A Comprehension of the Design of Loss Function in Face Recognition

Yong-Tao Ge
School of Automation, Southeast University
Nanjing 210096, China
geyongtao@seu.edu.cn

Abstract

Over the last few years, deep artificial neural networks have gotten the most attention in computer science, especially in pattern recognition and machine learning. One of its excellent applications is face recognition. The face recognition task usually boils down to computing the distance between face vectors of different people. Ideal face features are expected to have smaller maximal intra-class distance than minimal inter-class distance under a suitably chosen metric space. This paper reviews serval state-of-the-art metric methods with different type loss functions and discuss some challenges remain unsolved in face recognition.

1. Introduction

Recent years have witnessed the great success of convolutional neural networks (CNNs) in face recognition (FR)[8, 11, 16, 13]. Typically, face recognition can be categorized as face verification and face recognition. The former determines whether a pair of faces belongs to the same identity, while the latter classifies a face to a specific identity.

The conventional face recognition pipeline consists of four stages: face detection, face alignment, feature extraction (or face representation) and classification. Perhaps the single most important stage is feature extraction.

Pioneering work [13, 11] learn face features via the widely-used softmax loss, but softmax loss only learns separable features that are not discriminative enough. To address this, some methods combine softmax loss with contrastive loss [10, 12] or center loss [14] to enhance the discrimination power of features. [9] adopts triplet loss to supervise the embedding learning, leading to state-of-the-art face recognition results. However, center loss only explicitly encourages intra-class compactness. Both contrastive loss [2] and triplet loss [9] can not constrain on each individual sample, and thus require carefully designed pair/triplet mining procedure, which is both time-consuming and performance-sensitive.

Most recent approaches [10, 14, 12] combine Euclidean margin based losses with softmax loss to construct a joint supervision. In [4, 3], the authors introduced an conceptually appealing angular margin to push the classification boundary closer to the weight vector of each class. [3] also provided a theoretical guidance of training a deep model for metric learning tasks using the classification loss functions. [5] also improved the softmax loss by incorporating differnet kinds of margins.

The rest of this paper is structured as follows. Section 2 presents some respresentive loss functions in face recognition and makes a comparsion of them. Section 3 explains some tricks used in face recogniton. Finally, conclusion is given in section 4.

2. Representive Loss Fuctions

2.1. Baseline:Softmax Loss Function

The most widely used classification loss function, Softmax loss, is presented as follows:

$$\mathcal{L}_{\mathcal{S}} = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{1}$$

where $x_i \in \mathbb{R}_d$ denotes the deep feature of the i-th samples, belonging to the y_i-th class. $W_j \in \mathbb{R}_d$ denotes the j-th column of the weights $W \in \mathbb{R}_{d\times n}$ in the last fully connected layer and $b \in \mathbb{R}_n$ is the bias term. The batch size and the class number is m and n, respectively. Traditional Softmax loss is widely used in deep face recognition . However, the Softmax loss function does not explicitly optimise the features to have higher similarity score for positive pairs and lower similarity score for negative pairs, which leads to a performance gap. Most recent approaches [11, 10, 9, 14, 15] combine Euclidean margin based losses with softmax loss to construct a joint supervision.

2.2. Euclidiean Distance Based Loss

How to develop an effective loss function to improve the discriminative power of the deeply learned features? Intuitively, minimizing the intra-class variations while keeping the features of different classes separable is the key. [14] propose the center loss function, as formulated in Eq.

$$\mathcal{L}_{intra} = \frac{1}{2}\sum_{i=1}^{m}\|x_i - c_{y_i}\|_2^2 \quad (2)$$

$$\mathcal{L}_C = \mathcal{L}_S + \mathcal{L}_{intra} \quad (3)$$

Another representative euclidiean distance based loss is range loss[15], it encourages both intra-class and inter-class compactness by adding constrains on intra-class and inter-class. Range loss can be formulated as:

$$\mathcal{L}_R = \mathcal{L}_S + \alpha\mathcal{L}_{intra} + \beta\mathcal{L}_{inter} \quad (4)$$

$$\mathcal{L}_{R_{intra}} = \sum_{i\subseteq I}\mathcal{L}^i_{R_{intra}} = \sum_{i\subseteq I}\frac{k}{\sum_{j=1}^{k}\frac{1}{D_j}} \quad (5)$$

Where I denotes the complete set of classes/identities in this mini-batch. D_j is the j-th largest distance.

$$\begin{aligned}\mathcal{L}_{R_{inter}} &= max(m - \mathcal{D}_{Center}, 0)\\ &= max(m - \|\overline{x}_{\mathcal{Q}} - \overline{x}_{\mathcal{R}}\|_2^2)\end{aligned} \quad (6)$$

where, D_{Center} is the shortest distance between class centers, that are defined as the arithmetic mean of all output features in this class. In a mini-batch, the distance between the center of class Q and class R is the shortest distance for all class centers. m denotes a super parameter as the max optimization margin that will exclude D Center greater than this margin from the computation of the loss.

Constract loss and triplet loss are also very representative among all the euclidiean loss functions. Both of them need to select propriate sample pairs. In triplet loss, if inappropriate anchor points selected, the network may fail to convergence. For contrast loss, if a third class lies between two negative sample pairs, the distance between the inter-class sample becomes smaller. Both contrastive loss and triplet loss can not constrain on each individual sample, and thus require carefully designed pair/triplet mining procedure, which is both time-consuming and performance-sensitive. Triplet loss wants to ensure that an image x_i^a (anchor) of a specific person is closer to all other images x_i^p (positive) of the same person than it is to any image x_i^n (negative) of any other person. Thus:

$$\|f(x_i^a)\|_2^2 + \alpha < \|f(x_i^a - x_i^n)\|_2^2 \quad \forall(f(x_i^a), f(x_i^p), f(x_i^n) \in \mathcal{T} \quad (7)$$

where α is a margin that is enforced between positive and negative pairs. $\mathcal{T}$ is the set of all possible triplets in the training set and has cardinality N. Triplet loss can be formulated as:

$$\sum_{i}^{N}[\|\|f(x_i^a\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+ \quad (8)$$

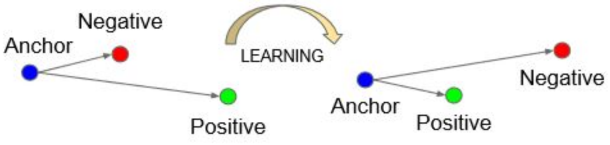


Figure 1. triplet loss

2.3. Angular/Cosine Distance Based Loss

Cosine similarity (CS) between two vectors x and y is defined as:

$$CS(x,y) = \frac{x^T y}{\|x\|\|y\|} \quad (9)$$

Cosine similarity[6] has a special property that makes it suitable for metric learning: the resulting similarity measure is always within the range of -1 and $+1$.

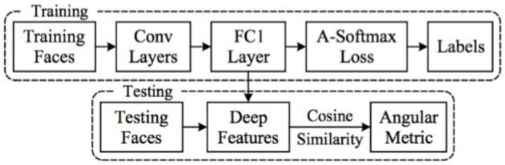


Figure 2. The Learning and Inference Pipeline of Sphereface

It seems to be a widely recognized choice to impose Euclidean margin to learned features, but a question arises: Is Euclidean margin always suitable for learning discriminative face features? In some sense, Euclidean margin based losses are incompatible with softmax loss, so it is not well motivated to combine these two type of losses. [3] propose to incorporate angular margin instead.

$$\psi(\theta_{y_i,i}) = (-1)^k\cos(m\theta_{y_i,i}) - 2k \quad (10)$$

where $\theta_{yi,i} \in [0, \frac{\pi}{m}]$. In order to get rid of this restriction and make it optimizable in CNNs, the denition range of $m\theta_{y_i,i}$ is expanded by generalizing it to a monotonically decreasing angle function $\psi(\theta_{y_i,i})$ which should be equal to $\cos(\theta_{y_i,i}) \in [0,\pi]$. Therefore, A-Softmax loss is formulated as:

$$L_{ang} = \frac{1}{N}\sum_i -\log(\frac{\exp(\|x_i\|\psi(\theta_{yi,i}))}{\exp(\|x_i\|\psi(\theta_{yi,i}))+\sum_{j\neq y_i}\exp(\|x_i\|\cos(\theta_{j,i}))}) \quad (11)$$

in which $\psi(\theta_{y_i,i}) = (-1)^k\cos(m\theta_{y_i,i}) - 2k, \theta_{yi,i} \in [0, \frac{\pi}{m}]$ and $k \in [0, m-1]$. $m1$ is an integer that controls the size of angular margin. When $m = 1$, it becomes the modied softmax loss.

Geometry Interpretation of A-Softmax Loss:A-Softmax is based on the assumption that different classes are located in different areas of the surface of a unit of hypersphere. From the above it can also be known that its geometric meaning is represented by the weight of the unit in the surface of the superficial point in the training process, the same type of input maps to the surface will slowly move to the center, weights of different class slowly dispersed. The size of m is to control the degree of aggregation of the same kind of point, thus controlling the distance between different classes. Note that larger m leads to smaller hypercircle-like region for each class, which is an explicit discriminative constraint on a manifold. For better understanding, Fig.3 provides 2D and 3D visualizations. One cansee that A-Softmax loss imposes arc length constrainton a unit circle in 2D case and circle-like region constraint on a unit sphere in 3D case. Our analysis shows that optimizing angles with A-Softmax loss essentially makes the learned features more discriminative on a hypersphere.

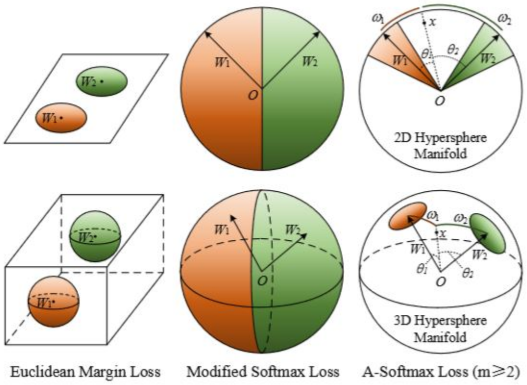


Figure 3. Geometry Interpretation of Euclidean margin loss(e.g. contrastive loss, triplet loss, center loss, etc.), modied softmax loss and A-Softmax loss. The rst row is 2D feature constraint,and these condrow is 3D feature constraint. The orange region indicates the discriminative constraint for class 1, while the green region is for class 2.

3. Tricks Used in Face Recognition

3.1. The Problem of Imblanced Training Data

Abundant training data and well-designed training strategies are indispensable for effective deep face models. However, many large scale face datasets exhibit long-tail distribution where a small number of entities have large number of face images while a large number of persons only have very few face samples (long tail).

3.2. Zeroing out the Biases in Softmax layer

Standard CNNs usually preserve the bias term in the fully connected layers, but these bias terms make it difficult to analyze the angular margin of the proposed A-Softmax loss. To facilitate the analysis, [3] zero out the bias of FC2. By setting the bias of FC2 to zero, the A-Softmax loss has clear geometry interpretation and therefore becomes much easier to analyze. Fig. 5 shows all the biases of FC2 from a CASIA-pretrained model. One can observe that the most of the biases are near zero, indicating these biases are not necessarily useful for face verification. Futhermore. The authors in [3] visualize the 2D feature distribution in MNIST dataset with and without bias in Fig. 6. One can observe that zeroing out the bias has no direct influence on the feature distribution. The features learned with and without bias can both make full use of the learning space.

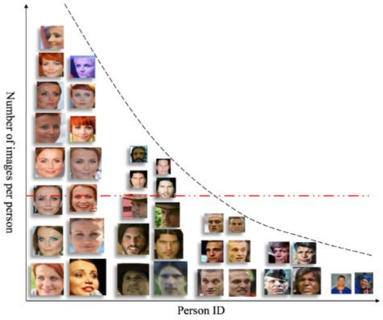


Figure 4. Long Tailed Training Data

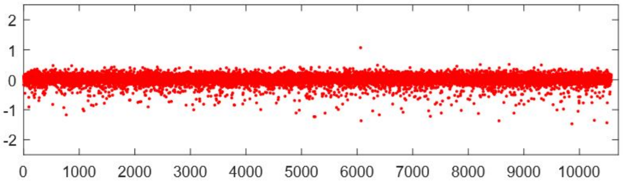


Figure 5. SphereFace:Biases of last fully connected layer learned in CASIA-WebFace dataset.

3.3. What is weight normalization and Feature Normation?

As is discussed in section 3.1, bias have litte effect on the performance of the face recognition. Let bias $b_j = 0$ and fix$\|W_j\| = 1$ (softmax layer) by L2 normalization, the target logit can be transformed as follows:

$$W_j^T x_i = \|W_j\|\|x_i\|\cos\theta_j + bj = \|x_i\|\cos\theta_j \qquad (12)$$

Futhermore, fix $\|x_i\|$ by L2 normalisation to push every feature to distribute on a hypersphere manifold. That is so called feature normalization.

$$W_j^T x_i = \|x_i\|\cos\theta_j = \cos\theta_j \qquad (13)$$

3.4. Why does weigth normalization work?

In paper[1], the author have studied the problem of one-shot face recognition, by creating a benchmark dataset consisting of 20,000 persons(each id has 50-100 images) for face feature learning and 1,000 persons(each id has 20 images) for one-shot learning. The paper reveals that the norm of the weight vector is proportional to the number of face

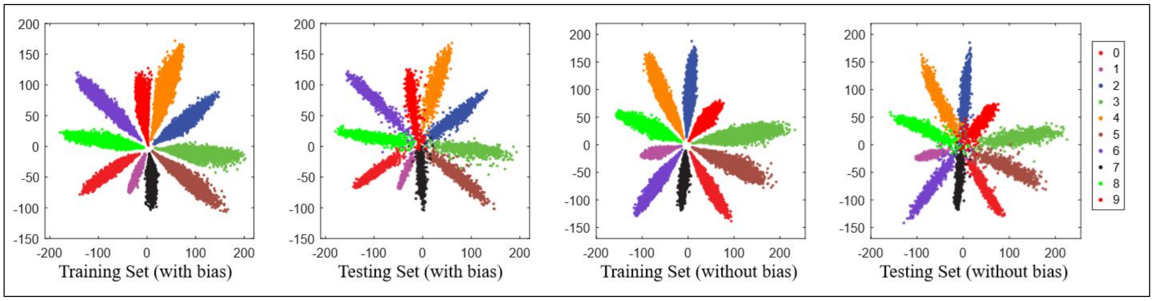


Figure 6. 2D visualization with and without bias of last fully connected layer in MNIST.

pictures with the same id, and propose a novel loss called underrepresented-classes promotion to effectively address the data imbalance problem in one-shot learning. The evaluation results on the benchmark dataset show that the new loss termbringsa significant gain by improving the recognition coverage rate from 25.65% to 77.48% at the precision of 99% for one-shot classes, while still keeping an overall accuracy of 99.8% for normal classes.

[1] have an empirical study on the relation between the sample number of each class and the 2-norm of the weights corresponding to the same class (the i-th column of W is associated to the i-th class). By computing the norm of W_i and sample number of class i with respect to each class (see Fig.7), the authors find that the larger sample number a class has, the larger the associated norm of weights tends to be. It can be argued that the norm of weights W_i with respect to class i is largely determined by its sample distribution and sample number. Therefore, norm of weights W_i ,$\forall i$ can be viewed as a learned prior hidden in training datasets. Eliminating such prior is often beneficial to face verification. This is because face verification requires to test on a dataset whose idenities can not appear in training datasets, so the prior from training dataset should not be transferred to the testing stage. This prior may even be harmful to face verification performance. To eliminate such prior, normalize the norm of weights of FC2 is neccessary(see Fig.8).

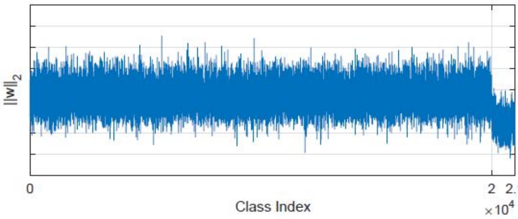


Figure 7. Without weight normalizaiton

In the experiments of SphereFace[3], L2 weight normalisation only improves little on performance. But the author emphasized that normalizing the weights can give better geometric interpretation. Besides this, we also justify why we want to normalize the weights from a different perspective. We find that normalizing the weights can implicitly reduce the prior brought by the training data imbalance issue (*e.g.*, the long-tail distribution of the training data). In other words, the authors argue that normalizing the weights can partially address the training data imbalance problem.

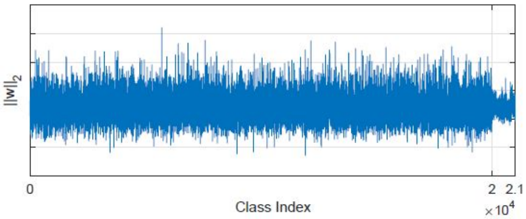


Figure 8. With weight normalizaiton

3.5. Is Feature Normalization Necessary?

Feature normalisation is widely used for face verification, *e.g.* L2-normalised Euclidean distance and cosine distance [6]. Parde et al.[7] observe that the L2-norm of features learned using Softmax loss is informative of the quality of the face. Features for good quality frontal faces have a high L2-norm while blurry faces with extreme pose have low L2-norm. Fig.10 shows that the L2-norm of features tend to be larger if a image contains more information about a face. Based on that, it is not necessarily to do feature normalization if the dataset has abundunt low-quality faces. That's why many existing works simply cutting the low-quality data for uniform distributions across the classes. Feature normalization may have little effect on improving face recongition performance. As a conclusion, feature normalization is most suitable for tasks whose image quality is very low.

4. Conclusion

There is still lots of potentials for the research of the large margin strategies which can maximum intra-class distance and minimum inter-class distance. SphereFace[3] is a pioneering work in face recognition. However, there could

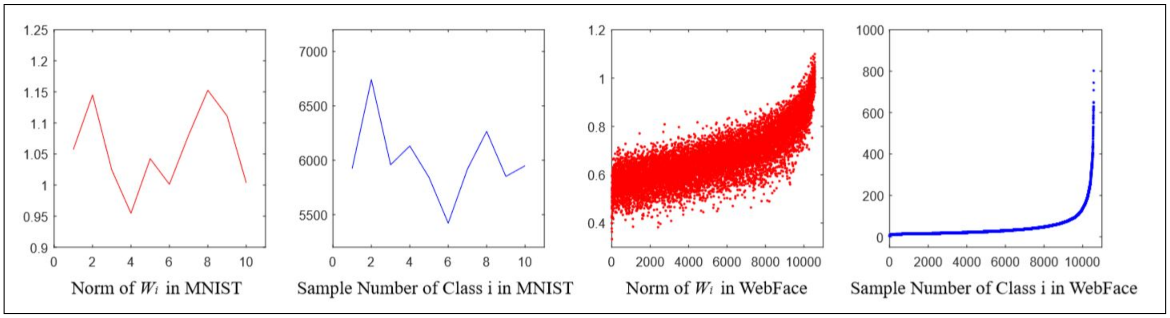


Figure 9. Norm of W_i and sample number of class i in MNIST dataset and CASIA-WebFace dataset.

Method	Data	LFW	YTF
Softmax Loss	$WebFace$	97.88	93.1
Softmax Loss+Contrastive	$WebFace$	98.78	93.5
Triplet Loss	$WebFace$	98.70	93.4
Softmax Loss+Center Loss	$WebFace$	99.05	94.4
SphereFace	$WebFace$	**99.42**	**95.0**

Table 1. Accuracy(%) on LFW and YTF dataset

be more creative way of specifying the function $\psi(\theta)$ other than multiplication and addition. How to automatically determine the margin and how to incorporate class-specific or sample-specific margins remain open questions and are worth studying.

References

[1] Y. Guo and L. Zhang. One-shot face recognition by promoting underrepresented classes. 2017.

[2] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[3] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746, 2017.

[4] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. *international conference on machine learning*, pages 507–516, 2016.

[5] Y. Liu, H. Li, and X. Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *neural information processing systems*, 2017.

[6] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV'10 Proceedings of the 10th Asian conference on Computer vision - Volume Part II*, pages 709–720, 2010.

[7] C. J. Parde, C. D. Castillo, M. Q. Hill, Y. I. Colon, S. Sankaranarayanan, J.-C. Chen, and A. J. O'Toole. Deep convolutional neural network features and the original image. *arXiv preprint arXiv:1611.01751*, 2016.

[8] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[9] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[10] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. *neural information processing systems*, pages 1988–1996, 2014.

[11] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[12] Y. Sun, X. Wang, and X. Tang. Sparsifying neural network connections for face recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4856–4864, 2016.

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[14] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515, 2016.

[15] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tail. 2016.

[16] E. Zhou, Z. Cao, and Q. Yin. Naive-deep face recognition: Touching the limit of lfw benchmark or not? *arXiv preprint arXiv:1501.04690*, 2015.

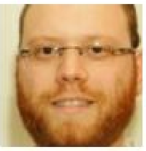
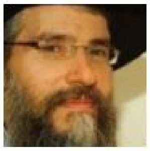
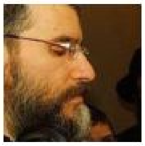
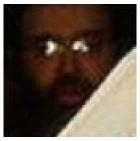
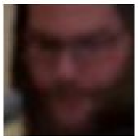
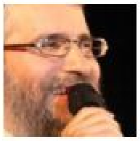
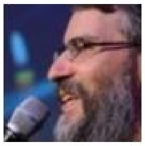
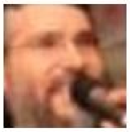
Img_id	img	Feature_norm
0001_01.png		26.0379
0001_02.png		21.901
0011_03.png		15.2779
0011_04.png		5.76883
0038_02.png		2.42962
0027_01.png		18.6879
0043_01.png		16.8457
0052_05.png		11.057

Figure 10. Relationship of Face Picture and Feature Norm