Deep Isometric Learning for Visual Recognition

Chong You

University of California, Berkeley

Jointly with:


Haozhi Qi

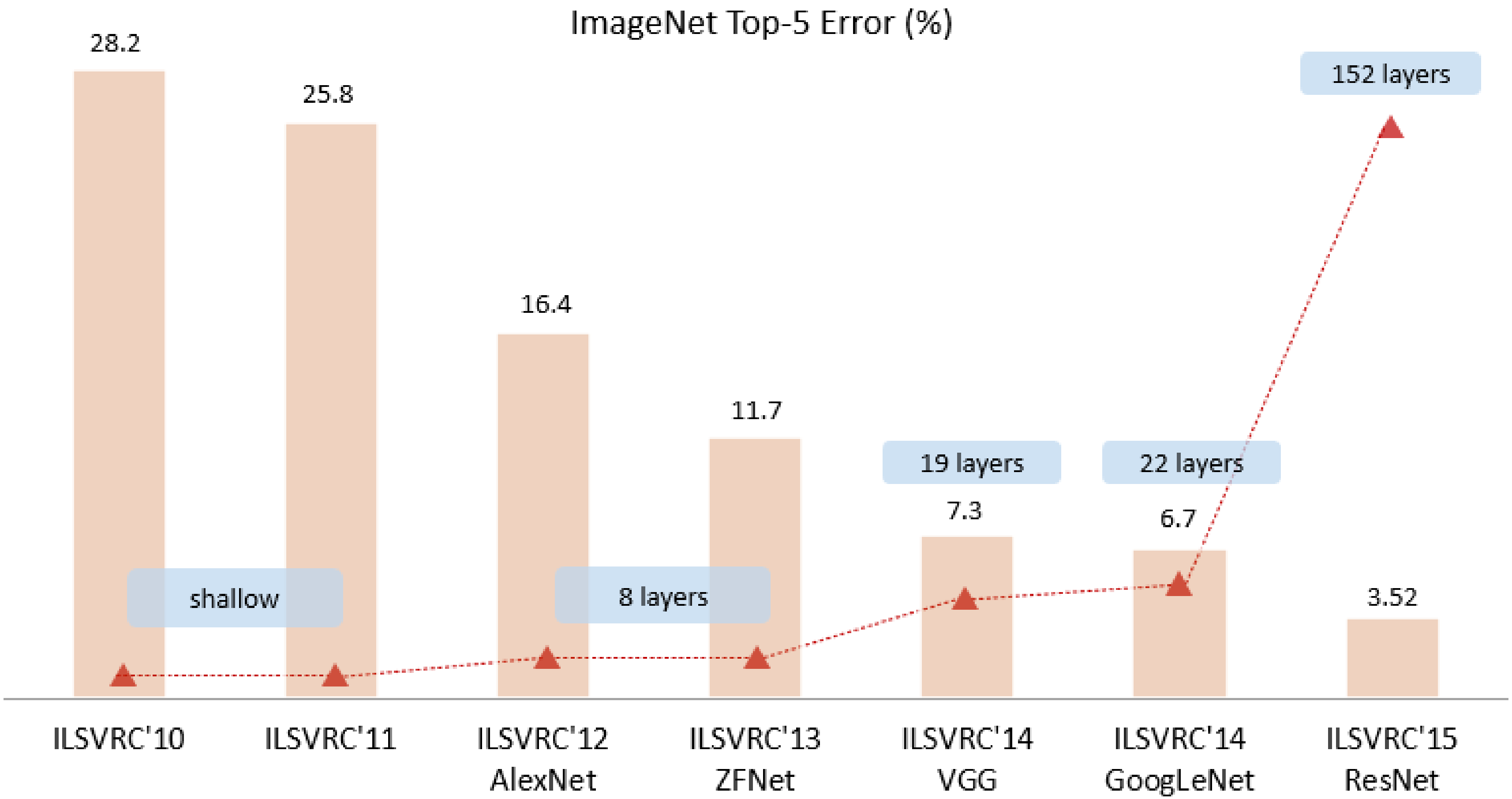

Xiaolong Wang


Yi Ma


Jitendra Malik

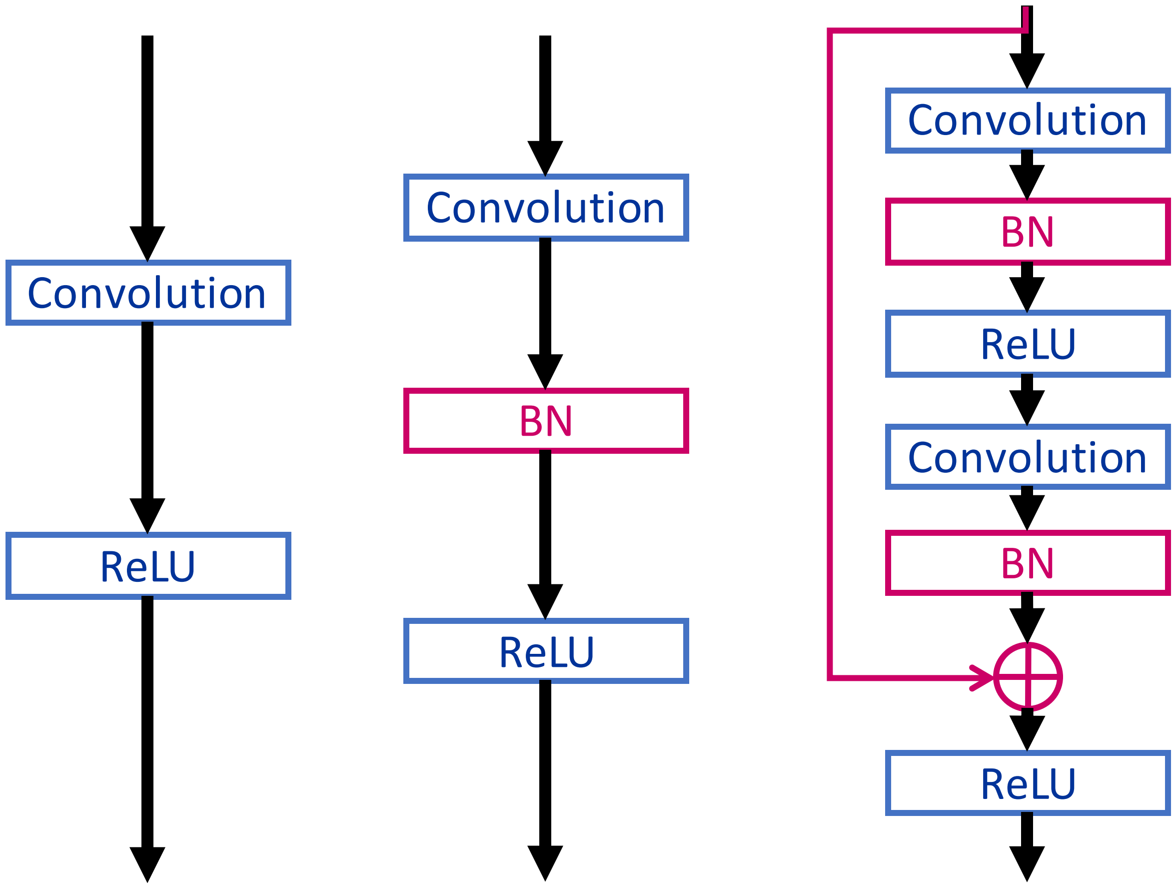
Benefit of Depth

- We make progress by gaining the ability to train very deep neural networks



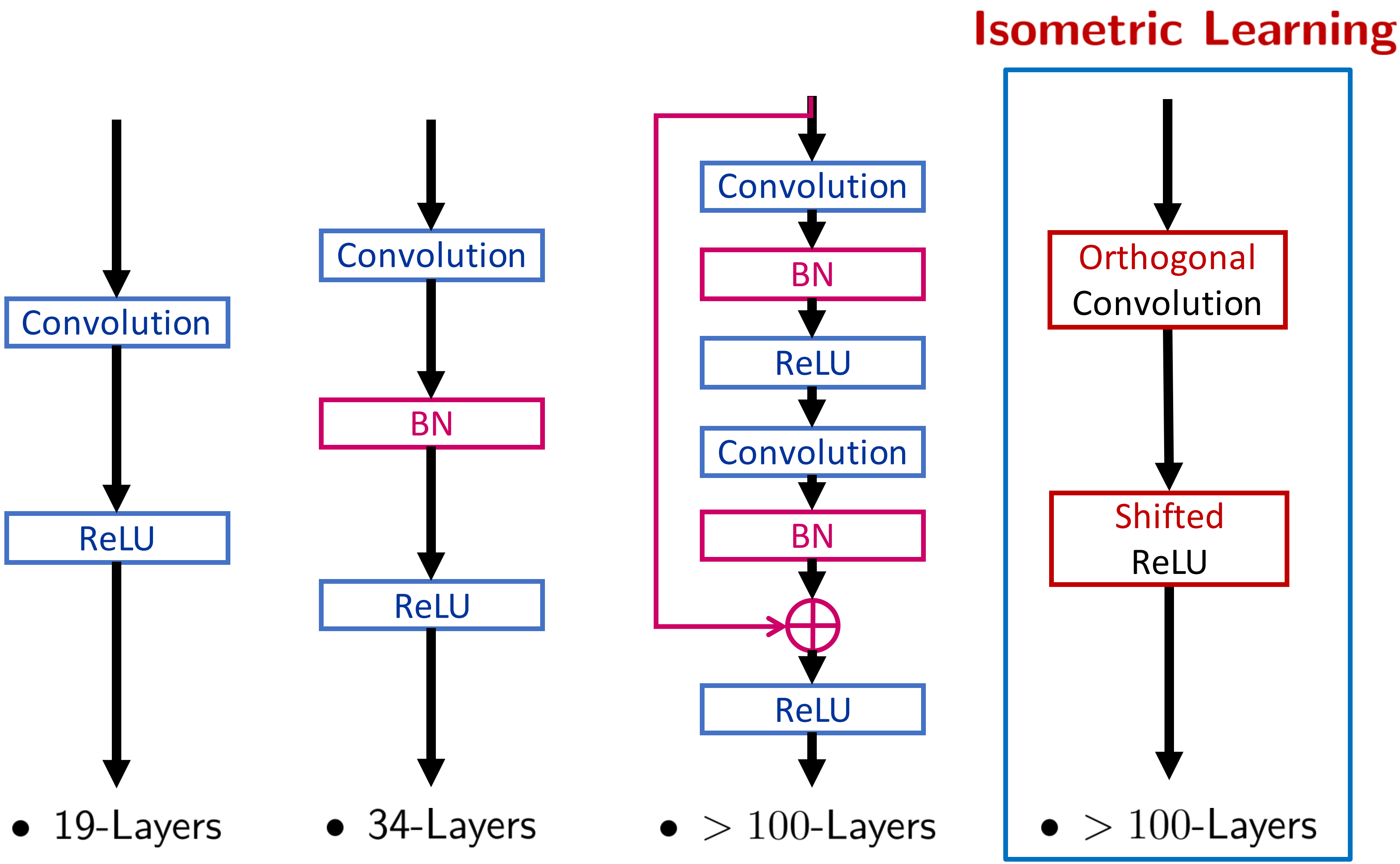
- Deep Residual Learning for Image Recognition, IEEE Conference on Computer Vision and Pattern Recognition 2016

How to Train a Deep Network?



- 19-Layers
- 34-Layers
- >100-Layers

Contribution: Isometric Learning



- With isometry, deep networks can be trained without BN and shortcut

Outline

- **(Conceptually)** What enables training very deep neural networks?

  Isometric Network (ISONet): Training 101-layer *vanilla* ConvNets (i.e., conv & nonlinear layers only) with $> 70\%$ accuracy on ImageNet

- **(Practically)** How to design better neural network architectures?

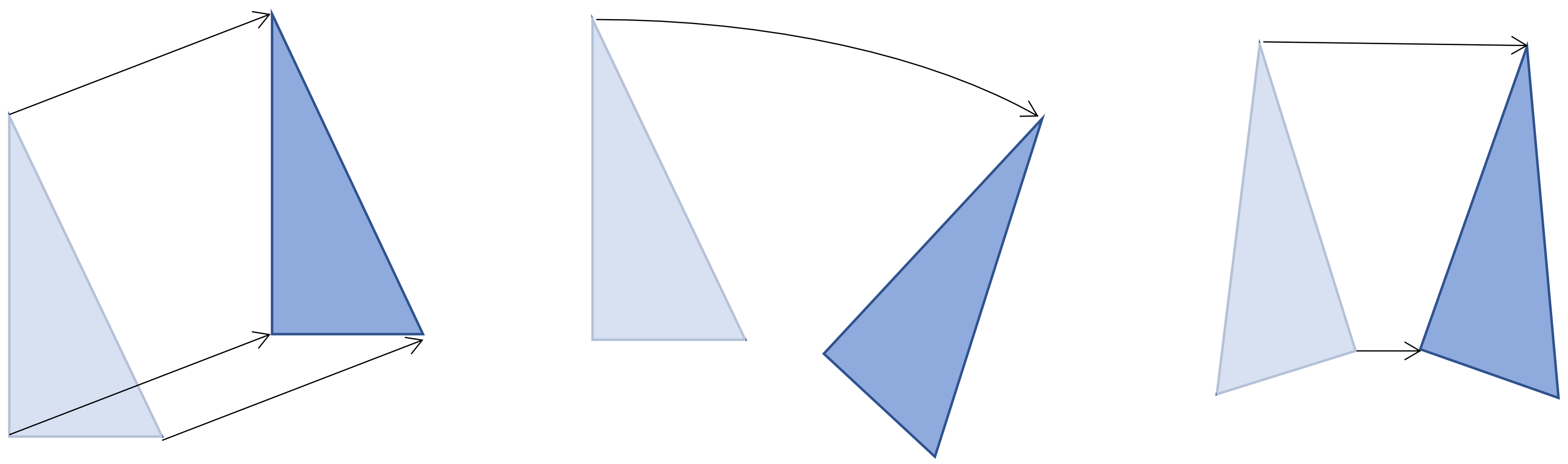
  Residual ISONet (R-ISONet): 1) SOTA performance on ImageNet without BatchNorm, 2) Better transfer ability for object detection

Isometry

Definition. A map $\mathcal{A} : \mathbb{R}^C \to \mathbb{R}^M$ is called an isometry if

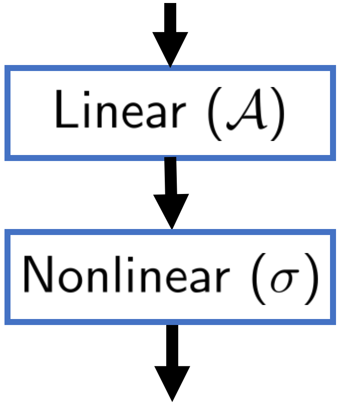
$$\langle \mathcal{A}\boldsymbol{x}, \mathcal{A}\boldsymbol{x}' \rangle = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle, \;\; \forall \{\boldsymbol{x}, \boldsymbol{x}'\} \subseteq \mathbb{R}^C.$$

In words, isometry preserves distances and angles between a preimage and its image



Translation Rotation Reflection

Isometry in Multi-Layer Neural Networks



- Consider a network with interleaved linear $\mathcal{A}$ and nonlinear $\sigma()$ layers
  - Forward propagation:

$$\boldsymbol{x}^L = \sigma(\mathcal{A}^L \cdots \sigma(\mathcal{A}^1 \boldsymbol{x}^0))$$

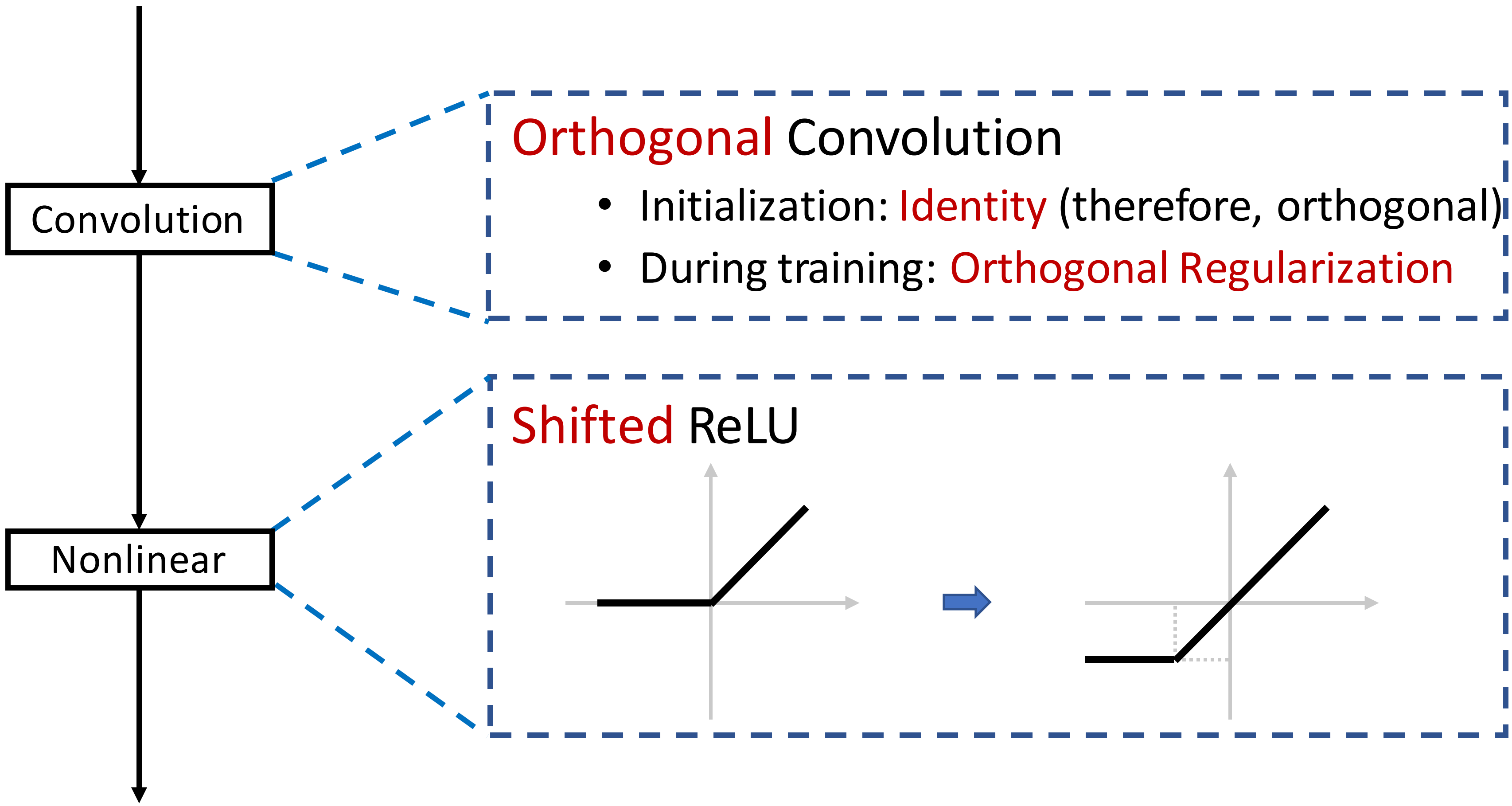
  - Backward propagation:

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{x}^0} = (\mathcal{A}^1)^* \mathcal{D}^1 \cdots (\mathcal{A}^L)^* \mathcal{D}^L (\boldsymbol{y} - \boldsymbol{x}^L),$$

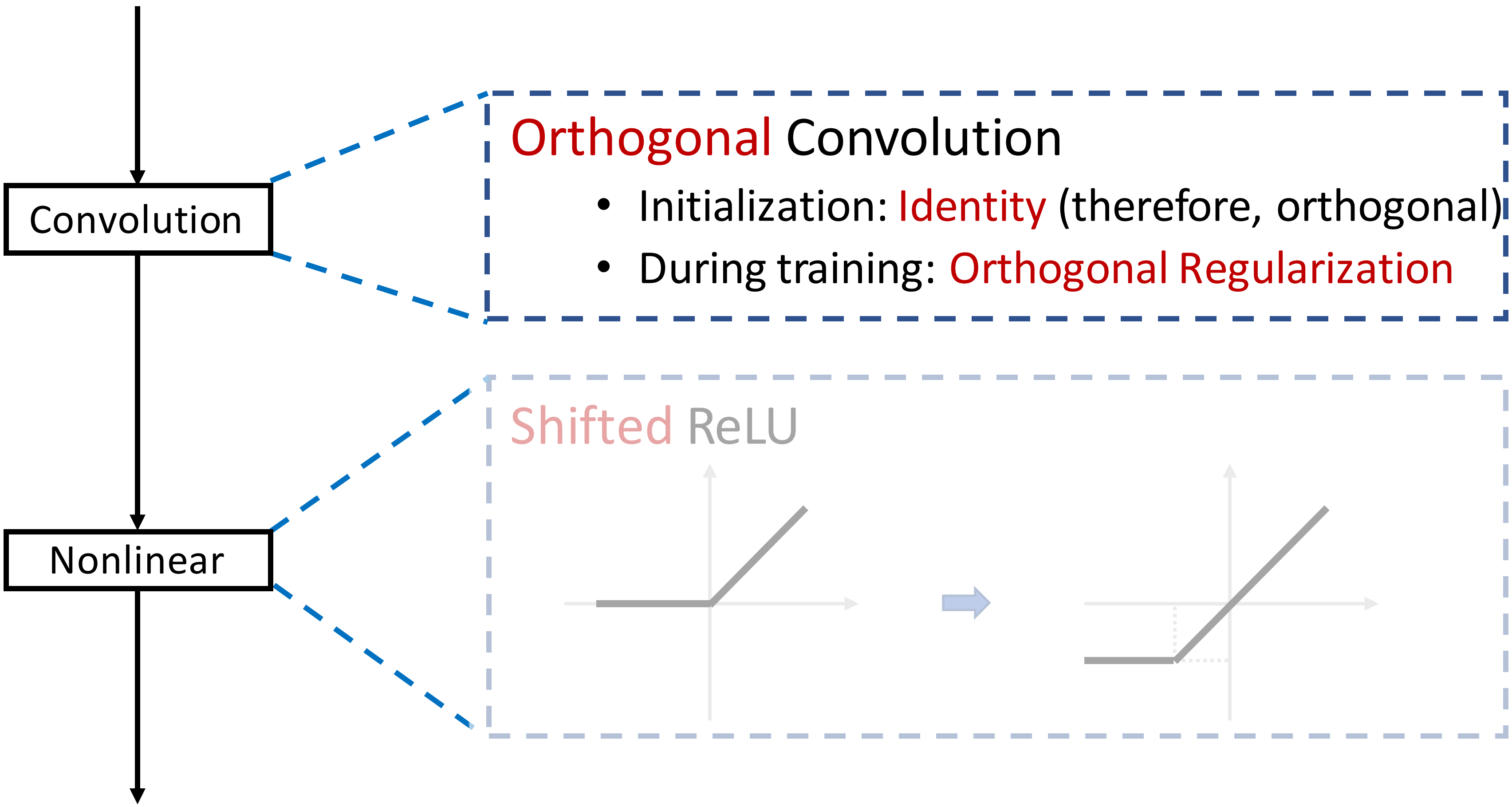
where $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$, and $\mathcal{D}$ is derivative of $\sigma()$

- Isometric Learning: Enforce isometry in forward/backward propagation
  - For linear layer: both $\mathcal{A}$ and $\mathcal{A}^*$ are (close to) an isometry
  - For nonlinear layer: both $\sigma()$ and $\mathcal{D}$ are (close to) an isometry

Isometric Network (ISONet)



Isometric Network (ISONet)



Isometry in Convolution

- Notations: Convolution for multi-channel images
  - Let $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_C) \in \mathbb{R}^{C\times H\times W}$ be the input signal
  - Let $\mathbf{A} = \begin{pmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} & \boldsymbol{a}_{13} & \cdots & \boldsymbol{a}_{1C} \\ \boldsymbol{a}_{21} & \boldsymbol{a}_{22} & \boldsymbol{a}_{23} & \cdots & \boldsymbol{a}_{2C} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{M1} & \boldsymbol{a}_{M2} & \boldsymbol{a}_{M3} & \cdots & \boldsymbol{a}_{MC} \end{pmatrix} \in \mathbb{R}^{M\times C\times k\times k}$ be the kernel
  - Define $\mathcal{A}\boldsymbol{x} \doteq \sum_{c=1}^{C} (\boldsymbol{a}_{1c} \star \boldsymbol{x}_c, \dots, \boldsymbol{a}_{Mc} \star \boldsymbol{x}_c) \in \mathbb{R}^{M\times H\times W}$

2D correlation

A Common Misnomer

- A plethora of work on "orthogonal" neural network
  - For CNNs: [Harandi & Fernando '16; Jia et al. '17; Cisse et al. '17; Bansal et al. '18; Zhang et al. '19a; Li et al. '19a; Huang et al. '20]
  - For RNNs: [Arjovsky et al. '16; Lezcano-Casado & Martinez-Rubio, '19]
  - For GANs: [Brock et al. '19; Liu et al. '20]

4D conv kernel

$$\begin{pmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} & \cdots & \boldsymbol{a}_{1C} \\ \boldsymbol{a}_{21} & \boldsymbol{a}_{22} & \cdots & \boldsymbol{a}_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{M1} & \boldsymbol{a}_{M2} & \cdots & \boldsymbol{a}_{MC} \end{pmatrix} \in \mathbb{R}^{M \times C \times k \times k}$$

$\Rightarrow$

2D matrix

$$\begin{bmatrix} \mathrm{vec}(\boldsymbol{a}_{11})^\top, \mathrm{vec}(\boldsymbol{a}_{12})^\top, \dots, \mathrm{vec}(\boldsymbol{a}_{1C})^\top \\ \mathrm{vec}(\boldsymbol{a}_{21})^\top, \mathrm{vec}(\boldsymbol{a}_{22})^\top, \dots, \mathrm{vec}(\boldsymbol{a}_{2C})^\top \\ \vdots \\ \mathrm{vec}(\boldsymbol{a}_{M1})^\top, \mathrm{vec}(\boldsymbol{a}_{M2})^\top, \dots, \mathrm{vec}(\boldsymbol{a}_{MC})^\top \end{bmatrix} \in \mathbb{R}^{M \times (C \times k \times k)}$$

Isometry for 4D conv kernel $\neq$ Isometry for 2D matrix!

Main Result

$$\mathbf{A} = \begin{pmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} & \boldsymbol{a}_{13} & \cdots & \boldsymbol{a}_{1C} \\ \boldsymbol{a}_{21} & \boldsymbol{a}_{22} & \boldsymbol{a}_{23} & \cdots & \boldsymbol{a}_{2C} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{M1} & \boldsymbol{a}_{M2} & \boldsymbol{a}_{M3} & \cdots & \boldsymbol{a}_{MC} \end{pmatrix}$$

Theorem: Orthogonal Convolution

Given a convolution kernel $\boldsymbol{A} \in \mathbb{R}^{M \times C \times k \times k}$, the operator $\mathcal{A}$ is an isometry if and only if

$$\sum_{m=1}^{M} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{mc'} = \begin{cases} \delta & \text{if } c = c', \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

and the operator $\mathcal{A}^*$ is an isometry if and only if

$$\sum_{c=1}^{C} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{m'c} = \begin{cases} \delta & \text{if } m = m', \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

In above, δ is the Kronecker delta function defined on $\mathbb{Z} \times \mathbb{Z}$ that takes value 1 at coordinate $(0, 0)$ and 0 otherwise.

In addition, Isometry for 4D conv kernel $\Longrightarrow$ Isometry for 2D matrix!

(Concurrent work: Wang et al., Orthogonal Convolutional Neural Networks)

Enforcing Orthogonality

$$\mathbf{A} = \begin{pmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} & \boldsymbol{a}_{13} & \cdots & \boldsymbol{a}_{1C} \\ \boldsymbol{a}_{21} & \boldsymbol{a}_{22} & \boldsymbol{a}_{23} & \cdots & \boldsymbol{a}_{2C} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{M1} & \boldsymbol{a}_{M2} & \boldsymbol{a}_{M3} & \cdots & \boldsymbol{a}_{MC} \end{pmatrix}$$

- Orthogonality at Initialization: Delta Initialization

$$\boldsymbol{a}_{ii} = \delta,\ \forall i = 1, \cdots, \min(C, M)$$
$$\boldsymbol{a}_{ij} = 0,\ \forall i \neq j$$

  - With Delta init, $\mathcal{A}$ and $\mathcal{A}^*$ are identity maps (when $C = M$)
  - Commonly adopted in RNNs

- Orthogonality during Training: Orthogonal Regularization

$$R(A) \doteq \sum_{c=c'} \| \sum_{m=1}^{M} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{mc'} - \delta \|_F^2 + \sum_{c \neq c'} \| \sum_{m=1}^{M} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{mc'} \|_F^2$$

$$R^*(A) \doteq \sum_{m=m'} \| \sum_{c=1}^{C} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{m'c} - \delta \|_F^2 + \sum_{m \neq m'} \| \sum_{c=1}^{C} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{m'c} \|_F^2$$

Fast Implementation

- Consider the computation of

$$R^*(A) \doteq \sum_{m=m'} \| \sum_{c=1}^{C} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{m'c} - \delta \|_F^2 + \sum_{m \neq m'} \| \sum_{c=1}^{C} \boldsymbol{a}_{mc} \star \boldsymbol{a}_{m'c} \|_F^2$$

  - Naive implementation

```
for m in range(M):
    for m_p in range(M):
        for c in range(C):
            # Compute correlation ...
```

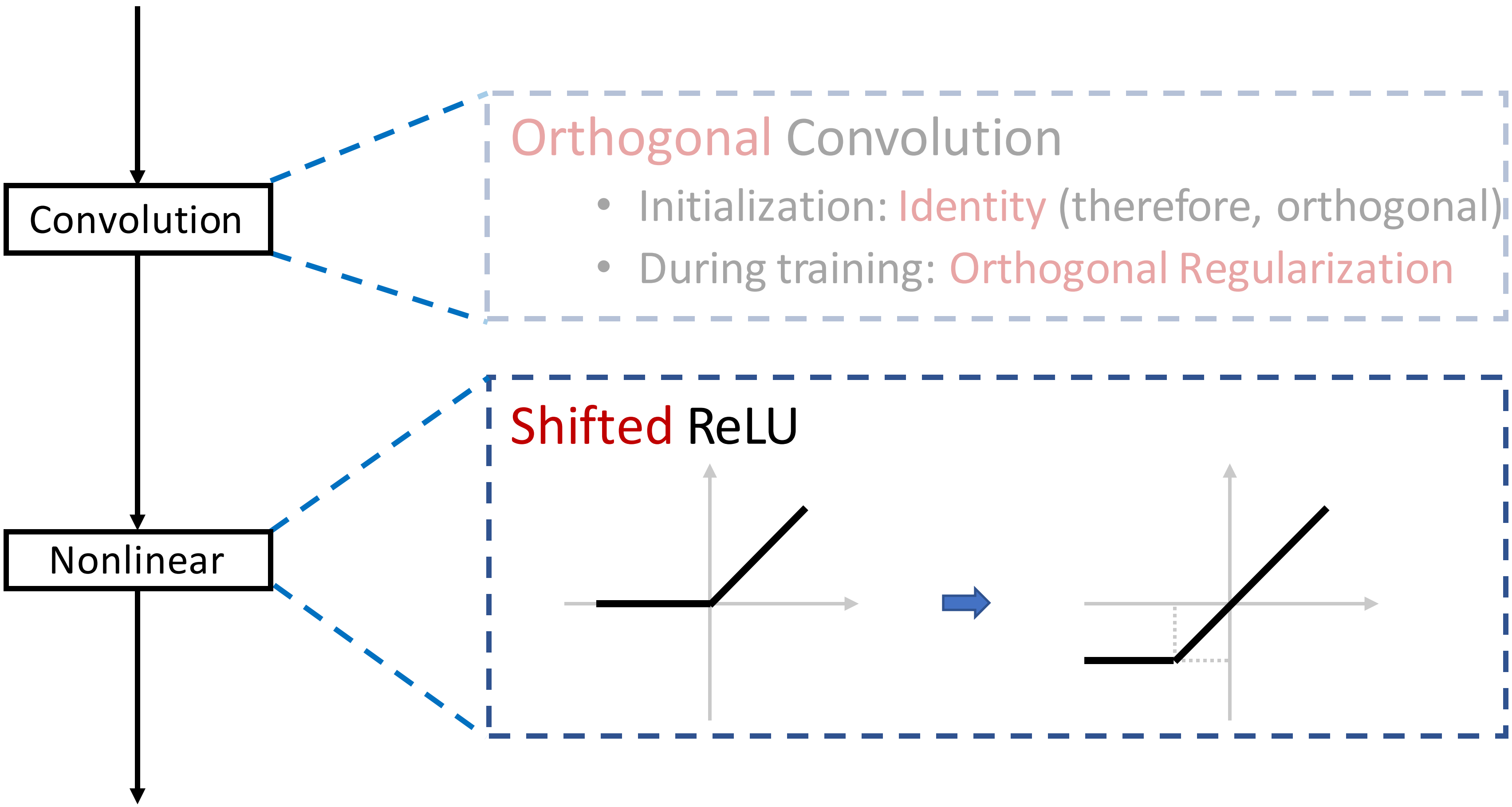
  - Implementation by deep learning packages

```
sum((conv2d(A, A) - identity)**2.0) / 2.0
```

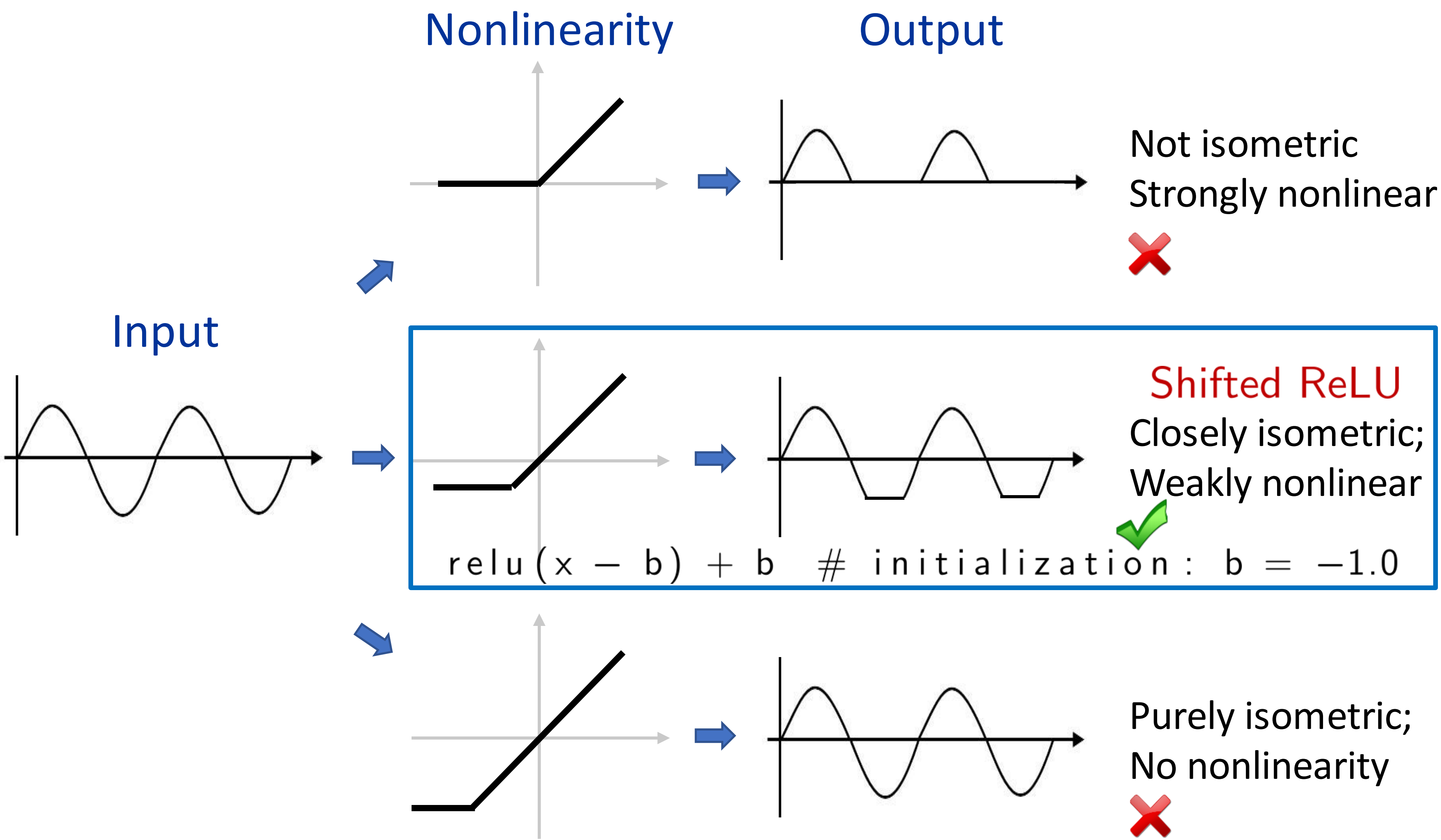
TensorFlow tf.nn.conv2d

PyTorch torch.nn.Conv2d

Isometric Network (ISONet)

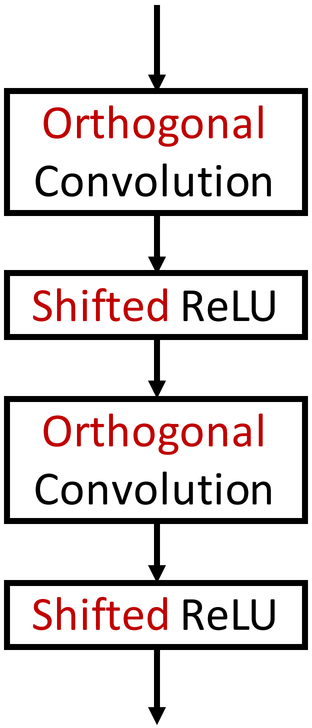


Enforcing Orthogonality in Nonlinear Layers



Unfortunately, isometry is at odds with nonlinearity (by Mazur-Ulam theorem)

ISONet: Training Deep *Vanilla* Net on ImageNet?

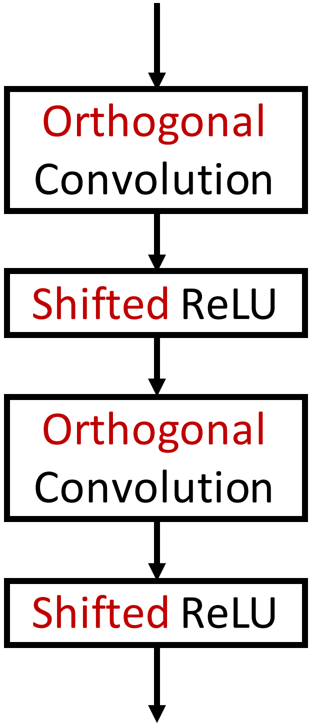


ImageNet Top-1 Accuracy

	Depth 18	Depth 34	Depth 50	Depth 101
ResNet	69.67	73.29	76.60	77.37
Vanilla	65.67	63.09	N/A	N/A
Vanilla+Shortcut	65.66	N/A	N/A	N/A
Vanilla+BN	68.98	69.43	70.00	N/A
ISONet	67.94	70.45	70.73	70.38

Training deep vanilla network on ImageNet, for the first time!

ISONet: Necessity of Isometric Components



	Method	SReLU	Delta Init.	Ortho. Reg.	Top-1 Acc. (%)
(a)	ResNet				73.29
(b)	Vanilla				63.09
(c)			✓	✓	46.83
(d)		✓			67.35
(e)		✓		✓	68.50
(f)		✓	✓		68.55
(g)	ISONet	✓	✓	✓	**70.45**

All three isometric components are needed!

Outline

- **(Conceptually)** What enables training very deep neural networks?

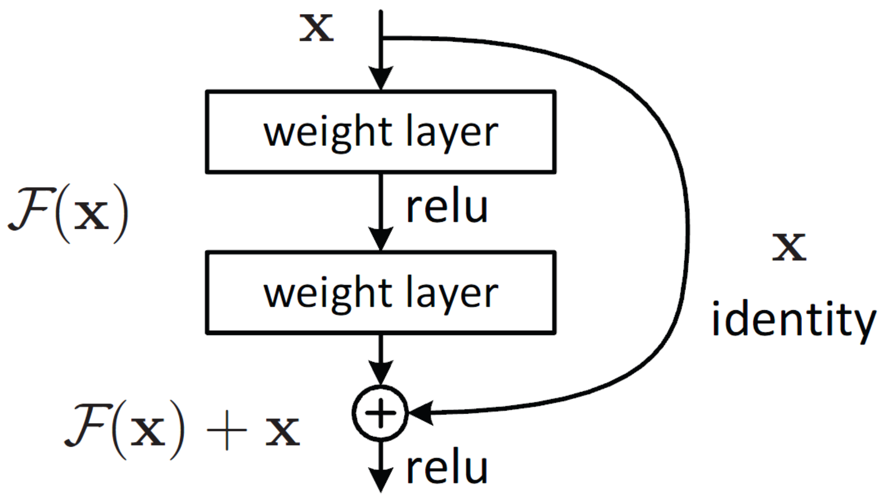
  Isometric Network (ISONet): Training 101-layer *vanilla* ConvNets (i.e., conv & nonlinear layers only) with $> 70\%$ accuracy on ImageNet

- **(Practically)** How to design better neural network architectures?

  Residual ISONet (R-ISONet): 1) SOTA performance on ImageNet without BatchNorm, 2) Better transfer ability for object detection

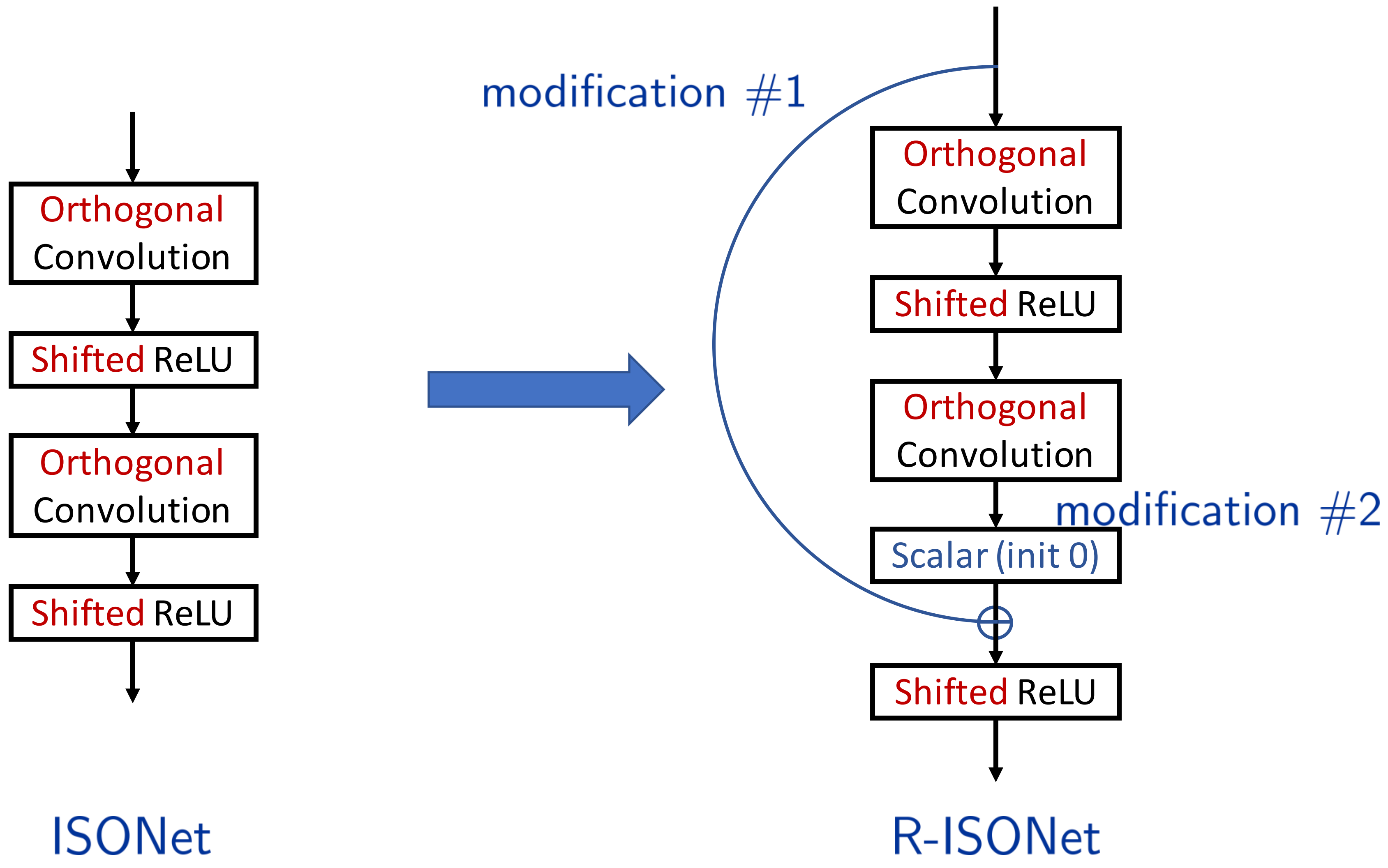
ResNet as Isometric Learning?

- ResNet is (almost) an isometry if residual is small



- Towards a more powerful isometric network: Combine residual learning with Orthogonal Conv. & Shifted ReLU?

Residual ISONet (R-ISONet)



A Remark about BatchNorm

- We do not use BatchNorm since:

introduces problems — BN's error increases rapidly when the batch size becomes smaller, caused by inaccurate batch statistics estimation. This limits BN's usage for training larger models and transferring features to computer vision tasks including detection, segmentation, and video, which require small batches constrained by memory consumption.

Detection / Segmentation
[Wu-He. '18]

experiments, we found that using BN prevents the model from learning good representations, as similarly reported in [35] (which avoids using BN). The model appears to "cheat" the pretext task and easily finds a low-loss solution. This is possibly because the intra-batch communica-

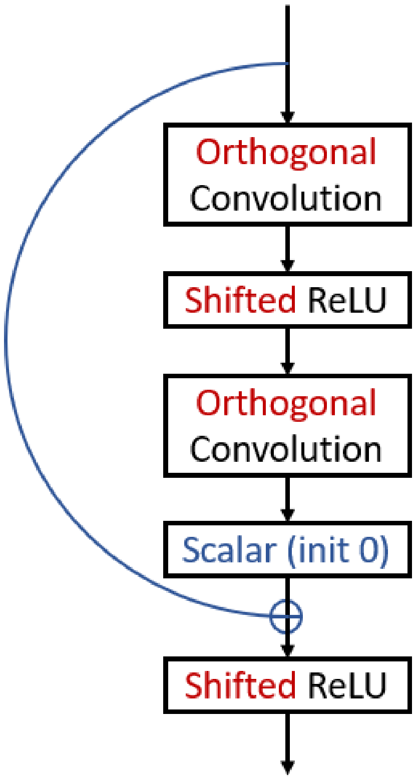
Constrastive Learning
[He et al. '19]

As discussed in the results section, Batch Normalization (BN) is ineffective for small batches, which are the inputs for Test-Time Training (both standard and online version)

Test-time Training [Sun et al. '20]

- However, using BatchNorm further improves the performance

R-ISONet



ImageNet Top-1 Accuracy

	depth 18	depth 34	depth 50	depth 101
ResNet	69.67	73.29	76.60	77.37
ResNet+Dropout	68.91	73.35	76.40	77.99
Vanilla+Shortcut	65.66	N/A	N/A	N/A
Fixup*	68.63	71.28	72.40	73.18
Fixup+Mixup*	67.37	72.56	76.00	76.17
R-ISONet	69.06	72.17	74.20	75.44
R-ISONet+Dropout	69.17	73.43	76.18	77.08

Best performing method without BatchNorm, on par with ResNet

* re-train from the released code for 100 epochs

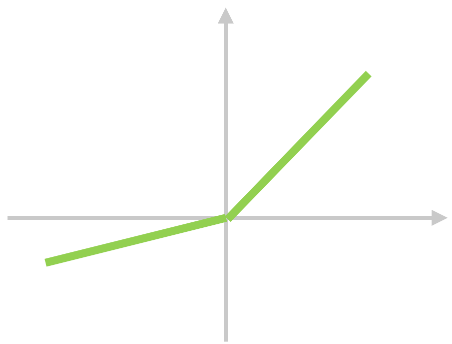
R-ISONet: Better Transfer Ability

- Tranfer learning on COCO for object detection & instance segmentation

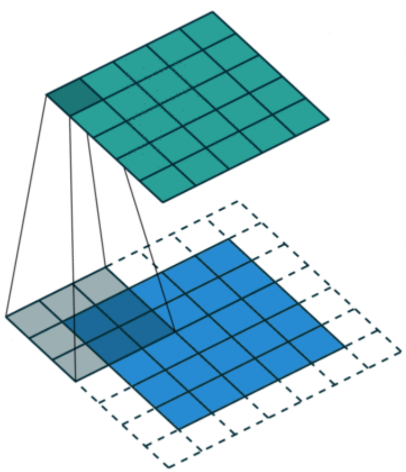
	Methods	mAP^{bbox}	mAP^{mask}
34 layer	ResNet	35.0	32.2
	R-ISONet	**36.2**	**33.0**
50 layer	ResNet	37.0	33.9
	R-ISONet	**37.3**	**34.4**

Finally, Evolution of Network Architectures

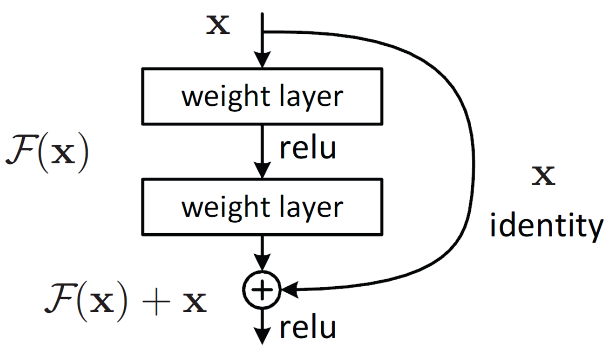
Nonlinear



Convolution

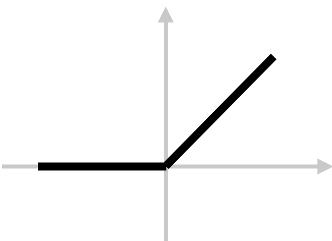


Residual

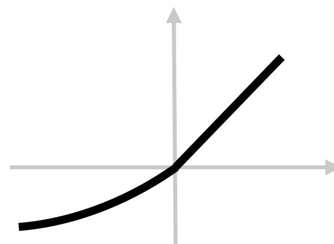


Nonlinear Activation

ReLU
early work

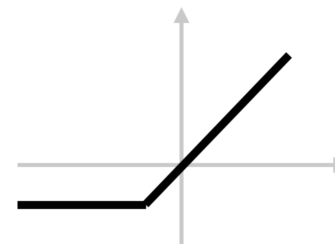


ELU / SELU
[Clevert'15, Klambauer'17]



"Input signal is shifted towards positivity"

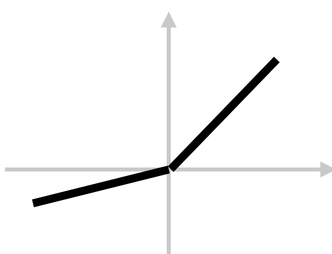
Shifted ReLU
[Xiang'17, Singh'19]



"Alleviates bias shifting"

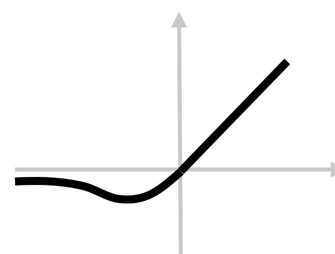
Isometry?

Leaky ReLU / PReLU
[Mass'13, He'15]



"Dead neurons prevent effective training"

Swish
[Ramachandran'17]



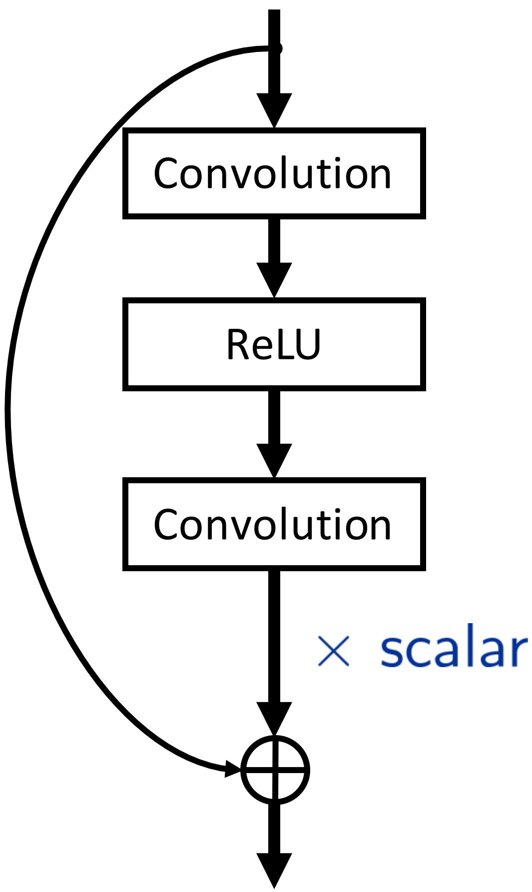
"Automatically search the best activation"

Weight Initialization / Regularization

- **Gaussian initialization:** early work
  "Variance of the signal maintains constant through multiple layers"
  - For tanh activation: Xavier initialization [Glorot'10]
  - For ReLU activation: Kaiming initialization [He'15]
  - For general activation (mean-field theory): [Poole'16, Schoenholz'16]
- **Orthogonal initialization**
  distance preserving (i.e., orthogonality) $\Longrightarrow$ variance preserving
  - For linear network: Provable benefits of ortho. init. [Saxe'13, Hu'20]
  - For nonlinear network: Dynamic isometry [Pennington'18, Xiao'18]
- **Orthogonal regularization**
  - For ConvNets: [Jia'17, Cisse'17, Bansal'17, Zhang'19, Huang'20]
  - For RNNs: [Arjovsky'16, Lezcano-Casado'19]
  - For GANs: [Brock'18, Liu'19]

Isometry?

Residual Learning

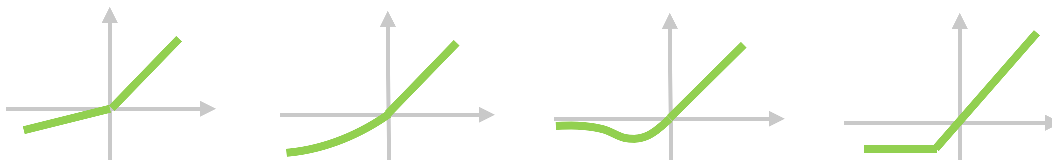


- **scalar** $= 1$ [He' 16]
- **scalar** $= 0$ [Goyal'17, Bachlechner'20]
- **scalar** $= 0.1$ [Zagoruyko'17]
- **scalar** $\sim \frac{1}{\sqrt{L}}$, where L=#layers

[Taki'17, Balduzzi'17, Qiu'18, Tarnowski'19, Zhang'19]

Isometry?

Finally, Evolution of Network Architectures



[Mass'13, He'15, Clevert'15, Klambauer'17, Ramachandran'17, Singh'19]

Nonlinear

Isometry

(energy preserving)

Convolution

- **Initialization** [Glorot'10, He'15, Poole'16, Schoenholz'16, Pennington'18, Xiao'18]
- **Regularization** [Arjovsky'16, Jia'17, Cisse'17, Bansal'17, Brock'18, Zhang'19, Huang'20]

Residual

Many variances [He'16, Szegedy'16, Goyal'17, Zagoruyko'17, Taki'17, Zhang'19, Bachlechner'20]

Open Problems: Theory

- Optimization: Improving optimization landscape and alleviating vanishing/exploding gradient?
  (Work of [Hu et al. '20] for linear networks. How about nonlinear networks?)

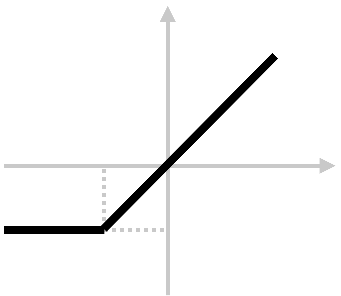
- Generalization: Better generalization bounds?
  (Existing work: [Jia et al. '19])

- Robustness: Improved adversarial robustness?
  (Empirical work: [Cisse et al. '17])

- GANs: Performance predicted by generator conditioning?
  (Empirical work: [Odena et al. '18])

Open Problems: Method

– Isometry vs. Nonlinearity: What is the best trade-off?



– Isometry vs. Restricted Isometry: Low-dim. structure of data?