Group Softmax Loss with Discriminative Feature Grouping

Takumi Kobayashi
National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Japan
takumi.kobayashi@aist.go.jp

Abstract

In the supervised learning framework, a softmax cross-entropy loss is commonly applied to train deep neural networks for high-performance classification. It, however, demands large amount of annotated data and fails to learn the discriminative networks on a smaller amount of data. In this paper, we propose a novel loss measure to train the networks such that discriminative feature representation can be learned even on the smaller-scale dataset. By means of feature grouping, we effectively expose non-discriminative feature components to representation learning and formulate two types of group softmax losses to cope with the grouped features. The proposed method encourages discriminative representation across all feature components, and from a theoretical viewpoint it renders adversarial training which works for alleviating over-fitting especially on scarce training data. The experimental results on image classification tasks demonstrate that the proposed loss favorably improves performance of CNNs on various-scale data.

1. Introduction

Deep neural networks (DNNs) have been producing promising performance on various fields of pattern recognition, such as by convolutional neural networks (CNNs) [22, 34] for image classification and recurrent neural networks [9, 27] for recognizing time-series signals. While exhibiting favorable applicability in the framework of unsupervised learning [44, 3], they perform extraordinarily well through an end-to-end supervised training based on ground-truth labels [15, 41]. The DNNs are equipped with lots of parameters, therefore demanding huge amount of labeled data for being successfully trained. Though much research effort has been devoted to mitigating the problem from the viewpoints of network architecture [15, 41, 17], layer-wise operation [35, 18] and data augmentation [7, 43], the data-hunger nature is the major drawback of DNNs; actually, it is hard to train the networks from scratch on the smaller-scale datasets due to over-fitting.

Toward effective training, there is also a research direction to focus on a *loss* function which is fundamental for optimizing networks. In the supervised learning, the softmax cross-entropy loss is arguably the most commonly used loss measure, and for improving generalization performance, as in SVMs [37], a large margin criterion can be introduced to the softmax loss in various ways [26, 38, 5, 23]. In the end-to-end learning framework of DNNs, the large-margin losses contribute not only to improving a classifier but also to providing a more discriminative feature representation. The other variants of softmax loss are also proposed to mitigate the bottleneck of the softmax formulation [20, 4]. These works directly modify the loss function itself based on the logits that the last fully-connected (FC) layer produces.

In this work, we focus on the other aspect of loss. While the previous works mainly investigate *what* type of loss is useful, we analyze *how* the loss is applied to features for further effective training. Softmax loss is applied to the logits, i.e., inner-products between the feature vector at the penultimate layer and FC weights, and thus deals with the features as a *whole* rather ignoring their individual characteristics. In contrast, we focus on discriminativity of *each* feature component for further encouraging effective feature learning across all the feature components. For that purpose, the whole feature components are separated into several *groups* to reveal each feature's discriminativity in a loss, and the same partitioning can be applied to the FC classifier in accordance with the feature grouping. Thereby, this grouping keeps the form of FC classifier embedded in NNs, without degeneration nor reformulation in the layer. To cope with the grouping in a loss, we formulate two types of *group* softmax loss as well as provide an effective feature grouping to gather non-discriminative feature components which are fed into the group softmax loss separately from the discriminative ones for fairly promoting discriminative learning across all the components. From a theoretical viewpoint, our method renders adversarial training [13] at the level of loss which works as a favorable regularization

to mitigate overfitting especially on scarce data. Thus, the group softmax loss equipped with the discriminative feature grouping facilitates training networks even on smaller-scale datasets. Our contributions are summarized as follows; 1) Grouping feature components naturally induces two types of group softmax loss. 2) The feature grouping is formulated in a computationally efficient form to fairly encourage discriminative feature representation across all the feature components. 3) The proposed loss is thoroughly analyzed/evaluated in the experiments on various datasets.

1.1. Related works

Loss and regularization. There are a variety of softmax loss variants such as for enhancing large-margin [26, 5] and mitigating softmax bottleneck [20, 4]. These methods design novel types of loss functions, thus being different from ours which considers how to apply a loss function to features via feature grouping. To work with the losses, some regularization methods, such as DropOut [35, 30] and CenterLoss [39], are proposed; in this work, we consider DropOut applied only at the penultimate layer for fair comparison. From an adversarial viewpoint, our method is also regarded as regularization via manipulating feature components like [35, 30] without imposing any explicit regularization term [39]; the discussion on the connection to DropOut [35, 30] is shown in Sec. 3.2.

Dividing feature space. Grouping feature components is related to the concept of splitting a feature space found in product quantization (PQ) [19, 12] which are extended in the framework of deep learning [42, 21]. The PQ methods leverage the feature sub-spaces to efficiently encode/quantize feature vectors into a large number of codes which are combinatorially described by means of a Cartesian product of sub-codes. In this work, we harness the feature partition in a *loss* function for effectively learning the discriminative features, apart from the Cartesian product of sub-space codes. Splitting feature channels into groups can be embedded in CNNs by group convolution [22, 41, 16]. It is one of the CNN techniques to make the convolution operation computationally efficient with less number of weight parameters, which exhibits clear difference from our feature grouping in the loss. In addition, the group convolution slices the input feature channels in a fixed way, sharply contrasting with our discriminative feature grouping which dynamically changes groups during the training.

Product of softmax. The group softmax loss is connected to a product form of the softmax probabilities which is utilized in the hierarchical softmax for efficiently computing the posterior over a lot of class categories [28] and for exploiting semantic structure of the visual categories [31, 25, 6]. These methods are based on the hierarchical structure of the class categories which probabilistically induces the product (chain) of the softmax probabilities. Recently, the balanced group softmax loss [24] has been proposed by grouping imbalanced class categories to cope with imbalanced long-tail object detection. In contrast to those works, we formulate the group softmax loss through dividing *feature* space without considering class categories, and it is an upper bound on the softmax loss (Sec. 3.1).

Upper bound on the loss. In the field of distance metric learning (DML), the upper bounds on the *triplet loss* are presented in [8, 29] for efficiently training DML models based on the triplet loss which requires to cope with a huge number of triplet samples in a naive approach. In this work, we derive the upper bound on the *softmax loss*, which is a standard classification loss applied in versatile classification tasks, for enhancing discriminative feature representation. It might be noteworthy that our approach to discriminative feature grouping is slightly connected to the sample mining technique used in the pairwise loss of DML [33]. The proposed upper bounds on the softmax loss are also different from the method [2] that formulates some bounds on the softmax function, actually the log-sum-exp function contained in the softmax, for efficient Bayesian inference. In this work, while retaining the fundamental form of the softmax loss function, we propose a novel way to apply the loss function to a feature vector through grouping its components.

2. Method

We begin with the standard softmax cross-entropy loss while introducing the notations used in this paper, and then propose the grouped form of softmax loss as well as discriminative feature grouping.

2.1. Softmax cross-entropy loss

A feature vector $\boldsymbol{x} \in \mathbb{R}^D$ is produced at the penultimate layer in NNs. It is fed into the final linear classification layer parameterized by weights $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C] \in \mathbb{R}^{D\times C}$ and biases $\boldsymbol{b} \in \mathbb{R}^C$ to provide C-class logits $\bar{\boldsymbol{z}} \in \mathbb{R}^C$ for the softmax loss ℓ_1 as

$$\bar{\boldsymbol{z}} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}, \tag{1}$$

$$\mathrm{p}_y(\bar{\boldsymbol{z}}) = \frac{\exp(\bar{z}_y)}{\sum_{c=1}^{C} \exp(\bar{z}_c)}, \quad \ell_1(\bar{\boldsymbol{z}}, y) = -\log \mathrm{p}_y(\bar{\boldsymbol{z}}), \tag{2}$$

where $y \in \{1, \cdots, C\}$ is the ground-truth label assigned to the feature vector $\boldsymbol{x}$ and $\bar{z}_c$ indicates the c-th element of $\bar{\boldsymbol{z}}$. In this work, we further decompose the logit $\bar{z}_c$ into *ingredient logits* $\{z_{jc}\}_{j=1}^D$;

$$z_{jc} = W_{jc} x_j + \frac{1}{D} b_c \tag{3}$$

$$\Rightarrow \bar{z}_c = \sum_{j=1}^{D} z_{jc}, \;\; \mathrm{p}_y(\bar{\boldsymbol{z}}) = \frac{\exp(\sum_{j=1}^{D} z_{jy})}{\sum_{c=1}^{C} \exp(\sum_{j=1}^{D} z_{jc})}, \tag{4}$$

where W_{jc}, x_j and b_c are the (j,c)-th, j-th and c-th elements of $\boldsymbol{W}$, $\boldsymbol{x}$ and $\boldsymbol{b}$, respectively, and the whole ingredient logits are represented by a matrix form of $\boldsymbol{Z} = \{z_{jc}\}_{j=1,c=1}^{D,C} \in \mathbb{R}^{D\times C}$. This is computed by element-wise multiplication between the feature vector $\boldsymbol{x}$ and the classifier weights $\boldsymbol{W}$.

2.2. Group softmax loss

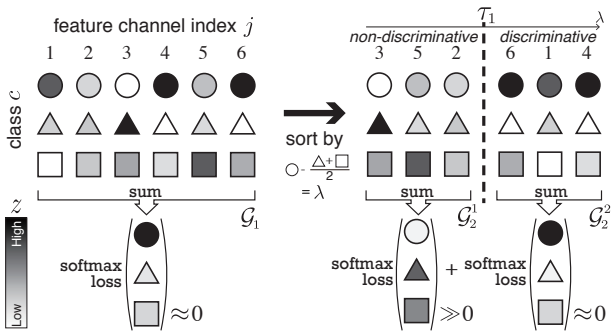
The standard softmax loss (4) uniformly aggregates all the ingredient logits $\boldsymbol{Z}$, correspondingly all feature components of $\boldsymbol{x}$, and hence could be dominated by only a few discriminative feature components as follows. Let such a discriminative feature index set be denoted by $\mathcal{D}$, and it produces $\ell_1(\bar{\boldsymbol{z}}) \approx \ell_1(\bar{\boldsymbol{z}}^{\mathcal{D}})$ by $\bar{\boldsymbol{z}}^{\mathcal{D}} = \{\bar{z}_c^{\mathcal{D}} = \sum_{j\in\mathcal{D}} z_{jc}\}_{c=1}^{C}$ in which $\bar{z}_y^{\mathcal{D}}$ is significantly larger than the other logits. Those prominent features are implied in the literature of network pruning [10]. Thus, through the uniform aggregation (4), the softmax loss disregards how much *each* feature component is learnt while concentrating on a small number of prominent features of high discriminativity (Fig. 1a). The few prominent features hinder *non*-discriminative components from being properly learned, although diversely discriminative features are required to improve generalization performance; it is especially hard to propagate discriminative learning across all the components on the scarce training data. To mitigate the issue, we partition the D feature components, actually ingredient logits z_{jc} in (3), into several *groups* in a loss formulation so that even non-discriminative features can enjoy effective learning separately from the discriminative ones (Fig. 1b). We here formulate a loss to cope with so grouped features and then provide discriminative feature grouping in Sec. 2.3 for endowing effective feature leaning with all components of $\boldsymbol{x}$.

Let the whole feature index set be denoted by $\mathcal{G}_1 = \{1,\cdots,D\}$, and suppose $\mathcal{G}_1$ is sliced into G groups $\{\mathcal{G}_G^i\}_{i=1}^G$ which are disjoint and non-empty; $|\mathcal{G}_G^i| > 0$, $\mathcal{G}_G^i \cap \mathcal{G}_G^{i'} = \emptyset$, $\sum_{i=1}^G |\mathcal{G}_G^i| = D$ where $|\mathcal{G}|$ indicates the cardinality of the set $\mathcal{G}$ and $i \neq i'$. This partitioning is applied to the features $\boldsymbol{x}$ as well as the classifier weights $\boldsymbol{W}$ to produce *group*-wise logits as

$$\bar{z}_c^{\mathcal{G}_G^i} = \left\{\sum_{j\in\mathcal{G}_G^i} W_{jc}x_j\right\} + \frac{|\mathcal{G}_G^i|}{D}b_c = \sum_{j\in\mathcal{G}_G^i} z_{jc} \qquad (5)$$

$$\Rightarrow\ \mathtt{p}_y(\bar{\boldsymbol{z}}) = \frac{\prod_{i=1}^G \exp(\bar{z}_y^i)}{\sum_{c=1}^C \prod_{i=1}^G \exp(\bar{z}_c^i)}. \qquad (6)$$

which aggregates the ingredient logit z_{jc} in the i-th group $\mathcal{G}_G^i$; for brevity, $\bar{z}_c^{\mathcal{G}_G^i}$ is simply denoted by $\bar{z}_c^i$. The group logits (5) which satisfy $\bar{z}_c = \sum_{i=1}^G \bar{z}_c^i$ further rewrite the softmax (4) into the group-wise form (6). By factorizing it with respect to groups, we can formulate two types of *group softmax losses* as



(a) Softmax loss ($G=1$) (b) Ours ($G=2$)

Figure 1. Group softmax loss with discriminative feature grouping. The ingredient logits z_{jc} (3) are depicted by gray-scale colors for values and shapes for class c; the circle ◯ indicates the ground-truth class y. The indexes j are equally divided into two groups of $\mathcal{G}_2^1$ and $\mathcal{G}_2^2$ after sorting by the discriminativity score λ, and partial softmax losses are computed on the respective groups. Non-discriminative features are exposed in our loss (b), while they are buried in the standard softmax loss (a).

$$\ell_G^I(\boldsymbol{Z},y;\mathcal{G}_G) = -\frac{1}{G}\sum_{i=1}^G \log\left[\frac{\exp(G\bar{z}_y^i)}{\sum_{c=1}^C \exp(G\bar{z}_c^i)}\right], \qquad (7)$$

$$\ell_G^{II}(\boldsymbol{Z},y;\mathcal{G}_G) = -\sum_{i=1}^G \log\left[\frac{\exp(\bar{z}_y^i)}{\sum_{c=1}^C \exp(\bar{z}_c^i)}\right]. \qquad (8)$$

These losses partition the softmax loss in terms of feature grouping $\{\mathcal{G}_G^i\}_{i=1}^G$ in contrast to [24] which groups class categories. The type-I loss ℓ_G^I introduces scaling by G to make the scale of $\bar{z}_c^i$ close to that of $\bar{z}_c$ as done in DropOut [35], while the type-II loss ℓ_G^{II} is simply composed of group-wise softmax loss based on group logits $\{\bar{z}_c^i\}_{c=1}^C$. Both types of losses are naturally extended from the softmax loss ℓ_1 (2) since they are reduced into $\ell_1 = \ell_1^I = \ell_1^{II}$ in case of $G = 1$ (single group); theoretically, the group softmax losses (7, 8) are upper bound on the softmax loss as described in Sec. 3.1.

2.3. Discriminative feature grouping

For enhancing discriminativity of *all* feature components, it is necessary to apply the group softmax loss (7, 8) with the proper feature grouping that effectively conveys *non*-discriminative feature components to the loss; actually, a naive *random* grouping is less effective in the group loss as empirically shown in Sec. 4.1. That is, we require the group that contains only the less discriminative features, excluding discriminative ones, so as to let the group loss (7, 8) focus on improving them (Fig. 1b). To this end, we measure discriminative power of each feature component based on ingredient logits.

The discriminativity at each component j can be evalu-

Algorithm 1 Group softmax loss with discriminative feature grouping

Input: $\boldsymbol{x} \in \mathbb{R}^D$: feature vector produced by CNN, $y \in \{1, \cdots, C\}$: ground-truth label,
$\boldsymbol{W} \in \mathbb{R}^{D\times C}, \boldsymbol{b} \in \mathbb{R}^C$: classifier parameters, G: number of feature partition

1: *Ingredient logit* (3): $\boldsymbol{Z} = \boldsymbol{W} \odot (\boldsymbol{x}\mathbf{1}_C^\top) + \frac{1}{D}\mathbf{1}_D\boldsymbol{b}^\top \in \mathbb{R}^{D\times C}$ where $\mathbf{1}_C \in \mathbb{R}^C$ is the vector whose elements are 1 and $\odot$ indicates Hadamard product.

2: *Discriminativity score* (9): $\boldsymbol{\lambda} = \boldsymbol{Z}(\boldsymbol{e}_y - \frac{1}{C}\mathbf{1}_C) \in \mathbb{R}^D$ where $\boldsymbol{e}_y$ is one-hot vector activating the y-th element.

3: *Grouping* (10): determine the thresholds $\{\tau_i\}_{i=0}^G$ via sorting $\{\lambda_j\}_{j=1}^D$ to provide the discriminative grouping $\{\tilde{\mathcal{G}}_G^i\}_{i=1}^G$.

4: *Group logit* (5): $\{\bar{\boldsymbol{z}}^i = \boldsymbol{Z}^\top(\sum_{j\in\tilde{\mathcal{G}}_G^i}\boldsymbol{e}_j) \in \mathbb{R}^C\}_{i=1}^G$.

Output: $\ell_G^I = -\frac{1}{G}\sum_{i=1}^G \log\frac{\exp(G\bar{z}_y^i)}{\sum_{c=1}^C\exp(G\bar{z}_c^i)}$ [type I], or $\ell_G^{II} = -\sum_{i=1}^G \log\frac{\exp(\bar{z}_y^i)}{\sum_{c=1}^C\exp(\bar{z}_c^i)}$ [type II]

ated by the difference between the ingredient logit z_{jy} of ground-truth label y and the others as

$$\lambda_j = z_{jy} - \frac{1}{C-1}\sum_{c\neq y} z_{jc} = \frac{C}{C-1}\left[z_{jy} - \frac{1}{C}\sum_{c=1}^C z_{jc}\right]. \tag{9}$$

According to the discriminativity score λ_j, we can group the feature index set $\mathcal{G}_1 = \{1, \cdots, D\}$ into G groups. Since it is generally hard to clearly partition the (continuous) discriminativity scores into two groups of discriminative and non-discriminative ones, we here simply consider $G \geq 2$ groups through *sorting* λ_j and *equal partitioning*; the process is described by using $G-1$ thresholds $\{\tau_i\}_{i=1}^{G-1}$ as

$$\tilde{\mathcal{G}}_G^i = \{j|\tau_{i-1} < \lambda_j \leq \tau_i\} \wedge |\tilde{\mathcal{G}}_G^i| = \frac{D}{G}, \forall i, \tag{10}$$

$$\text{where } \tau_{i-1} < \tau_i, \tau_0 = -\infty, \tau_G = \infty, \tag{11}$$

which provides highly *imbalanced* grouping in terms of discriminativity; $\tilde{\mathcal{G}}_G^1$ contains least discriminative features while $\tilde{\mathcal{G}}_G^G$ is highly discriminative. In practice, to further promote computational efficiency, the feature grouping is constructed in the mini-batch by averaging the discriminativity score λ_j over the mini-batch samples and the discriminative grouping $\tilde{\mathcal{G}}_G$ is shared in the mini-batch.

By integrating the discriminative grouping with the group softmax loss (7, 8), discriminative feature components are encapsulated in one group and their effects are restricted in one part of loss (Fig. 1b). Thereby, the non-discriminative feature components are freed from the effects of those prominent features and encouraged to effectively increase the discriminativity; throughout training, the less-discriminative components are exposed to the loss of the group that those components belong to. The practical procedure to compute the loss is summarized in Algorithm 1.

As the training proceeds, the number of non-discriminative components are reduced and accordingly the optimal cardinality of group $|\mathcal{G}_G^i|$ to capture them could be smaller. To cope with the situation, we integrate *multiple* group softmax losses by

$$\ell^I(\boldsymbol{Z}, y) = \frac{1}{|\mathbb{G}|}\sum_{G\in\mathbb{G}} \ell_G^I(\boldsymbol{Z}, y; \tilde{\mathcal{G}}_G), \tag{12}$$

$$\ell^{II}(\boldsymbol{Z}, y) = \frac{1}{|\mathbb{G}|}\sum_{G\in\mathbb{G}} \ell_G^{II}(\boldsymbol{Z}, y; \tilde{\mathcal{G}}_G), \tag{13}$$

where ℓ_G^I and ℓ_G^{II} are defined in (7, 8) computed by Algorithm 1, and $\mathbb{G}$ is the set of G, say $\mathbb{G} = \{1, 2\}$; $G = 1$ indicates the standard softmax loss (2). The losses (12, 13) hierarchically group feature components.

3. Discussion

3.1. Adversarial training

The proposed method can be analyzed from an adversarial viewpoint. We first show the following relationship among the group losses (7, 8) and the softmax loss (2).

Proposition 1. *Given the feature grouping $\{\mathcal{G}_G^i\}_{i=1}^G$, the softmax loss ℓ_1 (2) is upper bounded by the group softmax losses (7, 8) as*

$$\ell_1(\bar{\boldsymbol{z}}, y) \leq \ell_G^I(\boldsymbol{Z}, y; \mathcal{G}_G) \leq \ell_G^{II}(\boldsymbol{Z}, y; \mathcal{G}_G). \tag{14}$$

Proof. The proof is given in the supplementary material. □

In addition, these upper-bound losses ℓ_G^I (7) and ℓ_G^{II} (8) are *roughly* maximized with respect to grouping $\mathcal{G}$ by the discriminative feature grouping $\tilde{\mathcal{G}}_G$ (10) which provides the groups of highly imbalanced discriminativity (9); the detail is described in the supplementary material.

Therefore, minimizing the group softmax loss (7, 8) with the discriminative feature grouping (10) roughly corresponds to the following adversarial training [30] at the loss;

$$\min_{\boldsymbol{\theta}} \max_{\mathcal{G}} \ell_G(\boldsymbol{Z}(\boldsymbol{\theta}), y; \mathcal{G}), \tag{15}$$

where $\boldsymbol{\theta}$ is a parameter set of the network to be optimized and note that ℓ_G is reduced to the softmax loss by $\mathcal{G} = \mathcal{G}_1$. In other words, the proposed method *adversarially*

increases the softmax loss via feature grouping and thus in end-to-end learning it works as adversarial regularization at the level of *loss* which is effective for training especially on scarce data by alleviating over-fitting. While the standard adversarial training [13] is applied to input data (images) in a similar way to data augmentation, it should be noted that the proposed method works on features (at the penultimate layer) as in the framework of adversarial dropout [30].

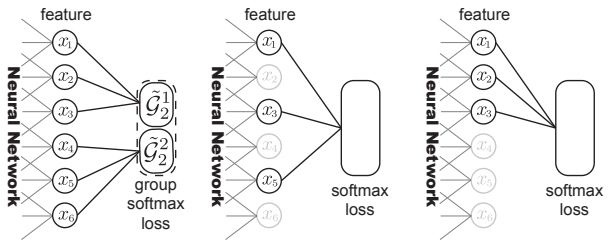
3.2. Comparison to DropOut

From a regularization perspective, the proposed method can be related to DropOut [35, 30]; we here consider the DropOut procedure that is applied to the features $\boldsymbol{x}$ produced by the penultimate layer. As illustrated in Fig. 2, the DropOut methods also operate on feature indexes through *dropping* some of feature components randomly [35] or adversarially [30], and the dropped components cannot enjoy back-propagation updates. In contrast, our method groups features without excluding any components and therefore leverages *all* the features to representation learning through *discriminative* grouping beyond random one; i.e., our method lets *all* feature components join in back-propagation learning.

It is also possible to view the proposed method as merging *group*-wise DropOuts into a single loss (updating); see Fig. 2. In our loss, group-wise parts of loss are exclusive to each other and thus each part of loss can be regarded as DropOut loss which drops the other groups; Fig. 2a contains *two* DropOuts of the one excluding $\tilde{\mathcal{G}}_2^1$ and the other excluding $\tilde{\mathcal{G}}_2^2$. While standard DropOut methods [35, 30] repeat such a process to drop components temporally along the mini-batch iterations, our method merges them to *efficiently* update *all* the feature components at one iteration.

In addition, we apply the discriminative feature grouping which is more sophisticated than random one similar to the standard DropOut [35]. In that sense, adversarial DropOut [30] could be closely related to ours as the method adversarially drops the prominent (important) feature components. It, however, might deteriorate stability in training due to the lack of the prominent features throughout the training, thus requiring the drop-out ratio to be carefully tuned. Actually speaking, in the experiment on Food-101 (Sec. 4.1), the adversarial DropOut failed to properly train ResNet50 by the drop-out ratio of $p = 0.05$[1]~ 0.01, while $p = 0.001$ produces almost the same performance (error rate $19.58\%_{\pm 1.06}$) as the standard softmax loss (error rate $19.67\%_{\pm 0.48}$); therefore, we apply $p = 0.005$ in the experiments. In contrast, the proposed method is based on the adversarial training in terms of feature *grouping*, not *dropping* [30], which enables us to adversarially train networks efficiently and stably by exploiting all the feature components via grouping.

[1] In [30], $p = 0.05$ is applied to train the smaller network.



(a) Ours (b) DropOut [35] (c) adv. DropOut [30]

Figure 2. Comparison to DropOut methods. For simplicity, we regard the feature components of $x_4 \sim x_6$ as discriminative (in ours) and prominent (in [30]); note that those two characteristics are not necessarily shared in the identical components due to the difference criteria of ours and the adversarial DropOut [30].

3.3. Gradient-based update

The back-propagation in an end-to-end learning is evoked by the gradient of a loss function. The standard and group softmax losses provide the following derivatives,

$$\frac{\partial \ell_1}{\partial z_{jc}} = \mathtt{p}_c(\bar{\boldsymbol{z}}) - \mathbb{I}[c = y], \tag{16}$$

$$\frac{\partial \ell_G^I}{\partial z_{jc}} = \mathtt{p}_c(G\bar{\boldsymbol{z}}^g) - \mathbb{I}[c = y], \quad \frac{\partial \ell_G^{II}}{\partial z_{jc}} = \mathtt{p}_c(\bar{\boldsymbol{z}}^g) - \mathbb{I}[c = y], \tag{17}$$

where $\mathbb{I}$ is an indicator function and $\bar{\boldsymbol{z}}^g = \{\bar{z}_c^{\mathcal{G}_G^g}\}_{c=1}^C \in \mathbb{R}^C$ and the group index g is such that $j \in \mathcal{G}_G^g$. These derivatives are similarly formulated based on the difference between the posterior probability $\mathtt{p}_c$ and the ground-truth label $\mathbb{I}(c = y)$. Thus, the training procedure and techniques developed for the standard softmax loss are directly applied to the proposed losses.

Through our gradient-based updating, all the feature components can effectively gain the discriminative power due to the factorized form of the gradient (17) via grouping. Namely, the less discriminative feature components enjoy the larger update since the posterior $\mathtt{p}_c(\bar{\boldsymbol{z}}^g)$ of the less discriminative group $\mathcal{G}_G^g$ is different from the ground-truth one $\mathbb{I}[c = y]$. On the other hand, those components coupled with the highly discriminative ones in the softmax loss receive less update since the discriminative feature components dominate the posterior leading to $\mathtt{p}_c(\bar{\boldsymbol{z}}) \approx \mathbb{I}[c = y]$ in (16). Therefore, the proposed feature grouping (Sec. 2.3) separates the less discriminative feature components from the highly discriminative ones to fairly improve discriminativity of all the features components.

3.4. Computation cost

The group softmax loss is based on the partial inner-product between the feature vector $\boldsymbol{x}$ and the classifier weights $\boldsymbol{W}$, and thus the computation cost for logits is the

same as that of the inner product in the standard softmax loss, in disregard of the bias addition which requires negligible extra cost of GC operation (addition) in the group softmax loss. The difference regarding computation cost is mainly in the process of discriminative feature grouping and computing softmax function. The feature grouping, however, is performed in a computationally efficient manner by sorting the score λ_j (9) which is also efficiently computed. The group softmax loss requires G-times softmax computation while the standard softmax loss performs only one computation, though the difference of the computation costs is negligible.

4. Experimental results

We evaluate the performance of CNNs trained by the proposed loss on image classification tasks. In the experiments, all the CNNs are trained by applying SGD with the batch size of 256, the momentum of 0.9, the weight decay of 0.0001, and the initial learning rate of 0.1 which is divided by 10 at some epochs; the details are shown in the supplementary material. The performance is measured by single-crop top-1 error rate (%).

4.1. Ablation study

We apply the loss to train ResNet50 [15] on Food-101 dataset [1] for analyzing the method in an ablation manner; ResNet50 produces 2048-dimensional feature vector $\boldsymbol{x}$, i.e., $D = 2048$. Food-101 [1] contains images of 101 food categories, each of which comprises 750 training and 250 test images, to provide 101,000 food images in total; this is regarded as a middle-scale dataset. While the test images are *clean* through manual review, the training samples contain realistic issues regarding such as intense colors and wrong labels. Thus, we can say that Food-101 dataset reflects the realistic setting in terms of both the number of training samples and quality of annotation/images. ResNet50 is trained/evaluated on the predefined training/test set three times with different initial random seeds.

Number of groups G. Our loss form (12, 13) integrates multiple group softmax losses of which the numbers of groups G are gradually increased to hierarchically incorporate various grouping resolution. Table 1a shows the performance results regarding the number of partition G, in comparison with the standard softmax loss which is realized by $\mathbb{G} = \{1\}$ in (12, 13). The grouped loss either of type-I (7) or type-II (8) favorably improves performance; especially in this Food-101 classification, type-II loss is superior to type-I. As discussed in Sec. 3, type-II is a further upper bound on type-I, providing higher (adversarial) regularization to cope with the smaller number of training samples by mitigating over-fitting; type-I loss, the tighter bound on the softmax loss, performs well on the rather larger-scale dataset as shown in the later. The performance is further improved by integrating multiple grouping; we apply $\mathbb{G} = \{1, 2\}$ for type-I and $\mathbb{G} = \{1, 2, 4\}$ for type-II in the following experiments.

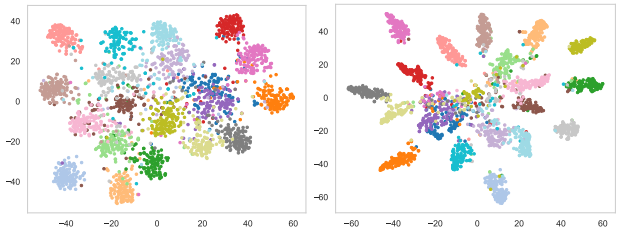
Table 1. Performance results (error rate %) on Food-101 by ResNet-50; our loss form is given in (12, 13).

(a) Various numbers of feature groups

Loss	# of groups	Error (%)
Softmax, ℓ_1	$\mathbb{G} = \{1\}$	19.67±0.48
Type-I, ℓ_G^I	$\mathbb{G} = \{1, 2\}$	**18.23**±0.27
Type-I, ℓ_G^I	$\mathbb{G} = \{1, 2, 4\}$	18.43±0.31
Type-II, ℓ_G^{II}	$\mathbb{G} = \{1, 2\}$	17.45±0.69
Type-II, ℓ_G^{II}	$\mathbb{G} = \{1, 2, 4\}$	**16.10**±0.29
Type-II, ℓ_G^{II}	$\mathbb{G} = \{1, 2, 4, 8\}$	16.23±0.25

(b) Various types of feature grouping

Grouping	Type-I $\mathbb{G} = \{1, 2\}$	Type-II $\mathbb{G} = \{1, 2, 4\}$
Discriminative	**18.23**±0.27	**16.10**±0.29
Random	19.63±0.50	17.70±0.69
Pre-fixed	19.53±1.28	18.18±0.96

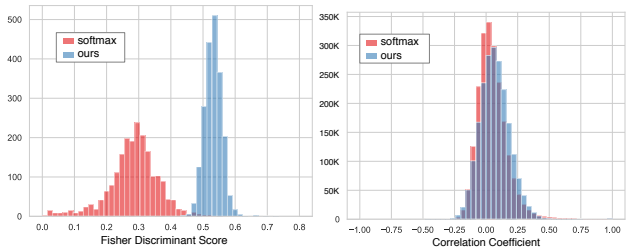


(a) Softmax loss (b) Ours (type-II)

Figure 3. Feature distribution on Food-101. Class categories are indicated by colors. Best viewed in color.

Type of grouping. The discriminative feature grouping (Sec. 2.3) is compared with the naive approaches that divide the whole indexes $\mathcal{G}_1$ in *random* and *pre-fixed* manners; the whole set $\mathcal{G}_1$ is *randomly* split into G groups at every mini-batch while the *pre-fixed* one uses the constant grouping $\{\mathcal{G}_G^i\}_{i=1}^G$ throughout the training. Table 1b demonstrates that the discriminative grouping outperforms the others. The naive approaches are problematic in completely ignoring discriminativity of feature components and fail to expose non-discriminative features to training. In contrast, our grouping enhances discriminativity of all the features by revealing (non-)discriminative components (Fig. 1).

Feature analysis. The distribution of the feature vector $\boldsymbol{x} \in \mathbb{R}^{2048}$ is shown in Fig. 3 by applying t-SNE [36] to test samples randomly drawn from the first 20 classes. We apply type-II loss in comparison to the standard softmax loss. Compared to the feature distribution learned by the softmax



(a) Discriminativity (b) Diversity

Figure 4. Distribution regarding two characteristics of feature components.

loss (Fig. 3a), our loss provides the discriminative distribution with clear class separability (Fig. 3b). We then measure the discriminativity of feature components by means of Fisher discriminant score [11]; the score ranges from 0 to 1 and the higher score means the higher discriminativity. Fig. 4a shows the distribution of the Fisher scores computed on the respective feature components of $\boldsymbol{x} \in \mathbb{R}^{2048}$. The proposed loss endows all the feature components with high discriminative power in contrast to the softmax loss which provides lower discriminativity; only the limited number of feature components gain the higher discriminative power by the softmax loss. On the other hand, the diversity (or redundancy) of the feature components is also measured by correlation coefficients among components. While redundant feature components exhibit higher correlation with the others, the diverse features are less correlated, producing coefficients close to zero. Fig. 4b shows that both the softmax loss and ours favorably produce features of high diversity, exhibiting less correlation among feature components; the two distributions are heavily overlapped around zero. These results validate that our loss is capable of learning discriminative and diverse features across *all* components.

Comparison. Our loss is compared with the other methods related to loss, the softmax variants [26, 20, 4, 5] and the regularization methods [39, 35, 30], on various scenarios; ResNet50 [15] and VGG16 [34] with batch-norm [18] are trained from scratch and fine-tuned. Note that for fair comparison, DropOut [35, 30] is applied only at the penultimate layer and backbone CNNs contain no DropOut. To report the performance results, we implemented those comparison methods by utilizing the codes that authors provide and properly tuned hyper-parameters based on suggestion in the papers/codes; for example, as discussed in Sec. 3.2, the drop-out ratio of adversarial DropOut [30] is set to $p = 0.005$. The performance comparison in Table 2 demonstrates that the proposed method is superior to the others. The comparison methods except for DropOut [35, 30] are built on the inner product of the whole feature vector $\boldsymbol{x}$ as in the softmax loss, and thus are vulnerable to the issue regarding the discriminative feature learning (Sec. 2.2&Fig. 1a). In contrast, our methods exploit all

Table 2. Performance comparison (error rate %) of various methods on Food-101.

	ResNet50 ($D=2048$)		VGG16 ($D=4096$)	
Loss	From scratch	Fine tune	From scratch	Fine tune
Softmax	19.67±0.48	12.55	17.86±0.22	13.09
L-Softmax [26]	18.81±0.83	12.48	17.14±0.26	12.01
SigSoftmax [20]	21.12±0.60	13.09	18.15±0.14	13.78
Noisy Softmax [4]	18.64±0.21	12.61	17.49±0.53	12.96
Virtual Softmax [5]	18.25±0.32	15.49	20.78±2.44	16.80
CenterLoss [39]	17.98±0.09	12.95	16.26±0.41	13.35
DropOut ($p=0.2$) [35]	18.77±0.68	12.49	17.84±0.35	13.28
DropOut ($p=0.5$) [35]	18.96±0.42	12.53	19.15±0.96	12.92
Adv. DropOut [30]	19.31±0.94	12.30	18.39±0.45	12.41
Ours, Type-I	18.23±0.27	12.28	16.62±0.12	12.16
Ours, Type-II	**16.10**±0.29	**11.88**	**15.33**±0.22	**11.78**

feature components in the group softmax loss through discriminative grouping to encourage feature learning on non-discriminative components, thereby favorably outperforming those comparison methods as well as DropOut [35, 30] which randomly/adversarially drops some features.

In the fine-tuning scenario, CNNs are first pre-trained on ImageNet by the standard softmax loss and then the identical pre-trained CNNs are fine-tuned to Food-101 by the methods (losses) for fair comparison. In this case, CNNs are already endowed with the discriminative feature representation through pre-training on the large-scale ImageNet dataset, and thus the superiority of the proposed losses is slightly smaller compared to the case of the training from scratch. It, however, can be seen that the discriminative feature learning by our loss still contributes to performance improvement. It might be noteworthy that the proposed method can be combined with those methods except for DropOut; the softmax variants [26, 20, 4, 5] can replace the softmax function in (7, 8) and the regularization term [39] can be simply added to our loss. It is also noteworthy that the proposed method tackles data-hunger issue of neural networks in a form of loss and thous would be cooperative with data-augmentation techniques which work on the problem by touching input data; our future work includes such a practical combination with those related methods.

4.2. ImageNet dataset

We utilize the large-scale ImageNet for further analyzing the proposed loss from various perspectives.

Number of classes. We analyze performance on various numbers of classes, by randomly picking up the sub-set of $C \in \{100, 200, 500, 1000\}$ categories from the whole 1000 categories to construct C-class classification task; note that each class contains roughly 1000 training samples. As

Table 3. Performance results (error rate %) on ImageNet.

(a) Number of classes (ImageNet subset)				
# of classes	100	200	500	1000
# of total samples	127K	258K	639K	1281K
Softmax	15.4	18.1	20.2	23.5
Type-I	14.6	16.3	**19.0**	**23.3**
Type-II	**13.0**	**15.6**	19.1	24.1

(b) Various CNNs (ImageNet full-set)			
Loss	ResNet50	ResNeXt50	VGG16
Softmax	23.45	22.42	25.04
Type-I	**23.31**	**22.09**	**24.26**
Type-II	24.13	22.16	24.65

Table 4. Performance results on smaller-scale datasets.

	Caltech-256 [14]		SUN-397 [40]	
Loss	ResNet50	VGG16	ResNet50	VGG16
Softmax	48.4±0.5	51.3±2.0	55.7±0.6	58.8±1.4
L-Softmax [26]	43.0±1.6	44.5±1.0	56.3±1.6	53.1±0.7
SigSoftmax [20]	48.2±2.2	54.6±1.7	56.9±0.5	59.6±1.2
Noisy Softmax [4]	43.2±0.7	51.2±1.2	56.0±1.4	57.5±0.8
Virtual Softmax [5]	46.9±0.5	54.5±2.8	56.9±0.5	56.4±0.2
CenterLoss [39]	53.5±1.3	44.6±1.2	60.6±1.0	52.9±0.4
DropOut ($p=0.2$) [35]	44.9±0.6	50.5±0.3	56.5±0.3	57.2±0.7
DropOut ($p=0.5$) [35]	44.7±1.7	47.1±0.6	54.8±0.7	55.5±0.9
Adv. DropOut [30]	43.8±0.8	45.4±1.7	53.4±0.9	54.0±0.4
Ours, Type-I	42.8±0.6	46.1±0.8	52.9±0.5	52.2±0.9
Ours, Type-II	**37.5**±0.2	**41.8**±0.5	**50.4**±0.5	**50.0**±0.2
Hand-craft [32]	42.7±0.2		52.8±0.2	

shown in Table 3a, the type-II loss outperforms the type-I on the smaller number of classes, since type-II is further upper bound on type-I to effectively cope with smaller-scale data. On the other hand, as the number of classes increases, the type-I loss exhibits superiority, being competitive to the softmax loss on the full ImageNet dataset ($C = 1000$). Diverse training samples contribute to training discriminative feature representation and the tighter upper-bound loss (type-I) effectively leverages them to improve performance. Therefore, the type-I loss is rather preferable for the larger-scale data, while the type-II loss is quite effective on the small/middle-scale one; further analyses are conducted in the supplementary to support this claim.

Comparison on various CNNs. Then, the type-I loss is tested on the full set of ImageNet dataset by using three deep CNNs of ResNet50 [15], ResNeXt50 [41] and VGG16 [34]. Table 3b shows that the type-I loss contributes to performance improvement of various CNNs trained on the large-scale training samples. In particular, the proposed methods enhancing discriminative feature representation work favorably on the rather complicated CNNs of ResNeXt50 [41] which exploits wide feature channels and VGG16 [34] which contains huge number of parameters.

4.3. Smaller-scale datasets

Finally, we apply the proposed losses to train CNNs on the smaller-scale datasets to which the hand-crafted methods [32] have been successfully applied. For fair comparison, we train the CNNs of ResNet50 [15] and VGG16 [34] from scratch on the following two datasets.

Caltech-256 [14] is composed of 30,607 images in 256 object categories, each of which contains at least 80 images. According to the standard protocol, we randomly pick up 60 training images per category and use the rest for test. We report the averaged classification error rate over three trials.

SUN-397 [40] contains roughly 100K images of 397 scene categories covering as many of visual world scenes as possible. Following the protocol of [40], we used 50 training and 50 test samples per category and measured the classification error rate averaged over three trials.

Table 4 presents the performance comparison results of various methods including the Fisher kernel [32], a representative hand-crafted method. The CNNs trained by the standard softmax loss are inferior even to the hand-crafted method on these smaller-scale datasets. The softmax loss fails to learn generalizable feature representation on these small-scale datasets, while the other comparison methods improve the feature representation such as by incorporating large-margin criterion, regularization and perturbation. On the other hand, our loss, especially the type-II, enables the CNNs to outperform the hand-crafted method. It should be noted that the proposed losses are simply applied to the CNNs as in the other experiments without introducing tricks tailored for these small-scale datasets. Thus, the proposed loss would contribute to further enlarging the applicability of CNNs to various datasets including smaller-scale one.

5. Conclusion

We have proposed the group softmax loss with discriminative feature grouping method for encouraging discriminative representation across all the feature components. Through partitioning feature components into groups, we derive two types of group losses and the effective feature grouping to expose non-discriminative components to feature learning. These methods are tightly coupled toward learning discriminative features even on small/middle-scale data, while theoretically providing adversarial training. In the experiments on supervised image classification tasks, the proposed loss is thoroughly evaluated and exhibits favorable performance in comparison to the other methods.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, pages 446–461, 2014.

[2] Guillaume Bouchard. Efficient bounds for the softmax function and applications to approximate inference in hybrid models. In *NIPS 2007 Workshop for Approximate Bayesian Inference in Continuous/Hybrid Systems*, 2007.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018.

[4] Binghui Chen, Weihong Deng, and Junping Du. Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation. In *CVPR*, pages 4021–4030, 2017.

[5] Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. In *NeurIPS*, page 1946–1956, 2018.

[6] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64, 2014.

[7] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv,* 1708.04552, 2017.

[8] Thanh-Toan Do, Toan Tran, Ian Reid, Vijay Kumar, Tuan Hoang, and Gustavo Carneiro. A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning. In *CVPR*, pages 10404–10413, 2019.

[9] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[10] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019.

[11] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[12] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *CVPR*, pages 2946–2953, 2013.

[13] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv,* 1412.6572, 2014.

[14] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, Caltech, 2007.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv,* 1704.04861, 2017.

[17] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Journal of Machine Learning Research*, 37:448–456, 2015.

[19] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Journal on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[20] Sekitoshi Kanai, Yasuhiro Fujiwara, Yuki Yamanaka, and Shuichi Adachi. Sigsoftmax: Reanalysis of the softmax bottleneck. In *NeurIPS*, page 284–294, 2018.

[21] Benjamin Klein and Lior Wolf. End-to-end supervised product quantization for image search and retrieval. In *CVPR*, pages 5041–5050, 2019.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[23] Xiaoxu Li, Dongliang Chang, Tao Tian, and Jie Cao. Large-margin regularized softmax cross-entropy loss. *IEEE Access*, 7:19572–19578, 2019.

[24] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *CVPR*, pages 10991–11000, 2020.

[25] Baoyuan Liu, Fereshteh Sadeghi, Marshall Tappen, Ohad Shamir, and Ce Liu. Probabilistic label trees for efficient large scale image classification. In *CVPR*, pages 843–850, 2013.

[26] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, pages 507–516, 2016.

[27] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *ECCV*, pages 132–149, 2018.

[28] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, pages 246–252, 2005.

[29] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368, 2017.

[30] Sungrae Park, Jun-Keon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *AAAI*, pages 3917–3924, 2018.

[31] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, pages 6517–6525, 2017.

[32] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[33] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, pages 118–126, 2015.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[35] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[37] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[38] Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. In *CVPR*, pages 9117–9126, 2018.

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

[40] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[41] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017.

[42] Tan Yu, Junsong Yuan, Chen Fang, and Hailin Jin. Product quantization network for fast image retrieval. In *ECCV*, pages 186–201, 2018.

[43] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv,* 1708.04896, 2017.

[44] Chengxu Zhuang, Alex Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6002–6012, 2019.