Supplementary Material for: Direct-Sum Approach to Integrate Losses Via Classifier Subspace

Takumi Kobayashi[1,2]
takumi.kobayashi@aist.go.jp

[1] National Institute of Advanced Industrial Science and Technology
Tsukuba, Japan

[2] University of Tsukuba
Tsukuba, Japan

A Derivatives of Proto loss

The discussion in Sec. 2.3 holds for the Proto loss since the derivative of Proto loss is explicitly described as follows.

The Proto loss is formulated in

$$\text{Proto:}\ \ \ell_{Proto}(\{\boldsymbol{x}_i, y_i\}_{i=1}^n) = -\mathop{\mathrm{E}}_i \log \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i\backslash i}\|_2^2)}{\sum_{c=1}^C \exp(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{c\backslash i}\|_2^2)},\ \text{where}\ \boldsymbol{\mu}_{c\backslash i} = \mathop{\mathrm{E}}_{j\neq i|y_j=c} \boldsymbol{x}_j. \tag{i}$$

The loss gradient w.r.t $\boldsymbol{x}_i$ is given by

$$\begin{aligned}
\frac{\partial \ell_{Proto}}{\partial \boldsymbol{x}_i} = \frac{2}{n}\Bigg[&(\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i\backslash i}) - \mathop{\mathrm{E}}_{j\neq i|y_j=y_i} (\boldsymbol{x}_j - \boldsymbol{\mu}_{y_j\backslash j}) && \text{(ii)}\\
&- \sum_{\hat{c}=1}^C \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{\hat{c}\backslash i}\|_2^2)}{\sum_{c=1}^C \exp(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{c\backslash i}\|_2^2)} (\boldsymbol{x}_i - \boldsymbol{\mu}_{\hat{c}\backslash i}) && \text{(iii)}\\
&+ \sum_{j\neq i}^n \frac{1}{n_{y_i} - [\![y_j = y_i]\!]} \frac{\exp(-\|\boldsymbol{x}_j - \boldsymbol{\mu}_{y_i\backslash j}\|_2^2)}{\sum_{c=1}^C \exp(-\|\boldsymbol{x}_j - \boldsymbol{\mu}_{c\backslash j}\|_2^2)} (\boldsymbol{x}_j - \boldsymbol{\mu}_{y_i\backslash j}) \Bigg] && \text{(iv)}\\
&\in \mathtt{span}(\boldsymbol{X}), && \text{(v)}
\end{aligned}$$

where $[\![y_j = y_i]\!]$ produces 1 for $y_j = y_i$ and otherwise 0. This shows that the derivative lies in a subspace spanned by samples similarly to that of NCA loss.

B Projection onto classifier subspace

We validate the approximated form (8) of projection onto a classifier subspace $\mathtt{W} \in \mathbb{R}^{d\times \mathtt{rank}(\boldsymbol{W})}$ for the classifier weight $\boldsymbol{W} \in \mathbb{R}^{d\times C}$; while the classifier rank is C in most case, rank reduction $\mathtt{rank}(\boldsymbol{W}) < C$ could practically happen.

Let the classifier $\boldsymbol{W}$ be decomposed by $\boldsymbol{W} = \tilde{\mathtt{W}}\mathrm{diag}(\boldsymbol{\lambda})\mathtt{V}^\top$ via singular value decomposition, and then we have

$$\boldsymbol{W}(\boldsymbol{W}^\top\boldsymbol{W}+\varepsilon\mathtt{I})^{-1}\boldsymbol{W}^\top = \tilde{\mathtt{W}}\mathrm{diag}\left(\left\{\frac{\lambda_j}{\varepsilon+\lambda_j}\right\}_{j=1}^{C}\right)\tilde{\mathtt{W}}^\top \tag{vi}$$

$$\approx \tilde{\mathtt{W}}\mathrm{diag}\left(\{[\![\lambda_j > 0]\!]\}_{j=1}^{C}\right)\tilde{\mathtt{W}}^\top = \mathtt{W}\mathtt{W}^\top, \tag{vii}$$

where $\frac{\lambda}{\varepsilon+\lambda}$ smoothly approximates a step function $[\![\lambda > 0]\!]$. Thus, the parameter ε makes the computation of inverse matrix stable as well as controls smoothness of the approximation.

C Spectral sum of losses

In the spectral-sum loss (Sec. 3.4), we constructed a *pseudo* complementary space by means of soft weighting; it is described by the projection matrix $\tilde{\mathtt{W}}\mathrm{diag}(\mathbf{1}-\tilde{\boldsymbol{\lambda}}^p)\tilde{\mathtt{W}}^\top$. We can measure its overlapness with the classifier $\boldsymbol{W} = \tilde{\mathtt{W}}\mathrm{diag}(\boldsymbol{\lambda})\mathtt{V}^\top$ (9) by using the the spectral norm of

$$\|\boldsymbol{W}^\top\tilde{\mathtt{W}}\mathrm{diag}(\mathbf{1}-\tilde{\boldsymbol{\lambda}}^p)\tilde{\mathtt{W}}^\top\|_2 = \|\mathtt{V}\,\mathrm{diag}(\boldsymbol{\lambda})\tilde{\mathtt{W}}^\top\tilde{\mathtt{W}}\mathrm{diag}(\mathbf{1}-\tilde{\boldsymbol{\lambda}}^p)\tilde{\mathtt{W}}^\top\|_2 \tag{viii}$$

$$= \|\mathtt{V}\,\mathrm{diag}(\boldsymbol{\lambda}\odot(\mathbf{1}-\tilde{\boldsymbol{\lambda}}^p))\tilde{\mathtt{W}}^\top\|_2 = \max_{j\in\{1,\cdots,d\}} \lambda_j(1-\tilde{\lambda}_j^p) = \lambda_{max}\max_{j\in\{1,\cdots,d\}}\tilde{\lambda}_j(1-\tilde{\lambda}_j^p), \tag{ix}$$

where $\odot$ indicates Hadamard product, $\lambda_{max} = \max_j \lambda_j$, and $\tilde{\lambda}$ is a normalized weight, $\tilde{\boldsymbol{\lambda}} = \frac{\boldsymbol{\lambda}}{\max_j \lambda_j} \in [0,1]^d$ (10). A function $\tilde{\lambda}(1-\tilde{\lambda}^p)$ for various p is depicted in Fig. A, demonstrating that lower p contributes to reduce the overlap; especially, the overlap is reduced to 0 by $p=0$. For $d \le C$, however, $p=0$ provides trivial projection of $\tilde{\mathtt{W}}\mathrm{diag}(\mathbf{1}-\tilde{\boldsymbol{\lambda}}^p)\tilde{\mathtt{W}}^\top = \mathbf{0}$ since $\boldsymbol{W}$ usually has d rank, being full column rank, with $\lambda_j > 0\,\forall j$ to render $\mathbf{1}-\tilde{\boldsymbol{\lambda}}^p = \mathbf{0}$. Thus, there is a trade-off between valid complementary space via larger $1-\tilde{\lambda}^p$ and overlap reduction by smaller p; the experimental results in Table 6 imply that $p = 0.3$ provides a good trade-off.

On tht other hand, in case of $d > C$ for the direct-sum loss (Sec. 2.4), $p = 0$ builds the complementary classifier subspace as $\boldsymbol{\lambda}$ definitely contains zeros due to padding and $\mathbf{1}-\tilde{\boldsymbol{\lambda}}^p$ works as binary weighting to pick up the complementary bases $\mathtt{W}_\perp$ from $\tilde{\mathtt{W}} = [\mathtt{W}, \mathtt{W}_\perp]$;

$$\tilde{\boldsymbol{\lambda}} = \frac{1}{\lambda_{max}}[\lambda_1,\cdots,\lambda_C,0,\cdots,0] \in \mathbb{R}^d \Rightarrow \tilde{\boldsymbol{\lambda}}^0 = [\underbrace{1,\cdots,1}_{C},\underbrace{0,\cdots,0}_{d-C}] \in \{0,1\}^d, \tag{x}$$

which means

$$\tilde{\mathtt{W}}\mathrm{diag}(\mathbf{1}-\tilde{\boldsymbol{\lambda}}^0)\tilde{\mathtt{W}}^\top = [\underbrace{\mathtt{W}}_{C},\underbrace{\mathtt{W}_\perp}_{d-C}]\,\mathrm{diag}(\mathbf{1}-\tilde{\boldsymbol{\lambda}}^0)[\mathtt{W},\mathtt{W}_\perp]^\top = \mathtt{W}_\perp\mathtt{W}_\perp^\top. \tag{xi}$$

D Experimental setting

For training a backbone model $\phi_{\boldsymbol{\theta}}$, we can apply several types of sampling to construct a mini-batch as detailed in the followings.

In a standard way, we randomly draw n mini-batch samples, e.g., $n = 512$, from M training samples distributed over C classes; in this case, the number of intra-class samples

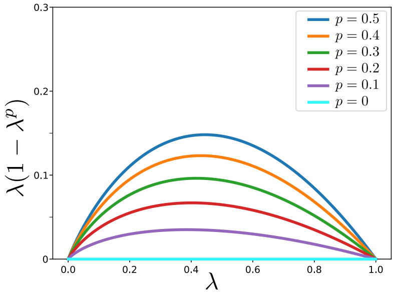

Figure A: Function of $\lambda(1-\lambda^p)$ with various p.

Table A: Various N-way K-shot sampling strategies for mini-batch in training.

	mini-ImageNet [5]		tiered-ImageNet [4]		CUB200 few-shot [2]		Cifar100 few-shot [1]	
Training classes	64		351		100		64	
Mini-batch size	512		512		128		512	
N-way K-shot	64-way	8-shot	64-way	8-shot	64-way	2-shot	64-way	8-shot
	32-way	16-shot	32-way	16-shot	32-way	4-shot	32-way	16-shot
	16-way	32-shot	16-way	32-shot	16-way	8-shot	16-way	32-shot
	8-way	64-shot	8-way	64-shot	8-way	16-shot	8-way	64-shot
	4-way	128-shot	4-way	128-shot	4-way	32-shot	4-way	128-shot

per class in a mini-batch is supposed to be roughly $\frac{nM}{C}$. The number of intra-class samples in a mini-batch would affect the performance especially for a metric-based loss, and thus we apply N-way K-shot strategy to the mini-batch sampling in a manner similar to episodic learning [5]; we draw K samples for each of N classes to build a mini-batch of $n = NK$ samples. Under the same budget of mini-batch size, we can consider several configurations for (N, K) as shown in Table A&Da. In Sec. 3, we report the best performance across those sampling strategies for fair comparison of all the methods even including the classification losses of SCE and BCE which are usually applied with randomly sampled mini-batches. The detailed performances are shown in Table B,C&Db.

References

[1] Luca Bertinetto, Jo ao Henriques, Philip H.S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019.

[2] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[3] Bharath Hariharan Davis Wertheimer. Few-shot learning with localization in realistic settings. In *CVPR*, 2019.

[4] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

[5] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016.

Table B: Classification accuracies (%) using various mini-batch sampling strategies in training (Table A); in each cell, left/right number shows accuracy in 1/5-shot evaluation setting, respectively. The right-most column shows the performance of random mini-batch sampling while the others are those of N-way K-shot mini-batch sampling. In performance comparison of Sec. 3, we pick up the best performance on each method, indicated by a gray-colored cell, that exhibits the best 1-shot accuracy across six types of sampling approaches.

mini-ImageNet

Method	64way - 8shot	32way - 16shot	16way - 32shot	8way - 64shot	4way - 128shot	512batch sample
Classification loss						
SCE	63.85, 79.68	63.47, 80.61	63.35, 81.00	61.62, 79.63	53.47, 70.08	63.75, 80.52
BCE	62.91, 79.26	63.97, 80.74	63.86, 80.79	62.81, 80.07	54.50, 71.91	64.00, 80.20
Metric loss						
NCA	62.39, 77.44	62.33, 78.08	63.07, 79.18	63.95, 79.47	62.13, 77.97	61.98, 76.78
Proto	60.43, 76.70	61.33, 77.95	61.40, 79.01	61.32, 78.38	60.87, 77.49	61.71, 77.56
Sum loss						
SCE+NCA	61.63, 77.15	62.14, 78.65	64.20, 80.09	63.81, 80.98	61.06, 78.30	61.49, 77.55
SCE+Proto	61.16, 77.95	61.91, 79.64	62.84, 80.82	62.02, 80.69	60.08, 77.59	61.15, 77.91
BCE+NCA	60.46, 75.52	61.71, 78.03	63.72, 79.84	64.64, 81.06	61.33, 77.82	61.90, 77.74
BCE+Proto	61.93, 78.09	62.60, 79.59	61.88, 79.86	63.43, 80.74	60.79, 78.37	62.41, 78.77
Direct-Sum loss (Ours)						
XE⊕NCA	63.08, 78.29	64.66, 80.08	64.53, 80.81	65.16, 81.89	61.37, 77.94	63.55, 78.51
XE⊕Proto	62.37, 77.94	63.11, 79.28	64.16, 81.19	64.05, 81.16	60.58, 77.08	63.65, 79.43
BCE⊕NCA	63.26, 78.17	63.86, 79.95	65.43, 80.92	65.61, 81.98	62.14, 78.31	62.59, 78.83
BCE⊕Proto	61.99, 77.86	63.46, 79.95	63.36, 80.51	64.27, 81.44	61.10, 77.77	63.19, 78.81

tiered-ImageNet

Method	64way - 8shot	32way - 16shot	16way - 32shot	8way - 64shot	4way - 128shot	512batch sample
Classification loss						
SCE	71.67, 86.37	71.33, 86.52	71.02, 86.33	68.27, 84.12	64.51, 81.44	71.35, 85.69
BCE	71.79, 86.13	72.14, 86.14	70.93, 86.19	69.55, 85.26	64.86, 80.48	71.59, 85.68
Metric loss						
NCA	69.90, 84.69	70.40, 84.95	70.19, 85.22	70.24, 84.75	68.31, 82.83	68.78, 83.60
Proto	70.23, 84.83	70.40, 85.39	69.91, 85.18	69.12, 84.89	67.20, 83.21	68.57, 82.69
Sum loss						
SCE+NCA	69.34, 84.63	70.08, 84.87	69.68, 85.13	68.56, 84.29	65.71, 81.91	68.94, 83.78
SCE+Proto	70.95, 85.66	70.48, 86.81	69.43, 86.38	69.01, 85.95	66.22, 82.85	69.18, 82.95
BCE+NCA	69.59, 84.79	70.84, 85.48	70.09, 85.40	69.48, 84.46	66.36, 82.10	69.79, 84.20
BCE+Proto	71.36, 85.95	70.62, 86.38	71.13, 86.42	70.02, 85.82	66.09, 83.22	69.27, 82.93
Direct-Sum loss (Ours)						
SCE ⊕NCA	71.79, 85.82	72.20, 86.50	71.57, 86.62	70.82, 85.87	66.87, 82.58	70.82, 84.63
SCE ⊕Proto	72.06, 85.87	72.27, 86.75	71.23, 86.26	70.52, 85.87	67.16, 82.85	71.03, 84.17
BCE⊕NCA	71.34, 85.85	71.32, 85.98	71.61, 86.69	70.66, 85.49	67.04, 83.16	70.67, 84.80
BCE⊕Proto	72.14, 85.74	72.14, 86.52	71.73, 86.44	70.43, 85.69	66.78, 82.46	70.48, 84.19

Table C: Classification accuracies (%) across various sampling strategies. (cont.)

CUB200 few-shot

Method	64way - 2shot	32way - 4shot	16way - 8shot	8way - 16shot	4way - 32shot	128batch sample
Classification loss						
SCE	72.57, 88.03	72.94, 88.06	73.37, 88.53	70.19, 86.54	53.75, 69.89	72.70, 87.87
BCE	74.83, 89.04	75.28, 89.47	75.80, 89.57	72.65, 87.81	58.35, 74.23	74.15, 88.64
Metric loss						
NCA	70.42, 82.51	75.78, 87.74	75.26, 87.15	72.32, 85.32	64.81, 78.43	70.81, 84.72
Proto	53.42, 61.98	76.33, 88.53	75.82, 88.69	73.04, 87.44	67.03, 81.39	70.79, 84.03
Sum loss						
SCE+NCA	78.06, 90.13	77.42, 90.05	78.47, 90.66	76.27, 89.82	67.63, 82.46	77.85, 90.20
SCE+Proto	76.13, 87.72	76.33, 89.64	76.43, 90.16	74.39, 89.90	67.09, 83.42	76.85, 89.77
BCE+NCA	78.61, 90.06	78.95, 90.26	78.57, 90.46	77.11, 89.77	69.31, 83.42	77.40, 89.58
BCE+Proto	77.70, 88.51	78.43, 90.44	77.12, 90.49	75.98, 90.07	68.63, 84.44	78.66, 90.36
Direct-Sum loss (Ours)						
SCE⊕NCA	78.20, 89.92	78.91, 90.73	78.14, 90.76	75.39, 89.54	65.23, 80.72	77.67, 90.00
SCE⊕Proto	78.30, 90.10	78.14, 90.75	76.37, 90.35	75.21, 89.63	65.83, 81.48	77.62, 89.98
BCE⊕NCA	79.89, 90.49	79.62, 90.96	78.42, 90.86	76.46, 89.82	67.55, 82.13	78.62, 90.23
BCE⊕Proto	78.67, 89.99	78.09, 90.48	77.86, 90.49	76.50, 90.28	68.61, 83.19	78.48, 90.50

Cifar100 few-shot

Method	64way - 8shot	32way - 16shot	16way - 32shot	8way - 64shot	4way - 128shot	512batch sample
Classification loss						
SCE	70.88, 84.84	68.40, 84.01	66.90, 84.14	63.25, 81.45	56.09, 73.77	70.04, 84.82
BCE	70.15, 83.91	69.33, 84.63	68.22, 83.72	65.97, 83.05	60.08, 77.67	69.80, 83.97
Metric loss						
NCA	70.80, 82.80	70.53, 83.57	71.49, 84.49	70.45, 84.30	66.30, 81.59	69.44, 82.43
Proto	70.10, 83.97	69.60, 84.31	69.36, 83.95	68.56, 84.33	66.77, 82.47	69.51, 82.33
Sum loss						
SCE+NCA	70.56, 83.59	70.80, 84.57	70.85, 85.39	70.25, 84.95	63.75, 80.76	69.83, 83.09
SCE+Proto	68.85, 83.51	68.89, 84.67	68.83, 85.13	67.33, 84.67	62.24, 80.98	69.42, 84.15
BCE+NCA	69.93, 82.32	71.00, 83.45	72.03, 85.17	69.56, 84.41	66.62, 81.91	69.67, 82.74
BCE+Proto	69.78, 83.28	69.72, 84.25	68.96, 84.31	67.93, 84.64	65.92, 82.49	69.31, 83.21
Direct-Sum loss (Ours)						
SCE ⊕NCA	71.27, 83.47	71.71, 84.92	71.98, 85.69	69.74, 84.79	64.79, 80.29	70.85, 83.60
SCE ⊕Proto	71.68, 84.10	69.93, 84.94	70.04, 85.25	68.73, 84.69	66.88, 82.32	70.69, 84.34
BCE⊕NCA	70.65, 83.22	70.22, 84.18	72.06, 85.33	70.48, 84.93	66.30, 81.96	69.29, 82.54
BCE⊕Proto	70.41, 83.34	70.11, 84.27	70.38, 85.12	69.35, 84.65	65.77, 81.74	70.74, 83.95

Table D: Classification accuracies (%) on iNaturalist2017 dataset [3]. The dataset is detailed in (a) with various settings of mini-batch sampling. The detailed performance results over various sampling strategies are shown in (b). In Table 6, we report the performances of our methods with the best setting at $p = 0.5$; that is, we apply 128-way 4-shot mini-batch sampling to SCE$\tilde{\bigoplus}_p$NCA and random mini-batch sampling to SCE$\tilde{\bigoplus}_p$Proto for $\forall p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

(a) Dataset

	iNaturalist2017 few-shot [3] training / test
Classes	908 / 227
Samples	197612 / 46374
Mini-batch size	512
N-way K-shot	256-way 2-shot 128-way 4-shot 64-way 8-shot

(b) Performance

Method	256way - 2shot	128way - 4shot	64way - 8shot	512batch sample
Classification loss				
SCE	80.53, 91.96	80.12, 91.79	79.24, 91.64	81.76, 92.73
NCA	81.23, 90.82	82.09, 91.93	82.42, 92.22	75.72, 87.87
Proto	76.30, 85.28	82.95, 92.08	81.61, 92.52	76.32, 88.54
Sum loss				
SCE+NCA	82.43, 92.16	82.43, 92.29	81.65, 92.25	81.17, 91.42
SCE+Proto	81.22, 89.88	81.91, 92.41	80.70, 92.39	82.02, 91.87
Spectral-Sum loss (Ours)				
SCE$\tilde{\bigoplus}_p$NCA ($p = 0.5$)	83.09, 92.26	83.13, 92.57	82.94, 92.74	82.91, 92.45
SCE$\tilde{\bigoplus}_p$Proto ($p = 0.5$)	82.82, 91.37	83.17, 92.57	81.85, 92.95	83.30, 92.77