KNOT MAGNIFY LOSS FOR FACE RECOGNITION

Qiang Rao, Bing Yu, Yun Yang, Bailan Feng

Noah's Ark Laboratory, Huawei Technologies Co., Ltd.
Email:{raoqiang3, yubing5, yangyun18, fengbailan}@huawei.com

ABSTRACT

Deep Convolutional Neural Netowrks (DCNN) have significantly improved the performance of face recognition in recent years. Softmax loss is the most widely used loss function for training the DCNN-based face recognition system. It gives the same weights to easy and hard samples in one batch, which would leads to performance gap on the quality imbalanced data. In this paper, we discover that the rare hard samples in the training dataset has become a main obstacle for training a robust face recognition model. We propose to address this problem by a new supervisor signal that pays more attention to the rare hard samples and reduces the effects of the easy samples relatively. Our proposed novel Knot Magnify (KM) loss modulates the classical softmax loss to suppress the influence of easy samples and up-weight the loss of hard samples during training. Our results show that after training with KM loss, face recognition model is able to get competing accuracy on the well-known face recognition benchmark LFW dataset and the challenging CFP dataset.

Index Terms— Deep convolutional neural networks, Face recognition, Loss function, Quality imbalance

1. INTRODUCTION

Face Recognition with Deep Convolutional Neural Networks (DCNNs) has achieved great success in recent years. Despite the excellent performance of DCNN-based face reccognition methods on the most accredited face dataset Labeled Faces in the Wlid (LFW) [1], it is still diffcult to maintain accuracy for face images with extreme variations in view points, resolution, occlusion, and image quality. Imbalanced distribution of training data with various quality is the key reason for this performance decay. Many famous publicly available datasets, such as CASIA-Web-Face [2] and MS-Celeb-1M [3], suffer from such data quality imbalance and annotation noises. These datasets generally contain large amount of high quality frontal face images, whereas unconstrained and challenging face images with extreme variations occur rarely.

The general pipeline for training a face recognition system using DCNN is illustrated in Fig.1. Face images from different people are fed into DCNN. Face features are extracted after successive convolutional and fullly-connected layers,

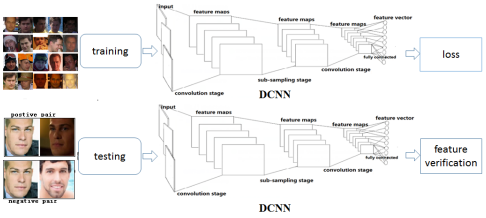

Fig. 1. The framwork of face verification system. Large amount of labeled faces are fed into the network at training stage to optimize the parameters, while at test stage, pairs of faces are verified with the similarity of extracted face features.

and then connected to specific metric loss function. Modeling the face recognition task as a large-scale multi-class classification is the most simple and effective way, and softmax loss is the most widely used loss function for such task. Unlike triplet loss [4] and contrastive loss [5] which specifically utilize finely selected hard samples, softmax loss minimizes the dissimilarity between the conditional probability of predicted label and the true label distribution in a minibatch. However, softmax loss is prone to learn from the frequently appeared samples since the loss weight of each sample is equal. Thus, after training on previously mentioned imbalanced dataset, the learned feature representations are biased to high quality samples, disgarding the effect of rare low quality faces. As a consequence, the recognition performance would drastically drop when encountering low quality faces. To conquer the gap of performace degradation, one possible solution is to enlarge training dataset with sufficient intrinsic variations. However, collecting such kind of datasets is very costy and time-consuming. Thus, it is quite appealing to design a loss function that can efficiently utilize the rare hard samples in the existing imbalanced dataset to ease the effort.

We propose a new loss function, Knot Magnify (KM) loss, to enhance the effect of hard samples in the training of DCNNs. A weight is added to each sample's loss during training. Self-adaptively, larger weights are assigned to the rare hard samples and samller weights are given to the easy samples when computing the training loss. It is observed that the softmax outputs reflect the image quality, as depicted in Fig.2, with samples from MsCeleb-1M [3]. Profile, occluded, and

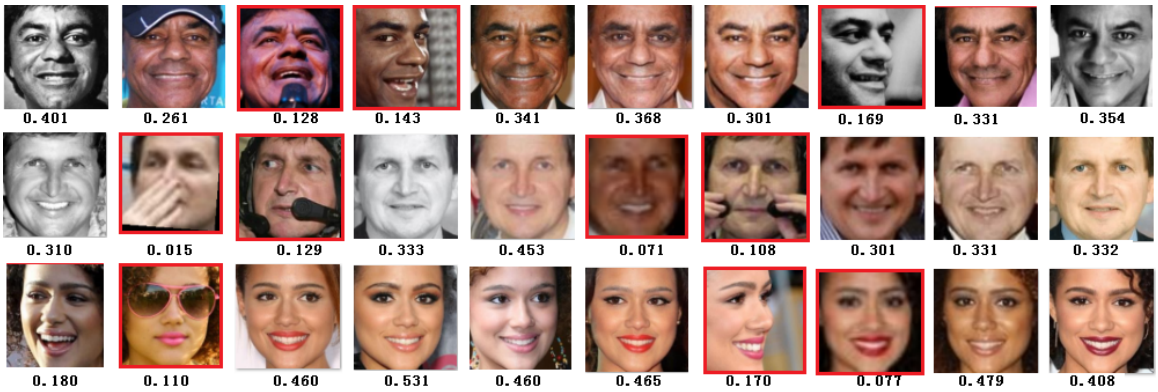

Fig. 2. The softmax outputs of samples frome MsCeleb-1M dataset. Each row contains randomly selected samples from one subject. The value under each image is its softmax output as conditional probability p_k. One may notice that the hard samples (the red boxed ones with extreme poses, occulsion or blurrness) have relatively smaller values than the clear frontal ones.

blurry faces have relatively lower output values. The softmax output can be treated as the conditional probability $p(y|x, D)$ in statistical perspective, where y is the label of sample x. K-M loss directly utilizes $p(y|x, D)$ to adaptively adjust the loss weight for each sample. The proposed KM loss has several advantages. First, KM loss is based on well-studied softmax loss and can be easily implemented by off-the-shelf public available deep learning toolboxes such as Caffe [6] and Tensorflow [7]. Secondly, it relatively enhances the loss effect of rare hard samples and guides DCNNs to learn more information from hard samples. Moreover, it can be combined with other auxilary loss function to further improve the disctiminative power.

In summary, our main contributions are as follows:

1. We propose a novel and simple KM loss to magnify loss effect of rare hard samples which can be used for training more robust face recognition models.

2. The single hyperpramater γ of knot magnify loss is theoretically studied. The process of selecting an appropriate γ is presented.

3. We show that the proposed KM loss is easy to implement and embed for DCNNs. The one-net-one-loss system is able to get competing performance on LFW. Combining KM loss with center loss [8], we get state-of-the-art result on the LFW dataset on both face verification and identification tasks.

2. RELATED WORK

Over the past few years, many face recognition methods based on deep learning [4, 5, 8, 9] have been proposed and have made significant improvement in the accuracy of face recognition. They achieved outstanding performances on LFW dataset and some even surpassed human performance. These methods differ in the type of loss functions whereas they all utilized DCNNs to learn face representation. For face verification, we expect the features of positive pair (from same subject) to be close while the features of negative pair (from different subjects) far apart. Chopra et al. [5] firstly proposed contrastive loss of mapping two face images to a distance. They train siamese networks by contrastive loss to minimize the distances between postive pairs and maxmize distances between negative pairs. FaceNet [4] introduced triplet loss to learn the metric using hard triplets of face samples.

Most recent approches [8, 9, 10, 11] use face images along with their subject labels to learn discriminative features in a multi-class classification framework. They train a DCNN with softmax loss to learn features which are later used either to directly compute the similarity score between a pair of faces or to train a discriminative metric embedding [12]. DeepID method [9] combined the identification task and verification together which jointly training with softmax loss and constrastive loss. [8] proposed to train the deep network under the supervision of softmax loss and center loss which pulls the deep features of the same subject to their centers. With the joint supervision, the learned features have more dicriminative power.

Recently, a few algorithms have used feature normalization during training to improve performance. SphereFace [13] proposes angular softmax (A-softmax) loss and introduces auglar margin to learn angularly discriminative features . Normalizing the feature on a hypersphere demonstrated its effectiveness in [14, 15]. By constrainting the norm of feature to a constant value [14], similar attention is payed to both good and bad quality faces. COCO loss [15] normalizes the features to a constant scale and compacts deep features to their class-reltated centers in softmax formulation. They achieve state-of-the-art results on LFW benchmark.

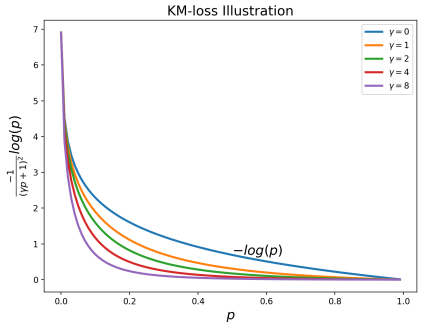

Fig. 3. The proposed KM loss which multiplies a factor $\frac{1}{(\gamma p_t+1)^2}$ with softmax loss. As one can see, setting γ to a suitable value, the KM loss will suppress the loss of easy samples and maginify that of hard samples. When $\gamma = 0$, the KM loss degenerates into softmax loss

3. PROPOSED METHOD

In this section, we introduce the proposed KM loss in details.

3.1. The softmax-based KM loss

Softmax loss is the most widely used loss function in training the deep neural netowrks. It is generally presented within a minibatch of n samples as:

$$L_s = -\sum_{k}^{n} log(p_k^i) \tag{1}$$

where $p_k^i = \frac{e^{w_i f(x_k)}}{\sum_j e^{w_j f(x_k)}}$ is the softmax output, $f(x_k)$ is the deep nets feature of sample x_k. The component of softmax loss $S(p_k^i) = -log(p_k^i)$ is the loss of single sample x_k.

To relatively magnify the loss effect of hard samples, we add a meticulously constructed weight term $\frac{1}{(\gamma p_t+1)^2}$ to each softmax loss component. The weight term will suppress the loss effect of easy samples and improve that of hard samples. Such that, our proposed KM loss is defined as:

$$L_k = -\sum_{k}^{n} \frac{1}{(\gamma p_k^i + 1)^2} log(p_k^i) \tag{2}$$

Accordingly, the component of KM loss is defined as $K(p_k^i) = -\frac{1}{(\gamma p_k^i+1)^2} log(p_k^i)$. The curve of the proposed KM loss is depicted in Fig.3, showing that modifying the loss weight of each sample will have different impacts on the easy samples that have larger softmax output p_k^i and hard samples that have samller p_k^j.

3.2. Theoretical analysis of the factor γ

We quantitatively analyze the loss effect by considering it as the cumulative loss corresponding to softmax output p_k^i which ranges in $[0, 1]$. First of all, we define the normalized softmax loss S^N and KM loss K^N as:

$$S^N(p_k^i) = \frac{S(p_k^i)}{\int_0^1 S(t)dt} = \frac{log(p_k^i)}{\int_1^0 log(d)dt} = -log(p_k^i) \tag{3}$$

$$K_\gamma^N(p_k^i) = \frac{K(p_k^i)}{\int_0^1 K(t)dt} = \frac{-\frac{log(p_k^i)}{(\gamma p_k^i+1)^2}}{\int_0^1 -\frac{log(t)}{(\gamma t+1)^2}dt} \tag{4}$$

$$= \frac{-\gamma log(p_k^i)}{log(\gamma+1)(\gamma p_k^i+1)^2} \tag{5}$$

Then we can get:

$$R_\gamma(p) = \frac{K_\gamma^N(p)}{S^N(p)} = \frac{\gamma}{log(\gamma+1)(\gamma p+1)^2} \tag{6}$$

A critical point p_c stands for the separation point of easy and hard samples in terms of softmax outputs. We expect $R_\gamma(p) \geq 1$ when $p \leq p_c$ (magnifying the effect of hard samples by introducing relatively larger loss) and $R_\gamma(p) < 1$ when $p > p_c$ (suppressing the influence of easy samples by introducing relatively smaller loss). Assuming $\gamma > 0, p > 0$, letting $R_\gamma(p) = 1$, we get the critical point p_c:

$$p_c(\gamma) = \sqrt{\frac{1}{\gamma log(1+\gamma)}} - \frac{1}{\gamma} \tag{7}$$

From the Eq.7, $p_c(\gamma)$ is a rigid monotonically decreasing function. By sampling different γ values, Table.1 is obtained. After examing the softmax output values of the training dataset (some examples shown in Fig.2), one may notice $p_c = 0.175$ is a valid critical point. This yields an appropriate choice for the hyperparameter γ as 2.

Table 1. Upward approximate critical point p_c for given γ

γ	0.1	1	2	4	6	8
$p_c(\gamma)$	0.244	0.202	0.175	0.145	0.126	0.114

4. EXPERIMENTS

4.1. Experiments settings

We train our model on publicly available MS-Celeb-1M dataset [3] which is carefully cleaned by ourselves. Inception-Resnet-50 [16] is adopted as the DCNN model and implemented with Tensorflow. We add Batch Normalization [17] to each convolutional layer and set penultimate layer to be 256-D as the deep face embedding size. We first align the input face images with MTCNN [18] and crop to 160×160. Each RGB face image is normalized to zero mean and unit variance. The network is trained on a Nvidia Tesla K80 GPU with batch size of 256. The learning rate starts from 0.01, and declines 10 times after 80k, 150k, 200k until converging.

4.2. Expriments on LFW

The Labeled Face in The Wild (LFW) dataset [1] contains 13233 images from 5749 distinct people, with large variations in poses, expressions and illuminations.

Face Verification. We follow the standard unrestricted protocol of face verification on the LFW dataset. 6000 pairs of faces are equally divided into 10 folds. Face verification judges whether a pair of faces to be of the same person by comparing the similarity of their features against a pre-set threshold. The accuracy is reported as the mean recogntion accuracy (mAcc) accoss all 10 folds. We test our model with both KM loss only and KM + Center loss respectively. We found that $\gamma = 2$ works best in our expriments. The results are listed in Table 2.

Table 2. Verification accuracy on LFW dataset. The proposed approach is compared with different state-of-the-art methods.

Methods	Data	#loss	#nets	mAcc.(%)
DeepFace [10]	4M	2	3	97.35
VGGFace [11]	2.6M	1	2	98.95
Facenet [4]	200M	1	1	99.63
DeepID2 [9]	300k	2	25	99.47
Center Loss [8]	700k	2	1	99.28
Sphereface [13]	500k	1	1	99.42
Softmax	3.7M	1	1	99.10
KM Loss	3.7M	1	1	**99.31**
Center+KM Loss	3.7M	2	1	**99.53**

Face Identification. The probe-gallery testing on the LFW was proposed in [19]. We follow the close-set protocol of face identification. The gallery contains 4249 identities, where each identity possesses only one face image. The probe set contains 3243 face images, which belong to the identities within the gallery set. The performance is measured by the rank-1 identification accuracy. From the results in Table 3 one can see that our method outperforms other state-fof-the-art methods on LFW dataset.

4.3. Expriments on CFP

Celebrities in Frontal-Profile (CFP) data set is proposed by Sengupta [21] and aims to isolate the factor of pose variation in terms of extreme poses like profile, where many features are occluded, along with other in the wild variations. CFP consists of 500 distinct subjects, each with 10 frontal and 4 profile images. The evaluation protocol includes frontal-frontal (FF) and frontal-profile (FP) face verification, each having 10 folds with 350 positive pairs and 350 negative pairs. In this paper, we report both the CFP-FF and CFP-FP verification performance.

Table 3. Rank-1 identification accuracy on LFW dataset. The evaluation follows the close-set protocol proposed by [19]. The proposed approach is compared with different state-of-the-art methods.

method	#Net	Protocol	Rank 1(%)
DeepFace[10]	7	unrestricted	64.90
Web-Scale[20]	4	unrestricted	82.50
DeepID2+[9]	25	unrestricted	95.00
VGGFace[11]	1	unsupervised	74.10
Center Loss[8]	1	unsupervised	94.05
Softmax	1	unsupervised	95.33
KM Loss	1	unsupervised	**97.26**
Center+KM Loss	1	unsupervised	**97.31**

Table 4. CFP verification accuracy

data	Loss	mAcc.(%)
CFP-FF	Softmax	99.19
	KM Loss	99.43
	Center+KM Loss	99.46
CFP-FP	Softmax	91.27
	KM Loss	91.71
	Center+KM Loss	93.39

From the results in Table 4, we find that the performance deteriorates after switching from Frontal-Frontal verification to Frontal-Profile verification, which validates our former analysis of softmax loss. With our KM loss, the performance is improved in both CFP-FF and CFP-FP protocols.

5. CONCLUSION

In this paper, we investigated once again the problem of learning deep representation of face. To address the problem of lacking learning ability from hard samples, we modified the widely adopted softmax loss by proposing $KM\ loss$ which assigns weights to training samples according to its softmax output in order to suppress the influence of easy samples and magnify the effect of rare hard samples. Our approach is simple and easy to impelment. Moreover, the proposed $KM\ loss$ can be easily combined with other auxilary losses which benefit to learn a more robust model. We demostrated our method's effetiveness on the well-known LFW dataset and the challengeing CFP dataset. Experimental results show that our method gets competing accuracy on both datasets.

6. REFERENCES

[1] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep., Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[2] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, "Learning face representation from scratch," *Computer Science*, 2014.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[5] Sumit Chopra, Raia Hadsell, and Yann LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 539–546.

[6] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[7] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[8] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.

[9] Web-Scale Training WST, "Deeply learned face representations are sparse, selective, and robust," *perception*, vol. 31, pp. 411–438, 2008.

[10] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[11] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al., "Deep face recognition.," in *BMVC*, 2015, vol. 1, p. 6.

[12] Jun-Cheng Chen, Vishal M Patel, and Rama Chellappa, "Unconstrained face verification using deep cnn features," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–9.

[13] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, vol. 1.

[14] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.

[15] Yu Liu, Hongyang Li, and Xiaogang Wang, "Rethinking feature discrimination and polymerization for large-scale recognition," *arXiv preprint arXiv:1710.00870*, 2017.

[16] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, 2017, vol. 4, p. 12.

[17] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[18] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[19] Lacey Best-Rowden, Hu Han, Charles Otto, Brendan F Klare, and Anil K Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2144–2157, 2014.

[20] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Web-scale training for face identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2746–2754.

[21] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs, "Frontal to profile face verification in the wild," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–9.