Gradient Flow Convergence Guarantee for General Neural Network Architectures

Yash Jakhmola
yj20ms028@iiserkol.ac.in

Abstract

A key challenge in modern deep learning theory is to explain the remarkable success of gradient-based optimization methods when training large-scale, complex deep neural networks. Though linear convergence of such methods has been proved for a handful of specific architectures, a united theory still evades researchers. This article presents a unified proof for linear convergence of continuous gradient descent, also called gradient flow, while training any neural network with piecewise non-zero polynomial activations. Our primary contribution is a single, general theorem that not only covers architectures for which this result was previously unknown but also consolidates existing results under weaker assumptions. While our focus is theoretical and our results are only exact in the infinitesimal step size limit, we nevertheless find excellent empirical agreement between the predictions of our result and those of the practical step-size gradient descent method.

***K*eywords** gradient flow · linear convergence · optimization

1 Introduction

Understanding why gradient-based optimization methods are so effective at training large, complex deep neural networks is a central question in modern machine learning theory. Despite the non-convex nature of the loss landscape, these methods can find solutions with low training error. This empirical success is well-known and is a reason for the success of neural networks in various real-life tasks like image recognition Krizhevsky et al. [2012], He et al. [2015], language processing Vaswani et al. [2017], Achiam et al. [2023], sequence tasks Schuster and Paliwal [1997], Hochreiter and Schmidhuber [1997], time series analysis, media generation Goodfellow et al. [2020], Sohl-Dickstein et al. [2015] among many more. However, finding theoretical guarantees that training via gradient methods converges, especially at exponential rates and across a broad class of architectures, remains an active area of research.

1.1 Previous Works

The seminal work by Jacot et al. [2018] proved that the training of DNNs with a smooth activation function in the infinite width limit for a finite time with gradient descent can be characterized by a kernel. Our proof technique relies on the same kernel matrix used by them.

Linear convergence of training error has been known under simple settings. Some works that use a trajectory-based analysis of the algorithm dynamics are as follows. Allen-Zhu et al. [2019b] proved that if inputs are separable and network is overparametrized, convergence of SGD is linear for DNNs, CNNs and ResNets. Allen-Zhu et al. [2019a] talks about learning and generalization bound for shallow networks. Awasthi et al. [2021] proved that if inputs are normally distributed and ground truth weights matrix has row norm unity for each row, convergence of GD is linear for GNNs initialized normally with sum-aggregation and degree of graph $o(\sqrt{n})$ with n being number of vertices. Chatterjee [2022] is one the closest to our work. It proves that for DNNs with twice differentiable activations, trained on a linearly independent input training dataset with either GF or GD with small step size, convergence is linear. Chen et al. [2021] proves that for biasless DNNs with ReLU activation trained on an orthonormal input training dataset with GD with small step size, loss decreases reciprocal to number of epochs. Du et al. [2019a] proved that for normalized inputs, GD converges in polynomial time for biasless DNNs with smooth or non-polynomial activations. Du et al. [2019b]

proved that for non-parallel inputs, GF converges linearly for overparametrized biasless shallow NNs with ReLU activations, initialized normally. Zou et al. [2020] proved that for ReLU DNNs without biases, with separable training data, loss decreases as reciprocal of number of epochs. Works by Boursier et al. [2022], Gopalani and Mukherjee [2025] prove linear convergence of GF for shallow networks. Liu et al. [2022], Nguyen and Mondelli [2020] prove linear convergence of GD for DNNs with specific activations.

The works by Min et al. [2023], Bah et al. [2022], Tarmoun et al. [2021], Xu et al. [2025] proved linear convergence of GF/GD, but for linear networks (ie. networks with linear activations, essentially making it linear regression). Hutzenthaler et al. [2023] provides a detailed analysis of convergence of both GF and GD, but for constant target functions.

Other approaches to finding convergence properties include landscape-based analysis Arora et al. [2022], Jin et al. [2017], mean-field approximation based analysis Chen et al. [2022], Ding et al. [2022] and using optimal transport theory Chizat and Bach [2018], Khamis et al. [2024].

1.2 Our Contributions

This paper generalizes the linear convergence results stated above by demonstrating that for a broad class of sufficiently over-parameterized neural networks whose initial weights are randomly sampled from an absolutely continuous probability distribution, training via gradient flow results in the training error converging to zero at a linear rate (ie., the loss decaying exponentially) almost surely (with respect to initialization) for almost all training datasets.

We consider a general function structure composed of polynomial layers and a piecewise polynomial activation function with finite zeroes, which covers many popular architectures like DNNs, ResNets He et al. [2016], CNNs Krizhevsky et al. [2012], ConvResNets Du et al. [2019a], GCNs Kipf and Welling [2017], ChebConvNets Defferrard et al. [2016], GraphConvNets Morris et al. [2019], U-Nets Ronneberger et al. [2015], MobileNets Howard et al. [2017] and SiameseNets Taigman et al. [2014] with activations like leaky ReLU Maas et al. [2013] or parametric ReLU He et al. [2015], just to name a few. Note that using leaky ReLU is not considered a huge handicap in comparison to using ReLU, as shown by Xu et al. [2015], Ramachandran et al. [2018], Dubey et al. [2022]. Morover, there has been no work for leaky ReLU or parametric ReLU activated networks in the past. This is still a non-comprehensive list of architectures supported by our result, and we keep our experiments in section 4 limited to the more popular architectures.

Such linear convergence results have been known for simpler architectures, larger over-parametrization and specific initializations, as detailed below. We unify the analysis by placing emphasis on the spectral properties of the Gram (NTK) matrix, in particular its positive definiteness during training.

We focus on gradient flow to not have to keep track of discrete step sizes, which complicates the analysis and pulls attention away from the more important insights. Moreover, Elkabetz and Cohen [2021] proves that GF is close to GD with small step sizes for DNNs. It is to be noted that even though analysis using gradient flow is usually less cumbersome than gradient descent, the proof in this paper is not anywhere in the literature - despite the simplicity of its arguments. Previous works do talk about the convergence of gradient flow, but they are again limited to either shallow networks Du et al. [2019b], Boursier et al. [2022], Gopalani and Mukherjee [2025], Tarmoun et al. [2021] or deep networks with smooth Chatterjee [2022] or linear activations Min et al. [2023], Bah et al. [2022], Xu et al. [2025].

The major novelty of our result is that it holds for a large variety of network architectures. Previous results have often been found for bias-less architectures Du et al. [2019b], Allen-Zhu et al. [2019b], Liu et al. [2022], real-valued Gopalani and Mukherjee [2025], Du et al. [2019a] or constant Hutzenthaler et al. [2023] targets, very high over-parameterization Du et al. [2019a], Allen-Zhu et al. [2019a], Chen et al. [2021], a specific initialization of weights Liu et al. [2022], Zou et al. [2020], Nguyen and Mondelli [2020] or strong assumptions on input data like orthogonality Boursier et al. [2022] or linear independence Chatterjee [2022]. Such convergence guarantees have also been found for other architectures like GNNs Awasthi et al. [2021], CNNs Allen-Zhu et al. [2019b] and ResNets Du et al. [2019a]. However, there has not been one work that combines all these results (and more) into a single, easy to follow theorem.

To summarise, our major contributions are as follows:

- We prove linear convergence of gradient flow for a wide variety of architectures. Previous analyses have worked out linear convergence on a case-by-case basis Awasthi et al. [2021], Bah et al. [2022]. We prove a general theorem that works for any general function structure composed of polynomial layers and any piecewise polynomial activation that is zero at finitely many points.

- We require initialization to be from any absolutely continuous distribution. Previous analyses usually have only considered normal distributions Du et al. [2019a], Nguyen and Mondelli [2020].

- We require overparametrization of only at least nM. Previous works have required over-parameterization to be of much larger magnitudes Du et al. [2019b], Chen et al. [2021].

Moreover, our proof stems directly from first principles - using standard results from the theory of ordinary differential equations and measure theory in conjunction with already established proof methods Du et al. [2019b,a], Chatterjee [2022]. Our proof radically simpler and shorter than almost all other proofs in this line of research, thus making it much simpler to follow and understand.

Despite being a rather short and concise proof, our contributions sharpen, generalize, and in some cases correct limitations in prior work by covering a wide range of architectures with weaker assumptions on activation functions, initialization, training data and over-parameterization.

2 Preliminaries

2.1 Setup

Consider a continuous function $f : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^M$ defined as follows for some $N, P, M \in \mathbb{N}$:

$$f(X, \theta) := g_L(\sigma(\dots g_2(\sigma(g_1(X, \theta))) \dots)) \tag{1}$$

for all inputs $X \in \mathbb{R}^N$ and parametrizations $\theta \in \mathbb{R}^P$, where $g_i : \mathbb{R}^{N_{i-1}} \times \mathbb{R}^P \to \mathbb{R}^{N_i}$ are polynomials (in each component) for all $i = 1, \dots, L$ with $N_0 = N, N_L = M$ and $N_1, \dots, N_{L-1} \in \mathbb{N}$ for some $L \in \mathbb{N}$. $\sigma : \mathbb{R} \to \mathbb{R}$ is a continuous, piecewise polynomial which is zero at finitely many values only. Note that σ applied to a vector means that it is applied component-wise.

We intend to find that parametrization of f that minimizes its error over a training dataset $(X_1, y_1), \dots, (X_n, y_n) \in \mathbb{R}^N \times \mathbb{R}^M$.

To do so, we utilize gradient flow. Start from some parameters $\theta(0)$ sampled from an arbitrary continuous probability distribution. Then, get $\theta(T)$ by solving the following ODE till $t = T$.

$$\frac{\mathrm{d}\theta(t)}{\mathrm{d}t} = -\frac{\partial L(\theta(t))}{\partial \theta(t)} \tag{2}$$

where $L : \mathbb{R}^P \to [0, \infty)$ is the error:

$$L(\theta) := \frac{1}{2} \sum_{j=1}^{n} ||f(X_j, \theta) - y_j||^2 \tag{3}$$

for all $\theta \in \mathbb{R}^P$.

3 Main Result

Theorem 3.1. *Appling gradient flow to the above kind of functions with $P \geq nM$, for almost every training dataset, the following holds almost surely with respect to the initialization distribution, for all* $t \in [0, T]$

$$||y - F_t|| \leq e^{-\lambda_0 t} ||y - F_0|| \tag{4}$$

where $\lambda_0 > 0$ and

$$F_t := ((f(X_1, \theta(t)))_1, \dots, (f(X_1, \theta(t)))_M, \dots, (f(X_n, \theta(t)))_1, \dots, (f(X_n, \theta(t)))_M) \in \mathbb{R}^{nM} \tag{5}$$

$$y := ((y_1)_1, \dots, (y_1)_M, \dots, (y_n)_1, \dots, (y_n)_M)^T \in \mathbb{R}^{nM} \tag{6}$$

Proof. The first half of this proof is inspired heavily by previous works Du et al. [2019b], where we bound the error by the least eigenvalue of the 'neural tangent kernel'.

Fix some $t \geq 0$. Note that, by chain rule,

$$\frac{\partial L(\theta(t))}{\partial \theta(t)} = \sum_{j=1}^{n} \frac{\partial f(X_j, \theta(t))}{\partial \theta(t)} (f(X_j, \theta(t)) - y_j) \tag{7}$$

Now, we calculate the dynamics of the predictions.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}f(X_i,\theta(t)) &= \left(\frac{\partial f(X_i,\theta(t))}{\partial\theta(t)}\right)^T\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}\\ &= -\left(\frac{\partial f(X_i,\theta(t))}{\partial\theta(t)}\right)^T\frac{\partial L(\theta(t))}{\partial\theta(t)}\\ &= -\left(\frac{\partial f(X_i,\theta(t))}{\partial\theta(t)}\right)^T\left(\sum_{j=1}^{n}\frac{\partial f(X_j,\theta(t))}{\partial\theta(t)}(f(X_j,\theta(t))-y_j)\right)\\ &= \sum_{j=1}^{n}\left(\left(\frac{\partial f(X_i,\theta(t))}{\partial\theta(t)}\right)^T\frac{\partial f(X_j,\theta(t))}{\partial\theta(t)}\right)(y_j-f(X_j,\theta(t)))\end{aligned}\tag{8}$$

Concatenating output vectors, we get $\frac{\mathrm{d}F_t}{\mathrm{d}t} = G(t)(y-F_t)$, where

$$G(t) := \left(\left(\frac{\partial F_t}{\partial\theta(t)}\right)_p^T\left(\frac{\partial F_t}{\partial\theta(t)}\right)_q\right)_{p,q=1}^{nM}\tag{9}$$

where $\frac{\partial F_t}{\partial\theta(t)} = \left(\frac{\partial(f(X_1,\theta(t)))_1}{\partial\theta(t)},\dots,\frac{\partial(f(X_1,\theta(t)))_M}{\partial\theta(t)},\dots,\frac{\partial(f(X_n,\theta(t)))_1}{\partial\theta(t)},\dots,\frac{\partial(f(X_n,\theta(t)))_M}{\partial\theta(t)}\right) \in \mathbb{R}^{P\times nM}$ and $P\in\mathbb{N}$ is the number of trainable parameters.

Now, $G(t)$ is a nonegative definite matrix, since it is a gram matrix. We further claim (from 3.2) that it is a positive definite matrix almost surely (with respect to initialization).

Now, consider

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(||y-F_t||^2\right) = -2(y-F_t)^TG(t)(y-F_t) \leq -2\lambda_0||y-F_t||^2\tag{10}$$

where $\lambda_0>0$ is the least eigenvalue of $G(t)$ for $t\in[0,T]$. $\lambda_0>0$ is true because of the following argument. The minimum eigenvalue of a matrix varies continuously with the matrix entries. $G_{pq}(t)$ is continuous since $\theta(t)$ varies continuously with t and σ is continuous. Thus, by extreme value theorem applied on the compact interval $[0,T]$, the minimum eigenvalue is achieved at some $t_0\in[0,T]$. Say the minimum eigenvalue of $G(t)$ at t_0 is λ_0. We know $\lambda_0>0$ due to 3.2.

This implies that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(e^{2\lambda_0t}||y-F_t||^2\right)\leq 0\tag{11}$$

Thus, putting in $t=0$ and $t=s\leq T$,

$$||y-F_s||\leq e^{-\lambda_0s}||y-F_0||\tag{12}$$

□

Lemma 3.2. *$G(t)$ is a positive definite matrix almost surely (with respect to initialization) for all t ≥ 0.*

Proof. We know that $G(t)$ is positive definite iff $\left\{\left(\frac{\partial F_t}{\partial\theta(t)}\right)_1,\dots,\left(\frac{\partial F_t}{\partial\theta(t)}\right)_{nM}\right\}\subset\mathbb{R}^P$ is linearly independent. Note that this set is always linearly dependent if $nM>P$. Thus, we shall assume that $nM\leq P$. We thus need the function to be overparametrized with respect to the training sample size.

Now, if this set is linearly dependent at initialization, then all sub-square matrices of $\left(\left(\frac{\partial F_0}{\partial\theta(0)}\right)_1,\dots,\left(\frac{\partial F_0}{\partial\theta(0)}\right)_{nM}\right)\in\mathbb{R}^{P\times nM}$ will be singular. Note that the determinants of these matrices are piecewise polynomials in the parameters and σ is zero only at finitely many points, since it is a piecewise nonzero polynomial. Thus, the determinants are zero for only a finite set of values of the parameters, unless they are the zero polynomial.

But, since the coefficients depend only on the training dataset, we can look at the determinants as a piecewise polynomial in the training dataset values, for some fixed value of $\theta(0)$. Then, this polynomial can be zero for only a finite number of datasets for a fixed $\theta(0)$, using the same argument as above. Thus, the determinants can be zero polynomials for only a finite number of datasets.

Since the initial parameters θ_0 are sampled from an absolutely continuous probability distribution, the probability that all the matrices will be singular (at initialization) is zero for almost all datasets.

Let θ_{degen} denote the set of those parameters for which the determinants are zero, or that set of parameters for which $\nabla L(0) = 0$, since gradient flow will not progress at all. Note that since $\nabla L(0)$ is also a piecewise polynomial which is not a zero polynomial over an open set, the measure of the set of weights such that $\nabla L(0) = 0$ is also finite. Thus, due to the continuous distribution of initial parameters, we have shown that $\mathbb{P}(\theta(0) \in \theta_{\text{degen}}) = 0$. We now need to show that $\mathbb{P}(\theta(t) \in \theta_{\text{degen}}) = 0$ for all $t > 0$.

Since $-\nabla L(\theta(t))$ is a piecewise polynomial in the parameters, it and its derivative (with respect to the parameters) are continuous in every piece. Thus, by Picard's existence and uniqueness theorem Nagle et al. [2011], the gradient flow equation has a unique solution in a neighbourhood of $t = 0$. Since there is no finite time blowup (proved below) of the solution, it can be extended beyond a neighbourhood of $t = 0$ (Hartman [1964], chapter II, corollary 3.2, pg 14). If this solution hits the boundary of the piece containing the initial condition $\theta(0)$, we can construct solution for the same ODE in the next piece, with initial condition being defined as the end point of the previous solution curve. These solutions can be joined together continuously (due to no finite time blowup) until we reach $t = T$.

We thus define this piecewise differentiable, but continuous everywhere solution of the gradient flow equation by $\theta(t) = \Phi_t(\theta(0))$ for all $t \geq 0$.

Now, Φ_t is piecewise differentiable with respect to initial conditions (since $-\nabla L_t$ is piecewise continuously differentiable with respect to initial conditions, Hartman [1964], chapter V, theorem 3.1, pg 95), invertible (since inverse is given by solution of $\dot{\theta}(t) = \nabla L(\theta(t))$ with initial condition $\theta(T)$ given that $\nabla L(\theta(T)) \neq 0$ (if $\nabla L(\theta(T)) = 0$, run the algorithm till $T - \epsilon$ for some appropriate $\epsilon > 0$), which also exists by the above arguments) and Φ_t^{-1} is also piecewise differentiable (by previous argument). We thus conclude that Φ_t is a piecewise diffeomorphism over the initial conditions space. It can be shown that piecewise differentiable functions (with finitely many pieces) map zero probability sets to zero probability sets (Rudin [2006], chapter 7, special case of lemma 7.25, pg 153).

We have shown that with respect to initialization, $\mathbb{P}(\theta_{\text{degen}}) = 0$. Thus, from above arguments, $\mathbb{P}(\Phi_t^{-1}(\theta_{\text{degen}})) = 0$, which implies that, with respect to initialization, $\mathbb{P}(\theta(0) | \, \Phi_t(\theta(0)) \in \theta_{\text{degen}}) = \mathbb{P}(\theta(t) \in \theta_{\text{degen}}) = 0$.

To finish the proof, we now prove $||\theta(s)|| < \infty$, for any $s > 0$.

Note that

$$\frac{\mathrm{d}L(\theta(t))}{\mathrm{d}t} = \left\langle \nabla L(\theta(t)), \frac{\mathrm{d}\theta(t)}{\mathrm{d}t} \right\rangle = -||\nabla L(\theta(t))||^2 \tag{13}$$

Thus,

$$\int_0^s ||\nabla L(\theta(t))||^2 dt = -\int_0^s \frac{\mathrm{d}L(\theta(t))}{\mathrm{d}t} dt = L(\theta(0)) - L(\theta(s)) \leq L(\theta(0)) < \infty \tag{14}$$

Now,

$$||\theta(s) - \theta(0)|| = \left|\left| \int_0^s \frac{\mathrm{d}\theta(t)}{\mathrm{d}t} dt \right|\right| = \left|\left| \int_0^s \nabla L(\theta(t)) dt \right|\right| \leq \int_0^s ||\nabla L(\theta(t))|| dt \tag{15}$$

Using Cauchy-Schwartz,

$$\left(\int_0^s 1 \cdot ||\nabla L(\theta(t))|| dt \right)^2 \leq s \int_0^s ||\nabla L(\theta(t))||^2 dt \tag{16}$$

Thus,

$$||\theta(s) - \theta(0)||^2 \leq s \int_0^s ||\nabla L(\theta(t))||^2 dt < \infty \tag{17}$$

Thus, $||\theta(s)|| < \infty$ for all $s > 0$. □

Remark. The above theorem is presented in such a way because the conclusion can be shown to hold for several different neural network architectures. Some of them are listed as corollaries below. Note than an analogous proof holds for matrix-valued inputs by concatenating the matrix into a vector. We omit the proof for that case for the sake of notational simplicity.

Corollary 3.2.1 (Linear convergence of DNNs). *A DNN is the function $f : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^M$ as defined in subsection 2.1 with $g_i : \mathbb{R}^{N_{i-1}} \times \mathbb{R}^P \to \mathbb{R}^{N_i}$, defined by*

$$g_i(X, \theta) = W_i X + B_i \tag{18}$$

for all $X \in \mathbb{R}^{N_{i-1}}$ with $W_i \in \mathbb{R}^{N_i} \times \mathbb{R}^{N_i - 1}, B_i \in \mathbb{R}^{N_i}$, for all $i \in [L]$, $N_0 = N, N_L = M$ and $\theta \in \mathbb{R}^P$ denoting the vector of all weights and biases. σ can be leaky ReLU or parametric ReLU, for example.

Appling gradient flow to ResNets with $P \geq nM$, for almost every training dataset of size n, the following holds almost surely with respect to the initialization distribution, for all $t \in [0, T]$

$$||y - F_t|| \leq e^{-\lambda_0 t}||y - F_0|| \tag{19}$$

where $\lambda_0 > 0$.

Corollary 3.2.2 (Linear convergence of ResNets)**.** *A ResNet is the function $f : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^M$ as defined in subsection 2.1 with $g_i : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^N$, defined by*

$$g_i(X, \theta) = X + W_i X + B_i \tag{20}$$

for all $X \in \mathbb{R}^N$ with $W_i \in \mathbb{R}^N \times \mathbb{R}^N, B_i \in \mathbb{R}^N$, for all $i \in [L]$ and $\theta \in \mathbb{R}^P$ denoting the vector of all weights and biases. σ can be leaky ReLU or parametric ReLU, for example.

Appling gradient flow to ResNets with $P \geq nM$, for almost every training dataset of size n, the following holds almost surely with respect to the initialization distribution, for all $t \in [0, T]$

$$||y - F_t|| \leq e^{-\lambda_0 t}||y - F_0|| \tag{21}$$

where $\lambda_0 > 0$.

Corollary 3.2.3 (Linear convergence of GCNs)**.** *A GCN is the function $f : \mathbb{R}^{m \times N} \times \mathbb{R}^P \to \mathbb{R}^M$ as defined in subsection 2.1 with $g_i : \mathbb{R}^{m \times N_{i-1}} \times \mathbb{R}^P \to \mathbb{R}^{m \times N_i}$, defined by*

$$g_i(X, \theta) = \hat{D}^{1/2} \hat{A} \hat{D}^{1/2} X W_i + B_i \tag{22}$$

for all $X \in \mathbb{R}^{m \times N_{i-1}}$ with $W_i \in \mathbb{R}^{N_{i-1}} \times \mathbb{R}^{N_i}, B_i \in \mathbb{R}^{m \times N_i}$, for all $i \in [L-1]$, $N_0 = N$, $g_L : \mathbb{R}^{m \times N_{L-1}} \to \mathbb{R}^m$ and $\theta \in \mathbb{R}^P$ denoting the vector of all weights and biases. $\hat{A} = A + \mathbb{I}_m$, $\hat{D} = diag\Big(\sum_{j=1}^m \hat{A}_{1j}, \ldots, \sum_{j=1}^m \hat{A}_{mj}\Big)$ where A is the adjacency matrix of the graph over m vertices. σ can be leaky ReLU or parametric ReLU, for example.

Appling gradient flow to ResNets with $P \geq nM$, for almost every training dataset of size n, the following holds almost surely with respect to the initialization distribution, for all $t \in [0, T]$

$$||y - F_t|| \leq e^{-\lambda_0 t}||y - F_0|| \tag{23}$$

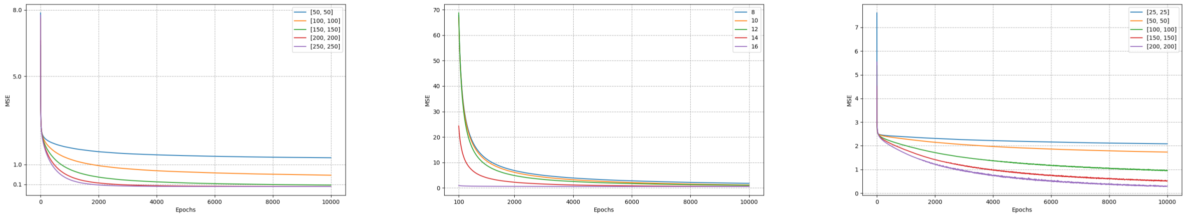
where $\lambda_0 > 0$.

4 Experiments

In this section, we use synthetic data to support our findings. For all architectures, data is generated uniformly from a ball in $\mathbb{R}^N$ with the $\mathbb{R}^M$-valued labels being generated normally. This is done in concordance to Zhang et al. [2017]. The intuition is similar to defining Rademacher complexity in learning theory - a highly over-parametrized model will be able to learn a lot of patterns - even ones close to random. Leaky ReLU activation Maas et al. [2013] is used everywhere, with the initialization being the default Kaiming-He Uniform initialization He et al. [2015]. The number of parameters are varied from being less than nM to slowly being overparametrized. The training curves verify the hypothesis that under overparametrization, these architectures converge linearly in training loss.

The rate of convergence improves with overparametrization. As stated in Du et al. [2019b], this might be because $G(t)$ matrix becomes more stable, and thus has larger least eigenvalue. We verify this claim for other architectures, and give a theoretical result supporting it as well.

The experimental setups are as follows. To simulate gradient flow, a very small learning rate is taken (lr= 10^{-5}) with a large number of epochs (10,000). Each model is trained over a training sample of size 1000. For DNN, the input dimension is 500 with output dimension 50 and the model is trained for varying hidden layer sizes - starting from two hidden layers of 50 neurons each ([50,50]) and ending with two hidden layers of 250 neurons each ([250,250]). For ResNet, the input dimension and output dimensions are 100 and the model is trained for varying number of hidden layers - starting from 8 hidden layers and ending with 16 hidden layers. For GCN, the input dimension is 500 with output dimension 50 and the model is trained for varying hidden layer sizes - starting from two hidden layers of 25 neurons each ([25,25]) and ending with two hidden layers of 200 neurons each ([200,200]). The underlying graph used is a k-nearest neighbours graph on the input data with $k = 10$.



(a) DNN error curves

(b) ResNet error curves

(c) GCN error curves

Figure 1: Results on synthetic data.

5 Conclusion and Future Directions

This paper studies linear convergence of GF to a global minimum almost surely (with respect to initial weights distribution) for any function that can be written as compositions of polynomials and piecewise polynomials with finite zeroes for almost all training datasets, given that the function is sufficiently overparametrized ($P \geq nM$). In particular, we prove linear convergence of GF to a global minimum almost surely (with respect to initial weights distribution) for a large variety of neural networks (like DNNs, GCNs, ResNets, etc.) with activations like leaky ReLU for almost all training datasets, given that the function is sufficiently overparametrized ($P \geq nM$). We prove this result by exploiting properties of the NTK and how it's lowest eigenvalue behaves under the operation of the GF equation. We shall now present some future directions.

The major concern regarding this result is its non-applicability to ReLU and non-polynomial activations like sigmoid and softmax. Though leaky ReLU is a great alternative to ReLU Xu et al. [2015], Ramachandran et al. [2018], Dubey et al. [2022], it is not as popular, perhaps due to the burden of having another hyperparameter to tune. Moreover, eliminating softmax from our discussion implies eliminating transformers. The polynomial structure of the activations was used to prove that sub-determinants of $\left(\left(\frac{\partial F_0}{\partial \theta(0)}\right)_1, \dots, \left(\frac{\partial F_0}{\partial \theta(0)}\right)_{nM}\right)$ will be zero for finite values of initialization. Proving this result using more sophisticated tools might allow for inclusion of non-polynomial activations.

Another issue might be that the eigenvalue λ_0 appears in the final rate of convergence, and we only know that it will be positive. However, it can be so small that the final rate, though linear, becomes unfavourable practically. Future works can try to bound λ_0 by a positive constant, thus improving the practicality of this result.

A minor concern can be that though the result hold for several architectures - this can be seen as a potential disadvantage, since this then fails to explain why one architecture can have a better rate than another in certain situations. However, solving that issue requires a more careful analysis, taking into account the exact problem that needs to be solved.

Finally, we rely on the loss being the squared loss. This is the most common loss used for regression, but not so much for classification. Future works can try extending this result to other losses like cross-entropy.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019a.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019b.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, pages 948–1024. PMLR, 2022.

Pranjal Awasthi, Abhimanyu Das, and Sreenivas Gollapudi. A convergence analysis of gradient descent on graph neural networks. *Advances in Neural Information Processing Systems*, 34:20385–20397, 2021.

Bubacarr Bah, Holger Rauhut, Ulrich Terstiege, and Michael Westdickenberg. Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. *Information and Inference: A Journal of the IMA*, 11(1):307–353, 2022.

Etienne Boursier, Loucas Pillaud-Vivien, and Nicolas Flammarion. Gradient flow dynamics of shallow relu networks for square loss and orthogonal inputs. *Advances in Neural Information Processing Systems*, 35:20105–20118, 2022.

Sourav Chatterjee. Convergence of gradient descent for deep neural networks. *arXiv preprint arXiv:2203.16462*, 2022.

Zhengdao Chen, Eric Vanden-Eijnden, and Joan Bruna. On feature learning in neural networks with global convergence guarantees. *arXiv preprint arXiv:2204.10782*, 2022.

Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much over-parameterization is sufficient to learn deep relu networks. In *International Conference on Learning Representations*, 2021.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Zhiyan Ding, Shi Chen, Qin Li, and Stephen J Wright. Overparameterization of deep resnet: zero loss and mean-field analysis. *Journal of machine learning research*, 23(48):1–65, 2022.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019a.

Simon S Du, Xiyu Zhai, Barnabás Poczós, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019b.

Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.

Omer Elkabetz and Nadav Cohen. Continuous vs. discrete optimization of deep neural networks. *Advances in Neural Information Processing Systems*, 34:4947–4960, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Pulkit Gopalani and Anirbit Mukherjee. Global convergence of sgd on two layer neural nets. *Information and Inference: A Journal of the IMA*, 14(1):iaae035, 2025.

Philip Hartman. *Ordinary Differential Equations*. John Wiley & Sons, Inc., 1964.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

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

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Martin Hutzenthaler, Arnulf Jentzen, Katharina Pohl, Adrian Riekert, and Luca Scarpa. Convergence proof for stochastic gradient descent in the training of deep neural networks with relu activation for constant target functions. *arXiv preprint arXiv:2112.07369*, 2023.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pages 1724–1732. PMLR, 2017.

Abdelwahed Khamis, Russell Tsuchida, Mohamed Tarek, Vivien Rolland, and Lars Petersson. Scalable optimal transport methods in machine learning: A contemporary survey. *IEEE transactions on pattern analysis and machine intelligence*, 2024.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

Hancheng Min, René Vidal, and Enrique Mallada. On the convergence of gradient flow on multi-layer linear models. In *International Conference on Machine Learning*, pages 24850–24887. PMLR, 2023.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

R. Kent Nagle, Edward B. Saff, and Arthur David Snider. *Fundamentals of Differential Equations and Boundary Value Problems*. Addison-Wesley, 6th edition, 2011.

Quynh N Nguyen and Marco Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. *Advances in Neural Information Processing Systems*, 33:11961–11972, 2020.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. In *International Conference on Learning Representations (Workshop)*, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Walter Rudin. *Real and Complex Analysis*. McGraw-Hill Education, 3rd edition, 2006.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

Salma Tarmoun, Guilherme Franca, Benjamin D Haeffele, and Rene Vidal. Understanding the dynamics of gradient flow in overparameterized linear models. In *International Conference on Machine Learning*, pages 10153–10161. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolution network. In *International Conference on Machine Learning (Deep Learning Workshop)*, 2015.

Ziqing Xu, Hancheng Min, Salma Tarmoun, Enrique Mallada, and Rene Vidal. A local polyak-łojasiewicz and descent lemma of gradient descent for overparametrized linear models. *Transactions on Machine Learning Research*, 2025.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109(3):467–492, 2020.