



Max Planck Institute for
Intelligent Systems



UNIVERSITY OF
CAMBRIDGE



EMORY
UNIVERSITY



Mila



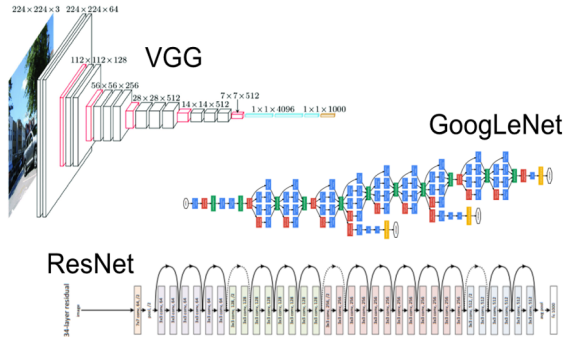
Orthogonal Over-parameterized Training

Weiyang Liu*, Rongmei Lin*, Zhen Liu

James Rehg, Liam Paull, Li Xiong, Le Song, Adrian Weller

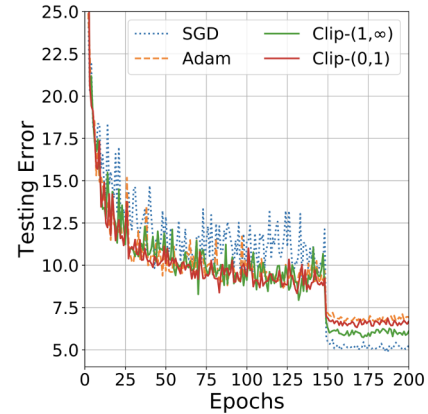
Empirical Generalization of Neural Networks

How the network is structured.



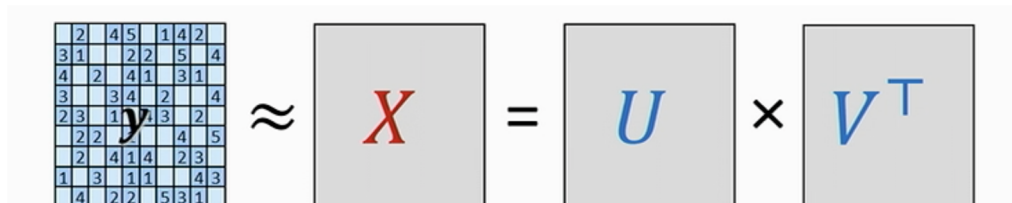
Optimization landscape

How the network is trained.

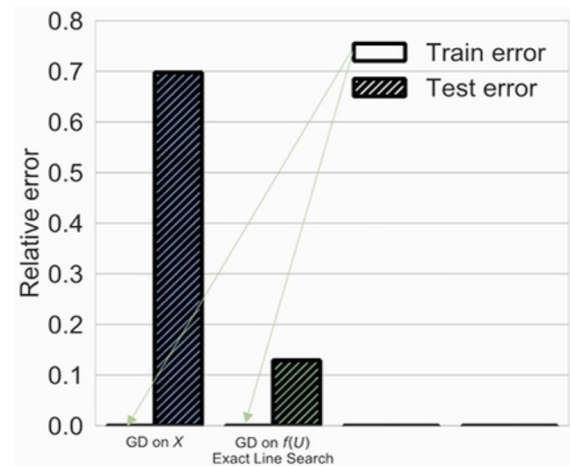


Optimization path

Motivation I: Over-parameterization (1)



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

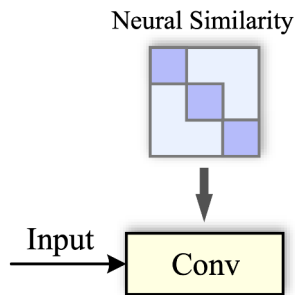


- Gradient descent on $f(U, V)$ finds better global minima.
- Gradient descent on $f(U, V)$ yields minimal nuclear norm solution.

Motivation I: Over-parameterization (2)

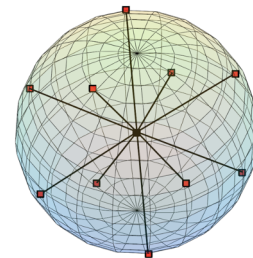
- In neural network, what if we directly replace all the inner products with the following bilinear form:

$$f_{\mathcal{M}}(\mathbf{W}, \mathbf{X}) = \mathbf{W}^{\top} \mathbf{M} \mathbf{X}$$



Method	CIFAR-10	CIFAR-100
Baseline CNN	7.78	28.95
Baseline CNN++	7.29	28.70
Static NSN w/ DNS	7.15	28.35
Static NSN w/ UNS	7.38	28.11
Dynamic NSN w/ DNS	6.85	27.81
Dynamic NSN w/ UNS	6.5	28.02

Motivation II: Minimum Hyperspherical Energy



- Hyperspherical Uniformity

- Hyperspherical energy characterizes the neuron diversity on the unit hypersphere:

$$\mathcal{L}_{\text{reg}} = \underbrace{\lambda_w \cdot \frac{1}{\sum_{j=1}^L N_j} \sum_{j=1}^L \sum_{i=1}^{N_j} \|\mathbf{w}_i\|}_{\text{Weight decay: regularizing the magnitude of kernels}} + \underbrace{\lambda_h \cdot \sum_{j=1}^{L-1} \frac{1}{N_j(N_j - 1)} \{\mathbf{E}_s\}_j + \lambda_o \cdot \frac{1}{N_L(N_L - 1)} \mathbf{E}_s(\hat{\mathbf{w}}_i^{\text{out}}|_{i=1}^c)}_{\text{MHE: regularizing the direction of kernels}}$$

$$\mathbf{E}_{s,d}(\hat{\mathbf{w}}_i|_{i=1}^N) = \sum_{i=1}^N \sum_{j=1, j \neq i}^N f_s(\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_j\|) = \begin{cases} \sum_{i \neq j} \|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_j\|^{-s}, & s > 0 \\ \sum_{i \neq j} \log(\|\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_j\|^{-1}), & s = 0 \end{cases}$$

Sum of pair-wise similarity

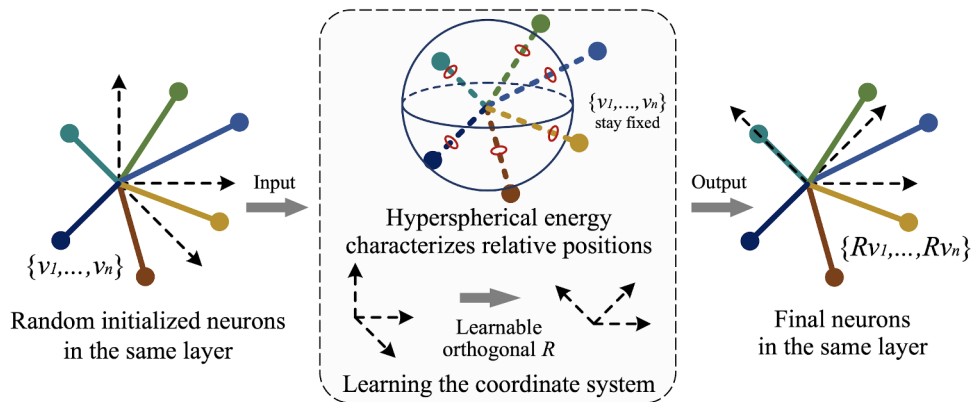
- Lower hyperspherical energy leads to better empirical generalization.

A principled training framework for neural networks

- Making use of the **over-parameterization** from linear matrix multiplication
- Naturally guarantee the **minimum hyperspherical energy**
- **Compatible** with different network architectures and different optimizers

Orthogonal Over-parameterized Training

- If we initialize each element of the neuron with zero-mean Gaussian, then we can prove that it follows uniform distribution on the hypersphere.
- Our framework:



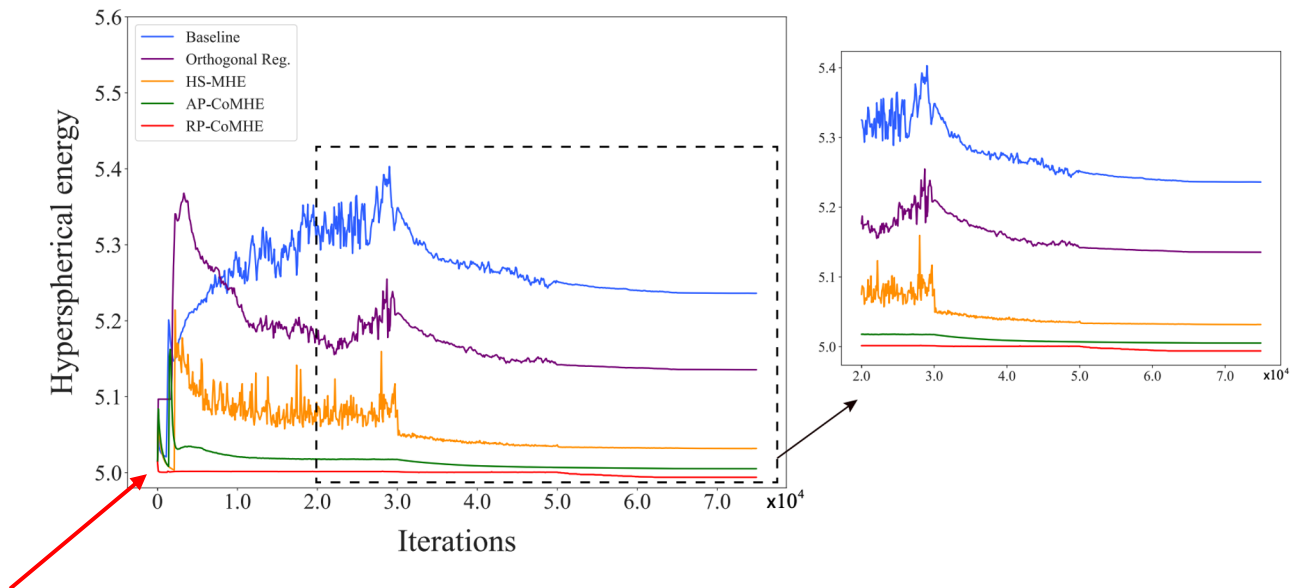
$$\min_{\{\mathbf{R}, u_i, \forall i\}} \sum_{j=1}^m \mathcal{L}(y, \sum_{i=1}^n u_i (\mathbf{R}v_i)^\top \mathbf{x}_j) \quad \text{s.t. } \mathbf{R}^\top \mathbf{R} = \mathbf{R}\mathbf{R}^\top = \mathbf{I}$$

Orthogonal
matrix

Neuron
weight

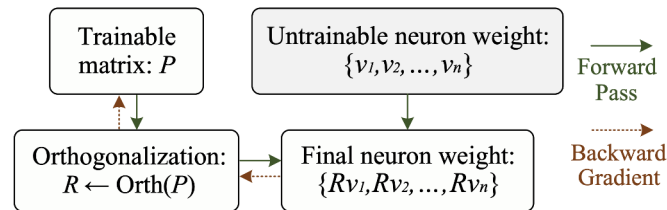
Why orthogonal?

- Training dynamics of hyperspherical energy



Even the baseline CNN starts with a very small hyperspherical energy!

Ways to guarantee orthogonality



- Unrolling orthogonalization algorithms, eg., Gram-Schmidt Process, Householder reflection, Lowdin's Symmetric Orthogonalization

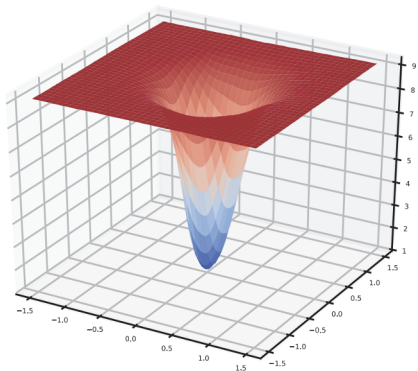
- Orthogonal parameterization: $\mathbf{R} = (\mathbf{I} + \mathbf{W})(\mathbf{I} - \mathbf{W})^{-1} \quad \mathbf{W} = -\mathbf{W}^\top$

- Orthogonality-preserving gradient descent (a special case of manifold gradient decent)

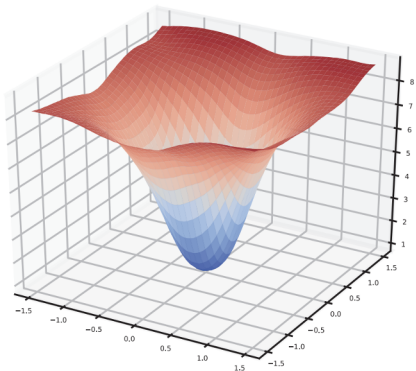
- Orthogonality as a regularization:
$$\min_{\mathbf{R}, u_i, \forall i} \sum_{j=1}^m \mathcal{L}(y, \sum_{i=1}^n u_i (\mathbf{R}v_i)^\top \mathbf{x}_j) + \beta \|\mathbf{R}^\top \mathbf{R} - \mathbf{I}\|_F^2$$

Intriguing Insights and guarantees

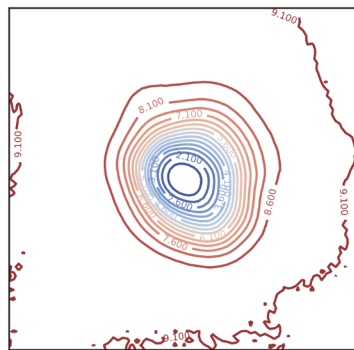
- More convex and smooth loss landscape



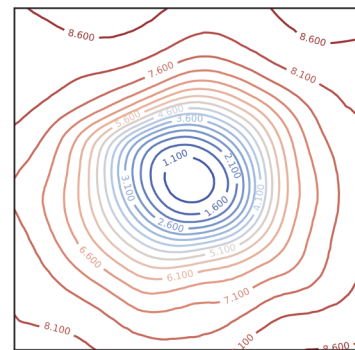
Standard Training



OPT



Standard Training



OPT

- Provable generalization: the training loss is on the order of the square norm of the gradient and the generalization error will have an additional term $\tilde{O}(1/\sqrt{m})$

Ablation and exploratory experiments

CIFAR-100

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

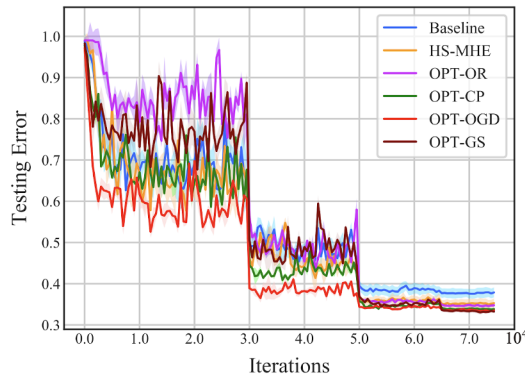
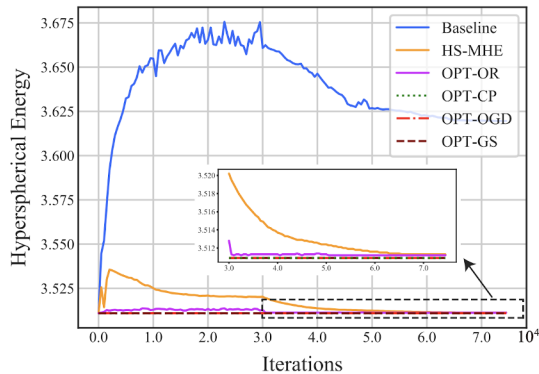
CIFAR-100

Mean Energy	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

FN: whether neurons are fixed after initialization

LR: whether we enforce orthogonality on R .

Different initial energy



Training dynamics

Results on training different neural networks

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

Training convolutional neural networks

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

Training graph convolution networks and point cloud networks

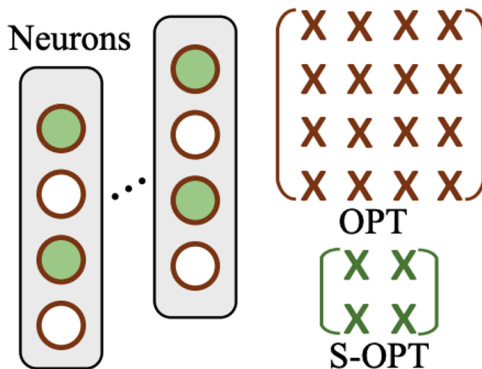
Method	5-shot Acc. (%)
MAML [13]	62.71 \pm 0.71
MatchingNet [70]	63.48 \pm 0.66
ProtoNet [65]	64.24 \pm 0.72
Baseline [9]	62.53 \pm 0.69
Baseline w/ OPT	63.27 \pm 0.68
Baseline++ [9]	66.43 \pm 0.63
Baseline++ w/ OPT	66.82 \pm 0.62

Training few-shot learning networks

What if the orthogonal matrix gets too large?

- Stochastic Orthogonal Over-parameterized Training

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



Experimental Results

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

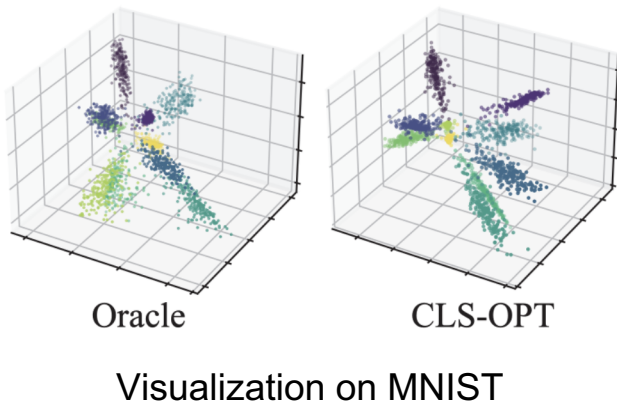
Training convolutional neural networks

$p =$	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

Sampling dimension

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)



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

ImageNet (1K 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

CASIA-WebFace (10K classes)