Orthogonal Over-Parameterized Training

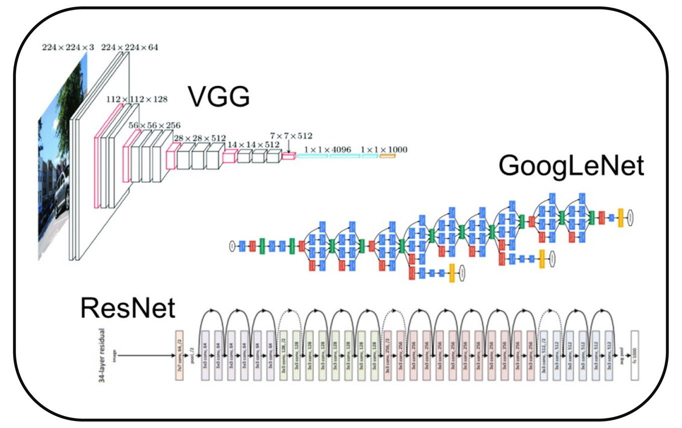
UNIVERSITY OF CAMBRIDGE · EMORY UNIVERSITY · Max Planck Institute for Intelligent Systems · Mila

Weiyang Liu* Rongmei Lin* Zhen Liu James Rehg Liam Paull Li Xiong Le Song Adrian Weller

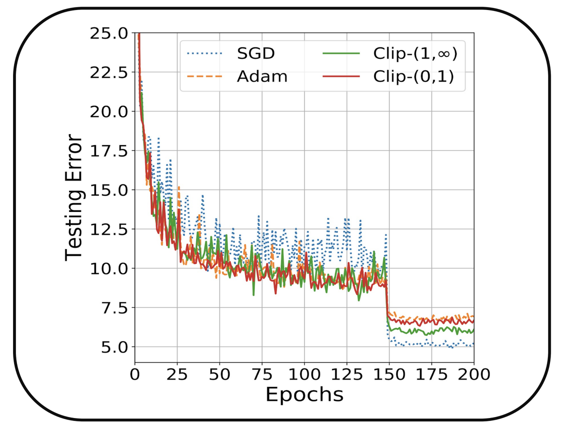
CVPR VIRTUAL JUNE 19-25

Introduction

Empirical Generalization of Neural Networks



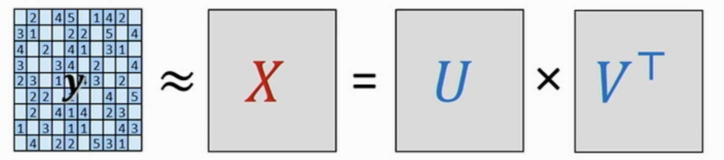
How the network is structured.



How the network is trained.

Motivation I: Over-parameterization

- Recent theories suggest the importance of over-parameterization in linear neural networks.
- For example,

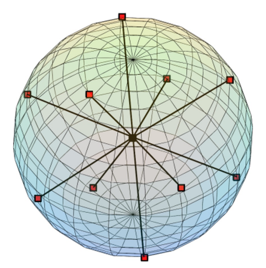


$$\min_{X\in\mathbb{R}^{n\times n}}\|observed(X)-y\|_2^2 \equiv \min_{U,V\in\mathbb{R}^{n\times n}}\|observed(UV^\top)-y\|_2^2$$

- Gradient descent on $f(U,V)$ finds better global minima.
- Gadient decent on $f(U,V)$ yields minimal nuclear norm solution.

Motivation II: Minimum Hyperspherical Energy (MHE)

- Definition of hyperspherical energy:



$$\min_{\{\hat{\boldsymbol{w}}_1,\cdots,\hat{\boldsymbol{w}}_n\in\mathbb{S}^{d-1}\}}\Big\{E_s(\hat{\boldsymbol{W}}_n):=\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}K_s(\hat{\boldsymbol{w}}_i,\hat{\boldsymbol{w}}_j)\Big\}$$

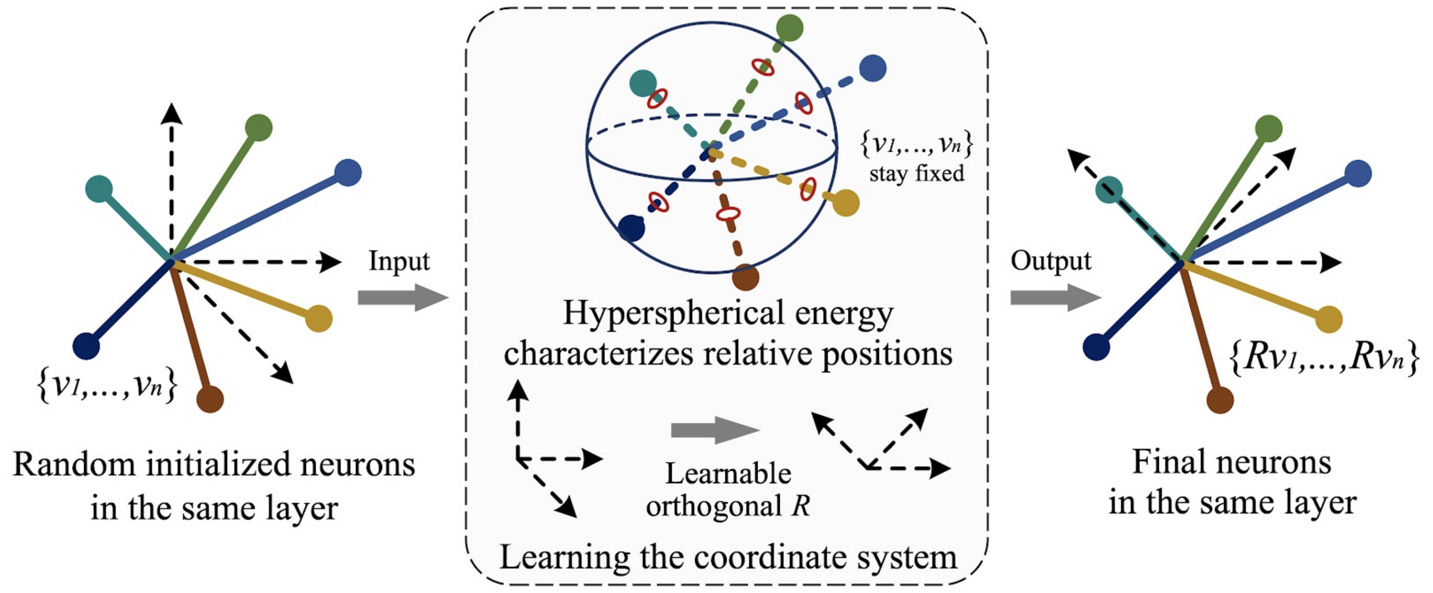
where $\hat{\boldsymbol{w}}_i:=\frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}$ $\quad K_s(\hat{\boldsymbol{w}}_i,\hat{\boldsymbol{w}}_j)=\begin{cases}\rho(\hat{\boldsymbol{w}}_i,\hat{\boldsymbol{w}}_j)^{-s}, & s>0\\ \log(\rho(\hat{\boldsymbol{w}}_i,\hat{\boldsymbol{w}}_j)^{-1}), & s=0\\ -\rho(\hat{\boldsymbol{w}}_i,\hat{\boldsymbol{w}}_j)^{-s}, & s<0\end{cases}$

- ρ denotes either Euclidean distance or angular distance on the unit hypersphere.
- Hyperspherical energy characterizes the neuron diversity of the neural network.
- Previous work shows that lower hyperspherical energy leads to better empirical generalization

Goal: A Principled Training Framework for Neural Networks

- Make use of the over-parameterization within each neuron
- Naturally guarantee the minimum hyperspherical energy
- Compatible to different network architectures and optimizers

Orthogonal Over-Parameterized Training (OPT)



- As an illustrative example, we consider a two-layer MLP:

$$\min_{\{\boldsymbol{R},u_i,\forall i\}}\sum_{j=1}^{m}\mathcal{L}\Big(y,\sum_{i=1}^{n}u_i(\boldsymbol{R}\boldsymbol{v}_i)^\top\boldsymbol{x}_j\Big)\quad\text{s.t. }\boldsymbol{R}^\top\boldsymbol{R}=\boldsymbol{R}\boldsymbol{R}^\top=\boldsymbol{I}$$

Orthogonal Matrix — What OPT learns

Neuron Weight — Randomly initialized and fixed

- Note that in OPT, neurons are randomly initialized and then fixed throughout the entire training. Only the layer-shared orthogonal matrices are learned.
- If each element of the neuron is initialized with zero-mean Gaussian, e.g., Xavier/Kaiming initialization, then we have that

OPT can provably guarantee the minimum hyperspherical energy.

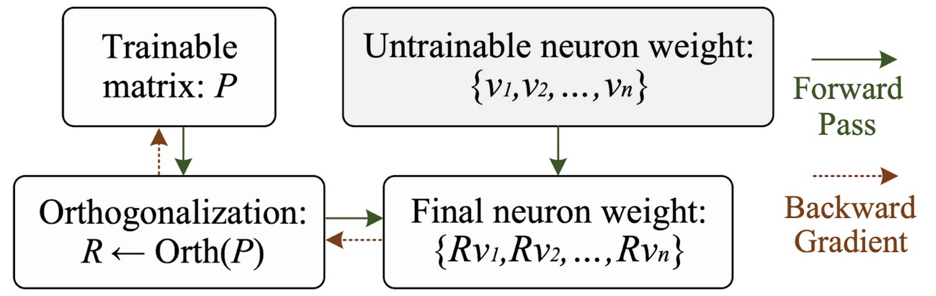
$$\boldsymbol{E}(\hat{\boldsymbol{R}\boldsymbol{v}}_i|_{i=1}^n)=\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\|\boldsymbol{R}\hat{\boldsymbol{v}}_i-\boldsymbol{R}\hat{\boldsymbol{v}}_j\|^{-1}$$

$$(\text{since }\|\boldsymbol{R}\|^{-1}=1)\quad=\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\|\hat{\boldsymbol{v}}_i-\hat{\boldsymbol{v}}_j\|^{-1}=\boldsymbol{E}(\hat{\boldsymbol{v}}_i|_{i=1}^n)$$

- Randonly initialized neurons provably lead to small energy.
- OPT does not change the hyperspherical energy during training.

Ways to guarantee orthogonality

- Unrolling orthogonalization algorithms:
  - ➢ Gram-Schmidt Process
  - ➢ Householder reflection
  - ➢ Lowdin's Symmetric Orthogonalization



- Orthogonal parameterization:

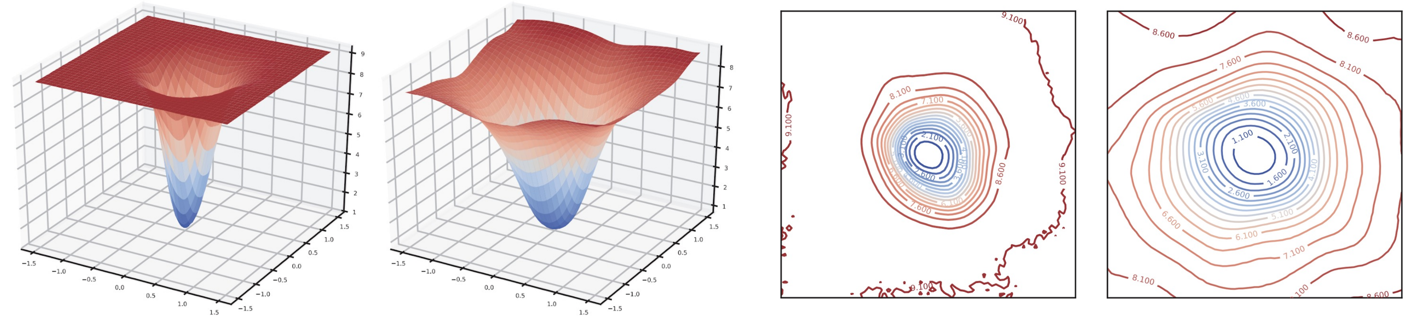
$$\boldsymbol{R}=(\boldsymbol{I}+\boldsymbol{W})(\boldsymbol{I}-\boldsymbol{W})^{-1}\qquad \boldsymbol{W}=-\boldsymbol{W}^\top$$

- Orthogonality-preserving gradient descent
- Orthogonality as a regularization:

$$\min_{\boldsymbol{R},u_i,\forall i}\sum_{j=1}^{m}\mathcal{L}\Big(y,\sum_{i=1}^{n}u_i(\boldsymbol{R}\boldsymbol{v}_i)^\top\boldsymbol{x}_j\Big)+\beta\|\boldsymbol{R}^\top\boldsymbol{R}-\boldsymbol{I}\|_F^2$$

Intriguing Insights and Extension

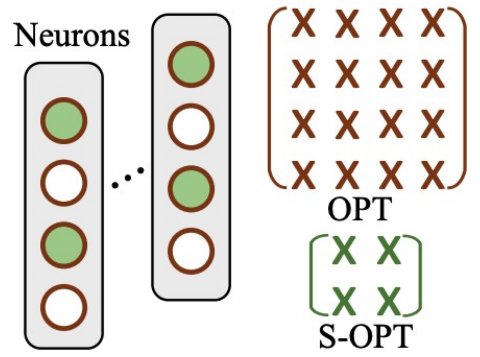
Loss Landscape Comparison



Standard Training OPT Standard Training OPT

Stochastic OPT for Better Scalability

- Approximate a large orthogonal matrix with many small orthogonal ones.
- Similar to the idea of DropOut.
- Randomly selecting a subset of the neuron dimensions in each iteration and perform OPT on this subset.



Experiments and Results

Ablation and exploratory experiments

Ablation

Method	FN	LR	CNN-6	CNN-9
Baseline	-	-	37.59	33.55
UPT	✗	U	48.47	46.72
UPT	✓	U	42.61	39.38
OPT	✗	GS	37.24	32.95
OPT	✓	GS	**33.02**	**31.03**

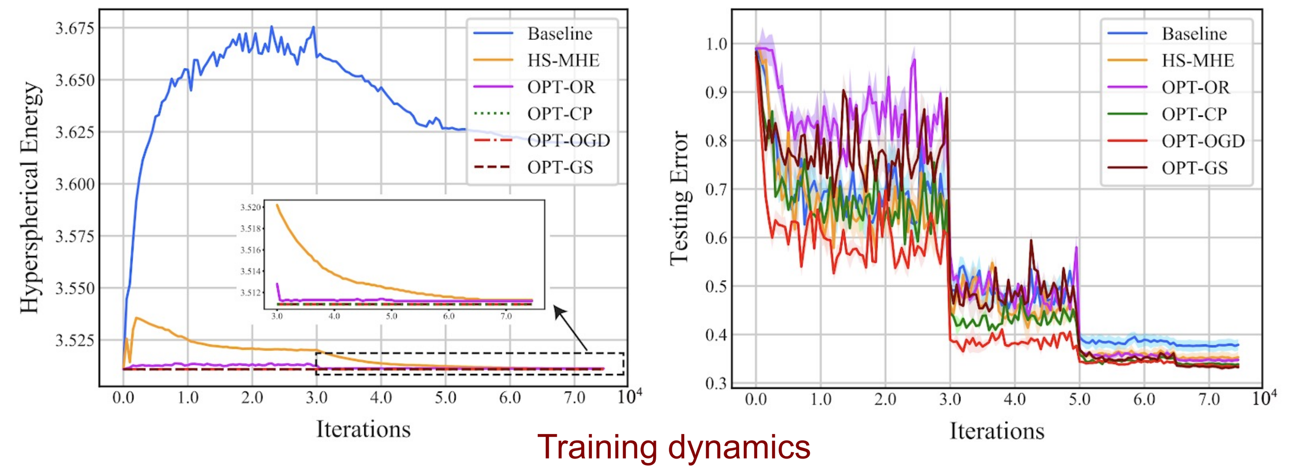
CIFAR-100

FN: whether neurons are fixed after initialization
LR: whether we enforce orthogonality on $\boldsymbol{R}$.

Different initial energy

Mean	Energy	Error (%)
0	**3.5109**	**32.49**
1e-3	3.5117	33.11
1e-2	3.5160	39.51
2e-2	3.5531	53.89
3e-2	3.6761	N/C

CIFAR-100



Training dynamics

Results on OPT and Stochastic OPT

OPT for MLP, plain CNN and ResNet

Method	MNIST		CIFAR-100			
	MLP-N	MLP-X	CNN-6	CNN-9	ResNet-20	ResNet-32
Baseline	6.05	2.14	37.59	33.55	31.11	30.16
Orthogonal [7]	5.78	1.93	36.32	33.24	31.06	30.05
SRIP [4]	-	-	34.82	32.72	30.89	29.70
HS-MHE [49]	5.57	1.88	34.97	32.87	30.98	29.76
OPT (GS)	**5.11**	**1.45**	**33.02**	**31.03**	30.49	29.34
OPT (HR)	5.31	1.60	35.67	32.75	30.73	29.56
OPT (LS)	5.32	1.54	34.48	31.22	30.51	29.42
OPT (CP)	5.14	1.49	33.53	31.28	**30.47**	**29.31**
OPT (OGD)	5.38	1.56	33.33	31.47	30.50	29.39
OPT (OR)	5.41	1.78	34.70	32.63	30.66	29.47

OPT for PointNet

Method	GCN		PointNet
	Cora	Pubmed	MN-40
Baseline	81.3	79.0	87.1
OPT (GS)	81.9	79.4	87.23
OPT (CP)	82.0	79.4	87.81
OPT (OGD)	**82.3**	**79.5**	**87.86**

OPT for few-shot learning

Method	5-shot Acc. (%)
MAML [13]	62.71 ± 0.71
MatchingNet [70]	63.48 ± 0.66
ProtoNet [65]	64.24 ± 0.72
Baseline [9]	62.53 ± 0.69
Baseline w/ OPT	**63.27 ± 0.68**
Baseline++ [9]	66.43 ±0.63
Baseline++ w/ OPT	**66.82 ± 0.62**

S-OPT for plain CNN and ResNet

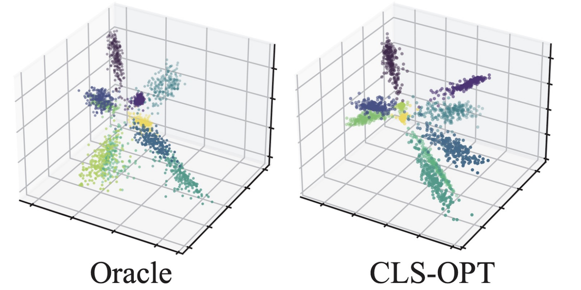
Method	CIFAR-100				ImageNet	
	CNN-6	Params	Wide CNN-9	Params	ResNet-18	Params
Baseline	37.59	258K	28.03	2.99M	32.95	11.7M
HS-MHE [49]	34.97	258K	25.96	2.99M	32.50	11.7M
OPT (GS)	**33.02**	1.36M	OOM	16.2M	OOM	46.5M
S-OPT (GS)	33.70	**90.9K**	**25.59**	**1.04M**	**32.26**	**3.39M**

Sampling dimension for S-OPT

$p=$	Error (%)	Params
d	OOM	16.2M
$d/4$	**25.59**	1.04M
$d/8$	28.61	278K
$d/16$	32.52	88.7K
16	33.03	27.0K
3	45.22	26.0K
0	60.64	**25.6K**

Large Categorical Training

- An interesting application is to apply OPT to the classifier layer (as comparison, the other layers are trained normally)
- It enables scalable categorical training (many classes)



Oracle CLS-OPT

ImageNet (1K classes)

Method	ResNet-18A		ResNet-18B	
	Error	Params	Error	Params
Oracle	**18.08**	64.0K	12.12	512K
CLS-OPT	21.12	**8.13K**	**12.05**	**131K**

CASIA-WebFace (10K classes)

Method	512 Dim.		1024 Dim.	
	Acc.	Params	Acc.	Params
Oracle	**95.7**	5.41M	**96.4**	10.83M
CLS-OPT	94.9	**131K**	95.8	**524K**