Du\textsuperscript{1}, Xiaoxuan Han\textsuperscript{2,*}, Yu Qin\textsuperscript{3}, Dong Wang\textsuperscript{4}

\textsuperscript{1,2,3,4} Kiwi Technology, Nanjing, Jiangsu, China
* Corresponding author and e-mail: Xiaoxuan Han, xiaoxuan.han@kiwiar.com

Abstract. Loss functions is one of the main challenges in face recognition problems. Recent works focus on designing loss functions that make learned features more discriminative by a larger angular or cosine distance. In this paper, in addition to the method based on additional angle margins, we propose a Multi-Sphere Radius Loss (MsrFace) to add radius constraints. MsrFace pushes learned features to hyperspheres with different spherical radii and the classes can be separated more strictly. We present experiments on several widely used benchmarks to show that MsrFace has a better performance in comparison with some recent state-of-the-art face recognition methods.

1. Introduction

Face recognition has been widely used in security, finance, and identity verification due to its non-intrusiveness. Earlier methods relied on the combination of hand-crafted features and machine learning techniques, such as principal component analysis (PCA) and linear discrimination (LDA). Both PCA and LDA adopt the idea of reducing dimensional complexity of the input. However, due to the complex variance of human faces in the real world, such as posture, age, occlusion, lighting, and expression, these traditional methods are not robust enough. In recent years, deep learning method such as convolutional neural network (CNN) has been used in face recognition problems and significantly improved the performance. Face verification and face identification are the two main categories in face recognition problems. Face verification computes the similarity of two images to determine whether they belong to the same subject, while face identification computes the similarity between one image and several images to determine whether the face belongs to the specific subject. Training data, network architecture and loss function are the three main aspects that affect the performance of CNN-based face recognition. In this paper, we focus on loss function and the optimization of the network architecture.

Softmax loss has been used in early classification works. Contrastive loss and triplet loss enlarge inter-variance and compress intra-variance in Euclidean space. Center loss learns a center for each class and minimize the distance between features and centers. SphereFace penalized large angles between features in an angular space. CosFace and ArcFace further added an additive cosine/angular margin and were able to converge without softmax supervision. These angular/cosine-margin-based methods measure the angular or cosine distance on the same hypersphere. In this paper, we creatively turn our attention to the sphere radius and propose a Multi-Sphere Radius Loss (MsrFace) to further improve the discriminativeness in face recognition. MsrFace combines angles and radius metrics to push the learned features to multiple surfaces of hyperspheres with different radii. The major contributions in this paper are as follows:
1. We propose a novel loss function named MsrFace with its math formula. We further compare MsrFace with some recent loss functions based on angular/cosine margins in terms of decision boundaries and geometric interpretation.

2. We perform experiments on several widely used benchmarks to show that MsrFace can obtain better discriminative features for face recognition comparing with other recent works.

3. Our work shows that based on MsrFace, keeping the shift-invariance of the CNN architecture can reduce the impact of input shifts and achieve better verification performance.

2. Related Work

2.1. Loss Function
Softmax loss is widely used in previous works, however, it cannot ensure the compactness of intra-class and discriminativeness of inter-class, thus it is not robust in large-scale face recognition scenarios. Some researchers focused on Euclidean-distance-based loss functions such as Contractive loss [1], Triplet loss [2] and Center loss [3]. Contractive loss requires image pairs as the training data. Triplet loss defines face triplets and separates the distance between positive pairs from distance between negative pairs by a margin. The training speed of Contrastive loss or Triplet loss is much slower due to the large number of pairs or triplets and it is hard to choose margin parameters and construct training samples. Center loss provides a class center for each category and minimizes the distance between features and their corresponding class center, which fulfills the target of reducing intra-class distance.

In recent years, some angular/cosine-margin-based methods have been proposed. L-Softmax introduced margins between classes, forcing the angle between the weights and features corresponding to each class to be increased to m times the original angle, so the distance between classes can be pulled apart. SphereFace [4] normalized the weights on the basis of L-Softmax and proposed a multiplicative margin, but this makes network training more difficult. CosFace [5] and ArcFace [6] introduced an additive margin on angular and cosine respectively, they are easy to implement and able to converge without tricky hyper-parameters.

2.2. Network Architecture
Deep learning method based on convolutional neural network (CNN) allows training on large amounts of input data to learn better feature representations. In 2012, AlexNet [7] used an 8-layer CNN to win the ImageNet competition. Since then, more deeper networks have been proposed. In 2014, DeepFace [8] achieved an accuracy of 97.35% on the LFW benchmark, which improved the state-of-the-art performance by 27%. The representative networks in recent years are VGGNet [9], GoogleNet [10], ResNet [11], SENet [12], etc. Among them, ResNet was proposed in 2016. By learning the residual mapping between the input and the output, the degradation problem of the deep network is effectively alleviated, and deeper network can be used for face recognition. ResNet is also widely used as the backbone network for feature extraction nowadays. In our experiments we employ ResNet100 as the embedding network.

In recent years, some traditional image recognition ideas have been introduced into the optimization of CNN. For instance, Richard Zhang [13] proposed to add a low-pass filter before subsampling to provide anti-aliasing capability. Subsampling methods such as max-pooling and average-pooling make CNNs lose shift-invariance. Therefore, a slight translation or shift of the input will greatly affect the output. Our experiments show that by integrating low-pass filter with max-pooling layer in ResNet100 boost verification accuracy in several widely used benchmarks.

3. Proposed Approach

3.1. Multi-Sphere Radius Loss (MsrFace)
The most widely used softmax loss can be presented as follows:
\[
L_{\text{softmax}} = -\frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{e^{\theta_i W^T x + b_i}}{\sum_{j=1}^{N} e^{\theta_j W^T x + b_j}} \right)
\]  

(1)

where \( x_i \in \mathbb{R}^d \) denotes the deep feature of the \( i \)-th samples, belonging to the \( y_i \)-th class. \( W_i \in \mathbb{R}^{d \times n} \) denotes the \( j \)-th column of the weights in the last fully connected layer. \( b_i \in \mathbb{R}^n \) is the bias term. \( N \) is the batch size. \( n \) is the class number. Softmax loss does not necessarily ensure the inter-class diversity and intra-class similarity. Thus, it is not sufficient when dealing with large-scale face recognition scenarios. ArcFace is an angular-margin-based loss which is shown in equation (2):

\[
L_{\text{arcface}} = -\frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{e^{\cos(\theta_i + m)}}{\sum_{j \neq i} e^{\cos\theta_j}} \right) + \frac{1}{M} \sum_{j=1}^{M} \sum_{i=1}^{N} \left( ||x_i||_2 - C_{y_i} - m \right)
\]

(2)

The learned features lie on a hypersphere of a fixed radius \( s \). If we expand \( \cos(\theta + m) \) to \( \cos \theta \cos m - \sin \theta \sin m \), it is discovered that the additive angular margin \( \cos(\theta + m) \) is simpler than \( \cos \theta - m \). ArcFace introduced an additive margin instead of multiplicative margin to discriminate different classes. The feature vectors are projected onto the hypersphere with a radius length of \( s \).

We propose to project features onto hyperspheres with different radii and add a new loss term based on the original ArcFace loss function. The MsrFace loss function is presented as follows:

\[
L_{\text{MSR}} = L_{\text{arcface}} + \frac{1}{N} \sum_{i=1}^{N} \left( ||x_i||_2 - C_{y_i} \right) + \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1, j \neq y_i}^{M} \left( ||x_i||_2 - C_{y_j} - m \right)
\]

(3)

Vector \( x_i \) is the feature of \( i \)-th sample in the \( y_i \)-th class. \( x_i \) is projected onto the hypersphere manifold with radius \( C_{y_i} \), on which all the vectors of the same class are compacted. \( M \) denotes the number of all class labels except \( y_i \)'s class. \( C_{y_j} \) denotes the \( j \)-th radius in \( M \). \( m \) is a constant value which is configured before training. When \( m \) becomes larger, the distance between different classes’ sphere radii become larger, and the inter-class distances is increased. The gradient of MsrFace is computed as follows:

\[
\frac{\partial L_{\text{MSR}}}{\partial x_i} = \frac{\partial L_{\text{arcface}}}{\partial x_i} + \frac{1}{N} \sum_{i=1}^{N} \left( ||x_i||_2 - C_{y_i} \right) \frac{\partial ||x_i||_2}{\partial x_i}
\]

\[
+ \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1, j \neq y_i}^{M} \left( m - ||x_i||_2 + C_{y_j} \right) \frac{\partial ||x_i||_2}{\partial x_i}
\]

\[
= \frac{\partial L_{\text{arcface}}}{\partial x_i} + \frac{2}{N} \sum_{i=1}^{N} \left( ||x_i||_2 - C_{y_i} - m \right) x_i
\]

\[
+ \frac{2}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1, j \neq y_i}^{M} \left( m - ||x_i||_2 + C_{y_j} \right) x_i
\]

(4)

The added loss term is independent of angular margin in ArcFace loss and the implementation of this part consumes little computing resources. We summarize the algorithm of MsrFace as follows:

Step 1: Initialize parameters \( \{\theta_i\} \) in convolution layers, parameters \( W \) and different radii for different labels \( \{C_j \mid j = 1, 2, ..., n\} \) in loss layers, respectively;

Step 2: If the network is not converged, \( t = t + 1 \);

Step 3: Compute MsrFace loss as:
\[ L_{\text{MSR}}^{t} = L_{\text{arcface}}^{t} + \frac{1}{N} \sum_{i=1}^{N} \left( \|x_i\|_2 - C_{y_i} - m \right) + \frac{1}{N \cdot M} \sum_{j=1}^{N} \sum_{k=1}^{M} \left( m - \|x_i\|_2 + C_{y_i} \right) \]

Step 4: Compute the backpropagation error \( \frac{\partial L_{\text{MSR}}}{\partial x_i} \) for each \( i \);

Step 5: Update parameter: \( W : W^{t+1} = W^t - \mu \cdot \frac{\partial L_{\text{MSR}}}{\partial W^t} \);

Step 6: Update the radius for each label: \( C_{y_i}^{t+1} = C_{y_i}^t + \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} \|x_i\|_2 - C_{y_i}^t \);

Step 7: Update: \( \theta_i : \theta_i^{t+1} = \theta_i^t - \mu \sum_{i=1}^{N} \frac{\partial L_{\text{MSR}}}{\partial \theta_i^t} \cdot \frac{\partial x_i^t}{\partial \theta_i^t} \);

Step 8: Continue 1-7 until converge.

3.2. Comparison with Other Losses

In MsrFace, we add a sphere radius margin in addition to additive angular margin. The geometry interpretation of ArcFace and MsrFace is shown in figure 1. We further compare the decision boundary under binary classification scenario for SphereFace, CosFace, ArcFace and MsrFace which is presented in table 1.

![Figure 1](image-url)

Figure 1. Geometry interpretation of ArcFace and MsrFace. (a) ArcFace loss. (b) MsrFace loss. (c) ArcFace and MsrFace in 3D hypersphere manifold respectively.

| Loss Function | Decision Boundaries |
|---------------|---------------------|
| Softmax       | \((W_1 - W_2)x + b_1 - b_2 = 0\) |
| SphereFace    | \(x (\cos m\theta_1 - \cos \theta_1) = 0\) |
| CosFace       | \(s(\cos \theta_1 - m - \cos \theta_2) = 0\) |
| ArcFace       | \(s (\cos (\theta_1 + m) - \cos \theta_2) = 0\) |
| MSR           | \(\|x_1\|_2 - \|x_2\|_2 = 0\) |
4. Keep Shift-invariance for CNNs
In our experiments, we follow [13] to replace MaxPool as MaxBlurPool in ResNet100. MaxPool layers are divided into two steps shown in figure 3. The first step is densely evaluated max-pooling, which preserves the shift-equivariance. The second step is subsampling, which violates the shift-equivariance. The main idea is to add a low-pass filter that serves as an anti-alias filtering between these two steps.

5. Experiment and Analysis
5.1. Data Sets and Data Processing
In our experiments we select VGGFace2 [14] and MS1MV2 as the training data. MS1MV2 is a refined version of MS-Celeb-1M [15] dataset. We compare performance of different implementations on four benchmarks, namely LFW [16], AgeDB-30 [17], CFP-FP [18] and IJB-C [19]. Dataset details are shown in table 2. MTCNN is used to obtain the position of the face and key points (eyes, nose, and corners of the mouth). After face alignment, the image is cropped into size of 112 * 112 pixels.
5.2. Experimental Conditions and Settings
The experiments are performed on a machine with an Intel i7 CPU and 8 NVIDIA GeForce RTX 2080Ti (11G). The operating system is Ubuntu 18.04. The program is implemented with MXNet. We adopt memonger [20] as the memory optimization technique to make the best use of GPUs. Take ResNet100 as an example, the memory compression ratio of network architecture before the fully connected layer fc7 is 62.37%. Thus, the batch size during training can be increased.

| Data Set   | Number of class | Number of image/videos |
|------------|-----------------|------------------------|
| VGGFace2   | 9.1K            | 3.3M                   |
| MS1MV2     | 85K             | 5.8M                   |
| LFW        | 5,749           | 13,233                 |
| AgeDB-30   | 568             | 16,488                 |
| CFP-FP     | 500             | 7000                   |
| IJB-C      | 3,531           | 148.8K                 |

Table 2. Basic information of public data sets.

| Method     | Accuracy (%) |
|------------|--------------|
| LFW        | AgeDB-30     | CFP-FP       |
| MsrFace    | 99.78        | 98.32        | 98.56        |
| ArcFace    | 99.77        | 98.28        | 98.27        |
| CosFace    | 99.51        | 95.44        | 94.56        |
| SphereFace | 99.11        | 91.70        | 94.38        |
| Softmax    | 99.08        | 92.33        | 94.39        |

Table 3. Test results of MsrFace model on LFW, AgeDB-30, CFP-FP.

5.3. Experiment Conclusions

5.3.1. Comparison of Sphere Radius Distribution
We compare sphere radii of 200 classes when training with Arcface and MsrFace. The result is shown in figure 4. It is clear to notice that the radii are concentrated in Arcface. In MsrFace, the radii are sparser. The features of different classes are pushed onto hyperspheres with different radii, which makes the classification more discriminative.

Figure 4. Radius distribution of 200 classes, sphere radii are normalized.
5.3.2. Experiment Results on LFW, AgeDB-30 and CFP-FP
We train our MsrFace model on ResNet100 with VGGFace2 and MS1MV2 dataset. For fair comparison, we employ the same embedding network and training dataset for ArcFace, CosFace and SphereFace. The accuracy results on LFW, AgeDB-30 and CFP-FP are shown in Table 3. MsrFace achieves best performance on these three benchmarks.

5.3.3. Experiment Results on IJB-C
In this experiment, we employ VGG2 as the training dataset for each method. We compare the performance on IJB-C (The IARPA Janus Benchmark-C face challenge) dataset. The original IJB-C dataset contains 138000 face images, 11000 face videos and 10000 non-face images. By checking the whole dataset, we found 1487 non-face images and relabel 5097 face images manually. The refined IJB-C contains 140327 images with 3539 classes. In order to fully evaluate the performance, we compare faces with different media id (from different video source) in every class for intra-class comparison. For inter-class comparison, we construct class pairs firstly. Then two random media ids are selected respectively for the pairs. All faces with chosen ids are compared. Finally, we perform 4.924393 million intra-class comparisons and 581.172488 billion inter-class comparisons. The ROC curve is shown in Figure 5. MsrFace-SI refers to the implementation of ResNet100 with MaxBlurPool layers. In Table 4, we compare TARs with different FARs (from 1e-7 to 0.1). MsrFace-SI and MsrFace achieve better TARs on IJB-C dataset.

|                | FAR@1e-7 | FAR@1e-6 | FAR@1e-5 | FAR@1e-4 | FAR@1e-3 | FAR@1e-2 | FAR@1e-1 |
|----------------|----------|----------|----------|----------|----------|----------|----------|
| MsrFace-SI     | 0.0145   | 0.2210   | 0.6026   | 0.8280   | 0.9113   | 0.9541   | 0.9844   |
| MsrFace        | 0.0144   | 0.2189   | 0.5967   | 0.8200   | 0.9024   | 0.9448   | 0.9748   |
| ArcFace        | 0.0390   | 0.2005   | 0.5881   | 0.8195   | 0.9012   | 0.9427   | 0.9721   |
| CosFace        | 0.0446   | 0.1392   | 0.2987   | 0.4719   | 0.6433   | 0.7998   | 0.9279   |
| SphereFace     | 0.0009   | 0.0037   | 0.0261   | 0.2085   | 0.4985   | 0.7295   | 0.8877   |

Figure 5. ROC curves of MsrFace, ArcFace, CosFace and SphereFace on IJB-C.

6. Conclusion
In this paper, we propose a novel loss function named MsrFace, which adds a radius margin to the angular-margin-based loss. By learning angular and radius margins at the same time, normalized features are more compacted intra-class and discriminative inter-class. It is easy to implement MsrFace and converge during the training stage. Experiments on several widely used benchmarks show that MsrFace has a better performance in comparison with some recent state-of-the-art face recognition methods.
References

[1] Hadsell, R., Chopra, S., and LeCun, Y. 2006. June dimensionality reduction by learning an invariant mapping. 2006 IEEE Computer Society Conference on CVPR'06 Vol. 2 1735-42

[2] Schroff, F., Kalenichenko, D., and Philbin, J. 2015 facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on CVPR'15 815-23

[3] Wen, Y., Zhang, K., Li, Z., and Qiao, Y. 2016, October a discriminative feature learning approach for deep face recognition. European conference on computer vision Springer, Cham 449-515

[4] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. 2017 sphereface: Deep hypersphere embedding for face recognition. Proceedings of the IEEE conference on CVPR'17 212-20

[5] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z. and Liu, W. 2018 cosface: Large margin cosine loss for deep face recognition. Proceedings of the IEEE Conference on CVPR'18 5265-74

[6] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. 2019 arcface: Additive angular margin loss for deep face recognition. Proceedings of the IEEE Conference on CVPR'19 4690-99

[7] Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012 imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 1097-1105

[8] Taigman, Y., Yang, M., Ranzato, M. A., and Wolf, L. 2014 deepface: Closing the gap to human-level performance in face verification. Proceedings of the IEEE conference on CVPR'14 1701-08

[9] Simonyan, K., and Zisserman, A. 2014 very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. 2015 going deeper with convolutions. Proceedings of the IEEE conference on CVPR'15 1-9

[11] He, K., Zhang, X., Ren, S., and Sun, J. 2016 deep residual learning for image recognition. Proceedings of the IEEE conference on CVPR'16 770-8

[12] Hu, J., Shen, L., and Sun, G. 2018 squeeze-and-excitation networks. Proceedings of the IEEE conference on CVPR'18 7132-41

[13] Zhang, R. 2019 making convolutional networks shift-invariant again. arXiv preprint arXiv:1904.11486

[14] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. 2018, May vggface2: A dataset for recognising faces across pose and age. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018) 67-74

[15] Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. 2016, October ms-celeb-1m: A dataset and benchmark for large-scale face recognition. European conference on computer vision Springer, Cham. 87-102

[16] Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. 2008, October labeled faces in the wild: A database forstuding face recognition in unconstrained environments

[17] Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., and Zafeiriou, S. 2017 agedb: the first manually collected, in-the-wild age database. Proceedings of the IEEE Conference on CVPR'17 Workshops 51-9

[18] Sengupta, S., Chen, J. C., Castillo, C., Patel, V. M., Chellappa, R., and Jacobs, D. W. 2016, March frontal to profile face verification in the wild. 2016 IEEE WACV 1-9

[19] Maze, B., Adams, J., Duncan, J. A., Kalka, N., Miller, T., Otto, C., Jain, A., Niggel, W., Anderson, J., Cheney, J. and Grother, P. 2018, February iarpa janus benchmark-c: Face dataset and protocol. 2018 ICB 158-165

[20] Chen, T., Xu, B., Zhang, C., and Guerstrin, C. 2016 training deep nets with sublinear memory cost. arXiv preprint arXiv:1604.06174