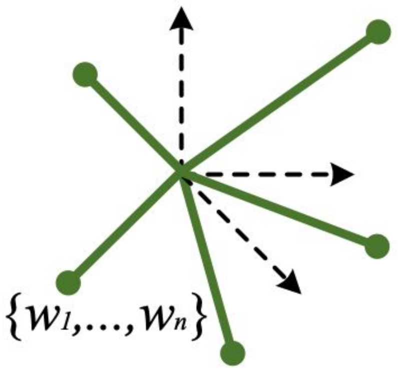
Regularizing Neural Networks via Minimizing Hyperspherical Energy

Rongmei Lin, Weiyang Liu, Zhen Liu, Chen Feng, Zhiding Yu,
James Rehg, Li Xiong, Le Song

CVPR SEATTLE WASHINGTON JUNE 16-18 2020

Hyperspherical Energy and Motivation

Neurons in one layer:



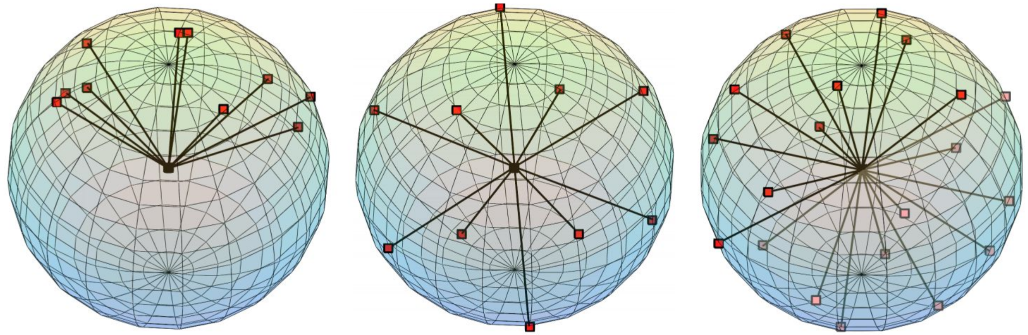
Hyperspherical energy: $\left(\hat{\boldsymbol{w}}_i = \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}\right)$

$$\boldsymbol{E}_{s,d}(\hat{\boldsymbol{w}}_i|_{i=1}^N) = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} f_s\Big(\|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|\Big)$$

$$= \begin{cases} \sum_{i\neq j} \|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|^{-s}, & s>0 \\ \sum_{i\neq j} \log\Big(\|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|^{-1}\Big), & s=0 \end{cases}$$

Minimizing the hyperspherical energy promotes the **diversity** of neurons on a hypersphere.

Minimum hyperspherical energy (MHE):



[1] shows that minimum hyperspherical energy leads to **better generalization**.

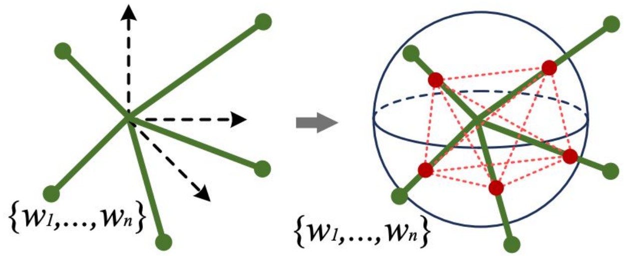
Naively minimizing hyperspherical energy in [1]:

- Higher neuron dimension makes the optimization difficult.
- Highly non-linear and non-convex objective leads to many bad local minima.
- Deterministic gradients from naive MHE is sub-optimal to run away from bad local minima.

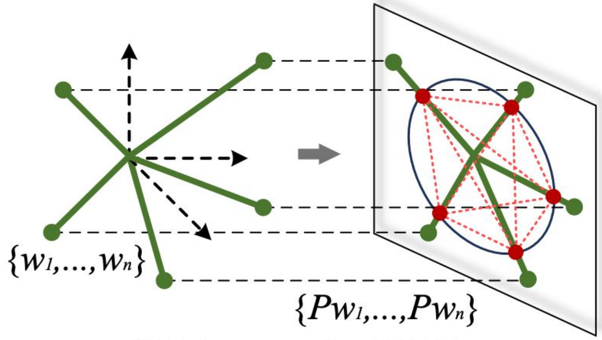
[1] W. Liu*, R. Lin*, Z. Liu*, L. Liu, Z. Yu, B. Dai, L. Song. Learning towards Minimum Hyperspherical Energy, NeurIPS 2018

Compressive Minimum Hyperspherical Energy (CoMHE)

Overview of CoMHE:



(a) Original MHE



(b) Compressive MHE

CoMHE uses projections to reduce the neuron dimension and perform MHE in the projected space.

- Stochastic and dynamic regularization (CoMHE gradients also have stochasticity)
- Low neuron dimension benefits the optimization

Random Projection CoMHE:

The projection matrices $\boldsymbol{P}$ are randomly initialized every certain number of iterations:

$$\boldsymbol{E}_s^R(\hat{\boldsymbol{W}}_N) := \frac{1}{C}\sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} f_s\Big(\Big\|\frac{\boldsymbol{P}_c\hat{\boldsymbol{w}}_i}{\|\boldsymbol{P}_c\hat{\boldsymbol{w}}_i\|}-\frac{\boldsymbol{P}_c\hat{\boldsymbol{w}}_j}{\|\boldsymbol{P}_c\hat{\boldsymbol{w}}_j\|}\Big\|\Big)$$

Angle-preserving Projection CoMHE:

The projections are learned to preserve angles:

$$\boldsymbol{E}_s^A(\hat{\boldsymbol{W}}_N,\boldsymbol{P}^\star) := \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} f_s\Big(\Big\|\frac{\boldsymbol{P}^\star\hat{\boldsymbol{w}}_i}{\|\boldsymbol{P}^\star\hat{\boldsymbol{w}}_i\|}-\frac{\boldsymbol{P}^\star\hat{\boldsymbol{w}}_j}{\|\boldsymbol{P}^\star\hat{\boldsymbol{w}}_j\|}\Big\|\Big)$$

$$\text{s.t.}\quad \boldsymbol{P}^\star = \arg\min_{\boldsymbol{P}}\sum_{i\neq j}\big(\theta_{(\hat{\boldsymbol{w}}_i,\hat{\boldsymbol{w}}_j)}-\theta_{(\boldsymbol{P}\hat{\boldsymbol{w}}_i,\boldsymbol{P}\hat{\boldsymbol{w}}_j)}\big)^2$$

Adversarial Projection CoMHE:

The projections are learned adversarially:

$$\min_{\hat{\boldsymbol{W}}_N}\max_{\boldsymbol{P}} \boldsymbol{E}_s^V(\hat{\boldsymbol{W}}_N,\boldsymbol{P}) := \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} f_s\Big(\Big\|\frac{\boldsymbol{P}\hat{\boldsymbol{w}}_i}{\|\boldsymbol{P}\hat{\boldsymbol{w}}_i\|}-\frac{\boldsymbol{P}\hat{\boldsymbol{w}}_j}{\|\boldsymbol{P}\hat{\boldsymbol{w}}_j\|}\Big\|\Big)$$

Theoretical Guarantees for RP-CoMHE:

$$\frac{\cos(\theta_{(\boldsymbol{w}_1,\boldsymbol{w}_2)})-\epsilon}{1+\epsilon} < \cos(\theta_{(\boldsymbol{P}\boldsymbol{w}_1,\boldsymbol{P}\boldsymbol{w}_2)}) < \frac{\cos(\theta_{(\boldsymbol{w}_1,\boldsymbol{w}_2)})+\epsilon}{1-\epsilon}$$

It holds with probability $\big(1-2\exp(-\frac{k\epsilon^2}{8})\big)^2$

Experiments and Results

Convolutional neural networks (CNN):

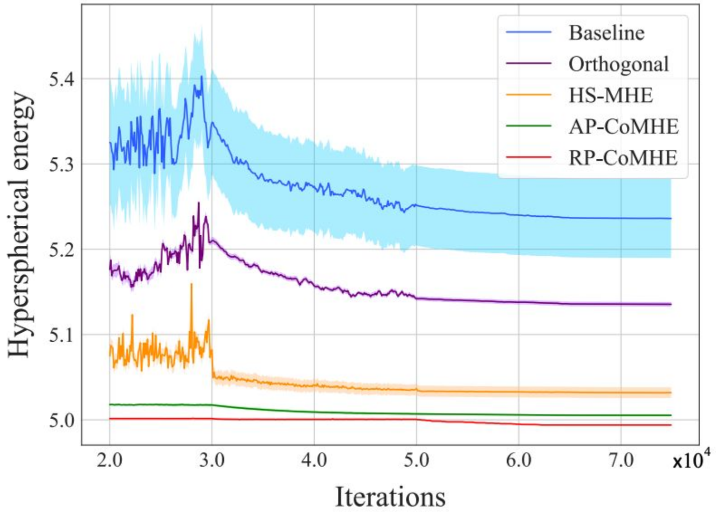
Method	C-10	C-100
ResNet-110 [1]	6.61	25.16
ResNet-1001 [60]	4.92	**22.71**
Baseline	5.19	22.87
Orthogonal [29]	5.02	22.36
SRIP [9]	4.75	22.08
MHE [12]	4.72	22.19
HS-MHE [12]	4.66	22.04
RP-CoMHE	4.59	21.82
AP-CoMHE	**4.57**	**21.63**

CIFAR-10/100

Method	Res-18	Res-34	Res-50
baseline	32.95	30.04	25.30
Orthogonal [29]	32.65	29.74	25.19
Orthnormal [32]	32.61	29.75	25.21
SRIP [9]	32.53	29.55	24.91
MHE [12]	32.50	29.60	25.02
HS-MHE [12]	32.45	29.50	24.98
RP-CoMHE	31.90	29.38	**24.51**
AP-CoMHE	**31.80**	**29.32**	24.53

ImageNet-2012

Hyperspherical energy:



Point cloud networks (PointNet):

Method	PN	PN (T)	PN++
Original	87.1	89.20	90.07
MHE [12]	87.31	89.33	90.25
HS-MHE [12]	87.44	89.41	90.31
RP-CoMHE	87.82	89.69	90.52
AP-CoMHE	**87.85**	**89.70**	**90.56**

ModelNet-40

Different network configurations:

Depth	CNN-6	CNN-9	CNN-15
Baseline	32.08	28.13	N/C
MHE [12]	28.16	26.75	26.90
HS-MHE [12]	27.56	25.96	25.84
RP-CoMHE	26.73	24.39	**24.21**
AP-CoMHE	**26.55**	**24.33**	24.55

Graph convolution networks (GCN):

Method	Citeseer	Cora	Pubmed
GCN Baseline	70.3	81.3	79.0
HS-MHE [12]	71.5	82.0	79.0
RP-CoMHE	**72.1**	**82.7**	**79.5**
AP-CoMHE	72.0	82.6	**79.5**

CoMHE can effectively minimize hyperspherical energy and can improve different types of neural networks. (i.e., CoMHE is **architecture-agnostic**.)

Thank you!

- For any question, please feel free to send emails to rongmei.lin@emory.edu or wyliu@gatech.edu

- Welcome to try CoMHE! The code is made available at https://github.com/rmlin/CoMHE

- For our full paper and related material, please visit https://wyliu.com/