Supplementary Material for
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

A. Detailed discussion about *adversariality*

In Sec.3.1, the proposed method is analyzed from an adversarial viewpoint. We here present further detailed discussion on it by showing the proof of Proposition 1 regarding the group softmax loss as well as explaining how the discriminative feature grouping approximately maximizes the group softmax loss.

Notations

- C : Number of classes.
- D : Number of feature elements.
- G : Number of feature groups.
- $y \in \{1, \cdots, C\}$: The ground-truth class label.
- $\{\mathcal{G}_G^i\}_{i=1}^G$: G feature groups; $\mathcal{G}_G^i \subset \{1, \cdots, D\}, \bigcup_{i=1}^G \mathcal{G}_G^i = \{1, \cdots, D\}, \mathcal{G}_G^i \cap \mathcal{G}_G^{i'} = \emptyset \, \forall i \neq i'$.
- z_{jc} : Ingredient logit at the j-th feature element for the c-th class; $j \in \{1, \cdots, D\}$, $c \in \{1, \cdots, C\}$.
- $\bar{z}_c = \sum_{j=1}^D z_{jc}$: (Whole) logit for the c-th class.
- $\bar{z}_c^i \triangleq \bar{z}_c^{\mathcal{G}_G^i} = \sum_{j \in \mathcal{G}_G^i} z_{jc}$: Group logit of the i-th group $\mathcal{G}_G^i$ for the c-th class.

  Note that the whole logit is decomposed into group and ingredient logits as follows.
$$\bar{z}_c = \sum_{j=1}^D z_{jc} = \sum_{i=1}^G \bar{z}_c^i. \tag{A.1}$$

A.1. Proof of Proposition 1

We prove Proposition 1 as follows.

Proposition 1. *Given the feature grouping $\{\mathcal{G}_G^i\}_{i=1}^G$, the softmax loss ℓ_1 is upper bounded by the group softmax losses ℓ_G^I, ℓ_G^{II} as*
$$\underbrace{-\log\left[\frac{\exp(\bar{z}_y)}{\sum_{c=1}^C \exp(\bar{z}_c)}\right]}_{\ell_1(\bar{\boldsymbol{z}},y)} \leq \underbrace{-\frac{1}{G}\sum_{i=1}^G \log\left[\frac{\exp(G\bar{z}_y^i)}{\sum_{c=1}^C \exp(G\bar{z}_c^i)}\right]}_{\ell_G^I(\boldsymbol{Z},y;\mathcal{G}_G)} \leq \underbrace{-\sum_{i=1}^G \log\left[\frac{\exp(\bar{z}_y^i)}{\sum_{c=1}^C \exp(\bar{z}_c^i)}\right]}_{\ell_G^{II}(\boldsymbol{Z},y;\mathcal{G}_G)}. \tag{A.2}$$

Proof. (A.2) is equivalent to the following relationship among the product form of softmax probabilities,
$$\mathrm{p}_y(\bar{\boldsymbol{z}}) = \frac{\exp(\bar{z}_y)}{\sum_{c=1}^C \exp(\bar{z}_c)} \geq \left\{\prod_{i=1}^G \frac{\exp(G\bar{z}_y^i)}{\sum_{c=1}^C \exp(G\bar{z}_c^i)}\right\}^{\frac{1}{G}} \geq \prod_{i=1}^G \frac{\exp(\bar{z}_y^i)}{\sum_{c=1}^C \exp(\bar{z}_c^i)}. \tag{A.3}$$

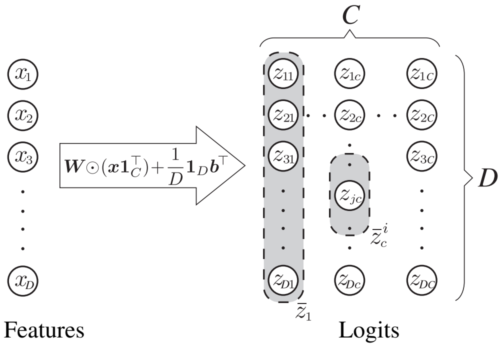

Figure A.1. Features $\boldsymbol{x}$ are mapped into logits. The ingredient logits in the gray-colored rectangle are summed up to produce the whole logit $\bar{z}_1$ and the group logit $\bar{z}_c^i$.

Based on the relationship (A.1), all the numerators in (A.3) are identical,

$$\exp(\bar{z}_y) = \exp\left(\frac{1}{G}\sum_{i=1}^{G} G\bar{z}_y^i\right) = \exp\left(\sum_{i=1}^{G}\bar{z}_y^i\right). \tag{A.4}$$

Therefore, it is enough to prove the following inequalities, regarding the denominators,

$$\sum_{c=1}^{C}\exp(\bar{z}_c) \leq \left\{\prod_{i=1}^{G}\sum_{c=1}^{C}\exp(G\bar{z}_c^i)\right\}^{\frac{1}{G}} \leq \prod_{i=1}^{G}\sum_{c=1}^{C}\exp(\bar{z}_c^i). \tag{A.5}$$

- The first inequality in (A.5)
By using (A.1), we can transform the sum of exponentials as

$$\sum_{c=1}^{C}\exp(\bar{z}_c) = \sum_{c=1}^{C}\exp\left(\frac{1}{G}\sum_{i=1}^{G}G\bar{z}_c^i\right) = \sum_{c=1}^{C}\left\{\prod_{i=1}^{G}\exp(G\bar{z}_c^i)\right\}^{\frac{1}{G}}. \tag{A.6}$$

Here, for the positive variables $\boldsymbol{\xi} = \{\xi_i\}_{i=1}^{G} \in \mathbb{R}_+^G$, we introduce the geometric mean function of

$$f(\boldsymbol{\xi}) = \left(\prod_{i=1}^{G}\xi_i\right)^{\frac{1}{G}}. \tag{A.7}$$

By using the geometric mean f and (A.6), the first inequality in (A.5) is rewritten into

$$\sum_{c=1}^{C} f(\boldsymbol{\xi}_c) \leq f\left(\sum_{c=1}^{C}\boldsymbol{\xi}_c\right), \tag{A.8}$$

where $\boldsymbol{\xi}_c = \{\exp(G\bar{z}_c^i)\}_{i=1}^{G} \in \mathbb{R}_+^G$. The concavity of the geometric mean function f with Jensen's inequality provides

$$\frac{1}{C}\sum_{c=1}^{C} f(\boldsymbol{\xi}_c) \leq f\left(\frac{1}{C}\sum_{c=1}^{C}\boldsymbol{\xi}_c\right) = \left(\prod_{i=1}^{G}\frac{1}{C}\sum_{c=1}^{C}\xi_{ci}\right)^{\frac{1}{G}} = \left(\frac{1}{C^G}\prod_{i=1}^{G}\sum_{c=1}^{C}\xi_{ci}\right)^{\frac{1}{G}} \tag{A.9}$$

$$= \frac{1}{C} f\left(\sum_{c=1}^{C}\boldsymbol{\xi}_c\right), \tag{A.10}$$

where the equality holds when $\boldsymbol{\xi}_c = \boldsymbol{\xi}_{c'},\ \forall c \neq c'$. This is obviously equivalent to the inequality (A.8).

- The second inequality in (A.5)
The inequality is equivalent to

$$\prod_{i=1}^{G}\sum_{c=1}^{C}\exp(G\bar{z}_c^i) \leq \prod_{i=1}^{G}\left\{\sum_{c=1}^{C}\exp(\bar{z}_c^i)\right\}^G. \tag{A.11}$$

Considering $\exp(\cdot) > 0$, the following holds

$$\sum_{c=1}^{C}\exp(G\bar{z}_c^i) = \sum_{c=1}^{C}\left\{\exp(\bar{z}_c^i)\right\}^G \leq \left\{\sum_{c=1}^{C}\exp(\bar{z}_c^i)\right\}^G, \tag{A.12}$$

where the equality holds by $G = 1$. This straightforwardly leads to the inequality (A.11). □

A.2. Discriminative feature grouping

We then explain how the discriminative feature grouping of

$$\tilde{\mathcal{G}}_G^i = \{j|\tau_{i-1} < \lambda_j \leq \tau_i\} \wedge |\tilde{\mathcal{G}}_G^i| = \frac{D}{G}, \forall i \in \{1, \cdots, G\}, \tag{A.13}$$

$$\text{where } \tau_{i-1} < \tau_i, \tau_0 = -\infty, \tau_G = \infty, \;\; \lambda_j = z_{jy} - \frac{1}{C-1}\sum_{c\neq y} z_{jc}, \tag{A.14}$$

approximately maximizes the group softmax loss w.r.t grouping $\{\mathcal{G}_G^i\}_{i=1}^G$. Such an optimization problem is written by

$$\max_{\{\mathcal{G}_G^i\}_{i=1}^G} -\sum_{i=1}^{G}\log\left[\frac{\exp(\bar{z}_y^{\mathcal{G}_G^i})}{\sum_{c=1}^{C}\exp(\bar{z}_c^{\mathcal{G}_G^i})}\right] \Leftrightarrow \max_{\{\mathcal{G}_G^i\}_{i=1}^G}\sum_{i=1}^{G}\mathtt{sp}\left[\bar{z}_y^{\mathcal{G}_G^i} - \log\left\{\sum_{c\neq y}\exp(\bar{z}_c^{\mathcal{G}_G^i})\right\}\right], \tag{A.15}$$

where we use the function of $\mathtt{sp}(x) = \log\{1+\exp(-x)\}$ like softplus function, and explicitly show the dependency of group logits on group $\mathcal{G}_G^i$. Though we consider (A.15) using the type-II loss, the following discussion also holds for the type-I by simply replacing $\bar{z}_c^{\mathcal{G}_G^i}$ with $G\bar{z}_c^{\mathcal{G}_G^i}$. (A.15) is a combinatorial optimization problem being computationally inefficient, mainly because the ingredient logits are mixed up across the indexes $j \in \mathcal{G}_G^i$ via the non-linear function of log-sum-exp. Thus, instead of directly solving (A.15), we relax it especially in terms of log-sum-exp function and show that the discriminative grouping $\{\tilde{\mathcal{G}}_G^i\}_{i=1}^G$ in (A.13) maximizes the relaxed problem, as follows.

The log-sum-exp term in (A.15) is lower-bounded by the simple mean as

$$\log\left\{\sum_{c\neq y}\exp(\bar{z}_c^{\mathcal{G}_G^i})\right\} > \frac{1}{C-1}\sum_{c\neq y}\bar{z}_c^{\mathcal{G}_G^i}, \tag{A.16}$$

because

$$\left\{\sum_{c\neq y}\exp(\bar{z}_c^{\mathcal{G}_G^i})\right\}^{C-1} > \prod_{c\neq y}\exp(\bar{z}_c^{\mathcal{G}_G^i}) = \exp\left(\sum_{c\neq y}\bar{z}_c^{\mathcal{G}_G^i}\right), \tag{A.17}$$

where $\sum_{c\neq y}\exp(\bar{z}_c^{\mathcal{G}_G^i}) > \exp(\bar{z}_{c'}^{\mathcal{G}_G^i}), \forall c' \neq y$. Thus, (A.15) can be roughly relaxed into the *feature component*-wise (j) form as

$$\text{(A.15)} > \max_{\{\mathcal{G}_G^i\}_{i=1}^G}\sum_{i=1}^{G}\mathtt{sp}\left[\bar{z}_y^{\mathcal{G}_G^i} - \frac{1}{C-1}\left(\sum_{c\neq y}\bar{z}_c^{\mathcal{G}_G^i}\right)\right] \tag{A.18}$$

$$= \max_{\{\mathcal{G}_G^i\}_{i=1}^G}\sum_{i=1}^{G}\mathtt{sp}\left[\sum_{j\in\mathcal{G}_G^i}\left\{z_{jy} - \frac{1}{C-1}\left(\sum_{c\neq y}z_{jc}\right)\right\}\right] = \max_{\{\mathcal{G}_G^i\}_{i=1}^G}\sum_{i=1}^{G}\mathtt{sp}\left[\sum_{j\in\mathcal{G}_G^i}\lambda_j\right], \tag{A.19}$$

where we use the discriminativity score λ_j in (A.14). Besides, for computational efficiency, we simply consider the *equal* partitioning that produces groups containing equal numbers of components; $\mathcal{U}_G = \{\{\mathcal{G}_G^i\}_{i=1}^G \mid \forall i, |\mathcal{G}_G^i| = \frac{D}{G}\}$. The optimization problem (A.15) is finally relaxed into the tractable form of

$$\max_{\{\mathcal{G}_G^i\}_{i=1}^G \in \mathcal{U}_G} \sum_{i=1}^G \mathtt{sp}\left[\bar{\lambda}^{\mathcal{G}_G^i}\right], \text{ where } \bar{\lambda}^{\mathcal{G}_G^i} = \sum_{j \in \mathcal{G}_G^i} \lambda_j. \tag{A.20}$$

Lemma 1. *The optimum partitioning for* (A.20) *holds the following condition;*

$$\lambda_j > \lambda_{j'},\ \{\forall j \in \mathcal{G}_G^i, \forall j' \in \mathcal{G}_G^{i'} | \, \bar{\lambda}^{\mathcal{G}_G^i} > \bar{\lambda}^{\mathcal{G}_G^{i'}}\}. \tag{A.21}$$

Proof. If the other partitioning not satisfying (A.21), that is, the partitioning which contains the groups $\mathcal{G}_G^i, \mathcal{G}_G^{i'}$ such that

$$\lambda_j < \lambda_{j'},\ \exists j \in \mathcal{G}_G^i, \exists j' \in \mathcal{G}_G^{i'}, \tag{A.22}$$

is the optimizer of (A.20), it causes the following contradiction. We consider to swap the indexes of j and j' of (A.22) over the two groups of $\mathcal{G}_G^i$ and $\mathcal{G}_G^{i'}$ as

$$\hat{\mathcal{G}}_G^i = \{\mathcal{G}_G^i \backslash j\} \cup j',\ \hat{\mathcal{G}}_G^{i'} = \{\mathcal{G}_G^{i'} \backslash j'\} \cup j, \tag{A.23}$$

where $\backslash$ indicates the operator to remove the (right-hand-side) element from the (left-hand-side) set. So created groups $\hat{\mathcal{G}}_G^i$ and $\hat{\mathcal{G}}_G^{i'}$ can increase the score by

$$\mathtt{sp}\left[\bar{\lambda}^{\mathcal{G}_G^i}\right] + \mathtt{sp}\left[\bar{\lambda}^{\mathcal{G}_G^{i'}}\right] < \mathtt{sp}\left[\bar{\lambda}^{\hat{\mathcal{G}}_G^i}\right] + \mathtt{sp}\left[\bar{\lambda}^{\hat{\mathcal{G}}_G^{i'}}\right], \tag{A.24}$$

where

$$\bar{\lambda}^{\mathcal{G}_G^i} + \bar{\lambda}^{\mathcal{G}_G^{i'}} = \bar{\lambda}^{\hat{\mathcal{G}}_G^i} + \bar{\lambda}^{\hat{\mathcal{G}}_G^{i'}},\ \ \bar{\lambda}^{\hat{\mathcal{G}}_G^{i'}} < \bar{\lambda}^{\mathcal{G}_G^{i'}} < \bar{\lambda}^{\mathcal{G}_G^i} < \bar{\lambda}^{\hat{\mathcal{G}}_G^i}. \tag{A.25}$$

This is because, given the constant scalar s, the sum of two functions $q(\alpha, \beta) = \mathtt{sp}(\alpha) + \mathtt{sp}(\beta)$ where $\alpha + \beta = s$ provides

$$q(\alpha, \beta) < q(\hat{\alpha}, \hat{\beta}),\ \{\forall(\alpha, \beta), \forall(\hat{\alpha}, \hat{\beta}) | \, \hat{\beta} < \beta < \alpha < \hat{\alpha}\}, \tag{A.26}$$

due to $\frac{d}{d\alpha} q(\alpha, s - \alpha) = \frac{1}{1+\exp(s-\alpha)} - \frac{1}{1+\exp(\alpha)} > 0$ by $\alpha > s - \alpha = \beta$. (A.24) contradicts to that the partitioning including (A.22) maximizes (A.20). Thus, the optimum grouping for (A.20) satisfies (A.21). □

Based on the lemma, the optimum grouping of (A.20) is uniquely determined by *sorting* $\{\lambda_j\}_{j=1}^D$ and *equal partitioning* as shown in the discriminative feature grouping (A.13).

Based on these two discussions in Sec. A.1&A.2, the proposed method can be roughly viewed as adversarial training of

$$\min_{\boldsymbol{\theta}} \max_{\mathcal{G}} \ell_G(\boldsymbol{Z}(\boldsymbol{\theta}), y; \mathcal{G}), \tag{A.27}$$

where the loss is adversarially increased with respect to *grouping*.

B. Experiments

B.1. Training procedures

As described in Sec.4, all the CNNs that we use in the experiments are trained by applying SGD with the batch size of 256, the momentum of 0.9, the weight decay of 0.0001. The initial learning rate is set to 0.1, and is divided by 10 at the t-th epoch, $t \in \mathbb{T}$, during the T training epochs. Those epoch points $\mathbb{T}$ and the total number of training epochs T are determined based on the number of training samples and classes as shown in Table B.1(i). In the fine-tuning on Food-101 in Sec.4.1 (Table 2), the learning rate is initially set to 0.0045 and divided by 10 at the t-th epoch, $t \in \mathbb{T}$, as well; see Table B.1(ii). On all the datasets, we simply apply the standard preprocessing and data augmentation techniques applied to ImageNet classification [1]. The methods are implemented by PyTorch and tested on NVIDIA Tesla V100.

Table B.1. The number of training epoch T and the epochs $\mathbb{T}$ at which the learning rate is divided by 10. The numbers of classes C and training samples N are also shown.

(i) Training from scratch (initial learning rate: 0.1)

(i-1) Various datasets

Dataset	C	N	$\mathbb{T}$	T
ImageNet	1000	1281K	$\{30, 60, 90\}$	120
Food-101	101	75K	$\{60, 120, 150\}$	180
Caltech-256	256	15K	$\{120, 240, 270\}$	300
SUN-397	397	19K	$\{120, 240, 270\}$	300

(i-2) ImageNet subset (Sec.4.2)

Condition	C	N	$\mathbb{T}$	T
100 classes	100	127K	$\{60, 120, 150\}$	180
200 classes	200	258K	$\{60, 120, 150\}$	180
500 classes	500	639K	$\{30, 60, 90\}$	120
100 samples per class	1000	100K	$\{90, 180, 210\}$	240
200 samples per class	1000	200K	$\{90, 180, 210\}$	240
500 samples per class	1000	500K	$\{60, 120, 150\}$	180

(ii) Fine-tuning (initial learning rate: 0.0045) (Sec.4.1)

Dataset	C	N	$\mathbb{T}$	T
Food-101	101	75K	$\{30, 60, 90\}$	120

B.2. Additional results on ImageNet subsets

We supplement the experimental results on Sec.4.2 by showing detailed analysis in the case of smaller number of classes as well as additional experimental results regarding number of training samples.

Number of classes. The experimental result is shown in Table B.2a which is the same as Table 3a in the main manuscript. We analyze the results from the viewpoint of gradient-based updates.

The standard softmax loss updates the feature $\boldsymbol{x}$ by

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta \boldsymbol{W} \left[\mathtt{p}(\bar{\boldsymbol{z}}) - \boldsymbol{e}_y\right], \tag{B.1}$$

where $\mathtt{p}(\cdot) = [\mathtt{p}_1(\cdot), \cdots, \mathtt{p}_C(\cdot)]^\top \in \mathbb{R}^C$ is the C-dimensional posterior vector, $\boldsymbol{e}_y$ is a one-hot vector activating the y-th element, and η is a learning rate. In (B.1), the updating direction is restricted in the subspace of $\boldsymbol{W} \in \mathbb{R}^{D\times C}$ which is at most $\min(C, D)$ rank. On the other hand, our loss updates the feature $\boldsymbol{x}$ as

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta\, \mathtt{diag}[\{\boldsymbol{W}^i\}_{i=1}^G]\, \mathtt{vec}[\{\mathtt{p}(\bar{\boldsymbol{z}}^i) - \boldsymbol{e}_y\}_{i=1}^G], \tag{B.2}$$

where $\boldsymbol{W}^i = \{W_{jc}\}_{j\in\tilde{\mathcal{G}}_G^i, c=1}^C \in \mathbb{R}^{|\tilde{\mathcal{G}}_G^i|\times C}$ indicates the classifier weights of the i-th group $\tilde{\mathcal{G}}_G^i$ and `diag` stacks matrices along the diagonal, $\mathtt{diag}[\{\boldsymbol{W}^i\}_{i=1}^G] \in \mathbb{R}^{D\times CG}$, and `vec` stacks vectors vertically, $\mathtt{vec}[\{\mathtt{p}(\bar{\boldsymbol{z}}^i) - \boldsymbol{e}_y\}_{i=1}^G] \in \mathbb{R}^{CG}$. Thus, the updating direction in (B.2) lies in the direct sum of the subspaces by $\{\boldsymbol{W}^i\}_{i=1}^G$ which have the rank of $\min(CG, D)$, higher than $\min(C, D)$ of the updating (B.1) by the standard softmax loss.

Based on the above analysis, we can say that the proposed method is quite effective on the smaller number of classes (C) to more flexibly update the feature $\boldsymbol{x}$; it is empirically shown in Table B.2a.

Number of samples. To analyze the dependency on the number of training samples, we randomly pick up N training samples, $N \in \{100, 200, 500, 1000^*\}$, in each of the 1000 class categories; $N = 1000^*$ indicates the full ImageNet training set. As shown in Table B.2b, the type-II loss outperforms the type-I on the smaller number of samples since type-II which is further upper bound on type-I in (A.2) alleviates over-fitting more effectively. On the other hand, as the number of samples increases, the type-I loss exhibits superiority, being competitive to the softmax loss on the full ImageNet dataset ($N = 1000^*$), by leveraging plenty of training samples to effectively training discriminative feature representation.

Table B.2. Performance results (error rate %) on ImageNet subsets.

(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) Number of samples (ImageNet subset)

# of samples per class	100	200	500	1000*
# of total samples	100K	200K	500K	1281K
Softmax	49.3	40.3	30.2	23.5
Type-I	47.6	38.0	**28.8**	**23.3**
Type-II	**46.8**	**38.0**	29.3	24.1

Based on the above two experimental results in Table B.2, we can roughly say that the type-II loss effectively works on <500K training samples while the type-I favorably copes with >500K samples. The type-I loss, the tighter upper bound on the softmax loss, is rather preferable for the larger-scale data. On the other hand, the type-II loss is quite effective on the smaller number of training samples as well as the smaller number of classes.

References

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