Weighted Softmax Loss for Face Recognition via Cosine Distance

Hu Zhang(✉), Xianliang Wang, and Zhixiang He

Beijing Hisign Corp., Ltd., Hanwei International Square, Area 4, no. 186, West Road, 4th South Ring Road, Fengtai District, Beijing 100160, China
{zhanghu, wangxianliang, hezhixiang}@hisign.com.cn

Abstract. Softmax loss is commonly used to train convolutional neural networks (CNNs), but it treats all samples equally. Focal loss focus on training hard samples and takes the probability as the measurement of whether the sample is easy or hard one. In this paper, we use cosine distance of features and the corresponding centers as weight and propose weighted softmax loss (called C-Softmax). Unlike focal loss, we give greater weight to easy samples. Experiment results show that the proposed C-Softmax loss can train many well known models like ResNet, ResNeXt, DenseNet and Inception V3, and the performance of the proposed loss is better than softmax loss and focal loss.

Keywords: Face recognition · Focal loss · Softmax loss · C-Softmax loss

1 Introduction

Over the past few years, due to the success of convolutional neural networks, the accuracy of face recognition has improved greatly. Although there are many new loss functions [1–5], the most commonly used one is still softmax loss, which mainly optimizes the inter-class difference, and gives same weight to all samples. Although most training samples are easy samples in face recognition, there are still hard samples. These hard samples may degrade the generalization performance of the model. Focal loss [6] is proposed for dense object detection, it down-weights the loss assigned to easy samples, and focuses on training hard samples in order to prevent the vast number of easy samples from overwhelming the model during training. Although its performance is better, it is difficult to apply to face recognition, because most of the time, the number of training samples of one subject is not large. Meanwhile, we think it is unreasonable to measure the difficulty of training samples by probability. One main difference between face recognition and detection is the variation of one person is small (although there are still changes in pose, expression and illuminations), and thus we can obtain the feature's centers of each subject. We think it is more reasonable to use the angle between features and its corresponding centers than probability to measure whether it is easy sample or hard one. We also think it may degrade the generalization performance of the model when focus on training hard samples, so we give greater weight to those easy samples. In this paper, we use cosine distance of features and its corresponding centers as the weight and propose a new loss function called C-Softmax loss.

J. Zhou et al. (Eds.): CCBR 2018, LNCS 10996, pp. 340–348, 2018.
https://doi.org/10.1007/978-3-319-97909-0_37

The advantages of C-Softmax loss is as follows: 1. It is easier to convergence than L-Softmax [4] and A-Softmax [5]. When training data has too many subjects, the convergence of L-Softmax and A-Softmax will be more difficult than softmax loss, and thus they used a learning strategy. The proposed loss is based on softmax loss, so it is easy to convergence. 2. It does not need any pre-trained model. Both COCO loss [7] and NormFace [8] use a pre-trained model and fine tune the model by their loss. We use softmax loss in the first few epochs to get the rough centers, which could not be considered as the pre-trained model, because the total number of training epoch remains unchanged, and the performance of the model is poor at this time. 3. It does not need to design pair selection procedure like triplet loss [2] and contrastive loss [3].

Although C-Softmax has many advantages, it still faces some problems. One main problem is it has to maintain feature centers like center loss [1], and we update feature centers the same way center loss does. Another problem is we have to train the model by softmax in the first few epochs, and decrease the number of epoch by C-Softmax loss, so as to keep the total number of training epoch unchanged.

2 Related Work

Given an input image x_i with label y_i, original softmax loss function is defined as:

$$Ls = -\frac{1}{m}\sum_{i=1}^{m}\log\left(\frac{e^{W_{y_i}^T f(x_i)+b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T f(x_i)+b_j}}\right) \tag{1}$$

where m is the batch size, n is the number of training class, $f(x_i)$ is the feature, $\mathbf{W} \in R^{n\times d}$ and $\mathbf{b} \in R^n$ are the weight and bias of the fully-connected layer before softmax loss, W_j is the j-th column of $\mathbf{W}$ and d is the feature dimension.

Focal loss [6] is proposed for dense object detection. It is used to handle extreme imbalance between foreground and background classes. The α-balanced variant of the focal loss is defined as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{2}$$

$$p_t = \begin{cases} p_i & if\ y_i = 1 \\ 1-p_i & otherwise \end{cases} \tag{3}$$

$$\alpha_t = \begin{cases} \alpha & if\ y_i = 1 \\ 1-\alpha & otherwise \end{cases} \tag{4}$$

where $\alpha \in [0,1]$ is a weighting factor, $p_i \in [0,1]$ is the model's estimated probability for the class with label $y_i = 1$, γ is set to 2 in the paper.

We can apply focal loss to face recognition. But the performance is worse than softmax loss. We think the reason is that it is unreasonable to use probability to measure the degree of difficulty of samples, and it may degrade the performance when focus on training hard samples. Inspired by focal loss, we modified softmax loss and

proposed weighted Softmax loss via Cosine Distance (C-Softmax) to train deep models for face recognition.

3 Proposed C-Softmax Loss

Given two vectors $f \in R^d$ and $C \in R^d$, the cosine distance of them is:

$$d = \frac{f \cdot c^T}{\|f\|_2 \|c\|_2} \tag{5}$$

The range of the cosine distance is [−1, 1]. The greater the distance, the more similar these two vectors is. The proposed C-Softmax loss is defined as:

$$CS_i = -w_i^r \times \log(p_i) \tag{6}$$

where w_i is the modified cosine distance of the current features f_i and the corresponding centers c_i. γ is set to 2, so there is no hyper-parameter in C-Softmax loss. As the angle between the feature and its corresponding center is greater than 90°, the weight is negative, so w_i is defined as follows to keep its monotony.

$$w_i = \begin{cases} d & if\ d \geq 10^{-6} \\ 10^{-6} & otherwise \end{cases} \tag{7}$$

We do not use α-balanced variant of C-Softmax loss in order to keep it concise. If all the weights are 1, then C-Softmax loss becomes softmax loss. If the weight of hard examples are greater than easy ones, C-Softmax loss is more like focal loss.

4 Results and Analysis

4.1 Experiment Details

Experiment Settings: We implement the proposed loss using PyTorch [11] framework. The face landmarks are detected by MTCNN [12]. The aligned face images are of size 112 * 96. The weight decay is $5e^{-4}$. The batch size is 256 and we use stochastic gradient descent to train the model. The learning rate begins with 0.1 and is divided by 10 at 11, 16 and 19 epochs, and finishes at 20 epochs. There are three ways to obtain the centers. 1 initialize the centers randomly and train the model by C-Softmax from the beginning. 2 fine tune the model by C-Softmax loss from a pre-trained model and the corresponding centers. 3 train the model by Softmax for a few epochs and by C-Softmax for the remaining epochs. For the first one, the centers could not be 0 because the cos distance between vector **0** and any vector is 0, result in C-Softmax loss always be 0. When the centers is initialized improper (cosine distance of the features and its centers is negative), the performance of C-Softmax loss will be bad. We will get the best performance with the second way, but it will consume twice as much time

(train by softmax and fine tune by C-Softmax). We choose the third way. The feature's centers are more stable when the epochs trained by softmax loss increases and the epochs trained by C-Softmax decreases as the total number of training epochs is fixed. We found the performance is the best when trained with softmax for 3 epochs. So we set all centers to be 0 at the beginning, train the model by softmax loss for 3 epochs, and update the centers like center loss. We use C-Softmax loss to train the model from epoch 4, the training finishes at 20 epochs.

Network Structure: We compare the performance of different loss functions with four network structures. model-A is the same as [5]. model-B has Batch Normalization (BN) [13] layer after FC1 layer. Model-C has BN layer after each convolution layer and FC1 layer. Model-D uses RReLU [14] instead of PReLU [15] as activation function, and it has BN layer after each convolution layer and FC1 layer.

Training: We use CASIA-WebFace [9] to train our CNN models. CASIA-WebFace has 494414 face images belonging to 10575 different individuals. In [16] they reported 17 overlapped identities between CASIA-WebFace and LFW [10], and 42 overlapped identities between CASIA-WebFace and MegaFace [17] set1. We checked their result and found 3 mismatched overlapped identities, meanwhile we also found another 5 overlapped identities, so there are totally 19 overlapped identities between CASIA-WebFace and LFW. We removed all these 61 identities, and use the remaining 447020 images from 10541 identities to train the model.

Evaluation: We extract the features from the output of the FC1 layer, and if there is BN layer after FC1 layer, we thus use the output of BN layer as the features instead. Features from the original image and its horizontally flipped one are extracted, and then merged by element-wise mean as the representation. The dimension of the feature is 512. We use LFW [10] and MegaFace [17] set1 for evaluation. We follow the unrestricted with labeled outside data protocol [18] on both datasets. We also evaluate the performance through BLUFR protocols [19], it is more challenging and generalized for LFW because it utilize all 13233 images while the standard evaluation protocol only evaluated on 6000 image pairs.

4.2 Experiment Results

The 3 to 5 columns in Table 1 show the performance of different network structures trained with A-Softmax loss [5], softmax loss, center loss [1], focal loss [6] and the proposed C-Softmax loss. We can see that the performance of A-Softmax with model-A and model-B are both good, but when BN layer is added after convolution layer, DIR@FAR = 1% drops from 82.03% to 75.99%. Although it increases to 80.61% when use RReLU (model-D), the performance is still lower than the original model.

When BN layer is added after FC1 layer (changed from model-A to model-B), and trained with softmax loss, focal loss and center loss, the performance of DIR@FAR = 1% increase greatly. The performance are further improved when BN layer is added after each convolution layer (model-C). When we replace PReLU with RReLU, the performance of these three loss all decrease (model-D). Although focal loss

outperforms softmax loss in dense object detection [6], its performance is worse than softmax loss in face recognition.

Although the performance of the C-Softmax loss is not very good to train model-A, it works quite well with other three model structures. DIR@FAR = 1% increases to 86.17% when trained model-D, and it outperforms the performance of model-B trained with A-Softmax loss, which is 82.03%. Meanwhile, C-Softmax loss outperforms both focal loss and softmax loss when trained with same model (except model-A), and the improvement is obvious. The improvement benefits from not only the cosine distance instead of probability as the measurement of easy or hard samples, but also gives greater weight to easy samples than hard samples. We ignored some difficult samples, but the generalization performance of the model was improved. If the proportion of hard samples in the training datasets is low, and we focus on training them, it may degrade the generalization performance of the model, like focal loss used in face recognition. Otherwise we should give greater weight to hard samples and focus on training them, like focal loss used in object detection [6].

As is analyzed in [13], the distributions of features trained by softmax changed significantly over time without BN layer, both in mean and variance, and the features are not necessarily discriminative [5]. On the contrary, A-Softmax can learn discriminative features [5]. Focal loss and C-Softmax loss are both based on softmax loss, so the features are not as discriminative as A-Softmax loss. This is why the performance

Table 1. Performance (%) comparison for different loss functions with different structures on LFW and MegaFace dataset.

Model	Loss	LFW			MegaFace	
		Acc.	VR@FAR = 0.1%	DIR@FAR = 1%	Rank-1	VR@FAR = 10^{-6}
Model-A	A-Softmax loss	**99.12**	**97.7**	**81.75**	**62.77**	**72.48**
	Softmax loss	97.55	87.6	59.82	49.79	55.48
	Center loss	98.01	91.7	68.96	59.42	67.74
	Focal loss	97.38	84.87	58.05	49.25	54.45
	C-Softmax loss	97	82.93	63.29	47.53	52.79
Model-B	A-Softmax loss	**99.2**	**97.56**	**82.03**	**64.81**	**75.97**
	Softmax loss	98.61	93.53	75.43	61.82	73.65
	Center loss	98.5	93.53	76.55	62.2	75.17
	Focal loss	98.32	92.27	72.6	60.45	72.31
	C-Softmax loss	98.78	93.6	80.93	61.93	74.11
Model-C	A-Softmax loss	**99.16**	96.67	75.99	58.93	68.18
	Softmax loss	98.57	95.5	77.41	64.46	78.01
	Center loss	98.62	95.73	77.81	64.59	78.85
	Focal loss	98.36	94	75.93	63.26	76.27
	C-Softmax loss	99.1	**96.93**	**83.17**	**65.41**	**79.67**
Model-D	A-Softmax loss	99.15	96.47	80.61	63.48	74.93
	Softmax loss	98.38	89.43	76.05	63.13	74.51
	Center loss	98.48	94.1	76.65	63.5	74.54
	Focal loss	98.33	90.33	71.72	61.01	71.56
	C-Softmax loss	**99.2**	**98.2**	**86.17**	**68.66**	**83.15**

of model-A trained by softmax loss, focal loss and C-Softmax loss are poor. BN layer makes the distribution of the features more stable as training progresses and reduces the internal covariate shift [13], so the performance of the model trained by softmax loss, focal loss and C-Softmax loss improved greatly when BN layer is added, and the features are necessarily discriminative. At this time, BN layer may affect discriminant performance of A-Softmax loss.

From the above analysis we can also see that no loss function can work quite well with all structures. A-Softmax is more suitable for models without BN layer after convolution layer, while others are more suitable for models with it. A-Softmax and C-Softmax are more suitable for models with RReLU layer, while others are more suitable for models with PReLU layer. And we should train model with the most suitable loss function, so as to get best performance.

Table 2 list the accurate of different methods on LFW. Some methods use their own dataset, like FaceNet [2]; some methods trained on MS-Celeb-1 M [20], like SeqFace [21], ArcFace [22]; some methods trained on CASIA-WebFace [9], like LGM [23], NormFace [8]. We have the following observations. First, the performance of the methods trained on large datasets (The number of images is more than 1M) are quite good. Second, the performance will be further improved with more layers. The number of layers of SeqFace [21], SeqFace [21] and Ring Loss [24] are all greater than or equal to 64 layer, and their accurate are very high. Third, the performance of the proposed method is equal or better than LGM [23], NormFace [8] and AM-Softmax [16] when trained on the same dataset (Strictly speaking, the training images we used is the least). Generally speaking, we obtain state of the art performance by using the least number of training images.

Table 2. Detailed information and verification accuracy (%) of different methods on LFW

Method	Images	Networks	Layers	Acc. on LFW
FaceNet [2]	200M	1	–	99.63
CosFace [25]	5M	1	64	99.73
SeqFace [21]	4M+	1	64	99.83
ArcFace [22]	3.8M	1	100	99.83
Ring loss [24]	3.5M	1	64	99.52
Baidu [26]	1.2M	10	9	99.77
Center loss [1]	0.7M	1	27	99.28
SphereFace [5]	0.49M	1	64	99.42
LGM [23]	0.49M	1	27	99.2
NormFace [8]	0.49M	1	27	99.19
AM-Softmax [16]	0.44M	1	20	99.17
Proposed	0.44M	1	20	99.2

The last two columns in Table 1 show rank-1 identification accuracy with 1 M distractors and verification TAR for 10^{-6} FAR of various loss functions on MegaFace set1. C-Softmax outperforms the other loss functions and gets the best result when trained with the most suitable model.

To make our experiment more convincing, we also trained simplified Inception V3 [27], DenseNet [28], ResNeXt [29] with softmax loss, center loss [1], focal loss [6] and C-Softmax loss. The depth of Inception V3 is 37. The depth of ResNeXt is 29 with cardinality = 32 and bottleneck width = 4d. The depth of DenseNet is 21 with growth rate = 32, dense blocks = 4 while each have 2 layers. Table 3 lists the results. C-Softmax loss outperforms other loss functions and gets the best result with all these models.

Table 3. Performance (%) on LFW dataset with other well known models

Model	Loss	Acc.	VR@FAR = 0.1%	DIR@FAR = 1%
Inception V3	Softmax loss	98.45	92.56	71.38
	Center loss	**98.8**	94.77	71.35
	Focal loss	98.22	92.4	68.67
	C-Softmax loss	98.65	**97.47**	**76.62**
DenseNet	Softmax loss	97.33	86.87	60.74
	Center loss	97.51	85.93	64.16
	Focal loss	97.41	86.53	59.76
	C-Softmax loss	**98.17**	**93.73**	**71.29**
ResNeXt	Softmax loss	98.53	92.67	71.83
	Center loss	98.58	93.77	74.07
	Focal loss loss	98.17	91.4	69.18
	C-Softmax loss	**99**	**97.3**	**80.71**

5 Conclusion

Inspired by focal loss, we proposed a new loss function called C-Softmax loss in this paper. Firstly, we use the cosine distance of the features and the corresponding centers as the measurement of whether the sample is easy or hard, and add it as the modulating factor to the softmax loss. Secondly, we give greater weight to easy samples than hard samples in training phase. There is no hyper-parameter in the proposed loss. The results show that the proposed loss function provides a significant and consistent boost over softmax loss and focal loss, and can be used to train other well known models like ResNet, ResNeXt, DenseNet and Inception V3.

References

1. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 499–515. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_31
2. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)

3. Hadsell, R., Chopra, S., Lecun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1735–1742 (2006)
4. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: ICML, pp. 507–516 (2016)
5. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: SphereFace: deep hypersphere embedding for face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 212–220 (2017)
6. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. arXiv preprint arXiv:1708 (2017)
7. Liu, Y., Li, H., Wang, X.: Rethinking feature discrimination and polymerization for large-scale recognition. arXiv preprint arXiv:1710.00870 (2017)
8. Wang, F., Xiang, X., Cheng, J., Yuille, A.L.: NormFace: L2 hypersphere embedding for face verification. arXiv preprint arXiv:1704.06369 (2017)
9. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
10. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts (2007)
11. Paszke, A., Gross, S., Chintala, S., Chanan, G.: PyTorch: tensors and dynamic neural networks in Python with strong GPU acceleration (2017)
12. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Sig. Process. Lett. **23**, 1499–1503 (2016)
13. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
14. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
15. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
16. Wang, F., Liu, W., Liu, H., Cheng, J.: Additive margin softmax for face verification. arXiv preprint arXiv:1801.05599 (2018)
17. Kemelmachershlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The MegaFace benchmark: 1 million faces for recognition at scale. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4873–4882 (2016)
18. Huang, G.B., Learned-Miller, E.: Labeled faces in the wild: updates and new reporting procedures. Technical report, Department of Computer Science, University of Massachusetts Amherst, Amherst (2014)
19. Liao, S., Lei, Z., Yi, D., Li, S.Z.: A benchmark study of large-scale unconstrained face recognition. In: 2014 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–8 (2014)
20. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: a dataset and benchmark for large-scale face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 87–102. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_6
21. Hu, W., Huang, Y., Zhang, F., Li, R., Li, W., Yuan, G.: SeqFace: make full use of sequence information for face recognition. arXiv preprint arXiv:1803.06524 (2018)
22. Deng, J., Guo, J., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. arXiv preprint arXiv:1801.07698 (2018)

23. Wan, W., Zhong, Y., Li, T., Chen, J.: Rethinking feature distribution for loss functions in image classification. arXiv preprint arXiv:1803.02988 (2018)
24. Zheng, Y., Pal, D.K., Savvides, M.: Ring loss: convex feature normalization for face recognition. arXiv preprint arXiv:1803.00130 (2018)
25. Wang, H., et al.: CosFace: large margin cosine loss for deep face recognition. arXiv preprint arXiv:1801.09414 (2018)
26. Liu, J., Deng, Y., Bai, T., Wei, Z., Huang, C.: Targeting ultimate accuracy: face recognition via deep embedding. arXiv preprint arXiv:1506.07310 (2015)
27. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2015)
28. Huang, G., Liu, Z., Weinberger, K.Q., Laurens, V.D.M.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1 (2017)
29. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 5987–5995 (2017)