Fixed Non-negative Orthogonal Classifier: Inducing Zero-mean Neural Collapse with Feature Dimension Separation

Hoyong Kim, Kangil Kim
Artificial Intelligence Graduate School
Gwangju Institute of Science and Technology

Overview

- ## Preliminaries
  - **Neural Collapse (NC):** A recently discovered phenomenon that at the terminal phase of training, the last-layer features of the same class will **collapse** into a single vertex, and the vertices of all classes will be aligned with their classifier prototypes and be formed as a simplex equi-angular tight frame (ETF)
  - **Layer-Peeled Model (LPM):** concentrate exclusively on the last-layer features $\mathbf{h}$ and class weight vectors $\mathbf{w}$ disregarding the encoder

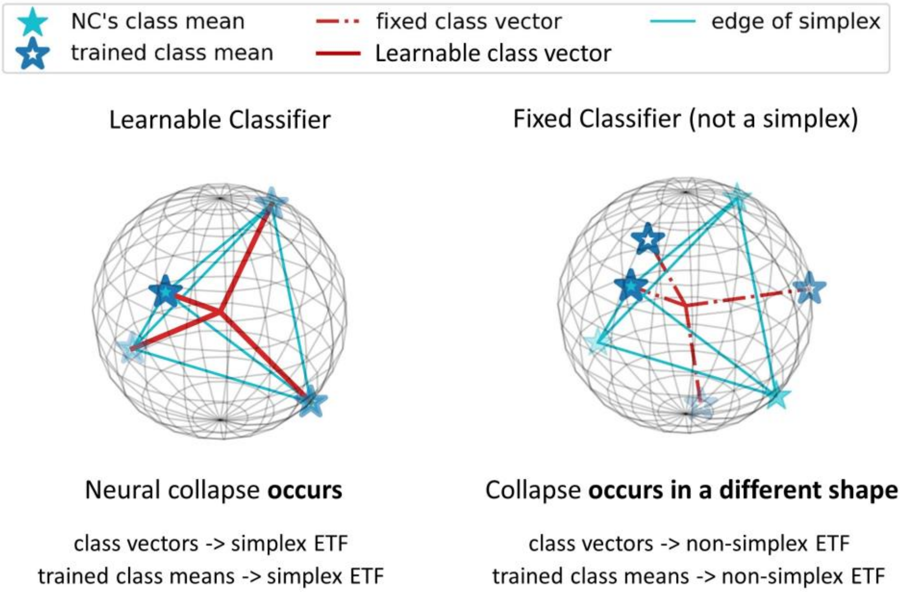
$$\min_{\mathbf{W},\mathbf{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}_{ce}\left(\mathbf{h}_{k,i},\mathbf{W}\right),$$
$$s.t. \|\mathbf{w}_k\|^2 \le E_W, \forall 1 \le k \le K,$$
$$\left\|\mathbf{h}_{k,i}\right\|^2 \le E_H, \forall 1 \le k \le K, \forall 1 \le i \le n_k$$

The LPM has been proven to achieve global optimality when satisfying NC properties:

$$\mathbf{h}_{k,i}^* = C\mathbf{w}_k^* = C'\boldsymbol{m}_k^*,$$

where the matrix $[\boldsymbol{m}_1^*, \ldots, \boldsymbol{m}_K^*]$ forms a K-simplex ETF

- ## Motivation



How does the **collapse** between class means and class weight vectors occur in the fixed classifier when their shape is not a simplex ETF?

Overview

- **Research Problem**
  - When the LPM with a fixed classifier which shape is non-simplex ETF, NC cannot explain the collapse phenomenon of it.

- **Solution**
  - **Non-negativity and Orthogonality**
    : These two additional constraints in the LPM, it can achieve the global optimality even in inducing the max-margin decision
    → **Fixed Non-negative Orthogonal Classifier**

  - **Zero-mean Neural Collapse (ZNC)**
    : when the LPM with a fixed non-negativity orthogonal classifier achieves the global optimality, it satisfies a different collapse properties.

    → To explain it, we propose a zero-mean neural collapse

Orthogonal Layer-Peeled Model (OLPM): the layer-peeled model with a fixed non-negative orthogonal classifier $\mathbf{Q}^*$

$$\min_{\mathbf{W},\mathbf{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}_{ce}\left(\mathbf{h}_{k,i},\mathbf{Q}^*\right),$$

$$s.t.\ \left\|\mathbf{h}_{k,i}\right\|^2 \le E_H,\ \text{and} \sum_{j\neq k}^{K}\mathbf{h}_{k,i}^{\top}\mathbf{q}_j^* \ge 0,$$

$$\forall 1\le k\le K, \forall 1\le i\le n_k$$

The OLPM achieves global optimality when satisfying ZNC properties:

$$\mathbf{h}_{k,i}^{*}{}^{\top}\mathbf{q}_{k'}^{*} = (K-1)\delta_{k,k'}$$

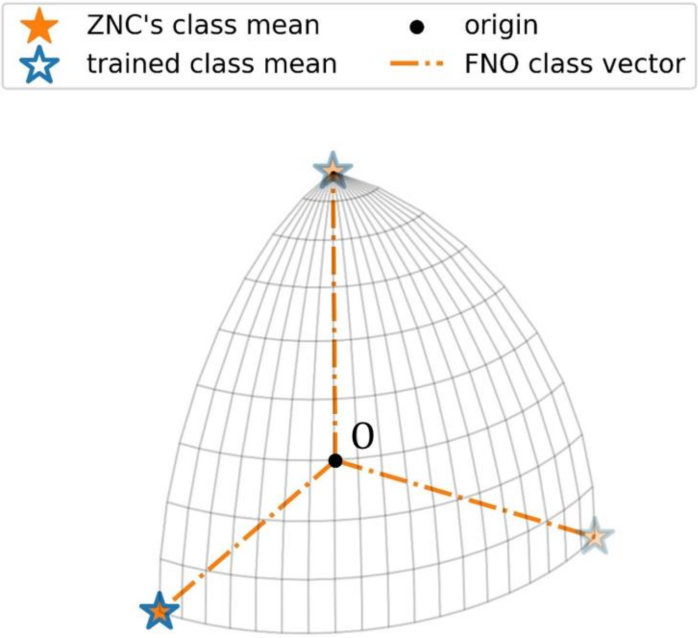
where $\delta_{k,k'}$ is a kronecker delta function

Proposed Method: Fixed Non-negative Orthogonal Classifier

Definition 1 (Non-negative Orthogonal Classifier). *A non-negative orthogonal classifier has a partial orthogonal weight matrix* $\mathbf{Q} \in \mathbb{R}_{\geq 0}^{D \times K}$, *which satisfies below properties:*

$$\mathbf{Q}^\intercal \mathbf{Q} = \boldsymbol{I}_K, \qquad s.t. Q_{i,j} \geq 0, \forall 1 \leq i \leq D, \forall 1 \leq j \leq K,$$

where $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ *is the identity matrix and* $Q_{i,j}$ *is the (i, j) element of* $\mathbf{Q}$



* The properties of Zero-mean Neural Collapse

: The only difference with NC is that class means are centered to the origin, not their global mean, as still satisfying the properties of neural collapse

Feature Dimension Separation

- FNO classifier invokes *feature dimension separation* (FDS), which **reduces the interference between class weight vectors**

→ FDS is useful in continual learning and imbalanced learning !

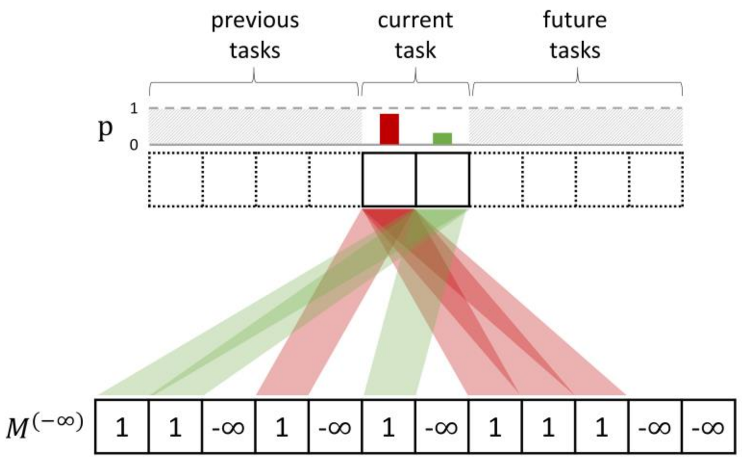
Definition 2 (Feature Dimension Separation)**.** *Let* $\boldsymbol{q}_k = \{q_j\}_{1 \le j \le D}$ *the k-th class weight vector in the fixed non-negative orthogonal classifier and* $\mathbb{J}_k = \{j \mid q_j > 0, 1 \le j \le D\}$ *an index set of* $\boldsymbol{q}_k$ *where* q_j *is not zero. Then, as the definition of FNO classifier, any index set of class weight vectors has disjoint to any other class weight vector and we call this phenomenon as feature dimension separation, i.e.,*

$$\mathbb{J}_k \cap \mathbb{J}_{k'} = \emptyset, \qquad \forall k \neq k',$$

which means that the features' elements used for deciding the confidence to any class lose their utility to any other classes.

Enhancement of Masked Softmax in Continual Learning

$$\mathbf{p} = \mathrm{WeightedSoftmax}\left(M^{(-\infty)} \odot \mathrm{MatMul}(\mathbf{Q}, \mathbf{H})\right)$$



Definition 3 (Masked Softmax)**.** *To remove the class-wise interference of specific classes in softmax, the masked softmax multiplies a negative infinity mask* $\boldsymbol{M}^{(-\infty)}$ *to output vectors. When getting rid of k-th class's interference from i-th input sample, k-th element of the output vector is multiplied by negative infinity values, i.e.,*

$$\boldsymbol{M}_i^{(-\infty)} = (m_j)_{1 \le j \le K}$$

$$\mathbf{p} = \mathrm{Softmax}\left(\boldsymbol{M}_i^{(-\infty)} \odot (\mathbf{W}^\top \mathbf{h} + \mathbf{b})\right),$$

where $m_j = -\infty$ *if* $j = k$ *otherwise 1 and* $\odot$ *means the Hadamard product.*

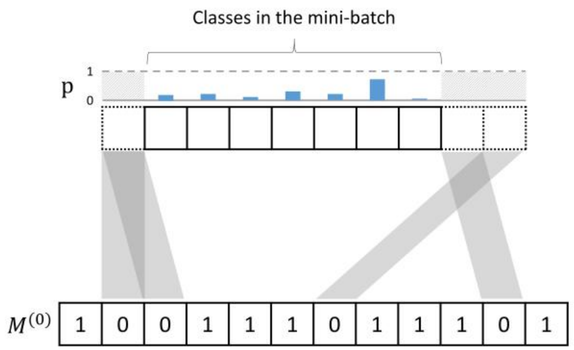
Algorithm 1 Masked and Weighted Softmax with FNO classifier in a Task T_t

Require: $(\boldsymbol{X}_t, \boldsymbol{Y}_t)$, $\boldsymbol{Q}$
Ensure: $\mathbf{P} \in \mathbb{R}^{N_t \times K}$

```
H ← ReLU(f_θ(X_t))                                              ▷ get features from X_t as Eq. 1
𝕂 = {c_i | c_i of X_t}                                          ▷ initialize a set of class labels in the task T_t.
M^(-∞) = (m_{i,j})_{1≤i≤N_t, 1≤j≤K} , where m_{i,j} = -∞ if j ∉ 𝕂 otherwise 1    ▷ initialize M^(-∞)
P = W-Softmax(M^(-∞) ⊙ MatMul(Q, H))                            ▷ get the confidence of H
```

Arc-mixup with Feature Masking in Imbalanced Learning

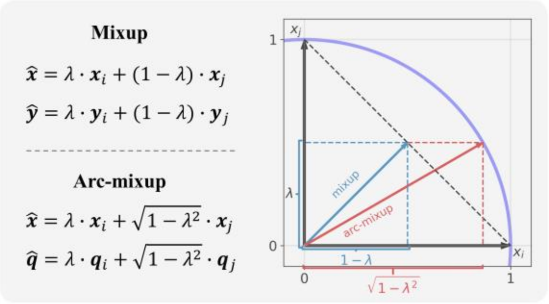
$$\mathbf{p} = \mathrm{MatMul}\left(\widehat{Q}, \mathrm{LayerNorm}\left(M^{(0)} \odot \widehat{\mathbf{H}}\right)\right)$$



$\mathbf{H}$: last-layer features

$\mathbf{Q}$: non-negative and orthogonal matrix

$M^{(-\infty)}$: negative infinite mask $\quad M^{(0)}$: zero mask



Definition 4 (Arc-mixup). *Arc-mixup is an interpolation-based method when all last-layer features and class weight vectors are located on the same hypersphere, while keeping the scale of mixed class weight vector* $\widehat{\mathbf{q}}$, *i.e.,*

$$\widehat{\boldsymbol{x}} = \lambda \cdot \boldsymbol{x}_i + \sqrt{1-\lambda^2} \cdot \boldsymbol{x}_j$$
$$\widehat{\mathbf{q}} = \lambda \cdot \boldsymbol{q}_i + \sqrt{1-\lambda^2} \cdot \boldsymbol{q}_j$$
$$\mathcal{L}_{cls}(\widehat{\boldsymbol{x}}, \widehat{\mathbf{q}}) = -\log \widehat{\mathbf{q}}^{\intercal} \widehat{\mathbf{h}}$$

and $\widehat{\mathbf{h}}$ *is the last-layer feature of* $\widehat{\boldsymbol{x}}$. *This means that* $\widehat{\mathbf{q}}$ *is still located on the hypersphere*

Algorithm 2 Arc-mixup with FNO classifier and feature masking in a mini-batch $\mathbb{B}$

Require: $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{B}, \boldsymbol{Q}$

Ensure: $\mathbf{P} \in \mathbb{R}^{|\mathbb{B}| \times K}$

$(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Q}}) \leftarrow ArcMixup(\boldsymbol{X}, \boldsymbol{Q})$ ▷ mixup input samples and class vectors as Eq. 11

$\hat{\mathbf{H}} \leftarrow \mathrm{ReLU}(f_\theta(\hat{\boldsymbol{X}}))$ ▷ get features from $\hat{\boldsymbol{X}}$ as Eq. 1

$\mathbb{K} = \{c_i \mid c_i \in \mathbb{B}, \forall 1 \le i \le |\mathbb{B}|\}$ ▷ initialize a set of class labels in the mini-batch

$\hat{\mathbb{J}} = \bigcup_{k \in \mathbb{K}} \mathbb{J}_k$ ▷ initialize an index set including all index sets of $\boldsymbol{Q}$

$M^{(0)} = (m_{i,j})_{1 \le i \le |\mathbb{B}|, 1 \le j \le D}$, where $m_{i,j} = \mathbf{1}_{j \in \hat{\mathbb{J}}}$ ▷ initialize a zero mask $M^{(0)}$

$\mathbf{P} = \mathrm{MatMul}(\hat{\boldsymbol{Q}}, \mathrm{LayerNorm}(M^{(0)} \odot \hat{\mathbf{H}}))$ ▷ get the confidence of $\hat{\mathbf{H}}$

Experimental Results

Intelligence Representation &Reasoning Lab

In classification results for standard CL bench-marks, our method demonstrated superior performance in all class-incremental learning settings *(Max: +5.23 in S-TinyImageNet with buffer size 200)*

$\mathcal{B}$	Method			S-MNIST		S-CIFAR-10		S-CIFAR-100		S-Tiny-ImageNet	
	RM	Clf	M	*Class-IL*	*Task-IL*	*Class-IL*	*Task-IL*	*Class-IL*	*Task-IL*	*Class-IL*	*Task-IL*
200	ER	FC	✓	$82.98_{1.03}$	$98.13_{0.16}$	$61.75_{6.07}$	$91.39_{2.13}$	$28.51_{0.44}$	$68.51_{0.87}$	$15.47_{0.67}$	$44.11_{0.50}$
	ER	FNO	✓	$84.26_{1.16}$	$98.45_{0.19}$	$63.84_{1.47}$	$92.03_{0.52}$	$\mathbf{32.43}_{0.58}$	$71.34_{1.17}$	$17.31_{0.74}$	$44.76_{0.90}$
				+1.28	+0.32	+2.09	+0.64	+3.92	+2.83	+1.84	+0.65
	DER++	FC	✓	$84.45_{0.88}$	$99.03_{0.09}$	$66.35_{1.52}$	$93.17_{0.54}$	$28.57_{1.11}$	$74.02_{0.76}$	$13.21_{0.56}$	$49.75_{0.99}$
	DER++	FNO	✓	$\mathbf{86.27}_{0.88}$	$\mathbf{99.11}_{0.08}$	$\mathbf{67.53}_{1.25}$	$\mathbf{93.98}_{0.39}$	$30.70_{1.16}$	$\mathbf{74.11}_{0.96}$	$\mathbf{18.44}_{0.94}$	$\mathbf{53.06}_{0.67}$
				+1.82	+0.08	+1.18	+0.81	+2.13	+0.09	+5.23	+3.31
500	ER	FC	✓	$89.35_{0.59}$	$\mathbf{99.20}_{0.16}$	$70.64_{1.28}$	$94.22_{0.41}$	$35.68_{0.89}$	$74.77_{0.71}$	$20.43_{0.38}$	$53.21_{0.84}$
	ER	FNO	✓	$\mathbf{89.42}_{0.72}$	$99.16_{0.17}$	$71.43_{0.95}$	$94.38_{0.43}$	$39.80_{0.68}$	$76.52_{0.86}$	$22.41_{0.57}$	$52.60_{0.58}$
				+0.07	-0.04	+0.79	+0.16	+4.12	+1.75	+1.98	-0.61
	DER++	FC	✓	$83.10_{1.22}$	$99.08_{0.09}$	$71.85_{3.76}$	$94.28_{1.49}$	$37.80_{0.92}$	$80.52_{0.60}$	$17.71_{0.58}$	$59.86_{1.08}$
	DER++	FNO	✓	$86.75_{0.75}$	$99.00_{0.10}$	$\mathbf{74.77}_{0.66}$	$\mathbf{95.56}_{0.16}$	$\mathbf{40.81}_{0.70}$	$\mathbf{80.61}_{0.46}$	$\mathbf{22.45}_{0.36}$	$\mathbf{59.87}_{1.91}$
				+3.65	-0.08	+2.92	+1.28	+3.01	+0.09	+4.74	+0.01
5120	ER	FC	✓	$93.51_{0.60}$	$99.38_{0.12}$	$82.63_{1.34}$	$96.45_{0.27}$	$52.95_{0.73}$	$84.20_{0.58}$	$35.73_{0.41}$	$67.50_{0.53}$
	ER	FNO	✓	$93.98_{0.39}$	$99.47_{0.09}$	$82.88_{1.35}$	$96.79_{0.38}$	$57.02_{0.52}$	$85.66_{0.42}$	$36.90_{0.41}$	$66.86_{0.32}$
				+0.47	+0.09	+0.25	+0.34	+4.07	+1.46	+1.17	-0.64
	DER++	FC	✓	$93.75_{0.23}$	$\mathbf{99.62}_{0.05}$	$84.71_{0.65}$	$96.78_{0.16}$	$58.18_{0.43}$	$\mathbf{87.97}_{0.33}$	$34.72_{0.46}$	$72.40_{0.25}$
	DER++	FNO	✓	$\mathbf{94.26}_{0.24}$	$99.59_{0.05}$	$\mathbf{85.65}_{0.38}$	$\mathbf{97.20}_{0.13}$	$\mathbf{58.82}_{0.43}$	$87.35_{0.45}$	$\mathbf{38.95}_{0.71}$	$\mathbf{72.70}_{0.27}$
				+0.51	-0.03	+0.94	+0.42	+0.64	-0.62	+4.23	+0.30

Experimental Results

In imbalanced learning on CIFAR10/100-LT, ImageNet-LT and Places-LT, our method performed better than other methods *(Max: +9.90 in Places-LT)*

Method			Reference	CIFAR10-LT			CIFAR100-LT		
Aug	Clf	$\mathcal{L}$		100	50	10	100	50	10
mixup	FC	CE	(Yang et al., 2022b)	$73.90_{0.30}$	$79.30_{0.20}$	$87.80_{0.10}$	43.00	48.10	-
B-mixup	FC	CE	(Zhang et al., 2022b)	78.70	-	**89.60**	-	-	-
mixup	ETF	CE	(Yang et al., 2022b)	$67.00_{0.40}$	$77.20_{0.30}$	$87.00_{0.20}$	-	-	-
mixup	ETF	DR	(Yang et al., 2022b)	$76.50_{0.30}$	$81.00_{0.20}$	$87.70_{0.20}$	45.30	50.40	-
mixup	FC	CE	(reproduced.)†	$74.24_{0.44}$	$80.00_{0.54}$	$89.08_{0.32}$	$43.80_{0.42}$	$49.57_{0.37}$	$\mathbf{63.90}_{0.33}$
mixup	ETF	DR	(reproduced.)†	$75.18_{0.49}$	$80.17_{0.31}$	$87.29_{0.23}$	$45.45_{0.38}$	$50.67_{0.37}$	$62.84_{0.38}$
arc-mixup	FNO	CE	(reproduced.)†	$\mathbf{82.59}_{0.26}$	$\mathbf{85.13}_{0.25}$	$89.50_{0.14}$	$\mathbf{49.26}_{2.82}$	$\mathbf{54.44}_{2.32}$	$63.14_{3.82}$

Method			Reference	ImageNet-LT (ResNet50)				Places-LT	
Aug	Clf	$\mathcal{L}$		Many	Median	Few	All	ResNet152	ResNet152 (FT)
mixup	FC	CE	(Yang et al., 2022b)*	-	-	-	44.30	-	-
mixup	ETF	DR	(Yang et al., 2022b)*	-	-	-	44.70	-	-
-	FC	CE	(reproduced)†	$66.53_{0.18}$	$40.34_{0.45}$	$12.03_{0.30}$	$45.88_{0.31}$	$22.62_{0.29}$	$24.16_{0.41}$
mixup	FC	CE	(reproduced)†	$\mathbf{67.40}_{0.55}$	$38.74_{0.83}$	$9.12_{0.40}$	$45.03_{0.64}$	$22.10_{0.27}$	$24.83_{1.19}$
mixup	ETF	DR	(reproduced.)†	$64.17_{0.27}$	$22.12_{0.38}$	$0.57_{0.16}$	$34.96_{0.27}$	$23.11_{0.16}$	$25.51_{0.08}$
arc-mixup	FNO	CE	(reproduced)†	$59.46_{0.51}$	$\mathbf{43.54}_{0.26}$	$\mathbf{24.48}_{0.55}$	$\mathbf{46.60}_{0.40}$	$\mathbf{30.07}_{0.40}$	$\mathbf{34.06}_{0.10}$

Conclusion

- **Contributions**

  - **Zero-mean Neural Collapse**
    : we propose a *zero-mean neural collapse* to analyze the collapse phenomenon in training classification model with the fixed classifier

  - **Fixed Non-negative Orthogonal Classifier**
    : we propose a *fixed non-negative orthogonal classifier* and prove its theoretical benefits in orthogonal layer-peeled model with the zero-mean neural collapse

  - **Benefits of Feature Dimension Separation**
    : we demonstrate the impacts of the proposed methods with masked softmax in continual learning and arc-mixup in imbalanced learning

Thank You

https://github.com/GIST-IRR/FNO-classifier

hoyong.kim.21@gm.gist.ac.kr