Ensemble Soft-Margin Softmax Loss for Image Classification

Xiaobo Wang\textsuperscript{1,2,*}, Shifeng Zhang\textsuperscript{1,2,*}, Zhen Lei\textsuperscript{1,2,†}, Si Liu\textsuperscript{3}, Xiaojie Guo\textsuperscript{4}, Stan Z. Li\textsuperscript{5,1,2}

\textsuperscript{1} CBSR&NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
\textsuperscript{2} University of Chinese Academy of Sciences, Beijing, China
\textsuperscript{3} School of Computer Science and Engineering, Beihang University, Beijing, China
\textsuperscript{4} School of Computer Software, Tianjin University, Tianjin, China
\textsuperscript{5} Faculty of Information Technology, Macau University of Science and Technology, Macau, China

\{xiaobo.wang,shifeng.zhang,zlei,szli\}@nlpr.ia.ac.cn, fifthzombiesi@gmail.com, xj.max.guo@gmail.com

\begin{abstract}
Softmax is arguably one of the most popular losses to train CNN models for image classification. However, recent works have exposed its limitation on feature discriminability. This paper casts a new viewpoint on the weakness of softmax loss. On the one hand, the CNN features learned using the softmax loss are often inadequately discriminative. We hence introduce a soft-margin softmax function to explicitly encourage the discrimination between different classes. On the other hand, the learned classifier of softmax loss is weak. We propose to assemble multiple these weak classifiers to a strong one, inspired by the recognition that the diversity among weak classifiers is critical to a good ensemble. To achieve the diversity, we adopt the Hilbert-Schmidt Independence Criterion (HSIC). Considering these two aspects in one framework, we design a novel loss, named as Ensemble soft-Margin Softmax (EM-Softmax). Extensive experiments on benchmark datasets are conducted to show the superiority of our design over the baseline softmax loss and several state-of-the-art alternatives.
\end{abstract}

1 Introduction

Image classification is a fundamental yet still challenging task in machine learning and computer vision. Over the past years, deep Convolutional Neural Networks (CNNs) have greatly boosted the performance of a series of image classification tasks, like object classification [Krizhevsky et al., 2012; He et al., 2016; Liu et al., 2016], face verification [Wen et al., 2016; Wang et al., 2015; Zhang et al., 2016; Liu et al., 2017a; Wang et al., 2017a] and hand-written digit recognition [Goodfellow et al., 2013; Lin et al., 2013], etc. Deep convolutional neural networks naturally integrate low/mid/high-level features and classifiers in an end-to-end multi-layer fashion, leading CNNs to the strong visual representation ability as well as their current significant positions.

\textsuperscript{*}These authors contributed equally to this work.
\textsuperscript{†}Corresponding author

To train a deep model, the loss functions, such as (square) hinge loss, contrastive loss, triplet loss and softmax loss, etc., are usually equipped. Among them, the softmax loss is arguably the most popular one [Liu et al., 2016], which consists of three components, including the last fully connected layer, the softmax function, and the cross-entropy loss\textsuperscript{1}. It is widely adopted by many CNNs [Krizhevsky et al., 2012; Simonyan and Andrew, 2014; He et al., 2016] due to its simplicity and clear probabilistic interpretation. However, the works [Liu et al., 2016; Wen et al., 2016; Zhang et al., 2016] have shown that the original softmax loss is inadequate due to the lack of encouraging the discriminability of CNN features. Recently, a renewed trend is to design more effective losses to enhance the performance. But this is non-trivial because a new designed loss usually should be easily optimized by the Stochastic Gradient Descent (SGD) [LeCun et al., 1998a].

To improve the softmax loss, existing works can be mainly categorized into two groups. One group tries to refine the cross-entropy loss of softmax loss. Sun et al. [Sun et al., 2014] trained the CNNs with the combination of softmax loss and contrastive loss, but the pairs of training samples are difficult to select. Schroff et al. [Schroff et al., 2015] used the triplet loss to minimize the distance between an anchor sample and a positive sample (of the same identity), as well as maximize the distance between the anchor and a negative sample (of different identities). However, requiring a multiple of three training samples as input makes it inefficient. Tang et al. [Tang, 2013] replaced the cross-entropy loss with the hinge loss, while Liu et al. [Liu et al., 2017b] employed a congerous cosine loss to enlarge the inter-class distinction as well as alleviate the inner-class variance. Its drawback is that these two losses are frequently unstable. Recently, Wen et al. [Wen et al., 2016] introduced a center loss together with the softmax loss. Zhang et al. [Zhang et al., 2016] proposed a range loss to handle the case of the long tail distribution of data. Both of them have achieved promising results on face verification task. However, the objective of the open-set face verification (i.e., mainly to learn discriminative features) is different from that of the closed-set image classi-

\textsuperscript{1}The details of each component will be described in section 2.1.
2 Preliminary Knowledge

2.1 Softmax Loss

Assume that the output of a single image through deep convolutional neural networks is \( x \) (i.e., CNN features), where \( x \in \mathbb{R}^d \), \( d \) is the feature dimension. Given a mini-batch of labeled images, their outputs are \( \{x_1, x_2, \ldots, x_n\} \). The corresponding labels are \( \{y_1, y_2, \ldots, y_n\} \), where \( y_i \in \{1, 2, \ldots, K\} \) is the class indicator, and \( K \) is the number of classes. Similar to the work [Liu et al., 2016], we define the complete softmax loss as the pipeline combination of the last fully connected layer, the softmax function and the cross-entropy loss. The last fully connected layer transforms the feature \( x \) into a primary score \( z = [z_1, z_2, \ldots, z_K]^T \in \mathbb{R}^K \) through multiple parameters \( W = [w_1^T, w_2^T, \ldots, w_K^T] \in \mathbb{R}^{d \times K} \), which is formulated as: \( z_k = w_k^T x = x^T w_k \). Generally speaking, the parameter \( w_k \) can be regarded as the linear classifier of class \( k \). Then, the softmax function is applied to transform the primary score \( z_k \) into a new predicted class score as: \( s_k = \frac{e^{z_k}}{e^{z_1} + \ldots + e^{z_K}} \). Finally, the cross-entropy loss \( L_S = -\log(s_y) \) is employed.

2.2 Hilbert-Schmidt Independence Criterion

The Hilbert-Schmidt Independence Criterion (HSIC) was proposed in [Gretton et al., 2005; Wang et al., 2018] to measure the (in)dependence of two random variables \( X \) and \( Y \).

**Definition 1 (HSIC)** Consider \( n \) independent observations drawn from \( p_{xy} \), \( Z = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq X \times Y \), an empirical estimator of HSIC(\( Z, F, G \)), is given by:

\[
\text{HSIC}(Z, F, G) = (n-1)^{-2} \text{tr}(K_1 H K_2 H),
\]

where \( K_1 \) and \( K_2 \) are the Gram matrices with \( k_{1,ij} = k_1(x_i, x_j) \), \( k_{2,ij} = k_2(y_i, y_j) \), \( k_1(x_i, x_j) \) and \( k_2(y_i, y_j) \) are the kernel functions defined in space \( F \) and \( G \), respectively. \( H = I - n^{-1} \mathbf{1} \mathbf{1}^T \) centers the Gram matrix to have zero mean. Note that, according to Eq. (1), to maximize the independence between two random variables \( X \) and \( Y \), the empirical estimate of HSIC, i.e., \( \text{tr}(K_1 H K_2 H) \) should be minimized.

3 Problem Formulation

Inspired by the recent works [Liu et al., 2016; Wen et al., 2016; Zhang et al., 2016], which argue that the original softmax loss is inadequate due to its non-discriminative features. They either reformulate the softmax function into a new desired one (e.g., Sparsemax [Martins and Astudillo, 2016] and L-softmax [Liu et al., 2016] etc.) or add additional constraints to refine the original softmax loss (e.g., contrastive loss [Sun et al., 2014] and center loss [Wen et al., 2016] etc.). Here, we follow this argument but cast a new viewpoint on the weakness, say the extracted features are not discriminative meanwhile the learned classifier is not strong.

3.1 Soft-Margin Softmax Function

To enhance the discriminability of CNN features, we design a new soft-margin softmax function to enlarge the margin between different classes. We first give a simple example to describe our intuition. Consider the binary classification and we have a sample \( x \) from class 1. The original softmax loss is to enforce \( w_1^T x > w_2^T x \) (i.e., \( \|w_1\|\|x\| \cos(\theta_1) > \|w_2\|\|x\| \cos(\theta_2) \)) to classify \( x \) correctly. To make this objective more rigorous, the work L-Softmax [Liu et al., 2016] introduced an angular margin:

\[
\|w_1\|\|x\| \cos(\theta_1) \geq \|w_1\|\|x\| \cos(m \theta_1) > \|w_2\|\|x\| \cos(\theta_2),
\]

and used the intermediate value \( \|w_1\|\|x\| \cos(m \theta_1) \) to replace the original \( \|w_1\|\|x\| \cos(\theta_1) \) during training. In that way, the discrimination between class 1 and class 2 is explicitly emphasized. However, to make \( \cos(m \theta_1) \) derivable, \( m \) should be a positive integer. In other words, the angular margin cannot go through all possible angles and is a hard one. Moreover, the forward and backward computation are complex due to the angular margin involved. To address these issues, inspired by the works [Sun et al., 2014; Liang et al., 2017; Bell and Bala., 2015], we here introduce a soft distance margin and simply let

\[
w_1^T x \geq w_1^T x - m > w_2^T x,
\]

\[\text{For convenience, we denote it as softmax classifier.}\]
where \( m \) is a non-negative real number and is a distance margin. In training, we employ \( w_i^T x - m \) to replace \( w_i^T x \), thus our multi-class soft-margin softmax function can be defined as: 
\[
s_i = \frac{e^{w_i^T x - m}}{e^{w_i^T x - m} + \sum_{k \neq i} e^{w_k^T x}}.
\]
Consequently, the soft-Margin Softmax (M-Softmax) loss is formulated as:
\[
L_M = - \log \frac{e^{w_i^T x - m}}{e^{w_i^T x - m} + \sum_{k \neq i} e^{w_k^T x}}.
\] (4)

Obviously, when \( m \) is set to zero, the designed M-Softmax loss Eq. (4) becomes identical to the original softmax loss.

### 3.2 Diversity Regularized Ensemble Strategy

Though learning discriminative features may result in better classifier as these two components highly depend on each other, the classifier may not be strong enough without explicitly encouragement. To learn a strong one, as indicated in [Guo et al., 2017], a combination of various classifiers can improve predictions. Thus we adopt the ensemble strategy. Prior to formulating our ensemble strategy, we revisit that the most popular way to train an ensemble in deep learning is arguably the dropout [Hinton et al., 2012]. The idea behind dropout is to train an ensemble of DNNs by randomly dropping the activations and average the results of the whole ensemble instead of training a single DNN. However, in the last fully connected layer of softmax loss, dropout is usually not permitted because it will lose the useful label information, especially with the limited training samples. Therefore, we need to design a new manner to assemble weak classifiers.

Without loss of generality, we take two weak softmax classifiers \( W_v = [w_1^v, w_2^v, \ldots, w_K^v] \in \mathbb{R}^{d \times K} \) and \( W_u = [w_1^u, w_2^u, \ldots, w_K^u] \in \mathbb{R}^{d \times K} \) as an example to illustrate the main idea. Specifically, it has been well-recognized that the diversity of weak classifiers is of utmost importance to a good ensemble [Guo et al., 2017; Li et al., 2012]. Here, we exploit the diverse/complementary information across different weak classifiers by enforcing them to be independent. High independence of two weak classifiers \( W_v \) and \( W_u \) means high diversity of them. Classical independence criteria like the Spearman's rho and Kendall's tau [Fredricks and Nelsen., 2007], can only exploit linear dependence. The recent exclusivity regularized term [Guo et al., 2017; Wang et al., 2017b] and ensemble pruning [Li et al., 2012] may be good candidates for classifier ensemble, but both of them are difficult to differentiate. Therefore, these methods are not suitable for assembling the weak softmax classifiers.

In this paper, we employ the Hilbert-Schmidt Independence Criterion (HSIC) to measure the independence (i.e., diversity) of weak classifiers, mainly for two reasons. One is that the HSIC measures the dependence by mapping variables into a Reproducing Kernel Hilbert Space (RKHS), such that the nonlinear dependence can be addressed. The other one is that the HSIC is computational efficient. The empirical HSIC in Eq. (1) turns out to be the trace of product of weak classifiers, which can be easily optimized by the typical SGD. Based on the above analysis, we naturally minimize the following constraint according to Eq. (1):
\[
\text{HSIC}(W_v, W_u) = (K - 1)^{-2} \text{tr}(K_v H K_u H).
\] (5)

For simplicity, we adopt the inner product kernel for the proposed HSIC, say \( K = W^T W \) for both \( W_v \) and \( W_u \). Considering the multiple ensemble settings and ignoring the scaling factor \((K - 1)^{-2}\) of HSIC for notational convenience, leads to the following equation:
\[
\sum_{u=1; u \neq v}^{V} \text{HSIC}(W_v, W_u) = \sum_{u=1; u \neq v}^{V} \text{tr}(K_v H K_u H) = \sum_{u=1; u \neq v}^{V} \text{tr}(W_v^T H W_u H W_u^T) = \text{tr}(W_v^T K_v^T W_u^T H),
\] (6)

where \( K_v = \sum_{v=1; v \neq u}^{V} H W_u^T W_u H \) is the centered matrix defined in section 2.2, and \( H^T = H \). However, according to the formulation Eq. (6), we can see that the HSIC constraint is value-aware, the diversity is determined by the value of weak classifiers. If the magnitude of different weak classifiers is quite large, the diversity may not be well handled. To avoid the scale issue, we use the normalized weak classifiers to compute the diversity. In other words, if not specific, the weak classifiers \( W_v \), where \( v \in \{1, 2, \ldots, V\} \) are normalized in Eq. (6). Merging the diversity constraint into the original softmax loss, leads to Ensemble Softmax (E-Softmax) loss as:
\[
L_E = \sum_{v=1}^{V} \left[- \log \frac{e^{x_i^T W_v^w}}{\sum_{k=1}^{K} e^{x_i^T W_k^w}} + \lambda \text{tr}(W_v K_v^T W_v^T) \right],
\] (7)

where \( \lambda \) is a hyperparameter to balance the importance of diversity. The backward propagation of \( W_v \) is computed as \( \frac{\partial L_E}{\partial W_v} = \frac{\partial L}{\partial W_v} + \lambda W_v K_v^T \). Clearly, the update of the weak classifier \( W_v \) is co-determined by the initializations and other weak classifiers (i.e., \( K_v \) is computed based on other classifiers). This means that the diversity of different weak classifiers will be explicitly enhanced.

Since feature extracting and classifier learning is an end-to-end framework, we prefer to simultaneously harness them. Now, putting all concerns, say Eqs. (4) and (6), together results in our final Ensemble soft-Margin Softmax (EM-Softmax) loss \( L_{EM} \) as follows:
\[
\sum_{v=1}^{V} \left[- \log \frac{e^{x_i^T W_v^w}}{\sum_{k=1}^{K} e^{x_i^T W_k^w}} + \lambda \text{tr}(W_v K_v^T W_v^T) \right].
\] (8)

### 3.3 Optimization

In this part, we show that the proposed EM-Softmax loss is trainable and can be easily optimized by the typical SGD. Specifically, we implement the CNNs using the well-known Caffe [Jia et al., 2014] library and use the chain rule to compute the partial derivative of each \( W_v \) and the feature \( x \) as:
\[
\frac{\partial L_{EM}}{\partial W_v} = \frac{\partial L_E}{\partial W_v} \frac{\partial W_v}{\partial W_v} + \lambda W_v K_v^T,
\] (9)
\[
\frac{\partial L_{EM}}{\partial x} = \sum_{v=1}^{V} \frac{\partial L_E}{\partial W_v} \frac{\partial W_v}{\partial x} + \frac{\partial L}{\partial W_v} \frac{\partial W_v}{\partial x},
\]
where the computation forms of \( \frac{\partial L_E}{\partial W_v}, \frac{\partial W_v}{\partial W_v}, \frac{\partial W_v}{\partial x}, \frac{\partial L}{\partial W_v} \) are the same as the original softmax loss.
4 Experiments

4.1 Dataset Description

**MNIST** [LeCun et al., 1998b]: The MNIST is a dataset of handwritten digits (from 0 to 9) composed of $28 \times 28$ pixel gray scale images. There are 60,000 training images and 10,000 test images. We scaled the pixel values to the $[0,1]$ range before inputting to our neural network.

**CIFAR10/CIFAR10+** [Krizhevsky and Hinton, 2009]: The CIFAR10 contains 10 classes, each with 5,000 training samples and 1,000 test samples. We first compare EM-Softmax loss with others under no data augmentation setup. For the data augmentation, we follow the standard technique in [Lee et al., 2015; Liu et al., 2016] for training, that is, 4 pixels are padded on each side, and a $32 \times 32$ crop is randomly sampled from the padded image or its horizontal flip. In testing, we only evaluate the single view of the original $32 \times 32$ image. In addition, before inputting the images to the network, we subtract the per-pixel mean computed over the training set from each image.

**CIFAR100/CIFAR100+** [Krizhevsky and Hinton, 2009]: We also evaluate the performance of the proposed EM-Softmax loss on CIFAR100 dataset. The CIFAR100 dataset is the same size and format as the CIFAR10 dataset, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. For the data augmentation set CIFAR100+, similarly, we follow the same technique provided in [Lee et al., 2015; Liu et al., 2016].

**ImageNet32** [Chrabaszcz et al., 2017]: The ImageNet32 is a downsampled version of the ImageNet 2012 challenge dataset, which contains exactly the same number of images as the original ImageNet, i.e., 1281, 167 training images and 50,000 validation images for I, 000 classes. All images are downsampled to $32 \times 32$. Similarly, we subtract the per-pixel mean computed over the downsampled training set from each image before feeding them into the network.

4.2 Compared Methods

We compare our EM-Softmax loss with recently proposed state-of-the-art alternatives, including the baseline softmax loss (Softmax), the margin-based hinge loss (HingeLoss [Tang, 2013]), the combination of softmax loss and center loss (CenterLoss [Wen et al., 2016]), the large-margin softmax loss (L-Softmax [Liu et al., 2016]), the angular margin softmax loss (A-Softmax [Liu et al., 2017a]) and the normalized features softmax loss (N-Softmax [Wang et al., 2017a]). The source codes of Softmax and HingeLoss are provided in Caffe community. For other compared methods, their source codes can be downloaded from the github or from authors’ webpages. For fair comparison, the experimental results are cropped from the paper [Liu et al., 2016] (indicated as *) or obtained by trying our best to tune their corresponding hyperparameters. Moreover, to verify the gain of our soft margin and ensemble strategy, we also report the results of the M-Softmax loss Eq. (4) and the E-Softmax loss Eq. (7).

4.3 Implementation Details

In this section, we give the major implementation details on the baseline works and training/testing settings. The important statistics are provided as follows:

Baseline works. To verify the universality of EM-Softmax, we choose the work [Liu et al., 2016] as the baseline. We strictly follow all experimental settings in [Liu et al., 2016], including the CNN architectures (LiuNet), the datasets, the pre-processing methods and the evaluation criteria.

**Training**. The proposed EM-Softmax loss is appended after the feature layer, i.e., the second last inner-product layer. We start with a learning rate of 0.1, use a weight decay of 0.0005 and momentum of 0.9. For MNIST, the learning rate is divided by 10 at 8k and 14k iterations. For CIFAR10/CIFAR10+, the learning rate is also divided by 10 at 8k and 14k iterations. For CIFAR100/CIFAR100+, the learning rate is divided by 10 at 12k and 15k iterations. For all these three datasets, the training eventually terminates at 20k iterations. For ImageNet32, the learning rate is divided by 10 at 15k, 25k and 35k iterations, and the maximal iteration is 40k. The accuracy on validation set is reported.

**Testing**. At testing stage, we simply construct the final ensemble classifier by averaging weak classifiers: \( W = \frac{1}{P} \sum_{v=1}^{P} W_v \). Finally, \( W \) is the learned strong classifier and we use it with the discriminative feature \( x \) to predict labels.

4.4 Accuracy vs HyperParameter

The soft-margin softmax function (4) involves one parameter \( m \). Inspired by [Bell and Bala., 2015], we try a few different \( m \in \{0.1, 0.5, 1, 5, 10\} \) and select the one that performs best. Regarding the diversity regularization (6), it involves the trade-off parameter \( \lambda \) and the ensemble number \( V \). In this part, we mainly report the sensitiveness of these two variables on CIFAR10 and CIFAR100. The subfigures of Figure 1 displays the testing accuracy rate vs. parameter \( \lambda \) plot of EM-Softmax loss. We set \( V = 2 \) and vary \( \lambda \) from 0 to 30 to learn different models. From the curves, we can observe that, as \( \lambda \) grows, the accuracy grows gradually at the very beginning and changes slightly in a relatively large range. The experimental results are insensitive to \( \lambda \in [0.01, 1.0] \). Too large \( \lambda \) may hinder the performance because it will degenerate the focus of classification part in Eq. (8). Moreover, it also reveals the effectiveness of the diversity regularization (\( \lambda \neq 0 \) vs. \( \lambda = 0 \)). The subfigures of Figure 2 is the testing accuracy rate vs. ensemble number \( V \) plot of EM-Softmax loss. We set \( \lambda = 0.1 \) and vary \( V \) from 1 to 10. From the curves, we can see that a single classifier (\( V = 1 \)) is weak for
classification. Our EM-Softmax ($V \geq 2$) benefits from the ensemble number of weak classifiers. But the improvement is slight when the ensemble number $V$ is big enough. The reason behind this may come from two aspects. One is that too many classifiers ensemble will bring too much redundant information, thus the improvement is finite. The other one is that the discriminative features help to promote weak classifiers, without needing assemble too many classifiers. Based on the above observations, we empirically suggest $V = 2$ in practice to avoid the parameter explosion of weak classifiers.

### 4.5 Classification Results on MNIST and CIFAR

Table 1 provides the quantitative comparison among all the competitors on MNIST and CIFAR datasets. The bold number in each column represents the best performance.

On MNIST dataset, it is well-known that this dataset is typical and easy for deep image classification, and all the competitors can achieve over 99% accuracy rate. So the improvement of our EM-Softmax is not quite large. From the experimental results, we observe that A-Softmax [Liu et al., 2017a], L-Softmax [Liu et al., 2016], the proposed EM-Softmax and its degenerations M-Softmax and E-Softmax outperform the other compared methods. Moreover, we have achieved a high accuracy rate 99.73% on this dataset.

On CIFAR10/CIFAR10+ dataset, we can see that our EM-Softmax significantly boosts the performance, achieving at least 2% improvement over the baseline Softmax. Considering all the competitors can achieve over 90% accuracy rate on this dataset, the improvement is significant. To the soft distance margin M-Softmax, it is slightly better than the hard angular margin L-Softmax [Liu et al., 2016] and A-Softmax [Liu et al., 2017a] because the soft margin can go through all possible desired ones. It is much better than Softmax because of the learned discriminative features. To the ensemble softmax E-Softmax, it is about 1% higher than the baseline Softmax because of the assembled strong classifier. Our EM-Softmax absorbs the complementary merits from these two aspects (i.e., discriminative features and strong classifier).

On CIFAR100/CIFAR100+ dataset, it can be observed that most of competitors achieve relatively low performance. The major reason is that the large variation of subjects, color and textures and the fine-grained category involve in this dataset. Even so, our EM-Softmax still reaches significant improvements, at least 5% higher than the baseline Softmax. Compared with the recent L-Softmax [Liu et al., 2016] and A-Softmax [Liu et al., 2017a], our EM-Softmax can achieve about 3% improvement. Moreover, we can see a similar trend as that shown on CIFAR10/CIFAR10+ dataset, that is, the EM-Softmax loss is generally better than its degenerations M-Softmax and E-Softmax.

### 4.6 Classification Results on ImageNet

We report the top-1 and top-5 validation accuracy rates on ImageNet32 in Table 2. From the numbers, we can see that the results exhibit the same phenomena that emerged on CIFAR dataset. In particular, the proposed EM-Softmax achieves a higher top-1 accuracy by 2.4% and top-5 accuracy by 2.2% in comparison with the baseline Softmax. The improvements are significant as the imagenet is very large and difficult for image classification, especially with such a smaller down-sampled size ($32 \times 32$). Compared with other competitors, our EM-Softmax can achieve at least 1% improvement. The re-

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST</th>
<th>CIFAR10</th>
<th>CIFAR10+</th>
<th>CIFAR100</th>
<th>CIFAR100+</th>
</tr>
</thead>
<tbody>
<tr>
<td>HingeLoss [Tang, 2013]</td>
<td>99.53*</td>
<td>90.09*</td>
<td>93.31*</td>
<td>67.10*</td>
<td>68.48</td>
</tr>
<tr>
<td>CenterLoss [Wen et al., 2016]</td>
<td>99.41</td>
<td>91.65</td>
<td>93.82</td>
<td>69.23</td>
<td>70.97</td>
</tr>
<tr>
<td>A-Softmax [Liu et al., 2017a]</td>
<td>99.66</td>
<td>91.72</td>
<td>93.98</td>
<td>70.87</td>
<td>72.23</td>
</tr>
<tr>
<td>N-Softmax [Wang et al., 2017a]</td>
<td>99.48</td>
<td>91.46</td>
<td>93.90</td>
<td>70.49</td>
<td>71.85</td>
</tr>
<tr>
<td>L-Softmax [Liu et al., 2016]</td>
<td>99.69*</td>
<td>92.42*</td>
<td>94.08*</td>
<td>70.47*</td>
<td>71.96</td>
</tr>
</tbody>
</table>

Baseline

<table>
<thead>
<tr>
<th>Softmax</th>
<th>99.60*</th>
<th>90.95*</th>
<th>93.50*</th>
<th>67.26*</th>
<th>69.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-Softmax</td>
<td>99.70</td>
<td>92.50</td>
<td>94.27</td>
<td>70.72</td>
<td>72.54</td>
</tr>
<tr>
<td>E-Softmax</td>
<td>99.69</td>
<td>92.38</td>
<td>93.92</td>
<td>70.34</td>
<td>71.33</td>
</tr>
<tr>
<td>EM-Softmax</td>
<td>99.75</td>
<td>93.31</td>
<td>95.02</td>
<td>72.21</td>
<td>75.69</td>
</tr>
</tbody>
</table>

Table 1: Recognition accuracy rate (%) on MNIST, CIFAR10/CIFAR10+ and CIFAR100/CIFAR100+ datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>HingeLoss [Tang, 2013]</td>
<td>46.52</td>
<td>71.56</td>
</tr>
<tr>
<td>CenterLoss [Wen et al., 2016]</td>
<td>47.43</td>
<td>71.98</td>
</tr>
<tr>
<td>A-Softmax [Liu et al., 2017a]</td>
<td>48.12</td>
<td>72.51</td>
</tr>
<tr>
<td>N-Softmax [Wang et al., 2017a]</td>
<td>47.52</td>
<td>72.06</td>
</tr>
<tr>
<td>L-Softmax [Liu et al., 2016]</td>
<td>47.85</td>
<td>72.63</td>
</tr>
</tbody>
</table>

Baseline

<table>
<thead>
<tr>
<th>Softmax</th>
<th>46.89</th>
<th>71.94</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-Softmax</td>
<td>48.21</td>
<td>72.77</td>
</tr>
<tr>
<td>E-Softmax</td>
<td>48.16</td>
<td>72.99</td>
</tr>
<tr>
<td>EM-Softmax</td>
<td>49.22</td>
<td>74.22</td>
</tr>
</tbody>
</table>

Table 2: The top-1 and top-5 recognition accuracy rate (%) on ImageNet32 validation set.
results presented in Table 2 also reveal that our EM-Softmax can benefits from the discriminative features (M-Softmax) and the strong classifier (E-Softmax).

### 4.7 EM-Softmax vs. Model Averaging

To validate the superiority of our weak classifiers ensemble strategy (i.e., EM-Softmax) over the simple model averaging, we conduct two kinds of model averaging experiments on both CIFAR10 and CIFAR100 datasets. One is the model averaging of the same architecture but with different numbers of filters (i.e., 48/48/96/192, 64/64/128/256 and 96/96/192/382)\(^4\). For convenience, we use CNN(48), CNN(64) and CNN(96) to denote them, respectively. The other one is the model averaging of different CNN architectures. We use AlexNet [Krizhevsky et al., 2012] (much larger than LiuNet [Liu et al., 2016]) and CIFAR10 Full (much smaller than LiuNet [Liu et al., 2016]) architectures as an example, which have been provided in the standard Caffe [Jia et al., 2014] library\(^5\). For comparison, all the architectures of these two kinds of model averaging strategies are equipped with the original softmax loss. Table 3 provides the experimental results of model averaging on CIFAR10 and CIFAR100 datasets, from which, we can see that the strategy of model averaging is beneficial to boost the classification performance. However, the training is time-consuming and the model size is large. Compared our weak classifiers ensemble (EM-Softmax) with these two kinds of model averaging, we can summarize that the accuracy of our EM-Softmax is generally higher and our model size is much lower than the simple model averaging.

### 4.8 EM-Softmax vs. Dropout

Dropout is a popular way to train an ensemble and has been widely adopted in many works. The idea behind it is to train an ensemble of DNNs by randomly dropping the activations\(^6\) and averaging the results of the whole ensemble. The adopted architecture LiuNet [Liu et al., 2016] contains the second last FC layer and is without the dropout. To validate the gain of our weak classifiers ensemble, we add the dropout technique to the second last fully-connected layer and conduct the experiments of Softmax, Softmax+Dropout and EM-Softmax+Dropout. The proportion of dropout is tuned in \([0.3, 0.5, 0.7]\) and the diversity hyperparameter \(\lambda\) of our EM-Softmax is 0.1. Table 4 gives the experimental results of dropout on CIFAR10 and CIFAR100 datasets. From the numbers, we can see that the accuracy of our EM-Softmax is much higher than the Softmax+Dropout, which has shown the superiority of our weak classifier ensemble over the dropout strategy. Moreover, we empirically find that the improvement of dropout on both Softmax and EM-Softmax losses is not big with the adopted CNN architecture. To sum up, our weak classifiers ensemble is superior to the simple dropout strategy and can seamlessly incorporate with it.

### 4.9 Running Time

We give the time cost vs. accuracy of EM-Softmax, Softmax and two kinds of model averaging on CIFAR10. From Table 5, the training time on 2 Titan X GPU is about 1.01h, 0.99h, 4.82h and 3.02h, respectively. The testing time on CPU (Intel Xeon E5-2660v0@2.20Ghz) is about 3.1m, 2.5m, 8.1m and 10m, respectively. While the corresponding accuracy is 93.31%, 90.90%, 90.97% and 91.99%, respectively. Considering time cost, model size and accuracy together, our weak classifiers ensemble EM-Softmax is a good candidate.

### 5 Conclusion

This paper has proposed a novel ensemble soft-margin softmax loss (i.e., EM-Softmax) for deep image classification. The proposed EM-Softmax loss benefits from two aspects. One is the designed soft-margin softmax function to make the learned CNN features to be discriminative. The other one is the ensemble weak classifiers to learn a strong classifier. Both of them can boost the performance. Extensive experiments on several benchmark datasets have demonstrated the advantages of our EM-Softmax loss over the baseline softmax loss and the state-of-the-art alternatives. The experiments have also shown that the proposed weak classifiers ensemble is...
generally better than model ensemble strategies (e.g., model averaging and dropout).

Acknowledgments
This work was supported by National Key Research and Development Plan (Grant No.2016YFC0801002), the Chinese National Natural Science Foundation Projects #61473291, #61572501, #61502491, #61572536, the Science and Technology Development Fund of Macau (No.151/2017/A, 152/2017/A), and AuthenMetric R&D Funds.

References