Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 6660961, 8 pages
https://doi.org/10.1155/2021/6660961

Research Article

Improved Loss Function for Image Classification

Chenrui Wen, Xinhao Yang, Ke Zhang, and Jiahui Zhang

School of Mechanical and Electrical Engineering, Soochow University, Suzhou, China

Correspondence should be addressed to Xinhao Yang; yangxinhao@163.com

Received 2 November 2020; Revised 27 December 2020; Accepted 11 January 2021; Published 23 January 2021

Academic Editor: José Alfredo Hernández-Pérez

An improved loss function free of sampling procedures is proposed to improve the ill-performed classification by sample shortage. Adjustable parameters are used to expand the loss scope, minimize the weight of easily classified samples, and further substitute the sampling function, which are added to the cross-entropy loss and the SoftMax loss. Experiment results indicate that improvements in all classification performance of our loss function are shown in various network architectures and on different datasets. To summarize, compared with traditional loss functions, our improved version not only elevates classification performance but also lowers the difficulty of network training.

1. Introduction

Loss function is used to measure the difference between the output data of the model and the actual sample data and its role is to guide the model to move towards convergence in the training process, during which minimizing the loss value is virtually to achieve model fitting of training data and the minimum test error of the model and eventually to accurately classify new samples [1].

SoftMax loss is considered as the very most fundamental loss function in image classification, featuring easy optimization and quick contract. SoftMax loss is often in combined application with cross-entropy loss to guarantee accurate classification of the known categories [2, 3]. For some simply classified image datasets, it is adequate to only ensure accurate classification of the known categories. But when it comes to fine image classification, adopting SoftMax alone is far from enough. To achieve better generalization performance, more elaborated classification characteristics are required, such as "intra-class sampling variation" and "inter-class sampling variation," which are beyond the reach of SoftMax loss's direct optimized targets. Therefore, researchers began to turn their insights from Euclidean space to metric space to obtain fine features for fine classification [4–6].

The core idea of loss function on the metric space is as follows: to shorten the similar images embedded in the space and to push the dissimilar images far away. Simple metric learning methods, such as DeepID2, are designed to gain features by combining SoftMax loss and contrastive loss [7], while the renowned Facenet further employs triplet loss. But it is not enough [8] for loss function by employing simple metric space. Given N samples, the complexity of SoftMax loss after traversing all samples is only $O(N)$; nevertheless, the complexities of contrastive loss or triplet loss go up to $O(N^2)$. Otherwise, it is impossible to traverse simply. Effectively searching for good training samples, or hard example mining, is indispensable in the complex training process, especially when categories in large quantity add difficulty to find good examples. Margin loss on the basis of image triplets, proposed by Wu et al. [5], adopts distance weighed sampling to iron out training problems and to search for good training samples. This sampling strategy cushions the impact from data imbalanced samples, but there is a high probability of omitting samples with huge relevance and influencing training.

The loss function we propose decreases the weight of easily classified samples from the perspective of probability. The training difficulty is lower in our loss function and a large volume of computation employed by metric methods is avoided. Additionally, our loss function avoids the issue of omitting concerned samples in traditional sampling methods. The innovative points of our research are listed as

follows: (1) propose a new loss function, (2) employ the new version to successfully lower training difficulty, and (3) apply the new version to deliver better classification performance compared with traditional loss functions.

To outline the paper, Section 2 focuses on related work of current research development in image multi-classification. Section 3 is an introduction of theoretical knowledge employed in this paper and our loss function. Section 4 describes experiment methods and outcomes demonstrating our loss function. Section 5 is a summary and discussions of all the above experiments. Section 6 summarizes all the contents.

2. Related Work

In recent years, the accuracy of image classification has been significantly improved under the framework of deep learning. Many researchers lay much emphasis on polishing the framework rather than noticing loss functions [9–12]. The work of image classification can be generally separated into two parts: binary classification and multi-classification [13, 14].

SoftMax is widely applied in image classification for its easy optimization and quick contract. It enables all the categories to possess the maximum logarithm likelihood in the probability space, or, in other words, to guarantee the accurate classification of all categories [2, 15]. The emergence of classical SoftMax makes indispensable contributions to the development of image multi-classification including accomplishing certain level of classification effect in the current superior network architectures of deep learning. However, better outcomes will be achieved through adopting the improved loss function in some simple network architectures.

One of the most technological routes is to reduce intra-class distances and expand inter-class distances. Then, these improved methods are combined with SoftMax. Hadsell et al. introduced contrastive loss function which simultaneously minimizes image pairs in positive samples and predefines borders to expand distance between image pairs in negative samples [16]. Similarly, Hoffer and Ailon proposed a triplet loss substituting previous image pairs with image triplets [6]. Combinatorial explosion of the quantity of image pairs is the major weakness of the above two methods. Contrastively, center loss [17] shuns this shortcoming without the necessity to calculate distances between image pairs or image triplets. However, center loss minimizes the distance between features and relevant classification centers, which leads to inconsistent distance measurement in feature space. Targeting this issue, Liu et al. added phase margin to SoftMax loss function [2]. Though such deep learning methods show well-performed features, triplet loss may lead to training difficulties. Therefore, to reduce training difficulties, various sampling methods rise to the occasion in the course of searching for training samples on datasets. Wu et al. [5] proposed a margin loss on the basis of triplet to lower the difficulty of network training, which adopts distance weighted sampling to facilitate back propagation of loss more prudently.

All methods mentioned above play a significant role in image classification; however, poor classification results and training difficulties also exist in the computation of loss functions based on distance. From another perspective, training difficulties caused by distance calculation can also be diminished by distance itself. Apart from various sampling methods in the classification, taking hard example mining into account and reducing the weight of easily classified samples can be an effective option.

3. Improved Loss Function

Metric learning in collocation with sampling method is often used to reduce computational complexity in simple image classification. Margin loss based on triplet employs distance weighted sampling method resulting in omission of relevant samples in the sampling course. Therefore, this paper explores the direct reduction of the weight of easily classified samples free of any sampling form to deal with the issue of sample losses.

The loss function proposed in this paper mainly solves the problem of ignoring small inter-class samples by reducing the weight of easily classified inter-class samples. On the other hand, our loss function reduces the computational complexity and diminishes the training difficulty.

3.1. Loss Function Design. Our loss function formulas are listed as follows:

$$L(x) = -f(-\beta x)e^{\gamma}\log(f(x)). \tag{1}$$

The loss is visualized for several values of $\beta \in [4, 10]$, $\gamma \in [3, 10]$ in Figures 1 to 4 . From the figures, the best experiment result is achieved when β equals 4 and γ equals 5 simultaneously. Details are shown in the fourth part of the paper. The definition of $f(x)$ in formula (1) is $f(x) = \begin{cases} p, & y = 1, \\ 1 - p, & \text{otherwise.} \end{cases}$, and the framework of the whole formula derives from cross-entropy loss for binary classification [18]:

$$L_{ce} = \begin{cases} -\log(p), & y = 1, \\ -\log(1 - p), & \text{otherwise.} \end{cases} \tag{2}$$

In formula (2), the ground-truth class is specified by $y \in \{\pm 1\}$. Moreover, $p \in [0, 1]$ is the estimated probability of the model for the class with label $y = 1$. The similar definition is $p_n = \begin{cases} p, & y = 1, \\ 1 - p, & \text{otherwise.} \end{cases}$, and rewrite $L_{ce} = -\log(p_n)$.

We mainly engage in image multi-classification where most researchers combine loss functions with SoftMax loss to elevate classification performance. To accomplish image multi-classification, we introduce SoftMax function to formula (3) to make that integration. The SoftMax function formula is formula (3) and the combined version is formula (4). In formula (3), the SoftMax value of z_c is the ratio of the index of the element to the sum of the indices of all elements.

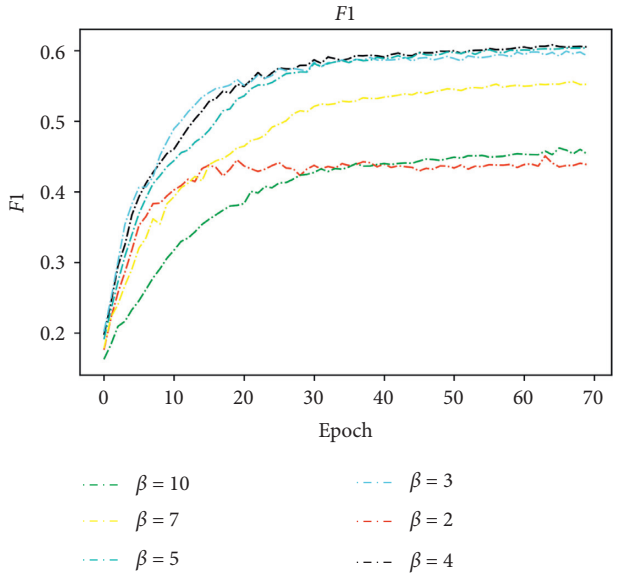

Figure 1: $F1$ change curve at different β values. Some curves represented by other values are omitted due to coincidence.

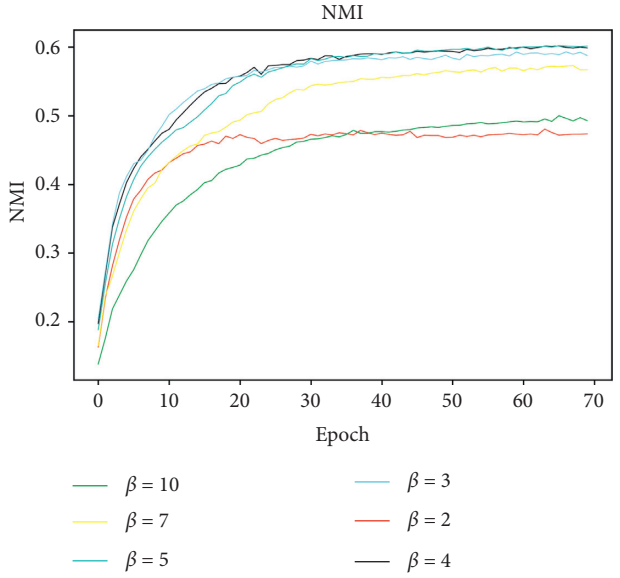

Figure 2: NMI change curve at different β values. Some curves represented by other values are omitted due to coincidence.

$$y_c = \varsigma(z)_c = \frac{e^{z_c}}{\sum_{d=1}^{C} e^{z_d}}, \quad \text{for } c = 1 \ldots C, \tag{3}$$

$$L(x) = -\text{softmax}(-\beta x)e^{\gamma}\log(\text{softmax}(x)). \tag{4}$$

Our loss function has two properties. (1) When the sample classification is inaccurate and $f(x)$ is relatively small, $f(-\beta x)$ approaches 1 and no impact on loss occurs. When $f(x)$ tends to 1, $f(-\beta x)$ approaches 0 and there is a loss decline of well-classified samples. (2) The parameter e^{γ} expands differences among various samples. When $f(x) \approx 0$, the value of $f(-\beta x)e^{\gamma}$ is huge and there is a huge discrepancy between $L(x)$ calculated by this formula and the loss

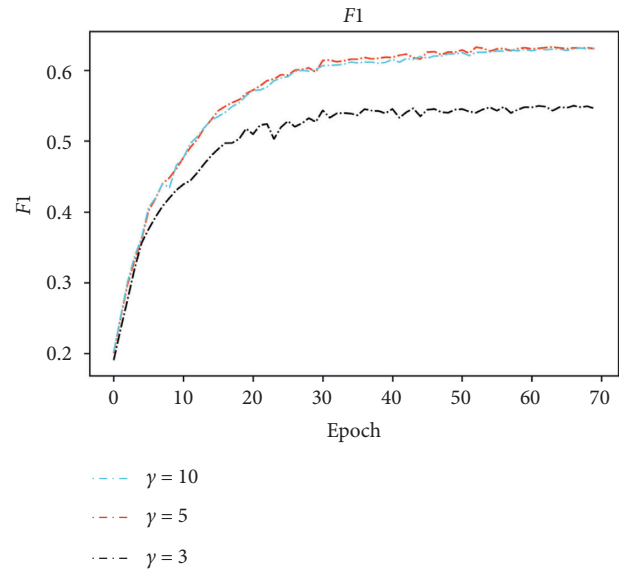

Figure 3: $F1$ change curve at different γ values. Some curves represented by other values are omitted due to coincidence.

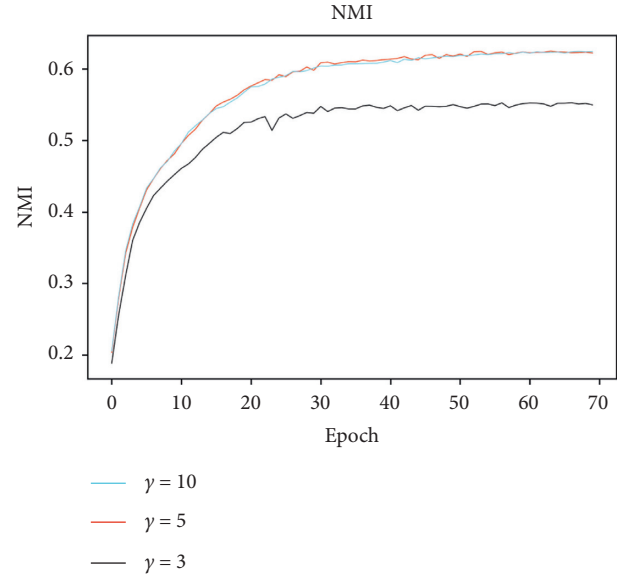

Figure 4: NMI change curve at different γ values. Some curves represented by other values are omitted due to coincidence.

calculated without the parameter. When $f(x) = 0.88$, the parameter also exerts influence on the result. It can be clearly seen that adjusting the parameter not only reduces the control of easily classified sample loss but also enlarges low loss scope; for example, when an example is classified with $f(x) = 0.88$, the loss value is lower than the cross-entropy loss by 10^7.

3.2. Derivatives. The gradient computations for the improved loss are given in equations (5) to (8). Since the loss function is mostly used for image multi-classification, gradient computations are performed on the basis of the final equation (4). For conciseness, we denote $\text{softmax}(x)$ as $s(x)$ in all these equations.

$$\frac{\partial L}{\partial x} = \beta e^{\gamma} s(-\beta x)\log(s(x)) - \frac{s(-\beta x)}{s(x)} e^{\gamma} \frac{\partial s}{\partial x}. \quad (5)$$

Considering both cross-entropy (CE) and our proposed loss function, specifically, we define a quantity x_t as follows:

$$x_t = yx, \quad (6)$$

where $y \in \{\pm 1\}$ specifies the ground-truth class as before. We can then write $p_n = \sigma(x_t)$, which is compatible with the definition of p_n in equation (1). An example is correctly classified when $x_t > 0$, in which case $p_n > 0.5$.

For reference, derivates for cross-entropy function w.r.t. x are

$$\frac{\partial C\,E}{\partial x} = y(p_n - 1), \quad (7)$$

$$\frac{\partial L}{\partial x} = e^{\gamma} y p_n (p_n - 1)(\log(p_n) + 1). \quad (8)$$

It is worth noting that x is an input after being processed by $-\beta$.

3.3. Assessment Criteria. To assess loss function, we applied two criteria: $F1$ and NMI. $F1$, also called macro-$F1$ [19], counts TP, FP, FN, and TN of various classifications and calculates their precision and recall rate, respectively, to obtain the average values of the $F1$ separately. Macro-$F1$ has a more superb assessment capability by treating each classification equally and considering both their precision and recall rate. NMI can be used to measure the fitting of the distribution of these two datasets [20]. The larger the value is, the more consistent the classification is with the real situation.

4. Experiments

We mainly exhibit four groups of experiments in this section. In the first group, we determined the values of β and γ in the formula above and selected the optimal parameters. In the second group, we compared different kinds of loss function on the CIFAR-10 [13], including margin loss, triplet loss, and our loss function, and reported the $F1$ and NMI values. In the third group, we verified that our loss function boasted good performance in various network architectures. In the fourth group, our loss function also displayed the performance strength in other datasets. All experiments were implemented under the PyTorch framework, on NVIDIA 2080Ti GPU.

Figure 5 shows four main categories: frog, boat, truck, and dog. Because the original dataset image is only 32 *32 in size, the sample obtained by sampling shows a lower sharpness. And these images are sampled by margin loss framework.

4.1. Determination of Parameters. We have two variable parameters β and γ in our loss function, both of which serve the smoothness of the loss function in order to strengthen its information transmission capability in the course of back propagation. In our experiments, we set β to 2, 3, 4, 5, 7, and 10 for CIFAR-10. Similarly, we set γ to 3, 5, and 10. We first set γ to 3 and then adjusted β to the above six values separately, tested the CIFAR-10 dataset through GoogLeNet training, and selected the β value that maximized the values of $F1$ and NMI. Provided the designated β value, we proceeded to the determination of the γ value by selecting one that also maximized the values of $F1$ and NMI in the experimental analysis. The outcome is shown in Figures 1–4.

To determine the β value, we first randomly selected the first and the last values within the range of [2, 16]. Since experiments showed little difference between the two corresponding $F1$ values and the slightly smaller $F1$ values, we continuously tried different values by dichotomy within that range and eventually determined the optimal β value as 4. Several curves that overlapped are dismissed in Figure 1 due to the relatively large number of curves.

In addition to $F1$ standard, we also adopted NMI standard for assessment in the course of the determination of β value. It can be seen from Figure 2 that NMI curve shares the same changing trend with $F1$ curve under different β values, also verifying that the optimal β value is 4. Again, several curves that overlapped are dismissed due to the relatively large number of curves.

Similarly, we employed both $F1$ and NMI standards to determine the γ value. We selected the γ value within the range of [2, 21] and determined the optimal γ value as 5. Likewise, we also omitted several curves that overlapped in the following graph due to the relatively large number of curves. The $F1$ curve and the NMI curve are shown in Figures 3 and 4, respectively.

4.2. Comparison with Other Loss Functions. The CIFAR-10 dataset altogether had 60000 colorful images of size 32 *32 in Passage 3, which are classified into 10 categories, each with 6000 images. 50000 images constituted 5 training groups on average and the other 10000 images formed one single testing group. In the testing group, we selected 1000 images in each of the 10 categories, leaving the rest randomly arranged to constitute the training group. The number of images for each category in a training group is not necessarily the same, but each category has 5000 images generally [13].

For CIFAR-10, we trained the GoogLeNet [22] of depth with different loss functions. The network was trained for 70 epochs and we set 0.00001 for the learning rate. We used a weight decay of 0.0004 and the momentum number of Adam was set to 0.9. The models were trained by ourselves on CIFAR-10 dataset since the GoogLeNet with triplet loss or margin loss had not been used in the previous works. The proposed loss function outperformed the triplet loss and margin loss for GoogLeNet models.

The classification criteria are shown in Figure 6. Compared with traditional loss functions including proxyna [23], *n*pair [24], triplet [6], and margin loss [5], whose $F1$ values were eventually stable from 0.5 to 0.6, our loss function has achieved a higher $F1$ value above 0.6 after training 70 epochs. To be more specific, our loss function is eventually stable at 0.633 and demonstrates better classification results.

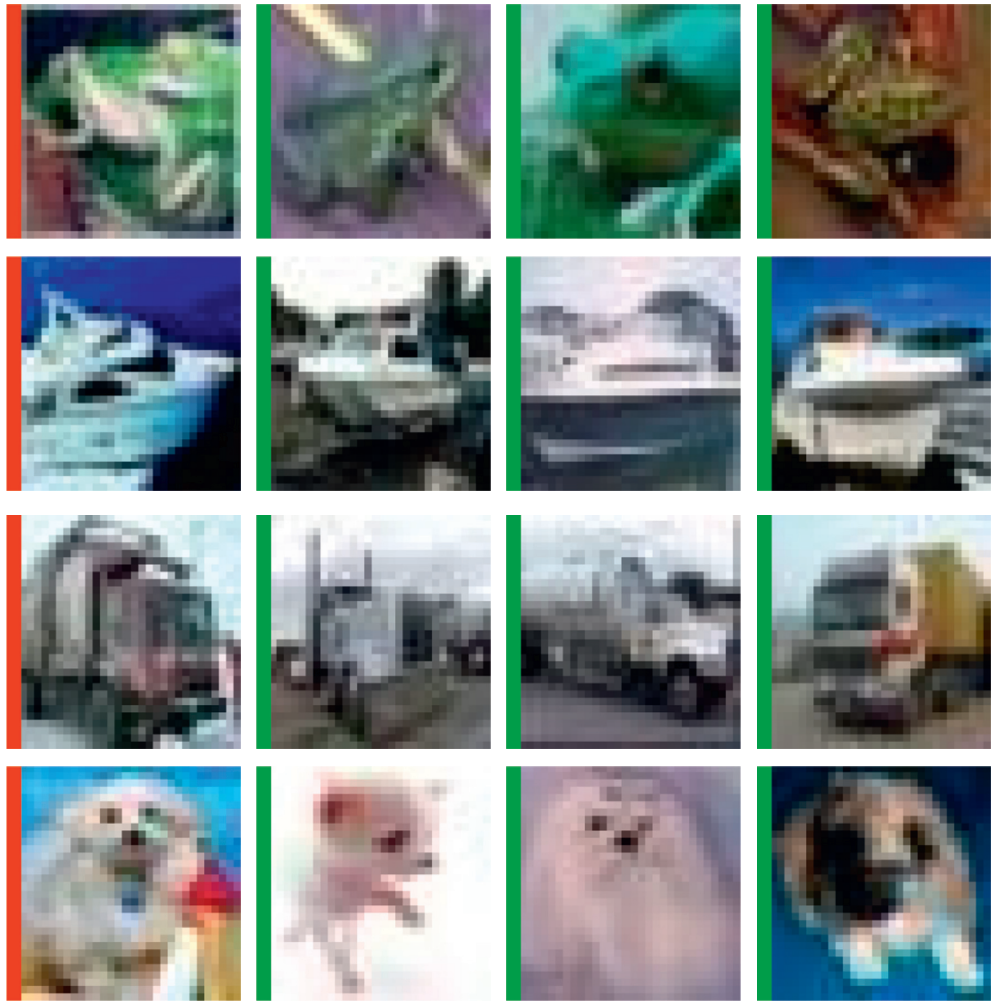

Figure 5: Some example images from the CIFAR-10 dataset. Here are four types of pictures in the CIFAR-10 dataset, namely, frogs, boats, trucks, and dogs.

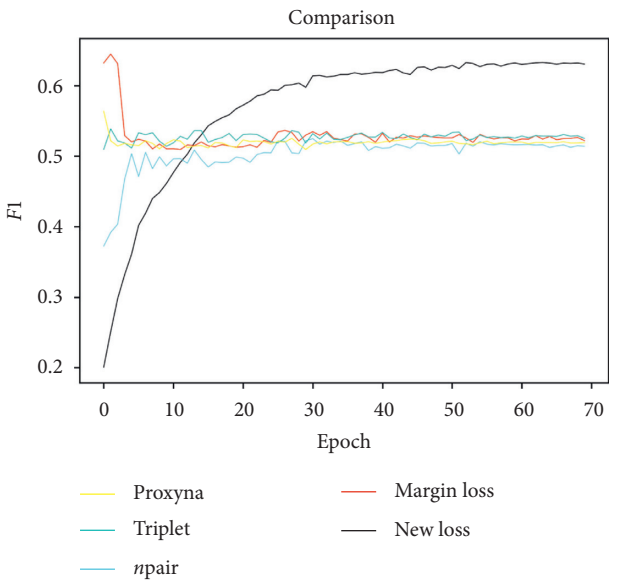

Figure 6: Comparison graph of different loss functions with the $F1$ standard. A higher $F1$ value above 0.6 after training 70 epochs.

As can be seen from Figure 7, the NMI value of the tradition loss function was eventually stable from 0.4 to 0.5. But our loss function has achieved a higher NMI value above 0.6, which is stable at 0.625 eventually.

4.3. Comparison in Other Network Architectures. The above training and testing experiments were conducted under GoogLeNet; furthermore, we also did similar comparison experiments under ResNet [3]. On the CIFAR-10 dataset, we trained the ResNet with triplet loss, margin loss, and other loss functions. The network was trained for 14 epochs, and the learning rate was set to 0.00001. We trained the network for 70 epochs and 0.00001 was set for the learning rate. We used a weight decay of 0.0004 and the momentum number of Adam was set to 0.9. The models were trained by ourselves on CIFAR-10 dataset since the ResNet with triplet loss or margin loss had not been used in the previous works. The proposed loss function outperformed the triplet loss and margin loss for ResNet models. The comparison of classification performance is shown in Figure 8.

According to Figure 8, the two curves representing NMI value and $F1$ value, respectively, by our loss function are

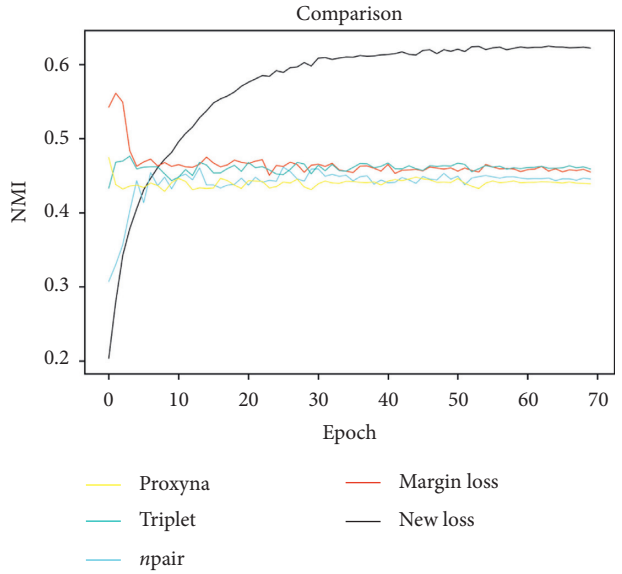

Figure 7: Comparison graph of different loss functions under the NMI standard. A higher NMI value above 0.6.

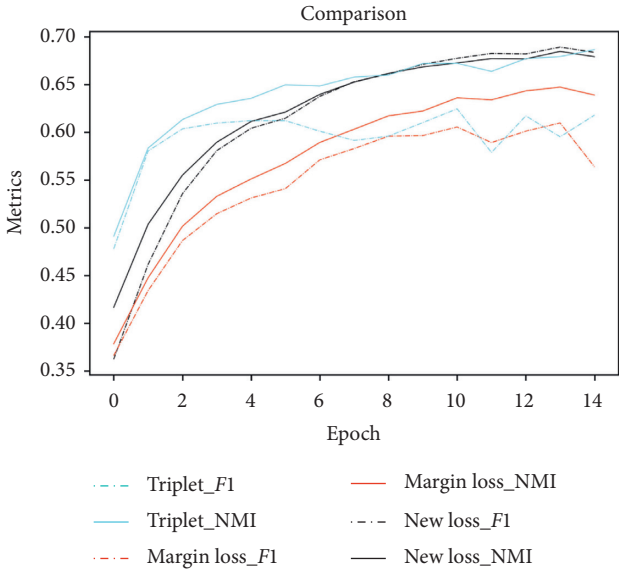

Figure 8: Comparison graph of different loss functions. The comparison experiments, implemented on ResNet, also showed that our loss has better performance.

above the curves by other loss functions. It is indicated that our loss function has better classification performance. Despite the slightly smaller NMI value of our loss than that of triplet, the $F1$ value of ours is 0.06 higher. Therefore, viewing comprehensively from the two standards, our loss contributes greater than triplet loss and margin loss.

4.4. Accomplishments on Other Datasets. We introduce other commonly used datasets for the experiments. Then, we compare our loss function with other loss functions by using renset50 or GoogLeNet on these datasets. The CIFAR-100 dataset, a fine classified dataset in image classification we used, was divided into 20 categories and 100 subcategories. Each subcategory contains 600 images: 500 for training and 100 for testing. Each image has two labels, "fine" and "coarse." In addition, Fashion-MNIST dataset was used to test the algorithm performance. The image content of this dataset is more complex than that of MNIST dataset, but its data distribution is the same as that of MNIST, which is a basic image dataset [25].

On the CIFAR-100 dataset, we trained GoogLeNet with triplet loss, margin loss, and other loss functions. We trained the network for 70 epochs, and 0.00001 was set for the learning rate. We used a weight decay of 0.0004 and the momentum number of Adam was set to 0.9. The models were trained by ourselves on CIFAR-100 dataset since the GoogLeNet with triplet loss or margin loss had not been used in the previous works. The proposed loss function outperformed the triplet loss and margin loss on GoogLeNet models. The classification criteria are shown in Table 1.

It can be seen from Table 1 that the advantage of the proposed loss on the CIFAR-100 dataset is not as evident as that on the CIFAR-10 dataset since the NMI value and $F1$ value of our loss are only slightly higher than those of other loss functions. From the top-1 recall rate, the highest proportion of correct classification for a certain class of samples is 0.632, which was generated by the model with our proposed loss function. Though the experiment results indicate that metric methods used in margin loss and triplet loss are capable of the strong classification for fine image classification, our loss function is superior to other loss functions in general.

On the Fashion-MNIST dataset, we changed the backbone of the training model to ResNet50. We trained this model for 70 epochs, and 0.00001 was set for the learning rate. We used a weight decay of 0.0004 and the momentum number of Adam was set to 0.9. Evidently, the proposed loss function outperformed the triplet loss and margin loss. Figure 9 shows the classification performance of different loss functions on Fashion-MNIST dataset. According to Figure 9, the NMI value and $F1$ value of our loss function are higher than those of the other two loss functions. The values of β and γ in the new loss are set to 4 and 10, respectively. It means that the new loss is adopted to different datasets or different classification tasks by changing the value of the function parameter.

Furthermore, we used recall rate to verify the classification effect of different loss functions on the validation sets. The following conclusion is given in Table 2. Compared with margin loss and triplet loss, our new loss obtains higher top-8 recall rate for all architectures on different datasets. Although the top-1 recall rate obtained by our new loss is lower than other loss functions for the ResNet with depth 50, the overall trend of recall rate generated by the new loss is better than others.

5. Discussions

In this paper, we applied our loss to the CIFAR-10 dataset representing simple image classification and to the CIFAR-

Table 1: Comparison on the CIFAR-100 dataset.

Loss function	NMI	*F*1
Margin loss	0.512	0.541
Triplet loss	0.533	0.548
New loss*	**0.542**	**0.550**

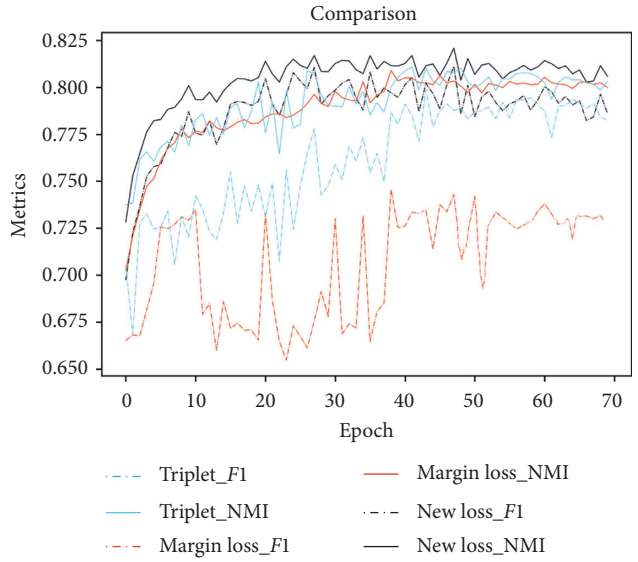

Figure 9: Comparison graph of different loss functions on Fashion-MNIST dataset.

Table 2: Comparison of recall rates with three loss functions.

Dataset	Model	Loss function	Recall@1	Recall@2	Recall@4	Recall@8
CIFAR-100	GoogLeNet	Margin loss	0.607	0.712	0.763	0.815
		Triplet loss	0.610	0.708	0.772	0.810
		New loss*	**0.632**	**0.720**	**0.779**	**0.817**
	GoogLeNet	Margin loss	0.712	0.810	0.873	0.916
		Triplet loss	0.695	0.801	0.869	0.912
		New loss*	**0.722**	**0.8291**	**0.8892**	**0.9294**
CIFAR-10	ResNet50	Margin loss	0.7807	0.8496	0.8922	0.9312
		Triplet loss	**0.8067**	**0.8661**	0.9045	0.9319
		New loss*	0.7743	0.8638	**0.918**	**0.9504**
Fashion-MNIST	ResNet50	Margin loss	0.8864	0.9273	0.9545	0.9704
		Triplet loss	**0.8909**	**0.9304**	0.959	0.9742
		New loss*	0.8749	0.9259	**0.961**	**0.9801**

100 dataset symbolizing fine image classification. The experiments proved that our loss function boasts relatively strong generalization ability in image classification and better classification performance than other loss functions in both fine and simple image classification. It is worth noting that our loss function has better classification performance than others in simple image classification, demonstrating that our loss's classification performance is not lowered when completing fine classification. In addition, our loss function reduces the training difficulty of image classification through ingeniously avoiding the complex computation of image triplet distance. Currently, our research has not considered the issue of classification imbalance; it is considered to adjust the loss functions or the network architectures in the future.

6. Conclusions

We proposed a loss function for reducing the difficulty of training. In order to solve this, we proposed an improved loss function which adds a modulation measure on the basis of the cross-entropy loss for giving less learning weights on easy samples. Extensive experiments demonstrate that the loss function we proposed outperforms the effect of margin loss or triplet loss was used for other frames on both small-scale and large-scale datasets.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

Acknowledgments

This work was supported by National Natural Science Foundation of China 61971297.

References

[1] N. Chen, J. Zhu, F. Sun, and E. P. Xing, "Large-margin predictive latent subspace learning for multiview data analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 12, pp. 2365–2378, 2012.

[2] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proceedings of the 33rd International Conference on Machine Learning*, pp. 507–516, New York, NY, USA, June 2016.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[4] W. Wan, Y. Zhong, T. Li, and J. Chen, "Rethinking feature distribution for loss functions in image classification," 2018, http://arxiv.org/abs/1803.02988.

[5] C. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2859–2867, Venice, Italy, October 2017.

[6] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition. SIMBAD 2015. Lecture Notes in Computer Science*Springer, New York, NY, USA, 2015.

[7] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," 2014, http://arxiv.org/abs/1406.4773.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, Boston, MA, USA, June 2015.

[9] S. Ren, K. He, R. Girshick et al., "Faster R-CNN: towards real-time object detection with region proposal networks," 2015, http://arxiv.org/abs/1506.01497.

[10] W. Zhu, C. Liu, W. Fan et al., "DeepLung: deep 3D dual path nets for automated pulmonary nodule detection and classification," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, March 2018.

[11] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa, "Triplet probabilistic embedding for face verification and clustering," in *Proceedings of the 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Niagara Falls, NY, USA, September 2016.

[12] B. Zhuang, G. Lin, C. Shen, and I. Reid, "Fast training of triplet-based deep binary embedding networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Technical Report, University of Toronto, Toronto, Canada, 2012.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1106–1114, 2012.

[15] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," 2017, http://arxiv.org/abs/1704.06369.

[16] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning aninvariant mapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1735–1742, New York, NY, USA, June 2006.

[17] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proceedings of the European Conference on Computer Vision*, pp. 499–515, Amsterdam, Netherlands, October 2016.

[18] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*, Springer Science & Business Media, Berlin, Germany, 2004.

[19] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F_1 measure," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 225–239, Nancy, France, September 2014.

[20] L. Danon, A. Díaz-Guilera, and A. Arenas, "The effect of size heterogeneity on community identification in complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 11, p. 11010, 2006.

[21] K. K. Sung and T. Poggio, "Learning and example selection for object and pattern detection," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1994.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

[23] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[24] K. Sohn, "Improved deep metric learning with multi-class *N*-pair loss objective," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 1857–1865, Barcelona, Spain, December 2016.

[25] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," 2017, http://arxiv.org/abs/1708.07747.