Gradient Descent Optimizes Normalization-Free ResNets

Zongpeng Zhang[a,b], Zenan Ling[b,c], Tong Lin[b], Zhouchen Lin[b,d,e,*]
[a] Center for Data Science, Academy for Advanced Interdisciplinary Studies, Peking University
[b] National Key Lab. of General AI, School of Intelligence Science and Technology, Peking University
[c] EIC, Huazhong University of Science and Technology
[d] Institute for Artificial Intelligence, Peking University
[e] Peng Cheng Laboratory
Email: zhangzongpeng@stu.pku.edu.cn, lingzenan@hust.edu.cn, {lintong, zlin}@pku.edu.cn

***Abstract*—Recent empirical studies observe that even without normalization, a deep residual network can be trained reliably. We call such a structure as normalization-free Residual Networks (N-F ResNets), which add a learnable parameter α to control the scale of the residual block instead of normalization. However, the theoretical understanding on N-F ResNets is still limited despite their empirical success. In this paper, we provide the first theoretical understanding of N-F ResNets from two perspectives. Firstly, we prove that the gradient descent (GD) algorithm can find the global minimum of the training loss at a linear rate for over-parameterized N-F ResNets. Secondly, we prove that N-F ResNets can avoid the gradient exploding or vanishing problem, by initializing the key parameter α to be a small constant. Notably, we demonstrate that the gradients of N-F ResNets are more stable than those of ResNets with Kaiming initialization. Moreover, empirical experiments on benchmark datasets verify our theoretical results.**

I. Introduction

Deep neural networks (DNNs) have achieved great success in numerous fields, including computer vision [1], speech recognition [2] and natural language processing [3]. As one of the most popular modern deep network structures, ResNets proposed by He et al. [4], [5] have achieved remarkable performance on various challenging tasks. In practice, the effective training of ResNets requires normalization techniques such as the commonly adopted batch normalization (BN) [6]. Despite the enormous empirical success of training deep networks with skip connections, the use of normalization may introduce practical challenges, such as costly computing, memory overhead [7] and the reduction of model's accuracy under train-test distribution shifts [8]–[10]. On the other hand, there is currently no general consensus on why these normalization techniques help the training process [6], [11]–[13]. Recent works [7], [14]–[16] find that none of the perceived benefits is unique to normalization. Actually, even without normalization, ResNets can still be trained well. This has been demonstrated by multiple works. For example, N-F ResNets can achieve comparable empirical performance by adding a trainable parameter α to control the scale of the residual block [14], [15], which enables adaptively controlling the scale of the gradient at initialization. The scaling factor is initialized to a small positive value, which ensures that the training can be more stable. N-F ResNets perform well on many experiments of real world datasets. Despite empirical success, there is no theoretical understanding on its effectiveness up to date. In this paper, we prove the global convergence of N-F ResNets, and further provide a theoretical explanation of the empirical success of N-F ResNets.

*:The corresponding author.

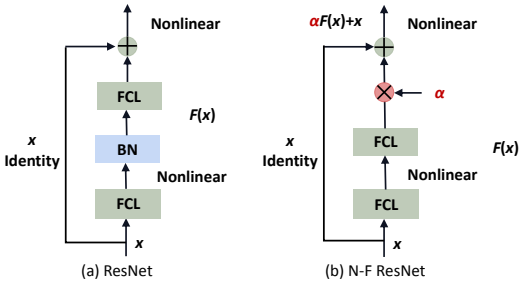

Fig. 1. The structure of ResNet (a) and N-F ResNet (b)

The structure of N-F ResNets is shown in Figure 1(b). For each layer, the N-F ResNet adds a residual connection for the input signal x and the non-linear transformation of the layer $F(x)$ which is modulated by a trainable parameter α. Compared with the original structure of ResNets (Figure 1(a)), N-F ResNets do not involve normalization but add a parameter α which controls the scale of the residual block. SkipInit [14], Rezero [15] and Fixup [16] are three common types of N-F ResNets. They initialize α to be either a small positive value or zero. The simple architectural change of N-F ResNets enables the well-conditioned Gram matrix induced by the gradient. Thus it is beneficial for making gradients well-behaved and arbitrarily deep signal propagation.

Specifically, the structure of N-F ResNets is defined as:

$$\begin{aligned} x^{(1)} &= \sqrt{\frac{c_\sigma}{m}}\sigma(\mathbf{W}^{(1)}x), \\ x^{(h)} &= x^{(h-1)} + \frac{\alpha_h}{H\sqrt{m}}\sigma(\mathbf{W}^{(h)}x^{(h-1)}),\ 2 \le h \le H, \\ f(x,\theta) &= a^\top x^{(H)}, \end{aligned} \tag{1}$$

where $x^{(0)} = x$ denotes the input, $\mathbf{W}^{(1)} \in \mathbb{R}^{m\times d}$ denotes the first weight matrix, $\mathbf{W}^{(h)} \in \mathbb{R}^{m\times m}$ denotes the weight at the h-th layer for $2 \le h \le H$, $a \in \mathbb{R}^m$ denotes

the weight of the output layer, $\sigma(\cdot)$ denotes the nonlinear activation function, $c_\sigma = \left(E_{x\sim\mathcal{N}(0,1)}\left[\sigma(x)^2\right]\right)^{-1}$ is a scaling factor to normalize the input in the initialization phase, and $\theta = (\{\mathbf{W}^{(h)}\}, a, \{\alpha_h\})$. See Section III-A for detailed descriptions of notations.

In this paper, we provide a solid theoretical analysis for N-F ResNets. The core step of our theoretical analysis is to prove that GD produces a sequence of iterations that stay inside a bounded perturbation region centered at the initial weights. And our main proof techniques are (i) the careful control of the magnitude of the change of network parameters, and (ii) a fine-grained analysis on the Gram matrix or Jacobian matrix induced by the ResNet structure.

Our contributions are summarized as follows:

- We prove the global convergence of training the N-F ResNets with GD. More specifically, over-parameterized N-F ResNets can achieve zero training loss at a linear convergence rate with GD.
- We analyze the gradient descent dynamics in the training process and demonstrate that N-F ResNets can solve the vanishing or exploding gradient problem, which is the main difficulty in training DNNs.
- We show that Kaiming initialized ResNets [17] without normalization will lead to exploding gradients, which from the opposite angle demonstrates that N-F ResNets have more stable training dynamics.

To the best of our knowledge, it is the first theoretical analysis on N-F ResNets.

II. Related Work

A. N-F ResNets

BN is often necessary to train ResNets, however its application can be limited and may introduce practical challenges [11]. For example, BN struggles when training with small batch-sizes which can incur computing and memory overhead, and enlarges the variance between different batches [7]. In settings with train-test distribution shifts, BN can reduce the generalization ability of the model and undermine a model's accuracy [12], [13], [18]. Besides, on meta-learning, it can lead to transductive inference [7]; and in adversarial training, it can hamper accuracy on both clean and adversarial examples by estimating incorrect statistics [8].

To either replace BN in general or address specific shortcomings of normalization, several recent works propose N-F ResNets. These methods introduce a learnable parameter α_h which is successively updated, such that $x_{h+1} = x_h + \alpha_h\sigma(\mathbf{W}_h x_h)$. Different kinds of N-F ResNets use different initialization methods for α. For example, SkipInit [14] sets α to a small constant or zero at initialization, i.e., the model at initialization is nearly an identity function, so there will be no worries about exploding variance of output. Fixup [16] uses the same technique and rescales α by the number of residual branches at the beginning of training. Rezero network [15] is another kind of N-F ResNets, which initializes α to zero. Empirical results in these works show that the normalization in the residual learning can be removed with the help of such kind of trainable scale parameter. Even without normalization, several kinds of very deep N-F ResNets [14]–[16] can be trained reliably with faster convergence. However, there is a lack of theoretical understanding on N-F ResNets' training dynamics in practice. These studies focus on initialization but the training dynamic of N-F ResNets remains unclear. Even under relatively simple conditions, it is unknown why N-F ResNets can achieve global convergence. Our work provides the first complete and dynamical analysis on N-F ResNets.

B. Theoretical analysis of neural networks

Several existing works have analyzed the convergence of neural networks theoretically. The global convergence of neural networks with different structures and activation functions have been proved in many papers. For example, Du et al. [19] analyze the global convergence of two-layer fully connected neural networks; Nguyen and Mondelli [20] analyze the global convergence of networks with one wide layer followed by pyramidal topology. The structure Zou et al. analyzed uses a linear activation function [21], while Zou et al. [18], Du et al. [19], Allen-Zhu et al. [22] and Zhang et al. [23], analyze networks with the ReLU activation function. Implicit equilibrium networks have also been analyzed [24].

Several related works [25], [22] and [23] consider the global convergence of ResNet. They use a fixed scale parameter, which can be seen as the static version of our N-F ResNets. These works support the usability of N-F ResNets from another perspective. However, the three works suffer from the inconsistence between real use and theory, i.e. their structure is far away from the real setting in use which needs to add normalization (e.g. BN) or the trainable parameter α in order to perform well. Moreover, the gradient stability is not proven in previous works. Our work differs from existing works in several ways: (i) we analyze the training dynamic of α, (ii) we analyze the N-F ResNet architecture, which can avoid vanish/exploding gradients even without normalization, and (iii) our analysis is consistent with the structure of N-F ResNets in practice. We corroborate that N-F ResNets can achieve global convergence in a linear rate.

III. Preliminaries

A. Notations

We consider an H-layer N-F ResNet with an activation function $\sigma(\cdot)$ and weights $\left\{\mathbf{W}^{(h)}\right\}_{h=1}^{H}$. Moreover, our network rescales the residual block by a parameter $\frac{\alpha_i}{H}$ (which is called the *residual weight*), where α_i is a learnable parameter. At the beginning of training, we set $\alpha_h(0) = 1$, $h = 1, 2, \cdots, H$. Namely, we initialize the coefficient of the residual block to be $\frac{1}{H}$, so the scale of residual weight is small at initialization. As we initialize the residual weight by the reciprocal of the network depth, we call our initialization method as *RecipDepth Init*. The commonly used notations are listed in Table 1. Using these notations, the signal of the N-F ResNet is propagated as

$$x^{(h)} = x^{(h-1)} + \frac{\alpha_h}{H\sqrt{m}}\sigma(\mathbf{W}^{(h)}x^{(h-1)}), \tag{2}$$

TABLE I
MAJOR NOTATIONS

Notation	Description
Bold capital letter	A matrix
$\mathbf{A}_{ij}$	The (i,j)-th entry of matrix $\mathbf{A}$
$\alpha_h(k)$	The rescaling parameter of the h-th layer at the k-th iteration
a	The weight of output layer, belonging to $\mathbb{R}^m$
$\|\cdot\|_2$	The ℓ_2 norm
$\sigma(\cdot)$	Nonlinear activation function
c_σ	$c_\sigma = \left(E_{x\sim N(0,1)}\left[\sigma(x)^2\right]\right)^{-1}$
H	The depth of network
m	The width of network
n	The number of data points in the training set
$[n]$	$[n] = \{1, 2, \cdots, n\}$
$x^{(0)}$	The input vector, belonging to $\mathbb{R}^d$
y	The labels of training inputs
$\mathbf{W}^{(1)}$	The first weight matrix, belonging to $\mathbb{R}^{m\times d}$
$\mathbf{W}^{(h)}$	The weight matrix of the h-th layer, $2 \le h \le H$
$u_i(k)$	The prediction of the i-th sample at the k-th iteration
$u(k)$	$u(k) = (u_1(k), u_2(k), \cdots, u_n(k))^\top$
λ_0	The smallest eigenvalue of the Gram matrix $\mathbf{K}^{(H)}$
$\lambda(\mathbf{A})$	The eigenvalues of matrix $\mathbf{A}$
$\lambda_{\min}(\mathbf{A})$	The smallest eigenvalue of matrix $\mathbf{A}$
$\lambda_{\max}(\mathbf{A})$	The largest eigenvalue of matrix $\mathbf{A}$
$O(\cdot)$	For two nonnegative sequences $\{b_k\}$ and $\{d_k\}$, there is $b_k = O(d_k)$, if $b_k \le C_1 d_k$ for some absolute constant $C_1 > 0$
$\Omega(\cdot)$	For two nonnegative sequences $\{b_k\}$ and $\{d_k\}$, there is $b_k = \Omega(d_k)$, if $b_k \ge C_2 d_k$ for some absolute constant $C_2 > 0$
$\Theta(\cdot)$	If $b_k = O(d_k)$ and $b_k = \Omega(d_k)$, there is $b_k = \Theta(d_k)$

for $2 \le h \le H$. α_h plays an important role in the convergence of the network, as it controls the magnitude of the entire gradient flow.

We make several assumptions as follows. Firstly, we use the following Gaussian noise initialization for GD algorithm to find the global minimizer of the empirical loss.

Assumption 1: Each entry of $\mathbf{W}$ and a uses standard initialization and is sampled from a Gaussian distribution: $\mathbf{W}_{ij}^{(h)} \sim \mathcal{N}(0, \frac{1}{m})$, $a_i \sim \mathcal{N}(0, \frac{1}{m})$. α_h is initialized to 1.

We extract the $\frac{1}{m}$ term of weights before activation function for the simplicity of proof.

Assumption 2: We suppose that $\sigma(\cdot)$ is analytic but is not a polynomial function. We further assume that there exists a constant $L > 0$ such that $\sigma(0) \le L$ and $\sigma(\cdot)$ is L-Lipschtiz and L-smooth. Namely, there exists a constant $L > 0$ such that for any $x, y \in \mathbb{R}$,

$$|\sigma(x) - \sigma(y)| \le L|x - y| \quad and \quad |\sigma'(x) - \sigma'(y)| \le L|x - y|.$$

Remark 1: Assumption 2 is used to show the stability of the training process. It is leveraged to guarantee the positive definiteness of certain Gram matrices which we will define later. Typical activation functions that satisfy Assumption 2 include softplus, sigmoid, tanh, GeLU, swish, etc.

B. Problem Setup

We define the i-th individual prediction at the k-th iteration as $u_i(k) = f(\theta(k), x_i)$, and denote $u(k) = (u_1(k), u_2(k), \cdots, u_n(k))^\top$. We use the square loss function and write the loss of all data points as $\mathcal{L}(\theta(k)) = \frac{1}{2}\|y - u(k)\|_2^2 = \frac{1}{2}\sum_{i=1}^n (y_i - u_i(k))^2$. Then the empirical risk minimization problem with the square loss function can be written as:

$$\min_\theta \mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^n (f(\theta, x_i) - y_i)^2,$$

where $\{x_i\}_{i=1}^n$ are the training inputs and $\{y_i\}_{i=1}^n$ are their labels. We train all layers by the GD algorithm with a constant positive step size η. Below are the equations of gradient update. For $k = 1, 2, \cdots,$ and $h = 1, 2, \cdots, H$,

$$\begin{cases} \mathbf{W}^{(h)}(k) = \mathbf{W}^{(h)}(k-1) - \eta \frac{\partial \mathcal{L}(\theta(k-1))}{\partial \mathbf{W}^{(h)}(k-1)}, \\ a(k) = a(k-1) - \eta \frac{\partial \mathcal{L}(\theta(k-1))}{\partial a(k-1)}, \\ \alpha_h(k) = \alpha_h(k-1) - \eta \frac{\partial \mathcal{L}(\theta(k-1))}{\partial \alpha_h(k-1)}. \end{cases} \tag{3}$$

IV. MAIN RESULTS

A. Global convergence of N-F ResNets

In this section, we consider the convergence of GD for training an N-F ResNet (1). We first provide the expressions of the Gram matrices $\mathbf{G}^{(h)}$ and $\mathbf{K}^{(h)}$. These two matrices play an important role in the analysis. Then we show how much over-parameterization is needed to ensure the global convergence of GD.

1) Proof sketch: The core of our proof technique is (1) the careful control of the magnitude of the change of α, and (2) a fine-grained analysis on the Gram matrix induced by the ResNet structure.

For Gram matrix $\mathbf{G}^{(h)}$ (Definition 1), we prove that there holds $\|y - u(k+1)\|_2^2 \le (1 - 2\eta\lambda_{\min}(\mathbf{G}^{(H)}(k))\|y - u(k)\|_2^2$ in the dynamics. If $\lambda_{\min}(\mathbf{G}^{(H)}(k))$ is uniformly bounded away from zero, then we can conclude that the loss decreases in a linear rate (Theorem 1).

We prove the uniform lower bound of $\lambda_{\min}(\mathbf{G}^{(H)}(k))$ (Lemma 2) in two steps. Firstly, we prove that at initialization, $\mathbf{G}^{(H)}(0)$ is close to $\mathbf{K}^{(H)}$ (Definition 3). Secondly, we prove that in the training process $\mathbf{G}^{(H)}(k)$ is close to $\mathbf{G}^{(H)}(0)$ for $k = 0, 1, \cdots$. This shows that $\left\{\mathbf{G}^{(H)}(k)\right\}$ is a matrix sequence that is close to Gram matrix $\mathbf{K}^{(H)}$.

Our main theoretical analysis is composed of two steps. Firstly, we prove that the gradient of initial N-F ResNet is stable. Secondly, we show that the sequence of iterations of α stay inside a bounded perturbation region, and the training loss function of ResNets achieves good locally linear convergence. These two results imply the global convergence of GD.

2) Detailed Results: The Gram matrix $\mathbf{G}^{(h)}$ is a kernel induced by the gradient to the weights of the h-th layer. The detailed definition is given below.

Definition 1: $\mathbf{G}^{(h)} \in \mathbb{R}^{n\times n}$ and

$$\mathbf{G}_{ij}^{(h)}(k) = \begin{cases} \left\langle \frac{\partial u_i(k)}{\partial \mathbf{W}^{(h)}(k)}, \frac{\partial u_j(k)}{\partial \mathbf{W}^{(h)}(k)} \right\rangle, & h = 1, \cdots, H, \\ \left\langle \frac{\partial u_i(k)}{\partial a(k)}, \frac{\partial u_j(k)}{\partial a(k)} \right\rangle, & h = H+1. \end{cases} \tag{4}$$

The Gram matrix contains rich information about the gradients. Especially, its eigenvalues are good proxies for assessing

how stable the gradient is. Note that $\mathbf{G}^{(h)}(k)$ is a positive semi-definite matrix for $h \in [H+1]$.

Definition 2: $\mathbf{G} \in \mathbb{R}^{n\times n}$ and

$$\mathbf{G}_{ij}(k) \triangleq \sum_{h=1}^{H+1} \mathbf{G}_{ij}^{(h)}(k) + \sum_{h=1}^{H} \left\langle \frac{\partial u_i(k)}{\partial \alpha_h(k)}, \frac{\partial u_j(k)}{\partial \alpha_h(k)} \right\rangle. \quad (5)$$

In the proving process, it is important to prove that $\mathbf{G}$ is positive definite. Because each term of $\mathbf{G}$ is positive semi-definite, it is enough to prove that there exists one term being positive definite. Thus, at the cost of a minor degradation in convergence rate, we focus on $\mathbf{G}^{(H)}(k)$, the Gram matrix induced by the weights of last but one layer.

$\mathbf{G}^{(h)}$ plays an important role in our analysis. On one hand, the eigenvalues of $\mathbf{G}^{(h)}$ are crucial indicators to measure the smallest and the largest gradient quantities: for any $i \in [n]$, there holds

$$\lambda_{\min}(\mathbf{G}^{(h)}) \leq \mathbf{G}_{ii}^{(h)} = \left\| \frac{\partial u_i}{\partial \mathbf{W}^{(h)}} \right\|_2^2 \leq \lambda_{\max}(\mathbf{G}^{(h)}). \quad (6)$$

This means that, if the eigenvalues are upper bounded and lower bounded, the gradient can also be bounded. On the other hand, the Gram matrix $\mathbf{G}^{(h)}$ influences the convergence speed of sequence $\{y - u(k)\}_k$. We will prove that

$$\|y-u(k+1)\|_2^2 \leq \left(1 - 2\eta\lambda_{\min}(\mathbf{G}^{(H)}(k)\right) \|y-u(k)\|_2^2, \quad (7)$$

i.e. there is a direct link between $\mathbf{G}^{(H)}(k)$ and the perturbation of two adjacent iterations. In order to achieve a uniform convergence rate, we need to lower bound the smallest eigenvalue of $\mathbf{G}^{(H)}(k)$ for any $k = 0, 1, \cdots$.

We first find the lower bound of $\lambda_{\min}(\mathbf{G}^{(H)}(0))$, then we uniformly bound $\lambda_{\min}(\mathbf{G}^{(H)}(k))$ for $k = 1, 2, \cdots$. Driven by the need of the lower bound of $\lambda_{\min}(\mathbf{G}^{(H)}(0))$, the definition of $\mathbf{K}^{(H)}$ arises. The recursive equation for the key Gram matrix $\mathbf{K}^{(H)}$ is given below.

Definition 3: For $(i,j) \in [n] \times [n]$ and $h = 2, \cdots, H-1$:

$$\begin{aligned}
\mathbf{K}_{ij}^{(0)} =& \langle x_i, x_j \rangle, \mathbf{A}_{ij}{}^{(1)} = \begin{pmatrix} \mathbf{K}_{ii}^{(0)} & \mathbf{K}_{ij}^{(0)} \\ \mathbf{K}_{ji}^{(0)} & \mathbf{K}_{jj}^{(0)} \end{pmatrix}, \\
\mathbf{K}_{ij}^{(1)} =& E_{(u,v)^\top \sim \mathcal{N}(0, \mathbf{A}_{ij}^{(1)})} \left[c_\sigma \sigma(u)\sigma(v) \right], \\
b_i^{(1)} =& \sqrt{c_\sigma} E_{u \sim \mathcal{N}(0, \mathbf{K}_{ii}^{(0)})} [\sigma(u)], \\
\mathbf{A}_{ij}^{(h)} =& \begin{pmatrix} \mathbf{K}_{ii}^{(h-1)} & \mathbf{K}_{ij}^{(h-1)} \\ \mathbf{K}_{ji}^{(h-1)} & \mathbf{K}_{jj}^{(h-1)} \end{pmatrix}, \\
\mathbf{K}_{ij}^{(h)} =& \mathbf{K}_{ij}^{(h-1)} + E_{(u,v)^\top \sim \mathcal{N}(0, \mathbf{A}_{ij}^{(h)})} \left[\frac{\alpha_h b_i^{(h-1)} \sigma(u)}{H} + \right. \quad (8) \\
& \left. \frac{\alpha_h b_j^{(h-1)} \sigma(v)}{H} + \frac{\alpha_h^2 \sigma(u)\sigma(v)}{H^2} \right], \\
b_i^{(h)} =& b_i^{(h-1)} + \frac{\alpha_h}{H} E_{u \sim \mathcal{N}(0, \mathbf{K}_{ii}^{(h-1)})} [\sigma(u)], \\
\mathbf{K}_{ij}^{(H)} =& \frac{\alpha_H^2}{H^2} \mathbf{K}_{ij}^{(H-1)} E_{(u,v)^\top \sim \mathcal{N}(0, \mathbf{A}_{ij}^{(H)})} \left[\sigma'(u)\sigma'(v) \right].
\end{aligned}$$

For $h = 1, \cdots, H-1$, the Gram matrix $\mathbf{K}^{(h)}$ reflects the correlation of forward propagation of the h-th layer in an infinitely wide N-F ResNet. And $\mathbf{K}^{(H)}$ reflects the correlation between gradients of the last layer. $\mathbf{K}_{ij}^{(h)}$ is the expectation of the correlation between the corresponding outputs of the i-th sample and the j-th sample in the h-th layer, i.e. $\mathbf{K}_{ij}^{(h)} = E\left\langle x_i^{(h)}, x_j^{(h)} \right\rangle$. When m becomes infinity, $b^{(h)}$ is equivalent to the output of the h-th layer. We can see that the definition of $\mathbf{K}^{(H)}$ also depends on the sequence $\{b^{(h)}\}_{h=1}^{H-1}$. This dependency comes from the skip connection block in the ResNet architecture.

The Gram matrix $\mathbf{K}^{(H)}$ is closely related to the uniform convergence rate of all iterations, because we can prove that Gram matrix $\mathbf{K}^{(H)}$ is the limit of the initial matrix $\mathbf{G}^{(H)}(0)$ as $m \to \infty$. The Gram matrix $\mathbf{K}^{(H)}$ is the key to the whole analysis, as its smallest eigenvalue λ_0 will determine the convergence rate and the amount of over-parameterization. The relation between the Gram matrix and convergence rate will be described in Theorem 1.

As a technical remark, we note that if no two input vectors are parallel, then $\mathbf{K}^{(H)}$ is positive definite. In Proposition 1 as follows, we can show that, if none of the data points are parallel and the activation function is analytic but not polynomial, then the eigenvalues of Gram matrix $\lambda(\mathbf{K}^{(H)}) > 0$.

Proposition 1: Assume that $\sigma(\cdot)$ satisfies Assumption 2, and for any $i, j \in [n]$, x_i and x_j are not parallel. Then we have $\lambda_0 > 0$.

The Gram matrices play an important role in displaying the dynamics of gradients. First, we present that the initial Gram matrices $\mathbf{G}^{(H)}(0)$ are closely connected to $\mathbf{K}^{(h)}$ via their smallest eigenvalues, as shown below.

Lemma 1 (The full rankness of $\mathbf{G}^{(H)}(0)$): Assume that the number m of neurons per layer is $\Omega\left(\frac{n^2 \log(Hn/\delta)}{\lambda_0^2}\right)$, then with probablity at least $1-\delta$ we have $\lambda_{\min}(\mathbf{G}^{(H)}(0)) \geq \frac{3}{4}\lambda_0$. Combining Lemma 1 and Proposition 1, we can show that $\mathbf{G}^{(H)}(0)$ is strictly positive definite. The proof of Lemma 1 uses the same technique as [25]. Several previous works also prove the positive definiteness of Gram matrix [19]–[21], but their settings differ from ours in the perspective of network structures and activation functions.

Meanwhile, the lemma characterizes how much over-parameterization is needed to ensure the connection between $\lambda_{\min}(\mathbf{G}^{(H)}(0))$ and λ_0. The assumed lower bound on m depends on the number n of samples and the depth H of network, etc.

As parameters of N-F ResNets perturb in the neighborhood of the initial point, we can also lower bound $\lambda_{\min}(\mathbf{G}^{(H)}(k))$ for $k = 1, 2, \cdots$, thus achieving the uniform lower bound of the smallest eigenvalues. With the help of Lemma 1 and Hoffman-Wielandt theorem [26], we get the lemma as follows.

Lemma 2 (The full rankness of $\mathbf{G}^{(H)}(k)$): Assume that the number m of neurons per layer is $\Omega\left(\frac{n^2 \log(Hn/\delta)}{\lambda_0^2}\right)$, then with probablity at least $1-\delta$ we have $\lambda_{\min}(\mathbf{G}^{(H)}(k)) \geq \frac{\lambda_0}{4}$.

Lemma 2 implies that the magnitude of gradients at any iteration is greater than $\frac{\lambda_0}{4}$, which is a positive scalar, because there holds $\left\| \frac{\partial u_i}{\partial \mathbf{W}^{(H)}} \right\|_2^2 \geq \lambda_{\min}(\mathbf{G}^{(H)})$ for any $i \in [n]$. This

theorem is key to prove the global convergence of the N-F ResNet structure (1), as it can uniformly lower bound $\mathbf{G}^{(H)}(k)$'s smallest eigenvalue at any k-th iteration, which contributes to the convergence rate between 0 and 1.

With these lemmas, we present our convergence result as shown in Theorem 1. The detailed proof is presented in Appendix.

Theorem 1 (The convergence of the loss): Assume for all $i \in [n]$, $\|x_i\|_2 = 1$, $y_i = O(1)$, $m = \Omega\left(\max\left\{\frac{n^4}{\lambda_0^4 H^6}, \frac{n^2}{\lambda_0^2 H^2}, \frac{n}{\delta}, \frac{n^2\log(Hn/\delta)}{\lambda_0^2}\right\}\right)$ and we set the step size $\eta = O\left(\frac{\lambda_0 H^2}{n^2}\right)$, then with probability at least $1-\delta$ over the random initialization, we have

$$\|y - u(k)\|_2^2 \leq \left(1 - \frac{\eta\lambda_0}{2}\right)^k \|y - u(0)\|_2^2, \quad k = 1, 2, \cdots. \tag{9}$$

The theorem shows that if the width m is large enough and we set step size η appropriately, the loss of the N-F ResNet (1) can converge to a global minimum at a linear rate. The main assumption of the theorem is that we need a large enough width m for each layer, which depends on n, H and λ_0. The dependency on n is only polynomial. m also polynomially depends on $\frac{1}{\lambda_0}$. The dependency on the number H of layers is logarithmic. Equation (9) holds with higher probability when m is larger, thus training process is more stable.

Without loss of generality, the training loss converges to zero as the iteration number k tends to infinity, because the training loss function $\mathcal{L}(\theta(k)) = \frac{1}{2}\|y - u(k)\|_2^2$ decreases geometrically as Theorem 1 declares. If we make rectifications of the input data, the loss function just changes by an offset which does not affect the GD dynamics. Thus, the global minimum is consistent with zero training loss.

To prove Theorem 1, our main idea of analysis is that random initialization followed by GD produces a sequence of iterations that stay inside a bounded perturbation region centered at the initial weights. In the perturation region, the training loss function of ResNets enjoys good locally linear convergence. Therefore, it is worth mentioning that we have each $\alpha_h(k)$ and weight matrix close to their initialization. The following lemma shows the result in detail.

Lemma 3 (Bounded weight pertubation): If Assumptions 1 and 2 hold and $\eta \leq c\frac{H^2}{m}$ for some small constant $c > 0$, then we have

$$\|\alpha_h(k) - \alpha_h(0)\|_F \leq O(1),$$
$$\left\|\mathbf{W}^{(h)}(k) - \mathbf{W}^{(h)}(0)\right\|_F \leq O(\sqrt{n}), \quad k = 0, 1, 2, \cdots.$$

Then we quantitatively study the bound of output during training. The following lemma shows that the outputs of any layer are close to its initial value, being within the range of $O(1)$ under the over-parameterization condition.

Lemma 4 (Bounded output pertubation): Suppose that $\sigma(\cdot)$ is L-Lipschitz and for $h \in [H]$, $\left\|\mathbf{W}^{(h)}(0)\right\|_2 \leq c_{w,0}\sqrt{m}$, $\left\|x^{(h)}(0)\right\|_2 \leq c_{x,0}$, $\|a_h(0)\|_2 \leq c_{a,0}$ and $\left\|\mathbf{W}^{(h)}(k) - \mathbf{W}^{(h)}(0)\right\|_F \leq \sqrt{m}R$ for some constant $c_{w,0}, c_{x,0}, c_{a,0} > 0$ and $R \leq c_{w,0}$. Then with probability at least $1-\delta$, we have

$$\left\|x^{(h)}(k) - x^{(h)}(0)\right\|_2 = O(1).$$

By the way, Lemma 4 claims that the output of the N-F ResNet structure can be bounded by the input.

Remark 2: We also analyze the magnitude change between two adjacent iterations as follows. It shows the perturbation during training. The magnitude of $\mathbf{W}$'s change is $O(\sqrt{n})$, that of α_h is $O(1)$, and that of x's change is $O(1)$ (Lemma 4). So the output change of two neighboring layers is $O(1)$. When the network is wide enough, it leads to a bounded perturbation. The perturbations from weight matrices propagate to the input of each layer.

Remark 3: It is noteworthy that our analysis works in both *RecipDepth Init* and zero Init [15] of α. The above analysis is under the condition of RecipDepth Init. When $\alpha(0) = 0$, the gradient of weight $\mathbf{W}$ is zero, i.e., $\mathbf{W}$ is not trained at the first step. Therefore, the process from the second step is consistent with our above analysis.

B. No vanishing or exploding gradient

Vanishing and exploding gradient is the main difficulty in training deep neural networks. In this subsection, we establish that the N-F ResNet structure can avoid vanishing and exploding gradients throughout the training. This is the advantage of N-F ResNet structure.

Below is the theoretical analysis that the N-F ResNet structure (1) can avoid vanishing gradient. Even without BN, N-F ResNets trained in the way of RecipDepth Init can help address the problem. It shows that $\mathbf{G}^{(H)}(k)$ can retain the good properties at the initial time, such as the full rankness, etc.

As declared in the previous section, we only need to prove that $\|y - u_0\|$ is independent of n and m. This is not obvious as it may explode at the initial time. Then we can choose an appropriate η to ensure that the loss of the N-F ResNet structure converges to a global minimum at a linear rate.

Lemma 5 (The boundedness of initial output): If $m = \Omega\left(\frac{n}{\delta}\right)$, we have with probability at least $1-\delta$ over random initialization that

$$\frac{1}{c_{x,0}} \leq \|x_i^{(h)}(0)\|_2 \leq c_{x,0}, \text{ for all } h \in [H] \text{ and } i \in [n].$$

for some universal constant $c_{x,0} > 1$ (only depending on σ).

Lemma 5 shows that the magnitude of initial output is of the order of $\Omega(1)$ with high probability. It is both lower-bounded and upper-bounded. The bound only depends on the properties of the activation function. Moreover, it shows that the initial loss is a finite value. Combined with the result of Lemma 4, we can show that the output is always bounded.

Then we quantitatively study the bound of gradients during training. The following theorem shows that N-F ResNets can avoid vanishing or exploding gradients.

Theorem 2 (No vanishing or exploding gradient throughout training): There exists a lower bound m_1 and upper

bound M of the magnitude of gradients during training for all iteration k, such that

$$m_1 \leq \left\| \frac{\partial u_i}{\partial \xi} \right\|_2 \leq M, \quad \text{for all } i \in [n], k \in \mathbb{N} \text{ and } \xi \in \theta,$$

where $m_1 > 0$ and $M = O(\sqrt{n})$.

Theorem 2 is the result of Lemmas 2 and 3. With a constant learning rate, Lemma 3 shows that N-F ResNets can counteract exploding gradient and benefit from a more stable training dynamics. On the other hand, Lemma 2 shows that the magnitude of gradients is greater than $\frac{\sqrt{\lambda_0}}{2}$, thus N-F ResNets can avoid vanishing gradients.

C. Comparison with conventional ResNet scheme

Standard initialization [17], [27], [28] is the commonly-used initialization scheme for ResNets. So it is natural to consider vanilla ResNets with standard initialization strategy in normalization free conditions. However, existing researches find that such an initialization does not work. It has been observed [16], [29], [30] that without normalization techniques, ResNets do not account properly for the effect of residual connections and this causes exploding gradients. In this subsection, we will give theoretical analysis on this phenomenon.

We assume that the ResNet structure is described by (10), and we apply Gaussian initialization, i.e., $\mathbf{W}_{ij}^{(h)}(0) \sim \mathcal{N}(0,1)$.

$$x^{(h)} = x^{(h-1)} + \frac{1}{\sqrt{m}} \sigma(\mathbf{W}^{(h)} x^{(h-1)}), \text{ for } 1 \leq h \leq H. \quad (10)$$

Through simple calculations, the gradients of N-F ResNets and vanilla ResNets can both be represented as follows,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(h)}} = \frac{\partial \mathcal{L}}{\partial x^{(H)}} \cdot \boxed{\prod_{l=h+1}^{H} \left(\frac{\partial x^{(l)}}{\partial x^{(l-1)}} \right)} \cdot \frac{\partial x^{(h)}}{\partial \mathbf{W}^{(h)}}. \quad (11)$$

The most important component of the gradients is the multiplicative term which is boxed in (11). And the boxed term of N-F ResNets is shown as a form of $\prod_{l=h+1}^{H} \left(\mathbf{I} + \frac{\alpha_l}{H\sqrt{m}} \mathbf{J}_i^{(l)} \mathbf{W}^{(l)} \right)$, where $\mathbf{J}^{(h')} \triangleq \operatorname{diag}\left(\sigma'\left((w_1^{(h')})^\top x^{(h'-1)} \right), \cdots, \sigma'\left((w_m^{(h')})^\top x^{(h'-1)} \right) \right) \in \mathbb{R}^{m \times m}$. Compared with N-F ResNets, the divergent multiplicative term of vanilla ResNets makes the gradients have worse performance. The rigorous mathematical analysis is shown as follows.

Let $\mathbf{J} = \frac{\partial x^{(H)}}{\partial x^{(0)}}$ denote the input-output Jacobian matrix of vanilla ResNets. The condition number of $\mathbf{J}\mathbf{J}^\top$ is good measurement for assessing how stable the gradient is. Based on the previous works [16], [29], [30], we can get the following conclusion.

Theorem 3 (Exploding gradient of standard initialized ResNets): For ResNets with standard initialization, the condition number of $\mathbf{J}\mathbf{J}^\top$ grows at least linearly with depth. More specifically,

$$\lambda_{\max}(\mathbf{J}\mathbf{J}^\top) = \Omega(H),$$
$$\lambda_{\min}(\mathbf{J}\mathbf{J}^\top) = O(1).$$

As Lemma 3 shows, the maximum eigenvalue of $\mathbf{J}\mathbf{J}^\top$ grows in an unbounded way with the network depth. So the Jacobian matrix is ill-conditioned and the learning dynamics is unstable when the network is deep. Even at initialization, the output of each layer is linearly dependent on the depth. The output will approach to infinite, which reflects that the gradient of standard initialization is unstable. So standard initialization is easier to suffer from the exploding gradient issue.

Compared with standard initialized ResNets, the initial state of N-F ResNets (1) has a much smaller gradient norm, and has the ability to propagate informative activation patterns in deeper layers. It reduces the dependence on normalization, via which we can train a deep residual network reliably even without normalization. Avoiding the ill-conditioned gradient, the N-F ResNet structure can use a larger learning rate.

The parameter α in the N-F ResNet plays a role in controlling the scale of the gradient. It can achieve the comparable empirical performance of the ResNet with normalization. The scaling factor ensures that the network preserves the size of every input in expectation.

V. Experiment

In this section, we implement several numerical experiments to verify our main theoretical conclusions in Section IV. The first experiment is performed to test the global convergence of N-F ResNet structure (1), and how the amount of over-parameterization affects the convergence rates. The second one is to test that the N-F ResNet structure can keep away from vanishing or exploding gradients and test the stability of parameters of various layers over time. The third one aims at verifying that N-F ResNets has better training dynamics than ResNets trained in other ways.

A. Training dynamic loss of different widths

In this subsection, we verify Theorem 1. We first evaluate our method on various datasets including MNIST [31], CIFAR10 [32], and a randomly generated dataset for classification tasks. As for the generation of data points and labels, we uniformly generate n = 1,000 data points from a d = 1,000 dimensional unit sphere; and labels are generated from a one-dimensional standard Gaussian distribution. The depth H of ResNet is set as 64. And we use the N-F ResNet shown in Figure 1(b), with the softplus activation function. In order to better observe the changing trend of the training loss, we run 2,000 iterations of GD with a fixed learning rate $\eta = 10^{-4}$ in this experiment.

B. No exploding or vanishing gradient

In this experiment, we test different values of network width m. It can be seen from Figure 2 that, the loss decreases at a linear rate, as m becomes larger, the loss convergence speed becomes faster, and the training loss at iteration 2,000 becomes smaller. We believe that the reason is as m increases, Gram matrices become more stable, and therefore have a larger smallest eigenvalue. So the above empirical results are consistent with Theorem 1.

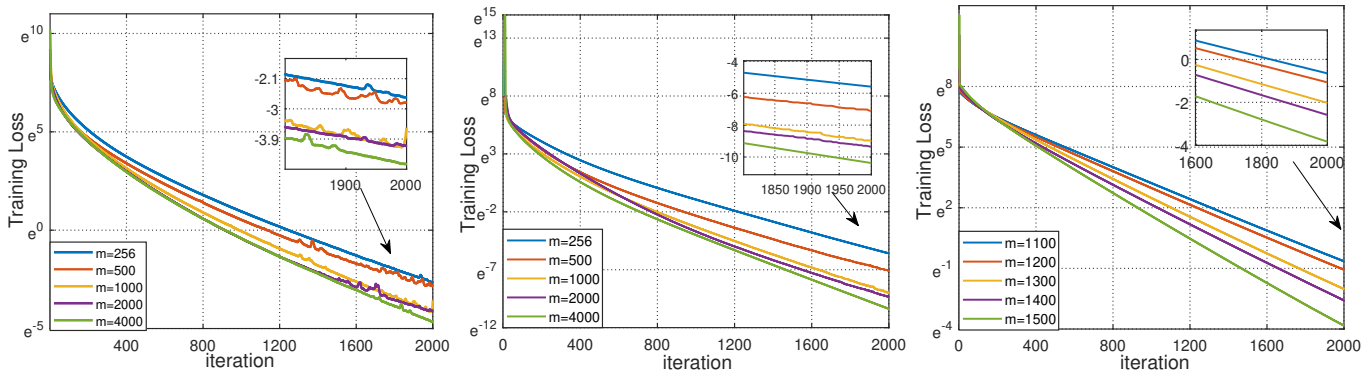

Fig. 2. Training dynamic loss of different widths (a) MNIST; (b) CIFAR10; (c) Randomly generated dataset.

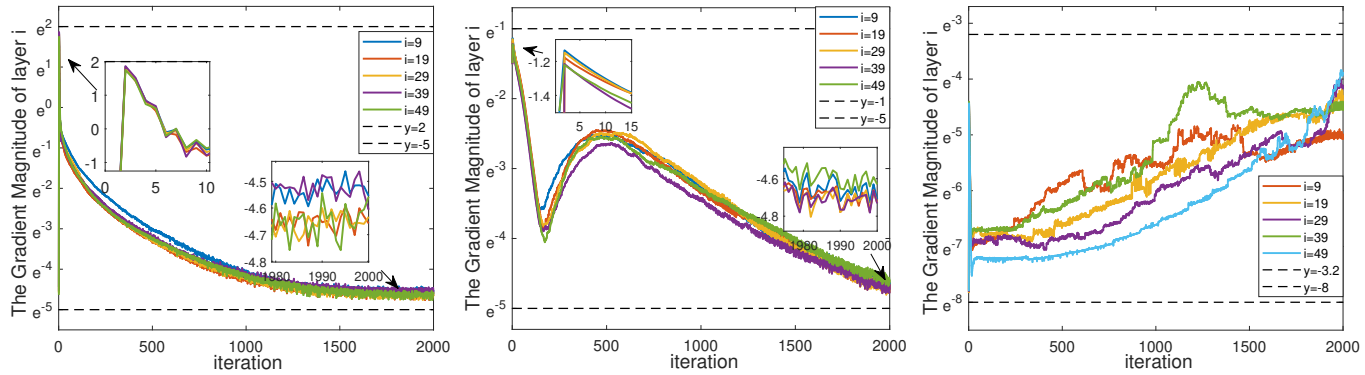

Fig. 3. The magnitude of gradient in different layers (a) MNIST; (b) CIFAR10; (c) Randomly generated dataset.

We conduct the second experiment to verify Theorem 2. The width m and depth H of structure (1) are set as 256 and 64, respectively. About the experimental setup of this experiment, except for the value of width m, other parameter settings are the same as those in Subsection V-A, including optimizer, network structure, loss function, learning rate, initialization method of α, etc.

Then we test the magnitude of gradients in different layers. We run 2,000 iterations of GD and use a fixed step size $\eta = 10^{-4}$. As Figure 3 shows, the gradient of each layer is always bounded, neither exploding nor vanishing. The upper bound and lower bound are drawn with a dotted line in the figure. The empirical results convince that N-F ResNets can avoid vanishing or exploding gradient.

C. The performance of ResNets trained in other ways

We first compare the gradient stability of N-F ResNets with standard initialized ResNets to verify Theorem 3. We train 1,000-layer ResNets on the MNIST dataset. Under the same setting, the ℓ_2 norm of output vector via two kinds of ResNets in the normalization-free circumstance is shown in Table II. It can be seen that when training via standard initialized ResNets, the magnitude of output will increase to an extraordinarily large value even at the first iteration, so the performance cannot be guaranteed to be stable in the subsequent process. Compared with standard initialized ResNet whose Gram matrix is ill conditioned, the N-F ResNet structure with RecipDepth Init is apparently much more stable, which significantly improves the gradient performance.

TABLE II
OUTPUT MAGNITUDE OF DIFFERENT RESNETS IN THE FIRST ITERATION UNDER NORMALIZATION-FREE CONDITION.

Layer Number	Standard Initialized ResNets	N-F ResNets(Ours)
1-st layer	6.36	5.52
64-th layer	1.57×10^{6}	5.59
128-th layer	4.21×10^{11}	5.68
256-th layer	NaN	5.85
512-th layer	NaN	6.25
1024-th layer	NaN	7.23

Besides, we compare the training dynamics of ResNets trained in different ways. As shown in Figure 4, N-F ResNets achieve the state-of-the-art performance. N-F ResNets converge faster than ResNets with batch normalization, which demonstrates its comparable performance to replace normalization. Moreover, the fixed scale ResNets analyzed in [25] has inferior performance to ours due to the inconsistency between theory and real use.

VI. CONCLUSIONS

In this paper, we analyze the gradient dynamics of N-F ResNets with the quadratic loss function. Firstly, we prove that the loss of deep over-parametrized N-F ResNets using GD can linearly converge to zero. Secondly, we demonstrate that N-F ResNets with RecipDepth Init have more stable gradients

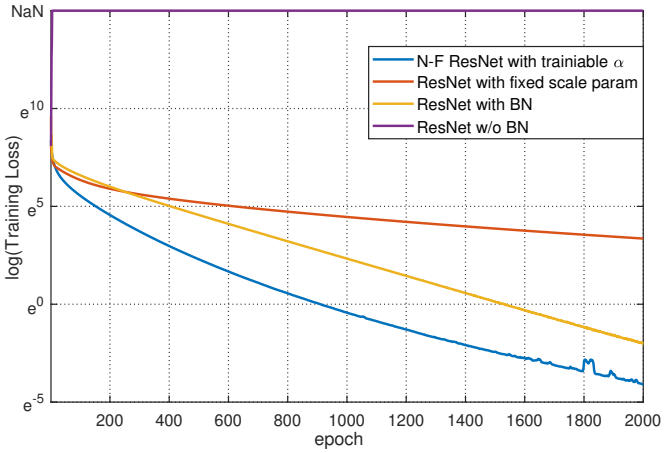

Fig. 4. Training dynamic loss of ResNets trained in different ways on the MNIST dataset.

than Kaiming initialization, avoiding vanishing or exploding gradients, hence enabling efficient training. To the best of our knowledge, it is the first theoretical analysis on N-F ResNets. All our theoretical results are verified by experiments. The theoretical and experimental results provide solid evidences that a deep residual network can be trained reliably without normalization. This work paves the way for future work on new analyses about N-F ResNets and the essential benefits of normalization.

We provide the full proof in the Appendix and the link to it is https://github.com/snowbbbb/Appendix-for-GD-Optimizes-N-F-ResNets.

ACKNOWLEDGMENT

Z. Lin was supported by National Key R&D Program of China (2022ZD0160302), the major key project of PCL, China (No. PCL2021A12), the NSF China (No. 62276004), and Qualcomm.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, vol. 25, pp. 1106–1114, 2012.

[2] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 14–22, 2011.

[3] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in International Conference on Machine learning, 2008, pp. 160–167.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[5] ——, "Identity mappings in deep residual networks," in European Conference on Computer Vision, 2016, pp. 630–645.

[6] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, 2015, pp. 448–456.

[7] A. Brock, S. De, and S. L. Smith, "Characterizing signal propagation to close the performance gap in unnormalized ResNets," in International Conference on Learning Representations, 2020.

[8] D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If ResNets are the answer, then what is the question?" in International Conference on Machine Learning, 2017, pp. 342–350.

[9] X. Wang, Y. Jin, M. Long, J. Wang, and M. I. Jordan, "Transferable normalization: Towards improving transferability of deep neural networks," Advances in Neural Information Processing Systems, vol. 32, pp. 1951–1961, 2019.

[10] C. Summers and M. J. Dinneen, "Four things everyone should know to improve batch normalization," in International Conference on Learning Representations, 2019.

[11] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" Advances in Neural Information Processing Systems, vol. 31, pp. 2488–2498, 2018.

[12] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," Advances in Neural Information Processing Systems, vol. 31, pp. 7705–7716, 2018.

[13] G. Yang, J. Pennington, V. Rao, J. Sohl-Dickstein, and S. S. Schoenholz, "A mean field theory of batch normalization," in International Conference on Learning Representations, 2018.

[14] S. De and S. Smith, "Batch normalization biases residual blocks towards the identity function in deep networks," Advances in Neural Information Processing Systems, vol. 33, pp. 19 964–19 975, 2020.

[15] T. Bachlechner, B. P. Majumder, H. Mao, G. Cottrell, and J. McAuley, "Rezero is all you need: Fast convergence at large depth," in Uncertainty in Artificial Intelligence, 2021, pp. 1352–1361.

[16] H. Zhang, Y. N. Dauphin, and T. Ma, "Fixup initialization: Residual learning without normalization," in International Conference on Learning Representations, 2018.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[18] 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.

[19] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," in International Conference on Learning Representations, 2018.

[20] Q. N. Nguyen and M. Mondelli, "Global convergence of deep networks with one wide layer followed by pyramidal topology," Advances in Neural Information Processing Systems, vol. 33, pp. 11 961–11 972, 2020.

[21] D. Zou, P. M. Long, and Q. Gu, "On the global convergence of training deep linear ResNets," in International Conference on Learning Representations, 2019.

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

[23] H. Zhang, D. Yu, W. Chen, and T.-Y. Liu, "Training over-parameterized deep resnet is almost as easy as training a two-layer network," *arXiv preprint arXiv:1903.07120*, 2019.

[24] Z. Ling, X. Xie, Q. Wang, Z. Zhang, and Z. Lin, "Global convergence of over-parameterized deep equilibrium models," in International Conference on Artificial Intelligence and Statistics, 2023.

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

[26] J. Sun, *Matrix perturbation analysis*, 2001, vol. 6.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.

[28] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, 2012, pp. 9–48.

[29] G. Yang and S. Schoenholz, "Mean field residual networks: On the edge of chaos," Advances in Neural Information Processing Systems, vol. 30, pp. 7103–7114, 2017.

[30] Z. Ling and R. C. Qiu, "Spectrum concentration in deep residual learning: a free probability approach," IEEE Access, vol. 7, pp. 105 212–105 223, 2019.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[32] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Technical report, 2009.