Norm-Preservation: Why Residual Networks Can Become Extremely Deep?

Alireza Zaeemzadeh
University of Central Florida
zaeemzadeh@eecs.ucf.edu

Nazanin Rahnavard
University of Central Florida
nazanin@eecs.ucf.edu

Mubarak Shah
University of Central Florida
shah@crcv.ucf.edu

Abstract

Augmenting deep neural networks with skip connections, as introduced in the so called ResNet architecture, surprised the community by enabling the training of networks of more than 1000 layers with significant performance gains. It has been shown that identity skip connections eliminate singularities and improve the optimization landscape of the network.

This paper deciphers ResNet by analyzing the of effect of skip connections in the backward path and sets forth new theoretical results on the advantages of identity skip connections in deep neural networks. We prove that the skip connections in the residual blocks facilitate preserving the norm of the gradient and lead to well-behaved and stable back-propagation, which is a desirable feature from optimization perspective. We also show that, perhaps surprisingly, as more residual blocks are stacked, the network becomes more norm-preserving. Traditionally, norm-preservation is enforced on the network only at beginning of the training, by using initialization techniques. However, we show that identity skip connection retain norm-preservation during the training procedure. Our theoretical arguments are supported by extensive empirical evidence.

Can we push for more norm-preservation? We answer this question by proposing zero-phase whitening of the fully-connected layer and adding norm-preserving transition layers. Our numerical investigations demonstrate that the learning dynamics and the performance of ResNets can be improved by making it even more norm preserving through changing only a few blocks in very deep residual networks. Our results and the introduced modification for ResNet, referred to as Procrustes ResNets, can be used as a guide for studying more complex architectures such as DenseNet, training deeper networks, and inspiring new architectures.

1 Introduction

Deep neural networks have progressed rapidly during the last few years, achieving outstanding, sometimes super human, performance [1]. It is known that the depth of the network, i.e., number of stacked layers, is of decisive significance. It is shown that as the networks become deeper, they are capable of representing more complex mappings [2]. However, deeper networks are notoriously harder to train. As the number of layers is increased, optimization issues arise and, in particular, avoiding vanishing/exploding gradients is essential to optimization stability of such networks. Batch normalization, regularization, and initialization techniques have shown to be useful remedies for this problem [3, 4].

Furthermore, it has been observed that as the networks become increasingly deep, the performance gets saturated and even deteriorates [5]. This problem has been addressed by many recent network designs [5–8]. All of these approaches use the same design principle: skip connections. This simple trick makes the information flow across the layers easier by bypassing the activations from one layer

Preprint. Work in progress.

to the next using skip connections. Highway Networks [7], ResNets [5, 6], and DenseNets [8] have consistently achieved state-of-the-art performances by using skip connections in different network topologies. The main goal of skip connection is to enable the information to flow through many layers without attenuation. In all of these efforts, it is observed empirically that it is crucial to keep the information path *clean* by using identity mapping in the skip connection. It is also observed that more complicated transformations in the skip connection lead to more difficulty in optimization, even though such transformations have more representational capabilities [6]. This observation indicates that *identity* skip connection, while provides adequate representational ability, has a great feature of optimization stability, enabling deeper well-behaved networks.

Since the introduction of Residual Networks (ResNets) [5, 6], there has been some effort on understanding how the residual blocks make help the optimization process and how they improve the representational ability of the networks. Authors in [9] showed that skip connection eliminates the singularities caused by the model non-identifiability. This makes the optimization of deeper networks feasible and faster. Similarly, to understand the optimization landscape of ResNets, authors in [10] prove that linear residual networks have no critical points other than the global minimum. This is in contrast to plain linear networks, in which other critical points may exist [11]. Furthermore, authors in [12] show that as depth increases, gradients of plain networks resemble white noise and become less correlated. This phenomenon, which is referred to as *shattered gradient* problem, makes training more difficult. Then, it is demonstrated that residual networks reduce shattering, compared to plain networks, leading to numerical stability and easier optimization.

In this paper, we present and analytically study another desirable effect of identity skip connection: *preserving the norm of error gradient* as it propagates in the backward path. We show theoretically and empirically that each residual block in ResNets is *increasingly norm-preserving* as the network becomes *deeper*. This interesting result is in contrast with hypothesis provided in [13], which states that residual networks avoid vanishing gradient *solely* by shortening the effective path of the gradient.

Furthermore, it is known that by modifying the distribution from which the initial weights are sampled, we can make the training easier [3, 14]. This is done by keeping the variance of weights gradient the same across layers. However, as observed in [14] and verified by our experiments, using such initialization methods, although the network is initially fairly norm-preserving, the norms of the gradients diverge as training progresses. Instead, we show that identity skip connection enforces the norm-preservation during the training, leading to well-conditioning and easier training.

We analyze the role of identity mapping as skip connection in the ResNet architecture from a theoretical perspective. Moreover, we use the insight gained from our theoretical analysis to propose modifications to some of the building blocks of the ResNet architecture. Two main contributions of this paper are as follows.

- **Proving the Norm Preservation of ResNets**: We show that having identity mapping in the shortcut path leads to norm-preserving building blocks. Specifically, identity mapping shifts all the singular values of the transformations towards 1. This makes the optimization of the network much easier by preserving the magnitude of the gradient across the layers. Furthermore, we show that, perhaps surprisingly, *as the network becomes deeper, its building blocks become more norm-preserving*. Hence, the gradients can flow smoothly through very deep networks, making it possible to train such networks. Our experiments validate our theoretical arguments.

- **Enforcing More Norm Preservation**: Using insights from our theoretical investigation, we propose important modifications to the fully connected layer and the transition blocks in the ResNet architecture. Since these blocks do not use identity mapping as the skip connection, they do not preserve the norm of the gradient in general. The transition blocks are blocks that change the number of channels and feature map size of the activations, making it impossible to use identity mapping as the skip connection. Thus, we propose to change the dimension of the activations in a norm preserving manner, such that the network becomes even more norm-preserving. Furthermore, we use zero-phase component analysis (ZCA) to ensure that all the singular values of the linear transformation in the fully connected layer are equal to 1. We refer to the proposed architecture as Procrustes ResNet (ProcResNet). Our experiments demonstrates that the proposed norm-preserving blocks are able to improve the optimization stability and performance of ResNets.

2 Norm-Preservation of Residual Networks

Due to the complexity of general non-linear networks, it is common practice to consider the simplified linear model to study the behavior of the networks and develop theoretical guarantees [10, 15]. Such models are analyzed to provide insights to general non-linear networks. Thus, in this paper, we model each residual block as:

$$\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + \boldsymbol{W}_l \boldsymbol{x}_l, \tag{1}$$

where, $\boldsymbol{x}_l, \boldsymbol{x}_{l+1} \in \mathbb{R}^N$ are the input and output of the l^{th} residual block, respectively, with dimension N. The weight matrix $\boldsymbol{W}_l \in \mathbb{R}^{N \times N}$ is the tunable linear transformation. The goal of learning is to compute a function $\boldsymbol{d} = \mathcal{M}(\boldsymbol{z}, \mathcal{W})$, where $\boldsymbol{z}$ is the input, $\boldsymbol{d}$ is its corresponding output, and $\mathcal{W}$ is the collection of all adjustable linear transformations, i.e., $\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_L$. In the case of simplified linear residual networks, function $\mathcal{M}(\boldsymbol{z}, \mathcal{W})$ is a stack of L residual blocks, as formulated in (1). Mathematically speaking, we have:

$$\boldsymbol{d} = \mathcal{M}(\boldsymbol{z}, \mathcal{W}) = \prod_{l=1}^{L} (\boldsymbol{I} + \boldsymbol{W}_l) \boldsymbol{z}, \tag{2}$$

where I is an $N \times N$ identity matrix. $\mathcal{M}(\boldsymbol{z}, \mathcal{W})$ is used to learn a linear mapping $\boldsymbol{R} \in \mathbb{R}^{N \times N}$ from its inputs and outputs. Furthermore, assume that $\boldsymbol{d}$ is contaminated with independent identically distributed (i.i.d) Gaussian noise, i.e., $\hat{\boldsymbol{d}} = \boldsymbol{R}\boldsymbol{z} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a zero mean noise vector with covariance matrix $\boldsymbol{I}$. Hence, our objective is to minimize the expected error of the maximum likelihood estimator as:

$$\min_{\mathcal{W}} \mathcal{E}(\mathcal{W}) = \mathbb{E}\{\frac{1}{2} \|\hat{\boldsymbol{d}} - \mathcal{M}(\boldsymbol{z}, \mathcal{W})\|_2^2\}, \tag{3}$$

where the expectation $\mathcal{E}$ is with respect to the population $(\boldsymbol{z}, \boldsymbol{d})$.

Our following main theorem states that the solution of the optimization problem stated in (3), using the residual mapping formulated in (2), is norm-preserving in the backward path. We show that, at each residual block, the norm of the gradient with respect to the input is close to the norm of gradient with respect to the output. In other words, *the residual block with identity mapping, as the skip connection, preserves the norm of the gradient in the backward path*. This results in several useful characteristics such as avoiding vanishing/exploding gradient, stable optimization, and performance gain.

Theorem 1. *For learning a linear map, $\boldsymbol{R} \in \mathbb{R}^{N \times N}$, between its input $\mathbf{z}$ and output $\mathbf{d}$ contaminated with i.i.d Gaussian noise, using a network consisting of L linear residual blocks of form $\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + \boldsymbol{W}_l \boldsymbol{x}_l$, there exists a global optimum for $\mathcal{E}(.)$, as defined in (3), such that for all residual blocks we have*

$$(1-\delta)\|\frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_{l+1}}\|_2 \leq \|\frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_l}\|_2 \leq (1+\delta)\|\frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_{l+1}}\|_2$$

$$\textit{and} \quad \delta \leq \frac{c}{L}$$

for $L \geq 3\gamma$, where $c = 2(\sqrt{\pi} + \sqrt{3\gamma})^2$ and $\gamma = \max(|\log \sigma_{max}(\boldsymbol{R})|, |\log \sigma_{min}(\boldsymbol{R})|)$.

Proof. See Section A in the supplementary materials. □

In words, the norm of the gradient does not change significantly, as it is backpropagated through the layers. This means that norm-preservation is enforced throughout the training process by the structure of the network, not just at the beginning of the training by good initialization. Thus, the gradient will have very similar magnitude at different layers, which leads to well-conditioning and faster convergence [14]. One interesting implication of Theorem 1 is that as L, the number of layers increases, the residual blocks become more norm-preserving. This is a very desirable feature because vanishing or exploding gradient often occurs in deeper network architectures. However, by utilizing residual blocks, as more blocks are stacked, we should be less concerned with the norm of the gradient.

Although Theorem 1 assumes a linear model for the residual block, our extensive empirical experiments, which will be presented in Section 4, show that non-linear residual blocks exhibit the same behavior.

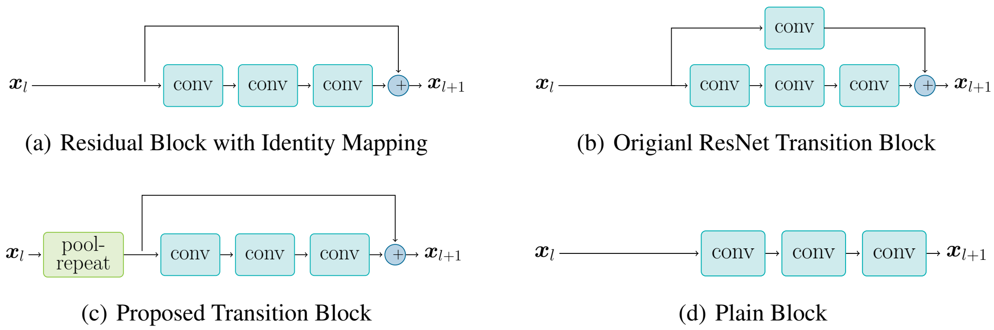


Figure 1: Block diagram of building blocks of different architectures. Each conv block represents a sequence of batch normalization, ReLU, and convolution layers.

3 Procrustes Residual Network (ProcResNet): Residual Network with Enhanced Norm-Preservation

In practice, residual networks contain four different types of blocks: (i) convolution layer (first layer), (ii) fully connected layer (last layer), (iii) transition blocks (which change the dimension of the activations), and (iv) residual blocks with identity skip connection, which we also refer to as non-transition blocks. Theoretical investigation presented in Section 2 holds only for residual blocks with identity mapping as the skip connection. If the benefits of residual networks can be explained, at least partly, by norm-preservation, then one can improve them by alternative methods for preserving the norm. In this section, we propose to modify the transition blocks and the fully connected layer of ResNet architecture, to make them norm-preserving. Due to multiplicative effect through the layers, making these layers norm-preserving may be important, although they make up a small portion of the network. In the following, we discuss how to preserve the norm of the back-propagated gradients across all the blocks of the network. Since the convolution layer is the first layer of the network (last layer in backward path) and we are not interested in the gradient at the input of network, we ignore its norm-preservation capability. However, there exist methods to make these layers norm-preserving as well [16].

3.1 Norm-Preserving Fully-Connected Layer

Since we are specifically interested in the norm-preservation in the backward path, the linear transformation in the fully-connected layer needs to be *co-isometric*. This means that for matrix $\boldsymbol{W^T}$ $\sigma_{min}(\boldsymbol{W}^T) = \sigma_{max}(\boldsymbol{W}^T) = 1$. This can be done using zero-phase component analysis (ZCA). Intuitively, we enforce the norm-preservation constraint by setting all the singular values of $\boldsymbol{W}^T$ equal to 1.

Assume $\boldsymbol{U\Sigma V}^T = svd(\boldsymbol{W}^T)$ be the singular value decomposition of $\boldsymbol{W}^T$. We can enforce the co-isometry constraint by projecting the matrix onto the feasible set, i.e., $\hat{\boldsymbol{W}}^T = \boldsymbol{UV}^T$, where $\hat{\boldsymbol{W}}$ is the updated norm-preserving matrix. This is known as the zero-phase whitening transformation [17, 18]. In other words, we are setting $\boldsymbol{\Sigma} = \boldsymbol{I}$, which makes $\hat{\boldsymbol{W}}$ a co-isometric transform. Since we want to keep the co-isometry throughout training, we implement this projection both at initialization and after each gradient descent iteration. This projection is mathematically equivalent to adding co-isometry constraint, i.e., $\boldsymbol{WW}^T = \boldsymbol{I}$, to the optimization problem [19, 20]. This is also closely related to *Procrustes* problems, in which the goal is to find the closest orthogonal matrix to a given matrix [21].

3.2 Norm-Preserving Transition Blocks

As depicted in Figure 1(b), in the original ResNet architecture, the dimension changing blocks, also known as transition blocks, use 1×1 convolution with stride of 2 in their skip connections to match the dimension of input and output activations. Such transition blocks are not norm-preserving in general.

The goal of the transition blocks in ResNet architecture is to double the number of channels and half the feature map size (in each dimension). To change the dimension in a norm-preserving manner, we utilize ℓ_2-norm pooling followed by a repetition. ℓ_2-pooling is utilized to reduce the feature map size and is always norm-preserving both in backward and forward paths. Repetition is used to double the number of channels. This block is not norm-preserving in general. However, our experiments show that, due to the special structure of gradients, the backward transformation is fairly norm-preserving.

Figure 1(c) shows the diagram of the proposed transition block, where ℓ_2-pooling and repetition are used to change the dimension. Hence, we are able to exploit a regular residual block with identity mapping. Compared to regular transition block in Figure 1(b), the proposed version uses identity skip connection, which makes it norm preserving.

4 Experiments

To validate our theoretical investigation, presented in Section 2, and to empirically demonstrate the behavior and effectiveness of the proposed modifications, we experimented with Residual Network (ResNet) and the proposed Procrustes Residual Network (ProcResNet) architecture on CIFAR10 and CIFAR100 datasets. Training and testing datasets contain 50,000 and 10,000 images of visual classes, respectively [17]. Standard data augmentation (flipping and shifting), same as [5, 6, 8], is adopted. Furthermore, channel means and standard deviations are used to normalize the images. The network is trained using stochastic gradient descent. The weights are initialized using the method proposed in [3] and the initial learning rate is 0.1. Batch size of 128 is used for all the networks, except networks with 1001 layers, for which batch size of 64 is used. The weight decay is 10^{-4} and momentum is 0.9. The results are based on the top-1 classification accuracy.

Experiments are performed on three different network architectures: 1. **ResNet** contains one convolution layer, L residual blocks, three of which are transition blocks, and one fully connected layer. Each residual block consists of three convolution layers, as depicted in Figure 1(a) and Figure 1(b), resulting in a network of depth $3L+2$. This is the same architecture as in [6]. 2. **ProcResNet** has the same architecture as ResNet, except the transition and fully connected layers are modified, as explained in Section 3. 3. **Plain** network is also same as ResNet without the skip connection in all the L residual blocks, as shown in Figure 1(d). Further details on network architectures and implementation details are presented in Section C of the supplementary materials.

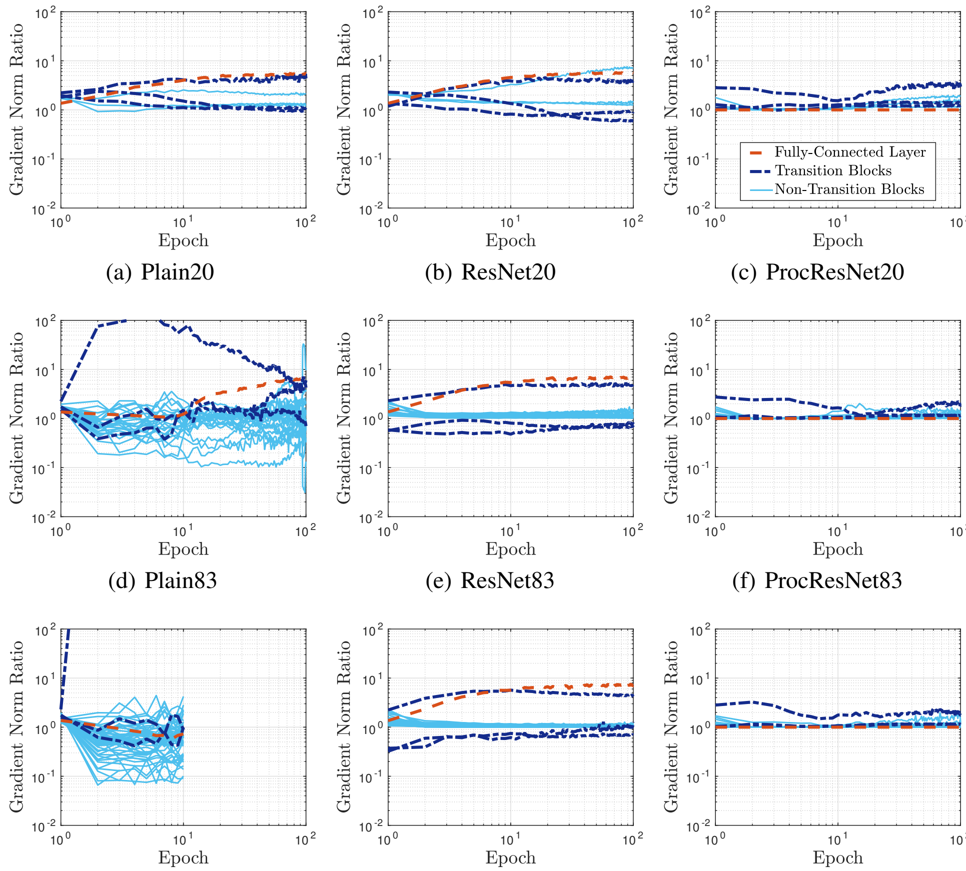
4.1 Norm-Preservation

In the first set of experiments, the behavior of different architectures is studied as the function of network depth. To this end, the ratio of gradient norm at output to gradient norm at input, i.e., $\|\frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_{l+1}}\|_2$ to $\|\frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_l}\|_2$, is captured for all the residual blocks[1], both transition and non-transition, and the fully connected layer, resulting in $L+1$ ratios. Figure 2 shows the ratios for different blocks over training epochs. We ran the training for 100 epochs, without decaying the learning rate. Plain network (Figure 2.(g)) with 164 layers became numerically unstable and the training procedure stopped after 10 epochs.

Several interesting observations can be made from this experiment:

- This experiment emphasizes the fact that one needs more than careful initialization to make the network norm-preserving. Although the plain network is initially norm-preserving, the range of the gradient norm ratios becomes very large and diverges from 1, as the parameters are updated. However, ResNet and ProcResNet are able to enforce the norm-preservation during training procedure by using identity skip connection.
- As the networks become deeper, the plain network becomes less norm preserving, which leads to numerical instability, optimization difficulty, and performance degradation. On the contrary, the non-transition blocks, the blocks with identity mapping as skip connection, of ResNet and

[1]In Plain architecture, which does not have skip connections, the gradient norm ratio is obtained at the input and output of its building blocks as depicted in Figure 1(d).



(a) Plain20 (b) ResNet20 (c) ProcResNet20

(d) Plain83 (e) ResNet83 (f) ProcResNet83

(g) Plain164 (h) ResNet164 (i) ProcResNet164

Figure 2: Training on CIFAR10. Gradient norm ratio over first 100 epochs for different block types: fully connected layer, transition blocks (blocks that change the dimension), and non-transition blocks (blocks that do not change the dimension).

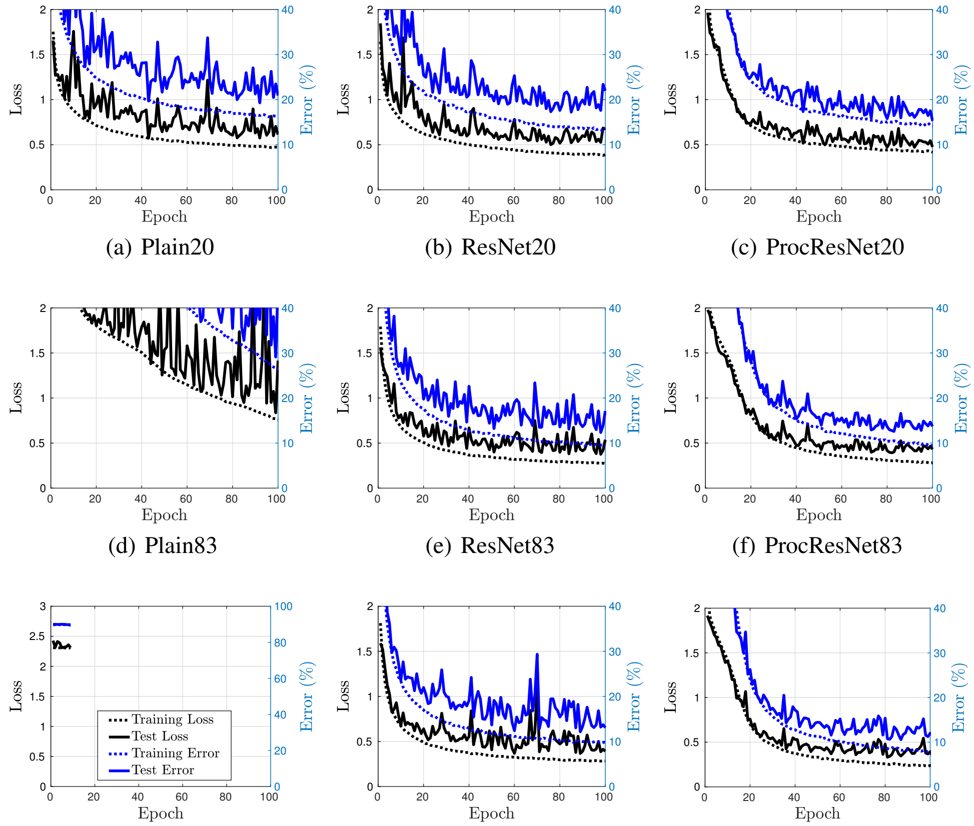
ProcResNet become more norm preserving. This is in line with our theoretical investigation for linear residual networks, which states that as we stack more residual blocks the network becomes more norm-preserving.

- Comparing Plain83 (Figure 2(d)) and Plain164 (Figure 2(h)) networks, it can be observed that most of the blocks behave fairly similar, except one transition block. Specifically, in Plain83, the gradient norm ratio of the first transition block goes up to 100 in the first few epochs. But it eventually decreases and the network is able to converge. On the other hand, in Plain164, the gradient norm ratio of the same block becomes too large, which makes the network unable to converge. Hence, a single block is enough to make the optimization difficult and numerically unstable. This highlights the fact that it is necessary to enforce norm-preservation on all the blocks.

- In ResNet83 (Figure 2(e)) and ResNet164 (Figure 2(h)), it is easy to notice that only 4 blocks are not norm-preserving: 3 transition blocks and the fully connected layer. As mentioned earlier, due to multiplicative effect, the magnitude of the gradient will not be preserved because of these few blocks. This issue is non-existent in ProcResNet, where all the blocks are more norm-preserving.

- The behaviors of ResNet and Plain architectures are fairly similar for depth of 20. This was somehow expected, since it is known that the performance gain achieved by ResNet is more significant in deeper architectures [5]. However, even for depth of 20, ProcResNet architecture is more norm preserving.

- Using ZCA projection, the gradient norm ratio of the fully connected layer is always equal to 1 in ProcResNet architecture.



(a) Plain20 (b) ResNet20 (c) ProcResNet20

(d) Plain83 (e) ResNet83 (f) ProcResNet83

(g) Plain164 (h) ResNet164 (i) ProcResNet164

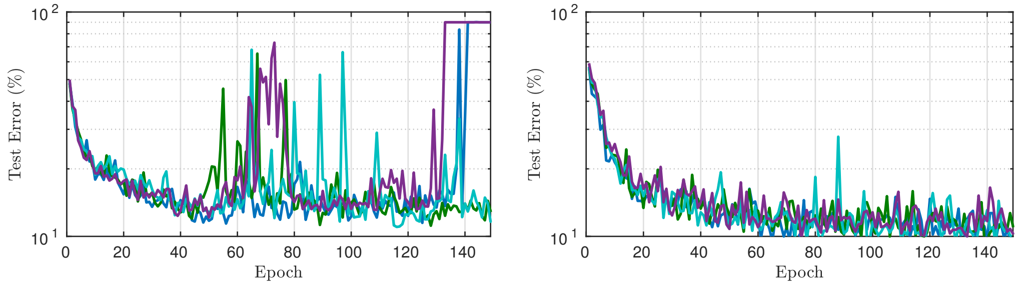
Figure 3: Loss (black lines) and error (blue lines) during training procedure on CIFAR10. Solid lines represent the test values and dotted lines represent the training values.

This experiment both validates our theoretical argument, clarifies some of the inner workings of ResNet architecture, and shows the effectiveness of the proposed modifications in ProcResNet. It is evident that, as stated in Theorem 1, addition of identity skip connection makes the blocks increasingly more norm-preserving, as the network becomes deeper. Furthermore, we have been able to make the network more norm-preserving by applying the changes proposed in Section 3. Additional experiments are provided in Section B of supplementary materials.

4.2 Optimization Stability and Learning Dynamics

In the next set of experiments, numerical stability and learning dynamics of different architectures is examined. For that, loss and classification error, in both training and testing phases, are depicted in Figure 3. This experiment illustrates that how optimization stability of deep networks is improved significantly and how it can be further improved by having norm preservation in mind during the design procedure.

As depicted in Figure 3, unlike the plain network, training error and loss curves corresponding to ResNet and ProcResNet architectures are consistently decreasing as the number of layers increases, which was the main motivation behind proposing residual blocks [5]. Moreover, Figure 3(a) and Figure 3(d) show that the plain networks have a poor generalization performance. The fluctuations in testing error shows that the points along the optimization path of the plain networks do not generalize well. This issue is also present, to a lesser extent, in ResNet architecture. Comparing Figure 3(h) and 3(b), we can see that the fluctuations are more apparent in deeper ResNet networks. However, In ProcResNet architecture, the amplitude of the fluctuations is smaller and does not change as the



(a) ResNet1001

(b) ProcResNet1001

Figure 4: Training on CIFAR10. Test error for (a) ResNet and (b) ProcResNet architectures with 1001 layers for 4 different runs. 2 of the ResNet runs did not converge. Each solid line represents the test error of a single run. Batch size is 64 and learning rate is 0.1

Table 1: Performance of different methods on CIFAR-10 and CIFAR-100 using moderate data augmentation.

Architecture	# Params	Error (%)	
		CIFAR10	CIFAR100
ResNet164	1.70M	5.46	24.33
ProcResNet164	1.67M	5.01 (5.06 $\pm$ 0.08)	23.85 (23.90 $\pm$ 0.07)
ResNet1001	10.32M	4.62 (4.69 $\pm$ 0.20)	22.71 (22.68 $\pm$ 0.22)
ProcResNet1001	10.29M	4.47 (4.47 $\pm$ 0.17)	21.87 (21.89 $\pm$ 0.12)

depth of the network is increased. This indicates that ProcResNet architecture is taking a better path toward the optimum and has better generalization performance.

This fact is more evident in the extreme case of 1001 layers. Figure 4 illustrates the testing error of ResNet and ProcResNet with 1001 layers for 4 different runs. We let the training to progress up to 150 epochs, without learning rate decay. It is evident that ResNet1001 is more unstable, compared to shallower ResNet architectures and ProcResNet1001. In fact, in 2 out of 4 runs, ResNet1001 ended up having error of 90% (random guess) at epoch 150. In original ResNet implementation [6], the learning rate is decreased by a factor of 10 at epoch 81, which avoids the instability.

This means that, by modifying only a few blocks in an extremely deep network, we have been able to make the network more stable and improved the learning dynamics. This emphasizes the utmost importance of norm-preservation of all blocks in avoiding optimization difficulties of very deep networks. Moreover, this sheds light on the reasons why architectures using residual blocks, or identity skip connection in general, perform so well and are easier to optimize.

4.3 Classification Performance

Finally, we show the impact of the proposed norm-preserving transition blocks and the fully connected layer on the classification performance of ResNet. Table 1 compares the performance of ResNet with and without the proposed transition blocks. The results are median of 5 runs with mean $\pm$ std inside the brackets. The results for standard ResNet are the best results reported by [6] and the results of ProcResNet is obtained by making the proposed changes to standard ResNet implementation provided by its authors at `https://github.com/KaimingHe/resnet-1k-layers`.

To leverage the stability achieved by the proposed changes, the learning rate for ProcResNet is divided by 10 at epochs 150 and 225 and the network is trained for 300 epochs. As shown in Figure 4, due to stability issues, original ResNet cannot be trained for 150 epochs without learning rate decay. Table 1 shows that the proposed network performs slightly better than the standard ResNet. This performance gain comes without increasing the number of parameters and by changing only 4 blocks. The total number of residual blocks for ResNet164 and ResNet1001 is 54 and 333, respectively. This illustrates that we are able to improve the performance by changing a tiny portion of the network and emphasizes the importance of norm-preservation in the performance of neural networks.

5 Conclusions

This paper theoretically analyzes building blocks of residual networks and shows that adding identity skip connection makes the residual blocks norm-preserving. Furthermore, the norm-preservation is enforced during the training procedure, which makes the optimization stable and improves the performance. This is in contrast to initialization techniques, such as [14], which ensure norm-preservation only at the beginning of the training. Our experiments validate our theoretical investigation by showing that (i) identity skip connection results in norm preservation, (ii) residual blocks become more norm-preserving as the network becomes deeper, and (iii) we can make the training more stable by making the network more norm-preserving. Our proposed modification of ResNet, Procrustes ResNet, enforces norm-preservation on the fully-connected layer and transition blocks of the network and is able to achieve better optimization stability and performance. Our findings can be seen as design guidelines for very deep architectures. By having norm-preservation in mind, we will be able to train extremely deep networks and alleviate the optimization difficulties of such networks. Furthermore, an interesting future direction is to see if and how our theoretical analysis applies to DenseNets or any general non-linear network with identity skip connection.

References

[1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, 10 2017.

[2] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in Neural Information Processing Systems 27*, pp. 2924–2932, Curran Associates, Inc., 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, IEEE, 12 2015.

[4] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456, 07–09 Jul 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, 6 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision – ECCV 2016*, pp. 630–645, Springer International Publishing, 2016.

[7] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems 28*, pp. 2377–2385, Curran Associates, Inc., 2015.

[8] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, IEEE, 7 2017.

[9] E. Orhan and X. Pitkow, "Skip connections eliminate singularities," in *International Conference on Learning Representations*, 2018.

[10] M. Hardt and T. Ma, "Identity Matters in Deep Learning," in *International Conference on Learning Representations*, 2017.

[11] K. Kawaguchi, "Deep learning without poor local minima," in *Advances in Neural Information Processing Systems 29*, pp. 586–594, Curran Associates, Inc., 2016.

[12] D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 342–350, 06–11 Aug 2017.

[13] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 550–558, Curran Associates, Inc., 2016.

[14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.), vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256, 13–15 May 2010.

[15] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *International Conference on Learning Representations*, 2014.

[16] D. Xie, J. Xiong, and S. Pu, "All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[17] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," tech. rep., 2009.

[18] A. J. Bell and T. J. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327 – 3338, 1997.

[19] R. Everson, "Orthogonal, but not orthonormal, procrustes problems," *Advances in computational Mathematics*, vol. 3, no. 4, 1998.

[20] A. Edelman, T. A. Arias, and S. T. Smith, "The Geometry of Algorithms with Orthogonality Constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, 1 1998.

[21] J. C. Gower, J. C. Gower, G. B. Dijksterhuis, *et al.*, *Procrustes problems*, vol. 30. Oxford University Press on Demand, 2004.

[22] N. A. Derzko and A. M. Pfeffer, "Bounds for the spectral radius of a matrix," *Mathematics of Computation*, vol. 19, no. 89, pp. 62–67, 1965.

[23] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

A Proof for Theorem 1

In the classical back-propagation equation, for a cost function $\mathcal{E}(.)$ and the Jacobian $\boldsymbol{J}$ of $\boldsymbol{x}_{l+1}$ with respect to $\boldsymbol{x}_l$, applying chain rule, following is true:

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_l} &= \boldsymbol{J} \frac{\partial \mathcal{E}}{\partial \boldsymbol{x}_{l+1}}, \\ \boldsymbol{J} &= \frac{\partial \boldsymbol{x}_{l+1}}{\partial \boldsymbol{x}_l} = \boldsymbol{I} + \boldsymbol{W}_l^T, \end{aligned} \tag{4}$$

To prove Theorem 1, we first state two lemmas.

Lemma 1. *For any non-singular matrix $\boldsymbol{I} + \boldsymbol{M}$, we have:*

$$1 - \sigma_{max}(\boldsymbol{M}) \leq \sigma_{min}(\boldsymbol{I} + \boldsymbol{M}) \leq \sigma_{max}(\boldsymbol{I} + \boldsymbol{M}) \leq 1 + \sigma_{max}(\boldsymbol{M}),$$

where $\sigma_{min}(\boldsymbol{M})$ and $\sigma_{min}(\boldsymbol{M})$ represent the minimum and maximum singular values of $\boldsymbol{M}$, respectively.

Proof. Since $\sigma_{min}(\boldsymbol{I} + \boldsymbol{M}) > 0$, the lower bound is trivial for $\sigma_{max}(\boldsymbol{M}) \geq 1$. For $\sigma_{max}(\boldsymbol{M}) < 1$, it is known that $|\lambda_{max}(\boldsymbol{M})| < 1$, where $\lambda_{max}(\boldsymbol{M})$ is the maximum eigenvalue of $\boldsymbol{M}$ [22]. Thus, we can show that:

$$\begin{aligned}\sigma_{min}(\boldsymbol{I}+\boldsymbol{M}) &= (\sigma_{max}((\boldsymbol{I}+\boldsymbol{M})^{-1}))^{-1} = \|(\boldsymbol{I}+\boldsymbol{M})^{-1}\|_2^{-1} \\ &\overset{(a)}{=} \|\sum_{k=0}^{\infty}(-1)^k\boldsymbol{M}^k\|_2^{-1} \geq (\sum_{k=0}^{\infty}\|(-1)^k\boldsymbol{M}^k\|_2)^{-1} \\ &\geq (\sum_{k=0}^{\infty}\|\boldsymbol{M}\|_2^k)^{-1} = (\frac{1}{1-\|\boldsymbol{M}\|_2})^{-1} = 1-\sigma_{max}(\boldsymbol{M}).\end{aligned}$$

Identity (a) is known as Neuman series of a matrix, which holds when $|\lambda_{max}(\boldsymbol{M})| < 1$ and $||.||_2$ represents the l_2-norm of a matrix.

The upper bound is easier to show. Due to triangle inequality:

$$\sigma_{max}(\boldsymbol{I}+\boldsymbol{M}) = \|\boldsymbol{I}+\boldsymbol{M}\|_2 \leq \|\boldsymbol{I}\|_2 + \|\boldsymbol{M}\|_2 = 1+\sigma_{max}(\boldsymbol{M}).$$

□

Lemma 2. *(Theorem 2.1 in [10]) Suppose $L \geq 3\gamma$. Then, there exist a global optimum for $\mathcal{E}(\mathcal{W})$, such that we have*

$$\sigma_{max}(\boldsymbol{W}_l) \leq \frac{2(\sqrt{\pi}+\sqrt{3\gamma})^2}{L}, \forall l = 1, 2, \ldots, L,$$

where γ is $\max(|\log\sigma_{max}(\boldsymbol{R})|, |\log\sigma_{min}(\boldsymbol{R})|)$.

To prove the theorem, using Lemma 1 and knowing that

$$\sigma_{min}(\boldsymbol{J})\|\frac{\partial\mathcal{E}}{\partial\boldsymbol{x}_{l+1}}\|_2 \leq \|\boldsymbol{J}\frac{\partial\mathcal{E}}{\partial\boldsymbol{x}_{l+1}}\|_2 \leq \sigma_{max}(\boldsymbol{J})\|\frac{\partial\mathcal{E}}{\partial\boldsymbol{x}_{l+1}}\|_2,$$

we conclude that

$$(1-\sigma_{max}(\boldsymbol{W}_l))\|\frac{\partial\mathcal{E}}{\partial\boldsymbol{x}_{l+1}}\|_2 \leq \|\frac{\partial\mathcal{E}}{\partial\boldsymbol{x}_l}\|_2 \leq (1+\sigma_{max}(\boldsymbol{W}_l))\|\frac{\partial\mathcal{E}}{\partial\boldsymbol{x}_{l+1}}\|_2.$$

Using the results from Lemma 2, Theorem 1 follows immediately.

B Additional Experiments

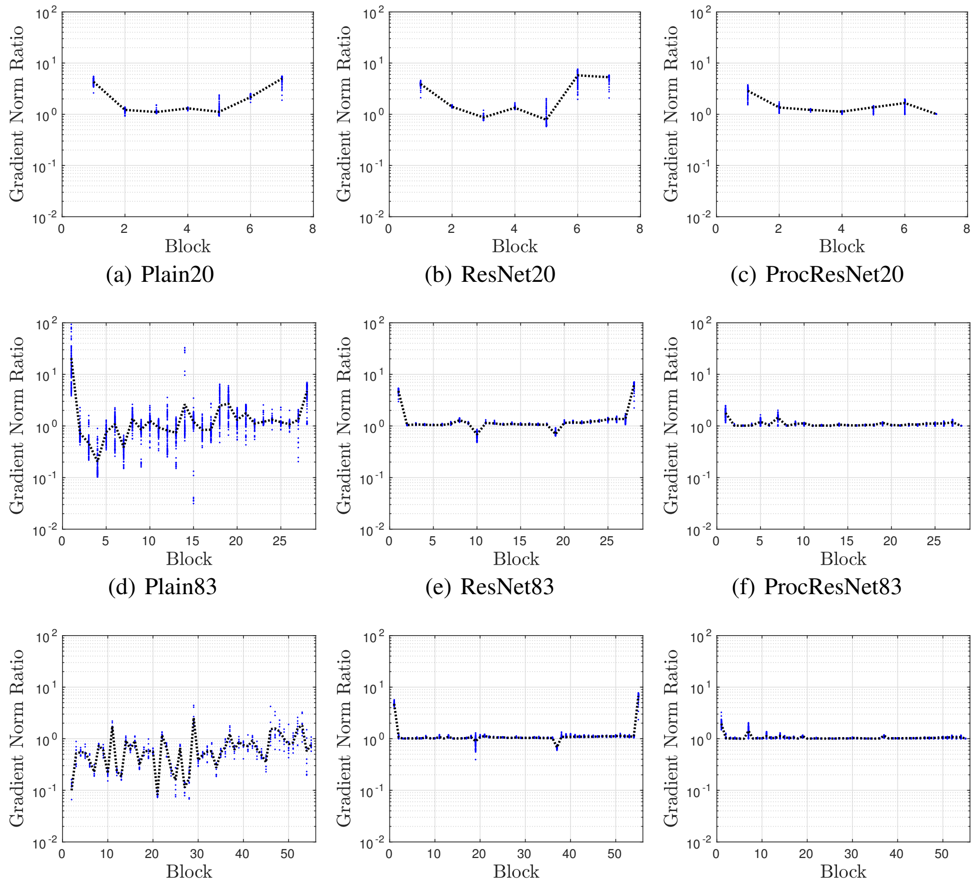
Figure 5 depicts the gradient norm ratio of different blocks for different network architectures for 100 epochs[2]. Each dot represents the ratio for a single dot and the dashed line is the mean value of the ratios over all 100 epochs. For a network of depth $3L+2$, its corresponding plot shows $L+1$ gradient norm ratios (representing L residual blocks[3] and 1 fully connected layer).

For instance, in Figure 5(e), which shows the gradient norm ratios of ResNet83 ($L = 27$), the first 27 blocks are corresponding to the residual blocks, transition and non-transition, and the last block, i.e. block #28, represents the fully-connected layer. It is also easy to notice that the blocks #1, #10, and #19, which represent the transition blocks, are not as norm-preserving as the blocks with identity skip connection. This is in contrast with Plain architecture, which does not exhibit any pattern. Moreover, the gradient ratios of the blocks in plain architecture varies significantly for different epochs. This experiment shows how the skip connections in ResNet architecture improve the norm-preservation of the network and emphasizes the importance of identity mapping as skip connection from norm-preservation perspective.

On the other hand, comparing ResNet with ProcResNet, it is evident that ProcResNet is more norm preserving in general, leading to optimization stability and performance gain, as discussed in more details in Section 4 of the main manuscript. One interesting observation is that, in ProcResNet20 architecture (Figure 2(c)), although only the transition blocks and fully connected layer are modified, all the blocks have become more norm-preserving, compared to ResNet20 (Figure 2(b). This

[2]Plain164 architecture became numerically unstable and training is stopped after 10 epochs.

[3]In Plain architecture, which does not have skip connections, the gradient norm ratio is obtained at the input and output of its building blocks as depicted in Figure 1(d).



(a) Plain20 (b) ResNet20 (c) ProcResNet20

(d) Plain83 (e) ResNet83 (f) ProcResNet83

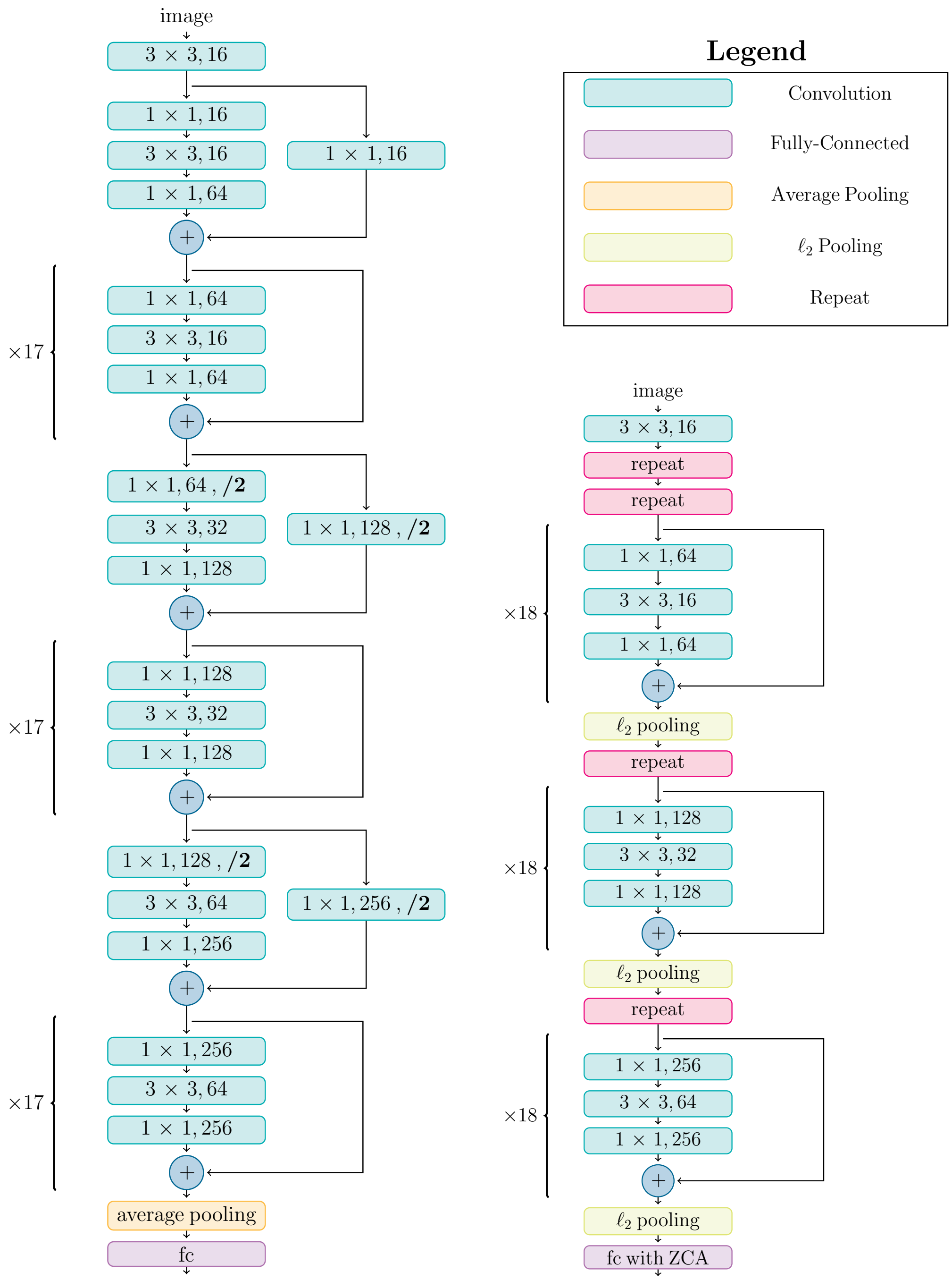
(g) Plain164 (h) ResNet164 (i) ProcResNet164

Figure 5: Gradient norm ratio for L residual blocks and fully-connected layer (last block). Each dot shows the ratio for a single epoch and dashed lines represent the mean of ratios over all epochs.

phenomena is existent, to a lesser extent, in deeper architectures as well. This highlights the importance of the proposed modifications and exploitation of norm-preserving blocks as a design principle in general. This observation also implies that the norm-preservation of different blocks are not independent of each other and should be analyzed accordingly, which is an interesting future direction.

C Implementation Details

As mentioned in Section 4.3 of the main manuscript, the classification accuracy of ProcResNet is obtained by changing the code of standard ResNet provided by its authors at `https://github.com/KaimingHe/resnet-1k-layers`. Furthermore, the results in Figure 4, which shows the instability of the standard ResNet, is generated by using the same code. In all of these experiments, the initial learning rate is 0.1. In ProcResNet, we take advantage of the added stability of the network and divide the learning rate at epochs 150 and 225. Also, the batch size is set to 64 for networks with 1001 layers and to 128 for other networks in all the experiments. The only exception is the classification error of standard ResNet1001 on CIFAR100, which is reported by its authors only for batch size of 128 [6]. We ran the same experiment for batch size of 64 and the classification accuracy of our single-run was 24.01%, which is larger than the results used for comparison in Table 1. In the other experiments provided in Section 4.1, Section 4.2, and Section B, which analyze the gradient signal on CIFAR10 data set, an implementation of the network in `chainer` framework [23] is used.



(a) ResNet164 Architecture

(b) ProcResNet164 Architecture

Figure 6: Example network architectures for CIFAR dataset.

Furthermore, Figure 6 illustrates the network architectures for ResNet164 and ProcResNet164. The numbers inside each convolution block represent kernel size, number of channels, and stride of the convolution. For instance, a 3×3 convolution layer with 16 channels and stride of 2 is denoted by $3 \times 3, 16, /2$. Batch normalization and ReLU blocks are not shown in the diagrams. We refer the reader to [6] for further details on the structure of the network and the placement of batch normalization and ReLU blocks. In ProcResNet, each pooling operation, either it is performed by an average pooling or by a convolution with stride of 2, is replaced by an ℓ_2 pooling. Furthermore, the each repeat block doubles the number of channels to match the dimension of input and output. Finally, to make the fully connected layer norm-preserving, the ZCA projection is performed after each iteration, as described in Section 3 of the main manuscript.