Journal of Machine Learning Research 27 (2026) 1-30 Submitted 12/22; Revised 9/26; Published 10/26

Convergence of Gradient Descent for Deep Neural Networks

Sourav Chatterjee SOURAVC@STANFORD.EDU
Department of Statistics, Stanford University
Sequoia Hall, 390 Jane Stanford Way
Stanford, CA 94305-4020, USA

Editor: Tong Zhang

Abstract

We give a simple *local Polyak–Łojasiewicz* (PL) criterion that guarantees linear (exponential) convergence of gradient flow and gradient descent to a zero-loss solution of a nonnegative objective. We then verify this criterion for the squared training loss of a feedforward neural network with smooth, strictly increasing activation functions, in a regime that is complementary to the usual over-parameterized analyses: the network width and depth are fixed, while the input data vectors are assumed to be linearly independent (in particular, the ambient input dimension is at least the number of data points). A notable feature of the verification is that it is *constructive*: it leads to a simple "positive" initialization (zero first-layer weights, strictly positive hidden-layer weights, and sufficiently large output-layer weights) under which gradient descent provably converges to an interpolating global minimizer of the training loss. We also discuss a probabilistic corollary for random initializations, clarify its dependence on the probability of the required initialization event, and provide numerical experiments showing that this theory-guided initialization can substantially accelerate optimization relative to standard random initializations at the same width.

Keywords: nonconvex optimization, gradient descent, deep learning, neural networks, convergence

2020 Mathematics Subject Classification. Primary 65K10; Secondary 90C26, 68T07.

1. A Convergence Criterion for Gradient Descent

We first review the background and then state the convergence criterion.

1.1 Background

The goal of gradient descent is to find a minimum of a differentiable function $f : \mathbb{R}^p \to \mathbb{R}$ by starting at some $x_0 \in \mathbb{R}^p$, and then iteratively moving in the direction of steepest descent, as

$$x_{k+1} = x_k - \eta_k \nabla f(x_k),$$

where ∇f is the gradient of f, and the step sizes η_k may remain fixed or vary with k. This is the discretization of the continuous-time gradient flow $\phi : [0, \infty) \to \mathbb{R}^p$, which is the solution of the differential equation

$$\frac{d}{dt}\phi(t) = -\nabla f(\phi(t))$$

with $\phi(0) = x_0$. Gradient descent and its many variants are indispensable tools in all branches of science and engineering, and particularly in modern machine learning and

©2026 Sourav Chatterjee.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v27/22-1457.html.

data science. The convergence properties of gradient descent are well-understood when the objective function f is convex (Nesterov, 2004; Boyd and Vandenberghe, 2004), and in general nonconvex optimization, even *certifying* local optimality can be computationally intractable (NP-hard, and related decision problems are NP-complete); see, for example, Murty and Kabadi (1987). Of course, for many "tame" objectives gradient methods do converge to critical points; the point here is that worst-case guarantees for locating (or certifying) good minima of arbitrary nonconvex functions are necessarily limited. In spite of this, gradient descent is widely used in practice to find local and global minima in highly nonconvex problems, especially in high dimensions. For example, it has been observed that gradient descent can often find global minima of training loss in deep learning (Zhang et al., 2021; Goodfellow et al., 2014), which is one of the reasons behind great success of the deep learning revolution (LeCun et al., 2015; Goodfellow et al., 2014).

Beyond worst-case guarantees, one motivation for our results is to obtain convergence theory that is not merely descriptive but *prescriptive*: it can suggest concrete algorithmic choices. In the neural-network application, the sufficient conditions can be verified by analyzing a local lower bound on the empirical tangent kernel, and this yields a particularly simple, sign-constrained starting point. In particular, the "positive" initialization singled out by the proof (zero first-layer weights, positive hidden-layer weights, and a suitably scaled output layer) is easy to implement. Our numerical experiments show that this theory-guided start can dramatically speed up training compared to standard random initializations at the same (small) width, serving as a proof-of-principle that optimization theory can be used to design effective initializations.

This article presents a novel criterion for convergence of gradient descent to a global minimum. The criterion is related to (and may be seen as a strengthening of) the classical Kurdyka–Łojasiewicz inequality (Attouch et al., 2010; Łojasiewicz, 1963; Kurdyka, 1998). It is also related to results from nonsmooth analysis, such as those in Ioffe (2000); Drusvyatskiy et al. (2015); Bolte et al. (2010); Corvellec and Motreanu (2008). Our proof idea bears close resemblance with the classical 'Łojasiewicz trapping argument' (Absil et al., 2005; Kurdyka, 1998). Nevertheless, the convergence criterion stated below has not appeared in this exact form previously in the literature.

1.2 Main Results

Let p be a positive integer. Let $f : \mathbb{R}^p \to [0, \infty)$ be a nonnegative C^2 function. Take any $x_0 \in \mathbb{R}^p$ and $r > 0$. Let $B(x_0, r)$ denote the closed Euclidean ball of radius r centered at x_0. Define

$$\alpha(x_0, r) := \inf_{x \in B(x_0,r),\, f(x) \neq 0} \frac{|\nabla f(x)|^2}{f(x)},$$

where $|\nabla f(x)|$ is the Euclidean norm of $\nabla f(x)$. If $f(x) = 0$ for all $x \in B(x_0, r)$, then we let $\alpha(x_0, r) := \infty$. Our main assumption is that for some $r > 0$,

$$4f(x_0) < r^2 \alpha(x_0, r). \tag{1.1}$$

The quantity $\alpha(x_0, r)$ in (1.1) is a *local PL constant*: on the ball $B(x_0, r)$, the inequality $|\nabla f(x)|^2 \geq \alpha(x_0, r) f(x)$ is exactly a local version of the Polyak–Łojasiewicz (PL) or *gradient-dominance* condition (compare, e.g., Łojasiewicz, 1963; Polyak, 1963; Karimi et al., 2016).

In particular, it implies a Kurdyka–Łojasiewicz inequality on the same ball with exponent $\frac{1}{2}$, but it is *stronger* than a generic KL inequality because it controls the objective value linearly.

Under the assumption (1.1), we have two results. The first result, stated below, shows that the gradient flow started at x_0 converges exponentially fast to a global minimum of f in $B(x_0, r)$ where f is zero. The existence of a global minimum in $B(x_0, r)$ is a part of the conclusion, and not an assumption. Since f is assumed to be nonnegative, any point x^* with $f(x^*) = 0$ achieves the minimum value of f and is therefore a (possibly non-unique) *global minimizer* of f; our theorems guarantee that such a point lies within the ball and is reached from x_0. The proof is in Appendix A.1.

Theorem 1. *Let f, x_0, and α be as above. Assume that (1.1) holds for some $r > 0$, and let $\alpha := \alpha(x_0, r)$. If $f(x_0) = 0$, the flow is stationary at x_0. Otherwise, there is a unique solution of the gradient flow equation*

$$\frac{d}{dt}\phi(t) = -\nabla f(\phi(t))$$

on $[0, \infty)$ with $\phi(0) = x_0$, and this flow stays in $B(x_0, r)$ for all time, and converges to some $x^* \in B(x_0, r)$ where $f(x^*) = 0$. Moreover, for each $t \geq 0$, we have

$$|\phi(t) - x^*| \leq re^{-\alpha t/2} \quad \text{and} \quad f(\phi(t)) \leq e^{-\alpha t} f(x_0).$$

Our second result is the analogue of Theorem 1 for gradient descent. It says that under the condition (1.1), gradient descent started at x_0, with a small enough step size, converges to a global minimum of f in $B(x_0, r)$. Again, the existence of a global minimum in $B(x_0, r)$ is a part of the conclusion. This result is proved in Appendix A.2.

Theorem 2. *Let f, x_0, and α be as above. Assume that (1.1) holds for some $r > 0$, and let $\alpha := \alpha(x_0, r)$. If $f(x_0) = 0$, gradient descent is stationary at x_0 for every $\eta > 0$. Otherwise, define*

$$\rho := \sqrt{\frac{4f(x_0)}{r^2\alpha}},$$

so that $\rho \in (0, 1)$ by (1.1). Fix any ϵ satisfying $0 < \epsilon < 1 - \rho$, and define

$$L_1 := \sup_{x \in B(x_0, r)} \max_{1 \leq i \leq p} |\partial_i f(x)| \qquad \text{and} \qquad L_2 := \sup_{x \in B(x_0, 2r)} \max_{1 \leq i,j \leq p} |\partial_i \partial_j f(x)|.$$

Choose any step size $\eta > 0$ such that

$$\eta \leq \min\left\{ \frac{r}{L_1\sqrt{p}}, \frac{2\epsilon}{L_2 p} \right\},$$

and iteratively define

$$x_{k+1} := x_k - \eta \nabla f(x_k)$$

for each $k \geq 0$. Then $x_k \in B(x_0, r)$ for all k, and as $k \to \infty$, x_k converges to a point $x^* \in B(x_0, r)$ where $f(x^*) = 0$. Moreover, with

$$\kappa := \min\{1, (1 - \epsilon)\alpha\eta\},$$

we have for each $k \geq 0$ *that*

$$|x_k - x^*| \leq (1-\kappa)^{k/2} r \quad \textit{and} \quad f(x_k) \leq (1-\kappa)^k f(x_0).$$

Note that the above results have nothing to say about variants of gradient descent, such as stochastic gradient descent. Adding a stochastic component to the gradient descent algorithm has various benefits, such as helping it escape saddle points (Jin et al., 2017). Since it is known that stochastic gradient methods often asymptotically follow the path of a differential equation (Duchi and Ruan, 2018), it would be interesting to see if analogues of Theorems 1 and 2 can be proved for stochastic gradient descent. We also do not have anything to say about algorithms that aim to find critical points instead of global minima in nonconvex problems, such as the ones surveyed in Danilova et al. (2020); Carmon et al. (2018). For a survey of the many variants of stochastic gradient descent and their applications in machine learning, see Netrapalli (2019). For a comprehensive account of all variants of gradient descent, see Ruder (2016). For some essential limitations of nonconvex optimization, see Arjevani et al. (2019).

Remark 3 (Beyond C^2 smoothness)**.** Theorems 1 and 2 are stated for C^2 objectives because our discrete-time analysis uses a second-order Taylor expansion and uniform bounds on the Hessian on a neighborhood of the iterates. For nonsmooth problems, one typically replaces the gradient by a suitable generalized gradient (for example, the limiting subdifferential) and works with a *descent lemma* for a specific algorithm (proximal gradient, forward–backward splitting, alternating minimization, etc.). Many nonsmooth objectives of interest in optimization are *weakly convex* or *prox-regular*, and many such functions satisfy a Kurdyka–Łojasiewicz (KL) inequality; see, e.g., Attouch et al. (2010); Bolte et al. (2010); Drusvyatskiy et al. (2015). A continuous-time analogue of Theorem 1 can often be obtained in that setting by combining a KL inequality with a trapping argument. Extending Theorem 2 in a comparably clean way to nonsmooth objectives would require additional machinery and is beyond the scope of this paper.

1.3 How the Criterion Will Be Used for Neural Networks

The application in Section 2 proceeds by verifying (1.1) for the squared training loss $S(w)$ of a neural network. A key observation is that for a least-squares objective of the form

$$S(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i, w))^2,$$

the ratio appearing in the definition of $\alpha(\cdot,\cdot)$ can be written in terms of the *empirical tangent kernel* (or empirical NTK)

$$H(w) := (\nabla f(x_i, w)^T \nabla f(x_j, w))_{1\leq i,j\leq n},$$

where u^T denotes the transpose of a column vector u. Indeed, letting $z(w) := (y_1 - f(x_1, w), \ldots, y_n - f(x_n, w))^T \in \mathbb{R}^n$, one checks that

$$|\nabla S(w)|^2 = \frac{4}{n^2} z(w)^T H(w) z(w) \quad \text{and} \quad S(w) = \frac{1}{n}|z(w)|^2,$$

so that

$$\frac{|\nabla S(w)|^2}{S(w)} = \frac{4}{n}\frac{z(w)^T H(w) z(w)}{|z(w)|^2} \geq \frac{4}{n}\lambda_{\min}(H(w)),$$

where $\lambda_{\min}(M)$ denotes the minimum eigenvalue of a Hermitian matrix M. Consequently, a sufficient route to (1.1) is to (i) find a lower bound for $\lambda_{\min}(H(w))$ on a ball around the initialization w_0, and (ii) choose the initialization so that $S(w_0)$ is small relative to that lower bound. In Section 2, we obtain such a lower bound in a high-dimensional data regime by controlling the first-layer gradients and using the positivity of certain products of weights.

2. Application to Deep Neural Networks

We introduce the network model and data assumptions before giving the convergence results.

2.1 Feedforward Neural Networks

A feedforward neural network has the following components:

- A positive integer L, which denotes the number of layers. It is sometimes called the depth of the network. The number $L-1$ denotes the 'number of hidden layers'. To avoid trivialities, we will assume that $L \geq 2$; i.e., there is at least one hidden layer.
- A positive integer d, called the 'input dimension'. The input data take value in $\mathbb{R}^d$.
- A sequence of positive integers $d_1, \ldots, d_L$, denoting the dimensions of layers $1, \ldots, L$. The maximum of $d_1, \ldots, d_L$ is called the width of the network. The dimension of the 'output layer', d_L, is often taken to be 1. We will henceforth take $d_L = 1$.
- A sequence of 'weight matrices' $W_1, \ldots, W_L$ with real entries, where W_ℓ has dimensions $d_\ell \times d_{\ell-1}$ (with $d_0 := d$).
- A sequence of 'bias vectors' $b_1, \ldots, b_L$, where $b_\ell \in \mathbb{R}^{d_\ell}$.
- A sequence of 'activation functions' $\sigma_1, \ldots, \sigma_L : \mathbb{R} \to \mathbb{R}$. Usually, the activation function for the output layer, σ_L, is taken to be the identity map, i.e., $\sigma_L(x) = x$. We will henceforth take σ_L to be the identity map.

The feedforward neural network with the above components is defined as follows. Denote $w := (W_1, b_1, \ldots, W_L, b_L)$, viewing it as a vector in $\mathbb{R}^p$, where

$$p := \sum_{\ell=1}^{L} d_\ell (d_{\ell-1} + 1). \tag{2.1}$$

Given a value of w, the neural network defines a map $f(\cdot, w)$ from the input space $\mathbb{R}^d$ into the output space $\mathbb{R}$, as

$$f(x, w) := \sigma_L(W_L \sigma_{L-1}(\cdots W_2 \sigma_1(W_1 x + b_1) + b_2 \cdots) + b_L),$$

where the activation functions $\sigma_1, \ldots, \sigma_L$ act componentwise on vectors of dimensions $d_1, \ldots, d_L$.

Suppose that we have a neural network as above, and some input data $x_1, \ldots, x_n \in \mathbb{R}^d$ and output data $y_1, \ldots, y_n \in \mathbb{R}$. Suppose that we want to 'fit' the model to the data by trying to minimize the squared error loss

$$S(w) := \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i, w))^2 \tag{2.2}$$

We study full-batch gradient flow and gradient descent on the training objective (2.2), namely

$$\frac{d}{dt} w(t) = -\nabla S(w(t)) \qquad \text{and} \qquad w_{k+1} = w_k - \eta \nabla S(w_k),$$

under smoothness assumptions on the activation functions that ensure S is C^2.

2.2 A High-Dimensional Data Regime

Our main convergence theorems below assume that the inputs $x_1, \ldots, x_n$ are linearly independent. In particular, this requires that the ambient input dimension satisfies $d \geq n$. This is a *different* regime from the by-now standard "over-parameterized" analyses of neural-network training, where one typically keeps d fixed and takes the *width* (and hence the number of parameters) to scale with n.

The $d \geq n$ assumption can be natural in high-dimensional statistics (small n, large d), or after applying a high-dimensional feature map/embedding to the inputs. It is also worth emphasizing what this paper does *not* claim: we do not provide a guarantee for generic modern deep-learning practice in the regime $n \gg d$. Rather, our contribution is to show that in the complementary high-dimensional setting, *width need not grow with* n in order for gradient descent to provably reach an interpolating solution.

In fact, linear independence of the inputs already implies strong interpolation capacity for very small networks. For example, when $L = 2$ and $d_1 = 1$, the map $x \mapsto f(x, w)$ is of the form $x \mapsto a\sigma_1(u^T x)$. In the centered-activation setting of Theorem 4, the range of σ_1 contains an interval about zero. Choose $a > 0$ large enough that each y_i/a belongs to this interval, and choose t_i with $\sigma_1(t_i) = y_i/a$. Since $X := [x_1\ x_2\ \cdots\ x_n]$ has full column rank, one can solve $u^T x_i = t_i$ simultaneously, yielding exact interpolation. Our results show that, for *any* fixed depth L and fixed hidden-layer dimensions $d_1, \ldots, d_{L-1}$, gradient descent can be initialized so as to find such an interpolating solution, and does so with a linear rate.

2.3 Our Results for Feedforward Neural Networks

There is an enormous body of work on optimization for neural networks. We only summarize the strands that are the most relevant for the results stated below, and refer to the survey by Sun (2019) (and its references) for broader background. For convex and linear models, convergence results for gradient-type methods are classical; for example, convex neural networks were analyzed in Bach and Moulines (2013); Bach (2017); Moulines and Bach (2011), and deep linear networks in Arora et al. (2018); Bartlett et al. (2018); Ji and Telgarsky (2018). For general (nonconvex, nonlinear) networks, many recent global-convergence guarantees are proved in an over-parameterized regime where the hidden layers are very wide and the empirical tangent kernel $H(w)$ is well conditioned and remains stable along training.

Representative results include Du et al. (2018); Allen-Zhu et al. (2019); Li and Liang (2018); Wu et al. (2018); E et al. (2020); Soltanolkotabi et al. (2018); Hardt and Ma (2016); Du et al. (2019); Jacot et al. (2018); Zou et al. (2020); Cao and Gu (2019, 2020); Sankararaman et al. (2020); Liu et al. (2022); see also the discussion in Jentzen and Riekert (2021). There are also works pushing beyond the "infinite-width" or NTK regime and obtaining convergence under milder (but still width-dependent) conditions, for example Jentzen and Riekert (2021); Robin and Lelarge (2022).

Our theorems below are complementary to the results cited above: we do not assume width scales with n, and our lower bound on $\lambda_{\min}(H(w))$ comes instead from the high-dimensional structure of the data (linear independence) together with an explicit initialization that prevents sign cancellations in the relevant products of weights. Technically, the proofs still proceed by verifying a local PL inequality for S via a lower bound on $\lambda_{\min}(H(w))$, but the mechanism is different from the typical over-parameterized stability arguments.

Finally, because our analysis is based on a PL/KL-type inequality, it is related in spirit to a classical line of work that uses (Kurdyka–) Łojasiewicz gradient inequalities to study convergence of gradient-like dynamics; see, e.g., Łojasiewicz (1963); Absil et al. (2005); Attouch et al. (2010); Bolte et al. (2010); Drusvyatskiy et al. (2015). More recently, Łojasiewicz-type inequalities have also been used to connect optimization dynamics with generalization in learning problems; see, for example, Liu et al. (2022).

Our first result about feedforward neural networks is the following. The proof is in Appendix A.3.

Theorem 4. *Consider a feedforward neural network with depth $L \geq 2$, output dimension $d_L = 1$, and $\sigma_L =$ identity. Assume that the activation functions $\sigma_1, \ldots, \sigma_{L-1}$ are twice continuously differentiable, satisfy $\sigma_\ell(0) = 0$, and are strictly increasing in the sense that $\sigma_\ell'(x) > 0$ for all x and ℓ. Given input data $x_1, \ldots, x_n \in \mathbb{R}^d$ and output data $y_1, \ldots, y_n \in \mathbb{R}$, define the squared loss S as in (2.2). Assume that the input vectors $x_1, \ldots, x_n$ are linearly independent, and let*

$$\lambda_X := \lambda_{\min}\left(\frac{1}{n} X^T X\right) > 0,$$

where X is the $d \times n$ matrix whose i^{th} column is x_i. Let $w_0 = (W_1, b_1, \ldots, W_L, b_L)$ be an initialization such that $b_1 = \cdots = b_L = 0$, $W_1 = 0$, and every entry of each hidden-layer weight matrix $W_2, \ldots, W_{L-1}$ is strictly positive. For $L \geq 3$, define

$$\delta := \min_{2 \leq \ell \leq L-1} \min_{i,j} (W_\ell)_{ij} \quad \text{and} \quad K := \max_{2 \leq \ell \leq L-1} \max_{i,j} (W_\ell)_{ij}.$$

When $L = 2$, set $\delta = K = 1$ by convention. Write

$$R_X := \max_{1 \leq i \leq n} |x_i|, \qquad Y_2 := \frac{1}{n} \sum_{i=1}^{n} y_i^2.$$

There is an explicit constant $A_0 > 0$ (see Remark 9) depending only on

$$\lambda_X,\ R_X,\ \delta,\ K,\ Y_2,\ L,\ d_1, \ldots, d_{L-1}, \quad (\sigma_\ell)_{\ell=1}^{L-1},$$

such that, for every choice of W_L whose entries are at least A_0:

- *the gradient flow* $\dot{w}(t) = -\nabla S(w(t))$ *with* $w(0) = w_0$ *converges to a point* w^* *with* $S(w^*) = 0$*; and*
- *there is a step-size bound* $\eta_0(w_0) > 0$ *such that gradient descent* $w_{k+1} = w_k - \eta \nabla S(w_k)$ *converges to a point* w^* *with* $S(w^*) = 0$ *whenever* $0 < \eta \leq \eta_0(w_0)$*.*

The bound $\eta_0(w_0)$ *may depend on the full data, the architecture, the activation functions, and the actual initialization* w_0*, including the entries of* W_L*. No uniform positive step-size bound over all unbounded choices of* W_L *is asserted.*

Remark 5 (Width versus ambient dimension)**.** Recall that the width of the network is $m := \max\{d_1, \ldots, d_L\}$ and does *not* include the input dimension $d_0 = d$. Theorem 4 works in the high-dimensional regime $d \geq n$ (linear independence of the inputs), but places *no* growth requirement on m as a function of n. This is in contrast with much of the over-parameterized literature, where m is taken to scale at least linearly (and often polynomially) with n; see, e.g., Soltanolkotabi et al. (2018); Du et al. (2018, 2019).

Remark 6 (Why the condition $W_1 = 0$ is not vacuous)**.** A possible source of confusion is the condition $W_1 = 0$ (and $b_1 = \cdots = b_L = 0$) in Theorem 4. At the initialization w_0 the network output is indeed constant (in fact, $f(x, w_0) = 0$ for all x because $\sigma_\ell(0) = 0$), so $S(w_0) = \frac{1}{n}\sum_i y_i^2$. Unless all labels are zero, the loss gradient at w_0 is *not* zero. The first-layer derivatives of the network output satisfy (see (A.5)–(A.6) in Appendix A.3)

$$\frac{\partial}{\partial (W_1)_{rs}} f(x, w_0) = x_s q_r(x, w_0),$$

where $q_r(x, w_0)$ is a sum of products of downstream weights and activation slopes. At w_0 it is positive and independent of x. Consequently, the gradient of S with respect to row r of W_1 is $-(2/n) q_r(w_0) \sum_i y_i x_i$, which is nonzero whenever the labels are not all zero, by linear independence of the inputs. The role of $W_1 = 0$ is simply to provide a convenient center point around which we can control the empirical tangent kernel $H(w)$.

Remark 7 (Why require hidden-layer weights to be positive?)**.** The proof goes by finding a lower bound on $\lambda_{\min}(H(w))$ in a neighborhood of the initialization by controlling certain vectors $q_r(x_i, w)$ that appear in the first-layer gradients. If the hidden-layer weights have mixed signs, then products of weights can cancel and $q_r(x_i, w)$ can be arbitrarily small even when individual weights are not. Requiring $W_2, \ldots, W_{L-1}$ to have strictly positive entries rules out such cancellations and yields a clean lower bound. We emphasize that this is a technical assumption used for a worst-case guarantee; it is not meant to model typical random initialization.

Remark 8 (Activation shifts, biases, and smooth leaky-ReLU)**.** The class of activation functions allowed by Theorem 4 includes many smooth monotone activations used in practice (e.g., tanh and smooth sigmoids). The condition $\sigma_\ell(0) = 0$ is not restrictive: if an activation σ does not satisfy $\sigma(0) = 0$, one can replace it by the shifted activation $\widetilde{\sigma}(x) := \sigma(x) - \sigma(0)$ and absorb the resulting constant offset into the biases. This preserves the represented function class; the convergence statements concern gradient descent in the centered-activation parameterization with the specified zero initial biases. The theorem also applies to smooth

approximations of ReLU and leaky-ReLU. One convenient "smooth leaky-ReLU" activation with parameter $a \in (0,1)$ is

$$\sigma(x) = ax + (1-a)\log(1+e^x) - (1-a)\log 2,$$

which satisfies $\sigma(0) = 0$ and has derivative bounded between a and 1. This is the activation used in the experiments in Section 3 below.

Remark 9 (Explicit constants in Theorem 4)**.** Let $r := \delta/2$ and $K' := K + r$, with the convention from Theorem 4 when $L = 2$. For $x \geq 0$, put

$$\gamma_\ell(x) := \max\{\sigma_\ell(x), |\sigma_\ell(-x)|\}.$$

Using R_X from the theorem, define the hidden-layer pre-activation bounds

$$a_1 := r(R_X + 1), \qquad a_\ell := K' d_{\ell-1}\gamma_{\ell-1}(a_{\ell-1}) + r, \quad 2 \leq \ell \leq L-1,$$

and set

$$c_\ell := \min_{|u| \leq a_\ell} \sigma_\ell'(u) > 0, \qquad P := \prod_{\ell=2}^{L-1} d_\ell, \qquad C := \prod_{\ell=1}^{L-1} c_\ell.$$

Empty products are interpreted as 1. An explicit, unoptimized choice is

$$A_0 := r + 1 + 2\left(\frac{Y_2}{\lambda_X d_1 r^{2L-2}(PC)^2}\right)^{1/2}. \tag{2.3}$$

For a chosen initialization, let $A := \min_j (W_L)_{1j} \geq A_0$. The proof gives

$$\alpha(w_0, r) \geq \underline{\alpha} := 4\lambda_X d_1 (A - r)^2 r^{2L-4}(PC)^2,$$

where α is defined for the objective S. Since $S(w_0) = Y_2$, this choice yields $4S(w_0)/(r^2\underline{\alpha}) < 1/4$ if $Y_2 > 0$. Thus $\epsilon = 1/4$ is admissible in Theorem 2. Writing

$$L_1(w_0) := \sup_{w \in B(w_0, r)} \max_j |\partial_j S(w)|,$$
$$L_2(w_0) := \sup_{w \in B(w_0, 2r)} \max_{j,k} |\partial_j \partial_k S(w)|,$$

one may take

$$\eta_0(w_0) := \min\left\{\frac{r}{L_1(w_0)\sqrt{p}}, \frac{1}{2L_2(w_0)p}\right\}. \tag{2.4}$$

The derivative bounds are evaluated around the actual initialization and therefore depend on its output weights. If $Y_2 = 0$, the initialization already has zero loss and both dynamics are stationary; any positive step size then suffices.

Remark 10 (Relation to "lazy training" results)**.** A number of general "lazy training" results show that, under suitable scaling assumptions, gradient flow remains close to initialization and behaves like kernel regression with the empirical tangent kernel; see, e.g., Chizat et al. (2019). Our proof is directly in this spirit: we lower bound $\lambda_{\min}(H(w))$ on a neighborhood of an explicit initialization and then apply the local PL criterion from Section 1. Because our scaling and initialization are quite different from the standard large-width settings, it is not immediate to deduce Theorem 4 as a black-box corollary of existing lazy-training theorems, and we therefore provide a self-contained argument.

Finally, note that Theorem 4 concerns *training error*: it guarantees convergence to a parameter w^* with $S(w^*) = 0$ (an interpolating global minimizer of S). It does not by itself imply any *generalization* guarantee.

Our second result about feedforward neural networks, stated below, shows that when initiated from a standard random initialization, the flow converges to a global minimum with positive probability. The proof is in Appendix A.4.

Theorem 11 (A conditional corollary for LeCun Gaussian initialization)**.** *Consider the architecture, data, and activation assumptions of Theorem 4, but use the random initialization specified below. Assume additionally that*

$$0 < C_1 \le \sigma_\ell'(x) \le C_2 < \infty \quad \text{for all } x \in \mathbb{R}, \quad 1 \le \ell \le L-1.$$

Let λ_X and Λ_X be the minimum and maximum eigenvalues of $X^T X/n$, and write $Y_2 = n^{-1}\sum_i y_i^2$. Define

$$\begin{aligned} P &:= \prod_{\ell=2}^{L-1} d_\ell, \qquad a := 4\lambda_X d_1 C_1^{2L-2} P^2, \\ b &:= 3^{L-1} C_2^{L-1} \sqrt{d_1}\, P, \qquad B := 2\Lambda_X b^2, \\ \beta &:= \sqrt{\frac{a}{16B}} = \sqrt{\frac{\lambda_X}{8\Lambda_X}} \left(\frac{C_1}{3C_2}\right)^{L-1}. \end{aligned} \tag{2.5}$$

As before, an empty product is 1. Fix $c > 0$ and choose

$$M \ge \max\left\{1, \left(\frac{32 Y_2}{a}\right)^{1/(2L)}\right\}. \tag{2.6}$$

Initialize all biases to zero and draw the entries of each W_ℓ independently from $\mathcal{N}(0, c/d_{\ell-1})$. Set

$$\begin{aligned} E_M &:= \{(W_\ell)_{ij} \in [2M, 3M] \text{ for all } \ell = 2, \dots, L \text{ and all } i, j\}, \\ F_M &:= \{\|W_1\|_F \le \beta M\}, \end{aligned} \tag{2.7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. On $E_M \cap F_M$, gradient flow converges to a point w^* with $S(w^*) = 0$. There is also a deterministic $\eta_0 > 0$, depending on the fixed data, architecture, activation functions, C_1, C_2, and M, such that gradient descent with any $0 < \eta \le \eta_0$ converges to such a point on this same event. This step-size bound can be chosen uniformly over the event, for these fixed data and functions. The event $E_M \cap F_M$ has strictly positive probability.

Remark 12 (Interpretation of Theorem 11)**.** The first-layer threshold is βM, rather than M. Appendix A.4 verifies that this choice and (2.6) imply the trapping inequality with a uniform margin. Independence of the weight matrices gives

$$\begin{aligned} \mathbb{P}(E_M \cap F_M) = \mathbb{P}&\left(\chi^2_{dd_1} \le \frac{d\beta^2 M^2}{c}\right) \\ &\times \prod_{\ell=2}^{L} \left[\Phi\left(3M\sqrt{d_{\ell-1}/c}\right) - \Phi\left(2M\sqrt{d_{\ell-1}/c}\right)\right]^{d_\ell d_{\ell-1}} > 0, \end{aligned}$$

where Φ is the standard normal distribution function and χ^2_k denotes a chi-squared random variable with k degrees of freedom. This is an explicit lower bound on the probability of convergence. It can be very small, even for small fixed networks if the required M is large. Thus the theorem gives a positive-probability guarantee, not a high-probability guarantee for typical Gaussian initialization. As M tends to infinity, $\mathbb{P}(F_M)$ tends to 1 while the downstream positive-box probability tends to 0.

3. Numerical Experiments

One appealing aspect of the neural network argument is that it produces a concrete initialization strategy rather than only an abstract convergence guarantee. In this section we illustrate this "theory-guided initialization" numerically on small synthetic examples. The experiments are not meant as performance benchmarks; rather, they show that the proof-motivated starting point can have a visible practical impact on optimization behavior at fixed (and relatively small) width.

3.1 Setup

We trained a depth-3 network (two hidden layers) with the smooth leaky-ReLU activation from Remark 8 with $a = 0.1$. We used full-batch gradient descent on (2.2) with step size $\eta = 10^{-3}$ for 4000 iterations, and we report the mean training loss over five random seeds.

3.2 Initializations Compared

We compared: (i) the deterministic "positive" initialization suggested by Theorem 4 (set $W_1 = 0$, set each entry of W_2 to $\delta = 0.5$, and set each entry of W_3 to $A = 1$); (ii) a standard LeCun Gaussian initialization (all weights i.i.d. $\mathcal{N}(0, 1/d_{\ell-1})$); (iii) a simple positive variant where the Gaussian weights are replaced by their absolute values (entrywise); and (iv) a "conditional-event" variant where W_2, W_3 are sampled from the LeCun Gaussian *conditioned* to lie entrywise in a positive box $[2M, 3M]$ (equivalently, sampled from a truncated Gaussian supported on $[2M, 3M]$), mimicking the event discussed around Theorem 11.

3.3 Baseline Result (Width $d_1 = d_2 = 5$)

Figure 1 plots the mean training loss (over five random seeds) versus iteration on a log scale. The theory-guided "positive" initialization exhibits a clear linear rate over several orders of magnitude. By contrast, at this fixed width the standard Gaussian initializations do not consistently reach zero training error, consistent with the discussion around Theorem 11: the random-initialization guarantee there is conditional on an event that may have very small probability at finite width. (The conditional-event curve (iv) shows that, *given* the required positivity event, behavior is much closer to the deterministic positive initialization.)

3.4 Wider Network

To see how behavior changes with width, we repeated the same experiment with widths $d_1 = d_2 = 10$. Figure 2 shows that the qualitative picture persists: the proof-motivated

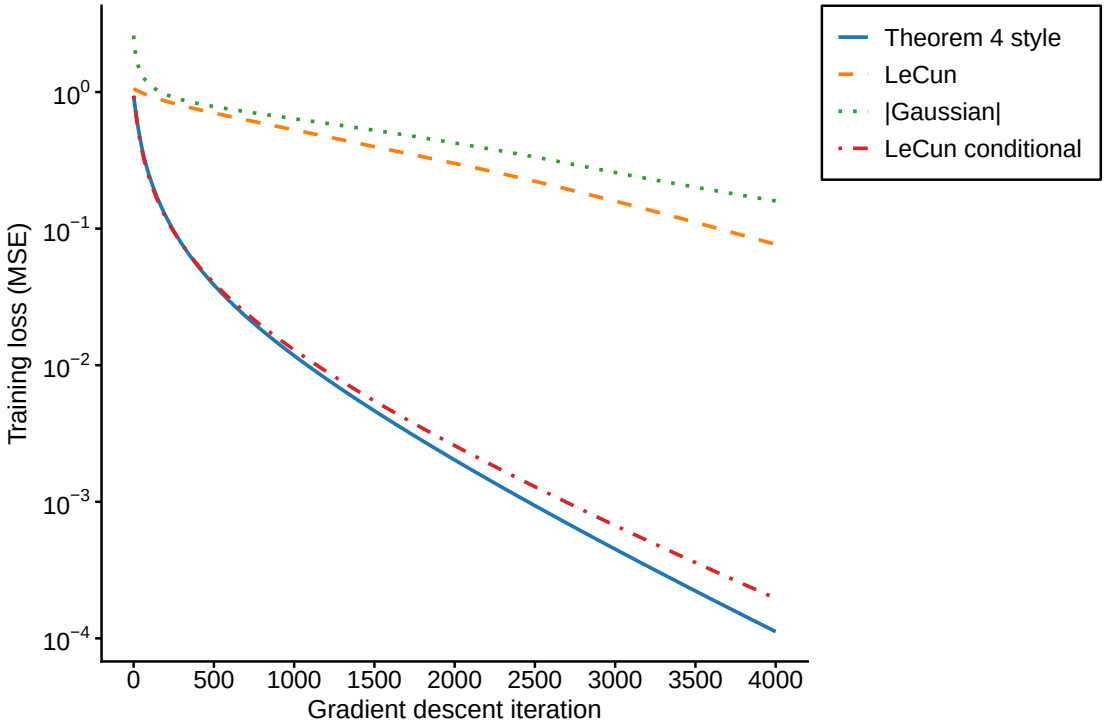


Figure 1: Mean training loss (MSE) versus gradient descent iteration in the baseline experiment. The y-axis is logarithmic. The legend is placed outside the plotting region and the curves use distinct line types so that color is not essential.

positive initialization remains reliably fast, while random initializations improve with width but can still be noticeably slower at these sizes.

3.5 Output-Layer Scaling

Theorem 4 requires the output-layer weights to be "sufficiently large." To visualize this dependence, we fixed the positive initialization pattern and swept the output-layer scale A over a range of values. Figure 3 shows a clear effect of the scale on optimization speed: once A is large enough, the loss decays rapidly and approximately linearly on a log scale, whereas smaller A can substantially slow down progress. This provides a simple example of how a convergence proof can translate into a practically useful initialization guideline.

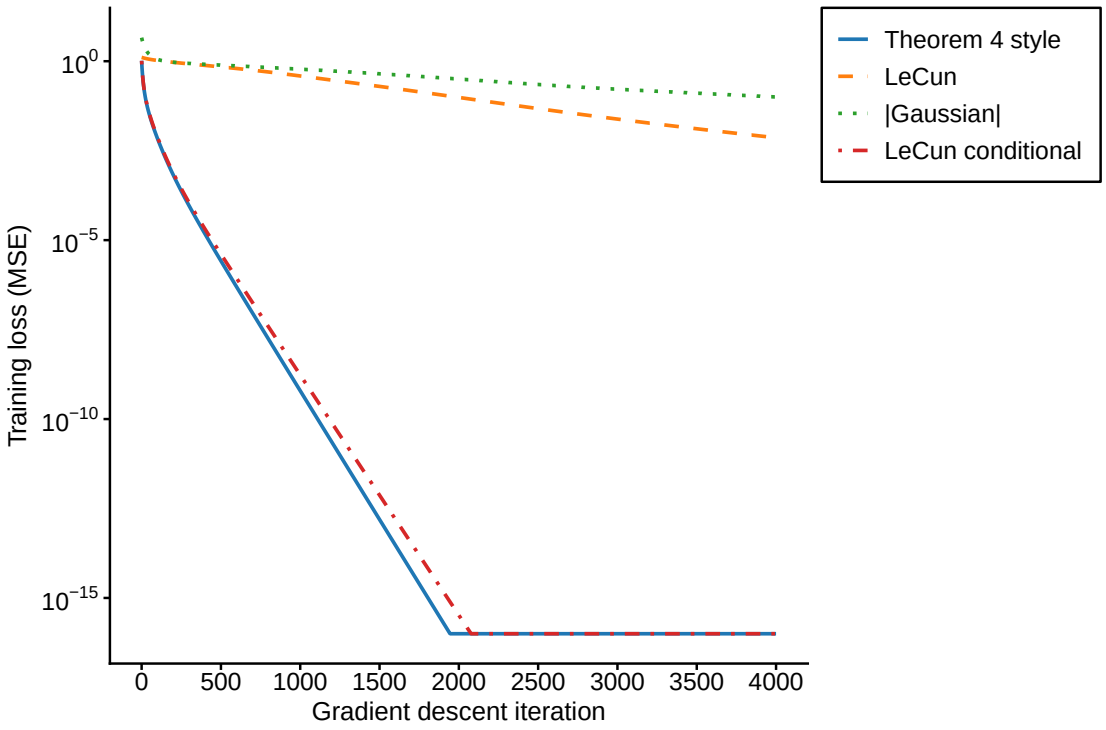


Figure 2: Mean training loss (MSE) versus gradient descent iteration for the same synthetic setup, but with widths $d_1 = d_2 = 10$. The y-axis is logarithmic.

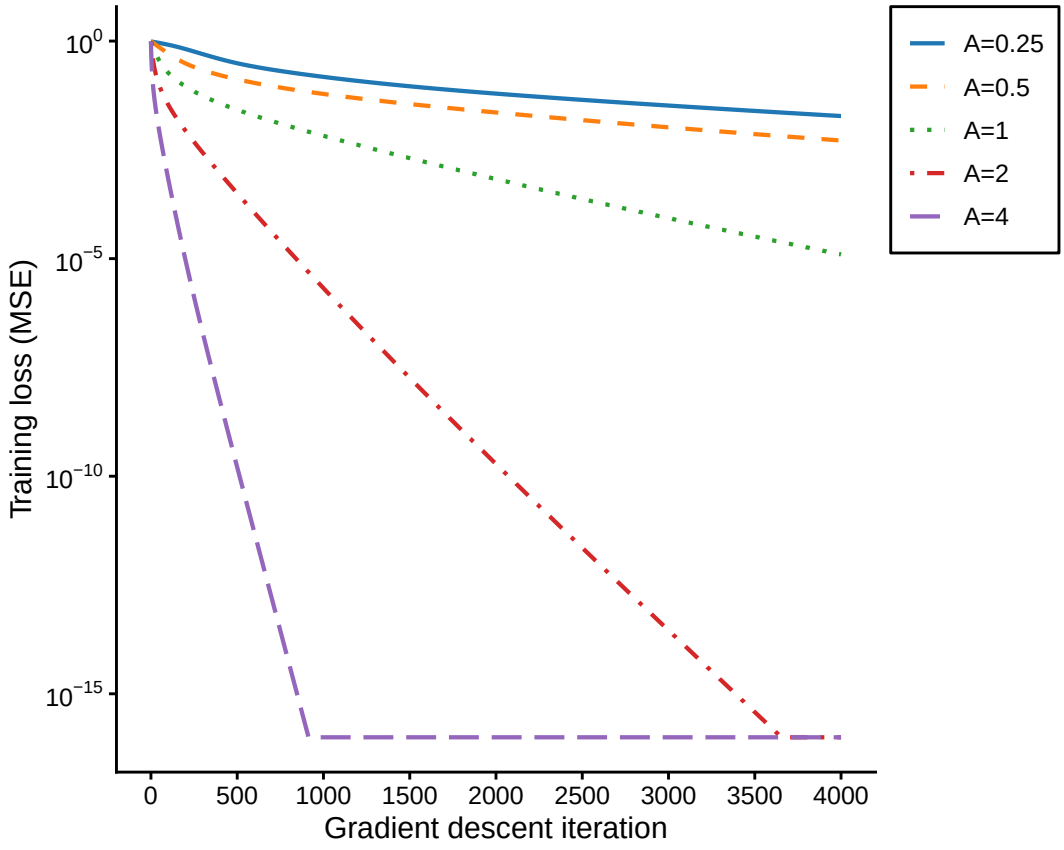


Figure 3: Mean training loss (MSE) versus iteration for the positive initialization with different output-layer scales A. The y-axis is logarithmic.

Acknowledgments

I thank Persi Diaconis, Dmitriy Drusvyatskiy, John Duchi, and Lexing Ying for useful feedback, references, and suggestions. The research was partially supported by NSF grant DMS-2413864. The author declares no competing interests.

Appendix A. Proofs

This appendix contains the proofs of the convergence theorems.

A.1 Proof of Theorem 1

To avoid trivialities, let us assume that f is not everywhere zero in $B(x_0, r)$, and hence $\alpha < \infty$. Throughout this proof, we will denote the closed ball $B(x_0, r)$ by B and its interior by U. We start with some supporting lemmas. Many of these are standard results, but we prove them anyway for the sake of completeness and to save the trouble of sending the reader to look at references. For each $x \in \mathbb{R}^p$, let T_x denote the supremum of all finite T such that for all $S \le T$, there is a unique continuous function $\phi_x : [0, S] \to \mathbb{R}^p$ satisfying the integral equation

$$\phi_x(t) = x - \int_0^t \nabla f(\phi_x(s))ds.$$

Since there is always such a unique function for $T = 0$ (namely, $\phi_x(0) = x$), T_x is well-defined. From the above definition, we see that there is a unique continuous function $\phi_x : [0, T_x) \to \mathbb{R}^p$ satisfying the above integral equation. We will refer to this ϕ_x as the gradient flow starting from x.

Lemma 13. *Take any $x \in \mathbb{R}^p$ and any $t \in [0, T_x)$ (assuming that $T_x > 0$). Let $y = \phi_x(t)$. Then $T_y = T_x - t$ (with the convention $\infty - t = \infty$), and $\phi_y(s) = \phi_x(t + s)$ for all $s \in [0, T_y)$.*

Proof. Take any $S \in [0, T_x - t)$. Define $g(s) := \phi_x(t + s)$ for $s \in [0, S]$. Then

$$\begin{aligned} g(s) = \phi_x(t + s) &= x - \int_0^{t+s} \nabla f(\phi_x(u))du \\ &= \phi_x(t) - \int_t^{t+s} \nabla f(\phi_x(u))du = y - \int_0^s \nabla f(g(v))dv. \end{aligned}$$

Thus, g satisfies the gradient flow integral equation starting from y, in the interval $[0, S]$. Let h be any other flow with these properties. Define $w : [0, t + S] \to \mathbb{R}^p$ as

$$w(u) := \begin{cases} \phi_x(u) & \text{if } u \le t, \\ h(u - t) & \text{if } u > t. \end{cases}$$

Then for $u \in [t, t+S]$,

$$\begin{aligned} w(u) = h(u-t) &= y - \int_0^{u-t} \nabla f(h(v))dv \\ &= \phi_x(t) - \int_0^{u-t} \nabla f(w(t+v))dv \\ &= x - \int_0^t \nabla f(w(s))ds - \int_t^u \nabla f(w(s))ds \\ &= x - \int_0^u \nabla f(w(s))ds. \end{aligned}$$

Obviously, w satisfies this integral equation also in $[0,t]$. Thus, the uniqueness of the flow in $[0, t+S]$ implies that $w = \phi_x$ in $[0, t+S]$. This implies, in particular, that $h = g$. Therefore, we conclude that $T_y \geq T_x - t$. This also shows that $\phi_y(s) = \phi_x(t+s)$ for all $s \in [0, T_x - t)$. So it only remains to show that $T_y \leq T_x - t$. Suppose not. Then T_x must be finite, and there is some $S > T_x - t$ such that the gradient flow integral equation starting from y has a unique solution in $[0, L]$ for every $L \leq S$. Take any $L \leq S$. Define $g : [0, t+L] \to \mathbb{R}^p$ as

$$g(s) := \begin{cases} \phi_x(s) & \text{if } s \leq t, \\ \phi_y(s-t) & \text{if } s > t. \end{cases}$$

Then for any $s \in [t, t+L]$,

$$\begin{aligned} g(s) &= y - \int_0^{s-t} \nabla f(\phi_y(u))du \\ &= x - \int_0^t \nabla f(g(u))du - \int_0^{s-t} \nabla f(g(u+t))du \\ &= x - \int_0^s \nabla f(g(u))du. \end{aligned}$$

Thus, g satisfies the integral equation for the gradient flow starting from x in the interval $[t, t+L]$. Obviously, it satisfies the equation also in $[0,t]$. Suppose that $h : [0, t+L] \to \mathbb{R}^p$ is any other continuous function satisfying this property. Then by the uniqueness of the gradient flow starting from x in the interval $[0,t]$, we get that $h(s) = \phi_x(s) = g(s)$ for all $s \leq t$. Next, define $h_1 : [0, L] \to \mathbb{R}^p$ as $h_1(u) := h(t+u)$. Then, since $h = \phi_x$ on $[0,t]$,

$$\begin{aligned} h_1(u) &= h(t+u) \\ &= x - \int_0^{t+u} \nabla f(h(v))dv \\ &= y - \int_t^{t+u} \nabla f(h(v))dv = y - \int_0^u \nabla f(h_1(s))ds. \end{aligned}$$

Thus, h_1 satisfies the integral equation for the gradient flow starting from y in the interval $[0, L]$. Thus, by the uniqueness assumption for this flow in $[0, L]$, we get that $h_1 = \phi_y$ in this interval. Consequently, $h = g$ in $[0, t+L]$. Thus, the gradient flow starting from x has a unique solution in $[0, T]$ for every $T \leq t+S$. Since $t+S > T_x$, this contradicts the definition of T_x. ■

Lemma 14. *For any compact set $K \subseteq \mathbb{R}^p$, $\inf_{x\in K} T_x > 0$.*

Proof. Let K' be the set of all points that are within distance 1 from K. Note that $K' \supseteq K$ and K' is compact. Since f is in C^2, its first and second order derivatives are uniformly bounded on the compact convex hull of K'. Thus, there is some L such that for any $x, y \in K'$,

$$|\nabla f(x)| \le \frac{L}{2}, \quad \text{and} \quad |\nabla f(x) - \nabla f(y)| \le \frac{L}{2}|x-y|,$$

where $|\cdot|$ denotes Euclidean norm. Take any $x \in K$ and any $T \le 1/L$. Let $\mathcal{B}$ denote the Banach space of all continuous maps from $[0,T]$ into $\mathbb{R}^p$, equipped with the norm

$$\|g\| := \sup_{t\in[0,T]} |g(t)|.$$

Let $\mathcal{A}$ be the subset of $\mathcal{B}$ consisting of all g such that $g(0) = x$ and $g(t) \in K'$ for all $t \in [0,T]$. It is easy to see that $\mathcal{A}$ is a closed subset of $\mathcal{B}$. Define a map $\Phi : \mathcal{A} \to \mathcal{B}$ as

$$\Phi(g)(t) := x - \int_0^t \nabla f(g(s))ds.$$

Then note that for any $t \in [0,T]$,

$$|\Phi(g)(t) - x| \le \int_0^t |\nabla f(g(s))|ds \le \int_0^t \frac{L}{2}ds \le \frac{1}{2},$$

where the second inequality holds because $g(s) \in K'$ for all s, and the third inequality holds because $t \le 1/L$. Since all points at distance ≤ 1 from x are in K', this shows that $\Phi(\mathcal{A}) \subseteq \mathcal{A}$. Next, note that for any $g, h \in \mathcal{A}$, and any $t \in [0,T]$,

$$\begin{aligned}
|\Phi(g)(t) - \Phi(h)(t)| &\le \int_0^t |\nabla f(g(s)) - \nabla f(h(s))|ds \\
&\le \frac{L}{2}\int_0^t |g(s) - h(s)|ds \\
&\le \frac{1}{2}\|g - h\|,
\end{aligned}$$

where the second inequality holds because $g(s), h(s) \in K'$, and the third inequality holds because $t \le 1/L$. This proves that Φ is a contraction mapping on $\mathcal{A}$, and therefore, by the Banach fixed point theorem, it has a unique fixed point $g^* \in \mathcal{A}$. Then g^* is a continuous map from $[0,T]$ into $\mathbb{R}^p$, that satisfies the integral equation

$$g^*(t) = x - \int_0^t \nabla f(g^*(s))ds.$$

We claim that g^* is the only such map. To prove this, suppose that there exists another map h with the above properties. If h maps into K', then the uniqueness of the fixed point implies that $h = g^*$. So, suppose that h ventures outside K'. Let

$$t_0 := \inf\{t : h(t) \notin K'\}.$$

Since $h(0) = x \in K'$ and h goes outside K', t_0 is well-defined and finite. Moreover, since K' is closed, $h(t_0) \in \partial K'$. But note that since $h(s) \in K'$ for all $s \le t_0$,

$$|h(t_0) - x| \le \int_0^{t_0} |\nabla f(h(s))| ds \le \frac{Lt_0}{2} \le \frac{1}{2}.$$

But this implies that the ball of radius $1/3$ centered at $h(t_0)$ is completely contained in K', and hence, $h(t_0) \notin \partial K'$. Thus, h cannot venture outside K'. We conclude that $T_x \ge 1/L$. ■

The above lemma has several useful corollaries.

Corollary 15. *If $T_x < \infty$ for some $x \in \mathbb{R}^p$, then $|\phi_x(t)| \to \infty$ as $t \to T_x$.*

Proof. Suppose that $|\phi_x(t)| \not\to \infty$. Then there is a compact set K such that for any $\epsilon > 0$, we can find $t \in [T_x - \epsilon, T_x)$ with $\phi_x(t) \in K$. By Lemma 14, we can find $\epsilon \in (0, \inf_{y \in K} T_y)$. Take any $t \in [T_x - \epsilon, T_x)$ such that $y := \phi_x(t) \in K$. Then, by Lemma 13, we have $T_y = T_x - t < \epsilon$. But this contradicts the fact that $T_y \ge \inf_{z \in K} T_z > \epsilon$. ■

Corollary 16. *If $\nabla f(x) = 0$ or $f(x) = 0$ for some $x \in \mathbb{R}^p$, then $T_x = \infty$ and $\phi_x(t) = x$ for all $t \in [0, \infty)$.*

Proof. If $f(x) = 0$, then since f is a nonnegative C^2 function, $\nabla f(x)$ must also be zero. Thus, it suffices to prove the result under the assumption that $\nabla f(x) = 0$. Taking $K = \{x\}$ in Lemma 14, we get that $T_x > 0$. Suppose that T_x is finite. Then let $t := T_x/2$. Note that since $\nabla f(x) = 0$, $g(s) \equiv x$ is a solution of the gradient flow equation starting from x in the interval $[0, t]$. By uniqueness, this shows that $\phi_x(s) = x$ for all $s \in [0, t]$. In particular, $\phi_x(t) = x$. By Lemma 13, this implies that $T_x = T_x - t = T_x/2$, which contradicts the fact that T_x is nonzero and finite. Thus, $T_x = \infty$. Again, the function $g(t) \equiv x$ is a solution of the flow equation in $[0, \infty)$. Thus, by uniqueness, $\phi_x(t) = x$ for all $t \ge 0$. ■

Corollary 17. *Take any $x \in \mathbb{R}^p$. If for some $t \in [0, T_x)$, we have $\nabla f(\phi_x(t)) = 0$ or $f(\phi_x(t)) = 0$, then $T_x = \infty$ and $\phi_x(s) = \phi_x(t)$ for all $s > t$.*

Proof. Take any t such that $\nabla f(\phi_x(t)) = 0$ or $f(\phi_x(t)) = 0$. Let $y := \phi_x(t)$. Then by Lemma 13, $T_y = T_x - t$. But by Corollary 16, $T_y = \infty$. Thus, $T_x = \infty$. Also by Corollary 16, $\phi_y(u) = y = \phi_x(t)$ for all $u > 0$, and by Lemma 13, $\phi_y(u) = \phi_x(t + u)$ for all $u > 0$. Thus, $\phi_x(s) = \phi_x(t)$ for all $s > t$. ■

In the following, let us fix x_0 and r as in Theorem 1 and write ϕ and T instead of ϕ_{x_0} and T_{x_0}, for simplicity of notation.

Lemma 18. *If $T < \infty$, then ϕ must visit the boundary of B.*

Proof. Suppose that $T < \infty$ and ϕ remains in U throughout. By Lemma 14, there is some $\delta > 0$ such that $T_y \ge \delta$ for all $y \in B$. Choose $t \in (T - \delta, T)$. Let $z := \phi(t)$. Then $z \in U$, and therefore $T_z \ge \delta$. This shows that the gradient flow starting from x_0 exists up to time at least $t + \delta$. Since $t + \delta > T$, this gives a contradiction which proves that ϕ cannot remain in U throughout. ■

Lemma 19. *Let $t \geq 0$ be any number such that $\phi(s) \in B$ for all $s \leq t$. Then for all $s \leq t$,*

$$f(\phi(s)) \leq e^{-\alpha s} f(x_0).$$

Proof. Note that by the flow equation for ϕ,

$$\frac{d}{ds} f(\phi(s)) = \nabla f(\phi(s)) \cdot \phi'(s) = -|\nabla f(\phi(s))|^2. \tag{A.1}$$

By the definition of α, the right side is bounded above by $-\alpha f(\phi(s))$ if $\phi(s) \in B$. It is now a standard exercise to deduce the claimed inequality. ∎

Lemma 20. *The flow ϕ cannot visit the boundary of B.*

Proof. Suppose that ϕ does visit the boundary of B. Let $t_0 := \inf\{t : \phi(t) \in \partial B\}$. Then $\phi(t_0) \in \partial B$ and $\phi(s) \in U$ for all $s < t_0$. By the flow equation,

$$|\phi(t_0) - x_0| \leq \int_0^{t_0} |\nabla f(\phi(s))| ds. \tag{A.2}$$

Now, since $\phi(s) \neq \phi(t_0)$ for all $s < t_0$, Corollary 17 shows that $f(\phi(s)) \neq 0$ for all $s < t_0$. Thus, the map

$$g(s) := \sqrt{f(\phi(s))}$$

is differentiable in $(0, t_0)$, with continuous derivative

$$g'(s) = \frac{1}{2g(s)} \frac{d}{ds} f(\phi(s)) = -\frac{|\nabla f(\phi(s))|^2}{2g(s)},$$

where the second identity follows from (A.1). Thus, for any compact interval $[a, b] \subseteq (0, t_0)$,

$$\int_a^b \frac{|\nabla f(\phi(s))|^2}{2g(s)} ds = -\int_a^b g'(s) ds = g(a) - g(b).$$

Since g is continuous on $[0, t_0]$, we can take $a \to 0$ and $b \to t_0$, and apply the monotone convergence theorem on the left, to get

$$\int_0^{t_0} \frac{|\nabla f(\phi(s))|^2}{2g(s)} ds = g(0) - g(t_0) \leq g(0) = \sqrt{f(x_0)}.$$

Therefore, by (A.2) and the Cauchy–Schwarz inequality,

$$\begin{aligned} |\phi(t_0) - x_0| &\leq \left(\int_0^{t_0} 2g(s) ds \int_0^{t_0} \frac{|\nabla f(\phi(s))|^2}{2g(s)} ds \right)^{1/2} \\ &\leq f(x_0)^{1/4} \left(\int_0^{t_0} 2g(s) ds \right)^{1/2}. \end{aligned}$$

On the other hand, since $\phi(s) \in B$ for all $s \leq t_0$, Lemma 19 shows that

$$g(s) \leq e^{-\alpha s/2} \sqrt{f(x_0)}$$

for all $s \le t_0$. Plugging this bound into the previous display, we get

$$\begin{aligned}|\phi(t_0) - x_0| &\le \sqrt{f(x_0)}\Big(\int_0^{t_0} 2e^{-\alpha s/2} ds\Big)^{1/2} \\ &\le \sqrt{f(x_0)}\Big(\int_0^{\infty} 2e^{-\alpha s/2} ds\Big)^{1/2} = 2\sqrt{\frac{f(x_0)}{\alpha}}.\end{aligned}$$

But the last quantity is strictly less than r, by assumption (1.1). Since $\phi(t_0) \in \partial B$, this gives a contradiction, which proves the lemma. ∎

We are now ready to prove Theorem 1.

Proof of Theorem 1. Combining Lemma 18 and Lemma 20, we see that $T = \infty$. Moreover by Lemma 20, ϕ stays in U forever. Therefore, by Lemma 19, it follows that $f(\phi(s)) \le e^{-\alpha s} f(x_0)$ for all s. It remains to establish the convergence of the flow and the rate of convergence. Let

$$t_0 := \inf\{t : f(\phi(t)) = 0\},$$

with the understanding that $t_0 = \infty$ if $f(\phi(t)) > 0$ for all t. Then note that for any $0 \le s < t < t_0$,

$$\begin{aligned}|\phi(t) - \phi(s)| &\le \int_s^t |\nabla f(\phi(u))| du \\ &\le \Big(\int_s^t 2g(u) du \int_s^t \frac{|\nabla f(\phi(u))|^2}{2g(u)} du\Big)^{1/2},\end{aligned}$$

where $g(u) = \sqrt{f(\phi(u))}$, as in the proof of Lemma 20. As in that proof, note that

$$\int_s^t \frac{|\nabla f(\phi(u))|^2}{2g(u)} du = g(s) - g(t) \le g(s) = \sqrt{f(\phi(s))} \le e^{-\alpha s/2}\sqrt{f(x_0)}.$$

On the other hand,

$$\int_s^t 2g(u) du \le \sqrt{f(x_0)} \int_s^t 2e^{-\alpha u/2} du \le \frac{4\sqrt{f(x_0)}}{\alpha} e^{-\alpha s/2}.$$

Combining the last three displays, and invoking assumption (1.1), we get

$$|\phi(t) - \phi(s)| \le 2\sqrt{\frac{f(x_0)}{\alpha}} e^{-\alpha s/2} \le r e^{-\alpha s/2}. \tag{A.3}$$

Note that the bound does not depend on t. If $t_0 < \infty$, then by Corollary 17, $\phi(t) = \phi(t_0)$ for all $t > t_0$. Thus, in this case $\phi(t)$ converges to $x^* := \phi(t_0)$. The rate of convergence is established by taking $t \to t_0$ in (A.3). If $t_0 = \infty$, then (A.3) proves the Cauchy property of the flow ϕ, which shows that $\phi(t)$ converges to some $x^* \in B$ as $t \to \infty$. Again, taking $t \to \infty$ in (A.3) proves the rate of convergence. ∎

A.2 Proof of Theorem 2

If $f(x_0) = 0$, then $\nabla f(x_0) = 0$ and therefore $x_k = x_0$ for all k, and there is nothing to prove. So, let us assume that $f(x_0) > 0$. The following lemma is the key step in the proof of Theorem 2.

Lemma 21. *For all $k \geq 0$, $x_k \in B(x_0, r)$.*

The proof of this lemma will be carried out via induction on k. We have $x_0 \in B(x_0, r)$. Suppose that $x_1, \ldots, x_{k-1} \in B(x_0, r)$ for some $k \geq 1$. We will use this hypothesis to show that $x_k \in B(x_0, r)$, with k remaining fixed henceforth.

Lemma 22. *For each j, define*

$$R_j := f(x_{j+1}) - f(x_j) + \eta |\nabla f(x_j)|^2. \tag{A.4}$$

Then for all $0 \leq j \leq k-1$, $|R_j| \leq \epsilon\eta |\nabla f(x_j)|^2$.

Proof. Since $x_{k-1} \in B(x_0, r)$, the assumed upper bound on η implies that

$$\eta |\nabla f(x_{k-1})| \leq L_1 \sqrt{p}\eta \leq r.$$

Thus, $|x_k - x_{k-1}| \leq r$, and therefore, $x_k \in B(x_0, 2r)$. Consequently, the line segment joining x_{k-1} and x_k lies entirely in $B(x_0, 2r)$. Since $x_0, \ldots, x_{k-1} \in B(x_0, r)$, the line segments joining x_j and x_{j+1} lie in $B(x_0, r)$ for all $j \leq k-2$. Thus, by Taylor expansion, we have that for any $0 \leq j \leq k-1$,

$$\begin{aligned} R_j &= f(x_j - \eta \nabla f(x_j)) - f(x_j) + \eta |\nabla f(x_j)|^2 \\ &= \frac{\eta^2}{2} \nabla f(x_j) \cdot \nabla^2 f(x_j^*) \nabla f(x_j), \end{aligned}$$

where x_j^* is a point on the line segment joining x_j and x_{j+1}, and $\nabla^2 f(x_j^*)$ is the Hessian matrix of f at x_j^*. Since L_2 is an upper bound on the magnitudes of all second order derivatives of f in $B(x_0, 2r)$, and $x_j^* \in B(x_0, 2r)$, this gives

$$\begin{aligned} |R_j| &\leq \frac{L_2 \eta^2}{2} \sum_{i,i'=1}^{p} |\partial_i f(x_j)| |\partial_{i'} f(x_j)| \\ &= \frac{L_2 \eta^2}{2} \Big(\sum_{i=1}^{p} |\partial_i f(x_j)| \Big)^2 \leq \frac{L_2 \eta^2 p}{2} \sum_{i=1}^{p} |\partial_i f(x_j)|^2, \end{aligned}$$

which proves that $|R_j| \leq \frac{1}{2} L_2 p \eta^2 |\nabla f(x_j)|^2$. Since $L_2 p \eta \leq 2\epsilon$, this proves the claim. ∎

Lemma 23. *We have $(1-\epsilon)\alpha\eta \leq 1$, and for any $0 \leq j \leq k$,*

$$f(x_j) \leq (1 - (1-\epsilon)\alpha\eta)^j f(x_0).$$

Proof. Since $x_j \in B(x_0, r)$ for $j \le k-1$, Lemma 22 and the definition of α imply that for $j \le k-1$,

$$\begin{aligned} f(x_{j+1}) &= f(x_j) - \eta|\nabla f(x_j)|^2 + R_j \\ &\le f(x_j) - (1-\epsilon)\eta|\nabla f(x_j)|^2 \\ &\le (1 - (1-\epsilon)\alpha\eta) f(x_j). \end{aligned}$$

Since $f(x_0) > 0$ and $f(x_1) \ge 0$, we can take $j = 0$ above and divide both sides by $f(x_0)$ to get that $(1-\epsilon)\alpha\eta \le 1$. Iterating the above inequality gives the desired upper bound for $f(x_j)$. ∎

Lemma 24. *For each $j \le k-1$,*

$$f(x_j) - f(x_{j+1}) \ge (1-\epsilon)\eta|\nabla f(x_j)|^2.$$

Proof. By Lemma 22,

$$\begin{aligned} \eta|\nabla f(x_j)|^2 &= f(x_j) - f(x_{j+1}) + R_j \\ &\le f(x_j) - f(x_{j+1}) + \epsilon\eta|\nabla f(x_j)|^2. \end{aligned}$$

Rearranging terms, we get the desired inequality. ∎

Lemma 25. *For all $0 \le j \le k-1$,*

$$\sum_{l=j}^{k-1} \eta|\nabla f(x_l)| \le (1 - (1-\epsilon)\alpha\eta)^{j/2} \sqrt{\frac{4f(x_0)}{\alpha(1-\epsilon)^2}}.$$

Proof. Let $\kappa := (1-\epsilon)^{-1}$. By Lemma 24,

$$\eta|\nabla f(x_j)| = \sqrt{\eta^2|\nabla f(x_j)|^2} \le \sqrt{\kappa\eta(f(x_j) - f(x_{j+1}))}.$$

Also by Lemma 24, $f(x_j) \ge f(x_{j+1})$ for each $j \le k-1$. Thus, for any $j \le k-1$, we use the Cauchy–Schwarz inequality to get

$$\begin{aligned} &\sum_{l=j}^{k-1} \eta|\nabla f(x_l)| \le \sum_{l=j}^{k-1} \sqrt{\kappa\eta(\sqrt{f(x_l)} + \sqrt{f(x_{l+1})})(\sqrt{f(x_l)} - \sqrt{f(x_{l+1})})} \\ &\le \Bigl(\sum_{l=j}^{k-1} \kappa\eta(\sqrt{f(x_l)} + \sqrt{f(x_{l+1})}) \sum_{l=j}^{k-1} (\sqrt{f(x_l)} - \sqrt{f(x_{l+1})})\Bigr)^{1/2} \\ &= \Bigl((\sqrt{f(x_j)} - \sqrt{f(x_k)}) \sum_{l=j}^{k-1} \kappa\eta(\sqrt{f(x_l)} + \sqrt{f(x_{l+1})})\Bigr)^{1/2} \\ &\le \Bigl(2\kappa\eta\sqrt{f(x_j)} \sum_{l=j}^{k-1} \sqrt{f(x_l)}\Bigr)^{1/2}. \end{aligned}$$

Using Lemma 23 to bound the terms on the right side, we get

$$\sum_{l=j}^{k-1} \eta|\nabla f(x_l)| \le (1-(1-\epsilon)\alpha\eta)^{j/4}\sqrt{f(x_0)}\Big(2\kappa\eta\sum_{l=j}^{k-1}(1-(1-\epsilon)\alpha\eta)^{l/2}\Big)^{1/2}.$$

Now, for $t \in [0,1]$, we have the inequality $1-t \le (1-t/2)^2$. This gives

$$\begin{aligned}\sum_{l=j}^{k-1}(1-(1-\epsilon)\alpha\eta)^{l/2} &\le (1-(1-\epsilon)\alpha\eta)^{j/2}\sum_{q=0}^{\infty}(1-(1-\epsilon)\alpha\eta)^{q/2}\\ &\le (1-(1-\epsilon)\alpha\eta)^{j/2}\sum_{q=0}^{\infty}\Big(1-\frac{(1-\epsilon)\alpha\eta}{2}\Big)^{q}\\ &= \frac{2(1-(1-\epsilon)\alpha\eta)^{j/2}}{(1-\epsilon)\alpha\eta}.\end{aligned}$$

Plugging this into the previous display completes the proof. ■

We are now ready to complete the proof of Lemma 21 and then use it to prove Theorem 2.

Proof of Lemma 21. Applying Lemma 25 with $j = 0$, and recalling that $\rho < 1-\epsilon$ by construction, we get

$$|x_k - x_0| \le \sum_{j=0}^{k-1}|x_{j+1}-x_j| = \sum_{j=0}^{k-1}\eta|\nabla f(x_j)| \le \sqrt{\frac{4f(x_0)}{\alpha(1-\epsilon)^2}} < r.$$

This proves that $x_k \in B(x_0, r)$, completing the induction step. ■

Proof of Theorem 2. By Lemma 21, we know that $x_k \in B(x_0, r)$ for all k. Thus, Lemma 25 holds for all $k \ge 1$ and all $j < k$. In particular,

$$\begin{aligned}|x_k - x_j| &\le \sum_{l=j}^{k-1}|x_{l+1}-x_l| = \sum_{l=j}^{k-1}\eta|\nabla f(x_l)|\\ &\le (1-(1-\epsilon)\alpha\eta)^{j/2}\sqrt{\frac{4f(x_0)}{\alpha(1-\epsilon)^2}} \le r(1-(1-\epsilon)\alpha\eta)^{j/2}.\end{aligned}$$

Note that the bound goes to zero as $j \to \infty$, and has no dependence on k. Thus, $\{x_k\}_{k\ge 0}$ is a Cauchy sequence in $B(x_0, r)$, and therefore, converges to a limit $x^* \in B(x_0, r)$. Moreover, the above bound is also a bound for $|x^* - x_j|$, since it has no dependence on k. Lastly, since $x_k \in B(x_0, r)$ for all k, Lemma 23 also holds for any j. This shows that $f(x^*) = 0$ and gives the required bound for $f(x_j)$. ■

A.3 Proof of Theorem 4

Take any $w = (W_1, b_1, \ldots, W_L, b_L) \in \mathbb{R}^p$. Define

$$f_1(x, w) := \sigma_1(W_1 x + b_1),$$

and for $2 \le \ell \le L$, define

$$f_\ell(x, w) := \sigma_\ell(W_\ell \sigma_{\ell-1}(\cdots W_2 \sigma_1(W_1 x + b_1) + b_2 \cdots) + b_\ell),$$

so that $f = f_L$. Note that $f_\ell(\cdot, w)$ is a map from $\mathbb{R}^d$ into $\mathbb{R}^{d_\ell}$. Define

$$g_1(x, w) := W_1 x + b_1,$$

and for $2 \le \ell \le L$, define

$$g_\ell(x, w) := W_\ell f_{\ell-1}(x, w) + b_\ell,$$

so that $f_\ell(x, w) = \sigma_\ell(g_\ell(x, w))$. Let $D_\ell(x, w)$ be the $d_\ell \times d_\ell$ diagonal matrix whose diagonal consists of the elements of the vector $\sigma_\ell'(g_\ell(x, w))$. Let $\partial_{ij} f_\ell$ denote the partial derivative of f_ℓ with respect to the $(i, j)^{\text{th}}$ element of W_1. Then note that

$$\partial_{ij} f_1 = D_1 \partial_{ij} g_1 = D_1 e_i e_j^T x = x_j D_1 e_i,$$

where $e_i \in \mathbb{R}^{d_1}$ is the vector whose i^{th} component is 1 and the rest are zero. Similarly, for $2 \le \ell \le L$,

$$\partial_{ij} f_\ell = D_\ell \partial_{ij} g_\ell = D_\ell W_\ell \partial_{ij} f_{\ell-1}.$$

Using this relation and the fact that $D_L = 1$, we get

$$\partial_{ij} f(x, w) = x_j q_i(x, w), \tag{A.5}$$

where

$$q_i(x, w) := W_L D_{L-1}(x, w) W_{L-1} \cdots W_2 D_1(x, w) e_i. \tag{A.6}$$

Let $H(w)$ denote the $n \times n$ matrix whose $(i, j)^{\text{th}}$ entry is

$$H_{ij}(w) := \nabla_w f(x_i, w) \cdot \nabla_w f(x_j, w).$$

Then the definition of S shows that for any w where $S(w) \ne 0$,

$$\begin{aligned} \frac{|\nabla S(w)|^2}{S(w)} &= \frac{4}{n} \frac{\sum_{i,j=1}^n (y_i - f(x_i, w))(y_j - f(x_j, w)) H_{ij}(w)}{\sum_{i=1}^n (y_i - f(x_i, w))^2} \\ &\ge \frac{4}{n} \lambda(w), \end{aligned} \tag{A.7}$$

where $\lambda(w)$ denotes the minimum eigenvalue of $H(w)$. Now note that for any vector $a = (a_1, \ldots, a_n)^T \in \mathbb{R}^n$ with norm 1, we have

$$\begin{aligned} \sum_{i,j=1}^n a_i a_j H_{ij}(w) &= \sum_{i,j=1}^n a_i a_j \nabla_w f(x_i, w) \cdot \nabla_w f(x_j, w) \\ &= \Big| \sum_{i=1}^n a_i \nabla_w f(x_i, w) \Big|^2 = \sum_{j=1}^p \Big(\sum_{i=1}^n a_i \partial_{w_j} f(x_i, w) \Big)^2. \end{aligned}$$

We obtain a lower bound on the above term by simply considering those j's that correspond to the entries of W_1. By (A.5), this gives

$$\sum_{i,j=1}^{n} a_i a_j H_{ij}(w) \geq \sum_{r=1}^{d_1}\sum_{s=1}^{d}\Big(\sum_{i=1}^{n} a_i \partial_{rs} f(x_i, w)\Big)^2$$
$$= \sum_{r=1}^{d_1}\sum_{s=1}^{d}\Big(\sum_{i=1}^{n} a_i x_{si} q_r(x_i, w)\Big)^2 \geq n\lambda_X \sum_{r=1}^{d_1}\sum_{i=1}^{n} a_i^2 q_r(x_i, w)^2,$$

where λ_X is the minimum eigenvalue of $\frac{1}{n}X^T X$, and $X = (x_{si})_{1\leq s\leq d, 1\leq i\leq n}$ the $d \times n$ matrix whose i^{th} column is the vector x_i. Since $x_1, \ldots, x_n$ are linearly independent, the Gram matrix $X^T X/n$ is strictly positive definite, which implies that $\lambda_X > 0$. Since $|a| = 1$, this gives

$$\lambda(w) \geq n\lambda_X \sum_{r=1}^{d_1} \min_{1\leq i\leq n} q_r(x_i, w)^2. \tag{A.8}$$

Now take any w as in the statement of Theorem 4, that is,

- the entries of $W_2, \ldots, W_{L-1}$ are all strictly positive, and
- $b_1, \ldots, b_L$ and W_1 are zero.

Then, since $\sigma_1(0) = 0$, we have that $f_1(x, w) = 0$ for each x. By induction, $f_\ell(x, w) = 0$ for each $\ell \leq L$, and hence $f(x, w) = 0$. Thus,

$$S(w) = \frac{1}{n}\sum_{i=1}^{n} y_i^2, \tag{A.9}$$

irrespective of the values of $W_2, \ldots, W_L$. Let $\delta, K, r, K', a_\ell, c_\ell, P, C$ be as in Theorem 4 and Remark 9, and take $w' \in B(w, r)$. Each entry of $W_2', \ldots, W_{L-1}'$ then lies in $[r, K']$. Each row of W_1' has Euclidean norm at most r, and each bias entry has absolute value at most r. Hence, for every input x_i, each coordinate of $g_1(x_i, w')$ has absolute value at most

$$r|x_i| + r \leq r(R_X + 1) = a_1.$$

Inductively, for $2 \leq \ell \leq L-1$, the coordinates of $g_\ell(x_i, w')$ have absolute value at most

$$K' d_{\ell-1}\gamma_{\ell-1}(a_{\ell-1}) + r = a_\ell.$$

Thus each diagonal entry of $D_\ell(x_i, w')$ is at least $c_\ell > 0$, for $1 \leq \ell \leq L-1$. These bounds have no dependence on the output weights.

Let $A := \min_j (W_L)_{1j} \geq A_0 > r$. Each entry of W_L' is at least $A - r$, so (A.6) gives, for every $1 \leq s \leq d_1$,

$$q_s(x_i, w') \geq (A - r)r^{L-2}PC.$$

When $L = 2$, this is simply $q_s(x_i, w') \geq (A - r)c_1$, with no intermediate hidden-layer weight matrices. Equations (A.7) and (A.8) therefore imply

$$\frac{|\nabla S(w')|^2}{S(w')} \geq 4\lambda_X d_1 (A - r)^2 r^{2L-4}(PC)^2 = \underline{\alpha}$$

whenever $w' \in B(w,r)$ and $S(w') > 0$. If $Y_2 > 0$, the choice (2.3) gives

$$\frac{4S(w)}{r^2\alpha(w,r)} \le \frac{Y_2}{\lambda_X d_1 (A-r)^2 r^{2L-2}(PC)^2} < \frac{1}{4}.$$

This verifies (1.1). Theorems 1 and 2, with the step-size bound (2.4) for the chosen w, prove the two conclusions. If $Y_2 = 0$, then $S(w) = 0$ and both dynamics are stationary. This completes the proof of Theorem 4. ∎

A.4 Proof of Theorem 11

Use the constants a, b, B, β, M from the theorem, and suppose that E_M occurs. For every $w' \in B(w_0, M)$, all entries of $W_2', \ldots, W_L'$ are at least M. The activation derivatives are globally bounded below by C_1, so (A.6) gives

$$q_s(x_i, w') \ge C_1^{L-1} M^{L-1} P \quad \text{for all } i, s.$$

By (A.7) and (A.8), the local PL constant for S satisfies

$$\alpha(w_0, M) \ge a M^{2L-2}. \tag{A.10}$$

At initialization, all biases are zero. Since $|\sigma_\ell(t)| \le C_2|t|$ for each hidden-layer activation, while the output activation is the identity,

$$\begin{aligned} |f(x, w_0)| &\le C_2^{L-1} \left(\prod_{\ell=2}^{L} \|W_\ell\|_F \right) |W_1 x| \\ &\le 3^{L-1} C_2^{L-1} M^{L-1} \left(\prod_{\ell=2}^{L} \sqrt{d_\ell d_{\ell-1}} \right) |W_1 x| \\ &= b M^{L-1} |W_1 x|. \end{aligned}$$

Here $\prod_{\ell=2}^{L} \sqrt{d_\ell d_{\ell-1}} = \sqrt{d_1} P$, since $d_L = 1$. Consequently,

$$\begin{aligned} S(w_0) &\le 2Y_2 + \frac{2b^2 M^{2L-2}}{n} \sum_{i=1}^{n} |W_1 x_i|^2 \\ &= 2Y_2 + \frac{2b^2 M^{2L-2}}{n} \operatorname{Tr}(W_1 X X^T W_1^T) \\ &\le 2Y_2 + B M^{2L-2} \|W_1\|_F^2. \end{aligned} \tag{A.11}$$

On F_M, the last term is at most $(a/16)M^{2L}$, because $B\beta^2 = a/16$. Also, (2.6) gives $2Y_2 \le (a/16)M^{2L}$. Thus

$$4S(w_0) \le \frac{a}{2} M^{2L} \le \frac{1}{2} M^2 \alpha(w_0, M) < M^2 \alpha(w_0, M). \tag{A.12}$$

In particular, the first-layer squared-norm threshold is explicitly

$$\beta^2 M^2 = \frac{aM^2}{16B}.$$

Theorem 1 now proves the gradient-flow assertion.

For completeness, we make the uniform step-size choice precise. Let $\mathcal{K}_M$ be the set of all initializations in $E_M \cap F_M$, with zero biases, and let

$$\mathcal{U}_M := \{w : \operatorname{dist}(w, \mathcal{K}_M) \le 2M\}.$$

Both sets are compact. As S is C^2, the constants

$$\widehat{L}_1 := \max\left\{1,\ \sup_{w\in\mathcal{U}_M}\ \max_j |\partial_j S(w)|\right\},$$
$$\widehat{L}_2 := \max\left\{1,\ \sup_{w\in\mathcal{U}_M}\ \max_{j,k} |\partial_j\partial_k S(w)|\right\}$$

are finite. For every $w_0 \in \mathcal{K}_M$ with $S(w_0) > 0$, (A.12) gives $\rho \le 1/\sqrt{2} < 3/4$, so $\epsilon = 1/4$ is admissible in Theorem 2. Therefore the single bound

$$\eta_0 := \min\left\{\frac{M}{\widehat{L}_1\sqrt{p}},\ \frac{1}{2\widehat{L}_2 p}\right\} > 0 \tag{A.13}$$

works for all $w_0 \in \mathcal{K}_M$. Initializations with $S(w_0) = 0$ are stationary. The derivative bounds in (A.13) may depend on the full data and on the activation functions, including their second derivatives on this compact set; no dimension-independent step-size bound is claimed.

Finally, $\|W_1\|_F^2$ has distribution $(c/d)\chi^2_{dd_1}$, independently of E_M. Each downstream Gaussian entry has a strictly positive probability of lying in $[2M, 3M]$, giving exactly the product in Remark 12. In particular, the sufficient event has strictly positive probability. This completes the proof. ■

References

P.-A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

Z. Allen-Zhu, Y. Li, and Y. Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems*, volume 32, pages 1–12, 2019.

Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

S. Arora, N. Cohen, N. Golowich, and W. Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka–Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017.

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

P. Bartlett, D. Helmbold, and P. Long. Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks. In *Proceedings of the 35th International conference on machine learning*, volume 80, pages 521–530. PMLR, 2018.

J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, pages 1–11, 2019.

Y. Cao and Q. Gu. Generalization error bounds of gradient descent for learning over-parameterized deep ReLU networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3349–3356, 2020.

Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

J.-N. Corvellec and V. V. Motreanu. Nonlinear error bounds for lower semicontinuous functions on metric spaces. *Mathematical Programming*, 114(2):291–319, 2008.

M. Danilova, P. Dvurechensky, A. Gasnikov, E. Gorbunov, S. Guminov, D. Kamzolov, and I. Shibaev. Recent theoretical advances in non-convex optimization. *arXiv preprint arXiv:2012.06188*, 2020.

D. Drusvyatskiy, A. D. Ioffe, and A. S. Lewis. Curves of descent. *SIAM Journal on Control and Optimization*, 53(1):114–138, 2015.

S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

S. S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1675–1685. PMLR, 2019.

J. C. Duchi and F. Ruan. Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimization*, 28(4):3229–3259, 2018.

W. E, C. Ma, and L. Wu. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Science China Mathematics*, 63(7):1235–1258, 2020. doi: 10.1007/s11425-019-1628-5.

I. J. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

M. Hardt and T. Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

A. D. Ioffe. Metric regularity and subdifferential calculus. *Russian Mathematical Surveys*, 55(3):501–558, 2000.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pages 8571–8580. Curran Associates, Inc., 2018.

A. Jentzen and A. Riekert. On the existence of global minima and convergence analyses for gradient descent methods in the training of deep neural networks. *arXiv preprint arXiv:2112.09684*, 2021.

Z. Ji and M. Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1724–1732. PMLR, 2017.

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak–łojasiewicz condition. arXiv preprint, 2016.

K. Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'Institut Fourier*, 48(3):769–783, 1998.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Y. Li and Y. Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, volume 31, pages 1–10, 2018.

C. Liu, L. Zhu, and M. Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022. doi: 10.1016/j.acha.2021.12.009.

S. Łojasiewicz. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117:87–89, 1963.

E. Moulines and F. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, volume 24, pages 1–9, 2011.

K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2004.

P. Netrapalli. Stochastic gradient descent and its variants in machine learning. *Journal of the Indian Institute of Science*, 99(2):201–213, 2019.

B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

S. Robin and M. Lelarge. Convergence beyond the over-parameterized regime using Rayleigh quotients. In *Advances in Neural Information Processing Systems*, 2022.

S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

K. A. Sankararaman, S. De, Z. Xu, W. R. Huang, and T. Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8469–8479. PMLR, 2020.

M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

R. Sun. Optimization for deep learning: theory and algorithms. *arXiv preprint arXiv:1912.08957*, 2019.

L. Wu, C. Ma, and W. E. How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

D. Zou, Y. Cao, D. Zhou, and Q. Gu. Gradient descent optimizes over-parameterized deep ReLU networks. *Machine Learning*, 109(3):467–492, 2020.