Journal of Information Hiding and Privacy Protection
DOI:10.32604/jihpp.2021.016835
Review

Survey on the Loss Function of Deep Learning in Face Recognition

Jun Wang[1], Suncheng Feng[2,*], Yong Cheng[3] and Najla Al-Nabhan[4]

[1]Director of Science and Technology Industry Department, Nanjing University of Information Science & Technology, Nanjing, China
[2]School of Computer & Software, Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, China
[3]Science and Technology Industry Department, Nanjing University of Information Science & Technology, Nanjing, China
[4]Deptratment of Computer Science, King Saud University, Riyadh, Saudi Arabia
[*]Corresponding Author: Suncheng Feng. Email: 20191220015@nuist.edu.cn
Received: 13 January 2021; Accepted: 28 March 2021

Abstract: With the continuous development of face recognition network, the selection of loss function plays an increasingly important role in improving accuracy. The loss function of face recognition network needs to minimize the intra-class distance while expanding the inter-class distance. So far, one of our mainstream loss function optimization methods is to add penalty terms, such as orthogonal loss, to further constrain the original loss function. The other is to optimize using the loss based on angular/cosine margin. The last is Triplet loss and a new type of joint optimization based on HST Loss and ACT Loss. In this paper, based on the three methods with good practical performance and the joint optimization method, various loss functions are thoroughly reviewed.

Keywords: Loss function; face recognition; orthogonality loss; ArcFace; the joint loss

1 Introduction

Let the machine have human intelligence, so as to realize artificial intelligence is the dream that human beings have been pursuing for a long time. Despite the rapid development of computer science in the past decades, the research and development of artificial intelligence is still very limited. With the development of neuroanatomy, the technical means to observe the microstructure of the brain have become increasingly rich, and the understanding of the morphology, structure and activity of the brain organization has become more and more profound, and the mystery of the information processing of the human brain has been gradually revealed. The research goal of "brain-like intelligence" is how to use the research achievements of neuroscience, brain science and cognitive science to study the information representation, transformation mechanism and learning rules of the brain, establish an intelligent computing model that simulates the process of brain information processing, and ultimately enable machines to master the cognitive rules of human beings. In recent years, brain-like intelligence has become a hot topic in research and competition all over the world. Following the United States and the European Union, a large number of researchers and research institutions at home and abroad are also making efforts to promote the neural and brain-like computing related research, large-scale "brain-like intelligence" research is poised for development. Face recognition and human action recognition [1] based on deep learning are the research hotspots.

Starting in 1940, M-P neurons [2] and Hebb learning rule [3], to the 1950s Hodykin-Huxley equation, the perceptron model and adaptive filter, and then to 1960s of the self-organizing map network, neurocognitive machine, adaptive resonance network, many neural computational model is developed as signal processing, computer vision, natural language processing and optimization calculation in areas such as the classic method, greatly promote the rapid progress in the field of neural network. At present, neural

network has developed hundreds of models and achieved very successful applications in such technical fields as handwriting recognition [4][5], image annotation [6] and semantic understanding [7]–[8].

With the improvement of the recognition accuracy of the neural network of deep learning [9]–[11] in various fields, more and more excellent loss functions emerge continuously. Start from the simplest 0–1 Loss function, because the 0–1 loss function is too idealistic, strict, and the mathematical properties is not very good, it is difficult to optimize problem, derived the perceptron loss, it for predicting result is not like a 0–1 loss function must be 0 or 1, but give an error range, as long as within the error range, is believed to be correct, and with the square of the difference between predicted values and real values as quadratic loss function loss function and a logarithmic as the log of loss function loss and so on.

For different application fields, different loss functions may produce completely different effects, so it is extremely important to design targeted loss functions. With the wide development and application of face recognition, many excellent designed to applied to face recognition of the loss function are constantly emerging, in the optimization of the distance between the class and class distance has made constant progress, the accuracy of face recognition neural network, and it has been far more than the human recognition accuracy is 97.53% [13], the latest face recognition accuracy was more than 99%, this is largely due to the loss function of optimization, so until now, there are many researchers devote themselves to the application of neural network in various fields of loss function design and optimization work.

2 Related Work of Loss Function in Face Recognition

In this section, this paper mainly introduces the defects of the application of traditional loss function in the field of face recognition and the requirements of designing an excellent neural network for face recognition.

2.1 Face Recognition

Early algorithms are based on geometric features algorithm, based on template matching algorithm, subspace algorithm and other types. Masi et al. [14] summarizes the important algorithms and comparisons in the DeepFace field. Gutta, etc., [15] proposed the hybrid neural network, such as Lawrence [16] by a multistage SOM sample clustering, the convolutional neural network (CNN) [17] is used in face recognition, Lin [18], such as the neural network method based on probabilistic decision, Demers, etc., principal component neural network method is proposed to extract the face image feature, using autocorrelation neural network further compression characteristics, finally MLP is used to realize face recognition. Er et al. used PCA [19] for dimension compression, then extracted features with LDA [20], and then performed face recognition based on RBF. Haddadnia et al. [21] used RBF neural network based on PZMI features and hybrid learning algorithm for face recognition. The advantage of neural network is to acquire the recessive expression of these rules and rules through the learning process, and it has strong adaptability. With the rapid development of deep learning and its strong learning ability, DeepFace [12] and DeepID [13] obtained the best effect on the LFW data set, surpassing humans for the first time in unconstrained scenarios, and the mainstream algorithms of face recognition also changed from traditional algorithms based on geometric features to algorithms based on neural networks. A complete and mainstream face recognition network is shown in Fig. 1.

FR differs from the general image classification task because of the natural particularity of human faces, that is, the intra-class gap is small and the inter-class gap is large. These problems also encourage a large number of researchers to work on the structure of neural network and loss function in the field of face recognition, which greatly improves the discriminability and generalization of depth model. At the same time, a large number of face training data sets and methods have emerged.

Images of the same person may look very different under different external conditions, such as posture, lighting and occlusion, resulting in large intra-class gaps and inter-class similarities. Therefore, how to narrow the intra-class gap and expand the inter-class gap is the key direction of face recognition research.

FR differs from the general image classification task because of the natural particularity of human faces, that is, the intra-class gap is small and the inter-class gap is large. These problems also encourage a large number of researchers to work on the structure of neural network and loss function in the field of face recognition, which greatly improves the discriminability and generalization of depth model. At the same time, a large number of face training data sets and methods have emerged.

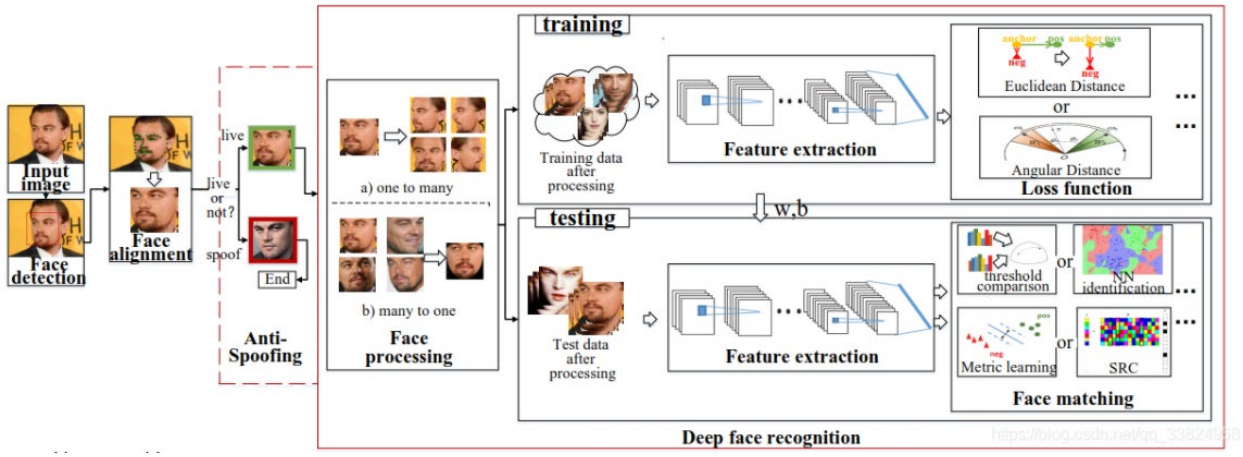


Figure 1: A complete and mainstream face recognition network [14]

Images of the same person may look very different under different external conditions, such as posture, lighting and occlusion, resulting in large intra-class gaps and inter-class similarities. Therefore, how to narrow the intra-class gap and expand the inter-class gap is the key direction of face recognition research.

In order to achieve better face recognition performance, the work mainly has two aspects, focusing on the construction of network structure [22]–[24] (such as VGGNet, GoogLeNet, ResNet) and loss function design. It is very important to construct an efficient learning loss function for face recognition. Softmax loss is able to solve classification problems directly. However, the disadvantage of Softmax is that it only encourages feature differentiation and does not work well in the face space. As an alternative, comparative loss [25] and triplet loss [26] constructed image pair and triad loss functions respectively. The DeepID [27] network was trained by combining classification and verification loss.

With the rapid progress and widespread popularity of GPU devices, the precision of face recognition based on deep learning method is also constantly improving, and has been widely applied in the real world.

2.2 Basic Concepts of Loss Function

Loss function (Cost function) is a function that maps a random event in real life into a non-negative real number in a European space, the loss of the random event is represented. The loss function can be a tool to better reflect the gap between the predicted value of the neural network and the actual tag, and the understanding of the loss function can better analyze and learn the subsequent optimization tools (gradient descent, etc.).

2.3 The Defects of Traditional Loss Function

The traditional loss function has a good effect in image processing, however, due to the traditional loss like softmax loss in the field of face recognition, at the time of convergence quickly, but the accuracy will not rise again in about 90%. On the one hand, it cannot be metric softmax as explicit learning optimization between class and class in the distance, so performance is not very good. Softmax encourages the real target category output to be larger than other categories, but does not require it to be much larger. For feature embedding of face recognition, softmax encourages separation of features of different categories, but does not encourage much separation of features, in addition, getting feature generalization ability is strong is the key to face recognition, and classification ability is not completely equivalent. Therefore, designing loss

functions with better performance in optimizing inter-class distance and intra-class distance is the most important task in the design of loss functions in the field of face recognition.

2.4 Design Criteria of Loss Function in Face Recognition Field

Face recognition task, which is different from traditional image recognition neural network, general test sets are much larger than the training set, and require training set and testing set does not overlap, so in the design of a great loss of face recognition function, as well as considering the optimization of the inter-class, as introduced in Section 2.4.1, also want to consider the optimization of the intra-class distance, as introduced in Section 2.4.2.

2.4.1 The Optimization of the Inter-Class

Because of face recognition is an important task to be distinguish different faces, so in the mapping characteristics of space, the distance between the different persons should be expanded, as far as possible. Sphereface, ArcFace, etc., are used, which will feature vector into the cosine vector, through training the different categories of cosine vector orthogonal to enlarge the distance between the classes.

2.4.2 The Optimization of the Inter-Class

Another face recognition task is the same kind of input image is to be able to identify the same class, so in the feature space, the distance between the vector of the same person should be as narrow as possible, center loss is based on the original loss function and the square of the distance within a class item as a function of punishment, to achieve the purpose of reduce the intra-class gap.

3 Excellent Loss Function in Face Recognition

This article focuses on four losses that practical perform very well in human face recognition and is widely used in the loss of function, they include the loss that add a penalty term method to optimize the original function method and the loss based on angular/cosine margin, so as to realize enlarge the distance between the classes at the same time to reduce the gap in the class of the optimization goal.

3.1 The Loss that Add a Penalty Term Method

In-line equations/expressions are embedded into the paragraphs of the text. For example, $\mathrm{E} = mc^2$. In-line equations or expressions should not be numbered and should use the same/similar font and size as the main text.

3.1.1 Center Loss

Center Loss [28] is an optimization of the original softmax loss function, it adds the square sum of the distance between samples and similar samples into the original loss function as a penalty term, as shown in formula (1):

$$L = L_s + \mu L_c = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} ||x_i - c_{y_j}||_2^2 \quad (1)$$

wherein, L_s is the traditional Softmax Loss, L_c is the penalty item in Center Loss, and μ is the penalty coefficient. The specific gravity between Softmax Loss and the penalty item can be controlled by adjusting the size of λ. Softmax cross entropy is responsible for increasing inter-class distance, while center-Loss is responsible for reducing intra-class distance, so that the learned feature discrimination degree will be higher. The selection of λ is also very important for the whole neural network. Because if λ is too small, it cannot narrow the inter-class distance, while λ is too large to cause the problem of underfitting.

Through data distribution visualization technology, we obtained data distribution diagrams under different λ through experiments, as shown in Fig. 2. It is obvious that Center Loss has a good effect on narrowing the in-class distance.

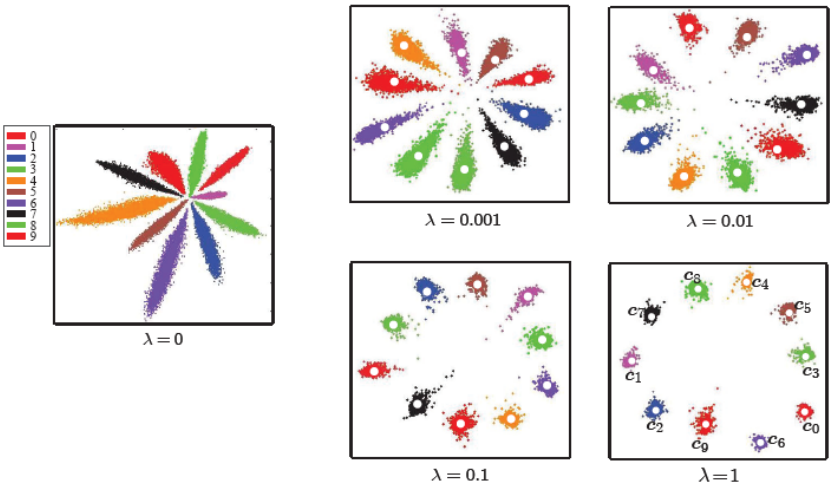


Figure 2: The impact of different λ on the data distribution in the Center Loss model [28]

3.1.2 Orthogonality Loss

Orthogonality Loss [29] was proposed in 2020, it makes a reasonable theoretical analysis of the effect of angular/cosine edge Loss on the increase of inter-class distance, and then proposes the Orthogonality Loss according to the characteristics of random vector distribution in high-dimensional feature space. It enhances the recognition ability of deep face features by punishing the mean value and second moment of the weight matrix of the generated feature vectors. Orthogonality Loss divides the entire Loss function into three parts, one is the traditional Softmax Loss function, and the other two are added penalty terms. As shown in Fig. 3, the axis of symmetry of the L_{fsm} constraint cosine similarity curve is around 0. However, it is not enough to distinguish W, in this case, the second sample moment is considered as another loss (L_{ssm}), which makes the cosine similarity curve steeper. L_{fsm} combined with L_{ssm} can provide a better minimum gradient and help the network to meet the constraint of weight vector being close to orthogonal in high dimension.

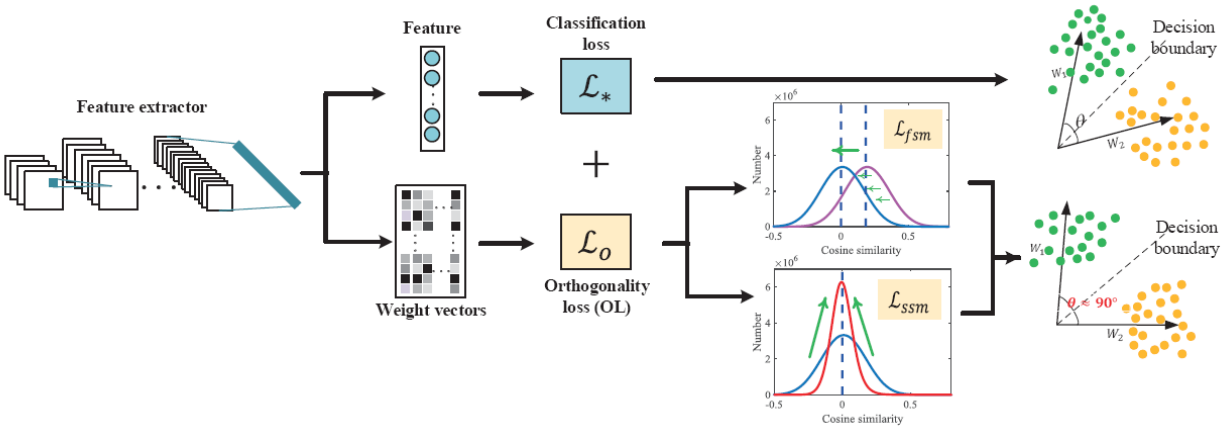


Figure 3: The overview of Orthogonality loss framework [29]

The mathematical formula for this is formula (2):

$$L_{\boldsymbol{Orthogonality}} = L_s + \mu L_{fsm} + \sigma L_{ssm} \tag{2}$$

Among them:

$$L_{fsm} = E^2(w^T w) = (\frac{2}{n(n-1)} \sum_{0 \le i < j \le n} w_i^T w_j)^2 \quad (3)$$

$$L_{ssm} = E((w^T w)^2) = \frac{2}{n(n-1)} \sum_{0 \le i < j \le n} (w_i^2 w_j)^2 \quad (4)$$

where, n represents the number of categories in the sample, and w represents the weight matrix.

According to the chain rule, the network can easily calculate the back propagation flow of orthogonal loss. Since this is a convex optimization problem, we use standard stochastic gradient descent to optimize the orthogonal loss. This paper summarizes the calculation process of loss value and gradient value in the algorithm, such as Algorithm 1.

Algorithm 1 Orthogonality Loss

Input: Training Data $\{x_i\}$, the last fully connected layer weights $W \in R^{d*n}$, hyperparameter α, β, learning rate μ.

Output: W

1. **while** not converged **do**
2. $t = t + 1$
3. Calculate the first loss $Loss_{fsm}^t = E^2(w^T w)$
4. Calculate the second loss $Loss_{ssm}^t = E((w^T \mathrm{w})^2)$
5. Calculate the joint loss $Loss_{Or}^t = Loss_*^t + \alpha Loss_{fsm}^t + \beta Loss_{ssm}^t$
6. Calculate the backpropagation process $\frac{\partial Loss_{Or}^t}{\partial w_i^t}$ for each i by $\frac{\partial Loss_{Or}^t}{\partial w_i^t} = \frac{\partial Loss_*^t}{\partial w_i^t} + \frac{\partial Loss_{fsm}^t}{\partial w_i^t} + \frac{\partial Loss_{ssm}^t}{\partial w_i^t}$
7. Update the parameters w_i by $w_i^{t+1} = w_i^t - \mu \frac{\partial Loss_{Or}^t}{\partial w_i^t}$
8. **return**

Orthogonality Loss has achieved superior performance in feature extraction compared with normal orthogonal methods, and the weights of the last full connection layer in the network represent the center of category. Softmax loss was used to optimize the sample features and make them approximate to the weight vector. Therefore, regularization of weight vectors is very significant for increasing the distance between classes. In Tab. 1.

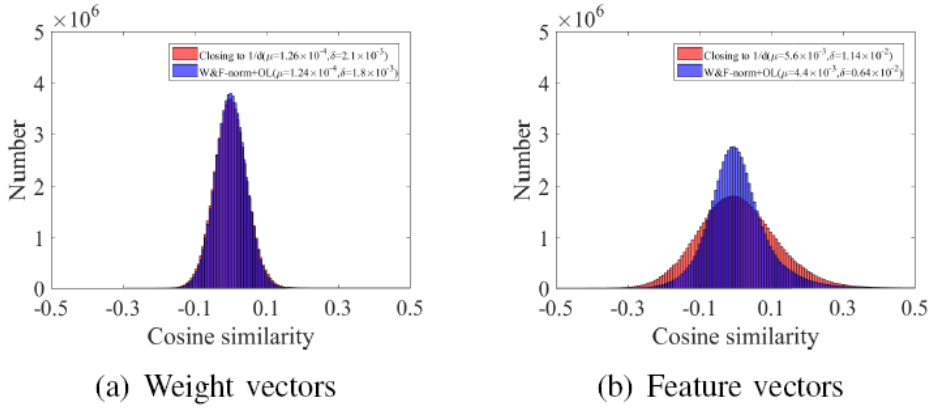
Table 1: Performance comparison of whether Orthogonality Loss should be applied [29]

Different Loss	IJB-A:Verif		IJB-A:Identif	
	0.01	0.1	Rank1	Rank5
LSFS [30]	73.3	89.5	82.0	92.9
Triplet [31]	79.0	89.5	88.0	05.0
W&F-norm [32]	88.54	95.41	90.06	93.96
W&F norm+OL	90.10	95.95	90.72	94.36
SphereFace [33]	90.34	96.21	91.13	94.83
SphereFace+OL	93.26	96.76	93.31	95.60
CosFace [34]	94.28	97.07	93.58	95.88
CosFace+OL	94.56	97.40	93.69	95.70
ArcFace [35]	93.68	96.75	93.22	95.51
ArcFace+OL	94.15	97.26	93.50	95.70

For the current mainstream facial recognition neural network structure, Orthogonality Loss has been applied, and the effect has been significantly improved. Fig. 4(b) also indicates that Orthogonality Loss

guarantees more Orthogonality of feature vectors. During each iteration, depth feature regularization punishes only 256 sample features, while Orthogonality Loss can simultaneously optimize up to five thousands weight vectors to obtain more accurate gradient directions.

For visual understanding, the histogram of cosine similarity between the weight vector and the feature vector is plotted in Fig. 4. The Orthogonality Loss method is used to approximate to $1/d$ to obtain the histogram of cosine similarity, and it is found that the histogram is sharper at this time, which means that our method is more likely to make the weight vector close to orthogonal.



(a) Weight vectors (b) Feature vectors

Figure 4: Performance comparison of whether Orthogonality Loss should be applied. By approximating $\frac{1}{d}$, the Orthogonality Loss has the weight vector shown in (a) and the cosine similarity histogram of the feature vector shown in (b). μ and $\sigma$are the mean and second order values of the cosine similarity[29]

3.2 The Loss Based on Angular/Cosine Margin

In the discriminative feature of human faces, our goal is to obtain discriminative feature. Since most of the discriminative features of face recognition are open-set identification, the training set cannot include all the faces. Therefore, at this time, margin is needed to implement the perfect constraint, namely, the maximum inter-class distance < the minimum inter-class distance.

The reason why intra-class aggregate classes are separated is that, for example, when the loss value drops to a certain fixed value during training, the e index terms with and without Margin are equal, so $\theta_{y_i}$with Margin needs to be reduced relatively. In this way, the training with Margin reduces the Angle between the input features and weights of category i. From the graphs of some angles, it can be seen that Margin squeezes θ_{y_i}into more classes and aggregates it, thus making θ_{y_i} more separate from other classes.

Sections 3.2.1–3.2.4 are introduced in this paper based on this idea of loss functions.

3.2.1 L-Softmax

Cross entropy loss and softmax are probably the most widely used loss functions in convolutional neural networks. Although relatively simple, popular and effective, it does not significantly encourage the learning of distinguishing features. Thus, Large-Magrin Softmax loss (LMSL, L-softmax) loss is proposed, which promotes the intra-class compactness and interclass separability between learning features to a great extent. In addition, L-softmax is able to adjust the required borders and avoid overfitting. We also show that L-softmax losses can be optimized for typical stochastic gradient descent. A lot of experiments show that the L-softmax lost feature with deep learning has stronger recognition ability, which significantly improves the performance of network for image classification and verification.

The author proposes that the motive of Large-Magrin Softmax loss (LMSL, L-softmax) [32] is to generate a decision margin by adding a margin that is positive, which can more strictly restrict the original formula, such as formula (5):

$$||W_1||||x||cos\theta_1 \geq ||W_1||||x||cos(m\theta_1) \geq ||W_2||||x||cos\theta_2 \quad (5)$$

When learning similar samples, we deliberately enhance the difficulty of similar learning, which is more difficult than that of different types. This kind of distinction treatment makes the feature distinguishability enhanced.

According to the idea in the previous section, L-Softmax loss can be written as formula (6):

$$Loss_{L-soft} = -\frac{1}{2N}\sum_i \frac{e^{||W_{y_i}||||x_i||\varphi(\theta_{y_i})}}{e^{||W_{y_i}||||x_i||\varphi(\theta_{y_i})} + \sum_{j\neq y_i} e^{||W_j||||x_i||\varphi(\theta_j)}} \quad (6)$$

In order to simplify forward and backward propagation, an approximate function formula (7) is constructed to replace $\varphi(\theta)$.

$$\omega(\theta) = (-1)^n \cos(m\theta) - 2n, \quad \theta\epsilon[\frac{n\pi}{m}, \frac{(n+1)\pi}{m}] \quad (7)$$

We applied the L-Softmax loss function to the neural network and trained and tested it on MINST. By visualizing the process, we could obtain the data distribution like Fig. 5.

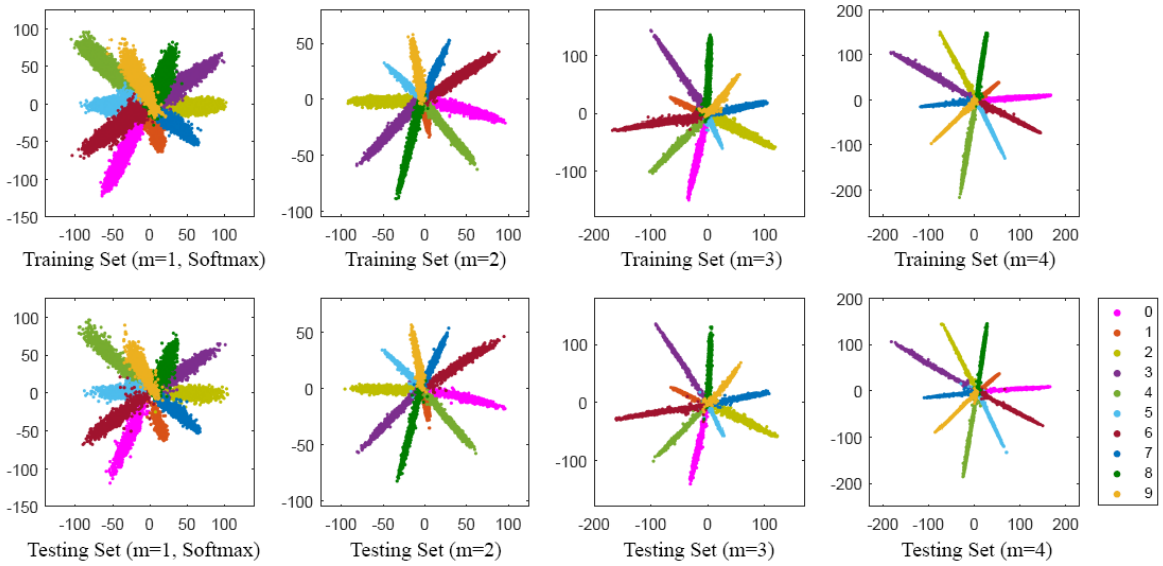


Figure 5: Comparison of distributions of different Margins on training sets and test sets [32]

Among them, the learning features L-softmax becomes more compact and well separated.

3.2.2 Sphereface

Sphereface(Angular softmax,A-softmax) [33] achieves its goal of narrowing the in-class distance while widening the inter-class distance by transforming the weight matrix and introducing margin methods. Sphereface first converts the original weight matrix into an included Angle by normalizing W and letting bias set zero, so that all features after the mapping are on the same hypersphere. At this point, the loss function is shown in formula (8):

$$L_{mod} = \frac{1}{N}\sum_i -log\left(\frac{e^{||x_i||cos(\theta_{y_i,i})}}{\sum_j e^{||x_i||cos(\theta_{j,i})}}\right) \quad (8)$$

At this point, the optimal inner product has changed to the optimal Angle, so the decision boundary of the category is completely determined by the Angle. The decision boundary becomes more concise and clear, and is also more in line with the image interpretation of the hypersphere. However, this optimization is not enough. In order to further narrow the distance between classes and expand the distance between classes, Sphereface introduces the idea of Margin, and the following formula is shown in formula (9):

$$L_{sphere} = -\frac{1}{N}\sum_i \log\left(\frac{e^{||x_i||cos\,(m\theta_{y_i,i})}}{e^{||x_i||cos\,(m\theta_{y_i,i})}+\sum_{j\neq y_j} e^{||x_i||cos\,(\theta_{j,i})}}\right) \quad (9)$$

It is worth noting that in order to ensure the monotonically decreasing nature of the $\cos(\theta)$ function, we use formula (10) as the approximate function of the $\cos(\theta)$.

$$\omega(\theta) = (-1)^n \cos(m\theta) - 2n, \quad \theta\epsilon[\frac{n\pi}{m}, \frac{(n+1)\pi}{m}] \quad (10)$$

The loss function of A-Softmax was applied to MNIST, and its two-dimensional characteristic distribution was visualized as shown in Fig. 6. It is obvious that when margin is larger, the discriminability of learning features will be stronger due to the larger inter-class angular margin.Most significantly, the feature distinctions it learns can be well applied to other data sets.

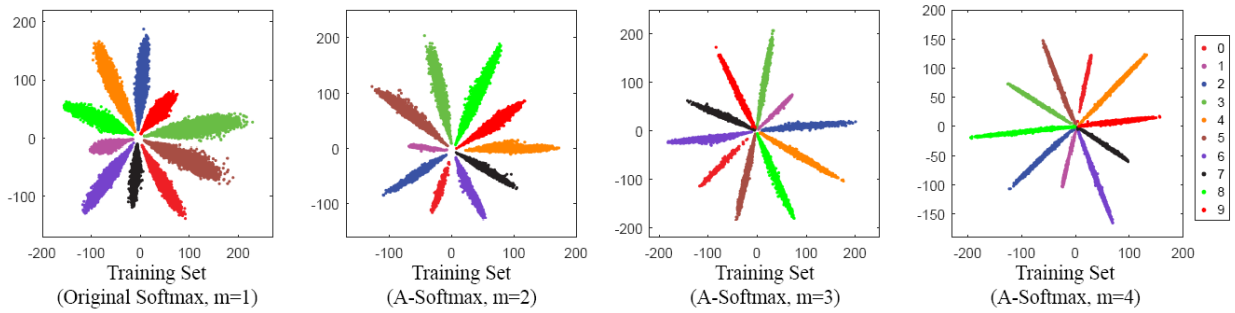


Figure 6: Comparison of data distributions in different margin between Original and A-softmax [33]

3.2.3 CosFace

CosFace (Large Margin Cosine Loss, LMCL) [34] is a further optimization of Sphereface, it respectively for W and x do L2 Normalization, the norm of them is 1, let all input after mapping space in the same hypersphere， but with x after Norm may appear too small, leading to softmax value is too small, eventually leading to training when the loss is too big, so the need for a scaling, fixed for s.

After this, the purpose of this article is from the Angle of inner product to optimize, but learning to feature still is separable, haven't reach the discriminative features. So we introduced a cosine margin constraint to measure, so that the category of the current sample still belonged to this category after subtracting a magin. Such as formula (11):

$$\cos(\theta_1) - m > \cos(\theta_2) \quad (11)$$

Then loss is shown in formula (12) [34]:

$$L_{cosine} = \frac{1}{N}\sum_i -log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)}+\sum_{j\neq y_i} e^{s(\cos(\theta_{j,i}))}} \quad (12)$$

among them:

$$W = \frac{W^*}{||W^*||} \quad (13)$$

$$x = \frac{x^*}{||x^*||} \quad (14)$$

$$\cos(\theta_j, i) = W_j^T x_i \quad (15)$$

We used a mini-experiment with Eight two-dimensional identities. The pictures in first row maps the feature to the Euclidean space, and the other row represents the feature to the angular space. With the increase of guarantee margin, this gap becomes obvious, as shown in Fig. 7.

The idea of CosFace is similar to SphereFace. It eliminates radial changes through L2 normalization features and weight vectors, and introduces margin, which improves the effect of the real model to a certain extent.

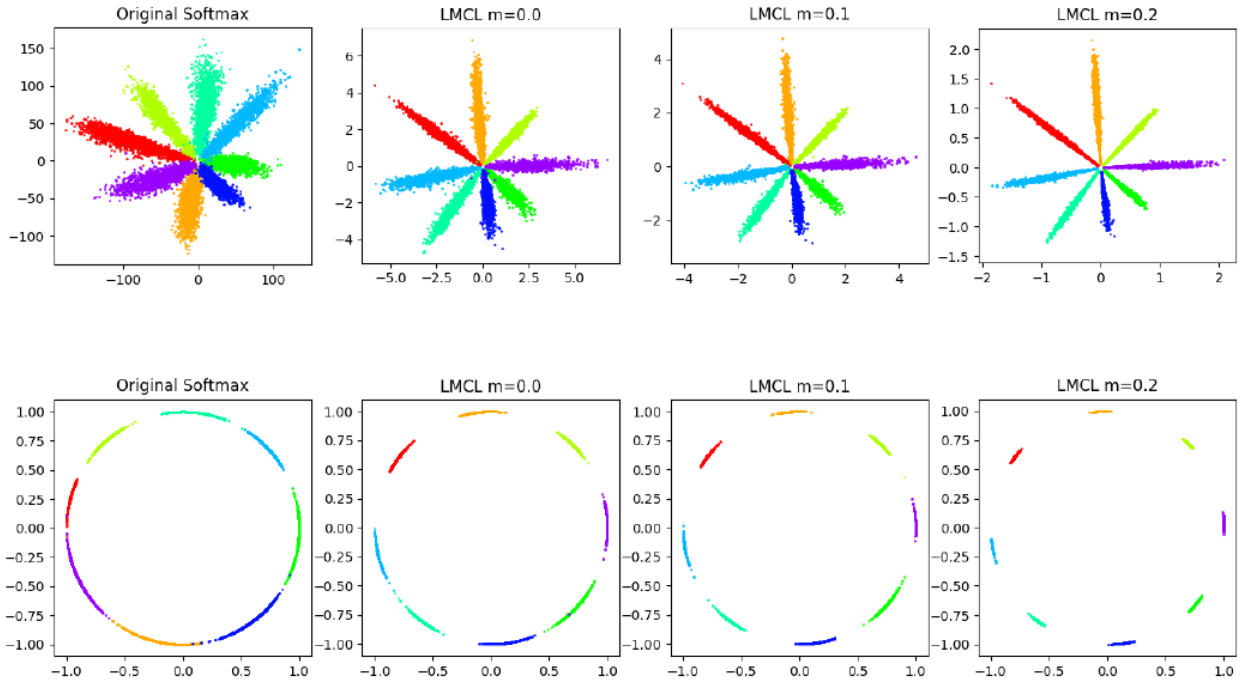


Figure 7: A toy experiment of LMCL [34]

3.2.4 ArcFace

ArcFace [35] was published by Imperial College London on January, 2018. ArcFace is an improved algorithm based on SphereFace, which significantly expand the distance of inter-class and decrease the intra-class distance by using the normalization and additive Angle interval of eigenvectorss.

Additive Angular Margin Loss is an Additive Angular Margin Loss that directly normalizes the eigenvectors and weights. An Angle interval m is added to theta, making the Angle interval more direct than the cosine interval affects the Angle. The biggest difference and optimization between ArcFace and Cosine loss was that ArcFace maximized classification limits directly in the angular space, while CosFace maximized classification limits in the Cosine space. From the perspective of hypersphere space, the angles in the normalized hypersphere correspond directly to radians, so it is more effective to optimize the angles directly. Meanwhile, ArcFace does not need to be optimized jointly with other original loss functions, which increases its stability and easy convergence, so ArcFace is more direct and effective. The ArcFace formula is shown in formula (16) [35]:

$$L_{arcface} = -\frac{1}{N}\sum_i log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j\neq y_i} e^{s(cos\theta_j)}} \tag{16}$$

The specific ArcFace network structure is shown in Fig. 8.

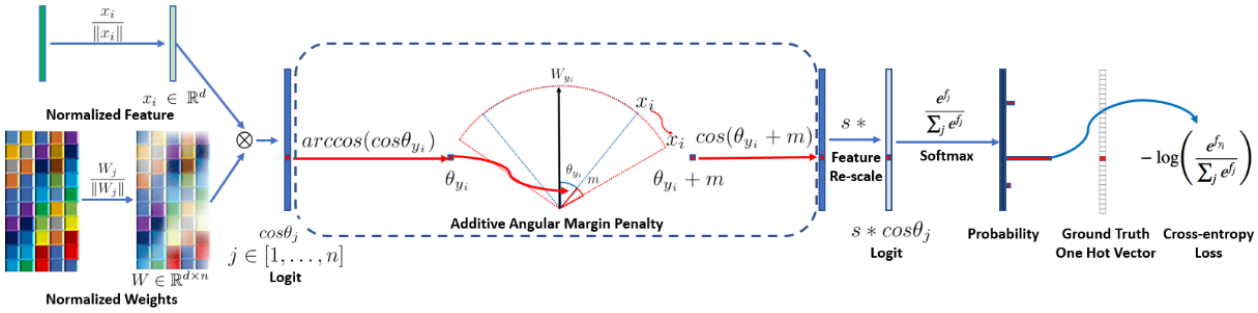
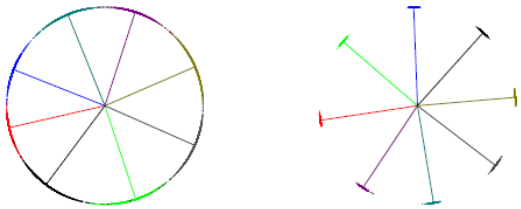


Figure 8: The network structure of ArcFace [35]

As shown in Fig. 2, firstly, the input weight matrix W was normalized and the feature x_i was normalized and scaled. Based on W_j and x_i, logit ($cos\theta_j$) could be obtained, and then the Angle θ_{y_i} between the feature and weight matrix was obtained by using the function $arccos\theta$. Then we add an angular distance penalty$m$on Angle θ_{y_i}, by characteristic scaling s， we multiply cos $(\theta_{y_i} + m)$ and all logits, and finally using softmax to obtain the cross entropy loss.

ArcFace is easy to program and easy to apply, so it can be easily applied to most existing mature networks to further improve performance. We use MxNet to write the pseudo-code of ArcFace, which is very easy to program and portability. Therefore, it can be flexibly applied in most mainstream networks.

We also carried out a mini experiment on ArcFace. Softmax and ArcFace under the experiment contained 8 identity and 2D functions. The dots represent samples, and the lines represent the central direction of each identification. On the basis of feature normalization, ArcFace pushes all face features into an arc space with a fixed radius. Due to the effect of margin, the distance gap between different classes with the minimum distance becomes obvious. As shown in Fig. 9.



(a) Softmax (b) ArcFace

Figure 9: A toy experiment between softmax and ArcFace

4 The Triplet Loss and the Joint Loss of HST Loss and ACT Loss

4.1 Triplet Loss

In order to solve the problem that the intra-class distance is not large enough and the distance of inter-classes is not compact enough when softmax is applied in the field of face recognition. In 2015 FaceNet introduced a new loss function called Triplet Loss [26]. It has shown very good performance when learning human face embedding. The reason is that similar faces are embedding very close in the embedding space and can be used for recognizing the same face.

The principle of Triplet Loss is that it enters a triple when calculating the loss. The triad includes a target image, a positive sample image (an image of the same category as the sample), and a negative sample image (an image of different categories as the sample). The example of the relationship between the three is shown in Fig. 10.

It is worth noting that the distance between samples of the same category in the final embedding space is small, and the distance between samples of different categories will be small as well. Therefore, we need

to add margin. Through optimization, the distance between the target image and the positive sample is continuously reduced ($d(a,p) \to 0$).

Meanwhile, the distance between the target image and the negative sample increases continuously ($d(a,n) > margin$), to achieve the goal of expanding the distance between classes and shrinking the distance within classes in the mapping space.

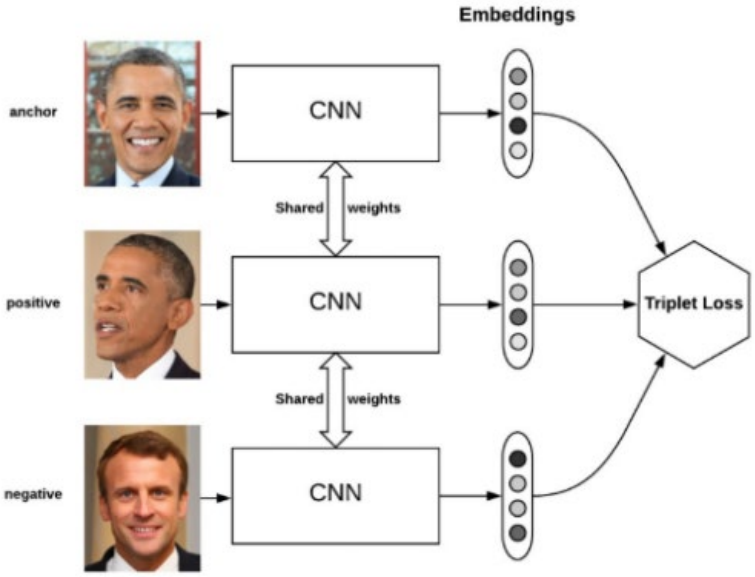


Figure 10: An example of the triplet in Triplet loss

So for the resulting triad, there are three different scenarios:

1. Easy: refers to the distance from the sample to the positive sample in a triplet is naturally closer than the distance from the negative sample, and the margin is greater than a margin, as shown in formula (17):

$$dsitance(a,p) + margin < distance(a,n) \tag{17}$$

2. Hard: Means that the distance from the sample to the positive sample in the triplet is greater than the distance from the negative sample, as shown in formula (18):

$$distance(a,p) > distance(a,n) \tag{18}$$

3. Semi-hard: Although the distance from the sample to the positive sample is less than the distance from the negative sample, the difference is less than a margin, i.e., a triplet with too small inter-class distance, as shown in Formula 19:

$$distance(a,p) < distance(a,n) < distance(a,p) + margin \tag{19}$$

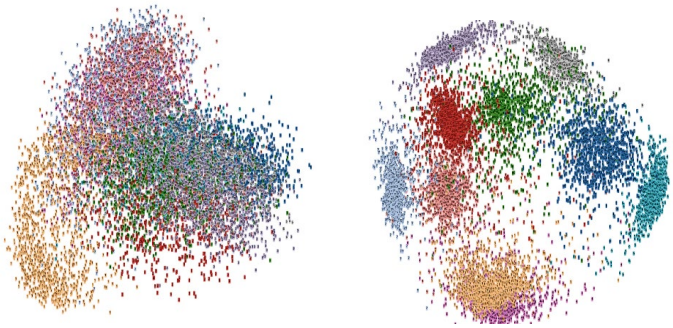
4. In FaceNet, semi-hard triples are usually selected for training, but semi-hard and hard triples can also be used for joint training. For each input sample, its loss function is shown in formula (20):

$$L_{sample} = \max\left(distance(a,p) - distance(a,n) + margin, 0\right) \tag{20}$$

So we have the final loss function, which is formula (21):

$$L_{triplet} = \frac{1}{2N}\sum_i^N [\left|\left|f(x_i^a) - f(x_i^p)\right|\right|_2^2 - \left|\left|f(x_i^a) - f(x_i^n)\right|\right|_2^2 + margin] \tag{21}$$

As you can see from Fig. 11, with just two epochs, the jumbled data has revealed a different class of features, suggesting that Triplet Loss has great performance, due to the huge amount of training, the traditional Triplet loss will suffer from problems such as slow convergence and instability, so that there is room for improvement.



(a). Original MINIST distribution (b). Distribution after Triplet loss training

Figure 11: Comparison of data distribution before and after Triplet Loss training

4.2 The Joint Loss of HST Loss and ACTLoss

4.2.1 HST Loss

Since Triplet Loss presents excellent effects in face recognition, such as face re-identification [36], it is of great significance to improve the Triplet Loss. To address the disadvantages of the Triplet loss, the optimized method is based on the Triple selected. But, due to the random selection of unconstrained triplet samples, the distance distribution between and within classes is not clear, which makes the network unstable and leads to convergence difficulties, and it is also quiet hard to find a perfect threshold for face verification.

Therefore, based on the hard sample triplet [26], the idea of HST loss [37] (that is to constrain the Triplet sample loss) is proposed to tackle the problem above, and it send the maximum intra-class distance and the minimum inter-class distance to the loss function. The HST loss alleviates the slow convergence and instability of the traditional Triplet loss to some extent under the condition that the distance within the largest category is less than the distance between any classes. As is shown in Fig. 12.

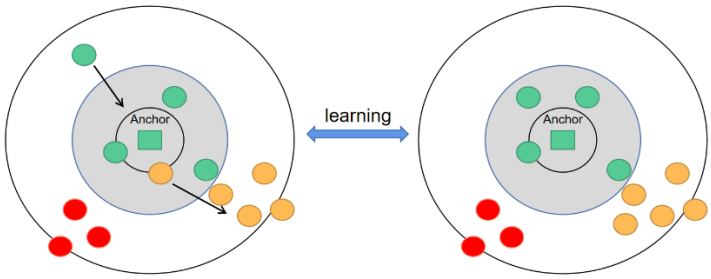


Figure 12: The network structure of ArcFace

Its mathematical formula is shown in formula (22):

$$Loss_{HST} = \frac{1}{C*N}\sum_{x_i}[\max_{x_j,y_i=y_j} distance\left(f(x_i), f(x_j)\right) - \min_{x_k,y_k\neq y_i} distance\left(f(x_i) - f(x_k)\right) + margin] \tag{22}$$

Among them, C stands for the number of categories in a batch and N represents the number of samples per class. If x_i and x_j are samples of the same class, then $y_i = y_j$. If x_i and x_j are samples of the different class, then $y_i \neq y_j$. $f(x)$ represents the eigenvector obtained by mapping the input x.

4.2.2 ACT Loss

Because when constructing a triple, the HST loss only consider the positive samples and negative samples relative distance of a given sample, without considering the other negative samples, therefore, the

problem that in the distance space some negative sample pairs might become positive sample pairs occurs due to the different categories of distance distribution, which affect the performance of the neural network.

Since that, ACT [37] loss imposes an absolute constraint that the distance within the largest class is less than the distance between any classes. It's shown in the formula which is formula (23).

$$Loss_{ACT} = \frac{1}{C*N}\sum_{x_i}[\max_{x_j, y_i=y_i} distance\left(f(x_i), f(x_j)\right) - \min_{x_m, x_n, y_m \neq y_n} distance\left(f(x_m), f(x_n)\right) + margin] \quad (23)$$

As shown in Fig. 13, under the absolute constraint of formula (17), ACT Loss will increase the distance for any two negative classes, while shorten the positive classes. Therefore, compared with HST loss, it enhances the recognition of learning features.

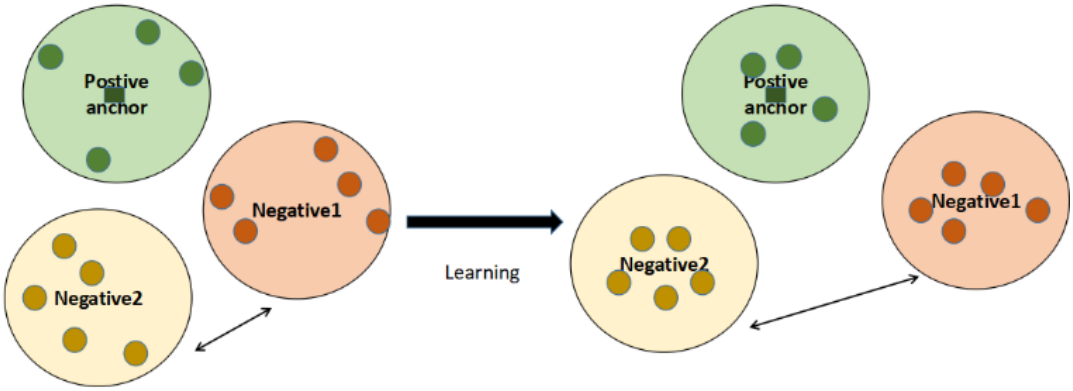


Figure 13: Comparison before and after ACT Loss training

4.2.3 The Combination of HST and ACT

In order to alleviate the defects of HST algorithm which has advantages of fast convergence speed and stable network, we combine HST algorithm and ACT algorithm to construct the final loss function, and the proportion of the two in the final loss is controlled by the super parameter α, as shown in formula (24) [37]:

$$Loss_{jiont} = \alpha Loss_{HST} + (1-\alpha) Loss_{ACT} \quad (24)$$

where $Loss_{HST}$ is formula 16 and $Loss_{ACT}$ is formula (17).

Through experiments, we can clearly see that HST Loss tends to achieve better results than ordinary Triplet loss and Softmax loss functions, and optimizes the loss function based on the combination of HST and ACT to get the best results. As shown in the Tab. 2.

Through adding an absolute constraint to the HST loss, the joint loss function alleviates the face authentication difficulties caused by the uneven distribution of distance between classes w ith different identities. The validity of the proposed method is verified on LFW.

Through adding an absolute constraint to the HST loss, the joint loss function alleviates the face authentication difficulties caused by the uneven distribution of distance between classes w ith different identities. The validity of the proposed method is verified on LFW. and YTF data sets, respectively. This loss function can also be used for video-based recognition.

Table 2: The verification rates at 1% FAR of different losses on LFW and YFT [37]

Loss	LWF (%)	YTF (%)
Triplet	96.40	85.45
HST	98.41	89.86
The joint	99.22	93.14

5 Conclusion

In this paper, we first introduce the background and development of the field of face recognition, and focus on the importance of loss function to face recognition network performance. After introducing the loss function of the traditional neural network, this paper also illustrates that the classification result of the neural network needs a small intra-class distance while a large inter-class distance for the classification result of the face, and points out the shortcomings of the traditional loss function. Then in order to solve this problem, this paper mainly three parts introduced three kinds of solutions, one is the method of using add penalty term governing the loss function of traditional: center loss and Orthogonality loss, another is to use presents margin method, not direct optimization characteristic vector, but the Angle between the optimal weight matrix and the eigenvector or cosine of the Angle, this article mainly introduced the performance of very excellent SphereFace, CosFace, ArcFace. The last method is Triplet Loss, and a new joint loss function based on HST Loss and ACT Loss is proposed to alleviate the shortcoming of slow convergence and unstable traditional Triplet loss. This paper analyzes the advantages, disadvantages and application scenarios of the above loss functions.

In addition to the above loss functions, there are also some rare but novel losses, such as [38]. Using von Mises-Fisher(vMF) hybrid model area as the theoretical basis, a novel vMF hybrid loss and its corresponding vMF depth characteristics were proposed. Chen et al. [39] proposed a method to simulate the early saturation of softmax by injecting annealing noise into it, etc.

Acknowledgement: This work was supported in part by the National Natural Science Foundation of China (Grant No. 41875184), Innovation Team of "Six Talent Peaks" In Jiangsu Province (Grant No. TD-XYDXX-004). Thanks for the funding of these two projects. First of all, I would like to thank Professor Wang Jun and Teacher Cheng Yong for their guidance and modification suggestions. At the same time, I would like to thank Nanjing University of Information Science & Technology for providing a very good learning and research environment.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China (Grant No. 41875184), Innovation Team of "Six Talent Peaks" In Jiangsu Province (Grant No. TD-XYDXX-004).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Gu, W. Xiong and Z. Bai, "Human action recognition based on supervised class-specific dictionary learning with deep convolutional neural network features," *Computers, Materials & Continua*, vol. 63, no. 1, pp. 243–262, 2020.

[2] W. S. McCulloch and P. Walter, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1043.

[3] D. O. Hebb and J. Wiley, "The organization of behavior: a neuropsychological theory," in *The Organization of Behavior: A Neuropsychological.* New York, NY, USA: Chapman & Hall, 1949.

[4] D. C. Cireşan, U. Mieier, L. M. Gambardella and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[5] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke *et al.,* "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2008.

[6] C. Farabet, C. Couprie, L. Najman and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.

[7] L. Wang, H. Lu, X. Ruan and M. H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015.

[8] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015.

[9] I. Arel, D. C. Rose and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[10] Y. Bengio, "Learning deep architectures for AI," in *Foundations and Trends® in Machine Learning*, vol. 2, no. 1. Norwell, MA, USA: Now Publishers, pp. 1–127, 2009.

[11] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[12] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014.

[13] Y. Sun, X. Wang and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014.

[14] I. Masi, Y. Wu, T. Hassner and P. Natarajan, "Deep face recognition: A survey," in *Proc. 2018 31st SIBGRAPI Conf. on Graphics, Patterns and Images (SIBGRAPI), IEEE*, Paraná, Brazil, 2018.

[15] S. Gutta and H. Wechsler, "Face recognition using hybrid classifier systems," In *Proc. Int. Conf. on Neural Networks*, vol. 2. IEEE, Washington DC, USA, 1996.

[16] P. C. Gibson, D. C. Noble and L. T. Larson, "Multistage evolution of the Calera epithermal Ag-Au vein system, Orcopampa District, southern Peru; first results," *Economic Geology*, vol. 85, no. 7, pp. 1504–1519, 1990.

[17] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proc. the 23rd ACM Int. Conf. on Multimedia*, New York, NY, USA, 2015.

[18] S. H. Lin, S. Y. Kung and L. J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114–132, 1997.

[19] S. Wold, K. Esbensen and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, 1987.

[20] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3., pp. 993–1022, 2003.

[21] J. Haddadnia, M. Ahmadi and K. Faez, "An efficient feature extraction method with pseudo-Zernike moment in RBF neural network-based human face recognition system," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 9, pp. 267692, 2003.

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

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.,* "Going deeper with convolutions," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015.

[24] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[25] R. Hadsell, S. Chopra and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition,* vol. 2. IEEE, New York, NY, USA, 2006.

[26] F. Schroff, D. Kalenichenko and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015.

[27] Y. Sun, Y. Chen, X. Wang and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Canada, 2014.

[28] Y. Wen, K. Zhang, Z. Li and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conf. on Computer Vision*, Springer, Cham, Amsterdam, Netherlands 2016.

[29] S. Yang, W. Deng, M. Wang, J. Du and J. Hu, "Orthogonality Loss: Learning Discriminative Representations for Face Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[30] D. Wang, C. Otto and A. K. Jain, "Face search at scale: 80 million gallery," arXiv:1507.07242, 2015.

[31] S. Sankaranarayanan, A. Alavi, C. D. Castillo and R. Chellappa, "Triplet probabilistic embedding for face verification and clustering," in *Proc. 2016 IEEE 8th Int. Conf. on Biometrics Theory, Applications and Systems (BTAS), IEEE*, Niagara Falls, NY, USA, 2016.

[32] W. Liu, Y. Wen, Z. Yu and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, Stockholm, Sweden, vol. 2, no. 3, 2016.

[33] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj *et al.,* "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, HI, USA, 2017.

[34] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong *et al.,* "Cosface: Large margin cosine loss for deep face recognition," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.

[35] J. Deng, J. Guo, N. Xue and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Los Angeles, CA, USA, 2019.

[36] A. Hermans, L. Beyer and B. Leibe, "In defense of the triplet loss for person re-identification." arXiv preprint arXiv:1703.07737, 2017.

[37] S. Wang and Y. Chen, "A joint loss function for deep face recognition," *Multidimensional Systems and Signal Processing*, vol. 30, no. 3, pp. 1517–1530, 2019.

[38] M. Hasnat, J. Bohné, J. Milgram, S. Gentric and L. Chen, "von mises-fisher mixture model-based deep learning: Application to face verification," arXiv preprint arXiv:1706.04264, 2017.

[39] B. Chen, W. Deng and J. Du, "Noisy softmax: Improving the generalization ability of DCNN via postponing the early softmax saturation," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, HI, USA, 2017.