A Mean-Field Theory of Training Deep Neural Networks

Raj Pabari
Stanford University
rajpabari@stanford.edu

Iris Zhou
Stanford University
iriszhou@stanford.edu

Abstract

Using concepts from statistical physics, we study signal propagation in deep neural networks with random weight and bias initialization, without the use of any architectural tricks to maintain gradient flow such as batch normalization or residual connections. First, we present a theoretical mean-field framework to study the depth scales of a fully-connected neural network as a function of the architecture. We provide an order-chaos phase transition and show that information can only propagate when the network architecture is in the ordered phase. Unfortunately, the mean-field literature has been largely contextualized to the $\tanh$ *activation function due to its nice properties; we prove in Theorem 1 that it does not generalize to the sigmoid activation function, which exhibits significantly lower trainability. However, we empirically verify that it does generalize to functions like* $\arctan$ *and softsign, and surprisingly, ReLU, which is unbounded.*

Next, we take a detour into dynamical isometry, a concept from random spectral matrix theory that demonstrates why orthogonal initialization is preferred to Gaussian initialization. We then present an extension of the mean-field framework to analyze depth scales of convolutional layers. As expected, the depth scales remain largely the same, but convolutional neural networks exhibit better much better performance on test sets. We empirically verify our theoretical predictions, investigating the independence of the depth scale to the choice of dataset and the learning with varying degrees dynamical isometry. Together, these results paint a fairly complete theoretical prediction of the effect of architecture and initialization on training deep vanilla neural networks with empirical verification to merit.

1. Introduction

Deep neural networks have exhibited tremendous success in recent years. However, the deeper the network, the harder it is for information about the dataset to be preserved as it propagates through each layer. For instance, deep neural networks often encounter problems with vanishing or exploding gradients, leading to an inability to properly update weights and biases. In order to maintain gradient flow in deep neural networks, techniques such as batch normalization [5], layer normalization [1], gradient clipping [10], and residual connections [4] help explicitly maintain gradient flow throughout deep neural networks.

However, inspired by mean field theory from statistical physics, it has been shown that specific initialization schemes for the weights and biases can implicitly maintain gradient flow. Such initialization renders even the deepest neural networks trainable without any other changes to architecture [12, 13]. Drawing on the theory of dynamical isometry, this mean field theory has been extended to more modern architectures, such as convolutional layers and transformers [15, 2]. Surprisingly, these works find an order-to-chaos phase transition as a function of the weight variance and the depth; there is a sharp phase boundary on one side of which the model is able to learn with very high accuracy and the other side of which the model does no better than random. The mean field theory allows us to approximate the phase boundary with a "depth scale" that intuitively tells us how far information can be propagated through the network.

In particular, Schoenholz *et al*. [13] shows that fully-connected neural networks become more trainable when the initial weights and biases are sampled from a Gaussian distribution with mean zero and variance within a certain depth scale. Additionally, Pennington *et al*. [11] takes inspiration from random matrix theory to show that orthogonal initialization schemes are able to learn faster than Gaussian initialization schemes. Xiao *et al*. and Cowsik *et al*. [15, 2] then combine these two results to create initialization schemes for convolutional neural networks and transformers, respectively.

Clearly, the mean-field framework has been used to explained the learning dynamics of many different architectures. However, in our paper, we seek to test its robustness. For instance, although the analysis of Schoenholz *et al*. is general enough to fit a fully-connected network with any activation function, their experiments focus almost excluively on the $\tanh$ activation function. Therefore, we test if their

theoretical framework can extend to other activation functions that are commonly used in practice, such as sigmoid and ReLU, which are not centered. On face from what has been shown thus far in the literature, it's not at all obvious if the mean field theory generalizes upon relaxing the assumptions of having a bounded, mean 0 activation function. Additionally, we attempt to obtain experimental results on different datasets, testing on CIFAR-10 [8] in addition to MNIST [3], verifying empirically that the framework is independent of the dataset.

We also explore the differences in Gaussian and orthogonal initialization, and use that to motivate our exploration of convolutional layers, a more complicated and modern architecture that is actually used in practice. In particular, the work of Xiao *et al*. [15] only demonstrates the viability of the framework on convolution-only architecture. We attempt to unify the fully-connected framework [13] with the convolutional framework to produce an initialization scheme for an architecture that alternates between convolutional and fully-connected layers. Though they present a theoretical result about dynamical isometry that motivates their Delta-Orthogonal initialization scheme for convolutional kernels, they don't provide empirical results comparing iid Gaussian weights; to make the theory more robust, we examine these learning dynamics.

2. Related Work

Currently, to solve the problem of vanishing and exploding gradients, the most common techniques are batch normalization [5], layer normalization [1], residual connections [4] and gradient clipping [10]. Both batch and layer normalization attempt to renormalize the data after each layer to be (roughly) distributed like a standard Gaussian to maintain gradient flow. Residual connections add the input of each layer to the output of a subsequent layer to facilitate gradient flow, and gradient clipping, as the name suggests, simply limits gradient values to be within a specified range. Although these work well in practice, they all serve as a somewhat artificial solution to the problem of vanishing and exploding gradients, and require additional computational capacity, which often serves as a limiting factor. Because of this, initialization techniques which directly enable backpropagation without additional computational cost, including the ones explored in this paper, are often preferred.

Mean field theory is an enormously successful analysis technique from statistical physics that approximates the dynamics of complex systems of interacting particles with a more analytically tractable system [7]. In a common example, the Ising model of interacting spins, the more analytically tractable system is one where the inter-spin interactions are neglected and are absorbed into an external applied magnetic field instead.

Poole *et al*. [12] provides a modern foundation for a mean field theory of deep neural networks, though the idea dates back to the 1980s [14]. They models the propagation of each input in a deep neural network as the motion of a particle throughout time, as it interacts with other particles. Following the standard physical mean field approach, they obtain an iterative map for the variance of a single input after each layer and another for the correlation between two inputs after each layer. After showing that these iterative maps each converge to a fixed point, they provides a criterion for the order-chaos phase transition which they conjecture renders deep neural networks trainable.

This is picked up by Schoenholz *et al*. [13], which uses it to study the depth scales of fully-connected neural networks. This paper explores gradient backpropagation in addition to the forward propagation analyzed in [12], and finds critical exponents to theoretically predict depth scales. Additionally, this paper empirically analyzes the $\tanh$ activation function by training a fully-connected neural net using Stochastic Gradient Descent (SGD) and RMSProp on the MNIST [3] and CIFAR10 [8] datasets. They produce depth scale and trainability plots which closely match their theoretical predictions. Unfortunately, although the theoretical formulation is meant to be generalized to all bounded activation functions, the authors of this paper do not empirically examine any functions other than $\tanh$. Although the authors do make an attempt at investigating more complicated architectures by showing that dropout inhibits trainability, they were not able to generalize their findings to more commonly used architectures. Since there is no publicly available code provided by the authors of this paper, we used the (unofficial) implementation given in [9] as a starting point.

The paper by Pennington *et al*. [11] attempts to expand the mean field formalism to other common initialization techniques. They draw on the concept of dynamical isometry from spectral random matrix theory, this being that the spectrum of the Jacobian concentrates about 1. With dynamical isometry, they show that initializing weights as an orthogonal matrix (as opposed to having entries be iid Gaussian), improves the trainability of a deep neural network. In particular, with orthogonal initialization, the number of epochs necessary for the network to converge scales sublinearly with depth instead of linearly. Interestingly, the authors show that in contrast to sigmoidal networks such as $\tanh$, sigmoid, etc., the ReLU activation function is unable to achieve dynamical isometry, meaning that it benefits less from the orthogonal initialization. This matches their empirical results when they train on CIFAR10 using SGD, SGD with Momentum, Adam and RMSProp. In this paper, more activation functions than simply $\tanh$ are explored, but this paper still only focuses on the most basic deep neural networks with only linear layers.

However, Xiao *et al*. [15] is able to generalize this framework to convolutional layers. Building off both the mean

field formalism for information propagation, and the idea of achieving dynamical isometry using orthogonal initialization, the authors are able to train a vanilla CNN with 10,000 layers using Delta-Orthogonal Initialization. They find a similar order-chaos phase transition, as in the fully-connected case, which determines the trainability of the network at various depths. However, they demonstrate that convolution layers with iid weights cannot achieve dynamical isometry, which again motivates orthogonal initialization. Empirically, the authors are able to achieve testing accuracies of 99% and 82% on MNIST and CIFAR10, respectively. Most importantly, this paper, unlike the previous ones, is able to extend the mean-field formalism to an architecture that is commonly used in practice. For the implementation of Delta-Orthogonal Initialization, we referred to [6] along with the authors' official implementation. In fact, a recent paper by Cowsik *et al*. [2] is able to further extend the mean field theory framework to another modern architecture of transformers, and another of Yang and Schoenholz [16] applies the framework to residual networks.

3. Theoretical Results and Formulation

3.1. Fully-Connected Layers

For fully-connected layers, we review the mean-field formulation from Poole *et al*. and Schoenholz *et al*. [12, 13]. Consider a deep fully-connected neural network with depth D and width N_l at each layer l. Let $W^1, ..., W^D$ denote the weight matrices, $b^1, ..., b^D$ denote the bias vectors, and φ denote the activation function. Suppose that the initial weights and biases are sampled from Gaussian distributions with mean zero, and variance σ_w^2/N_l and σ_b^2, respectively. The factor of $1/N_l$ in the weight variance is to ensure that the input to an individual neuron of the $l+1$ layer remains $O(1)$, independent of the width N_l. Inputs are propagated through the network by the pair of equations

$$z^l = W^l y^l + b^l \qquad y^{l+1} = \varphi(z^l) \tag{1}$$

with y^0 denoting the input to the first layer. Since the weights and biases are random variables, z_i^l and y_i^l are random variables as well. Here, we assume the "mean-field approximation" by supposing that the z_i^l are Gaussian with mean $\langle z_i^l \rangle$ and variance $\langle (z_i^l - \langle z_i^l \rangle)^2 \rangle$. Let y^0 be some arbitrary input, then since the weights and biases are iid with mean zero, the first two moments are given by

$$\langle z_i^l \rangle = 0 \qquad \langle z_i^l z_j^l \rangle = q^l \delta_{ij} \tag{2}$$

where δ_{ij} denotes the Kronecker delta, and q^l is the variance of the z^l. As derived in Poole *et al*., this is given by

$$q^l = \sigma_w^2 \int \mathcal{D}z \varphi(\sqrt{q^{l-1}} z) + \sigma_b^2 \tag{3}$$

where $\mathcal{D}z \equiv \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$ denotes the standard Gaussian measure. The initial condition is $q^0 = \frac{1}{N_0}\|y^0\|_2^2$ and $q^1 = \sigma_w^2 q^0 + \sigma_b^2$. If φ is bounded, then for any σ_w^2 and σ_b^2, 3 has a well-defined fixed point, $q^* = \lim_{l\to\infty} q^l$.

Now suppose that there is a pair of arbitrary inputs x^0 and y^0, then again since the weights and biases are iid with mean zero, the covariance at each layer is given by

$$\langle z_{i,x}^l z_{i,y}^l \rangle = q_{xy}^l \delta_{ij} \tag{4}$$

where again, as derived in Poole *et al*., q_{xy}^l is described by the recurrence

$$q_{xy}^l = \sigma_w^2 \int \mathcal{D}z_1 \mathcal{D}z_2 \varphi(u_1)\varphi(u_2) + \sigma_b^2 \tag{5}$$

$$u_1 = \sqrt{q_{xx}^{l-1}} z_1 \tag{6}$$

$$u_2 = \sqrt{q_{yy}^{l-1}} \left[c_{xy}^{l-1} z_1 + \sqrt{1 - (c_{xy}^{l-1})^2} z_2 \right] \tag{7}$$

where $c_{xy}^l = q_{xy}^l / \sqrt{q_{xx}^l q_{yy}^l}$ is the correlation between two inputs at layer l. Again $\mathcal{D}z_1$ and $\mathcal{D}z_2$ denote the standard Gaussian measure, and note that u_1 and u_2 are correlated approximations for the z^{l-1} with the correct covariance matrix. Examining (5), it is clear that $c^* = \lim_{l\to\infty} c_{xy}^l = 1$ is a fixed point. To determine whether the $c^* = 1$ fixed point is stable or not, Poole *et al*. computes the susceptibility

$$\chi_1 \equiv \frac{\partial c_{xy}^l}{\partial c_{xy}^{l-1}} = \sigma_w^2 \int \mathcal{D}z [\varphi'(\sqrt{q^*} z)]^2 \tag{8}$$

and deduces that $c^* = 1$ is stable if $\chi_1 < 1$ and unstable otherwise. This gives the critical line $\chi_1 = 1$ which separates the ordered phase, where $c^* = 1$ we achieve correlation of inputs in the limit, and the chaotic phase where inputs are decorrelated in the limit.

We can continue by examining the dynamics of q^l, c_{xy}^l near the fixed points. Indeed, continuing with our mean-field inspired analysis, we conjecture that there exists some critical exponents ξ_q, ξ_c such that near the critical point, $|q^l - q^*| \approx \exp\left(-\frac{l}{\xi_q}\right)$ and $|c_{xy}^l - c^*| \approx \exp\left(-\frac{l}{\xi_c}\right)$. These depth scales are derived in detail in Schoenholz *et al*. [13]; note that we use the assumption of being near the fixed point with a Taylor expansion and keeping terms up to lowest order only. We find that

$$\xi_q^{-1} = -\log\left(\chi_1 + \sigma_w^2 \int \mathcal{D}z \varphi''(\sqrt{q^*} z)\phi(\sqrt{q^*} z)\right) \tag{9}$$

$$\xi_c^{-1} = -\log\left(\sigma_w^2 \int \mathcal{D}z_1 \mathcal{D}z_2 \varphi''(u_1^*)\phi(u_2^*)\right) \tag{10}$$

where naturally, u_1^*, u_2^* are equivalent to u_1, u_2 from before with $q_{xx}^{l-1} = q^* = q_{yy}^{l-1}$ and $c_{xy}^{l-1} = c^*$. Fitting the exponential approximations with these depth scales sees a good

match with empirical measurements for the variance and correlation. In our experiments, we use this framework to examine the trainability of various fully-connected layers, including those integrated into other architecture.

3.2. Dynamical Isometry

Now that we have a condition for trainability, a natural next question is to consider the speed of convergence. Again analyzing fully-connected layers with the same setup as the previous section, [11] considers the input-output Jacobian matrix

$$J = \frac{\partial y^D}{\partial z^0} = \prod_{l=1}^{D} A^l W^l \tag{11}$$

where A^l is a diagonal matrix with random entries $A^l_{ij} = \delta_{ij}\varphi'(z^l_i)$ and W^l is a random weight matrix. First, note that, recall that the order to chaos phase transition occurs when $\chi_1 = 1$, and we can express

$$\chi_1 = \frac{1}{N}\langle \mathrm{Tr}(AW)^T AW\rangle = \sigma_w^2 \int \mathcal{D}z[\varphi'(\sqrt{q^*}z)]^2 \tag{12}$$

where A and W are the corresponding random matrices to A^l and W^l for when the weight variance is set to the fixed point q^*. Through this interpretation, [11] χ_1^D becomes the mean squared singular value of J, so our task reduces to computing the spectral density (probability density of singular values) of J. Intuitively, the more tightly concentrated the spectral density is about 1, the "sharper" the phase transition at $\chi_1 = 1$ is, rendering the neural network easier to train.

The term perfect dynamical isometry is used to describe a random matrix with spectral density $\rho(\lambda) = \delta(\lambda - 1)$, so we can equivalently state that more dynamical isometry that a neural network architecture has, the easier it will be to train. Interestingly, [11] proves that ReLU cannot achieve perfect dynamical isometry, and nor can architectures with weight matrices sampled iid from a Gaussian, but sigmoidal activation functions (eg. $\tanh$ and sigmoid) with orthogonal initialization can.

3.3. Convolutional Layers

To this point, we have considered only vanilla feedforward fully-connected neural networks, but we hope to extend the mean-field theory to more modern architectures. For the vision tasks considered herein, a common architectural choice is convolutional neural networks; hence we extend our analysis to include convolutional kernels with inspiration from Xiao *et al*. [15]. Consider a CNN with D layers, each applying 1D convolutions along two different dimensions, with periodic boundary conditions, and using kernel size $2k + 1$, spatial size n and c channels. Let φ denote the activation function, $\omega^l \in \mathbb{R}^{(2k+1)\times c\times c}$ denote the weight tensor, and $b^l \in \mathbb{R}^c$ denote the biases, for $l = 1, ..., D$. If $h^l_j(\alpha)$ denotes the pre-activation at layer l, channel j, and spatial location $\alpha \in \{1, ..., n\}$, then inputs are propagated through the network by the equation

$$h^{l+1}_j(\alpha) = \sum_{\substack{\text{channels } i\\ \text{kernel index } \beta}} \varphi(h^l_i(\alpha+\beta))\omega^{l+1}_{ij}(\beta) + b^{l+1}_j \tag{13}$$

where $\beta \in \mathbb{Z}$ such that $|\beta| \le k$. As in the fully-connected case, suppose that the weights and biases are sampled from Gaussian distributions with mean zero, and variance $\sigma_w^2/(c(2k+1))$ and σ_b^2, respectively. As before, we make the "mean-field approximation" by supposing that h^l_j are Gaussian with mean zero with covariance $\langle h^l_j(\alpha)h^l_j(\alpha')\rangle$, where the average is taken over the weights and biases. [15] shows that the covariance matrix at each layer $l+1$ can be expressed as

$$\Sigma^{l+1} = A \star \mathcal{C}(\Sigma^l) \tag{14}$$

with the following definitions

$$A = \frac{1}{2k+1} I_{2k+1} \tag{15}$$

$$[\mathcal{C}(\Sigma)]_{\alpha,\alpha'} = \sigma_w^2 \mathbb{E}_{h\sim\mathcal{N}(0,\Sigma)}[\varphi(h_\alpha)\varphi(h_{\alpha'})] + \sigma_b^2 \tag{16}$$

$$[A \star B]_{\alpha,\alpha'} = \frac{1}{2k+1}\sum_{\text{kernel index}\beta} B_{\alpha+\beta,\alpha'+\beta} \tag{17}$$

Then, similar to the fully-connected case, 14 has a fixed point (ie a point $\Sigma^* = A \star \mathcal{C}(\Sigma^*)$) given by

$$\Sigma^*_{\alpha\alpha'} = q^*(\delta_{\alpha\alpha'} + (1 - \delta_{\alpha\alpha'})c^*) \tag{18}$$

where c^* is the fixed point defined in the fully-connected case.

To analyze the dynamics near the fixed point Σ^*, we again compute the susceptibility by finding partial derivatives, similar to how χ_1 was computed in the fully-connected case. As shown in [15], the Jacobian of the $\mathcal{C}$-map in 16, evaluated at the fixed point Σ^*, decomposes into a diagonal eigenspace with eigenvalue χ_{q^*} and an off-diagonal eigenspace with eigenvalue χ_{c^*}, where

$$\chi_{q^*} = \sigma_w^2 \mathbb{E}_{h\sim\mathcal{N}(0,\Sigma^*)}[\varphi''(h_1)\varphi(h_1) + \varphi'(h_1)^2] \tag{19}$$

$$\chi_{c^*} = \sigma_w^2 \mathbb{E}_{h\sim\mathcal{N}(0,\Sigma^*)}[\varphi'(h_1)\varphi'(h_2)],\ h_1 \neq h_2 \tag{20}$$

Essentially, small perturbations in q^* and c^* affect Σ^* independently. This is formalized in [15] by moving to a Fourier basis, and showing that $|\Sigma^l - \Sigma^*|$ can be separated into two independently evolving Fourier modes. From this, one can quickly conclude that, exactly like the fully-connected case, the fixed point Σ^* is stable if and only if $\chi_{q^*} < 1$ and $\chi_{c^*} < 1$, meaning that $\chi_1 = 1$ still represents the critical line between the ordered and chaotic phases, and the

desired condition for neither exploding nor vanishing gradients. Again using the approximation of dropping higher order Taylor coefficients near the fixed point, we find that the depth scales of the different Fourier modes and the fixed point c^* are given by

$$\xi_{\alpha,\alpha'}^{-1} = -\log(\chi_{c^*}\lambda_{\alpha,\alpha'}) \tag{21}$$

$$\xi_{c^*}^{-1} = -\log\chi_{c^*} \tag{22}$$

where $\lambda_{\alpha,\alpha'}$ are the eigenvalues of A. As in the fully-connected scenario, empirical measurements for trainability seem to match these theoretical depth scales.

However, as exhibited in the discussion regarding dynamical isometry, the condition $\chi_1 = 1$ is not always a sufficient condition for trainability of deep networks. In particular, we again desire the singular values of the input-output Jacobian matrix J to be close to 1. Noting that convolution is still a linear operator, we are able to adopt the analysis given in [11] and explained in the previous section. In particular, by again writing J in the same form as 11, we note that both A^l and W^l should be close to orthogonal for $l = 1, ..., D$, in order to achieve dynamical isometry. For A^l, this can be done by choosing the appropriate value of q^* and appropriate activation function. However, for W^l, [15] shows that if the weights are initialized as iid Gaussian, then its singular value distribution converges to the Marcenko-Pastur distribution, meaning that it cannot achieve dynamical isometry. Therefore, it is again necessary to utilize orthogonal initialization.

Since there are multiple depth scales, one for each Fourier mode, we need to find a variance vector such that the depth scale $\xi_{\alpha,\alpha'}$ diverges for all modes. Indeed, [15] shows that there exists one such vector which makes all eigenvalues $\lambda_{\alpha,\alpha'}$ equal to 1, meaning that information can propagate along all Fourier modes. Combining this variance vector with orthogonal initialization yields the Delta-Orthogonal initialization scheme [15]. We employ this in our CNN architecture to empirically validate the theoretical hypothesis that deep convolutional networks become more trainable at the appropriate depth scale, and learn in fewer epochs using orthogonal initialization.

4. Empirical Results

In this section, we seek to empirically verify the theoretical results of the previous section that predict the trainability and learning dynamics of deep neural networks.

4.1. Dataset, Hyperparameters, and Architectures

For all of our results, we train on the MNIST and CIFAR10 datasets with Stochastic Gradient Descent (SGD) and a negative log likelihood loss function. For the MNIST dataset, we normalize the data to be in the interval $[-0.5, 0.5]$ for activation functions whose range is in $[-1, 1]$, including ReLU, and we normalize the data to be in the interval $[0.25, 0.75]$ for activation functions in the range $[0, 1]$, to avoid unnecessary issues with vanishing gradients. We use the following default hyperparameters: a learning rate of 1×10^{-3}, a batch size of 256, 180 steps per epoch.

For our architecture, we used a constant width of 300, as in [13], for each fully-connected network, with depth D layers in total. For convolutional layers, we follow the lead of [15] and first increase the number of channels to 256 with a $3 \times 3 \times 256$ convolutional layer. From there, we apply two $3 \times 3 \times 256$ convolutional layers with stride of 2 to downsample the image to 7×7 for MNIST, 8×8 for CIFAR-10. We conclude by applying D $3 \times 3 \times 256$ convolutional layers with stride 1. At the end, we applied an average pooling layer. For both architectures, we append a projection matrix at the end to output unnormalized logits for each class.

In terms of code, we used the existing codebases from [9] and [6] for the implementation of the fully-connected depth scale plots, including the Delta-Orthogonal initialization for CNNs. However, [9] only used the MNIST dataset and the $\tanh$ activation function, so we extended this code to CIFAR10 and other activation functions. We also implemented code to run various experiments on training and test sets, analyze the fixed points c^* and q^* as an intermediate result for generating the ξ_c plots, and to plot convergence rate for Gaussian and orthogonal initializations. The convolutional architecture, and integration of convolutional and fully-connected layers, was largely our code.

4.2. Different Activation Functions

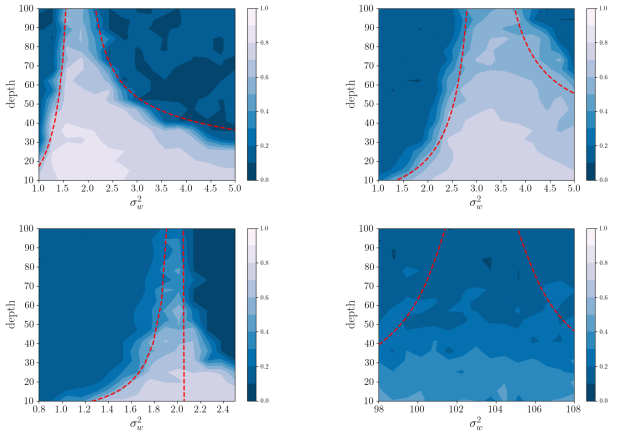

Figure 1. Empirical trainability of neural networks as a function of weight variance σ_w^2 and depth, with multiple of depth scale ($4.5\xi_c$), theoretical prediction, overlaid on top. Top Left: arctan, Top Right: softsign, Bottom Left: ReLU, Bottom Right: sigmoid.

First, to make the mean field framework more robust to different architectures, we explored various activation functions on the MNIST dataset in one epoch. Previous works focus heavily on the $\tanh$ activation function, and in fact

the theoretical analysis of [13] seems to only be valid for bounded, mean zero activation functions.

As a baseline, we attempted to reproduce the results for $\tanh$ found in [13], using $\sigma_b^2 = 0.05$ and plot it in the left side of Figure 2. We also worked with similar activation functions that are also bounded and symmetric around zero keeping the σ_b^2 the same, such as $\arctan$ and softsign, which is given by $f(x) = \frac{x}{|x|+1}$. Due to the similarity of these three functions, it is not a surprise that they produced similar results, with nearly identical weight and bias variances and depth scales, as seen in the top of Figure 1.

Next, as a preliminary test, we analyzed the sigmoid nonlinearity $\varphi(x) = \frac{1}{1+e^{-x}}$, since it is closely related to the $\tanh$ activation function via $\varphi(x) = \frac{\tanh(x/2)+1}{2}$. However, notice that $\varphi(x) \in [0, 1]$, so it is symmetric around 0.5, whereas $\tanh(x) \in [-1, 1]$, so it is symmetric around 0. Thus, despite the two activation functions having the same functional form, we hypothesized the depth scales and learning dynamics of sigmoid to be distinct from those of $\tanh$, with the nonzero mean of sigmoid worsening gradient flow. Indeed, in our empirical experiments, we found that sigmoid performed much worse than $\tanh$, barely better than 10% on MNIST, which demonstrates that it is more or less guessing. Indeed, we have that

Theorem 1 *For the sigmoid activation function* $\varphi(x) = \frac{1}{1+e^{-x}}$, *if* $\chi_1 = 1$, *then* $\sigma_w^2 \gtrsim 100$, *for any* $\sigma_b^2 \in [0, \infty)$.

The full proof of Theorem 1 is given in the appendix. In order to enable the model to train, we set $\sigma_b^2 = 0.09$ (instead of exactly zero). As demonstrated in the bottom right of Figure 1, sigmoid ended up performing very poorly on MNIST near ξ_c, because the predictions of the theory yield absurdly high weight variance for training a neural network.

From an intuitive perspective, this is expected, since if the weight variance σ_w^2 is too high, it essentially becomes random noise. This can lead to numerically unstable gradients and poor convergence. Clearly, the case of sigmoid demonstrates that the mean-field framework does not extend to all bounded activation functions, like [13] claims. Although the depth scales still exist for activation functions like sigmoid, which are not mean zero, the resulting weight and bias variances are simply unfeasible for any neural network to learn from.

Finally, although [13] only claims that their framework works on bounded activation functions, we also investigated ReLU, which is unbounded. The analysis of [13] requires boundedness to show the existence of a fixed point, since otherwise the integral may diverge. However, although this is a sufficient condition, it is not always necessary. Our empirical fixed point analysis demonstrated the existence of q^* and c^* for ReLU. Setting $\sigma_b^2 = 2.01 \times 10^{-5}$, we produced the depth scale plot in the bottom left of Figure 1. Although ReLU did not perform as well as $\tanh$, the mean-field framework still extended quite well, enabling us to find a viable initialization scheme for information propogation through many layers.

4.3. Generalization to Different Datasets

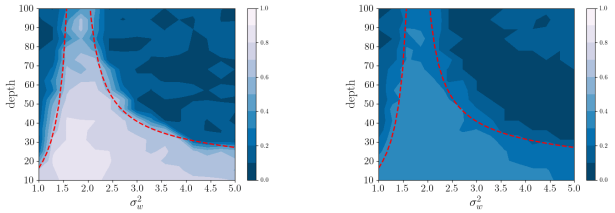

Figure 2. Empirical trainability after 5 epochs of neural networks with $\tanh$ activation as a function of weight variance σ_w^2 and depth, with multiple of depth scale $(4.5\xi_c)$, theoretical prediction, overlaid on top. Left: MNIST, Right: CIFAR-10.

An important feature of the theoretical analysis of Section 3 is that it predicts trainability purely as a function of the architecture and is independent of the choice of dataset. However, the analysis of [13] focuses almost exclusively on the fairly simple MNIST task. Thus, we show in Figure 2 that the depth scale generalizes to the more complicated CIFAR-10 task as well, as predicted by the theoretical analysis. This empirical verification of the theoretical prediction finding makes the results far more compelling and generalizable across tasks.

4.4. Dynamics of Orthogonal Initialization

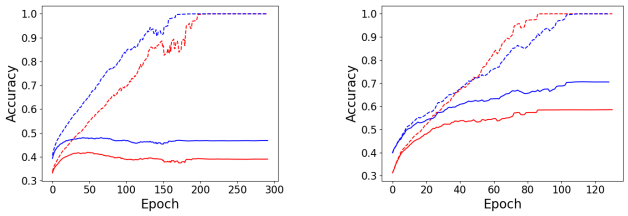

Figure 3. Learning dynamics with respect to dynamical isometry on CIFAR-10. Left: fully-connected, Right: convolutional. Red curves have Gaussian initialization and blue have orthogonal initialization; dotted lines denote test accuracy and solid lines denote training accuracy. Smoothed with 10 epoch moving average.

As explained in Section 3, we expect to see that within the ordered phase, the greater a degree of dynamical isometry a neural network arhchitecture exhibits, the more quickly it will converge. However, [11] only empirically tests this in the context of fully-connected networks on the MNIST task, and [15] only theorizes about this. Thus, to inform our future architectural choices, we test Gaussian and orthogonal initialization for both fully connected and convolutional architectures (which is detailed in the next

subsection). In Figure 3 we test the effect of orthogonal initialization with a depth of 40, $\sigma_w^2 = 2$, and $\sigma_b^2 = 0.05$ and the $\tanh$ activation function (which is squarely in the ordered phase, so is trainable). The test accuracy of orthogonal initialization quickly exceeds Gaussian initialization across both architectures, matching the predictions of the theory. Thus, we henceforth use orthogonal initialization for both our fully-connected and convolutional layers, as it improves the dynamical isometry of the architecture.

4.5. Convolutional Layers

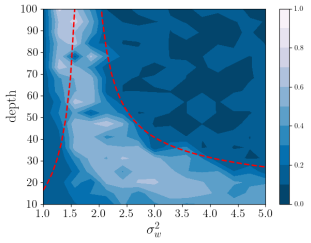

Figure 4. Trainability of CNN with $\tanh$ activation for 5 epochs on CIFAR-10, with multiple of depth scale ($4.5\xi_c$) overlaid on top.

We showed in Section 3 that the mean-field theory extends to convolutional neural networks with kernel matrix weights sampled according to σ_w^2, and in Figure 4 we demonstrate agreement between theory and practice. Being an architecture that is better for complicated vision tasks such as CIFAR-10, we see the performance within the ordered phase in Figure 4 far exceeds the performance within the ordered phase in the right side of Figure 2.

5. Discussion and Conclusion

In this paper, we developed a mean-field theory of information propagation through deep neural networks, and showed that there exists a stark phase transition between order (where deep neural networks are trainable) and chaos (where the vanishing/exploding gradient problem renders learning impossible). The nature of the heatmaps presented in this paper reveal something entirely nontrivial. Consider the aforementioned depth scale $\xi_c(\mathcal{A})$ that is a function of the architecture $\mathcal{A}$ alone and outputs a tuple (σ_w^2, D). This depth scale is such that such that a neural network achieves good accuracy at (σ_w^2, D) but fails to be able to learn at $(\sigma_w^2, D + \varepsilon)$. If the network is deeper than the maximum depth scale, our mean field theory explains that inputs will be uncorrelated after propagating through the network, which corresponds to inability to learn. However, a naive guess may predict that there no phase transition between order and chaos, but rather that there is a smooth gradient where neural networks get gradually harder to train as the depth increases. However, we show that the naive picture is the *exception, rather than the rule*, and is satisfied only by special activation functions such as sigmoid (see this "naive picture" in the bottom right of Figure 1). Rather, there is surprisingly a "cliff" with shape $\xi_c(\mathcal{A})$, such that falling off the cliff makes it impossible to train the neural network.

It is worth remarking that the effort herein is a theoretical study on trainability of deep neural networks rather than a one on optimizing accuracy on the test set. Due to computational constraints, the neural networks in the heatmaps were not able to be trained until convergence, hence for instance the poor performance on CIFAR in Figure 2. Within the ordered regime, neural networks are trainable and gradient flow is possible, and from there, architectural optimizations can be performed to provide the best generalization to the test set. See for instance that in Figure 3, when the neural networks are trained until convergence, they achieve much higher accuracy than what is displayed in the plots for the corresponding points.

In the same vein, for the purposes of fair comparison, when generating the heatmaps we did not try any more advanced techniques that are known to improve the performance of deep vision models such as dropout, learning rate decay, etc. The convolutional architecture that we tested with was clearly far from optimal; we immediately downsampled the images for ease of computation, aggressively pooled, and used a simple linear projection layer at the end. Not downsampling or pooling all at once, having a larger MLP to compute the logits, or a variety of other things could help improve the architecture. Our results should be taken as a theoretical prediction of the depth scales for trainability of neural networks, with astounding agreement with empirical experiments, rather than an attempt to design a model with the best performance on MNIST or CIFAR-10.

Revisiting our initial goal of training deep neural networks without explicit architectural techniques to ensure gradient flow, we have indeed shown trainability in a certain regime. However, as an important intermediate result, we also observed the importance of dynamical isometry, which motivates initializing with orthogonal weight matrices rather than iid Gaussian and using sigmoidal activation functions (because ReLU, for instance, cannot achieve dynamical isometry). Indeed, with orthogonal initialization, hyperparameter tuning, and some more compute, we can surprisingly achieve quite good performance with even extremely deep vanilla neural networks without running into challenges with vanishing or exploding gradients. Notably, [15] trains a 10,000 layer convolutional neural network using these techniques and achieves 80% accuracy on the test set; more reasonably sized networks achieve nearly 90% which is comparable to the state of the art for sigmoidal networks. Thus, maybe the first instinct when training deep neural networks should not be to add batch normalization between each layer, but rather to try to sample weight matrices in the ordered region and achieve dynamical isometry!

6. Appendix

Proof of Theorem 1. We analyze the following two equations given in [12] and [13]:

$$q^* = \sigma_w^2 \int \mathcal{D}z(\varphi(\sqrt{q^*}z))^2 + \sigma_b^2 \tag{23}$$

$$\chi_1 = \sigma_w^2 \int \mathcal{D}z(\varphi'(\sqrt{q^*}z))^2 \tag{24}$$

Let $k = \sqrt{q^*}$, and first assume that k can take any value in $[0, \infty)$. Noting that $\varphi'(x) = \varphi(x)(1 - \varphi(x))$, then

$$(\varphi'(kz))^2 = [\varphi(x)(1 - \varphi(x))]^2 = \varphi(x)^2 - 2\varphi(x)^3 + \varphi(x)^4$$

so if $\chi_1 = 1$ then 24 becomes

$$\frac{1}{\sigma_w^2} = \int \mathcal{D}z[\varphi(kz)^2 - 2\varphi(kz)^3 + \varphi(kz)^4] \tag{25}$$

To minimize σ_w^2, we would like to maximize the integral over all $k \in [0, \infty)$. We claim that this occurs precisely when $k = 0$. Indeed, define

$$\Phi(x) \equiv \varphi(x)^2 - 2\varphi(x)^3 + \varphi(x)^4 \tag{26}$$

and compute

$$\begin{aligned}\Phi'(x) &= 2\varphi(x) - 6\varphi(x)^2 + 4\varphi(x)^3 \\ &= 2\varphi(x)(1 - 2\varphi(x))(1 - \varphi(x))\end{aligned}$$

Notice that since $\varphi(x) \in (0, 1)$, then if $\Phi'(x) = 0$, it must be the case that $1 - 2\varphi(x) = 0$, or $\varphi(x) = \frac{1}{2}$, yielding $x = 0$. The second derivative test shows that this is indeed a maximum, since $\Phi''(0) = -1 < 0$. Since the Gaussian measure $\mathcal{D}z$ is always positive, then the right hand side of 25 is clearly maximized when $k = 0$: if $k \neq 0$, then $\Phi(kz) < \Phi(0)$ for all $z \neq 0$. This gives a preliminary bound for σ_w: if $k = 0$, then

$$\frac{1}{\sigma_w} = \frac{1}{16} \int \mathcal{D}z = \frac{1}{16} \implies \sigma_w = 16 \tag{27}$$

in view of the known fact that the integral of the Gaussian measure $\mathcal{D}z$ is equal to 1. However, it turns out that this is not achievable, since it requires $q^* = 0$. This would reduce 23 to

$$0 = \sigma_w^2 \int \mathcal{D}z(\varphi(0))^2 + \sigma_b^2 = \frac{\sigma_w^2}{4} + \sigma_b^2$$

which implies $\sigma_w^2 = \sigma_b^2 = 0$ since variances are necessarily nonnegative. But if $\sigma_w^2 = 0$ and $\chi_1 = 1$, then by 24, we cannot have $q^* = 0$.

This means that the preliminary bound in 27 can actually be improved. Again notice that since $\varphi(x) \in (0, 1)$ and it is monotonically increasing, then by computation, $\Phi'(x) < 0$ for $x > 0$, and $\Phi'(x) > 0$ for $x < 0$. In particular, this means that $\Phi(x)$ is strictly decreasing for $x > 0$ and strictly increasing for $x < 0$, so that if $|x_1| < |x_2|$ then $\Phi(x_1) > \Phi(x_2)$. This shows that for all z, it must be the case that $\Phi(k_1 z) \geq \Phi(k_2 z)$ if $|k_1| < |k_2|$, with equality only occurring when $z = 0$. Again since the Gaussian measure $\mathcal{D}z$ is positive, then the right-hand side of 25 is larger for smaller k (recall that $k = \sqrt{q^*}$ is nonnegative). Therefore, to minimize σ_w^2, we would like to minimize $k = \sqrt{q^*}$ as well, meaning that we should minimize q^*. Examining 23, it is clear that the integral on the right-hand side is nonnegative, meaning that q^* increases with respect to σ_b^2. To minimize q^*, we should therefore let $\sigma_b^2 = 0$. This yields a system of two equations

$$\begin{aligned}q^* &= \sigma_w^2 \int \mathcal{D}z(\varphi(\sqrt{q^*}z))^2 \\ 1 &= \sigma_w^2 \int \mathcal{D}z(\varphi'(\sqrt{q^*}z))^2 = \chi_1\end{aligned}$$

with two unknowns q^* and σ_w^2. Numerical computation gives $\sigma_w \approx 103.05$ (as shown in Figure 5), which proves the bound $\sigma_w^2 \gtrsim 100$.

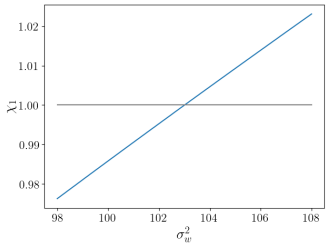

Figure 5. Susceptibility of sigmoid as a function of weight variance with $\sigma_b^2 = 0$. Recall that the phase transition from order to chaos occurs at $\chi_1 = 1$, plotted in gray.

7. Contributions & Acknowledgements

Iris initially scaffolded the codebase for the mean field for fully connected neural networks on MNIST, and Raj extended the functionality to include CIFAR10, orthogonal initialization, and convolutional layers. Both Raj and Iris contributed to running the experiments, writing the paper, and conducting the theoretical analysis. We have cited the publicly available code that we used elsewhere in the paper. Compute was through the class's AWS credits. Inspiration for this project came from Prof. Surya Ganguli's APPPHYS 229 class offered this quarter and Raj's background in statistical physics.

As a remark, we had a conversation with our TA (Saumya) during which we confirmed that our more theoretical project was acceptable for the CS231N project, despite not exactly conforming to the requirements/rubric provided by CS231N which was intended for more implementation-focused projects.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016.

[2] A. Cowsik, T. Nebabu, X. Qi, and S. Ganguli. Geometric dynamics of signal propagation predict trainability of transformers. *CoRR*, abs/2403.02579, 2024.

[3] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015.

[6] JiJingYu. Delta orthogonal initialization for pytorch. `https://github.com/JiJingYu/delta_orthogonal_init_pytorch`, 2018. GitHub repository.

[7] C. Kittel and H. Kroemer. *Thermal Physics*. W. H. Freeman, 1980.

[8] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Canada, 2009.

[9] G.-H. Liu and E. A. Theodorou. Deep learning theory review: An optimal control and dynamical systems perspective. *arXiv preprint arXiv:1908.10920*, 2019.

[10] T. Mikolov. Statistical language models based on neural networks. Master's thesis, Brno University of Technology, Brno, Czech Republic, 2012.

[11] J. Pennington, S. S. Schoenholz, and S. Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4785–4795, 2017.

[12] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3360–3368, 2016.

[13] S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein. Deep information propagation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[14] H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in random neural networks. *Phys. Rev. Lett.*, 61:259–262, Jul 1988.

[15] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. S. Schoenholz, and J. Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10, 000-layer vanilla convolutional neural networks. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5389–5398. PMLR, 2018.

[16] G. Yang and S. S. Schoenholz. Mean field residual networks: On the edge of chaos. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 7103–7114, 2017.