Spherization Layer
: Representation Using Only Angles

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

GIST Intelligence Representation &Reasoning Lab

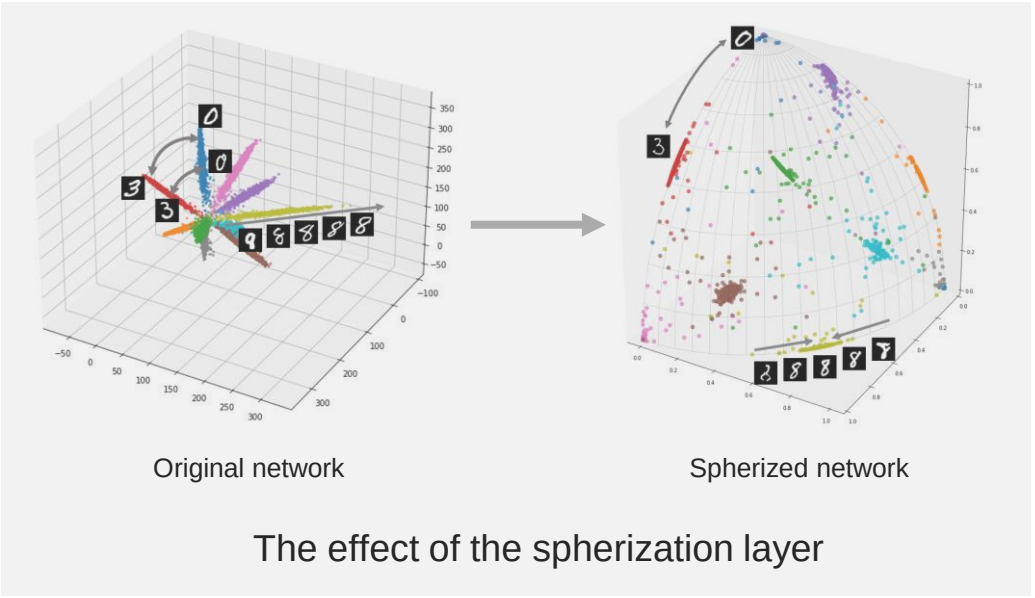
Overview

- **Background & Problem**
  - Inner product disperses information to scale and angles
  - Using only a factor loses information in re-using or analyzing representations

- **Motivation**

  How to address the dispersion problem?

  → Learning representations using only the angles!



The effect of the spherization layer

- **Spherization Layer**

  A layer to replace an ordinary layer in the network for learning representations on the hyperspherical surface

  The main effects from the spherization layer are as below
  - address the dispersion problem
  - maintain the training ability of original models
  - outperform in the tasks where the angle-based metric is crucial

Problem

- Is it real that the dispersion induces information loss?

Comparison with Angle-based Approach on Image Classification with CIFAR10

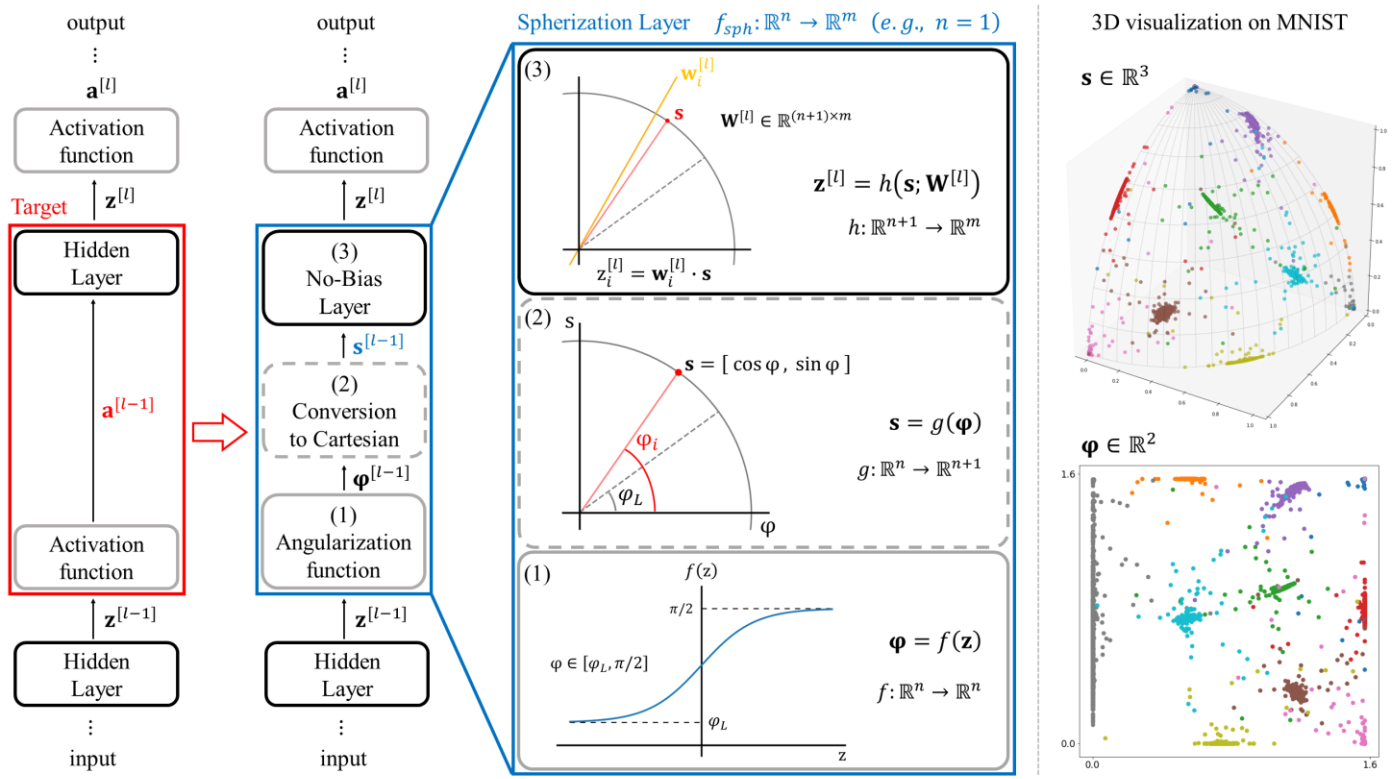
 # err. number of errors in overlapped samples
 # ovlp. number of overlapped samples
ratio ratio of # err. to # ovlp.

Role	Operator	No-bias	Train	Test			
			acc.	acc.	# err.	# ovlp.	(ratio)
base	Original Conv.		99.47 ± 0.42	92.46 ± 0.10	0 ± 0	0 ± 0	(0.00 ± 0.00 %)
direct	Sigmoid		81.74 ± 4.48	79.03 ± 2.71	1097 ± 266	5222 ± 391	(20.75 ± 3.70 %)
	Linear		74.71 ± 1.06	72.15 ± 0.32	1631 ± 39	6849 ± 286	(23.84 ± 0.63 %)
	Cosine		77.41 ± 0.89	76.15 ± 0.86	776 ± 132	3170 ± 330	(24.39 ± 2.66 %)
indirect	SW-Softmax	✓	98.32 ± 0.07	91.51 ± 0.19	167 ± 9	7925 ± 58	(2.11 ± 0.11 %)
	LW-Softmax	✓	86.74 ± 4.06	82.12 ± 3.82	946 ± 374	7669 ± 66	(12.30 ± 4.84 %)
	CW-Softmax	✓	99.66 ± 0.04	92.29 ± 0.18	80 ± 5	7051 ± 134	(1.14 ± 0.05 %)
proposed	Spherization	✓	99.66 ± 0.05	92.38 ± 0.14	0 ± 0	499 ± 106	(0.00 ± 0.00 %)

Large proportion of representations suffer the dispersion problem as ignoring the Euclidean norm by projection, and considerable errors are caused by the overlapped representations

Spherization Layer

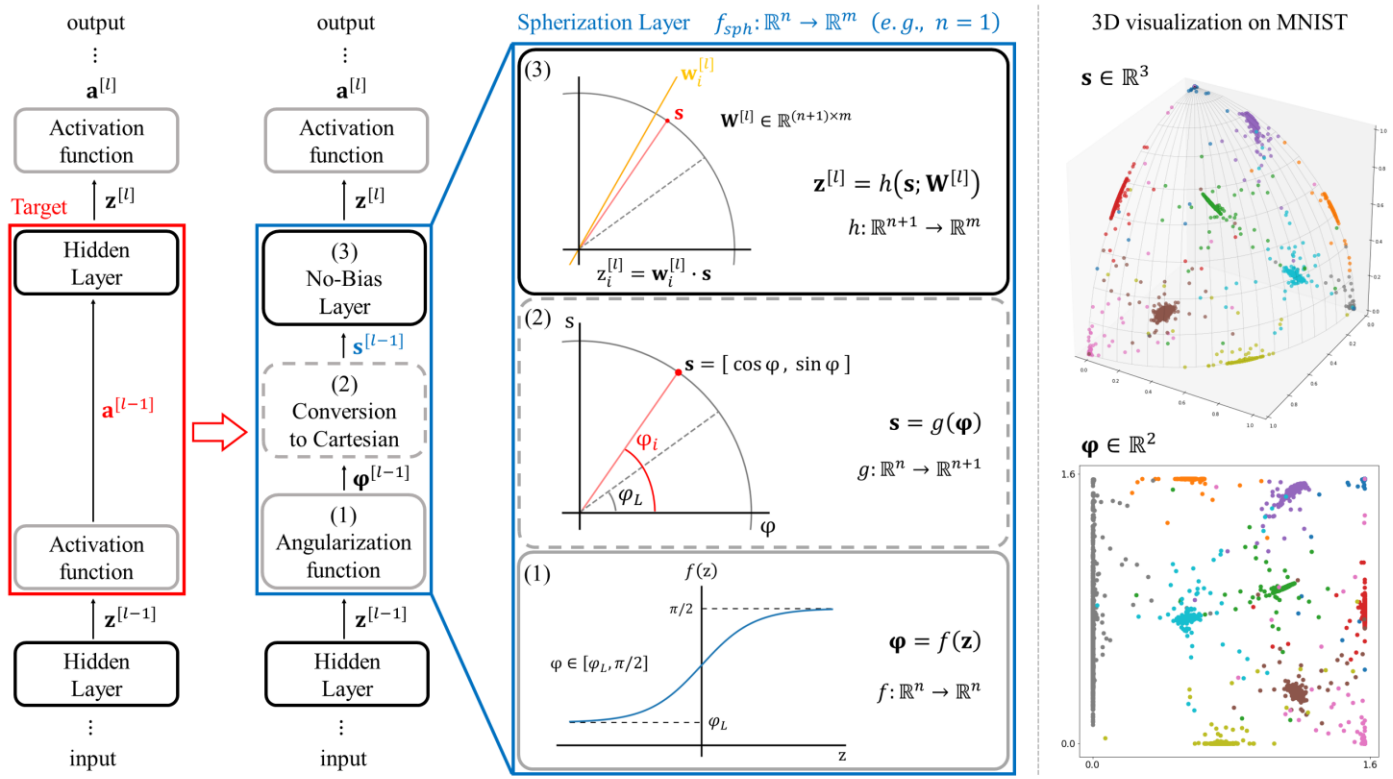
- An explicit solution for the dispersion to completely eliminate the interference of the norms in training without drawbacks



- Locate all representations onto a constrained region on the hyperspherical surface
- Train hyperplanes passing through the origin to learn representations with only the angles

Spherization Layer

- An explicit solution for the dispersion to completely eliminate the interference of the norms in training without drawbacks



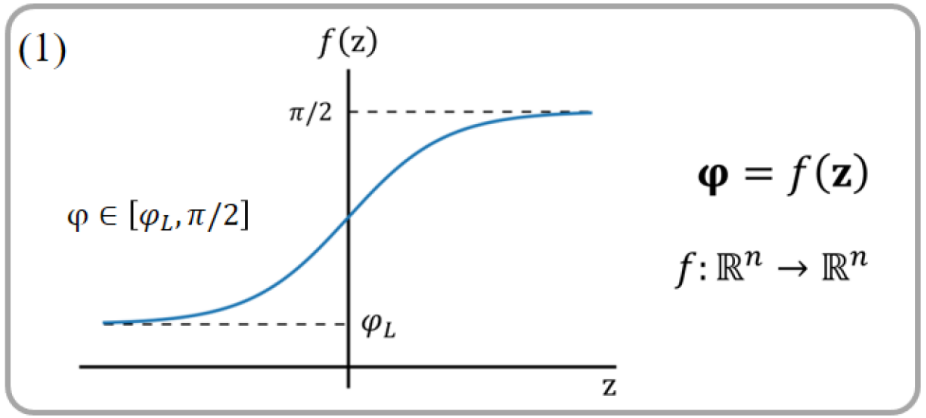
(1) Angularization function
converts pre-activations to angles

(2) Conversion-to-Cartesian
converts spherical coordinates to Cartesian coordinates

(3) No-bias Layer
determines decision boundaries by using only the angles

Spherization Layer

(1) Angularization



$$f(\mathbf{z}) = \left(\frac{\pi}{2} - \varphi_L\right) \cdot \sigma(\alpha \cdot \mathbf{z}) + \varphi_L$$

Converting Pre-Activation to the Angular Coordinate

$\boldsymbol{\varphi} \in [0, \pi/2]$

Tailoring Angular Representation Space $\boldsymbol{\varphi} \in [\varphi_L, \pi/2]$

$$\mathbf{s} = [r\cos\varphi_1, \cdots, r\cos\varphi_k \prod_{i=1}^{k-1}\sin\varphi_i, \cdots, r\prod_{i=1}^{n-1}\sin\varphi_i]$$

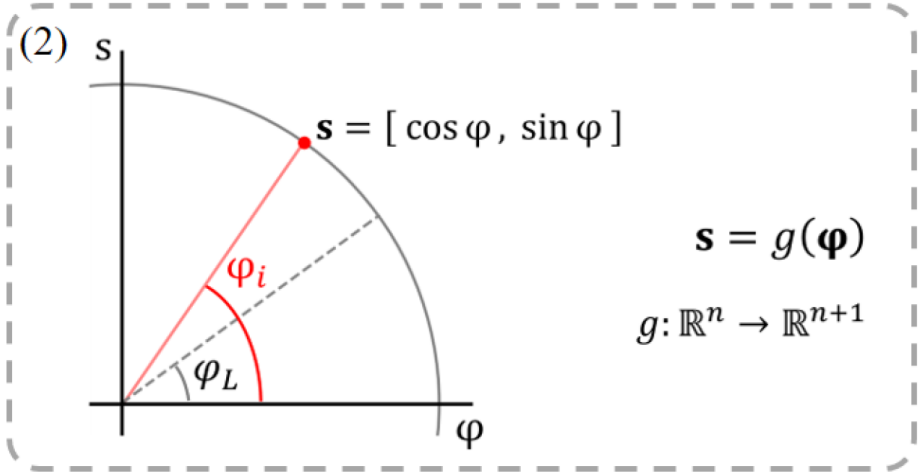
$$when\ \sin\phi = \alpha,\ \alpha^n \geq \delta \Leftrightarrow \alpha = \delta^{1/n}$$
$$\phi = \sin^{-1}\alpha \geq \sin^{-1}(\delta^{1/n})$$
$$\therefore\ \phi_L = \sin^{-1}(\delta^{1/n})$$

Scaling Pre-Activations α

To reduce concentration onto the small region by scaling pre-activations

Spherization Layer

(2) Conversion-to-Cartesian



Calculation Trick

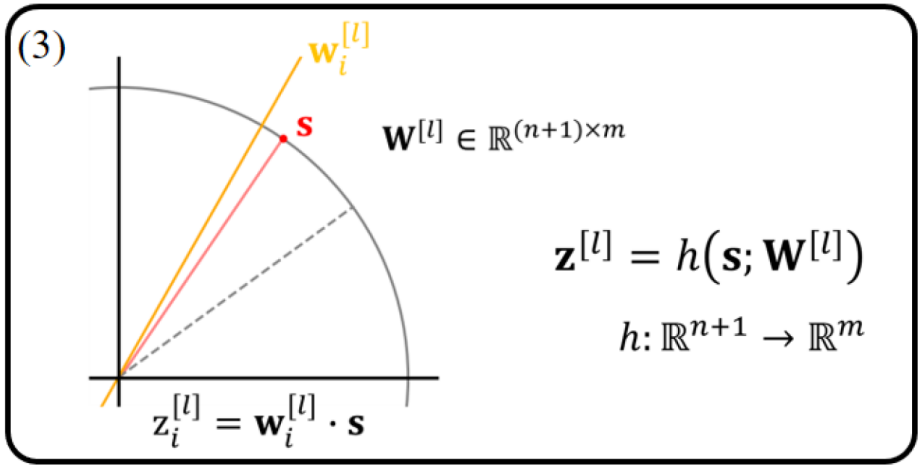
Implementation as a tensor operation requires the trick as below

$$\begin{aligned} \boldsymbol{\phi} &= \mathbf{W}_{\varphi}^{\intercal}\boldsymbol{\varphi} \\ \mathbf{s} &= r \cdot \exp\left(\mathbf{W}_{\phi}^{\intercal} \ln(\sin\boldsymbol{\phi}) + \ln\left(\cos\left(\boldsymbol{\phi} + \mathbf{b}_{\phi}\right)\right)\right) \end{aligned}$$

$$g(\boldsymbol{\varphi}) = [r\cos\varphi_1, \cdots, r\cos\varphi_k \prod_{i=1}^{k-1}\sin\varphi_i, \cdots, r\prod_{i=1}^{n}\sin\varphi_i], \quad \varphi_i \in \left[\varphi_L, \frac{\pi}{2}\right]$$

Spherization Layer

(3) No-bias Training



$$\mathbf{z}^{[l]} = \mathbf{W}^{[l]\top} \mathbf{s}$$

Effect of Bias Elimination to Training

In the ordinary layer, the problem of bias elimination is that *hyperplanes passing through the origin cannot be shifted to another parallel hyperplanes*

However, the problem disappears when all feature vectors are located on the (n+1)-spherical surface

Experiments

- **Functional Correctness Test on a Toy Task**

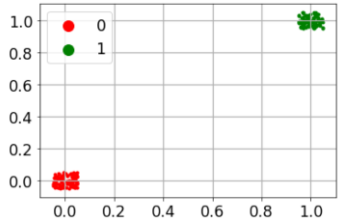


Figure 2: Input samples for the toy task

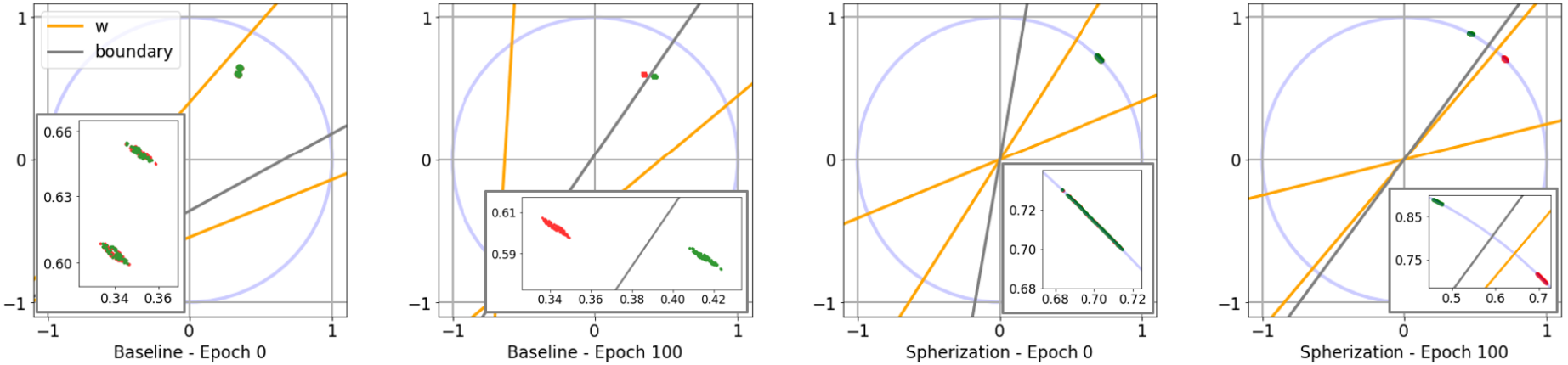


Figure 3: Visualization of Hyperplanes, Decision Boundary, and Representations in the Toy Task. (w: hyperplanes, boundary: decision boundary, red or green points: representations for label 0 or 1)

Experiments

- **Retention of Training Ability on Image Classification Benchmarks**

Table 1: Retention of the Training Ability on Image Classification with Various Datasets and Models (Accuarcy(%): $\mu \pm \sigma$)

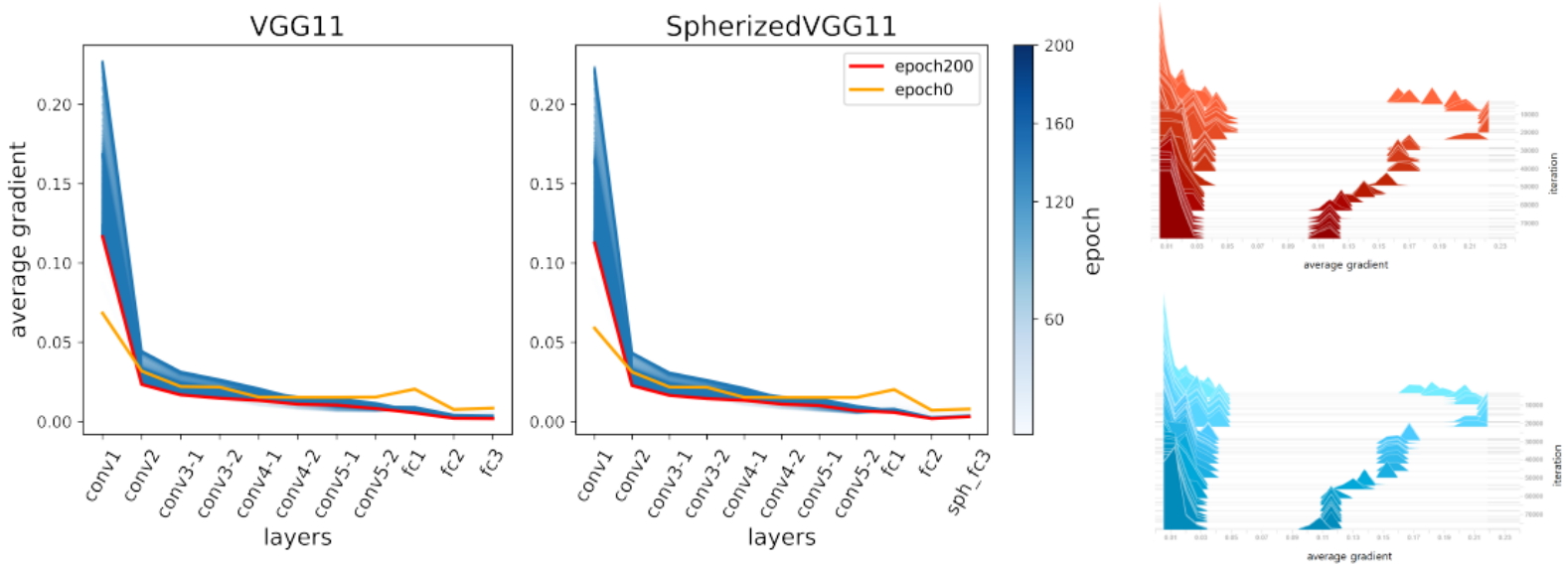
Network	Dataset	Reference		Reproduced		Spherized	
		train	test	train	test	train	test
SimpleFNN [8]	MNIST	–	98.47	99.99±0.01	98.58±0.03	99.99±0.01	**98.65±0.04**
LeNet-5 [11]	MNIST	–	99.05	99.55±0.09	99.10±0.05	99.79±0.09	**99.14±0.04**
VGG-11 [33]	F-MNIST	–	94.70	99.24±0.18	**94.36±0.06**	98.92±0.36	94.34±0.17
VGG-11 [33]	CIFAR10	–	90.90	100.00±0.00	92.38±0.06	100.00±0.00	**92.49±0.11**
VGG-11 [33]	CIFAR100	–	66.80	99.71±0.03	68.42±0.12	99.82±0.02	**69.03±0.24**

Table 2: Retention of Training Ability on Image Classification with CIFAR100 in Various Network Width and Depth Settings (Accuarcy(%): $\mu \pm \sigma$)

Depth	Width	Reference[2]		Reproduced		Spherized	
		train	test	train	test	train	test
VGG-11	16/32/64/128	–	–	79.23±5.94	60.17±0.21	77.48±5.60	**60.40±0.35**
VGG-11	32/64/128/256	–	–	98.34±0.37	64.89±0.38	96.98±3.67	**65.38±0.28**
VGG-11	64/128/256/512	–	–	99.71±0.03	68.42±0.12	99.82±0.02	**69.03±0.24**
VGG-11	128/256/512/1024	–	–	99.90±0.00	70.53±0.35	99.93±0.00	**70.89±0.19**
VGG-11	256/512/1024/1024	–	–	99.90±0.01	71.43±0.22	99.93±0.01	**71.94±0.19**
VGG-11	64/128/256/512	–	68.64	99.71±0.03	68.42±0.12	99.82±0.02	**69.03±0.24**
VGG-16	64/128/256/512	–	72.93	99.39±0.07	72.51±0.26	99.54±0.05	**72.53±0.17**
VGG-19	64/128/256/512	–	72.23	97.95±0.81	71.53±0.32	99.30±0.06	**72.17±0.33**

Experiments

- **Analysis**
  : Gradient Flows

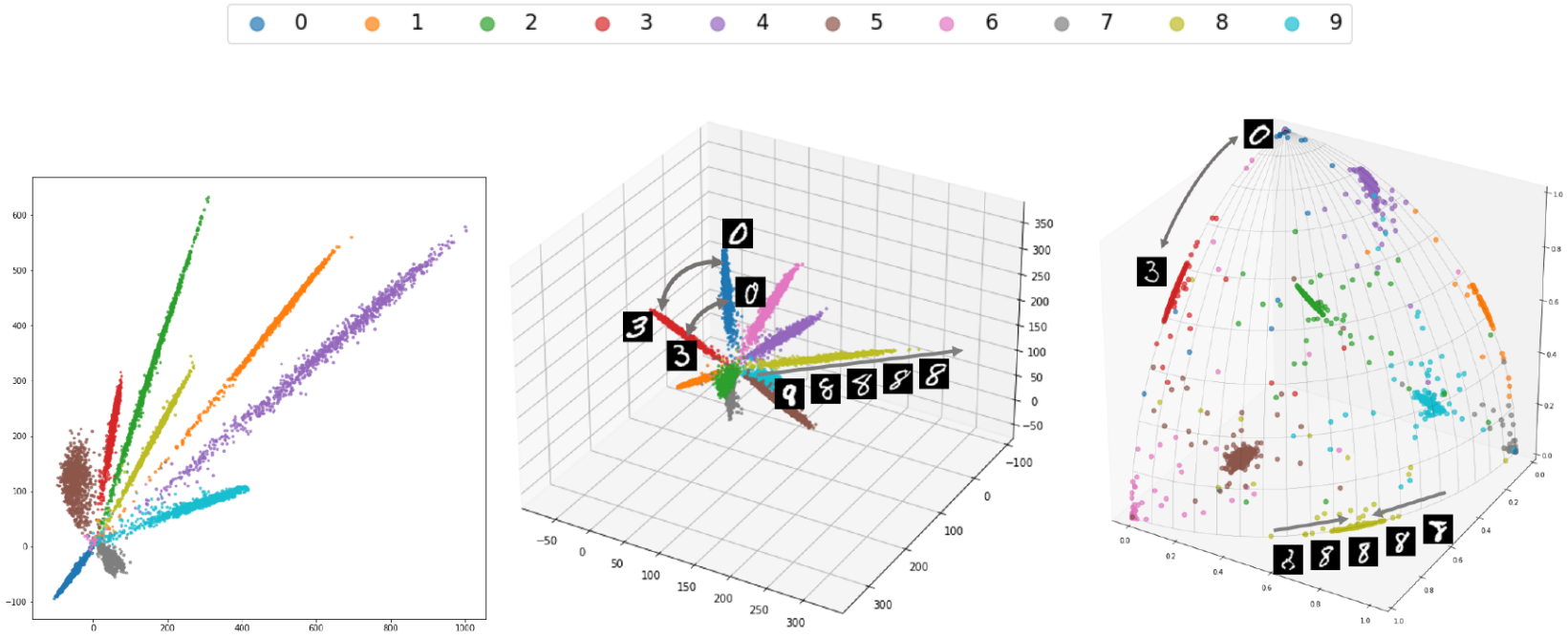


(a) Avg. of Absolute Gradients at Each Layer (b) Histograms

Figure 4: Analysis of Gradient Flow from Image Classification Model trained on CIFAR100. (a) The y-axis means the average of absolute gradients which occurred at each layer. The left side shows the gradient flow in VGG-11 (VGG11), and the right side shows the spherized VGG-11 (SpherizedVGG11), where the last fully connected layer is substituted with the spherization layer. (b) The histograms show the frequency of the average of absolute gradients in VGG-11 (red) and the spherized VGG-11 (cyan), respectively.

Experiments

- **Downstream Tasks: Visualization**



(a) Original 2D (b) Original 3D (c) Spherized 3D

Figure 5: Visualization of Feature Representations on MNIST. (a) and (b) are the visualization results of 2D and 3D feature vectors in the original networks, and (c) is the result in the spherized network

Experiments

- **Downstream Tasks: Word Analogy Test & Few-shot Learning**

Table 4: Performance on the Word Analogy Test ($S_{ppl}/S_{pmi}/S_{mppl}$ ↑)

Model	SAT	U2	U4	Google	BATS	Avg.
BERT	**29.4**/28.5/**28.8**	36.0/36.0/**36.8**	**38.7**/34.7/34.3	**33.0/33.8/33.0**	32.3/**35.0**/33.2	**33.9**/33.6/**33.2**
BERT + *sph*	29.1/**29.4**/27.9	**37.3/39.0**/36.0	36.8/**35.9/35.4**	32.4/32.6/32.2	**34.0**/34.2/**33.8**	**33.9/34.2**/33.1
RoBERTa	**29.4/31.2**/29.7	35.5/**35.5/36.4**	33.6/**34.3/34.5**	32.8/33.2/30.8	30.9/31.6/30.9	32.4/33.1/32.5
RoBERTa + *sph*	29.1/29.4/**30.0**	**36.4/35.5**/34.2	**34.0/34.3**/33.3	**34.2/33.6/32.8**	**35.0/33.9/34.8**	**33.7/33.3/33.0**

Table 5: Performance of Few-shot Learning on Mini-ImageNet (Accuarcy(%): $\mu \pm \sigma$)

Model	Test Acc.		Model	Test Acc.	
	Euclidean	Cosine		Euclidean	Cosine
ConvNet	50.29±0.18	52.87±0.18	ResNet	37.63±0.15	33.41±0.15
ConvNet + *sph*	43.41±0.16	**53.74±0.16**	ResNet + *sph*	31.77±0.13	**38.71±0.16**

Conclusion

- **Contributions**

  - To address the dispersion problem, we propose the ***spherization layer*** to represent all feature vectors on the hyperspherical surface and learn the representations with only the angles

  - We validate the wide-applicability and scalability of the spherization layer without any loss of performance through experiments on various well-known networks

  - We empirically show that the spherization layer can be used in many applications in which angular similarity is a critical metric.

Thank You

https://github.com/GIST-IRR/spherization_layer

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

GIST IRR Intelligence Representation &Reasoning Lab