The Effect of Network Depth on the Optimization Landscape

Behrooz Ghorbani
Department of Electrical Engineering
Stanford University
ghorbani@stanford.edu

Shankar Krishnan
Google Research
skrishnan@google.com

Ying Xiao
Google Research
yingxiao@google.com

May 2019

Abstract

It is well-known that deeper neural networks are harder to train than shallower ones. In this short paper, we use the (full) eigenvalue spectrum of the Hessian to explore how the loss landscape changes as the network gets deeper, and as residual connections are added to the architecture. Computing a series of quantitative measures on the Hessian spectrum, we show that the Hessian eigenvalue distribution in deeper networks has substantially heavier tails (equivalently, more outlier eigenvalues), which makes the network harder to optimize with first-order methods. We show that adding residual connections mitigates this effect substantially, suggesting a mechanism by which residual connections improve training.

1 Introduction

Practical experience in deep learning suggests that the increased capacity that comes with deeper models can significantly improve their predictive performance. It has also been observed that as the network becomes deeper, training becomes harder. In convolutional neural networks (CNNs), residual connections [6] are used to alleviate this problem. Various explanations are provided for this phenomenon: [7] suggests that residual connections reduce the flatness of the landscape, whereas [4] questions this premise, noting that the extremal eigenvalues of the loss Hessian are much larger when residual connections are present: large Hessian eigenvalues indicate that the curvature of the loss is much sharper, and less flat. In a different line of work, [1] observes that the gradients with respect to inputs in deeper networks decorrelate with depth, and suggest that residual connections reduce the 'shattering' of the gradients.

In this paper, we explore the interaction between depth and the loss geometry. We first establish that gradient explosion or vanishing is not responsible for the slowing down of training, as is commonly believed. Searching for an alternative explanation, we study the Hessian eigenvalue density (using the tools introduced in [4] to obtain estimates of the eigenvalue histogram or *density*). The classical theory of strongly convex optimization tells us that optimization is slow when the spectrum simultaneously contains very small and very large eigenvalues (i.e., optimization rate is dependent on $\kappa = \lambda_{\max}/\lambda_{\min}$). Following this intuition, we focus on examining the relative spread of the Hessian eigenvalues. In particular, we quantify the extent of the large outliers by computing some scale-invariant classical statistics of the Hessian eigenvalues, namely the skewness and kurtosis. Finally, we observe that in comparable models with residual connections, these magnitude of these outliers is substantially mitigated. In [4], it is hypothesised that batch normalization suppresses large outlier eigenvalues, thereby speeding up training; in this paper, we present evidence that residual connections speed up training through essentially the same channel.

Throughout, the dataset of interest is CIFAR-10; we describe the specific model architectures used in Appendix A.

2 Gradient Explosion: An Incomplete Reason For Poor Training

It is well-known that deeper CNNs are harder to train than shallower ones. We exhibit training loss curves depicting this for both residual and non-residual (we refer to these as *simple*) CNNs in Appendix B, at

various network depths (20 and 80). The most prevalent explanation for why very deep networks are hard to train is that the gradient explodes or vanishes as the number of layers increase [5]; this explanation has been infrequently challenged (Section 4.1 in [6]), but no definitive experiments have been shown. We study this hypothesis in Figure 1, where we compare the gradient norms of a depth 80 residual and non-residual networks.

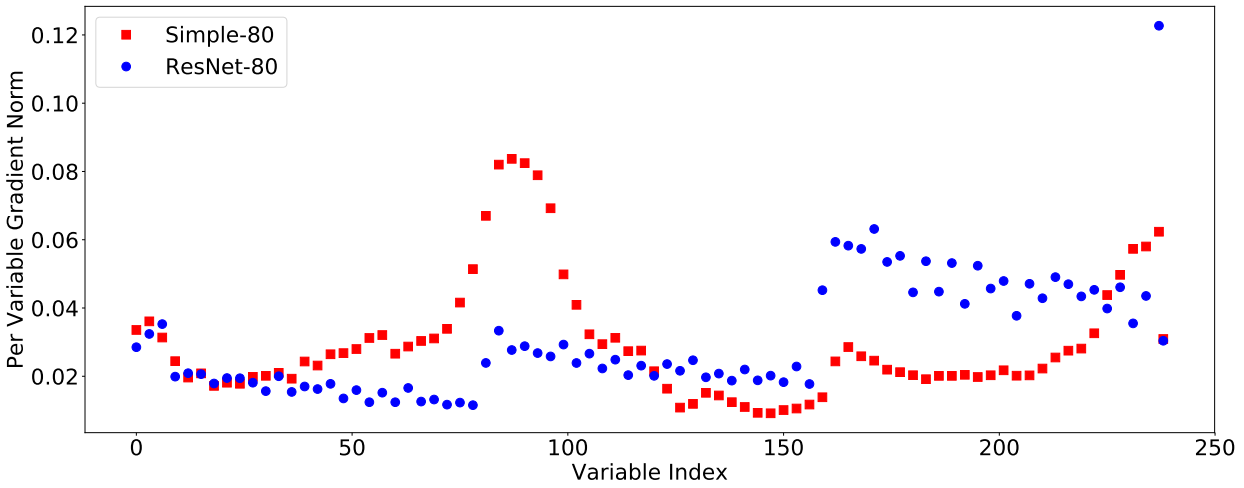

Figure 1: The norm of the gradients per variable for a ResNet and a simple CNN. The gradients are evaluated after $10k$ steps of training on the full dataset. The plot suggests that the two networks have similar gradient norm characteristics.

Two things become clear from this this plot. Firstly, there is no exponential increase or decrease in gradient norms (i.e., we would see vastly different gradient norm scales), as hypothesised in gradient explosion explanations. Secondly, residual connections do not consistently increase or decrease the gradient norms. In Figure 1, 49.4% of variables have lower gradient norm in residual networks (in comparison to a baseline of non-residual networks), making the exploding/vanishing gradient explanation untenable in this case.

3 Deeper Models Have Heavier Eigenvalue Tails

Let $H \in \mathbb{R}^{n \times n}$ be the Hessian of the training loss function with respect to the parameters of the model:

$$H_{i,j} = \frac{\partial^2 L(\theta)}{\partial \theta_i \partial \theta_j} \tag{1}$$

where $\theta \in \mathbb{R}^n$ is the parameter vector, and $L(\theta)$ is the training loss. The Hessian is a measure of (local) loss curvature. In the convex setting, optimization characteristics are largely determined by the loss curvature, we expect to be able to observe the factors slowing down the training from analyzing the Hessian spectrum along the optimization trajectory.

Let $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n$ be the eigenvalues of the Hessian. The theory of convex optimization suggests that first-order methods such as SGD slow down dramatically when the *relative* differences among the eigenvalues of the loss Hessian are large; in particular, results from convex analysis suggest that as $|\ \frac{\lambda_i}{\lambda_1}\ |$ becomes smaller, the optimization in the direction of the eigenvectors associated with λ_i slows down [2, 3] [1]. Following this intuition, when the distribution of the eigenvalues of H has heavy tails (equivalently large outliers), we expect the network to train slowly as there will be many eigenvalues where λ_i/λ_1 is small.

Figure 2 shows the (smoothed) density of the eigenvalues of the Hessian for a series of simple CNNs with increasing depth. This figure shows two prominent features of the loss Hessian:

[1] While these results are only formally valid in convex cases, the intuition they convey is observed everyday in training deep networks.

1. Most of the eigenvalues of the Hessian are concentrated near zero. This means that the loss surface is relatively flat, in agreement with [9, 4] and others.

2. As the network gets deeper, outliers appear in the spectrum of H. Moreover, the magnitude of these outliers increases with the depth of the network. This means that as the network becomes deeper, $\frac{\lambda_i}{\lambda_1}$ shrinks for almost all of the directions, making the training challenging.

To quantify the magnitude and extent of these outlier eigenvalues, we compute some scale-independent classical statistics of the Hessian eigenvalues. We are primarily interested in *skewness* and *kurtosis* defined as:

$$\text{skewness}(X) = \mathbb{E}[(X-\mu)^3]/\sigma^3 \qquad \text{kurtosis}(X) = \mathbb{E}[(X-\mu)^4]/\sigma^4$$

The skewness of a distribution measures its asymmetry, and the kurtosis measures how heavy (or non-Gaussian) the tails are – a heavy tailed distribution has a kurtosis greater than 3. In our case, we compute these statistics on the Hessian eigenvalues by observing that for $v \sim N(0, I_n)$:

$$\mathbb{E}[v^T H^k v] = \sum_{i=1}^{n} \lambda_i^k.$$

Due to the rapid concentration of the quadratic form in high dimensions (for concrete bounds, see [8]) we expect extremely accurate approximation of $\mathbb{E}[\lambda^k]$ using a few i.i.d. samples of the form $v^T H^k v$. Both skewness and kurtosis should dramatically increase as the tails of the eigenvalue density become heavier.

Figure 3 shows what happens to these metrics as we increase the depth: both the skewness and kurtosis increase dramatically as we increase the depth of the model. In particular, note that the kurtosis is far from being a Gaussian – these distribution of eigenvalues is extremely heavy tailed.

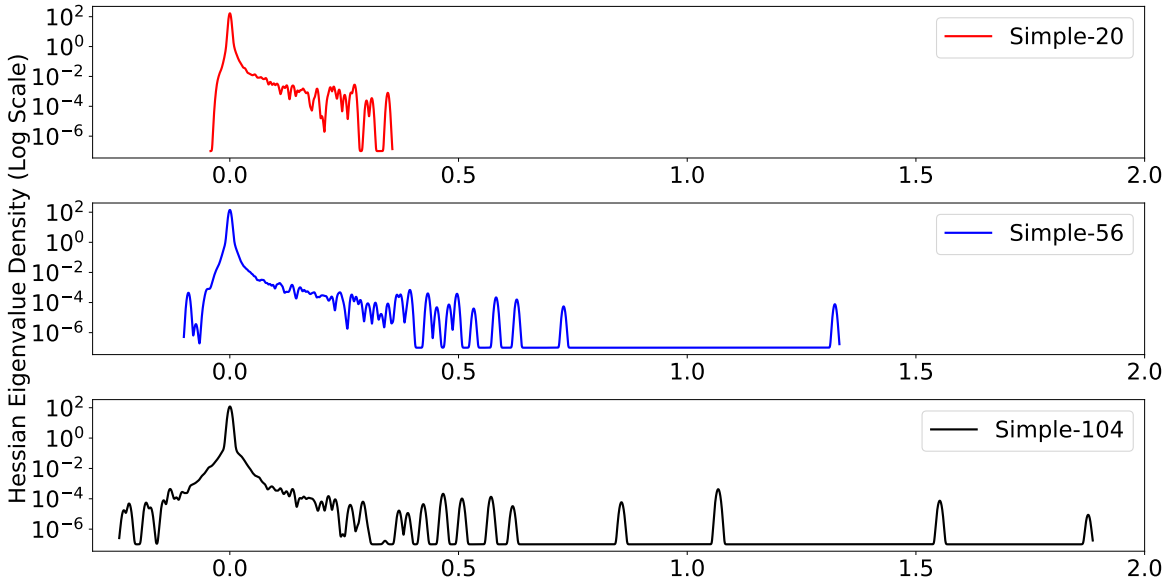

Figure 2: Density plots of the Hessian eigenvalues of models with increasing depth. All the spectra are normalized to have the same standard deviation. As the network becomes deeper, large outliers appear in the spectrum.

4 How Do Residual Connections Affect the Spectrum?

Given that residual connections allow us to train much deeper models, we would predict that the addition of residual connections should prevent the largest eigenvalues from being so extreme. Figure 4 compares

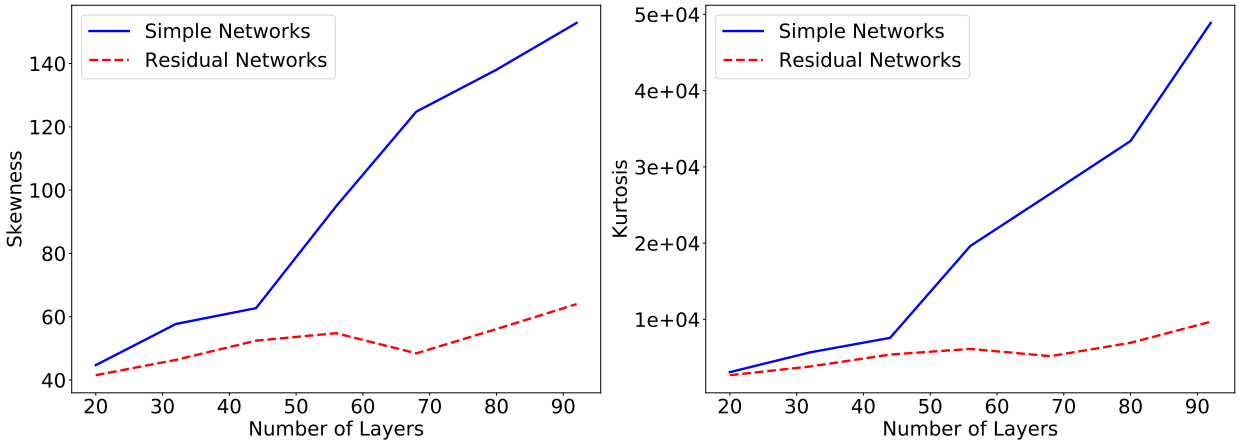

Figure 3: Skewness and kurtosis of networks by depth, and with residual connections. The Hessians are evaluated at $10k$ training steps.

the spectrum of the Hessian for residual networks and their corresponding simple networks (both networks are identical save for the residual connections). We can see that adding residual connections substantially reduces the extent of the outliers in the Hessian spectrum. More quantitatively, in Figure 3, we can see that models with residual connections have substantially lower skewness and kurtosis than models without residual connections. The effects are in substantial: a 90 layer model with residual connections has lower skewness and kurtosis than a non-residual model half its size.

5 Conclusion

In this paper, we have presented qualitative and quantitative evidence that depth increases outlier eigenvalues in the Hessian, and that residual connections mitigate this. We believe that this touches upon some of the fundamental dynamics of optimizing neural networks, and that any theoretical explanation of residual connections needs to explain this.

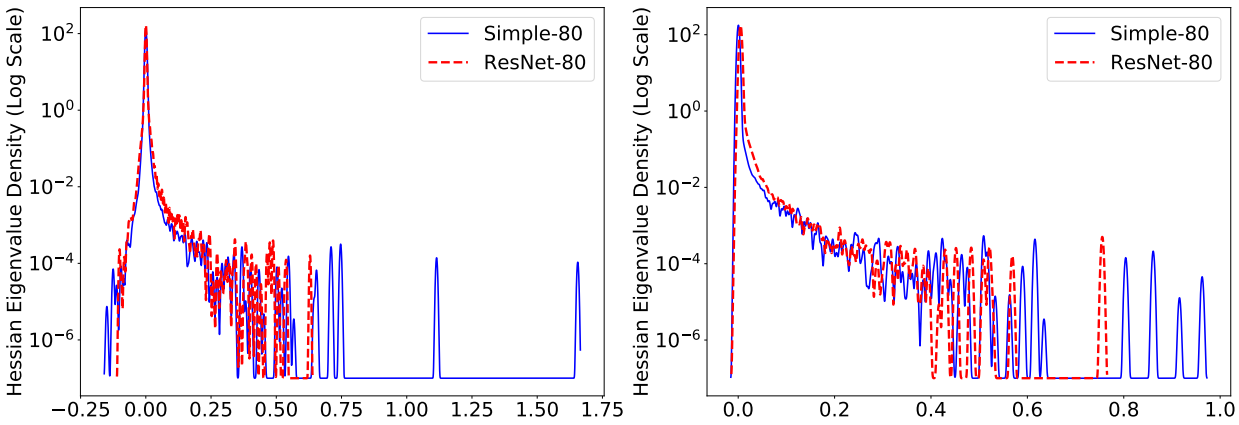

Figure 4: Comparison between the normalized Hessian spectrum of a ResNet-80 and a corresponding simple network after roughly $10k$ (left) and $80k$ (right) training steps. The outliers are significantly dampened in the residual network.

6 Acknowledgements

Behrooz Ghorbani was supported by grants NSF-DMS 1418362 and NSF-DMS 1407813.

References

[1] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian 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-Volume 70*, pages 342–350. JMLR. org, 2017.

[2] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[3] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[4] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. *arXiv preprint arXiv:1901.10159*, 2019.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] A Emin Orhan and Xaq Pitkow. Skip connections eliminate singularities. *arXiv preprint arXiv:1701.09175*, 2017.

[8] Mark Rudelson, Roman Vershynin, et al. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18, 2013.

[9] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

A Network Architectures

For the purposes of this short exposition, we adopt a class of networks for our study. We consider a standard residual networks trained on CIFAR-10. These types of networks have $6n$ layers of feature maps of sizes $\{32 \times 32, 16 \times 16, 8 \times 8\}$ ($2n$ layers for each type) with $\{16, 32, 64\}$ filters per layer. With the addition of the input convolution layer and the final fully-connected layer, this type of network has $6n + 2$ layers. Batch-Normalization is also present in these networks. In our experiments, when we don't include residual connections, we refer to the network 'simple-$6n + 2$' network and when residual connections are included, the network is referred to as ResNet-$6n + 2$. We use the SGD with momentum with the same learning rate schedule to train both these networks for $100k$ steps.

B Deeper CNNs are harder to train; skip connections help more at depth

We observe that for small n, both simple and ResNet networks train well (Figure 5). As the number of layers increase, training simple model becomes slower. Note however that as we increase the depth, that residual

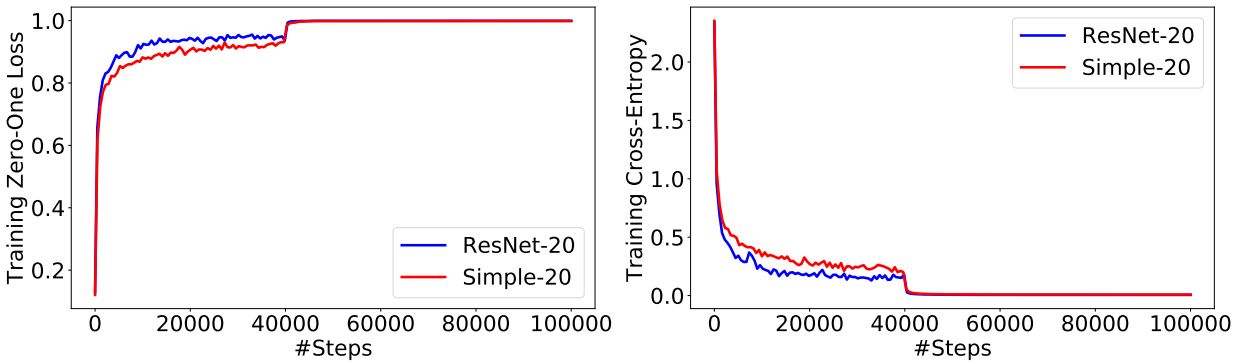

Figure 5: Training curves for a short stack of convolutional layers corresponding to $n = 3$ in Section 1.

connections improve the training loss substantially.

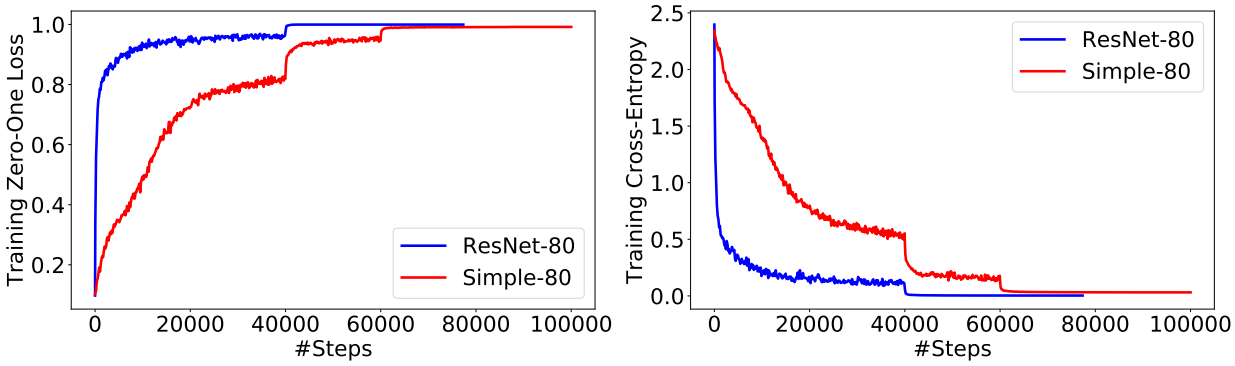

Figure 6: Training curves for a tall stack of convolutional layers corresponding to $n = 13$ in Section 1.