Stop-Gradient Softmax Loss for Deep Metric Learning

Lu Yang[1,2], **Peng Wang**[1,2,*], **Yanning Zhang** [1,2,*]

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2] National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, China
lu.yang@mail.nwpu.edu.cn, peng.wang@nwpu.edu.cn, ynzhang@nwpu.edu.cn

Abstract

Deep metric learning aims to learn a feature space that models the similarity between images, and feature normalization is a critical step for boosting performance. However directly optimizing $L2$-normalized softmax loss cause the network to fail to converge. Therefore some SOTA approaches appends a scale layer after the inner product to relieve the convergence problem, but it incurs a new problem that it's difficult to learn the best scaling parameters. In this letter, we look into the characteristic of softmax-based approaches and propose a novel learning objective function Stop-Gradient Softmax Loss (SGSL) to solve the convergence problem in softmax-based deep metric learning with L2-normalization. In addition, we found a useful trick named Remove the last BN-ReLU (RBR). It removes the last BN-ReLU in the backbone to reduce the learning burden of the model. Experimental results on four fine-grained image retrieval benchmarks show that our proposed approach outperforms most existing approaches, i.e., our approach achieves 75.9% on CUB-200-2011, 94.7% on CARS196 and 83.1% on SOP which outperforms other approaches at least 1.7%, 2.9% and 1.7% on Recall@1.

Introduction

Deep metric learning (DML) aims to learn a similarity metric, which can map samples to a high-dimensional space. In the high-dimensional space, the samples of the same instance are closer, while the samples of different instances are farther away. Typical deep metric learning applications include image retrieval, person re-identification, etc. Popular methods of deep metric learning include pairwise based methods and softmax based methods. Pairwise based methods focused on finding efficient ways to improve sample weighting strategies over the existing pairwise losses, such as contrastive loss and triplet loss. Pairwise based methods directly affect the distance between point pairs in the embedded space, which is strongly related to the goal of DML. Softmax based methods may seem unrelated to DML as it does not explicitly involve pairwise distances at the surface.

Some methods such as (Movshovitz-Attias et al. 2017; Wang et al. 2017a; Zhai and Wu 2019; Wang et al. 2018) only use softmax loss to train the model which can achieve good performance as well. In contrast to pairwise-based methods, the softmax-based method can be viewed as approximating each class using a proxy (Movshovitz-Attias et al. 2017), and uses all proxies to provide global context for each training iteration. Boudiaf et al. (Boudiaf et al. 2020) proves that optimizing the softmax-based method corresponds to an approximate bound-optimizer of an underlying pairwise loss, showing that minimizing the softmax loss is equivalent to maximizing a discriminative view of the mutual information between the features and labels. In practice, the inner product (last fully connection layer) without L_2-normalization is the most widely used similarity measure when training the softmax-based DML model, but the features are often L_2-normalized in the testing phase (Boudiaf et al. 2020; He et al. 2020), that means the distance metric used during training is different from that used in the testing phase. In order to make up for this gap, a simple method is

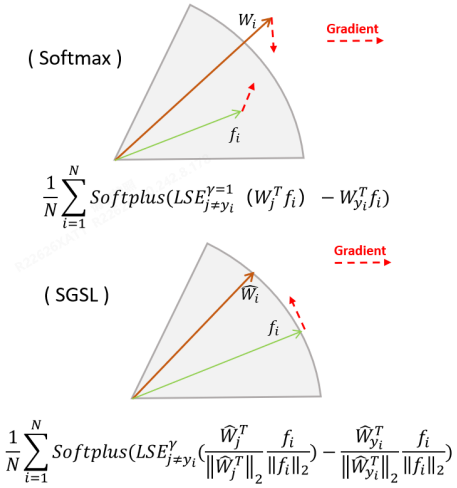

Figure 1: Illustration of Softmax and our proposed SGSL. SGSL and softmax share parameters, but there are three differences: 1) The value of γ. 2) The feature in SGSL is L2-normlized. 3) We do not allow gradient update through W_j, which is identified as $\hat{W}_j$. The derivation process and details can be found in Section Method.

*Corresponding author

to use L_2-normalization during training directly. However, after L_2-normalization, the network fails to converge easily. The softmax loss only decreases a little and then converges to a very big value within a few thousands of iterations. After that the loss does not decrease no matter how many iterations we train and how small the learning rate is. Wang et al. (Wang et al. 2017a) claim that this is mainly because the range of inner product output is only $[-1, 1]$ after $L2$-normalization, and it may prevent the probability getting close to 1 even when the samples are well-separated. In order to relieve this convergence problem, Wang et al. (Wang et al. 2017a) appends a scale layer after the inner product. The scale layer have a learnable parameter to scale the inner product output to a bigger value instead of 1, then the softmax loss can continue to decrease. However, this method can not guarantee that the network can learn the best scaling parameter. In this paper, we propose a new softmax based metric loss named Stop-Gradient Softmax Loss (SGSL), it used together with the original softmax. As Figure 1 shows, it shares parameters with the original softmax and has almost the same form, with only three differences: different γ, L2-normalized feature and stop gradient for W_j. Since the features used in SGSL are L2-normalized, the distance metric in the training phase is consistent with that in the test phase. SGSL and the softmax share parameters, so that the network can get a good proxy (class center). At the same time, the gradient of the class center is stopped in SGSL, but the sample feature does not stop the gradient, thus forcing the sample feature to approach the class center on the high spherical surface. To summarize, the contribution of our work is three-fold:

- We propose a novel and efficient Stop-Gradient Softmax Loss to solve the convergence problem in softmax-based DML with L_2-normalization. The proposed SGSL does not need for complex sample-mining in deep metric learning.
- In addition, we propose a useful trick named Remove the last BN-ReLU (RBR) for ResNet, it can reduce the learning burden of the model and improve performance.
- Experiments on CUB-200-2011 (Welinder et al. 2010), CARS-196 (Krause et al. 2013), Stanford Online Products (SOP) (Oh Song et al. 2016a), and In-shop Clothes Retrieval (Liu et al. 2016) show that our method achieves SOTA results than the current pairwise based and softmax based DML approaches.

Related Works

Deep Metric Learning

Deep Metric Learning learns a set of nonlinear transformation (He et al. 2016) for mapping the raw data points into other feature space with higher discrimination power. The deep architecture (Simonyan and Zisserman 2014) is exploited for better comparison and matching ability in the feature space. It combines metric learning and feature learning together as a joint framework. Image retrieval is one of the most common applications of deep metric learning. One kind of losses used in deep metric learning is based on classification, which is called softmax-based losses. And another is based on sample pairs, which is called pairwise-based losses.

Pairwise-based Losses

Pairwise-based losses use positive or negative sample pair to supervise the model for learning.

Center Loss (Wen et al. 2016)'s target is to make all samples as close to the center of their class as possible to increase the intra-class compactness, and it increase the distance between different categories and reduce the distance between the same class:

$$\mathcal{L}_{center} = \frac{1}{2}\mathbb{E}_{\mathbf{x}}\left[\|\boldsymbol{x} - \boldsymbol{c}_y\|_2^2\right], \tag{1}$$

where $\boldsymbol{c}_y$ is the learned center for class y. The definition of other symbols is same as above. Working with the ID cross-entropy loss, center loss can obtain excellent intra-class compactness and inter-class separability on the training set.

Triplet Loss (Schroff, Kalenichenko, and Philbin 2015) applies to a triplet of samples called anchor point, positive point and negative point. It aims to pull an anchor point closer to the positive point (same identity) than to the negative point (different identity) by a fixed margin.

$$\mathcal{L}_{tri} = \mathbb{E}_{\{\mathbf{x}_a,\mathbf{x}_p,\mathbf{x}_n\}}\left[D(\mathbf{x}_a,\mathbf{x}_p) - D(\mathbf{x}_a,\mathbf{x}_n) + m\right]_+, \tag{2}$$

where $D(\cdot)$ denotes euclidean distance, m is a fixed margin and $[\cdot]_+$ is the hinge function.

Soft Margin Triplet Loss (Hermans, Beyer, and Leibe 2017) replace the hinge function in triplet (Schroff, Kalenichenko, and Philbin 2015) by a smooth approximation using the softplus function $softpuls(\cdot) = ln(1 + exp(\cdot))$. The softplus function has similar behavior to the hinge, but it decays exponentially instead of having a hard cut-off.

Lifted Structure Loss (Oh Song et al. 2016b) tries to pull one positive pair as close as possible and pushes all negative samples farther than a margin.

Ranked List Loss (Wang et al. 2019b) proposes to build a set-based similarity structure by exploiting all instances in the gallery. Different from the above methods, which aim to pull positive pairs as close as possible in the embedding space, the Ranked List Loss only needs to pull positive examples closer than a predefined threshold (boundary).

Pairwise-based losses has a common problem, in which pairwise losses require careful sample mining and weighting strategies to obtain the most informative pairs, otherwise, the performance will be greatly affected.

Softmax-based Losses

Softmax-based losses modify the original softmax to make the model learn distinguishing features, without the need for complex sample-mining and optimization schemes. The original softmax is,

$$\mathcal{L}_{softmax} = \frac{1}{N}\sum_{i=1}^{N} -log\frac{e^{W_{y_i}^T f_i}}{\sum_{j=1}^{c} e^{W_j^T f_i}} \tag{3}$$

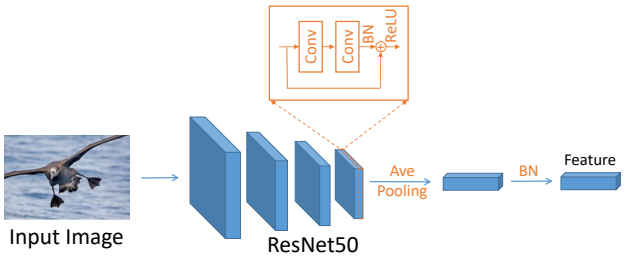

Figure 2: A common network architecture for image retrieval. Many approaches (Luo et al. 2019; Boudiaf et al. 2020; He et al. 2020) add batch normalization (without scaling and bias) on top of the backbone, as it can smoothen/normalize the feature distribution and enhance the intra-class compactness.

where N is the number of samples in the batch, c is the number of classes in training set, f_i is the ith sample's feature and y_i is the ith sample's label. W_j is the jth column of the last fully connection layer, which is corresponding to the jth class.

Ring Loss applies soft normalization, where it gradually learns to constrain the norm to the scaled unit circle while preserving convexity leading to more robust features.

$$\mathcal{L}_{Ring} = \frac{\lambda}{2m}\sum_{i=1}^{m}(\|\mathcal{F}(x_i)\|_2 - R)^2 \tag{4}$$

where $\mathcal{F}(x_i)$ is the deep network feature for the sample x_i. Here, R is the target norm value and λ is the loss weight enforcing a trade-off between the primary loss function. Boudiaf et al. (Boudiaf et al. 2020) proving that optimizing the softmax-based method corresponds to an approximate bound-optimizer of an underlying pairwise loss.

But these methods can not solve the convergence problem in L2-normalized softmax well.

Method

In this section, we will first describe the definition of the proposed loss function and discuss about the intuition and interpretation of the loss function. Then we describe and analyse the "remove last BN-ReLU" trick when batch normalization is added on top of the model.

Stop-Gradient Softmax Loss (SGSL)

When training the classification network for metric learning, many approaches (Wang et al. 2017a; Zhai and Wu 2019; Wang et al. 2018) remove the bias term in the last fully connection layer, and we follow this setting. To better understand our approach, we give a brief review of the original softmax and its variants. The original softmax loss (without bias) can be written as

$$\begin{aligned}\mathcal{L}_{softmax} &= \frac{1}{N}\sum_{i=1}^{N} -log\frac{e^{W_{y_i}^T f_i}}{\sum_{j=1}^{c} e^{W_j^T f_i}} \\ &= \frac{1}{N}\sum_{i=1}^{N} log(1 + e^{log\frac{\sum_{j=1,j\neq y_i}^{c} e^{W_j^T f_i}}{e^{W_{y_i}^T f_i}}}) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{Softplus}((log\sum_{j=1,j\neq y_i}^{c} e^{W_j^T f_i}) - W_{y_i}^T f_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{Softplus}(\mathbf{LSE}_{j\neq y_i}^{\gamma=1}(W_j^T f_i) - W_{y_i}^T f_i)\end{aligned} \tag{5}$$

where N is the number of samples in the batch, c is the number of classes in training set, f_i is the ith sample's feature and y_i is the ith sample's label. W_j is the jth column of the last fully connection layer, which is corresponding to the jth class. $Softplus(x) = log(1 + e^x)$ and $LSE^{\gamma}(x_1 \cdots x_n) = \frac{1}{\gamma}log(\sum_{j=1}^{c} e^{\gamma x_j})$.

The formula of our proposed Stop-Gradient Softmax Loss (SGSL) is similar to the standard softmax, but there are three differences: 1) The γ in SGSL is not fixed to 1, but a larger value. γ can be regarded as a scale parameter to control the "temperature" of the loss. But unlike traditional temperature scaling (Wu, Efros, and Yu 2018), our γ only added when W_j and f_i come from different classes; 2) Both W_j and f_i are L2-normalized; 3) We do not allow gradient update through W_j. Our proposed SGSL is defined as

$$\mathcal{L}_{SGSL} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{Softplus}(\mathbf{LSE}_{j\neq y_i}^{\gamma}(\frac{\hat{W}_j^T}{\left\|\hat{W}_j^T\right\|_2}\frac{f_i}{\|f_i\|_2}) - \frac{\hat{W}_{y_i}^T}{\left\|\hat{W}_{y_i}^T\right\|_2}\frac{f_i}{\|f_i\|_2}) \tag{6}$$

where $\frac{\cdot}{\|\cdot\|_2}$ stands for L2-normalization and $\hat{W}_j$ means we do not allow gradient update through W_j, γ is a pre-defined scalar (e.g., 30). Other symbols have the same meaning as Equation 5. In the experiment, we use the original softmax and SGSL together, then the total loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{softmax} + \mathcal{L}_{SGSL} \tag{7}$$

Why SGSL Works

We know that $Softplus(x) = log(1 + e^x)$ (Chapados et al. 2001) is a convex and monotone increasing function, and can be considered as a smooth version of the positive part function $max(0, x)$. The so-called Log Sum of Exponentials $LSE^{\gamma}(x_1 \cdots x_n) = \frac{1}{\gamma}log(\sum_{i=1}^{n} e^{\gamma x_i})$ is a functional form commonly encountered in dynamic discrete choice models, it can be considered as a smooth version of selecting the largest one in a set of data. And the larger γ is, the smaller the error is. However, if the error is too small, the information except the biggest one may be lost in the optimization process. So it is important to choose a suitable γ, and in the experiment we set an empiric value $\gamma = 30$ in Equation 6 as default. Based on the above, Equation 6 can be approximately expressed as

$$\mathcal{L}_{SGSL} \approx \frac{1}{N}\sum_{i=1}^{N}[\mathbf{max}_{j\neq y_i}(\frac{\hat{W}_j^T}{\left\|\hat{W}_j^T\right\|_2}\frac{f_i}{\|f_i\|_2}) - \frac{\hat{W}_{y_i}^T}{\left\|\hat{W}_{y_i}^T\right\|_2}\frac{f_i}{\|f_i\|_2}]_+ \tag{8}$$

where the $[\cdot]_+$ denotes $max([\cdot], 0)$, $\hat{W}_j$ means we do not allow gradient update through W_j, and the cosine similarity $\frac{u}{\|u\|_2}\frac{v}{\|v\|_2}$ is normalized version of inner-product of two vectors, it used to measure the similarities between features which is independent of magnitude, and it can be equivalently L2-normalized euclidean distance.

As we can see from Equation 8, the target of $\mathcal{L}_{SGSL}$ is to make cosine similarity between f_i and W_{y_i} greater than the maximum cosine similarity between f_i and $W_{j\neq y_i}$. In other words, $\mathcal{L}_{SGSL}$ requires that the features learned by the network should be closer (L2-normalized euclidean distance) to the proxy of its class and farther away from the proxies of other classes, which is obviously related to the goal of DML. Note that $\mathcal{L}_{SGSL}$ does not allow gradient update through W, therefore there is less convergence problem in SGSL.

Remove the Last BN-ReLU (RBR)

In deep metric learning, ResNet50 without last fully connection layer is often used as the backbone. As Figure 2 shows, many approaches (Luo et al. 2019; Boudiaf et al. 2020; He et al. 2020) add batch normalization (without scaling and bias) on top of the backbone, as it can smoothen/normalize the feature distribution and enhance the intra-class compactness. However, it makes the last three layers in backbone are BN-ReLU-BN which will increase the learning burden of the model. Continuous BN-ReLU modules added to the output feature do not bring any new information, but may drop some useful information for metric learning. The experiment result in Figure 4 further proves that the continuous BN-ReLU modules added to the output feature are not conducive to metric learning, so we remove the last BN-ReLU in the backbone.

Experiments

We conduct extensive experiments on four public image retrieval benchmarks, $i.e.$, CUB-200-2011 (Welinder et al. 2010), CARS-196 (Krause et al. 2013), Stanford Online Products (SOP) (Oh Song et al. 2016a), and In-shop Clothes Retrieval (Liu et al. 2016). The architecture as Figure 2 shows. We follow the same evaluation protocol commonly used in traditional image retrieval benchmarks with the standard train/test split and compare our proposed approach to state-of-the-art deep metric learning approaches. For a fair comparison, we only demonstrate the performance of the approaches trained with ResNet-50.

Datasets

CUB-200-2011 has 200 classes with $11,788$ images. The first 100 classes (5864 images) for training and the rest of the classes (5,924 images) for testing.

CARS-196 has 198 classes with $16,185$ images. The first 98 classes for training ($8,054$ images) and the other 98 classes ($8,131$ images) for testing.

Stanford Online Products has $22,634$ classes with $120,053$ images. The first $11,318$ classes ($59,551$ images) for training and the other $11,316$ classes ($60,502$ images) for testing.

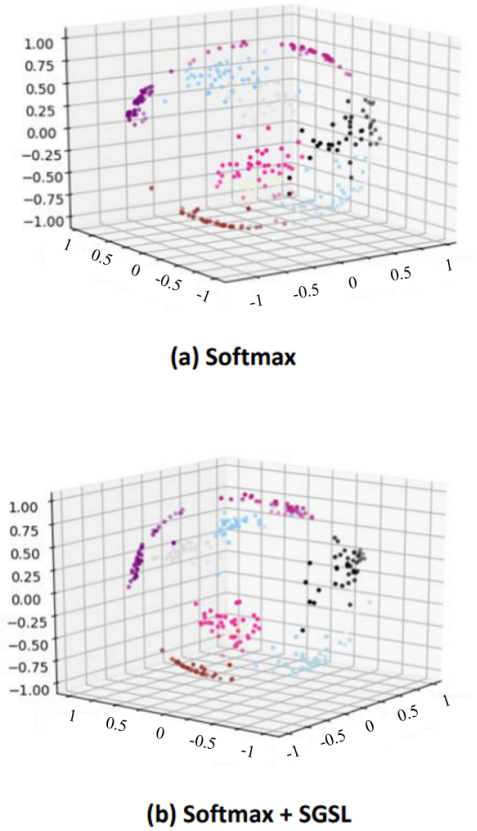

Figure 3: Feature distribution visualization of "softmax" and "softmax + SGSL" on fasion MNIST. We can see that the features obtained by our approach (softmax + SGSL) have a more compact intra-class distribution. Generally speaking, more compact intra-class distribution is useful for DML. Best viewed in color.

In-shop Clothes is a large-scale clothes dataset with comprehensive annotations. It has 50 fine-grained categories and $1,000$ attributes, and contains over $800,000$ images, which are richly annotated with massive attributes, clothing landmarks, and correspondence of images taken under different scenarios including store, street snapshot, and consumer.

Implementation Details

Our experiments were executed using PyTorch on GTX 2080Ti GPU. We use different numbers of GPUs for training according to the size of the data set, for example, SOP used 4 GPU for training, and others used 2. All the experiments use ResNet50 as the backbone which pre-trained on ImageNet, and we replace the global average pooling by generalized mean pooling (Filip, Giorgos, and Ondrej 2017). As many DML approaches (Luo et al. 2019; Boudiaf et al. 2020; He et al. 2020), we add Batch Normalization (BN), without scaling and bias, on top of the backbone. Like most meth-

		CUB-200-2011				CARS196				Stanford Online Products		
Method	**#dims**	**R@1**	**R@2**	**R@4**	**R@8**	**R@1**	**R@2**	**R@4**	**R@8**	**R@1**	**R@10**	**R@100**
Deep Spectral (Law, Urtasun, and Zemel 2017) *ICML17*	512	53.2	66.1	76.7	85.2	73.1	82.2	89.0	93.0	67.6	83.7	93.3
Angular Loss (Wang et al. 2017b) *ICCV17*	512	54.7	66.3	76	83.9	71.4	81.4	87.5	92.1	70.9	85.0	93.5
Hierarchical triplet (Ge 2018) *ECCV18*	512	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7	74.8	88.3	94.8
ABE (Kim et al. 2018) *ECCV18*	512	60.6	71.5	79.8	87.4	85.2	90.5	94.0	96.1	76.3	88.4	94.8
Normalized Softmax (Zhai and Wu 2019) *BMVC19*	512	61.3	73.9	83.5	90.0	84.2	90.4	94.4	96.9	78.2	90.6	96.2
RLL-H (Wang et al. 2019b) *CVPR19*	512	57.4	69.7	79.2	86.9	74.0	83.6	90.1	94.1	76.1	89.1	95.4
Multi-similarity (Wang et al. 2019a) *CVPR19*	512	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0
Relational Knowledge (Park et al. 2019) *CVPR19*	512	61.4	73.0	81.9	89.0	82.3	89.8	94.2	96.6	75.1	88.3	95.2
SoftTriple Loss (Qian et al. 2019) *ICCV19*	512	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9	78.3	90.3	95.9
HORDE (Jacob et al. 2019) *ICCV19*	512	66.3	76.7	84.7	90.6	83.9	90.3	94.1	96.3	80.1	91.3	96.2
Easy triplet mining (Xuan 2020) *WACV20*	512	64.9	75.3	83.5	-	82.7	89.3	93.0	-	78.3	90.7	96.3
Proxy NCA++ (Teh, DeVries, and Taylor 2020) *ECCV20*	512	69.0	79.8	87.3	92.7	86.5	92.5	95.7	97.7	80.7	92.0	96.7
Proxy Anchor (Kim et al. 2020) *CVPR20*	512	68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3	79.1	90.8	96.2
Proxy Few (Zhu et al. 2020) *NeurIPS20*	512	66.6	77.6	86.4	-	85.5	91.8	95.3	-	78.0	90.6	96.2
EFC(12,48) (Li et al. 2022) *IET CV21*	576	69.8	79.5	86.5	91.8	91.8	95.1	97.0	98.2	-	-	-
IBC (Seidenschwarz, Elezi, and Leal-Taixé 2021) *ICML21*	512	70.3	80.3	87.6	92.7	88.1	93.3	96.2	98.2	*81.4*	91.3	95.9
HIST (Lim et al. 2022) *CVPR22*	512	71.4	81.1	88.1	-	89.6	93.9	96.4	-	*81.4*	92.0	96.7
SGSL (Ours)	512	72.0	81.1	88.3	93.1	*94.1*	*96.7*	*98.0*	*99.0*	*81.4*	91.8	96.2
Normalized Softmax (Zhai and Wu 2019) *BMVC19*	2048	65.3	76.7	85.4	91.8	89.3	94.1	96.4	98.0	79.5	91.5	96.7
Cross-Entropy-cos (Boudiaf et al. 2020) *ECCV20*	2048	69.2	79.2	86.9	91.6	89.3	93.9	96.6	98.4	81.1	91.7	96.3
Proxy NCA++ (Teh, DeVries, and Taylor 2020) *ECCV20*	2048	72.2	82.0	*89.2*	*93.5*	90.1	94.5	97.0	98.4	*81.4*	*92.4*	*96.9*
KAE-Net (Moskvyak et al. 2021) *WACV21*	2048	*74.2*	*83.3*	89.1	93.2	91.1	94.9	96.9	98.1	-	-	-
SGSL (Ours)	2048	**75.9**	**85.0**	**90.7**	**94.5**	**94.7**	**97.2**	**98.3**	**99.1**	**83.1**	**93.0**	**97.0**

Table 1: Retrieval performance on *CUB-200-2011*, *CARS196* and *Stanford Online Products* datasets. Bold and Italic fonts represent the best and second best performance respectively.

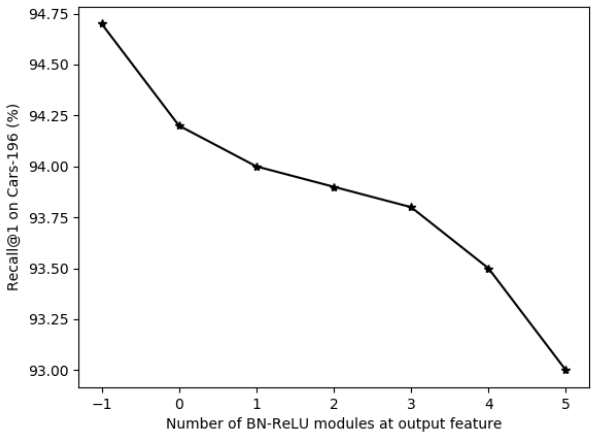

Figure 4: Performance of different number of continuous BN-ReLU modules on Cars-196. The BN-ReLU modules are added to the output feature. The -1 on the X-axis means that the last BN-ReLU in the backbone is removed.

ods (Ge 2018; Movshovitz-Attias et al. 2017; Wang et al. 2017b, 2019a; Xuan, Souvenir, and Pless 2018; Xuan 2020; Yuan, Yang, and Zhang 2017; Zhai and Wu 2019), we use the L2-normalized euclidean distances to compute the recall for the evaluation. All the input images were resized to 256×256 and croped to 224×224 with a batch size of 64 (4 images/ID and 16 IDs), and we use fp16 to improve the GPU memory utilization. The model is trained 100 epochs and we set the learning rate of parameter using cosine annealing schedule. We set $\gamma = 30$ as default. To build a robust model that can generalize well, we use label smoothing for $\mathcal{L}_{softmax}$. For the stability of training, SGSL starts to join the training only when the value of softmax is smaller than 3. Other settings are the same as (Boudiaf et al. 2020).

Comparison with other SOTAs

We evaluate our approach in comparison with state-of-the-art approaches on several image retrieval benchmarks. For a fair comparison, we only demonstrate the performance with embedding of 512 dimension and $2,048$ dimension. The comparison between our approach and other competitors on four public image retrieval datasets is presented in Table 1 and Table 2. Overall, our approach outperforms all the compared methods. On the datasets CUB-200-2011, CARS196 and SOP, the Recall@1 with $2,048$ dimension of our approach are much higher than previous SOTAs by $1.7\%, 2.9\%$ and 1.7% respectively. And for InShop, our proposed SGSL can also achieve state-of-the-art performance.

Comparison with other Losses

We conducted experiments with different losses on CUB-200-2011, CARS196 and In-shop Clothes, i.e., L2-normlization Loss, Triplet Loss (Schroff, Kalenichenko, and Philbin 2015), SoftMargin Loss (Hermans, Beyer, and Leibe 2017), Ring Loss (Zheng, Pal, and Savvides 2018). The results are shown in Table 3. Compared with these common loss functions, SGSL shows excellent performance both on 512 dimensions and $2,048$ dimensions. Especially on the CUB-200-2011, SGSL exceeds the other losses by more than 4.4% on Recal@1. In addition, we note that the per-

Method	#dims	R@1	R@10	R@20	R@40
A-BIER (Opitz et al. 2018) *PAMI20*	512	83.1	95.1	96.9	97.8
ABE (Kim et al. 2018) *ECCV18*	512	87.3	96.7	97.9	98.5
Normalized Softmax (Zhai and Wu 2019) *BMVC19*	512	88.6	97.5	98.4	98.8
Multi-similarity (Wang et al. 2019a) *CVPR19*	512	89.7	97.9	98.5	99.1
Learning to Rank (Cakir et al. 2019)	512	90.9	97.7	98.5	98.9
HORDE (Jacob et al. 2019) *ICCV19*	512	90.4	97.8	98.4	98.9
Proxy NCA++ (Teh, DeVries, and Taylor 2020) *ECCV20*	512	90.4	98.1	98.8	99.2
Proxy Anchor (Kim et al. 2020) *CVPR20*	512	91.5	98.1	98.8	99.1
IBC (Seidenschwarz, Elezi, and Leal-Taixé 2021) *ICML21*	512	92.8	*98.5*	**99.1**	99.2
SGSL (Ours)	512	**93.5**	**98.6**	**99.1**	*99.3*
Normalized Softmax (Zhai and Wu 2019) *BMVC19*	2048	89.4	97.8	98.7	99.0
Cross-Entropy-*cos* (Boudiaf et al. 2020) *ECCV20*	2048	90.6	98.0	98.6	99.1
Proxy NCA++ (Teh, DeVries, and Taylor 2020) *ECCV20*	2048	90.9	98.2	98.9	**99.4**
SGSL (Ours)	2048	*93.0*	**98.6**	**99.1**	*99.3*

Table 2: Retrieval performance on *In Shop Clothes*. Bold and Italic fonts represent the best and second best performance respectively.

		CUB-200-2011				CARS196				In-shop Clothes		
Method	**#dims**	**R@1**	**R@2**	**R@4**	**R@8**	**R@1**	**R@2**	**R@4**	**R@8**	**R@1**	**R@10**	**R@20**
Softmax (Boudiaf et al. 2020)	512	67.9	78.5	86.0	91.8	92.1	95.7	97.5	98.7	92.7	98.5	99.0
Softmax + L2-Norm	512	59.3	70.7	80.3	87.2	88.0	92.4	94.9	96.9	92.4	98.4	98.9
Softmax + Triplet	512	66.9	77.2	84.9	90.8	91.5	95.1	97.1	98.4	92.7	98.5	99.0
Softmax + SoftMargin	512	65.0	75.2	83.7	89.8	91.5	94.7	96.9	98.2	92.3	98.5	99.0
Softmax + Ring Loss	512	69.7	79.6	87.1	92.0	92.2	95.6	97.5	98.7	92.7	98.5	99.0
Softmax + **SGSL (Ours)**	512	72.0	81.1	88.3	93.1	94.1	96.7	98.0	99.0	**93.5**	**98.6**	**99.1**
Softmax (Boudiaf et al. 2020)	2048	69.2	79.2	86.9	91.6	89.3	93.9	96.6	98.4	90.6	98.0	98.6
Softmax + L2-Norm	2048	61.3	72.5	81.3	87.7	89.7	93.2	95.8	97.3	92.2	98.2	98.9
Softmax + Triplet	2048	68.0	78.5	86.3	91.7	92.5	95.7	97.4	98.5	92.2	98.3	98.9
Softmax + SoftMargin	2048	66.1	76.7	84.4	90.2	91.6	95.3	97.1	98.2	91.8	98.3	98.9
Softmax + Ring Loss	2048	71.5	81.2	88.3	92.8	92.2	95.6	97.5	98.6	91.2	98.1	98.7
Softmax + **SGSL (Ours)**	2048	**75.9**	**85.0**	**90.7**	**94.5**	**94.7**	**97.2**	**98.3**	**99.1**	93.0	**98.6**	99.1

Table 3: Comparison with some metric learning losses on *CUB-200-2011*, *CARS196* and *In-shop Clothes* datasets. The baseline is ResNet50 with softmax (cross entropy) loss only, the detail settings are same as (Boudiaf et al. 2020). Bold fonts represent the best performance (in %).

Base Method	R@1	R@2	R@4	R@8
Softmax (Boudiaf et al. 2020)	91.9	95.6	97.4	98.7
Softmax + RBR	92.4	96.0	97.8	98.9
Softmax + RBR + SGSL($\gamma = 1$)	94.1	96.9	98.1	99.0
Softmax + RBR + SGSL($\gamma = 10$)	94.6	97.1	98.3	**99.1**
Softmax + RBR + SGSL($\gamma = 30$)	**94.7**	**97.2**	98.3	**99.1**
Softmax + RBR + SGSL($\gamma = 50$)	94.5	97.0	98.3	**99.1**
Softmax + RBR + SGSL($\gamma = 70$)	94.4	**97.2**	**98.4**	99.0

Table 4: We analyzed the performance of SGSL and RBR on Cars196 by the ablation study experiments. SGSL is short for Stop-Gradient Softmax Loss and RBR is short for Remove the last BN-ReLU. The embedding dimension is 2048.

formance of L2-Norm, Triplet and SoftMargin on the CUB dataset is even lower than that of baseline, a possible reason is that the baseline Softmax uses the setting of (Boudiaf et al. 2020), which has carefully designed the hyper-parametric for each dataset, and directly using above losses may destroy this design. While our proposed SGSL achieved a performance improvement of 6.7% on the CUB-200-2011 dataset, which indicates that SGSL has good compatibility.

Feature Distribution Visualization

To better understand the effect of SGSL, we conduct an experiment on Fashion MNIST (Xiao, Rasul, and Vollgraf 2017) to visualize the feature distributions trained by original softmax and SGSL. We use a five-layer CNN model, and set the output number of the last hidden layer to 3, which allows us to plot the features on 3-D surface for visualization. The results are shown in Figure 3 in which different colors are used to denote samples from different classes. It can be seen that SGSL can significantly shrink the intra-class variance which is beneficial to metric learning.

Class Separability Criterion

We use the class separability criterion (CSC) (Sergios and Konstantinos 2003) to evaluate the effect of our proposed SGSL. It takes large values when samples in the embedding space are well clustered around their mean, within each class, and the clusters of the different classes are well separated. The CSC is calculated by between-ID scatter matrix

Dataset	Methos	$trace\{S_b\}$	$trace\{S_w\}$	CSC
CUB	Softmax	492.3	14.9	33.0
	Softmax+SGSL	456.5	12.7	**36.0**
CARS-196	Softmax	437.1	11.1	39.4
	Softmax+SGSL	375.9	9.1	**41.3**
SOP	Softmax	841.2	111.2	7.6
	Softmax+SGSL	772.1	86.7	**8.9**
In-shop	Softmax	450.2	33.6	13.4
	Softmax+SGSL	440.7	31.0	**14.2**

Table 5: The CSC comparison among different losses, the larger value of CSC, the better performance of the loss. The default dimension is $2,048$.

(S_b) and within-ID scatter matrix (S_w):

$$
\begin{aligned}
S_b &= \sum_{i=1}^{M} Prob_i(\mu_i - \mu_0)(\mu_i - \mu_0)^T, \\
S_w &= \sum_{i=1}^{M} Prob_i E[(x_i - \mu_i)(x_i - \mu_i)^T], \\
CSC &= \frac{trace\{S_b\}}{trace\{S_w\}},
\end{aligned}
\tag{9}
$$

where M is the number of classes; $Prob_i$ is the probability of class i; μ_i is the mean vector of class i, μ_0 is the global mean vector; x_i is the whole samples in class i. In Tabel 5, we can find that our proposed SGSL gets higher CSC than corresponding Softmax. Better class separability leads to better retrieval performance.

Ablation Study

The Impact of SGSL and RBR

In Table 4, we analyze the effect of applying SGSL and RBR on Cars196 with 2048 dimension. The experimental results show that both SGSL and RBR can improve the performance, and SGSL plays a greater role. γ is a pre-defined scalar in Equation 6. From $3th$ row to $7th$ row in Table 4, we can see that larger γ is beneficial to the model, and when γ is greater than 10, the performance tends to be saturated. This is because $LSE^{\gamma}(\cdot)$ in Equation 6 is used to select the most similar negative proxy with cosine similarity and the larger γ is, the smaller the selection error is.

The Impact of Continuous BN-ReLU

In this section, we conduct experiments to investigate the impact of different number of continuous BN-ReLU modules. ResNet is the most commonly used backbone. As Figure 2 shows, many approaches (Luo et al. 2019; Boudiaf et al. 2020; He et al. 2020) add batch normalization (without scaling and bias) on top of the backbone, as it can smoothen/normalize the feature distribution and enhance the intra-class compactness. Therefore, when extracting features, the last three layers are BN, ReLU and BN. In this sub-section, we added different number of continuous BN-ReLU modules before the last BN layer which on top of the backbone. Figure 4 shows that continuous BN-ReLU modules does not bring any performance improvement, but only reduce the performance. The possible reason is that the continuous BN-ReLU may lose some information, thus affecting the retrieval performance. The best performance is achieved by removing the last BN-ReLU in the backbone, and we adopted this trick in our approach.

Impact of Loss Weight

Some studies like (Kendall, Gal, and Cipolla 2018; Zheng et al. 2019) show that multi task learning has the ability to achieve advanced performance by extracting appropriate shared information between tasks. When multiple losses work together, the weight of the loss is usually important. We conducted experiments on CARS196 to evaluate the effect of loss weight of SGSL. We change the loss weight β of SGSL from 0.1 to 10.0 in equation 10, and the results are shown in table 6. The experimental results show that $\beta = 1.0$ is a better choice.

$$
L_{total} = L_{softmax} + \beta L_{SGSL} \tag{10}
$$

loss weight	R@1	R@2	R@4	R@8
$\beta = 0.1$	93.2	96.4	98.0	98.8
$\beta = 0.5$	94.1	96.8	98.1	98.9
$\beta = 1.0$	**94.7**	**97.2**	**98.3**	**99.1**
$\beta = 2.0$	93.5	96.7	97.9	98.9
$\beta = 10.0$	93.0	96.2	97.8	98.9

Table 6: Effect of weighting factor β of SGSL when working together with softmax loss(in %) on CARS196. The default dimension is $2,048$.

Conclusion

In this work, we propose a Stop-Gradient Softmax Loss (SGSL) and a trick named Remove the last BN-ReLU (RBR) for the task of deep metric learning. Without need for complex sample-mining, SGSL works with original softmax together. Standard softmax performs traditional learning and optimization for good class separability, while SGSL performs distance metric learning based on L_2-normalization. And we theoretically analyze that the target of SGSL is to make the L2-normalized distance between the anchor and its positive proxy smaller than that between anchor and its negative proxy. In addition, we remove the last BN-ReLU in backbone to lighten the learning burden of the model. The extensive experimental results on four public image retrieval benchmarks show clear advantages over current state-of-the-art approaches.

Acknowledgments

This work was supported by National Key RD Program of China (No. 2020AAA0106900), the National Natural Science Foundation of China (No. U19B2037, No. 61876152), Shaanxi Provincial Key RD Program (No. 2021KWZ-03), and Natural Science Basic Research Program of Shaanxi (No. 2021JCW-03).

References

Boudiaf, M.; Rony, J.; Ziko, I. M.; Granger, E.; Pedersoli, M.; Piantanida, P.; and Ayed, I. B. 2020. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *European Conference on Computer Vision*, 548–564. Springer.

Cakir, F.; He, K.; Xia, X.; Kulis, B.; and Sclaroff, S. 2019. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1861–1870.

Chapados, N.; Bengio, Y.; Vincent, P.; Ghosn, J.; and Meng, L. 2001. Estimating Car Insurance Premia: a Case Study in High-Dimensional Data Inference. *Advances in Neural Information Processing Systems*, 1369–1376.

Filip, R.; Giorgos, T.; and Ondrej, C. 2017. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 1655–1668.

Ge, W. 2018. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 269–285.

He, K.; Zhang, X.; Ren, S.; and Jian, S. 2016. Identity Mappings in Deep Residual Networks. In *European Conference on Computer Vision*.

He, L.; Liao, X.; Liu, W.; Liu, X.; Cheng, P.; and Mei, T. 2020. FastReID: A Pytorch Toolbox for General Instance Re-identification. *arXiv preprint arXiv:2006.02631*.

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

Jacob, P.; Picard, D.; Histace, A.; and Klein, E. 2019. Metric learning with horde: High-order regularizer for deep embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6539–6548.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Kim, S.; Kim, D.; Cho, M.; and Kwak, S. 2020. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kim, W.; Goyal, B.; Chawla, K.; Lee, J.; and Kwon, K. 2018. Attention-based ensemble for deep metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 736–751.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Law, M. T.; Urtasun, R.; and Zemel, R. S. 2017. Deep spectral clustering learning. In *International conference on machine learning*, 1985–1994. PMLR.

Li, S.; Guo, Y.; Ren, H.; Wang, Z.; Ren, K.; Liu, C.; Lin, H.; and Shi, J. 2022. FCNet: A feature context network based on ensemble framework for image retrieval. *IET Comput. Vis.*, 16(4): 295–306.

Lim, J.; Yun, S.; Park, S.; and Choi, J. Y. 2022. Hypergraph-Induced Semantic Tuplet Loss for Deep Metric Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 212–222.

Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; and Tang, X. 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1096–1104.

Luo, H.; Jiang, W.; Gu, Y.; Liu, F.; Liao, X.; Lai, S.; and Gu, J. 2019. A strong baseline and batch normalization neck for deep person re-identification. *TMM*.

Moskvyak, O.; Maire, F.; Dayoub, F.; and Baktashmotlagh, M. 2021. Keypoint-Aligned Embeddings for Image Retrieval and Re-identification. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 676–685.

Movshovitz-Attias, Y.; Toshev, A.; Leung, T. K.; Ioffe, S.; and Singh, S. 2017. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, 360–368.

Oh Song, H.; Xiang, Y.; Jegelka, S.; and Savarese, S. 2016a. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4004–4012.

Oh Song, H.; Xiang, Y.; Jegelka, S.; and Savarese, S. 2016b. Deep metric learning via lifted structured feature embedding. In *CVPR*, 4004–4012.

Opitz, M.; Waltner, G.; Possegger, H.; and Bischof, H. 2018. Deep metric learning with bier: Boosting independent embeddings robustly. *IEEE transactions on pattern analysis and machine intelligence*, 42(2): 276–290.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3967–3976.

Qian, Q.; Shang, L.; Sun, B.; Hu, J.; Li, H.; and Jin, R. 2019. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6450–6458.

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.

Seidenschwarz, J. D.; Elezi, I.; and Leal-Taixé, L. 2021. Learning intra-batch connections for deep metric learning. In *International Conference on Machine Learning*, 9410–9421. PMLR.

Sergios, T.; and Konstantinos, K. 2003. *Pattern Recognition*. Academic Press.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*.

Teh, E. W.; DeVries, T.; and Taylor, G. W. 2020. Proxynca++: Revisiting and revitalizing proxy neighborhood

component analysis. In *European Conference on Computer Vision (ECCV)*. Springer.

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7): 926–930.

Wang, F.; Xiang, X.; Cheng, J.; and Yuille, A. L. 2017a. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, 1041–1049.

Wang, J.; Zhou, F.; Wen, S.; Liu, X.; and Lin, Y. 2017b. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, 2593–2601.

Wang, X.; Han, X.; Huang, W.; Dong, D.; and Scott, M. R. 2019a. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5022–5030.

Wang, X.; Hua, Y.; Kodirov, E.; Hu, G.; Garnier, R.; and Robertson, N. M. 2019b. Ranked list loss for deep metric learning. In *CVPR*, 5207–5216.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001. *California Institute of Technology*.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*, 499–515. Springer.

Wu, Z.; Efros, A. A.; and Yu, S. X. 2018. Improving Generalization via Scalable Neighborhood Component Analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xuan, H. 2020. Improved embeddings with easy positive triplet mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2474–2482.

Xuan, H.; Souvenir, R.; and Pless, R. 2018. Deep Randomized Ensembles for Metric Learning. *European Conference on Computer Vision*.

Yuan, Y.; Yang, K.; and Zhang, C. 2017. Hard-Aware Deeply Cascaded Embedding. In *2017 IEEE International Conference on Computer Vision (ICCV)*.

Zhai, A.; and Wu, H.-Y. 2019. Classification is a strong baseline for deep metric learning. *BMVC*.

Zheng, F.; Deng, C.; Sun, X.; Jiang, X.; Guo, X.; Yu, Z.; Huang, F.; and Ji, R. 2019. Pyramidal person re-identification via multi-loss dynamic training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8514–8522.

Zheng, Y.; Pal, D. K.; and Savvides, M. 2018. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5089–5097.

Zhu, Y.; Yang, M.; Deng, C.; and Liu, W. 2020. Fewer is More: A Deep Graph Metric Learning Perspective Using Fewer Proxies. *arXiv preprint arXiv:2010.13636*.