Contrapositive Margin Softmax Loss for Face Verification

Dongxue Xu
College of Computer Science
Sichuan University, Chengdu, China
xu.dx@foxmail.com

Qijun Zhao
College of Computer Science
Sichuan University, Chengdu, China
qjzhao@scu.edu.cn

ABSTRACT
The performance of face recognition has been boosted by the features extracted from deep convolutional neural networks. Ideal features should have minimum intra-class variations and maximum inter-class variations. The most commonly used loss function for classification, softmax loss, however, does not necessarily learn features discriminative enough. Large margin classifiers have nice generalization properties in statistical machine learning. These properties have lead to the application of margin to deep learning in recent years. We hereby propose a new loss function called Contrapositive Margin Softmax loss for face verification task, which helps to learn invariant and discriminative features by introducing margins to both target logits and maximum negative logits of softmax loss. Competitive results on LFW (99.28%) and YTF (95.34%) demonstrate the effectiveness of our approach.

CCS Concepts
• **Computing methodologies** → **Biometrics**

Keywords
Face Recognition; Margin Loss Function; CNN

1. INTRODUCTION
Since the great success of AlexNet [1], deep convolutional neural networks (CNNs) have been prevailing in computer vision. Face recognition (FR), a common computer vision task, has also benefited from the advancement of CNNs.

Softmax loss is probably the most widely used loss function in classification tasks, but it cannot guarantee strong discrimination. Euclidean distance based loss functions [2, 3, 4] were proposed to provide an auxiliary loss or do embedding learning. [5, 6] introduce margin (so-called angular margin) into softmax loss, which further enhances intra-class compactness. Following their work, [7, 8] add cosine margin to softmax loss, and [9] performs an additive angular margin, in contrast to the original multiplicative angular margin [5].

In this paper, we propose a Contrapositive Margin Softmax loss (COPRA) to encourage intra-class compactness and inter-class discrimination simultaneously. Specifically, we introduce cosine margin to both the target logit and the maximum negative logit of softmax loss, whereas previous methods [7, 8] consider the target logit only. This way, our proposed method can better separate different classes. Our main contributions are summarized as follows:

(1) We propose COPRA loss for CNNs to learn invariant and discriminative features. In COPRA loss, margin for target classes can effectively improve the invariance of features within classes and thus result in better intra-class compactness, while margin for non-target classes can further enhance the inter-class discrimination capacity of the learned features.

(2) The proposed approach achieves competitive results on the Labeled Faces in the Wild (LFW) [10] and YouTube Faces (YTF) [11] datasets for face verification.

ICRCA '18, August 11–13, 2018, Chengdu, China

ACM ISBN 978-1-4503-6530-7/18/08…$15.00
DOI: https://doi.org/10.1145/3265639.3265679

2. RELATED WORK
Softmax loss can be seen as the combination of three components, i.e. an inner product, a softmax function and a cross-entropy loss (See Figure 1 and Equation 1).

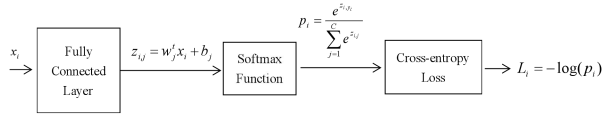

Figure 1. Softmax Loss

x_i is the extracted feature of sample i with label y_i, $i \in \{1,...,N\}$. w_j and b_j are weight and bias of fully connected layer corresponding to class j, $j \in \{1,...,C\}$, respectively. $z_{i,j}$ represents the logit of class j in softmax function for feature x_i. The original softmax loss is given by

$$L_s = \frac{1}{N}\sum_{i=1}^{N} L_i = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{w_{y_i}^t x_i + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^t x_i + b_j}} \quad (1)$$

Normalization. The inner product of the last fully connected layer's weights and the learned feature (ignoring the bias term) can be formulated as $z_{i,j} = w_j^{\mathrm{t}} x_i = \| w_j \| \cdot \| x_i \| \cdot \cos\theta_{i,j}$. To maximize the target logit, i.e. z_{i,y_i}, softmax loss would readily enlarge the norm of target class weight and the learned feature, which should weaken the capacity to decrease the cosine distance between the learned feature and its target weight comparatively. To encourage the learned feature to be closer to its corresponding weight, recent works [7, 8, 9, 12] all normalize the weight and the feature of softmax loss. We empirically find that normalization does help in performance. All our experiments are conducted with feature and weight normalization.

Margin based Loss Function. Large margin classifiers have nice generalization properties in statistical machine learning. These properties have lead to the application of margin to deep learning

in recent years. [5, 6, 7, 8, 9] impose margin to softmax loss and achieve state-of-art performances.

Unlike the above cosine distance based margin loss functions, which only impose margin to the target logit, we introduce margins for both the target logit and the maximum negative logit of softmax.

3. THE PROPOSED APPROACH

3.1 Additive Margin (AM) Softmax Loss

The original softmax loss function is defined in Equation 1. After weight and feature normalization, it becomes a *cosine loss* function. Besides normalization, considering that small feature norms would make CNNs hard to train [12], we rescale the loss with a scalar *s* after normalization (Equation 2). We refer to the normalized softmax loss as NSL, for short.

$$L_i = -\log \frac{e^{z_{y_i}}}{\sum_{j=1}^{C} e^{z_j}} = -\log \frac{e^{s\cos\theta_{y_i}}}{\sum_{j=1}^{C} e^{s\cos\theta_j}} \tag{2}$$

where L_i is the softmax loss for sample i. θ_j is the angle between the weight of class j and the feature from sample i. Considering a binary classification and a learned feature x from class 1. The cosine loss function must force $cos(\theta_1) > cos(\theta_2)$ to make correct classifications. However, if we introduce a non-negative margin m to class 1, $cos(\theta_1) - m > cos(\theta_2)$ then needs to be satisfied. The following inequality always holds when m is non-negative:

$$cos(\theta_1) \geq cos(\theta_1) - m > cos(\theta_2)$$

which guarantees a more rigorous decision boundary to be held for class 1. The decision boundary now becomes $cos(\theta_1) - m = cos(\theta_2)$, compared to the original decision boundary $cos(\theta_1) = cos(\theta_2)$ (see Figure 2). Similarly, if we want to classify class 2 correctly, $cos(\theta_2) - m > cos(\theta_1)$ is required. Multi-class classification with margin loss function is analogous to a binary classification case. Additive cosine margin loss functions [7, 8] (Equation 3) are effective but much simpler than angular margin loss functions [5, 6, 9] when implemented.

$$L_{AS} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s(\cos\theta_{y_i} - m)}}{e^{s(\cos\theta_{y_i} - m)} + \sum_{j=1, j\neq y_i}^{C} e^{s\cos\theta_j}} \tag{3}$$

3.2 Contrapositive Margin Softmax Loss

Since we have reformulated softmax loss to cosine loss, the only thing that matters is the angle between the learned feature and each classification weight. As the margin only changes the lower bound of cosine distance of the feature and the target weight in one sample classification, the principle effect it has is to make the learned feature get closer to the target weight, though *closer to the target weight* and *away from other non-target weights* are mutually beneficial.

Additionally, to further enhance the discriminative capacity of the learned feature, we impose an extra margin to the class which has the maximum logit in all the negative classes. The proposed loss function is called Contrapositive Margin Softmax Loss (COPRA), defined as:

$$L_{COPRA} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s(\cos\theta_{y_i} - m_1)}}{e^{s(\cos\theta_{y_i} - m_1)} + e^{s(\cos\theta_j + m_2)} + \sum_{k=1, k\neq y_i, k\neq j}^{C} e^{s\cos\theta_k}} \tag{4}$$

where j is the label of the maximum negative logit. m1 and m2, both non-negative, are the margins imposed on the target logit and the maximum negative logit, respectively. We denote the margin between $cos(\theta_i)$ and $cos(\theta_j)$ as $M_{AM/COPRA}(i, j)$, where the subscript AM or COPRA represents AM softmax loss or COPRA loss. When m1=m and m2=0, the proposed COPRA loss becomes AM softmax loss. As there's an extra m2 term, we expand the margin between the learned feature and the maximum negative class weight. In the binary classification case, our COPRA loss is equivalent to AM softmax loss if m=m1+m2. But in the multi-class classification case, when m1=m and m2>0, we have $M(y_i, j)$=m1+m2=m+m2, which is obviously larger than m. Nevertheless $M(y_i, k)$=m1=m, which remains unchanged. For better understanding, Figure 2 and Figure 3 provide the binary classification and the three-class classification with 2D and 3D visualization, respectively.

Compared to the original softmax loss or normalized softmax loss, AM softmax loss gives more rigorous conditions. Besides that, our proposed COPRA puts more stringent restrictions on the cosine distance than AM softmax loss.

3.3 Bounds on Hyperparameter s

After normalizing softmax loss to normalized softmax loss, the loss is only contributed to by the cosine distance of features and weights. Assuming that all the class weights are well separated, e.g. mutually orthogonal, the probability of correctly classifying a sample is:

$$p = \frac{e^s}{e^s + (C-1)e^0} \tag{5}$$

We can infer that

$$s = \ln \frac{p(C-1)}{1-p} \tag{6}$$

where C is the number of classes. If we expect a classification probability p, Equation 6 provides us the lower bound on s. For example, for our experimental settings, where the number of classes is 10,516, if we expect to achieve a probability 0.99, the lower bound on s is 13.86. As the feature dimension is far less than the number of classes, class weights cannot be orthogonal, and even normally distributed in the feature space could be challenging. Thus, to increase the classification probability, the scale parameter has to be larger in practice. We set s to 30 in our experiments empirically.

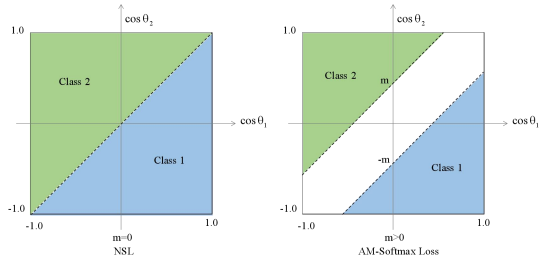

Figure 2. The comparison of decision margin for NSL and AM-Softmax loss in binary classification. Left: NSL without margin (m=0). Right: AM-Softmax loss with margin. Class 1 and 2 are represented by the blue and green regions,

respectively. The white areas are decision margins.

3.4 Discussions

Comparison with AM softmax loss. As mentioned, the proposed COPRA loss is equivalent to AM softmax loss in the binary classification but has lager margin with the maximum negative class in the multi-class classification. Note that in the two scenarios, the assumptions are different. In the equivalence case, we assume that m=m1+m2, while in the multi-class classification case, we assume that m1=m. If we still keep m=m1+m2 in multi-class classification, it's not necessary that COPRA loss has greater discriminative power than AM softmax loss. Assume that j is the maximum negative class and k represents any other negative class. If m=m1+m2, $M_{COPRA}(y_i, j) = M_{AM}(y_i, j)$ always holds, but it would harm $M_{COPRA}(y_i, k)$, leading to $M_{COPRA}(y_i, k) < M_{AM}(y_i, k)$. On the contrary, if m=m1, $M_{COPRA}(y_i, k) = M_{AM}(y_i, k)$ is always satisfied, and $M_{COPRA}(y_i, j) > M_{AM}(y_i, j)$ enables a larger margin between the learned feature and the maximum negative class weight.

From a backward perspective. We have discussed the intuition of COPRA in a forward-like manner, that is, margin imposed on the target logit mainly results in intra-class feature compactness, and margin on the maximum negative logit could improve the discriminative capacity of features. When training CNNs with SGD, weights get updated based on the gradient of the loss w.r.t the weights. The gradient of NSL w.r.t the target logit and other logits in softmax loss are:

$$\Delta s z_{y_i} = \Delta s \cos(\theta_{y_i}) = -\nabla_{s\cos(\theta_{y_i})} L_i = 1 - p_{y_i} = 1 - \frac{e^{s\cos(\theta_{y_i})}}{\sum_{j=1}^{C} e^{s\cos(\theta_j)}} \quad (7)$$

$$\Delta s z_j = \Delta s \cos(\theta_j) = -\nabla_{s\cos(\theta_j)} L_i = -p_j = -\frac{e^{s\cos(\theta_j)}}{\sum_{k=1}^{C} e^{s\cos(\theta_k)}} \quad (8)$$

where Δw represents the increment of w for update, which is proportional to the negative value of its corresponding gradient $\nabla_w L$ (here we ignore the learning rate). Compared to NSL, AM softmax loss only changes the probability terms (p_i, $i \in \{1,...,C\}$) for weight updates, leading to $|\Delta w_i|_{AM} > |\Delta w_i|_{NSL}$, for $i \in \{1,...,C\}$. Maybe that's why AM softmax loss works better than NSL. Now taking COPRA into consideration, one would find that $|\Delta w_i|_{COPRA} > |\Delta w_i|_{AM} > |\Delta w_i|_{NSL}$ when i represents the sample class or the class of maximum negative logit. We have given some discussions here for completeness, but at this time it is uncertain to us the relation between $|\Delta w|$ and the model performance.

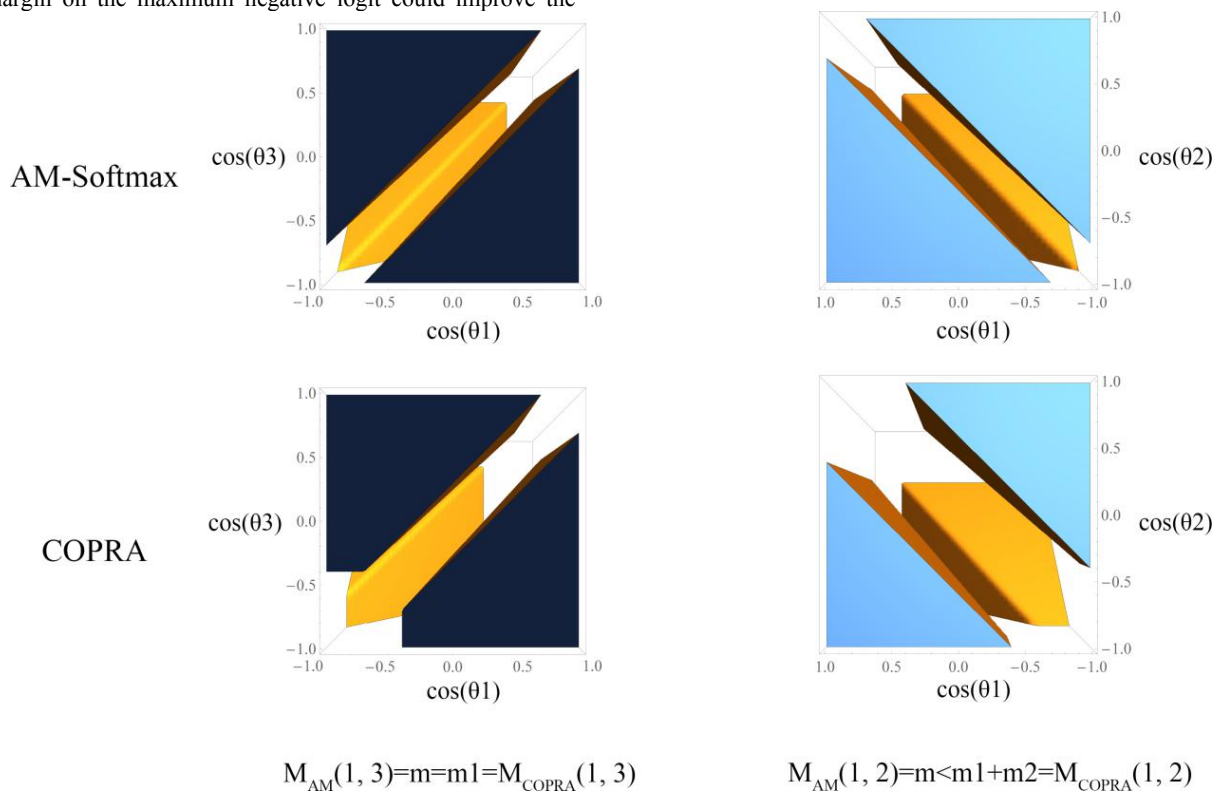

Figure 3. The comparison of decision margin for AM-Softmax loss and COPRA loss in three-class classification under the assumption that the maximum negative logit class for class 1 and class 3 are both class 2, and for class 2, the maximum negative logit class is class 1. The top and bottom rows compare the effects of AM Softmax loss to COPRA on the decision margins for class 1 and 3 (left column) and class 1 and 2 (right column). Notice the greater effect COPRA has on the decision margin between class 1 and 2.

4. EXPERIMENTS

4.1 Experimental Settings

Preprocessing. Following [5, 7], face and landmarks are detected by MTCNN [13] to perform image similarity transformation and cropping. The cropped image are then resized to 96×112. Each pixel ([0, 255]) in RGB images is normalized by subtracting 127.5 and then being divided by 128.

Training. We take ResNet-20 [5, 14] as our network structure. Loss function and experiments are performed under CAFFE [15] framework. The CNN models are trained by SGD, with a batch size of 256. Models trained from scratch start with an initial learning rate of 0.1, which is divided by 10 at the 16K, 24K iterations. We take the model snapshotted at 22Kth iteration for test. The hyperparameters s, m1 and m2 are empirically set to 30, 0.35, 0.01, respectively. We train our models on a cleaned version of CASIA-WebFace [16] dataset according to [7] for fair comparison. After duplicate removal, there are 10,516 identities remaining, in contrast to the original 10,575 identities and 10,572 identities used in [5]. During training, images are mirrored to augment the dataset.

Testing. For LFW benchmark, the representation of one image is obtained by taking max of feature from the original face image and feature from horizontally flipped image. For YTF benchmark, the representation is only extracted from the original image. Features all get normalized by zero-mean normalization. Cosine similarity and threshold are used to perform face verification.

4.2 Experiments on LFW and YTF

LFW is a widely used benchmark for unconstrained face recognition. It contains 13,233 images from 5,749 people. Performance reported on 10 fold cross validation is suggested. Experimental results are reported in Table 1.

YTF is a database of face videos designed for studying the problem of unconstrained face recognition in videos, specifically, the video pair-matching or template-to-template matching problem. It contains 3,425 videos of 1,595 different people. All the videos are divided into 5,000 video pairs and 10 fold cross validation is required. When computing the similarity score of two videos, we firstly normalize features. Then we compute the cosine distance matrix of two videos with each entry indicating two frames' similarity. Finally, we take the median of the distance matrix as the similarity metric of the two videos. We follow the typical unconstrained with labeled outside data protocol and the result is shown in Table 1.

Table 1. Accuracy on LFW and YTF datasets. For AM softmax loss, we evaluate the official released model under the same test settings as COPRA loss.

Model	#Images	#Layers	LFW(%)	YTF(%)
FaceNet [3]	200M	22	**99.63**	95.12
Center Face [4]	0.7M	7	99.28	94.90
L-Softmax [6]	0.49M	17	98.71	N/A
SphereFace [5]	0.49M	64	99.42	95.00
AM-Softmax [7] (m=0.35)	0.49M	20	98.98	94.96
COPRA (m1=0.35, m2=0.01)	0.49M	20	99.28	**95.34**

After imposing another margin to maximum negative logits, COPRA consistently achieves higher accuracy than AM softmax loss, which only utilizes margin on target logits. While this is with the smallest training dataset and a quite small neural network, our single model trained with COPRA loss achieves competitive results on both still-image and video-based face verification tasks, which verifies the potential of our approach.

5. CONCLUSIONS

In this paper, we propose a new loss function called COPRA, which helps to learn invariant and discriminative features by introducing margins to the target logits and the maximum negative logits as well. The target logits' margin mainly decreases intra-class variance, while the negative logits' margin enlarges inter-class variance. Experiments on unconstrained face datasets, i.e. LFW and YTF, have demonstrated the effectiveness of the proposed approach. To the best of our knowledge, this is the first study to simultaneously consider margins for the target and the negative logit terms.

6. ACKNOWLEDGEMENTS

This research is supported by the National Natural Science Foundation of China (No. 61773270).

7. REFERENCES

[1] KRIZHEVSKY, A., SUTSKEVER, I., and HINTON, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097-1105.

[2] SUN, Y., CHEN, Y., WANG, X., and TANG, X., 2014. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, 1988-1996.

[3] SCHROFF, F., KALENICHENKO, D., and PHILBIN, J., 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815-823.

[4] WEN, Y., ZHANG, K., LI, Z., and QIAO, Y., 2016. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision.* Springer, 499-515.

[5] LIU, W., WEN, Y., YU, Z., LI, M., RAJ, B., and SONG, L., 2017. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] LIU, W., WEN, Y., YU, Z., and YANG, M., 2016. Large-Margin Softmax Loss for Convolutional Neural Networks. In *ICML*, 507-516.

[7] WANG, F., CHENG, J., LIU, W., and LIU, H., 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters* 25, 7, 926-930.

[8] WANG, H., WANG, Y., ZHOU, Z., JI, X., LI, Z., GONG, D., ZHOU, J., and LIU, W., 2018. CosFace: Large Margin Cosine Loss for Deep Face Recognition. *arXiv preprint arXiv:1801.09414*.

[9] DENG, J., GUO, J., and ZAFEIRIOU, S., 2018. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698*.

[10] HUANG, G.B., RAMESH, M., BERG, T., and LEARNED-MILLER, E., 2007. *Labeled faces in the wild: A database for studying face recognition in unconstrained*

environments. Technical Report 07-49, University of Massachusetts, Amherst.

[11] WOLF, L., HASSNER, T., and MAOZ, I., 2011. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 529-534.

[12] WANG, F., XIANG, X., CHENG, J., and YUILLE, A.L., 2017. NormFace: L 2 Hypersphere Embedding for Face Verification. In *Proceedings of the 2017 ACM on Multimedia Conference.* ACM, 1041-1049.

[13] ZHANG, K., ZHANG, Z., LI, Z., and QIAO, Y., 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23, 10, 1499-1503.

[14] HE, K., ZHANG, X., REN, S., and SUN, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

[15] JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S., and DARRELL, T., 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 675-678.

[16] YI, D., LEI, Z., LIAO, S., and LI, S.Z., 2014. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.