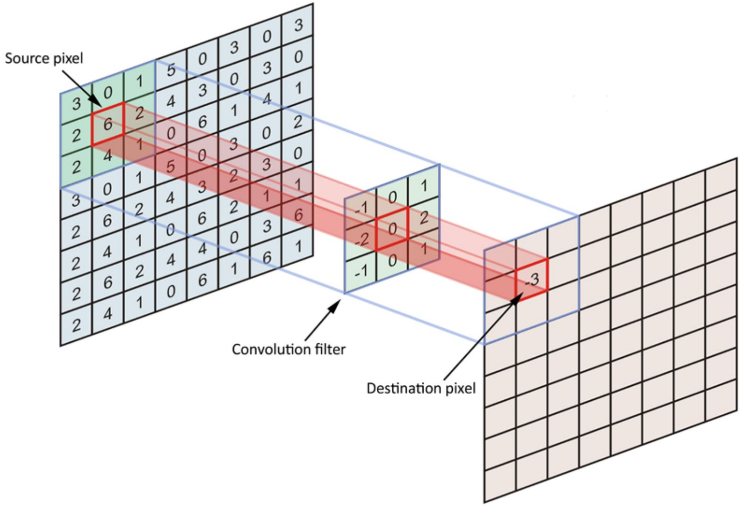
Decoupled Networks

Weiyang Liu*, Zhen Liu*, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M. Rehg, Le Song

(* indicates equal contributions)

Motivation I

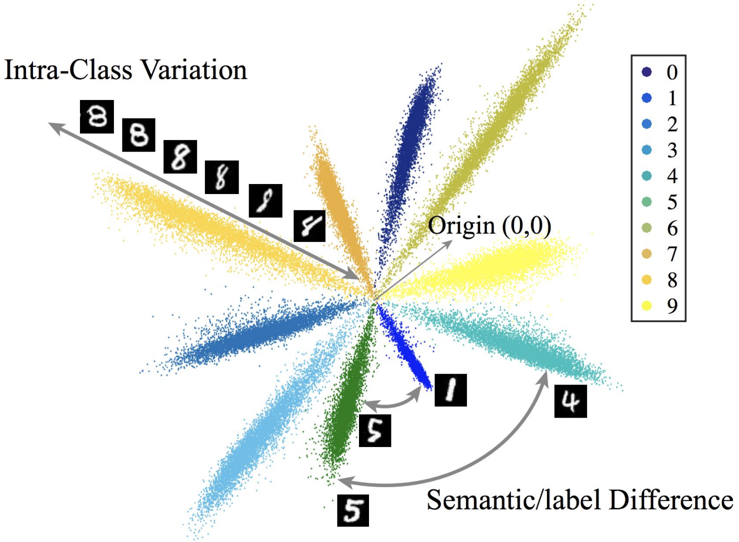
- Convolution works well by incorporating our prior knowledge on images (compared to MLP).
- Specifically, it computes inner product-based similarity in a sliding window fashion.



Questions:

- Why should inner product be the optimal similarity measure (despite the simplicity)?
- Is there better similarity prior for specific tasks?

Motivation II



- The original inner product based convolution operator learns naturally *decoupled* features.

- **norm** of feature = **intra-class variation**
- **angle** of feature = **semantic difference**

- We can explicitly model both intra-class variation and semantic difference with *decoupled* convolution framework.

Decoupled Convolution

- Original Inner Product-based Convolution

$$< w, x > = ||w|| \cdot ||x|| \cdot cos(\theta_{w,x})$$

naturally decoupled

- Decoupled Convolution

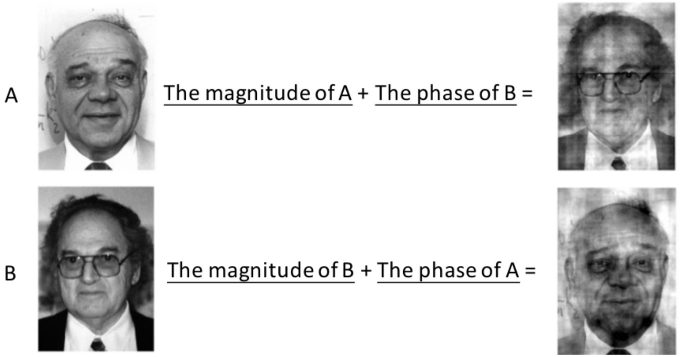
$$f(w, x) = \boxed{h(\|w\|, \|x\|)} \cdot \boxed{g(\theta_{w,x})}$$

Magnitude: intra-class variation

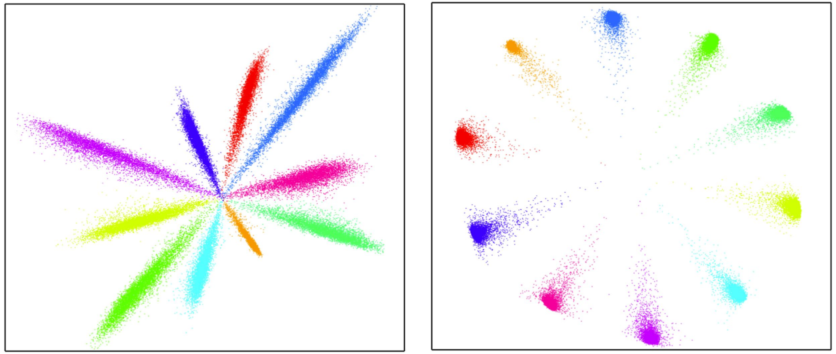
Angle: semantic difference

Case Study: Hyperspherical Convolution

- Hyperspherical Convolution (SphereConv) models the norm by $h(\|\boldsymbol{w}\|,\|\boldsymbol{x}\|)=\alpha\cdot$
- SphereConv ignores intra-class variation by discarding all magnitude information.



Intuition: The identity information is preserved by phase in 2D Fourier transform.



2D feature visualization on MNIST shows that SphereConv can compress the intra-class variation.

Example Designs

Magnitude (norm) Functions

- SphereConv - Project onto unit sphere

$$h(\|\boldsymbol{w}\|, \|\boldsymbol{x}\|) = \alpha$$

- BallConv - Project onto unit ball

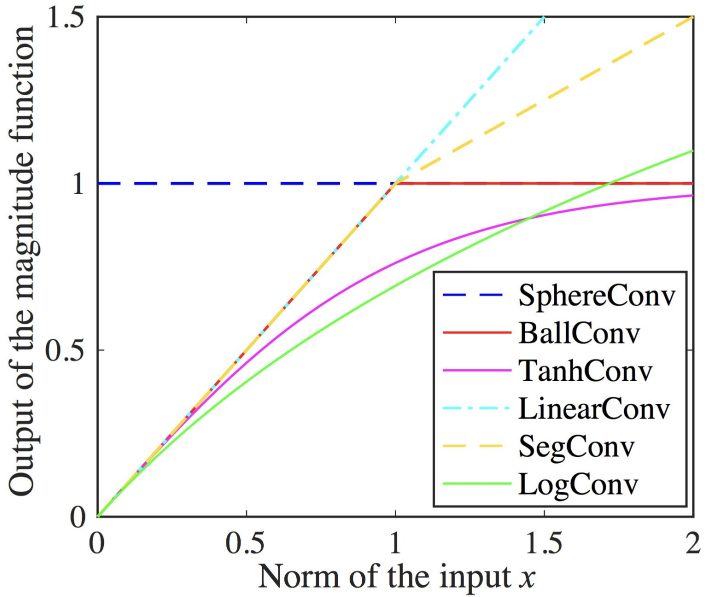
$$h(\|\boldsymbol{w}\|, \|\boldsymbol{x}\|) = \alpha \min(\|\boldsymbol{x}\|, \rho)/\rho$$

- TanhConv - Soft BallConv

$$h(\|\boldsymbol{w}\|, \|\boldsymbol{x}\|) = \alpha \tanh\left(\frac{\|\boldsymbol{x}\|}{\rho}\right)$$

- LinearConv - Only project weights

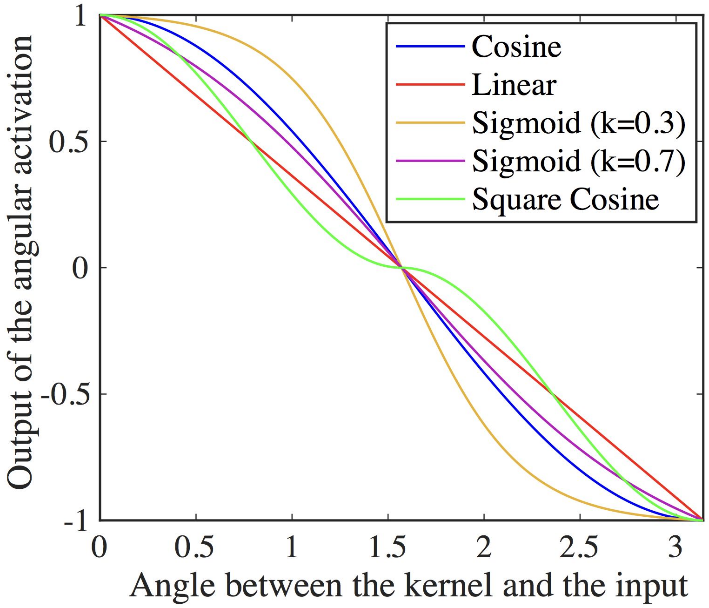
$$h(\|\boldsymbol{w}\|, \|\boldsymbol{x}\|) = \alpha \|\boldsymbol{x}\|$$



Example Designs

Angular Functions

- Square Cosine
$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = \mathrm{sign}(\cos(\theta)) \cdot \cos^2(\theta)$$
- Linear
$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = -\frac{2}{\pi}\theta_{(\boldsymbol{w},\boldsymbol{x})} + 1$$
- Cosine
$$g(\theta_{(\boldsymbol{w},\boldsymbol{x})}) = \cos(\theta_{(\boldsymbol{w},\boldsymbol{x})})$$
- Sigmoid (refer to the paper)



Properties of the magnitude function

- Smoothness: whether the magnitude function is differentiable everywhere. It affects the optimization of the neural network.

- Boundedness: whether the value of the magnitude function is bounded. It affects the adversarial robustness of the neural network (due to the Lipschitz constant).

Special Case: Bounded Magnitude Function

When the magnitude function is bounded, it has the following advantage:

- Adversarial Robustness: Bounded operators have **smaller Lipschitz constant**, which is more robust against adversarial attacks
- Optimization: Boundedness can improve optimization, which is both theoretically and empirically validated by some recent results*.

* [1] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, Aleksander Mądry. How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift), arXiv:1805.11604.
[2] Liu, et al. Deep Hyperspherical Learning, NIPS 2017.

CNN without BN or without ReLU on CIFAR-100

- Operators can significantly outperform baselines w/o BN

Method	Linear	Cosine	Sq. Cosine
CNN Baseline	-	35.30	-
LinearConv	33.39	31.76	N/C
TanhConv	**32.88**	31.88	**34.26**
SegConv	34.69	**30.34**	N/C

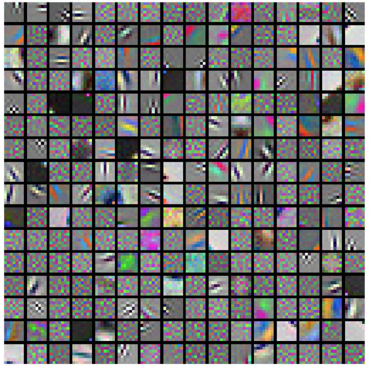
- Operators can significantly outperform baselines w/o ReLU

Method	Cosine w/o ReLU	Sq. Cosine w/o ReLU	Cosine w/ ReLU	Sq. Cosine w/ ReLU
Baseline	58.24	-	26.01	-
SphereConv	33.31	25.90	26.00	26.97
BallConv	**31.81**	25.43	25.18	26.48
TanhConv	32.27	25.27	25.15	26.94
LinearConv	36.49	24.36	**24.81**	25.14
SegConv	33.57	**24.29**	24.96	**25.04**
LogConv	33.62	24.91	25.17	25.85
MixConv	33.46	24.93	25.27	25.77

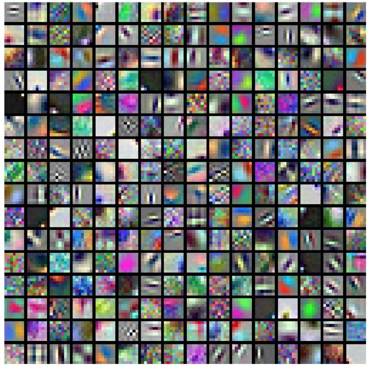
ImageNet Results

- Decoupled operators are exceptionally good on specific architectures.

Method	Standard ResNet-18 w/ BN	Modified ResNet-18 w/ BN	Modified ResNet-18 w/o BN
Baseline	12.63	12.10	N/C
SphereConv	12.68*	11.55	13.30
LinearConv	**11.99***	11.50	N/C
TanhConv	12.47*	**11.10**	**12.79**



DCNet (SphereConv)



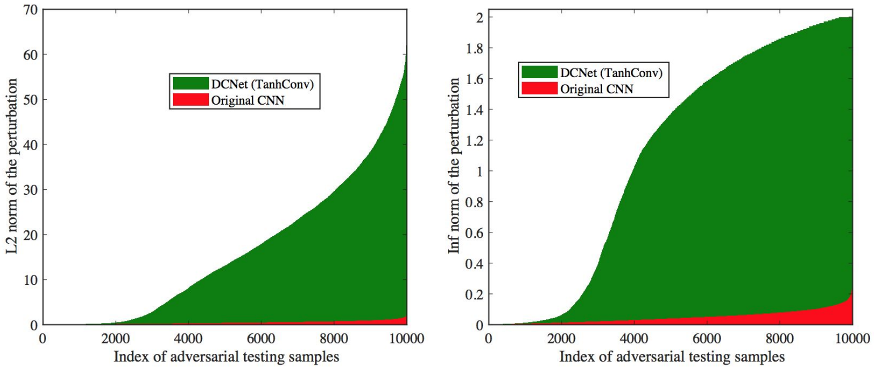
DCNet (TanhConv)

Robustness Against Adversarial Attacks

- Our operators are naturally robust against attacks.

	Attack	Target models: Baseline	SphereConv	BallConv	TanhConv
Natural Training	None	85.35	88.58	91.13	**91.45**
	FGSM	18.82	43.64	50.47	**52.60**
	BIM	8.67	8.89	7.74	**10.18**
Adversarial Training	None	83.70	87.41	87.47	**87.54**
	FGSM	78.96	**85.98**	82.20	81.46
	BIM	7.96	**35.07**	17.38	19.86

- To attack decoupled networks, it takes much more efforts (L2 norm).



From Function Approximation perspective

- Convolutional neural networks (CNNs) can be viewed as special cases of feed-forward neural networks (or multi-layer perceptrons).

- Although MLPs can achieve the universal approximation, but in practice, CNNs usually work much better than MLPs in vision tasks.

From Function Approximation perspective

- Implication I: which nonlinear function class we use really matters (even if they can be equivalent in theory).
- Implication II: the function class we use should be task-dependent. learning/regularizing/constraining the function class for different tasks is appealing.
- Implication III: **our decoupled convolution can be viewed as regularizing the function class of CNNs by using a different similarity measure.**

The End

Thank you!

Welcome to our poster for further questions!

F21, Tuesday 4:30-6:30 @ Halls C-E