Exploring Gradient Oscillation in Deep Neural Network Training

Chedi Morchdi[1], Yi Zhou[1], Jie Ding[2], Bei Wang[3]

Abstract— **Understanding optimization in deep learning is a fundamental problem, and recent findings have challenged the previously held belief that gradient descent stably trains deep networks. In this study, we delve deeper into the instability of gradient descent during the training of deep networks. By employing gradient descent to train various modern deep networks, we provide empirical evidence demonstrating that a significant portion of the optimization progress occurs through the utilization of oscillating gradients. These gradients exhibit a high negative correlation between adjacent iterations. Furthermore, we make the following noteworthy observations about these gradient oscillations (GO): (i) GO manifests in different training stages for networks with diverse architectures; (ii) when using a large learning rate, GO consistently emerges across all layers of the networks; and (iii) when employing a small learning rate, GO is more prominent in the input layers compared to the output layers. These discoveries indicate that GO is an inherent characteristic of training different types of neural networks and may serve as a source of inspiration for the development of novel optimizer designs.**

I. Introduction

In modern machine learning, there is a remarkable and consistent observation that simple gradient-based algorithms have proven to be highly effective in training deep neural networks [1]–[3]. This observation has sparked significant interest among researchers, leading them to investigate the fundamental mechanisms that contribute to the success of nonconvex neural network optimization. As a result, numerous studies have been conducted, each providing distinct explanations for this phenomenon.

Specifically, several theoretical works have provided explanations for the success of gradient-based deep learning optimization by focusing on specific types of amenable geometry found in deep networks. These include gradient dominant geometry [4]–[9] and local strong convexity [6], [10]–[13]. These theories have shed light on the favorable properties of deep network optimization. However, recent empirical observations have revealed a contrasting reality. It has been observed that gradient-based neural network optimization frequently operates in what is known as the "edge of stability" regime [14]–[16], in which the optimization tends to be 'unstable' [14]–[16]. More specifically, by training many deep networks using full-batch gradient descent, it is discovered that the maximum eigenvalue of the training loss Hessian hovers just above $2/(\text{step size})$, which is known to cause instability in training simple convex quadratic models. This surprising observation seems to be highly inconsistent with the aforementioned smooth geometries that are leveraged in the conventional optimization analysis. Consequently, there is a strong motivation to delve deeper into the dynamics of gradient-based algorithms in neural network training.

The research presented in [14] highlights that the maximum eigenvalue of the training loss Hessian remains above $2/(\text{step size})$, a value known to induce instability when optimizing convex quadratic functions. However, it's worth noting that the maximum eigenvalue of the Hessian differs significantly from the first-order gradients utilized in actual optimization updates. Consequently, it remains unclear how the actual gradients behave in the practical training of *nonconvex* deep networks. Additionally, while [14] focuses on the instability of gradient descent across the entire parameter space, modern neural network models are composed of multiple feed-forward layers with diverse architectures. Therefore, it is crucial to investigate how this instability propagates throughout the deep network layers during the optimization process. In light of these considerations, the objective of this study is to delve into the characteristics of gradient oscillation (GO) observed in the training of modern deep networks and address the aforementioned questions. By exploring GO, we aim to gain insights into the behavior of actual gradients and how the instability manifests and evolves across different layers of deep networks during the optimization process.

A. Our Contribution

We apply the standard gradient descent algorithm to train several popular deep networks using various datasets and study the gradients generated along the optimization trajectory using the gradient correlation metric defined in Section II. We consistently make the following observations.

- We first run gradient descent with a large learning rate to train various convolutional and residual networks, and consistently observe two distinct gradient correlation patterns: (i) gradients computed in adjacent iterations are highly positively correlated (in terms of the cosine similarity defined in eq. (2)); and (ii) gradients computed in adjacent iterations are highly negatively correlated, i.e., they suffer from intensive oscillation. Surprisingly, most of the training loss decrease is attained with oscillating gradients. Interestingly, we find that gradient oscillation (GO) occurs in the later phase of training for convolutional type networks, whereas it occurs in the initial phase of training for residual type networks.

  We further track the layer-wise gradient correlation (i.e., correlation of layer-wise gradients) throughout the training process, and observe the same correlation patterns as described above in all the layers of the networks.

[1] Department of ECE, University of Utah, UT, USA. {u1402320, yi.zhou}@utah.edu.

[2] School of Statistics, University of Minnesota, MN, USA. dingj@umn.edu.

[3] School of Computing, University of Utah, UT, USA. beiwang@sci.utah.edu.

- We then run gradient descent with a small learning rate and observe similar gradient correlation patterns for different networks as described above. However, in this case, the layer-wise gradient correlations have different patterns across the layers. Specifically, we consistently observe high GO in the input layers, whereas gradients of the output layers tend to be more stable and positively correlated. This shows that the vanilla gradients generated by gradient descent have highly imbalanced layer-wise gradient correlation across the layers of deep models.

Our discoveries suggest that GO is an essential and invariant feature in the optimization of different types of deep neural networks, and may inspire new optimizer designs.

B. Related Work

Many nonconvex optimization theories have been developed to explain the success of deep learning optimization. The key idea is to prove that deep neural networks have certain nice geometries that guarantee convergence to the global minimum in nonconvex optimization. For example, many types of deep neural networks such as over-parameterized residual networks [4]–[6], recurrent networks [7], nonlinear networks [8], [9], and linear networks [17], [18] have been shown to satisfy the so-called gradient dominant geometry [19]. On the other hand, shallow ReLU networks [10]–[13], deep residual networks [6], and some nonlinear networks [20], [21] have been shown to satisfy the local strong convexity geometry. Both geometry types guarantee the convergence of gradient-based algorithms to a global minimum at a linear rate. Moreover, some other works proved that no spurious local minimum exists for various nonlinear networks [22], [23] and linear networks [24], [25].

On the other hand, from an empirical perspective, researchers have found that skip connections and batch normalization of deep networks can substantially improve the smoothness of the optimization geometry [26]–[28]. Furthermore, some other works found that there is a continuous low-loss path between the minima of deep networks [29], [30]. In particular, it is observed that a simple linear interpolation between the initialization point and global optimum encounters no significant barrier for many deep networks [31]. Moreover, many networks have been shown to possess wide and flat minima that tend to generalize well [32], [33].

II. Preliminaries

A. Definition of Gradient Correlation

To understand the optimization behavior of gradient descent (GD) in deep learning, we investigate the gradients generated along the optimization trajectory of GD. Specifically, given a set of training samples $\{x_i, y_i\}_{i=1}^n$ where x_i denotes the data and y_i denotes the corresponding label, the training objective function and the GD update at each step ($k = 0, 1, \ldots$) are written as follows.

$$\text{(Objective function): } \mathcal{L}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(h_\theta(x_i), y_i),$$
$$\text{(GD): } \theta_{k+1} = \theta_k - \eta \nabla \mathcal{L}_n(\theta_k), \qquad (1)$$

where h_θ denotes the neural network model parameterized by θ, ∇ is the gradient operator with respect to the parameter θ, η is the learning rate, and ℓ is the loss function. We focus on classification tasks with cross-entropy loss in this paper. In the training, we collect a set of gradients generated along the optimization trajectory of GD, i.e., $\{\nabla \mathcal{L}_n(\theta_0), \nabla \mathcal{L}_n(\theta_1), \ldots, \nabla \mathcal{L}_n(\theta_k), \ldots\}$. These gradients determine the direction of model updates. To quantify the gradients' statistical behavior, we consider the following gradient correlation between an arbitrary pair of gradients $(\nabla \mathcal{L}_n(\theta_j), \nabla \mathcal{L}_n(\theta_k))$.

$$\text{(Gradient correlation): } \mu(j,k) := \frac{\langle \nabla \mathcal{L}_n(\theta_j), \nabla \mathcal{L}_n(\theta_k)\rangle}{\|\nabla \mathcal{L}_n(\theta_j)\| \cdot \|\nabla \mathcal{L}_n(\theta_k)\|} \in [-1, 1], \quad \forall\ j, k, \qquad (2)$$

where $\|\cdot\|$ denotes the ℓ_2-norm. To help visualize the gradient correlation, in all of our experiments, we fix a window size $h = 5$ and track the mean gradient correlation over all pairs of *adjacent iterations* within the window. The choice of h is not essential as it only affects the smoothness of the mean gradient correlation curves.

B. Gradient Correlation in Convex Optimization

We illustrate gradient correlation in training a simple convex linear classifier using gradient descent with learning rate $\eta = 0.01$ on the MNIST dataset [34]. In Figure 1, we plot the training loss and the corresponding mean gradient correlation over 2500 epochs. It can be seen that the mean gradient correlation is very close to $+1$ throughout the training, indicating that the gradients calculated between adjacent iterations are highly positively correlated and the training is stable. This is due to the well-conditioned convex geometry and the use of a small learning rate. In the following sections, however, we show that optimization in deep learning is usually driven by oscillating gradients.

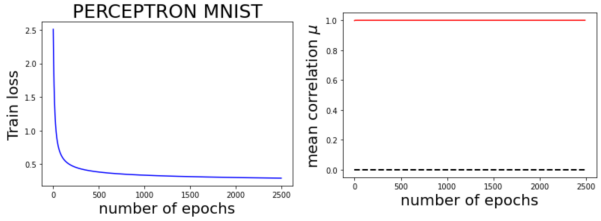

Fig. 1: Training perceptron using GD on MNIST.

III. Gradient Oscillation of GD in Neural Network Training

In this section, we explore the correlation of the gradients (defined in Section II) generated by GD in training various types of modern deep networks using a large learning rate.

A. Training Convolutional Networks

Simple CNN. We train a simple feed-forward CNN that consists of three convolution blocks and one fully-connected block on the CIFAR-10 dataset [35] using GD. We set the learning rate $\eta = 0.1$ and train it for 1500 epochs. We track

the training loss and mean gradient correlation throughout the training process, and the results are shown in Figure 2. It can be seen that the training loss decreases to almost zero after 800 epochs.

Interestingly, the mean gradient correlation changes drastically in different stages of training. Specifically, in the first 10 epochs at the very beginning of training, the gradient correlation between adjacent iterations is close to $+1$. This shows that the gradients computed in the initial epochs are very stable and highly positively correlated (i.e., they point toward the same direction), as can be seen from the pairwise gradient correlation matrix computed in epoch 5 shown in the second row. However, after 10 epochs, the gradient correlation drops to highly negative values close to -1, which implies that the gradients become unstable and start to oscillate. This phenomenon is further illustrated by the pairwise gradient correlation matrix computed in epoch 675 shown in the second row. Surprisingly, one can observe that most of the training loss decrease is achieved with oscillating gradients between 10 and 800 epochs.

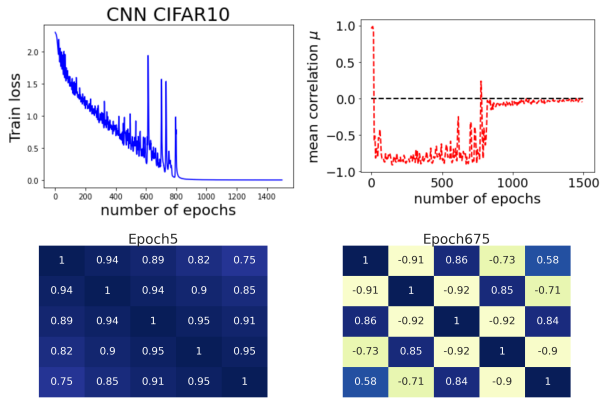

Fig. 2: CNN training using GD on CIFAR-10. From left to right: training loss curve, mean gradient correlation curve, pairwise correlations of gradients within a window of $h = 5$ consecutive iterations at epochs 5 and 675, respectively.

VGG-16. We further explore the gradient correlation in training a VGG-16 network on the CIFAR-10 dataset using GD with learning rate $\eta = 0.1$ for 1500 epochs. The results are shown in Figure 3, where we can observe a similar phenomenon. Specifically, the gradient correlation is highly positive in most of the initial 500 epochs. This is further illustrated by the pairwise gradient correlation matrix computed in epoch 5 shown in the second row, whose entries are uniformly highly positive. This shows that the gradients computed in the initial epochs are stable and pointing toward almost the same direction. However, in the later training phase after 500 epochs, the gradient correlation drops to highly negative values, indicating an occurrence of gradient oscillation. This phenomenon is further illustrated by the pairwise gradient correlation matrix computed in epoch 915 shown in the second row. Again, one can observe that most of the training loss decrease is achieved with oscillating gradients after 500 epochs.

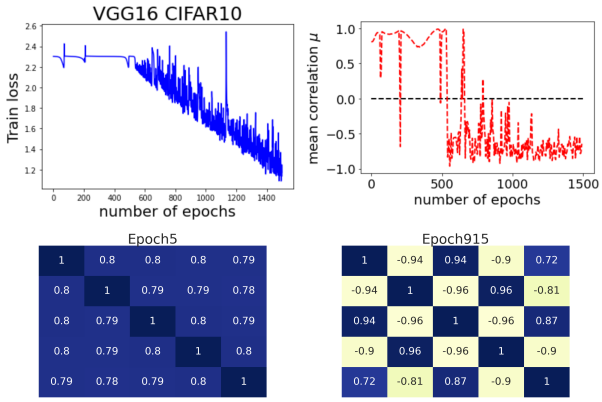

Fig. 3: VGG-16 training using GD on CIFAR-10.

Summary. We obtain additional results on training convolutional-type networks using other datasets including MNIST [34] and SVHN [36]. The observations are very similar and are not presented due to space limitation. From all these results, we observe a common phenomenon in training convolutional type networks using GD: most of the optimization progress is achieved with oscillating gradients, and moreover, the gradients generated along the optimization trajectory have a sharp transition from positive correlation to negative correlation. This is very different from the gradients in training convex models (see Section II-B), which are highly positive throughout the training.

B. Training Residual Networks

We train various residual networks using GD and track the gradient correlation. Interestingly, we also observe gradient oscillation in training residual networks, but the overall transition of gradient correlation is very different from that of convolutional networks.

ResNet-18. We train a ResNet-18 on CIFAR-10 using GD with learning rate $\eta = 0.1$ for 500 epochs, and track the training loss and gradient correlation. The results are shown in Figure 4, from which one can see a very different transition of gradient correlation compared to convolutional networks. Specifically, one can see that the gradient correlation is highly negative in the first 120 epochs. This shows that the gradients computed in adjacent iterations are negatively correlated in the initial training phase, indicating gradient oscillation. This phenomenon is further illustrated by the pairwise gradient correlation matrix computed in epoch 60. After 150 epochs, the gradient correlation increases to highly positive values, implying very stable and positively correlated gradients when approaching the minimizer. This is also illustrated by the pairwise gradient correlation matrix computed in epoch 490. To summarize, the gradients in training ResNet-18 have a sharp transition from negative correlation to positive correlation, which is opposite to that of convolutional networks. Moreover, most of the loss decrease is achieved with oscillating gradients.

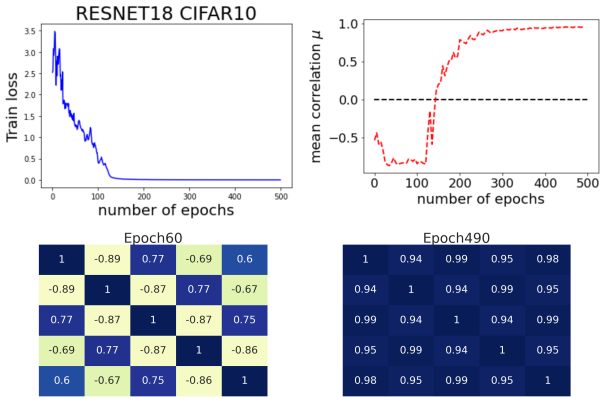

Fig. 4: ResNet-18 training using GD on CIFAR-10.

ResNet-34. We further train a ResNet-34 on CIFAR-10 using GD with $\eta = 0.1$ for 500 epochs. The results are shown in Figure 5, where one can observe a similar transition of gradient correlation to that of ResNet-18.

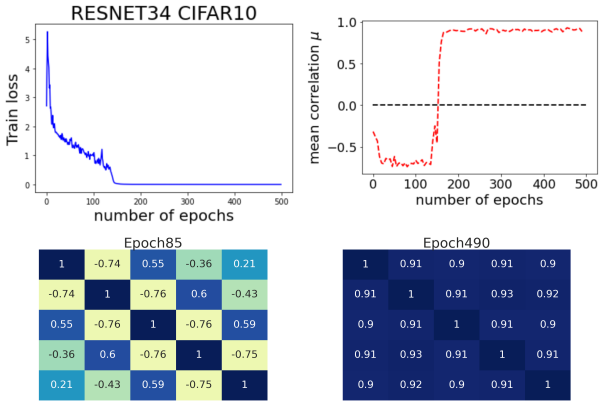

Fig. 5: ResNet-34 training using GD on CIFAR-10.

Summary. We obtain additional results on training residual networks using MNIST and SVHN. The observations are very similar and are not presented due to space limitation. From all these results, we observe a common phenomenon in training residual type networks using GD: the gradients generated along the optimization trajectory have a sharp transition from negative correlation to positive correlation. This is opposite to the transition observed in training convolutional networks.

C. Layer-wise Gradient Correlation

An important feature of deep networks is the composition of layers. In this subsection, we further investigate gradient correlation across different layers. Due to space limitation, we only present the results on a subset of the layers.

Simple CNN. The following Figure 6 tracks the layer-wise gradient correlation and gradient dot product (associated with the layer-wise gradient correlation) in training a CNN on CIFAR-10 using GD with learning rate $\eta = 0.1$. We only present the results on the first and last layers, and the intermediate layers have similar results. From the first row of the figure, it can be seen that gradient correlation behaves consistently across different layers, i.e., it is highly positive in the first 10 epochs and then drops to highly negative values later on. This implies that gradient oscillation occurs in all the layers. On the other hand, from the second row of the figure, we observe that the layer-wise gradient dot products are of similar numerical order. This shows that gradient oscillation occurs uniformly across the layers.

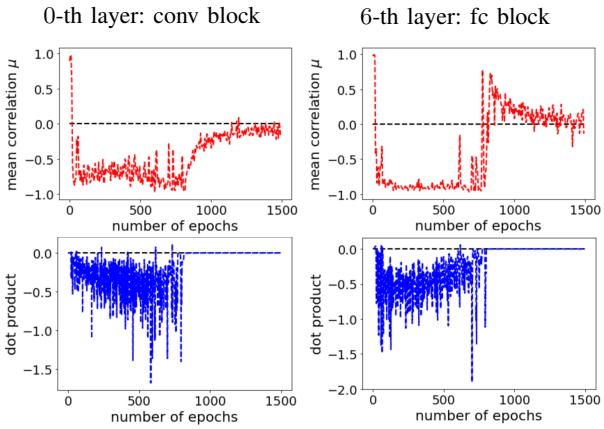

Fig. 6: Layer-wise gradient correlation and dot product in training CNN on CIFAR-10.

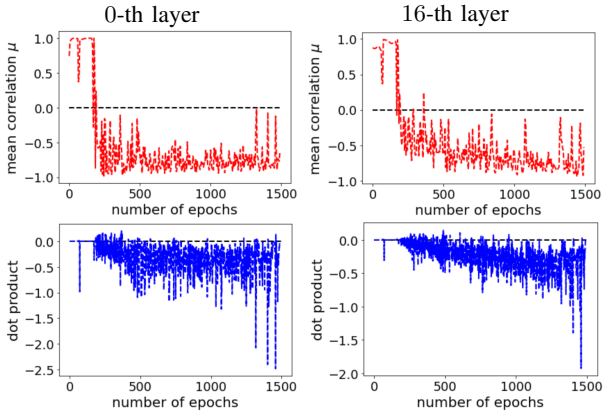

Fig. 7: Layer-wise gradient correlation and dot product in training VGG-16 on CIFAR-10.

VGG-16. Figure 7 shows the layer-wise gradient correlation and gradient dot product in training a VGG-16 on CIFAR-10 using GD with $\eta = 0.1$. One can make similar observations that gradient oscillation occurs uniformly across all the layers.

ResNet-18 & ResNet-34. Figures 8 and 9 show the layer-wise gradient correlation and gradient dot product in training a ResNet-18 and ResNet-34 on CIFAR-10 using GD with $\eta = 0.1$. One can see that the transition of the gradient correlation is highly consistent across all layers, and is opposite to that observed in training convolutional networks. Also, gradient oscillation occurs uniformly across all the layers.

Summary. From all these results on layer-wise gradient correlation, we consistently observe that gradient oscillation occurs uniformly across all the layers in training different

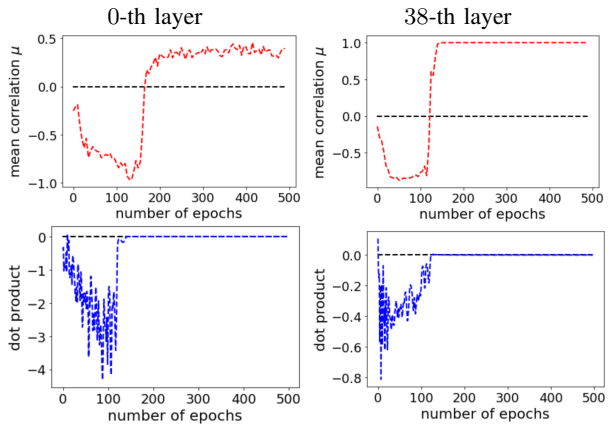

Fig. 8: Layer-wise gradient correlation and dot product in training ResNet-18 on CIFAR-10.

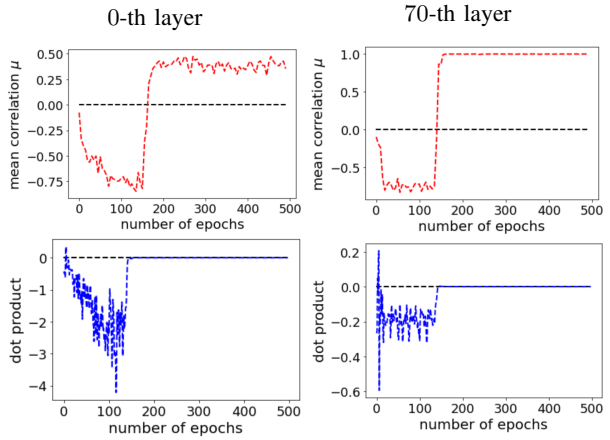

Fig. 9: Layer-wise gradient correlation and dot product in training ResNet-34 on CIFAR-10.

types of networks when a large learning rate is used.

IV. Gradient Oscillation under Small LR

All the previous experiments are conducted under a relatively large learning rate $\eta = 0.1$, which causes gradient oscillation in all the layers during the training. In this section, we explore the impact of learning rate on the transition of gradient correlation in the training.

Simple CNN. The following Figure 10 shows the training of a CNN on CIFAR-10 using GD with a small learning rate $\eta = 0.001$ for 30k epochs. From the second figure in the first row, it can be seen that the global gradient correlation follows a similar transition to that under the previous large learning rate, i.e., it is highly positive at the beginning and drops to highly negative values after about 8k epochs.

However, the figures in the second and third rows show that the layer-wise gradient correlations behave very differently from the global gradient correlation. Specifically, from the dot product figures in the third row, it can be seen that the dot product of the first convolutional block numerically dominates that of all the layers, because the scale of the inner product in this layer is substantially larger than other layers. This implies that the first convolutional block contributes the most to the global gradient correlation, and its layer-wise gradient correlation (first figure in the second row) is highly consistent with the global gradient correlation. On the other hand, the layer-wise gradient correlation of the last layer is substantially higher, implying that the gradient oscillation is less severe in the output layer.

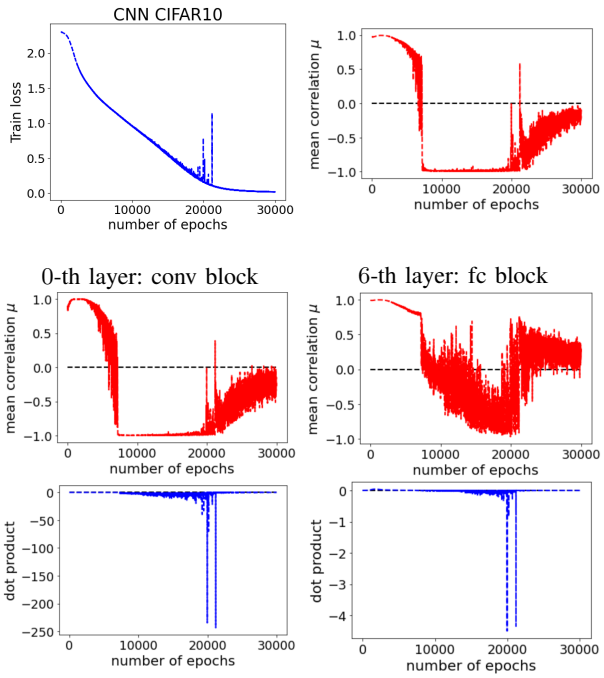

Fig. 10: CNN training using GD with $\eta = 0.001$ on CIFAR-10. The first row shows the training loss and the mean correlation of the global gradient. The second row shows the mean correlation of layer-wise gradients.

ResNet-18. Figure 11 shows the training of ResNet-18 on CIFAR-10 using GD with a small learning rate $\eta = 0.001$, from which one can make similar observations. Specifically, the first layer dominates the dot product among all the layers and hence contributes the most to the global gradient correlation. Consequently, its layer-wise gradient correlation is highly consistent with the global gradient correlation, and is highly negative in most of the epochs. As a comparison, the layer-wise gradient correlation of the last 38-th layer is close to $+1$, implying that the gradients of this layer are very stable in the training.

Summary. We obtain additional results on training ResNet-18 and ResNet-34 on CIFAR-10 (with small $\eta = 0.01$) and SVHN (with small $\eta = 0.01, 0.001$). The observations are very similar and are not presented due to space limitation. From all these results, we conclude that layer-wise gradient correlation can be diverse across different layers when the model is trained with small learning rates. In particular, the input layers are more likely to experience gradient oscillation, whereas the gradients of the subsequent layers are more and more stable. We think this is related to the composition structure of deep neural networks, which makes the parameters of the input layers suffer from a larger

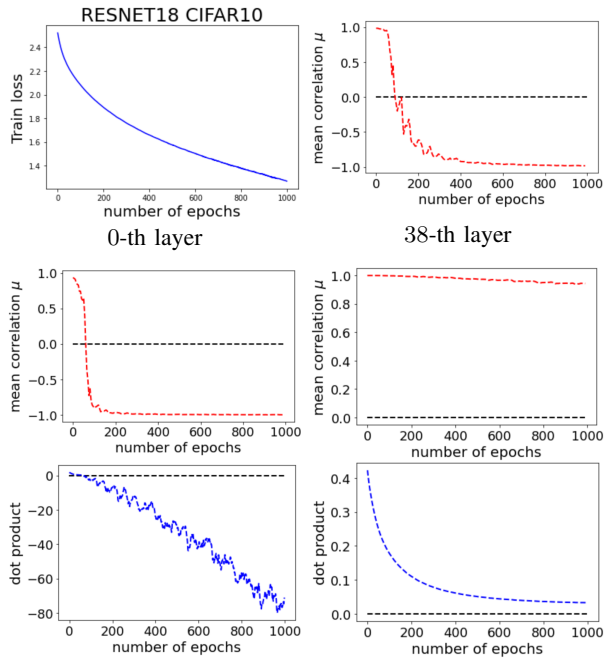

Fig. 11: ResNet-18 training using GD with η=0.001 on CIFAR-10.

Lipschitz constant that induces a smaller feasible learning rate.

V. Conclusion

In this paper, we reveal that gradient descent usually suffers from gradient oscillation in training modern deep networks. Such gradient oscillation follows diverse transition patterns depending on the learning rate, model architecture and different layers. As a comparison, the Adam adaptive optimizer applies normalized gradient updates to suppress the oscillation across different layers. These observations show that deep learning optimization cannot be fully characterized by the classic optimization theories, which crucially rely on simple and elegant geometrical assumptions. We hope to advance the understanding of neural network training in practice and inspire new optimizer designs.

References

[1] H. E. Robbins, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 2007.

[2] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., 2015.

[3] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, p. 2121–2159, jul 2011.

[4] H. Zhang, D. Yu, M. Yi, W. Chen, and T.-Y. Liu, "Convergence theory of learning over-parameterized resnet: A full characterization," *arXiv:1903.07120*, 2019.

[5] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. International Conference on Machine Learning*, vol. 97, 2019, pp. 242–252.

[6] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *Proc. International Conference on Machine Learning*, vol. 97, Jun 2019, pp. 1675–1685.

[7] Z. Allen-Zhu, Y. Li, and Z. Song, "On the convergence rate of training recurrent neural networks," in *Proc. International Conference on Neural Information Processing Systems*, 2019.

[8] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Gradient descent optimizes over-parameterized deep relu networks," *Machine Learning*, vol. 109, no. 3, pp. 467–492, 2020.

[9] Y. Zhou, H. Zhang, and Y. Liang, "Geometrical properties and accelerated gradient solvers of non-convex phase retrieval," *Annual Allerton Conference on Communication, Control, and Computing*, pp. 331–335, 2016.

[10] M. Soltanolkotabi, "Learning relus via gradient descent," in *Proc. Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon, "Recovery guarantees for one-hidden-layer neural networks," in *Proc. International Conference on Machine Learning*, vol. 70, Aug 2017, pp. 4140–4149.

[12] H. Fu, Y. Chi, and Y. Liang, "Local geometry of cross entropy loss in learning one-hidden-layer neural networks," in *Proc. IEEE International Symposium on Information Theory*, 2019, pp. 1972–1976.

[13] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," in *Proc. International Conference on Learning Representations*, 2019.

[14] J. Cohen, S. Kaur, Y. Li, J. Z. Kolter, and A. Talwalkar, "Gradient descent on neural networks typically occurs at the edge of stability," in *Proc. International Conference on Learning Representations*, 2021.

[15] C. Xing, D. Arpit, C. Tsirigotis, and Y. Bengio, "A walk with sgd," *arXiv:1802.08770*, 2018.

[16] L. Wu, C. Ma, and W. E, "How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective," in *Proc. Advances in Neural Information Processing Systems*, vol. 31, 2018.

[17] S. Frei and Q. Gu, "Proxy convexity: A unified framework for the analysis of neural networks trained by gradient descent," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 7937–7949.

[18] Y. Zhou and Y. Liang, "Characterization of gradient dominance and regularity conditions for neural networks," *ArXiv:1710.06910v2*, Oct 2017.

[19] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Proc. Machine Learning and Knowledge Discovery in Databases*, 2016, pp. 795–811.

[20] S. Mei, Y. Bai, and A. Montanari, "The landscape of empirical risk for nonconvex losses," *The Annals of Statistics*, vol. 46, no. 6A, pp. 2747–2774, 2018.

[21] S. Du and J. Lee, "On the power of over-parametrization in neural networks with quadratic activation," in *Proc. International Conference on Machine Learning*, vol. 80, Jul 2018, pp. 1329–1338.

[22] L. Venturi, A. S. Bandeira, and J. Bruna, "Spurious valleys in one-hidden-layer neural network optimization landscapes," *Journal of Machine Learning Research*, vol. 20, no. 133, pp. 1–34, 2019.

[23] J. Lederer, "No spurious local minima: on the optimization landscapes of wide and deep neural networks," *arXiv:2010.00885*, 2021.

[24] Y. Zhou and Y. Liang, "Critical points of linear neural networks: Analytical forms and landscape properties," in *Proc. International Conference on Learning Representations*, 2018.

[25] K. Kawaguchi, "Deep learning without poor local minima," in *Proc. Advances in Neural Information Processing Systems*, vol. 29, 2016.

[26] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Advances in Neural Information Processing Systems*, vol. 31, 2018.

[27] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proc. Advances in Neural Information Processing Systems*, vol. 31, 2018.

[28] Y. Zhou, J. Yang, H. Zhang, Y. Liang, and V. Tarokh, "SGD converges to global minimum in deep learning via star-convex path," in *Proc. International Conference on Learning Representations*, 2019.

[29] P. C. Verpoort, A. A. Lee, and D. J. Wales, "Archetypal landscapes for deep neural networks," *Proc. National Academy of Sciences*, vol. 117, no. 36, pp. 21 857–21 864, 2020.

[30] F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht, "Essentially no barriers in neural network energy landscape," in *Proc. International Conference on Machine Learning*, vol. 80, 10–15 Jul 2018, pp. 1309–1318.

[31] I. Goodfellow, O. Vinyals, and A. Saxe, "Qualitatively characterizing neural network optimization problems," in *Proc. International Conference on Learning Representations*, 2015.

[32] Y. Hao, L. Dong, F. Wei, and K. Xu, "Visualizing and understanding the effectiveness of BERT," in *Proc. Conference on Empirical Methods in Natural Language Processing and the International Joint Conference*

on Natural Language Processing (EMNLP-IJCNLP), Nov. 2019, pp. 4143–4152.

[33] R. Mulayoff and T. Michaeli, "Unique properties of flat minima in deep networks," in *Proc. International Conference on Machine Learning*, vol. 119, 13–18 Jul 2020, pp. 7108–7118.

[34] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[35] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[36] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.