Deep Isometric Learning for Visual Recognition

Haozhi Qi [1] **Chong You** [1] **Xiaolong Wang** [1 2] **Yi Ma** [1] **Jitendra Malik** [1]
https://haozhiqi.github.io/ISONet

Abstract

Initialization, normalization, and skip connections are believed to be three indispensable techniques for training very deep convolutional neural networks and obtaining state-of-the-art performance. This paper shows that deep vanilla ConvNets without normalization nor skip connections can also be trained to achieve surprisingly good performance on standard image recognition benchmarks. This is achieved by enforcing the convolution kernels to be near *isometric* during initialization and training, as well as by using a variant of ReLU that is shifted towards being *isometric*. Further experiments show that if combined with skip connections, such near isometric networks can achieve performances on par with (for ImageNet) and better than (for COCO) the standard ResNet, even without normalization at all. Our code is available at https://github.com/HaozhiQi/ISONet.

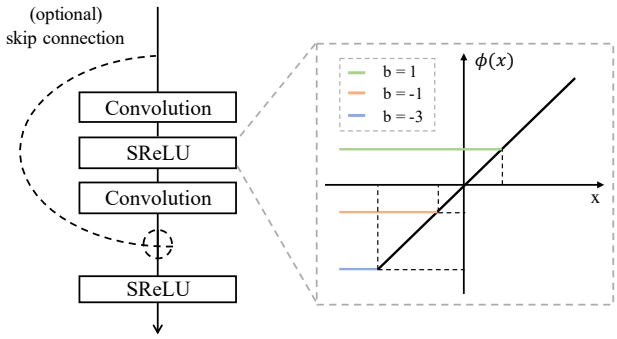

Figure 1. A basic block of an Isometric Network (ISONet). Each block contains only convolution and nonlinear activation layers, with an identity skip connections (for R-ISONet only). The convolution is initialized as the (Kronecker) delta kernel and regulated to be (near) orthogonal during training. The activation is Shifted ReLU (SReLU) $\phi(\cdot)$ with a learnable parameter b for obtaining a balance between nonlinearity and isometry. The figure shows three examples of the SReLU with $b = 1, -1$ and -3.

1. Introduction

Convolutional Neural Networks (ConvNets) have achieved phenomenal success in computer vision (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Szegedy et al., 2015; He et al., 2015; Ioffe & Szegedy, 2015; He et al., 2016; Xie et al., 2017b; Huang et al., 2017). While shallow ConvNets with a few layers have existed for decades (LeCun et al., 1989; Denker et al., 1989; LeCun et al., 1990; 1998), it is only until recently that networks with hundreds or even thousands of layers can be effectively trained. Such *deep* ConvNets are able to learn sophisticated decision rules for complex practical data, therefore are usually indispensable for obtaining state-of-the-art performance.

Training *deep* ConvNets is inherently difficult (Pascanu et al., 2013; Glorot & Bengio, 2010). Despite widespread interests in this problem, it has not been made possible until the development of proper weight initialization (Glorot & Bengio, 2010; He et al., 2015), feature map normalization (Ioffe & Szegedy, 2015), and residual learning techniques (He et al., 2016; Srivastava et al., 2015). Following these initial works, a wide variety of network architectural components including novel nonlinear activation, weight initialization, weight regularization, and residual learning techniques have been purposed in the literature (see Section 4 for an overview). Such techniques are motivated and justified from diverse perspectives, such as prevention of dead neurons (Maas et al., 2013), promotion of self-normalization (Klambauer et al., 2017), reduction of filter redundancy (Wang et al., 2020), improvement of generalization (Jia et al., 2019), to name a few.

The abundance of existing architectural components and the diversity of their design principles posit ConvNets architectural design a difficult task. After all, which combination of components should one use for their networks? Such a challenge motivates us to pose the following question: *Is there a central guiding principle for training very deep ConvNets?*

Isometric learning. We show that the *isometric* property, where each layer of the network preserves the inner product

[1]UC Berkeley [2]UC San Diego. Correspondence to: Haozhi Qi <hqi@berkeley.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

for both forward and backward propagation, plays a central role in training deep ConvNets. To illustrate this, we take a minimalist approach and show that a vanilla deep network (i.e., composed of interleaved convolution and nonlinear layers only) can be trained as long as both the convolution and nonlinear activation are close to an isometry.

Specifically, we design the Isometric Networks (ISONets) where the convolution layer is simply initialized as *the identity* and is regularized to be *near orthogonal* during training, and the nonlinear layer is the *Shifted ReLU* (SReLU) obtained by shifting regular ReLU towards being an identity around the origin with a learnable parameter (see Figure 1). We perform extensive experiments with so designed simple networks on image classification using the ImageNet dataset (Deng et al., 2009; Russakovsky et al., 2015). Our results show that an ISONet with more than 100 layers is trainable and can achieve surprisingly competitive performance. To the best of our knowledge, this is the best performing *vanilla* ConvNet for such tasks.

From a practical perspective, isometric learning provides an important guiding principle for addressing challenges in network architectural design. One such challenge is that the commonly used normalization layers in modern networks (Ioffe & Szegedy, 2015; Ba et al., 2016; Wu & He, 2018) require certain statistical independence assumptions to hold and large enough batch size or channel number for precise estimation of such statistics. This drawback significantly limits their applications to robust learning (Sun et al., 2019), contrastive learning (Chen et al., 2020), implicit models (Bai et al., 2020), object detection and so on.

To address such a challenge, we develop a Residual ISONet (R-ISONet) that does not contain any normalization layer. In particular, we introduce skip connection to ISONet as it helps to better promote isometry of the network (Tarnowski et al., 2019). We evaluate the performance of R-ISONet for object detection and instance segmentation on the COCO dataset (Lin et al., 2014). In these applications, the batch size is very small due to the high resolution of the images, so that batch normalization becomes ineffective. Our experiment shows that R-ISONet obtains better performance than standard ResNet. Comparing with existing techniques for addressing the small-batch challenge (Luo et al., 2018; Wu & He, 2018; Singh & Krishnan, 2020), R-ISONet has the benefit that it does not suffer from a slowdown during inference.

Isometry and many closely related notions have been implicitly and explicitly studied and utilized in many previous works for improving deep networks. In particular, techniques to promote isometric property have been widely practiced in the literature. However, they are often used together with many other design techniques and none of the previous work has clearly demonstrated that the isometry property *alone* ensures surprisingly strong performance for deep networks. At the end (Section 4), we discuss how isometric learning offers a simple but unified framework for understanding numerous seemingly different and diverse ideas in the literature, supporting the hypothesis that isometric learning offers a central guiding principle for designing and learning deep networks.

2. Isometric Networks

In this section, we introduce the isometric principle and present the isometric network (ISONet). In Section 2.1, we formally introduce the isometric principle in a vanilla net as enforcing both the convolution and nonlinear activation layers to be close to an *isometry*. Subsequently, Section 2.2 derives the notion of isometry for convolution, and explains how it can be enforced at initialization and in the training process. In Section 2.3, we discuss principles for designing nonlinear activation functions in isometric learning and argue that one needs to strike a balance between isometry and nonlinearity, which cannot be achieved at the same time. Finally, in Section 2.4, skip connections (or residual structure) can be naturally introduced to ISONet to further improve isometry, which leads to the R-ISONet.

2.1. Isometric Learning

We first develop a vanilla network that is composed of interleaved convolution[1] and nonlinear activation layers, i.e.,

$$\boldsymbol{x}^\ell = \phi(\boldsymbol{y}^\ell), \;\; \boldsymbol{y}^\ell = \mathcal{A}^\ell \boldsymbol{x}^{\ell-1}, \quad \ell = 1, \ldots, L, \tag{1}$$

where $\mathcal{A}^\ell : \mathbb{R}^{C^{\ell-1}\times H\times W} \to \mathbb{R}^{C^\ell \times H\times W}$ denotes a convolution operator, $\phi(\cdot)$ denotes a point-wise nonlinear activation function, and $\boldsymbol{x}^0 \in \mathbb{R}^{C^0\times H\times W}$ is the input signal. Assuming a squared loss given by Loss$= \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}^L\|_2^2$, the backward propagation dynamic at $\boldsymbol{x}^0$ is given by

$$\frac{\partial \text{Loss}}{\partial \boldsymbol{x}^0} = (\mathcal{A}^1)^* \mathcal{D}^1 \cdots (\mathcal{A}^L)^* \mathcal{D}^L (\boldsymbol{z} - \boldsymbol{x}^L), \tag{2}$$

where $(\mathcal{A}^\ell)^* : \mathbb{R}^{C^\ell\times H\times W} \to \mathbb{R}^{C^{\ell-1}\times H\times W}$ is the adjoint operator of $\mathcal{A}^\ell$, and $\mathcal{D}^\ell : \mathbb{R}^{C^\ell\times H\times W} \to \mathbb{R}^{C^\ell\times H\times W}$ operates point-wise by multiplying the (c,h,w)-th entry of the input by $\phi'(\boldsymbol{y}^\ell_{c,h,w})$. For the operator $\mathcal{A}^\ell$ and $\phi(\cdot)$ in the forward dynamic and $(\mathcal{A}^\ell)^*$ and $\mathcal{D}^\ell$ in the backward dynamic, we may define the notion of *isometry* as follows:

Definition 1 (Isometry). *A map $\mathcal{A} : \mathbb{R}^C \to \mathbb{R}^M$ is called an isometry if*

$$\langle \mathcal{A}\boldsymbol{x}, \mathcal{A}\boldsymbol{x}' \rangle = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle, \;\; \forall \{\boldsymbol{x}, \boldsymbol{x}'\} \subseteq \mathbb{R}^C. \tag{3}$$

Our ISONet is designed to maintain that all transformations

[1] We consider convolution for 2D signals (e.g., images), though our analysis trivially generalizes to arbitrary dimensional signals.

($\mathcal{A}$, $\mathcal{A}^*$, ϕ, and $\mathcal{D}$) in the forward and the backward dynamics are close to an *isometry*. In the following, we develop techniques for enforcing isometry in convolution and nonlinear activation layers. As we show in Section 3, combining these components enables effective training of ISONet with more than 100 layers on image classification tasks.

2.2. Isometry in Convolution

In this section, we show how to impose the property of isometry for convolutional layers in ConvNets, and explain how it can be achieved at initialization and through network training.

Notation. For computation concerning convolution, we treat a 2D signal $\boldsymbol{\xi} \in \mathbb{R}^{H\times W}$ as a function defined on the discrete domain $[1,\dots,H]\times[1,\dots,W]$ and further extended to the domain $\mathbb{Z}\times\mathbb{Z}$ by padding zeros. Similarly, we regard any 2D kernel $\boldsymbol{\alpha}\in\mathbb{R}^{k\times k}$ with $k=2k_0+1$ for some integer k_0 as a function defined on the discrete domain $[k_0-1,\dots,0,\dots,k_0+1]\times[k_0-1,\dots,0,\dots,k_0+1]$ and extended to $\mathbb{Z}\times\mathbb{Z}$ by padding zeros. $\boldsymbol{\xi}[i,j]$ represents the value of the function $\boldsymbol{\xi}$ at the coordinate (i,j).

Given any $\boldsymbol{\xi}\in\mathbb{R}^{H\times W}$ and $\boldsymbol{\alpha}\in\mathbb{R}^{k\times k}$, the convolution of $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}$ is defined as

$$(\boldsymbol{\alpha}*\boldsymbol{\xi})[i,j]=\sum_{p=-k_0}^{k_0}\sum_{q=-k_0}^{k_0}\boldsymbol{\xi}[i-p,j-q]\cdot\boldsymbol{\alpha}[p,q], \quad (4)$$

and the correlation of $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}$ is defined as

$$(\boldsymbol{\alpha}\star\boldsymbol{\xi})[i,j]=\sum_{p=-k_0}^{k_0}\sum_{q=-k_0}^{k_0}\boldsymbol{\xi}[i+p,j+q]\cdot\boldsymbol{\alpha}[p,q]. \quad (5)$$

In contrast to the conventions in signal processing, the convolution layers (i.e. $\{\mathcal{A}^\ell\}_{\ell=1}^L$ in (1)) in modern deep learning frameworks actually perform multi-channel *correlation* operations that map a C-channel 2D signal to an M-channel 2D signal. Let $\boldsymbol{x}=(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_C)\in\mathbb{R}^{C\times H\times W}$ be the input signal where $\boldsymbol{\xi}_c\in\mathbb{R}^{H\times W}$ for each $c\in\{1,\dots,C\}$, and let

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{\alpha}_{11}&\boldsymbol{\alpha}_{12}&\boldsymbol{\alpha}_{13}&\dots&\boldsymbol{\alpha}_{1C}\\ \boldsymbol{\alpha}_{21}&\boldsymbol{\alpha}_{22}&\boldsymbol{\alpha}_{23}&\dots&\boldsymbol{\alpha}_{2C}\\ \vdots&\vdots&\vdots&\ddots&\vdots\\ \boldsymbol{\alpha}_{M1}&\boldsymbol{\alpha}_{M2}&\boldsymbol{\alpha}_{M3}&\dots&\boldsymbol{\alpha}_{MC}\end{pmatrix}\in\mathbb{R}^{M\times C\times k\times k}, \quad (6)$$

be the convolution kernel where each $\boldsymbol{\alpha}_{m,c}$ is a kernel of size $k\times k$ for $c\in\{1,\dots,C\}$ and $m\in\{1,\dots,M\}$. The convolution operator $\mathcal{A}$ associated with $\boldsymbol{A}$ is given by

$$\mathcal{A}\boldsymbol{x}:=\sum_{c=1}^{C}\big(\boldsymbol{\alpha}_{1c}\star\boldsymbol{\xi}_c,\dots,\boldsymbol{\alpha}_{Mc}\star\boldsymbol{\xi}_c\big)\in\mathbb{R}^{M\times H\times W}. \quad (7)$$

The adjoint of $\mathcal{A}$ that appears in the backward dynamics (2), denoted as $\mathcal{A}^*$, is a mapping from $\mathbb{R}^{M\times H\times W}$ to $\mathbb{R}^{C\times H\times W}$:

$$\mathcal{A}^*\boldsymbol{y}:=\sum_{m=1}^{M}\big(\boldsymbol{\alpha}_{m1}*\boldsymbol{\eta}_m,\dots,\boldsymbol{\alpha}_{mC}*\boldsymbol{\eta}_m\big)\in\mathbb{R}^{C\times H\times W}, \quad (8)$$

where $\boldsymbol{y}=(\boldsymbol{\eta}_1,\dots,\boldsymbol{\eta}_M)\in\mathbb{R}^{M\times H\times W}$ and $\boldsymbol{\eta}_m\in\mathbb{R}^{H\times W}$.

We first study under what conditions of kernel $\boldsymbol{A}$ such that $\mathcal{A}$ and $\mathcal{A}^*$ are isometric. Starting from Definition 1 we arrive at the following theorem (see appendix for the proof).

Theorem 1. *Given a convolution kernel $\boldsymbol{A}\in\mathbb{R}^{M\times C\times k\times k}$ in (6), the operator $\mathcal{A}$ is an isometry if and only if*

$$\sum_{m=1}^{M}\boldsymbol{\alpha}_{mc}\star\boldsymbol{\alpha}_{mc'}=\begin{cases}\boldsymbol{\delta} & \text{if } c=c',\\ \boldsymbol{0} & \text{otherwise},\end{cases} \quad (9)$$

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

$$\sum_{c=1}^{C}\boldsymbol{\alpha}_{mc}\star\boldsymbol{\alpha}_{m'c}=\begin{cases}\boldsymbol{\delta} & \text{if } m=m',\\ \boldsymbol{0} & \text{otherwise}.\end{cases} \quad (10)$$

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

For the case of $M=C$, the isometry of $\mathcal{A}$ is equivalent to the isometry of $\mathcal{A}^*$[2]. We refer to $\boldsymbol{A}$ as an *orthogonal convolution* kernel if $M=C$ and either $\mathcal{A}$ or $\mathcal{A}^*$ is isometric.

On the other hand, if $M\neq C$ then $\mathcal{A}$ and $\mathcal{A}^*$ cannot be both isometric. Nonetheless, we can still enforce isometry of $\mathcal{A}$ (in the case $M\geq C$) or the isometry of $\mathcal{A}^*$ (in the case $M\leq C$) by the conditions (9) and (10), respectively.

Example 1 (Delta kernel)**.** *We refer to a convolution kernel $\boldsymbol{A}\in\mathbb{R}^{C\times M\times k\times k}$ as the (Kronecker) delta kernel, denoted as $\boldsymbol{\delta}^{C\times M\times k\times k}$, if its entries indexed by (i,i,k_0,k_0) (assuming $k=2k_0+1$) for all $i\in\{1,\dots,\min(C,M)\}$ are set to 1 and all other entries are set to zero. If $M\geq C$ (resp., $M\leq C$), then the operator $\mathcal{A}$ (resp., $\mathcal{A}^*$) associated with a delta kernel is an isometry. Moreover, if $M=C$ then a delta kernel is orthogonal.*

Isometry at initialization. As shown in Example 1, we may initialize all convolution kernels to be isometric by setting them to be the delta kernel, which we refer to as the Delta initialization. Other orthogonal initialization, such as the Delta Orthogonal initialization (Xiao et al., 2018), are also plausible choices. We empirically find that Delta initialization works well and often outperforms Delta Orthogonal initialization. Such initialization is also commonly used in initializing Recurrent Neural Networks (Le et al., 2015).

Isometry during training. Isometry at initialization does not guarantee that isometry will be preserved through the

[2]When $M=C$, the isometry of $\mathcal{A}$ implies that it is surjective, therefore unitary, hence $\mathcal{A}^*$ is also an isometry.

training process. In addition to Delta initialization, we enforce isometry by penalizing the difference between the left and right hand sides of (9) (or (10)). As we shown below, this can be easily implemented via modern deep learning packages.

In particular, given an input that contains a batch of N multi-channel signals $\boldsymbol{X} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{N\times C\times H\times W}$ where each $\boldsymbol{x}_n \in \mathbb{R}^{C\times H\times W}$ for each $n \in \{1, \ldots, N\}$, and a convolution kernel $\boldsymbol{A} \in \mathbb{R}^{M\times C\times k\times k}$ defined as in (6), a convolution function in a typical deep learning framework, denoted as $\text{Conv}(\boldsymbol{A}, \boldsymbol{X})$, performs correlation of $\boldsymbol{A}$ on each of the N signals and stack the resulting N multi-channel signals to a tensor:

$$\text{Conv}(\boldsymbol{A}, \boldsymbol{X}) = (\mathcal{A}\boldsymbol{x}_1, \ldots, \mathcal{A}\boldsymbol{x}_N). \tag{11}$$

By replacing $\boldsymbol{X}$ in (11) with the kernel $\boldsymbol{A}$ itself and using (7), we obtain

$$\begin{aligned} &\text{Conv}(\boldsymbol{A}, \boldsymbol{A}) \\ &= \big(\mathcal{A}(\boldsymbol{\alpha}_{11}, \ldots, \boldsymbol{\alpha}_{1C}), \ldots, \mathcal{A}(\boldsymbol{\alpha}_{M1}, \ldots, \boldsymbol{\alpha}_{MC})\big) \\ &= \begin{pmatrix} \sum_{c=1}^{C} \boldsymbol{\alpha}_{1c} \star \boldsymbol{\alpha}_{1c} & \cdots & \sum_{c=1}^{C} \boldsymbol{\alpha}_{Mc} \star \boldsymbol{\alpha}_{1c} \\ \vdots & \ddots & \vdots \\ \sum_{c=1}^{C} \boldsymbol{\alpha}_{1c} \star \boldsymbol{\alpha}_{Mc} & \cdots & \sum_{c=1}^{C} \boldsymbol{\alpha}_{Mc} \star \boldsymbol{\alpha}_{Mc} \end{pmatrix}. \end{aligned} \tag{12}$$

Comparing (12) with the condition in (10), we see that the operator $\mathcal{A}^*$ is an isometry if and only if $\text{Conv}(\boldsymbol{A}, \boldsymbol{A})$ is a delta kernel $\boldsymbol{\delta}^{M\times M\times k\times k}$. Similarly, $\mathcal{A}$ is an isometry if and only if $\text{Conv}(\boldsymbol{A}^\top, \boldsymbol{A}^\top) = \boldsymbol{\delta}^{C\times C\times k\times k}$, where $\boldsymbol{A}^\top$ denotes a kernel with the first two dimension of $\boldsymbol{A}$ transposed. From these results, we can enforce isometry by adding one of the following regularization terms to the objective function:

$$L(\boldsymbol{A}) = \frac{\gamma}{2}\|\text{Conv}(\boldsymbol{A}, \boldsymbol{A}) - \boldsymbol{\delta}^{M\times M\times k\times k}\|_F^2, \quad \text{or} \tag{13}$$

$$L(\boldsymbol{A}^\top) = \frac{\gamma}{2}\|\text{Conv}(\boldsymbol{A}^\top, \boldsymbol{A}^\top) - \boldsymbol{\delta}^{C\times C\times k\times k}\|_F^2, \tag{14}$$

where γ is a regularization coefficient. We use $L(\boldsymbol{A})$ when $C > M$ and $L(\boldsymbol{A}^\top)$ otherwise.

2.3. Isometry in Nonlinear Activation

The rectified linear unit (ReLU) is one of the most popular activation functions for deep learning applications (Nair & Hinton, 2010). Defined entry-wise on the input as $\phi(x) = \max(0, x)$, ReLU is an identity map for nonnegative input, but completely removes all negative component. For an input signal that is normalized to zero mean and unit variance, ReLU is far from being an isometry. How can we develop an isometric nonlinear activation layer?

Unfortunately, isometry is innately at odds with nonlinearity. By Mazur-Ulam theorem, any surjective isometry between two normed spaces over $\mathbb{R}$ is linear up to a translation (Nica, 2013). Therefore, isometry can never be achieved if any nonlinearity is present. On the other hand, a purely isometric network without nonlinearity is not interesting either, since the entire network becomes a linear transformation which has very limited learning ability. The design of the nonlinear activation function requires striking a good *trade-off between isometry and nonlinearity*.

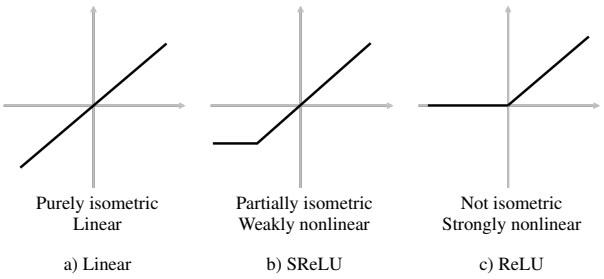

Figure 2. Illustration of isometry in nonlinear activation

We advocate using a Shifted ReLU (SReLU) for achieving such a trade-off. SReLU is obtained by interpolating regular ReLU (for obtaining nonlinearity) and the identity (for obtaining isometry). Given a signal $\boldsymbol{y} \in \mathbb{R}^{C\times H\times W}$ where C is the number of channels, SReLU is defined point-wise as

$$\phi_b(y) = \max(y, b), \tag{15}$$

where b is a parameter that is shared for each channel of $\boldsymbol{y}$. SReLU becomes the regular ReLU if $b = 0$ and the identity if $b \to -\infty$, therefore can be interpreted as a trade-off between nonlinearity and isometry (see Figure 2). In terms of backward dynamic (see (2)), the operator $\mathcal{D}$ associated with $\phi_b(\cdot)$ is given as $\mathcal{D}(x) = 1_{y\geq b} \cdot x$, which is an identity map if y is larger than b.

In our experiments, we find it beneficial to initialize b to be negative values, which indicates the importance of such a trade-off between ReLU and identity. For simplicity, in all our experiments we initialize b to be -1 and optimize it in the training process.

2.4. Isometry in Residual Structure

We may further improve the isometric property of each layer of ISONet with a residual component (He et al., 2016), which we call a residual ISONet (R-ISONet). The core idea is to introduce a skip connection to ISONet, so that the network learns a *residual* component that is added onto the path of signal propagation (see Figure 1). Such a network architecture is automatically near isometric if the residual component is small enough relative to the identity shortcut (Tarnowski et al., 2019). Motivated by such an observation, we add a scalar multiplier s at the end of each residual branch and initialize s to be zero (s is shared within each channel). Similar design ideas have previously been

explored to improve ResNet (Szegedy et al., 2017; Goyal et al., 2017; Zhang et al., 2019b).

Since skip connection helps to enforce isometry, R-ISONet exhibits better performance than ISONet. For image classification on ImageNet, R-ISONet obtains almost as good performance as regular ResNet, even though R-ISONet has no normalization layers. Since no normalization layer is required, R-ISONet is particularly suited for applications where normalization layers are ineffective. As we show in the next section, R-ISONet obtains better performance than regular ResNet for object detection tasks on COCO dataset.

3. Experiments

3.1. Experimental Setup

Setup. We test the performance of ISONets on the ILSVRC-2012 image classification dataset (Deng et al., 2009; Russakovsky et al., 2015). The training set contains $\sim$1.28 million images from 1,000 categories. Evaluation is performed on the validation set which contains 50,000 images. 1-Crop, Top-1 accuracy is reported.

Network architectures. The design of network architecture follows from the design of ResNet (He et al., 2016; Goyal et al., 2017), with the only difference being that we change the residual blocks to the block shown in Figure 1. In particular, the number of channels and layers in each stage remain the same as ResNet. We find that (R-)ISONet tends to overfit since BatchNorm is removed. Therefore we add one dropout layer right before the final classifier. The dropout probability is 0.4 for R-ISONet and 0.1 for ISONet.

Implementation details. The hyperparameter settings follow from prior works (He et al., 2016; Goyal et al., 2017). All models are trained using SGD with weight decay 0.0001, momentum 0.9 and mini-batch size 256. The initial learning rate is 0.1 for R-ISONet and 0.02 for ISONet. The models are trained for 100 epochs with the learning rate subsequently divided by a factor of 10 at the 30th, 60th and 90th epochs. To stabilize training in the early stage, we perform linear scheduled warmup (Goyal et al., 2017) for the first 5 epochs for both our methods and baseline methods. The orthogonal regularization coefficient γ in (13) (when used) is 0.0001 except for ISONet-101 where a stronger regularization (0.0003) is needed. For other baselines without normalization or skip connections, we choose the largest learning rate that makes them converge.

3.2. ISONet: Isometry Enables Training Vanilla Nets

In this section, we verify through extensive experiments that isometry is a central principle for training deep networks. In particular, we show that more than 100-layer ISONet can be effectively trained with neither normalization layers nor skip connections. In addition, we demonstrate through ablation study that all of the isometric components in ISONet are necessary for obtaining good performance.

	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**

Table 1. **Isometric learning (with SReLU, Delta initialization and orthogonal regularization) enables training ISONets on ImageNet without BatchNorm and skip connection.** (a) Regular 34-layer ResNet with ReLU activation, Kaiming initialization and no orthogonal regularization. (b) Same as ResNet but without BatchNorm and skip connection. (g) Our ISONet with the same backbone as Vanilla. (c-f) Ablation for (g) that shows that the combination of all three isometric learning components is necessary for effectively training ISONet.

Orthogonal coefficient γ	0	$1e^{-5}$	$3e^{-5}$	$1e^{-4}$	$3e^{-4}$
Top-1 Accuracy (%)	68.55	70.08	70.44	**70.45**	69.32

Table 2. **Effect of orthogonal coefficients for 34-layer ISONets**. The performance is not sensitive to the coefficient in a wide range.

Ablation study of ISONet components. To demonstrate how each isometric component affects the trainability of deep neural networks, we perform experiments with 34-layer vanilla neural networks and report results in Table 1. In Table 1 (a), we show the result of regular ResNet. If the normalization layers and skip connections in ResNet are removed, the network performance drops by more than 10 points as shown in Table 1 (b). This shows a plain vanilla network cannot be easily optimized. In fact, 34-layer is the deepest vanilla network we found trainable in our experiments. In contrast, ISONet with neither normalization layers nor skip connections can be effectively trained and obtain competitive 70.45% Top-1 accuracy, as shown in Table 1 (g). To our best knowledge, this is the only vanilla network that achieves $> 70\%$ accuracy on ImageNet.

We further show that all the three isometric components (i.e., Delta initialization, orthogonal regularization and SReLU) are indispensable for training deep ISONet. Firstly, enforcing isometry in convolution alone produces $\sim$46% Top-1 Accuracy, which is far lower than ISONet (see Table 1 (c)). The reason is that orthogonal convolution is not suitable for vanilla networks with ReLU activation functions, since the

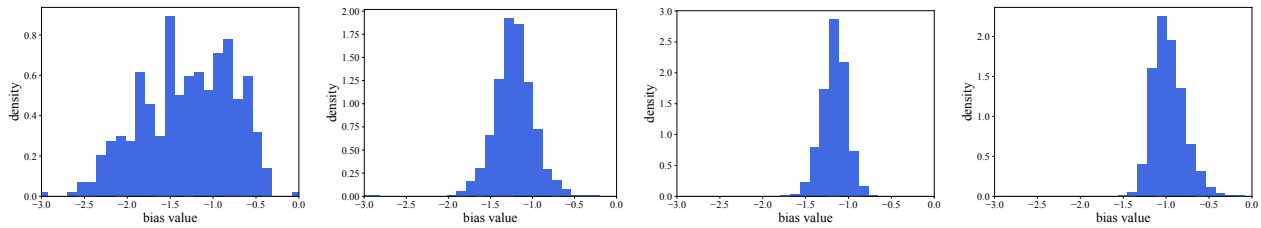

Figure 3. Histogram of the parameter b in SReLU computed for each of the four stages of a 34-layer ISONet.

	Activation	Adaptive b?	Initial b	Top-1 Acc. (%)
(a)	ReLU		-	46.83
(b)	SReLU		-1.0	70.27
(c)		✓	-1.0	70.45
(d)		✓	-2.0	70.04

Table 3. **Effect of parameter b in SReLU for 34-layer ISONets.**

energy may decrease exponentially through multiple layers of the network. Secondly, enforcing isometry in activation functions alone cannot achieve good performance, as shown in Table 1 (d). Thirdly, for convolution layers, isometry must be enforced both at initialization as well as during training. This can be seen in Table 1 (e) and (f), where the performance of ISONet drops by 2 points if either Delta initialization or orthogonal regularization is removed. This demonstrates the critical role of isometry during the entire training process.

Analysis for orthogonal regularization. To understand the effect of the orthogonal regularization term in (13), we conduct experiments with different values of the regularization coefficient γ and report Top-1 accuracy in Table 2. It is shown that our method is not sensitive to the choice of γ in a wide range (from $1e^{-5}$ to $1e^{-4}$), whereas too large ($3e^{-4}$) or no regularization will hurt the performance.

Analysis for SReLU. To verify the design idea of SReLU, we perform an ablation study on the effect of parameter b in (15) and report the results in Table 3. Comparing (b) and (c), a trainable threshold in SReLU is better than a fixed one. Comparing (c) and (d), changing the initial value of b from -1 to -2 reduces the accuracy by less than 0.5 percent. We also plot the histogram of the learned b in SReLU computed within each of the four stages (corresponding to spatial resolutions $\{56, 28, 14, 7\}$) of ISONet and report the results in Figure 3. In all cases, the histogram is bell-shaped and concentrates near -1. This explains why a fixed $b = -1$ produces a good performance as shown in Table 3 (b).

Training very deep ISONets. To demonstrate the effectiveness of isometric principle for training deep networks, we report performance of ISONet with increasing number of layers and report the results in Table 4. Firstly, we observe

	Method	Dropout	Top-1 Accuracy (%)			
			d=18	d=34	d=50	d=101
(a)	ResNet		69.67	73.29	76.60	77.37
(b)		✓	68.91	73.35	76.40	77.99
(c)	Vanilla		65.67	63.09	N/A	N/A
(d)	Vanilla+BN		68.98	69.43	70.00	N/A
(e)	ISONet		67.94	70.45	70.73	70.38
(f)		✓	68.10	70.90	71.20	71.01

Table 4. **ISONet with varying depth $d \in \{18, 34, 50, 101\}$ can be effectively trained on ImageNet.** (a, b) Regular ResNet with or without dropout. (c) Same as ResNet but without BatchNorm and skip connection. (d) Same as ResNet but without skip connection. (e, f) Our ISONet with the same backbone as Vanilla. N/A stands for not converging or obtaining an accuracy lower than 40.

that a standard vanilla network, either without (Table 4 (c)) or with (Table 4 (d)) BatchNorm, cannot converge when networks are extremely deep. In contrast, ISONet maintains a competitive accuracy when the depth is beyond 100 layers. Adding dropout to ISONet further improves the performance by ∼0.5 points. On the other hand, the performance ISONet is not as good as ResNet. In the following, we show that when combined with skip connection, the performance of our R-ISONet is on par with ResNet.

3.3. R-ISONet: a Practical Network without Normalization Layers

In this section, we evaluate the performance of R-ISONet for ImageNet classification and show that it is comparable to regular ResNet, despite the fact that R-ISONet does not contain BatchNorm. The results are reported in Table 5.

From Table 5 (c), ResNet without BatchNorm cannot converge for $d \geq 34$. In contrast, our R-ISONet (Table 5 (f)) can be effectively trained for varying network depth. In addition, R-ISONet is better than previous methods without BatchNorm such as Fixup (Zhang et al., 2019b) (Table 5 (d)).

Since R-ISONet is prone to overfitting due to the lack of BatchNorm, we add dropout layer right before the final

	Method	Dropout	Top-1 Accuracy (%)			
			d=18	d=34	d=50	d=101
(a)	ResNet		69.67	73.29	76.60	77.37
(b)		✓	68.91	73.35	76.40	77.99
(c)	R-Vanilla		65.66	N/A	N/A	N/A
(d)	Fixup		68.63	71.28	72.40	73.18
(e)	Fixup++		67.37	72.56	76.00	76.17
(f)	R-ISONet		69.06	72.17	74.20	75.44
(g)		✓	69.17	73.43	76.18	77.08

Table 5. **Performance of R-ISONet with varying depth $d \in \{18, 34, 50, 101\}$ on ImageNet.** All networks do not have BatchNorm except (a, b). (c) ResNet with all BatchNorm removed. (d, e) Same as (c), but with Fixup initialization (Zhang et al., 2019b). (f, g) Our R-ISONet performs comparable to ResNet in (a, b) and better than all networks in (c, d, e). N/A stands for not converging or obtaining an accuracy lower than 40. Fixup results are reproduced from running the code in (Zhang et al., 2019b) with 100 training epochs. Fixup++ denotes Fixup with Mixup data augmentation.

classifier and report the results in Table 5 (g). The results show that R-ISONet is comparable to ResNet with dropout (Table 5 (b)) and is better than Fixup with Mixup regularization (Zhang et al., 2018).

The superior performance that R-ISONet obtains even without any normalization layers makes it a favorable choice for tasks where statistics in BatchNorm cannot be precisely estimated. In Section 3.4, we demonstrate such advantage of R-ISONet for object detection and instance segmentation tasks.

3.4. Transfer Learning on Object Detection

We further evaluate our method for object detection and instance segmentation tasks on COCO dataset (Lin et al., 2014). The dataset contains 115k training images and 5k validation images. We use the mAP (over different IoU threshold) metric to evaluate the performance of the model, and the "dilated conv5" variant of Faster RCNN (Ren et al., 2015) as our detector. The backbone of the network includes all the convolution layers in R-ISONet. The RoIAlign operator is added on top of the last convolution with the last stage having convolution with stride 1 and dilation 2. Two fully-connected layers are then used to predict the bounding box scores and regression output using this feature. This design follows prior works (Wu et al., 2019; Dai et al., 2017; He et al., 2020). The hyper-parameter settings of both ResNet and R-ISONet follows detectron2 (Wu et al., 2019). We train our model for 90k iterations. The learning rate is initialized to be 0.01 and divide by 10 at 60k and 80k iterations. The training is performed on 8 GPUs, each of which holds 2 images. To keep a fair comparison with standard protocols in object detection, dropout is not added.

	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**

Table 6. **Performance of R-ISONet with varying depth for object detection on COCO dataset.** R-ISONet *outperforms* standard ResNet, indicating that our model has better transfer ability.

The results are reported in Table 6. Although the classification accuracy of our R-ISONet is lower than that of ResNet with the same depth, the detection and instance segmentation performance of R-ISONet is better. This demonstrates that our model has better feature transfer abilities and can mitigate the disadvantages introduced by BatchNorm.

4. Related Work and Discussion

In this paper, we contend that isometry is a central design principle that enables effective training and learning of deep neural networks. In the literature, numerous instantiations of this principle have been explicitly or implicitly suggested, studied, and exploited, often as an additional heuristic or regularization, for improving existing network training or performance. Now supported by the strong empirical evidence presented earlier about isometric learning, we reexamine these ideas in the literature from the unified perspective of isometry.

Arguably, the notion of isometry was first explored in the context of weight initialization (Section 4.1) and then in diverse contexts such as weight regularization (Section 4.2), design of nonlinear activation (Section 4.3), and training techniques for residual networks (Section 4.4).

4.1. In Context of Weight Initialization

Early works on weight initialization are based on the principle that the variance of the signal maintains a constant level as it propagates forward or backward through multiple layers (LeCun et al., 2012; Glorot & Bengio, 2010). A popular method that provides such a guarantee is the Kaiming Initialization (He et al., 2015) which derives a proper scaled Gaussian initialization for weights in vanilla convolutional networks with ReLU activation functions. Derivations for general activation functions beyond ReLU is more difficult, but nonetheless can be achieved by working in the regime where the network is infinitely wide using mean-field theory (Poole et al., 2016; Schoenholz et al., 2016). From the perspective of isometry, the aforementioned works guarantee that the average of the squared singular values of the input-

output Jacobian matrix is close to 1. However, this *does not mean that the Jacobian is an isometry*, which requires that *all* the singular values concentrate at 1. In fact, with the common practice of Gaussian weight initialization, isometry can never be achieved regardless of the choice of activation functions (Pennington et al., 2018).

To address such an issue, orthogonal weight initialization has been extensively studied in the past few years. For linear networks with arbitrary depth, orthogonality of each composing layers trivially leads to an isometry of the input-output Jacobian, and the benefit over Gaussian initialization in terms of training efficiency has been empirically observed (Saxe et al., 2013) and theoretically justified (Hu et al., 2020). For deep nonlinear networks, isometry may also be achieved if the network works in a local regime that the nonlinearity becomes approximately linear (Pennington et al., 2018), and empirical good performance of orthogonal initialization is also observed (Mishkin & Matas, 2016) particularly when combined with proper scaling. This line of work culminates at a recent work (Xiao et al., 2018) which shows that orthogonal initialization enables effective training of ConvNets with 10,000 layers. Nonetheless, the performance of the network in Xiao et al. (2018) is far below the state-of-the-art networks, perhaps due to the fact that isometry beyond the initialization point is not guaranteed.

The Delta initialization adopted in our method is a particular case of orthogonal initialization for convolution kernels. Despite the existence of other orthogonal initialization (Xiao et al., 2018), we advocate the Delta initialization due to its simplicity and good empirical performance in our evaluation for visual recognition tasks.

4.2. In Context of Weight Regularization

A plethora of works has explored the idea of regulating the convolution operators in a ConvNet to be orthogonal in the training process (Harandi & Fernando, 2016; Jia et al., 2017; Cisse et al., 2017; Bansal et al., 2018; Zhang et al., 2019a; Li et al., 2019a; Huang et al., 2020). It is also found that orthogonal regularization helps with training GANs (Brock et al., 2019; Liu et al., 2020) and RNNs (Arjovsky et al., 2016; Lezcano-Casado & Martínez-Rubio, 2019). Despite widely observed performance improvement, the explanation for the effectiveness of orthogonal regularization is rather diverse: it has been justified from the perspective of alleviating gradient vanishing or exploding (Xie et al., 2017a), stabilizing distribution of activation over layers (Huang et al., 2018), improving generalization (Jia et al., 2019), and so on.

In our framework, the benefit of orthogonal regularization lies in that it enforces the network to be close to an isometry. To make our argument precise, we first distinguish between the two related concepts of orthogonal weights and orthogonal convolution. When enforcing orthogonality of convolution kernels in convolutional neural networks, all of the works mentioned above are based on flattening a 4D kernel into a 2D matrix and imposing orthogonality on the matrix. We refer to such a method as imposing *weight orthogonality*, which is not the same as the orthogonality of convolution as discussed in Section 2.2. For example, in Bansal et al. (2018) a kernel $\boldsymbol{A} \in \mathbb{R}^{M\times C\times k\times k}$ is reshaped into a matrix of shape $M \times (C \times k \times k)$ and is enforced to be row-orthogonal. That is,

$$\sum_{c=1}^{C}\langle \text{vec}(\boldsymbol{\alpha}_{mc}), \text{vec}(\boldsymbol{\alpha}_{m'c})\rangle = \begin{cases} 1 & \text{if } m = m', \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

where $\text{vec}(\boldsymbol{\alpha}_{mc})$ denotes a vector of length $k \times k$ obtained from flattening $\boldsymbol{\alpha}_{mc}$. Comparing (16) with (10), it is clear that such a regularization is necessary, but not sufficient, for the operator $\mathcal{A}^*$ to be isometric. Therefore imposing weight orthogonality as in previous works can be interpreted as partially enforcing network isometry as well.

To the best of our knowledge, the only works that have derived orthogonality for convolution kernels are Xiao et al. (2018) and Wang et al. (2020). Xiao et al. (2018) provides a means of constructing orthogonal convolution kernels, but it can only represent a subset of all orthogonal convolutions (Li et al., 2019b). In addition, while the method in Xiao et al. (2018) can be used to generate a random orthogonal convolution, it does not provide a means of enforcing orthogonality in the training process. We become aware of a very recent work (Wang et al., 2020) during the preparation of this paper, which derives the notion of orthogonal convolution that is equivalent to ours. However, the objective of our work is different from theirs. Instead of improving an existing network by adding an additional regularization, our work aims to show that isometry is the most central principle to design effective deep vanilla networks. And as shown in Table 1, using orthogonal regularization alone is not enough for training deep vanilla networks. Meanwhile, the derivation in (Wang et al., 2020) is based on expressing convolution as matrix-vector multiplication using a doubly block-Toeplitz matrix and using the notion of orthogonality for matrices. Such a derivation is limited to 2D convolution while generalization to higher dimensional convolution may become very cumbersome. Moreover, the definition is restricted to discrete time signals and does not adapt to continuous time signals. In contrast, our definition of orthogonal convolution can be extended for higher-dimensional convolution and for continuous times signals by properly redefining the operator "*" in (4), therefore may bear broader interests.

4.3. In Context of Nonlinear Activation

Many of the important variants of nonlinear activation functions developed over the past few years can be interpreted

as obtaining closer proximity to isometry. Early works on improving ReLU such as Leaky ReLU (Maas et al., 2013), Parametric ReLU (He et al., 2015) and Randomized ReLU (Xu et al., 2015), which are generically defined as

$$f(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha x & \text{otherwise}, \end{cases} \tag{17}$$

with the parameter $\alpha > 0$ being fixed, learned and randomly chosen, respectively. Motivation for this family of activation functions comes from the dead neuron issue of ReLU, which states that negative input values give rise to zero gradients that prevents effective training of the network.

The Exponential Linear Unit (ELU) (Clevert et al., 2016) and its variants (Trottier et al., 2017; Klambauer et al., 2017), defined as

$$f(x) = \begin{cases} \gamma x & \text{if } x \geq 0, \\ \alpha(\exp(x/\beta) - 1) & \text{otherwise}, \end{cases} \tag{18}$$

with α, β and γ being trainable or fixed parameters, is another family of activation functions that may solve the dead neuron issue. Another important motivation for ELU is that it pushes the mean of the activation closer to zero, therefore alleviates the issue with regular ReLU that the bias of the input signal is always shifted towards positivity. Such benefit is further confirmed by a theoretical study (Klambauer et al., 2017) which shows that a zero mean and unit variance signal converges towards being zero mean and unit variance after propagating over multiple layers. Empirically, a very recent work (Huang et al., 2019) demonstrates that a 50-layer neural network without batch normalization and skip connection may be effectively trained for speech recognition tasks by using ELU instead of ReLU.

The SReLU that we adopt in the isometric network is closely related to the ELU family. To the best of our knowledge, a variant of SReLU (in which the parameter is fixed and not optimizable) first appears in Clevert et al. (2016), the paper that proposes ELU, where it is shown to outperform ReLU and Leaky ReLU by a large margin and has similar performance as ELU. Besides, SReLU is also adopted in (Qiu et al., 2018) as an important baseline. Recently, SReLU is rediscovered in (Xiang & Li, 2017; Singh & Krishnan, 2020) for alleviating the bias shifting issue.

We argue that aside from the benefits claimed by the original papers, the activation function families (17), (18) as well as the SReLU are advantageous over the regular ReLU as they bring the network closer to an isometry. At an intuitive level, both (17) (with $0 < \alpha < 1$), (18) (with $\gamma = 1$ and $0 < \alpha < 1$ and $\beta = 1$) and SReLU lie between ReLU and identity, therefore may be considered as a tradeoff between obtaining nonlinearity and isometry. From a theoretical perspective, the analysis from (Pennington et al., 2018) reveals that SReLU combined with random orthogonal initialization can achieve dynamic isometry, while regular ReLU cannot.

4.4. In Context of Residual Learning

Ever since the inception of residual learning (He et al., 2016), there has been no lack of effort in further enhancing its performance. In particular, there is a line of work that demonstrates the benefit of reducing the energy on the residual branch of a ResNet. This is studied from different perspectives such as dynamic isometry (Tarnowski et al., 2019; Qiu et al., 2018), training failure modes (Taki, 2017; Hanin & Rolnick, 2018; Balduzzi et al., 2017). Such design is also empirically applied in different ways such as BatchNorm initialization (Goyal et al., 2017), adding a small scalar in the residual branch (Zagoruyko & Komodakis, 2017; Szegedy et al., 2017), and 0-initialized convolution (Zhang et al., 2019b), all of which brings the network closer to an isometry. With a careful design along this line of study (Zhang et al., 2019b), deep ResNet can be effectively trained to obtain competitive performance on visual recognition tasks, even without the help of BatchNorm layers. This line of research as well as our simple R-ISONet verifies the effectiveness of isometry in the design the network structures of residual learning.

5. Conclusion

In this paper, we have demonstrated through a principle-guided design and strong empirical evidence why isometry is likely to be the main key property that enables effective learning of deep networks and ensures high performance on real visual recognition tasks. With this design principle, one may achieve competitive performance with much simplified networks and eased training. We also argue that isometric learning provides a unified principle that helps explain numerous ideas, heuristics and regularizations scattered in the literature that exploit this property and are found effective. We believe that the isometry principle may help people design or discover new simple network operators and architectures with much-improved performance in the future.

Acknowledgements

The authors acknowledge support from Tsinghua-Berkeley Shenzhen Institute Research Fund. Haozhi is supported in part by DARPA Machine Common Sense. Xiaolong is supported in part by DARPA Learning with Less Labels. We thank Yaodong Yu and Yichao Zhou for insightful discussions on orthogonal convolution. We would also like to thank the members of BAIR for fruitful discussions and comments.

References

Arjovsky, M., Shah, A., and Bengio, Y. Unitary evolution recurrent neural networks. In *ICML*, 2016.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv*, 2016.

Bai, S., Koltun, V., and Kolter, J. Z. Multiscale deep equilibrium models. *arXiv*, 2020.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question? In *ICML*, 2017.

Bansal, N., Chen, X., and Wang, Z. Can we gain more from orthogonality regularizations in training deep networks? In *NIPS*, 2018.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv*, 2020.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *ICML*, 2017.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). In *ICLR*, 2016.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. Deformable convolutional networks. In *ICCV*, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Denker, J. S., Gardner, W., Graf, H. P., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., Baird, H. S., and Guyon, I. Neural network recognizer for hand-written zip code digits. In *NIPS*, 1989.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv*, 2017.

Hanin, B. and Rolnick, D. How to start training: The effect of initialization and architecture. In *NIPS*, 2018.

Harandi, M. and Fernando, B. Generalized backpropagation,\'{E} tude de cas: Orthogonality. *arXiv*, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Hu, W., Xiao, L., and Pennington, J. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *ICLR*, 2020.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *CVPR*, 2017.

Huang, L., Liu, X., Lang, B., Yu, A. W., Wang, Y., and Li, B. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *AAAI*, 2018.

Huang, L., Liu, L., Zhu, F., Wan, D., Yuan, Z., Li, B., and Shao, L. Controllable orthogonalization in training dnns. In *CVPR*, 2020.

Huang, Z., Ng, T., Liu, L., Mason, H., Zhuang, X., and Liu, D. Sndcnn: Self-normalizing deep cnns with scaled exponential linear units for speech recognition. *arXiv*, 2019.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Jia, K., Tao, D., Gao, S., and Xu, X. Improving training of deep neural networks via singular value bounding. In *CVPR*, 2017.

Jia, K., Li, S., Wen, Y., Liu, T., and Tao, D. Orthogonal deep neural networks. *PAMI*, 2019.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-normalizing neural networks. In *Advances in neural information processing systems*, pp. 971–980, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Le, Q. V., Jaitly, N., and Hinton, G. E. A simple way to initialize recurrent networks of rectified linear units. *arXiv*, 2015.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1990.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural networks: Tricks of the trade*. Springer, 2012.

Lezcano-Casado, M. and Martínez-Rubio, D. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. *arXiv preprint*, 2019.

Li, J., Li, F., and Todorovic, S. Efficient riemannian optimization on the stiefel manifold via the cayley transform. In *ICLR*, 2019a.

Li, Q., Haque, S., Anil, C., Lucas, J., Grosse, R. B., and Jacobsen, J.-H. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *NIPS*, 2019b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Liu, B., Zhu, Y., Fu, Z., de Melo, G., and Elgammal, A. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In *AAAI*, 2020.

Luo, P., Ren, J., Peng, Z., Zhang, R., and Li, J. Differentiable learning-to-normalize via switchable normalization. *arXiv*, 2018.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

Mishkin, D. and Matas, J. All you need is a good init. In *ICLR*, 2016.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Nica, B. The mazur-ulam theorem. *arXiv*, 2013.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

Pennington, J., Schoenholz, S. S., and Ganguli, S. The emergence of spectral universality in deep networks. In *AISTATS*, 2018.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. In *NIPS*, 2016.

Qiu, S., Xu, X., and Cai, B. Frelu: Flexible rectified linear units for improving convolutional neural networks. In *ICPR*, 2018.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv*, 2013.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. *arXiv*, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Singh, S. and Krishnan, S. Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. In *CVPR*, 2020.

Srivastava, R. K., Greff, K., and Schmidhuber, J. Highway networks. *arXiv*, 2015.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. A., and Hardt, M. Test-time training for out-of-distribution generalization. *arXiv*, 2019.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *CVPR*, 2015.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

Taki, M. Deep residual networks and weight initialization. *arXiv*, 2017.

Tarnowski, W., Warchol, P., Jastrzebski, S., Tabor, J., and Nowak, M. Dynamical isometry is achieved in residual networks in a universal way for any activation function. In *AISTATS*, 2019.

Trottier, L., Gigu, P., Chaib-draa, B., et al. Parametric exponential linear unit for deep convolutional neural networks. In *ICMLA*, 2017.

Wang, J., Chen, Y., Chakraborty, R., and Yu, S. Orthogonal convolutional neural networks. In *CVPR*, 2020.

Wu, Y. and He, K. Group normalization. In *ECCV*, 2018.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

Xiang, S. and Li, H. On the effects of batch and weight normalization in generative adversarial networks. *arXiv*, 2017.

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S. S., and Pennington, J. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *ICML*, 2018.

Xie, D., Xiong, J., and Pu, S. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *CVPR*, 2017a.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017b.

Xu, B., Wang, N., Chen, T., and Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv*, 2015.

Zagoruyko, S. and Komodakis, N. Diracnets: Training very deep neural networks without skip-connections. *arXiv*, 2017.

Zhang, G., Niwa, K., and Kleijn, W. Approximated orthonormal normalisation in training neural networks. *arXiv*, 2019a.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Zhang, H., Dauphin, Y. N., and Ma, T. Fixup initialization: Residual learning without normalization. In *ICLR*, 2019b.