Mean Field Residual Networks: On the Edge of Chaos

Greg Yang*
Microsoft Research AI
gregyang@microsoft.com

Samuel S. Schoenholz
Google Brain
schsam@google.com

Abstract

We study randomly initialized residual networks using mean field theory and the theory of difference equations. Classical feedforward neural networks, such as those with tanh activations, exhibit exponential behavior on the average when propagating inputs forward or gradients backward. The exponential forward dynamics causes rapid collapsing of the input space geometry, while the exponential backward dynamics causes drastic vanishing or exploding gradients. We show, in contrast, that by converting to residual connections, with most activations such as tanh or a power of the ReLU unit, the network will adopt subexponential forward and backward dynamics, and in many cases in fact polynomial. The exponents of these polynomials are obtained through analytic methods and proved and verified empirically to be correct. In terms of the "edge of chaos" hypothesis, these subexponential and polynomial laws allow residual networks to "hover over the boundary between stability and chaos," thus preserving the geometry of the input space and the gradient information flow. For each activation function we study here, we initialize residual networks with different hyperparameters and train them on MNIST. Remarkably, our *initialization time* theory can accurately predict *test time* performance of these networks, mostly by tracking the expected gradient explosion of random residual networks. Importantly, we show, theoretically as well as empirically, that common initializations such as the Xavier or the He schemes are not optimal for residual networks, because *the optimal initialization variances depend on the depth*. Finally, we have made mathematical contributions by deriving several new identities for the kernels of powers of ReLU functions by relating them to the zeroth Bessel function of the second kind.

1 Introduction

Previous works [9, 3, 11] have shown that randomly initialized neural networks exhibit a spectrum of behavior with depth, from stable to chaotic, which depends on the variance of the initializations: the cosine distance of two input vectors converges exponentially fast with depth to a fixed point in [0, 1]; if this fixed point is 1, then the behavior is stable; if this fixed point is 0, then the behavior is chaotic. It has been argued in many prior works [1, 9] that effective computation can only be supported by a dynamical behavior that is on **the edge of chaos**. Too much stability prevents the neural network from telling apart two different inputs. While some chaotic behavior can increase the expressivity of a network, too much chaos makes the neural network think two similar inputs are very different. At the same time, the same initialization variances also control how far gradient information can be propagated through the network; the networks with chaotic forward dynamics will tend to suffer from exploding gradients, while networks with stable forward dynamics will tend to suffer from vanishing gradients.

These works have focused on vanilla (fully connected) feedforward networks. Here we consider residual networks [6, 7] (with fully-connected layers and without batchnorm), which are a family

*Work done while at Harvard University

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

of recently proposed neural network architectures that has achieved state-of-the-art performance on image recognition tasks, beating all other approaches by a large margin. The main innovation of this family of architectures is the addition of a passthrough (identity) connection from the previous layer to the next, such that the usual nonlinearity computes the "residual" between the next-layer activation and the previous-layer activation.

In this work, we seek to characterize randomly initialized residual networks. One of our main results is that random residual networks for many nonlinearities such as $\tanh$ **live on the edge of chaos**, in that the cosine distance of two input vectors will converge to a fixed point at a polynomial rate, rather than an exponential rate, as with vanilla tanh networks. Thus a typical residual network will slowly cross the stable-chaotic boundary with depth, hovering around this boundary for many layers. In addition, for most of the nonlinearities considered here, the mean field estimate of the gradient grows subexponentially with depth. In fact, for α-ReLU, the αth-power of ReLU, for $\alpha < 1$, the gradient grows only polynomially. These theoretical results provide some theoretical justification for why residual networks work so well in practice. In our experiments, we are also able to predict surprisingly well the relative performances of *trained* residual networks based only on their initialization hyperparameters, in a variety of settings. We establish theoretically and empirically that the best initialization variances for residual networks depend on the depth of the network (contrary to the feedforward case [11]), so that common initialization schemes like Xavier [4] or He [5] cannot be optimal.

In the body of this paper, we give account of general intuition and/or proof strategy when appropriate for our theoretical results, but we relegate all formal statements and proofs to the appendix.

2 Background

Consider a vanilla feedforward neural network of L layers, with each layer l having $N^{(l)}$ neurons; here layer 0 is the input layer. For the ease of presentation we assume all hidden layer widths are the same $N^{(l)} = N$ for all $l > 0$. Let $x^{(0)} = (x_1^{(0)}, \dots, x_{N^{(0)}}^{(0)})$ be the input vector to the network, and let $x^{(l)}$ for $l > 0$ be the activation of layer l. Then a neural network is given by the equations

$$x_i^{(l)} = \phi(h_i^{(l)}), \qquad\qquad h_i^{(l)} = \sum_{j=1}^{N} w_{ij}^{(l)} x_j^{(l-1)} + b_i^{(l)}$$

where (i) $h^{(l)}$ is the pre-activation at layer l, (ii) $w^{(l)}$ is the weight matrix, (iii) $b^{(l)}$ is the bias vector, and (iv) ϕ is a nonlinearity, for example $\tanh$ or ReLU, which is applied coordinatewise to its input.

To lighten up notation, we suppress the explicit layer numbers l and write

$$x_i = \phi(h_i), \qquad\qquad h_i = \sum_j w_{ij} \underline{x}_j + b_i$$

where $\bullet$ implicitly denotes $\bullet^{(l)}$, and $\underline{\bullet}$ denotes $\bullet^{(l-1)}$ (and analogously, $\overline{\bullet}$ denotes $\bullet^{(l+1)}$).

A series of papers [9, 10, 11] investigated the "average behavior" of random neural networks sampled via $w_{ij}^{(l)} \sim \mathcal{N}(0, \sigma_w^2/N), b_i^{(l)} \sim \mathcal{N}(0, \sigma_b^2)$, for fixed parameters σ_w and σ_b, independent of l. Consider the expectation of $\frac{1}{N}\sum_{i=1}^{N} x_i^2$, the normalized squared length of x, over the sampling of w and b. Poole et al. [9] showed that this quantity converges to a fixed point exponentially fast for sigmoid nonlinearities. Now suppose we propagate two different vectors $x^{(0)}$ and $(x^{(0)})'$ through the network. Poole et al. [9] also showed that the expectation of the normalized dot product $\frac{1}{N}\sum_{i=1}^{N} x_i x_i'$ converges exponentially fast to a fixed point. The ratio between the normalized squared length and the normalized dot product is the cosine distance between x and x'. Thus these two exponential convergence results show that the cosine distance converges exponentially fast to a fixed point as well. Intuitively, this means that a vanilla feedforward network "forgets" the geometry of the input space "very quickly," after only a few layers.

In addition, Schoenholz et al. [11], under certain independence assumptions, showed that the expected normalized squared norm of the gradient also vanishes or explodes in an exponential fashion with depth, with the "half-life" controlled by σ_w and σ_b. They verified that this theoretical "half-life" correlates in practice with the maximal number of layers that are admissible to good performance.

At the same time, Daniely et al. [3] published work of similar nature, but phrased in the language of reproducing kernel Hilbert spaces, and provided high probability estimates that are meaningful for the case when the width N is finite and the depth is logarithmic in the input dimension. However, they adopted a simplifying assumption that the activation functions are "normalized," and furthermore, their framework, for example, the notion of a "skeleton," does not immediately generalize to the residual network case.

In this work, we show that residual networks have very different dynamics from vanilla feedforward networks. In most cases, the cosine distance convergence rate and the gradient growth rate are subexponential in a residual network, and in most cases, these rates may be polynomial.

3 Preliminaries

Residual networks were first introduced by [6] and later refined by [7], and they are now commonplace among deployed neural systems. The key innovation there is the addition of a shortcut connection from the previous layer to the next. We define the following idealized architectures for ease of analysis. Note that we only consider fully-connected affine layers instead of convolutional layers. A **reduced residual network (RRN)** has the recurrence

$$x_i = \phi(h_i) + \underline{x}, \qquad h_i = \sum_j w_{ij}\underline{x}_j + b_i.$$

A **(full) residual network (FRN)** in addition has an affine connection given by weights v and biases a from the nonlinearity $\phi(h)$ to the next layer:

$$x_i = \sum_j v_{ij}\phi(h_j) + \underline{x}_i + a_i, \qquad h_i = \sum_j w_{ij}\underline{x}_j + b_i$$

We are interested in the "average behavior" of these network when the weights and biases, $w_{ij}^{(l)}, b_i^{(l)}, v_{ij}^{(l)}$, and $a_i^{(l)}$ are sampled i.i.d. from Gaussian distributions resp. with standard deviations $\sigma_w, \sigma_b, \sigma_v$, and σ_a, independent from l. Here we take the variance of $w_{ij}^{(l)}$ to be σ_w^2/N so that the variance of each h_i is σ_w^2, assuming each $\underline{x}_j$ is fixed (similarity for $v_{ij}^{(l)}$). Such an initialization scheme is standard in practice.

We make several key "physical assumptions" to make theoretical computations tractable:

Axiom 3.1 (Symmetry of activations and gradients)**.** *(a) We assume* $\langle (h_i^{(l)})^2\rangle = \langle (h_j^{(l)})^2\rangle$ *and* $\langle (x_i^{(0)})^2\rangle = \langle (x_j^{(0)})^2\rangle$ *for any* i, j, l*. (b) We also assume that the gradient* $\partial E/\partial x_i^{(l)}$ *with respect to the loss function* E *satisfies* $\langle (\partial E/\partial x_i^{(l)})^2\rangle = \langle (\partial E/\partial x_j^{(l)})^2\rangle$ *for any* i, j, l*.*

One can see that Axiom 3.1(a) is satisfied if the input $x^{(0)} \in \{\pm 1\}^N$ and Axiom 3.1(b) is satisfied if Axiom 3.2 below is true and the gradient at the last layer $\partial E/\partial xL \in \{\pm 1\}^N$. But in general it is justified both empirically and theoretically as an approximation, because $(h_i^{(l)})^2 - (h_j^{(l)})^2$ stays about constant with l, but $(h_i^{(l)})^2$ and $(h_j^{(l)})^2$ grow rather quickly at the same pace with l (as will be seen later in calculations), so that their additive difference becomes negligible; similarly for $(x_i^{(l)})^2$ and $(\partial E/\partial h_i^{(l)})^2$.

Axiom 3.2 (Gradient independence)**.** *(a) We assume the we use a different set of weights for back-propagation than those used to compute the network outputs, but sampled i.i.d. from the same distributions. (b) For any loss function* E*, we assume that the gradient at layer* l*,* $\partial E/\partial x_i^{(l)}$*, is independent from all activations* $h_j^{(l)}$ *and* $x_j^{(l-1)}$ *from the previous layer.*

Axiom 3.2(a) was first made in [11] for computing the mean field theory of gradients for feedforward tanh networks. This is similar to the practice of feedback alignment [8]. Even though we are the first to explicitly formulate Axiom 3.2(b), in fact it was already applied implicitly in the gradient calculations of [11]. Note that a priori Axiom 3.2(b) is not true, as $\partial E/\partial x_i^{(l)}$ depends on $\dot{\phi}(h_k^{(l+1)})$ for every k, which depend on $h_j^{(l)}$ for each j, and which depends on $x_k^{(l-1)}$ for every k. Nevertheless, in practice both subassumptions hold very well.

Now we define the central quantities studied in this paper.

Definition 3.3. Fix an input $x^{(0)}$. Define the **length quantities** $q^{(l)} := \langle (h_1^{(l)})^2 \rangle$ and $p^{(l)} := \langle (x_1^{(l)})^2 \rangle$ for $l > 0$ and $p^{(0)} = \|x^{(0)}\|^2/N$. Here the expectations $\langle \bullet \rangle$ are taken over all random initialization of weights and biases for all layers l, as $N \to \infty$ (large width limit).

Note that in our definition, the index 1 does not matter by Axiom 3.1.

Definition 3.4. Fix two inputs $x^{(0)}$ and $x^{(0)\prime}$. We write $\bullet'$ to denote a quantity $\bullet$ with respect to the input $x^{(0)\prime}$. Then define **the correlation quantities** $\gamma^{(l)} := \langle h_1^{(l)} h_1^{(l)\prime} \rangle$ and $\lambda^{(l)} := \langle x_1^{(l)} x_1^{(l)\prime} \rangle$ for $l > 0$ and $\gamma^{(0)} = x^{(0)} \cdot x^{(0)\prime}/N$, where the expectations $\langle \bullet \rangle$ are taken over all random initialization of weights and biases for all layers l, as $N \to \infty$ (large width limit). Again, here the index 1 does not matter by Axiom 3.1. Additionally, define **the cosine distance quantities** $\mathfrak{c}^{(l)} := \gamma^{(l)}/\sqrt{p^{(l)} p^{(l)\prime}}$ and $c^{(l)} := \lambda^{(l)}/\sqrt{q^{(l)} q^{(l)\prime}}$.

Definition 3.5. Fix an input $x^{(0)}$ and a gradient vector $(\partial E/\partial x_i^{(L)})_i$ of some loss function E with respect to the last layer $x^{(L)}$. Then define **the gradient quantities** $\beth^{(l)} := \langle (\partial E/\partial x_1^{(l)})^2 \rangle, \chi_\bullet^{(l)} := \langle (\partial E/\partial \bullet_1^{(l)})^2 \rangle$ for $\bullet = a, b$, and $\chi_\bullet^{(l)} := \langle (\partial E/\partial \bullet_{11}^{(l)})^2 \rangle$ for $\bullet = w, v$. Here the expectations are taken with Axiom 3.2 in mind, over both random initialization of forward and backward weights and biases, as $N \to \infty$ (large width limit). Again, the index 1 or 11 does not matter by Axiom 3.1.

Asymptotic notations. The expressions $f = O(g) \iff g = \Omega(f)$ have their typical meanings, and $f = \Theta(g)$ iff $f = O(g), g = O(f)$. We take $f(x) = \tilde{O}(g(x)) \iff g(x) = \tilde{\Omega}(f(x))$ to mean $f(x) = O(g \log^k x)$ for some $k \in \mathbb{Z}$ (this is slightly different from the standard usage of $\tilde{O}$), and $f = \tilde{\Theta}(g) \iff f = \tilde{O}(g) \ \& \ g = \tilde{O}(f)$. We introduce a new notation: $f = \breve{\Theta}(g)$ if $f(x) = O(g(x) \cdot x^\epsilon)$ and $f(x) = \Omega(g(x) \cdot x^{-\epsilon})$, as $x \to \infty$, for any $\epsilon > 0$. All asymptotic notations are sign-less, i.e. can indicate either positive or negative quantities, unless stated otherwise.

4 Overview

The primary reason we may say anything about the average behavior of any of the above quantities is the central limit theorem: every time the activations of the previous layer pass through an affine layer whose weights are sampled i.i.d., the output is a sum of a large number of random variables, and thus follows approximately Gaussian distributions. The mean and variance of these distributions can be computed by keeping track of the mean and variances of the activations in the previous layer.

In what follows, we use this technique to derive recurrence equations governing $p, q, \gamma, \lambda, \beth$ for different architectures and different activation functions. We use these equations to investigate the dynamics of $\mathfrak{c}$, the key quantity in the forward pass, and the dynamics of $\beth$, the key quantity in the backward pass.

The cosine distance $\mathfrak{c}$ in some sense measures the geometry of two vectors. If $\mathfrak{c} = 1$, then the vectors are parallel; if $\mathfrak{c} = 0$, then they are orthogonal. Just as in [9] and [11], we will show that in all of the architectures and activations we consider in this paper, $\mathfrak{c}^{(l)}$ converges to a fixed point $\mathfrak{c}^*$ as $l \to \infty$ [1]. Thus, on the average, as vectors propagate through network, the geometry of the original input space, for example, linear separability, is "forgotten" by residual networks as well as by vanilla networks. But we will prove and verify experimentally that, while Poole et al. [9] and [11] showed that the convergence rate to $\mathfrak{c}^*$ is exponential in a vanilla network, the convergence rate is rather only polynomial in residual networks, for tanh and α-ReLU (Defn 5.2) nonlinearities; see Thm B.5, Thm B.11, Thm B.17, and Thm B.18. This slow convergence preserves geometric information in the input space, and allows a typical residual network to "hover over the edge of chaos": Even when the cosine distance $\mathfrak{c}^{(l)}$ converges to 0 (corresponding to a "chaotic" regime), for the number of layers usually seen in practice, $\mathfrak{c}^{(l)}$ will reside well away from 0.

On the other hand, $\beth^{(l)}$ measures the size of gradient at layer l, and through it we track the dynamics of gradient backpropagation, be it explosion or vanishing. In contrast to vanilla tanh networks, which can experience both of these two phenomenon depending on the initialization variances, typical residual networks cannot have vanishing gradient, in the sense of vanishing $\beth^{(l)}$ as $l \to 1$; see Thm B.5 and Thm B.12. Furthermore, while vanilla tanh networks exhibit exponentially vanishing

Table 1: Main Recurrences

Antisymmetric/RRN		Any/FRN	
$q = \sigma_w^2 \underline{p} + \sigma_b^2$	$p = \mathrm{V}\phi(0, q) + \underline{p}$	$q = \sigma_w^2 \underline{p} + \sigma_b^2$	$p = \sigma_v^2 \mathrm{V}\phi(0, q) + \sigma_a^2 + \underline{p}$
$\lambda = \sigma_w^2 \underline{\gamma} + \sigma_b^2$	$\gamma = \mathrm{W}\phi(0, q, q', \lambda) + \underline{\gamma}$	$\lambda = \sigma_w^2 \underline{\gamma} + \sigma_b^2$	$\gamma = \sigma_v^2 \mathrm{W}\phi(0, q, q', \lambda) + \sigma_a^2 + \underline{\gamma}$
	$\beth = (\sigma_w^2 \mathrm{V}\dot{\phi}(0, q) + 1)\beth$		$\beth = (\sigma_v^2 \sigma_w^2 \mathrm{V}\dot{\phi}(0, q) + 1)\beth$
Theorems B.2, B.3, B.5		Theorems B.8, B.10, B.12	

or exploding gradients, all of the activation/architecture pairings considered here, except the full residual network with ReLU, have subexponential gradient dynamics. While tanh residual networks (reduced or full) has $\beth^{(0)} \approx \exp(\Theta(\sqrt{l}))\beth^{(l)}$ (Thm B.13), α-ReLU residual networks for $\alpha < 1$ have $\beth^{(0)} \approx \mathrm{poly}(l)\beth^{(l)}$ (Thm B.20). Instead of $\partial E / \partial x_i$, we may also consider the size of gradients of actual trainable parameters. For tanh and α-ReLU with $\alpha < 1$, they are still subexponential and polynomial (Thm B.21). On the other hand, while $\beth^{(0)} = \exp(\Theta(l))\beth^{(l)}$ for a ReLU resnet, its weight gradients have size independent of layer, within $O(1)$ (Thm B.21)! This is the only instance in this paper of gradient norm being completely preserved across layers.

Over the course of our investigation of α-ReLU, we derived several new identities involving the associated kernel functions, first defined in [2], which relate them to the zeroth Bessel functions (Lemmas C.31 to C.34).

5 Theoretical Results

We are interested in the two major categories of nonlinearities used today: tanh-like and rectified units. We make the following formal definitions as a foundation for further consideration.

Definition 5.1. We say a function ϕ is **tanh-like** if ϕ is antisymmetric ($\phi(-x) = -\phi(x)$), $|\phi(x)| \leq 1$ for all x, $\phi(x) \geq 0, \forall x \geq 0$, and $\phi(x)$ monotonically increases to 1 as $x \to \infty$.

Definition 5.2. Define the α-ReLU $\psi_\alpha(x) = \begin{cases} x^\alpha & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$

Note that in practice, to avoid the diverging gradient $\dot{\psi}_\alpha(x) \to \infty$ as $x \to 0$, we can use a tempered version $\Psi_\alpha(x)$ of α-ReLU, defined by $\Psi_\alpha(x) = (x + \epsilon)^\alpha - \epsilon^\alpha$ on $x > 0$ and 0 otherwise, for some small $\epsilon > 0$. The conclusions of this paper on ψ_α should hold similarly for Ψ_α as well.

By applying the central limit theorem as described in the last section, we derive a set of recurrences for different activation/architecture pairs, shown in Table 1 (see appendix for proofs). They leverage certain integral transforms [2] as in the following

Definition 5.3. Define the transforms V and W by $\mathrm{V}\phi(\mu, \rho) := \mathrm{E}[\phi(z)^2 : z \sim \mathcal{N}(\mu, \rho)]$ and $\mathrm{W}\phi(\mu, \rho, \rho', \nu) := \mathrm{E}[\phi(z)\phi(z') : (z, z') \sim \mathcal{N}(\mu, \begin{pmatrix} \rho & \nu \\ \nu & \rho' \end{pmatrix})]$.

These recurrences are able to track the corresponding quantities in practice very well. For example, Fig. 1 compares theory vs experiments for the tanh/FRN pair. The agreement is very good for tanh/RRN (not shown, but similar to the case of tanh/FRN with $\sigma_v = 1$ and $\sigma_a = 0$) and α-ReLU/FRN as well (see Fig. A.1).

As mentioned in previous sections, we seek to characterize the long term/high depth behavior of all of the quantities defined in Section 2. To do so, we solve for the asymptotics of the recurrences in Table 1, where ϕ is instantiated with tanh or α-ReLU. Our main dynamics results are summarized in Table 2.

Table 2: Summary of Main Dynamics Results. Note that while $\daleth^{(l)}$ is exponential for ReLU/FRN, the gradients with respect to weight parameters have norms (χ_w and χ_v) constant in l (Thm B.21). Also, the $\daleth^{(l)}$ entry for α-ReLU is for $\alpha \in (3/4, 1)$ only

	Tanh/RRN	Tanh/FRN	ReLU/FRN	α-ReLU/FRN, $\alpha < 1$
$p^{(l)}$	$\Theta(l)$, B.2	$\Theta(l)$, B.9	$\exp(\Theta(l))$, B.16	$\Theta(l^{1/(1-\alpha)})$, B.16
$\mathfrak{e}^{(l)} - \mathfrak{e}^*$	$\check{\Theta}(l^{\frac{2}{\pi}-1})$, B.4	$\mathsf{poly}(l)$, B.11	$\Theta(l^{-2})$, B.17	$\mathsf{poly}(l)$, B.18
$\daleth^{(l)}$	$\exp(\Theta(\sqrt{l}))$, B.6	$\exp(\Theta(\sqrt{l}))$, B.12	$\exp(\Theta(l))$, B.20	$\Theta(l^{\frac{\alpha^2}{(1-\alpha)(2\alpha-1)}})$, B.20

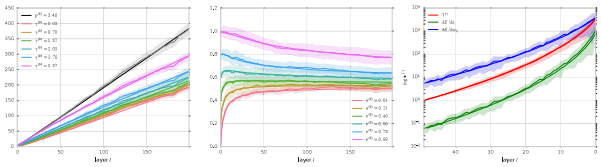

Figure 1: Our equations predict the relevant quantities very well in practice. These plots make the comparison between prediction and measurements for the full resnet with tanh activation, with $\sigma_v^2 = 1.5$, $\sigma_a^2 = .5$, $\sigma_w^2 = 1.69$, $\sigma_b^2 = .49$. Left-to-right: **(a)** $p^{(l)}$ and $\gamma^{(l)}$ against layer l for 200 layers. **(b)** $\mathfrak{e}^{(l)} = \gamma^{(l)}/p^{(l)}$ against l for 200 layers. Both (a) and (b) trace out curves for different initial conditions. **(c)** Different gradient quantities against l for 50 layers. From left to right the layer number l decreases, following the direction of backpropagation. Notice that the gradient increases in norm as $l \to 1$. All three figures exhibit smooth curves, which are theoretical estimates, and irregular curves with shades around them, which indicate empirical means and standard deviations (both of which taken in regular scale, not log scale). (a) and (b) are made with 20 runs of resnets of width 1000. (c) is made with 25 runs of resnets of width 250.

5.1 Tanh

Forward dynamics. When $\phi = \tanh$, $p^{(l)}$ and $q^{(l)}$ increase as $\Theta(l)$ in either RRN or FRN (Thm B.2), as one might expect by observing that $\mathrm{V}\tanh(0, q) \to 1$ as $q \to \infty$ so that, for example in the RRN case, the recurrence $p = \mathrm{V}\tanh(0, q) + p$ becomes $p = 1 + p$. This is confirmed graphically by the black lines of the leftmost chart of Fig. 1. We carefully verify that this intuition is correct in its proof in the appendix, and find that in fact $p^{(l)} \sim l$ in the RRN case and $p^{(l)} \sim (\sigma_v^2 + \sigma_a^2)l$ in the FRN case.

What about $\gamma^{(l)}$? The middle chart of Fig. 1 shows that over time, $\mathfrak{e}^{(l)} = \gamma^{(l)}/p^{(l)}$ contracts toward the center of the interval $[0, 1]$, but from the looks of it, it is not clear whether there is a stable fixed point $\mathfrak{e}^*$ of $\mathfrak{e}$ or not. We prove that, in fact, **all trajectories of $\mathfrak{e}$ not starting at 1 do converge to a single fixed point, but only at a polynomial rate**, in both the RRN and FRN cases (Thm B.2 and Thm B.10); we can even explicitly compute the fixed point and the rate of convergence: For FRN, there is a **unique stable fixed point** $\mathfrak{e}^* < 1$ determined by the equation

$$\mathfrak{e}^* = \frac{1}{\sigma_v^2 + \sigma_a^2}[\sigma_v^2 \frac{2}{\pi} \arcsin(\mathfrak{e}^*) + \sigma_a^2],$$

and $|\mathfrak{e}^* - \mathfrak{e}^{(l)}|$ decreases like $l^{-\delta^*}$, where

$$\delta^* := 1 - \frac{2}{\pi} \frac{1}{\sqrt{1 - (\mathfrak{e}^*)^2}} \frac{\sigma_v^2}{\sigma_v^2 + \sigma_a^2}.$$

The case of RRN can be viewed as a special case of the above, setting $\sigma_v^2 = 1$ and $\sigma_a^2 = 0$, which yields $\mathfrak{e}^* = 0$ and $\delta^* = 1 - \frac{2}{\pi}$. We observe that both $\mathfrak{e}^*$ and δ^* only depend on the ratio $\rho := \sigma_a/\sigma_v$, so in Fig. 2 we graph these two quantities as a function of ρ. $\mathfrak{e}^*$ and δ^* both increase with ρ and asymptotically approach 1 and $1/2$ respectively from below. When $\rho = \sigma_a = 0$, $\mathfrak{e}^* = 0$ and $\delta^* = 1 - \frac{2}{\pi}$. Thus the rate of convergence at its **slowest** for tanh/FRN is $\delta^* = 1 - \frac{2}{\pi} \approx 0.36338$, where asymptotically the network tends toward a **chaotic regime** $\mathfrak{e}^* = 0$, corresponding to a large weight variance and a small bias variance; it at its **fastest** is $\delta^* = 1/2$, where asymptotically the network tends toward a **stable regime** $\mathfrak{e}^* = 1$, corresponding to a large bias variance and small weight variance. We verify δ^* by comparing $\mathfrak{e}^{(l)} - \mathfrak{e}^{(l-1)}$ to $l^{-\delta^*-1}$ in log-log scale. If $\mathfrak{e}^{(l)} = \Theta(l^{-\delta^*})$, then $\mathfrak{e}^{(l)} - \mathfrak{e}^{(l-1)} = \Theta(l^{-\delta^*-1})$ and should obtain the same slope as $l^{-\delta^*-1}$ as $l \to \infty$. The middle figure of Fig. 2 ascertains that this is indeed the case, starting around layer number 400.

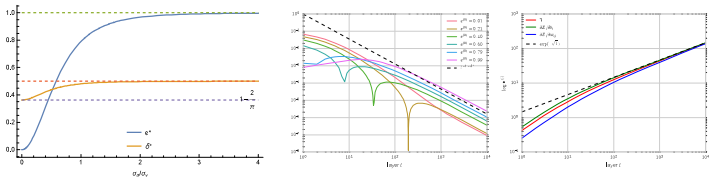

Figure 2: Left-to-right: **(a)** Plots of $\mathfrak{e}^*$ and δ^* against σ_a/σ_v. **(b)** In log-log scale: the dashed line is $l^{-\delta^*-1}$, and the colored lines are $\mathfrak{e}^{(l)} - \mathfrak{e}^{(l-1)}$ for different initial conditions $\mathfrak{e}^{(0)}$. That they become parallel at about $l = 400$ on verifies that $\mathfrak{e}^{(l)} = \Theta(l^{-\delta^*})$. [3] **(c)** In log-log scale: The dashed line is $\mathcal{A}\sqrt{l}$ ($\mathcal{A}$ given in Thm B.13), and the colored lines are $\log(\bullet^{(1)}/\bullet^{(l)})$ for $\bullet = \beth, \chi_b, \chi_w$. That they all converge together starting around $l = 1000$ indicates that the approximation in Thm B.13 is very good for large l.

Backward dynamics. Finally, we show that the gradient is approximated by

$$\beth^{(m)} = \exp(\mathcal{A}(\sqrt{l} - \sqrt{m}) + O(\log l - \log m))\beth^{(l)} \qquad (\star)$$

where $\mathcal{A} = \frac{4}{3}\sqrt{\frac{2}{\pi}}\sigma_w$ in the RRN case and $\mathcal{A} = \frac{4}{3}\sqrt{\frac{2}{\pi}}\frac{\sigma_v^2\sigma_w}{\sqrt{\sigma_v^2+\sigma_a^2}}$ in the FRN case (Thm B.6 and Thm B.13). The rightmost plot of Fig. 2 verifies that indeed, for large $l \geq 1000$, this is a very good approximation. This demonstrates that the mean field assumption of independent backpropagation weights is very practical and convenient even for residual networks.

Note that in the FRN case, the constant $\mathcal{A}$ can be decomposed into $\mathcal{A} = \frac{4}{3}\sqrt{\frac{2}{\pi}} \cdot \sigma_v \cdot \sigma_w \cdot (1 + \sigma_a^2/\sigma_v^2)^{-1/2}$. Consider the ratio $\rho := \sigma_a/\sigma_v$. If $\rho \gg 1$, then $\mathfrak{e}^* \approx 1$ (Fig. C.17), meaning that the typical network essentially computes a constant function, and thus unexpressive; at the same time, large ρ makes $\mathcal{A}$ small, and thus ameliorating the gradient explosion problem, making the network more trainable. On the other hand, if $\rho \ll 1$, then $\mathfrak{e}^* \approx 0$ (Fig. C.17), the typical network can tease out the finest differences between any two input vectors, and a final linear layer on top of such a network should be able to express a wide variety of functions [9]; at the same time, small ρ increases $\mathcal{A}$, worsening the gradient explosion problem, making the network less trainable. This is the same expressivity-trainability tradeoff discussed in [11].

5.2 α-ReLU

Forward dynamics. As with the tanh case, to deduce the asymptotic behavior of random α-ReLU resnets, we need to understand the transforms $\mathrm{V}\psi_\alpha$ and $\mathrm{W}\psi_\alpha$. Fortunately, $\mathrm{V}\psi_\alpha$ has a closed form, and $\mathrm{W}\psi_\alpha$ has been studied before [2]. In particular, if $\alpha > -\frac{1}{2}$, then $\mathrm{V}\psi_\alpha(0, q) = \mathsf{c}_\alpha q^\alpha$, where c_α is a constant with a closed form given by Lemma B.15. In addition, by [2], we know that $\mathrm{W}\psi_\alpha(0, q, q, qc) = \mathrm{V}\psi_\alpha(0, q)\mathbb{J}_\alpha(c)$ for $\mathbb{J}_\alpha$ given in Appendix C.7.1. Fig. C.17 shows a comparison of $\mathbb{J}_\alpha$ for different αs along with the identity function.

Substituting in $\mathsf{c}_\alpha q^\alpha$ for $\mathrm{V}\psi_\alpha$, we get a difference equation $p - \underline{p} = \sigma_v^2\mathsf{c}_\alpha(\sigma_w^2\underline{p} + \sigma_b^2)^\alpha + \sigma_a^2$ governing the evolution of p. This should be reminiscent of the differential equation $\dot{P}(l) = CP(l)^\alpha$, which has solution $\propto l^{1/(1-\alpha)}$ for $\alpha < 1$, and $\propto \exp(Cl)$ when $\alpha = 1$. And indeed, the solutions $p^{(l)}$ to these difference equations behave asymptotically exactly like so (Thm B.16). Thus **ReLU behaves very explosively compared to α-ReLU with $\alpha < 1$.** In fact, in simulations, for $\sigma_w^2 = 1.69$ and $\sigma_v^2 = 1.5$, the ReLU resnets overflows into `infs` after around 100 layers, while there's no problem from any other kind of networks we consider.

Regardless, **α-ReLU for all α massages $\mathfrak{e}^{(l)}$ toward a fixed point $\mathfrak{e}^*$ that depends on α.** When $\phi = \psi_1$, the standard ReLU, $\mathfrak{e}^{(l)}$ converges to 1 asymptotically as Cl^{-2} for an explicit constant C depending on σ_v and σ_w only (Thm B.17). When $\phi = \psi_\alpha$ for $\alpha < 1$, then $\mathfrak{e}^{(l)}$ converges to the nonunit fixed point $\mathfrak{e}^*$ of $\mathbb{J}_\alpha$ at a rate of $\check{\Theta}(l^{-\mu})$, where $\mu = (1 - \dot{\mathbb{J}}_\alpha(\mathfrak{e}^*))/(1 - \alpha)$ is independent of the variances (Thm B.18). These rates are verified in Fig. A.2.

Backward dynamics. Finally, we have also characterized the rate of gradient growth for any $\alpha \in (\frac{3}{4}, 1]$. [4] **In the case of $\alpha = 1$, the dynamics of $\beth$ is exponential**, the same as that of p, $\beth^{(l-m)} = \beth^{(l)}B^m$ where $B = \frac{1}{2}\sigma_v^2\sigma_w^2 + 1$. **For $\alpha \in (\frac{3}{4}, 1)$, the dynamics is polynomial**, but

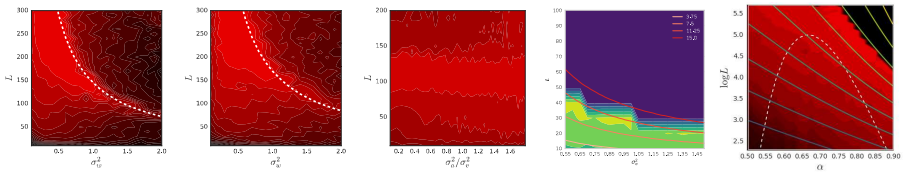

Figure 3: From left to right: **(a)** and **(b)**: σ_w^2, L, and test set accuracy of a grid of tanh reduced (left) and full (right) resnets trained on MNIST. Color indicates performance, with lighter colors indicating higher accuracy on test set. Other than the values on the axes, we have fixed $\sigma_b^2 = \sigma_a^2 = \frac{1}{2}$ and $\sigma_v^2 = 1$. The white dotted lines are given by $\sigma_w^2 L = C$, where $C = 170$ on the left and $C = 145$ on the right. We see that both dotted lines accurately predicts the largest optimal σ_w for each depth L. **(c)** Varying the ratio σ_a^2/σ_v^2 while fixing $\sigma_v/\sqrt{1+\sigma_a^2/\sigma_v^2}$, and thus fixing $\mathcal{A}$, the leading constant of $\log \daleth^{(0)}/\daleth^{(L)}$. **(d)**: Purple-green heatmap gives the test accuracies of ReLU FRN for varying σ_w^2 and L. Red curves give level sets for the log ratios $\log p^{(L)}/p^{(0)} = \log \daleth^{(0)}/\daleth^{(L)} = L\log(1+\sigma_v^2\sigma_w^2/2)$. **(e)** Red heatmap shows the test accuracies of a grid of α-ReLU FRN with varying α and L as shown, but with all $\sigma_\bullet$s fixed. The white dashed curve gives a typical contour line of $L^R = \text{const}$, where $R = \frac{\alpha^2}{(1-\alpha)(2\alpha-1)}$. The yellow-to-blue curves form a set of level curves for $p^{(l)} - \gamma^{(l)} = \text{const}$, with yellow curves corresponding to higher levels.

with different exponent in general from that of the forward pass: $\daleth^{(l-m)} = \Theta(1)\daleth^{(l)}(l/(l-m))^R$ for $R = \frac{\alpha^2}{(1-\alpha)(2\alpha-1)}$, where the constants in $\Theta(1)$ do not depend on l or m. This exponent R is minimized on $\alpha \in [\frac{3}{4}, 1)$ at $\alpha = 3/4$, where $R = 9/2$ (but on $\alpha \in (\frac{1}{2}, 1)$ it is minimized at $\alpha = 2/3$, where $R = 4$); see Fig. B.8. These exponents are verified empirically in Fig. A.2.

Looking only at $\daleth$ and the gradients against the biases, it seems that ReLU suffers from a dramatic case of exploding gradients. But in fact, because $\daleth$ gains a factor of B moving backwards while p loses a factor of B, the gradient norm $\chi_w^{(l-m)}$ (and similarly for $\chi_v^{(l-m)}$) is independent of how far, m, the gradient has been propagated (Thm B.21) — this is certainly the best gradient preservation among all of the models considered in this paper. Thus strangely, random ReLU FRN exhibits both the best (constant for v and w) and the worse (exponential for a and b) gradient dynamics. This begs the question, then, is this a better deal than other α-ReLU for which for any learnable parameter we have at most a polynomial blowup with depth in its gradient? Our experiments (discussed below) show that α-ReLU is useful to the extent that smaller α avoids numerical issues with exponentiating forward and backward dynamics, but the best performance is given by the largest α that avoids them (Fig. 3(c, d)); in fact, an expressivity quantity, $p^{(l)} - \gamma^{(l)}$, determines performance, not gradient explosion (see α-ReLU experiments).

6 Experimental Results

Schoenholz et al. [11]'s experiments suggest that

> *Similar levels of gradient explosion/vanishing at initialization time induce similar levels of performance at test time.* ($\diamondsuit$)

Our experiments show that this hypothesis is true most of the time, but in the sole exception, it is the expressivity quantity $p^{(l)} - \gamma^{(l)}$ that determines performance.

Tanh, vary σ_w. We train a grid of reduced and full tanh resnets on MNIST, varying the variance σ_w^2 and the number of layers (for FRN we fix $\sigma_v = 1$). The results are indicated in Fig. 3(a, b). We see that in either model, deeper resnets favor much smaller σ_w than shallower ones. The white dotted lines in Fig. 3(a, b) confirm the predictions of $\diamondsuit$: according to Eq. ($\star$), for the same gradient ratio $R = \daleth^{(0)}/\daleth^{(L)}$, we want $\log R \approx \sigma_w\sqrt{L}$. Indeed, the white dotted lines in Fig. 3(a, b) trace out such a level curve and it remarkably pinpoints the largest σ_w that gives the optimal test set accuracy for each depth L. This suggests the following procedure for finding optimal initialization parameters for Tanh-residual networks: train an ensemble of shallow networks quickly with depth L', and find σ_w' with the best performance. Compute $C := \sigma_w'\sqrt{L'}$. When training deep networks with depth $L > L'$, initialize with $\sigma_w' := C/\sqrt{L}$.

Tanh, vary σ_a^2/σ_v^2. As suggested in the analysis of Eq. ($\star$), the ratio $\rho^2 = \sigma_a^2/\sigma_v^2$ determines the fixed point $\mathfrak{e}^*$ and its convergence rate by itself while also contributes to the rate of gradient explosion in tanh FRN. We seek to isolate its effect on forward dynamics by varying σ_v with ρ such that $\sigma_v/\sqrt{1+\rho^2}$ is kept constant, so that the leading term of the log gradient ratio is kept approximately equal for each L and ρ. Fig. 3(c) shows the test accuracies of a grid of tanh FRN initialized with such an ensemble of $\sigma_\bullet$s. What stands out the most is that performance is maximized essentially around a fixed value of L regardless of ρ, adding another piece of major evidence toward $\diamondsuit$. There is also a minor increase in performance with increasing ρ regardless of L; this is counterintuitive as increasing ρ means "decreasing expressivity." It is currently not clear what accounts for this effect.

ReLU, vary σ_w We train a grid of ReLU FRN on MNIST, varying $\sigma_w^2 \in [.55, 1.5]$ while fixing $\sigma_v^2 = 1, \sigma_a^2 = \sigma_b^2 = \frac{1}{2}$. The resulting test set accuracies are shown in Fig. 3(d). The dark upper region signifies failure of training caused by numerical issues with exploding activation and gradient norms. We see that the best test accuracies are given by depths just below where the numerical issues occur. Our theory predicts (the ordering of) accuracies very well under the assumption that networks with similar levels of gradient explosion should perform similarly: The red lines give contours of the gradient ratio $\gimel^{(0)}/\gimel^{(L)}$ (which is also the activation norm ratio $p^{(L)}/p^{(0)}$), and they track the level sets of accuracies. It seems that a ratio of $\exp(11.25)$ is optimal.

α-ReLU, vary α. We similarly trained a grid of α-ReLU FRN on MNIST, varying only α and the depth, fixing all $\sigma_\bullet$. Fig. 3(e) shows their test accuracies. We see similar behavior to ReLU, where when the net is too deep, numerical issues doom the training (black upper right corner), but the best performance is given by L just below where this problem occurs. Contrary to $\diamondsuit$, gradient ratio contours (white dashed line) do not predict the performances at all, but the gap $p^{(l)} - \gamma^{(l)}$ does, remarkably well.

In all of our experiments, we did not find $\mathfrak{e}$ dynamics to be predictive in neural network performance. Our α-ReLU experiment suggests that we should replace $\mathfrak{e} = \gamma/p$ with $p - \gamma$ as the central expressivity quantity.

7 Conclusion

In this paper, we have extended the mean field formalism developed by [9, 10, 11] to residual networks, a class of models closer to practice than classical feedforward neural networks as were investigated earlier. We proved and verified that in both the forward and backward passes, most of the residual networks discussed here do not collapse their input space geometry or the gradient information exponentially. We found our theory incredibly predictive of test time performance despite saying nothing about the dynamics of training. We in particular found that random tanh residual networks seem to perform best when the product $\sigma_w\sqrt{l}$ hovers around a constant value. In addition, we overwhelmingly find, through theory and experiments, that an optimal initialization scheme must take into account the depth of the residual network. The reason that Xavier [4] or He [5] scheme are not the best is in fact not that their statistical assumptions are fragile — theirs are similar to our mean field theoretic assumptions, and they hold up in experiments for large width — but rather that their structural assumptions on the network break very badly on residual nets.

Open Problems. Our work thus have shown that optimality of initialization schemes can be very unstable with respect to architecture. We hope this work will form a foundation toward a mathematically grounded initialization scheme for state-of-the-art architectures like the original He et al. residual network. To do so, there are still two major components left to study out of the following three: 1. Residual/skip connection 2. Batchnorm 3. Convolutional layers. Recurrent architectures and attention mechanisms are also still mostly unexplored in terms of mean field theory. Furthermore, many theoretical questions still yet to be resolved; the most important with regard to mean field theory is: why can we make Axioms 3.1 and 3.2 and still be able to make accurate predictions? We hope to make progress on these problems in the future and encourage readers to take part in this effort.