SAD Neural Networks: Divergent Gradient Flows and Asymptotic Optimality via o-minimal Structures

Julian Kranz[a,b] **Davide Gallon**[c,d] **Steffen Dereich**[e] **Arnulf Jentzen**[f,g]

[a]Department of Information Systems,
University of Münster, Germany, email: julian.kranz@uni-muenster.de
[b]Applied Mathematics: Institute for Analysis and Numerics,
University of Münster, Germany, email: julian.kranz@uni-muenster.de
[c]Applied Mathematics: Institute for Analysis and Numerics,
University of Münster, Germany, email: davide.gallon@uni-muenster.de
[d]RiskLab Switzerland, ETH Zürich, Switzerland, email: dgallon@ethz.ch
[e]Applied Mathematics: Institute for Mathematical Stochastics,
University of Münster, Germany, email: steffen.dereich@uni-muenster.de
[f]School of Data Science and School of Artificial Intelligence, The Chinese University
of Hong Kong, Shenzhen (CUHK-Shenzhen), China, email: ajentzen@cuhk.edu.cn
[g]Applied Mathematics: Institute for Analysis and Numerics,
University of Münster, Germany, email: ajentzen@uni-muenster.de

Abstract

We study gradient flows for loss landscapes of fully connected feedforward neural networks with commonly used continuously differentiable activation functions such as the logistic, hyperbolic tangent, softplus or GELU function. We prove that the gradient flow either converges to a critical point or diverges to infinity while the loss converges to an asymptotic critical value. Moreover, we prove the existence of a threshold $\varepsilon > 0$ such that the loss value of any gradient flow initialized at most ε above the optimal level converges to it. For polynomial target functions and sufficiently big architecture and data set, we prove that the optimal loss value is zero and can only be realized asymptotically. From this setting, we deduce our main result that any gradient flow with sufficiently good initialization diverges to infinity. Our proof heavily relies on the geometry of o-minimal structures. We confirm these theoretical findings with numerical experiments and extend our investigation to more realistic scenarios, where we observe an analogous behavior.

1 Introduction

The success of deep learning in practice has far outpaced our theoretical understanding of neural network training dynamics. While gradient-based optimization methods – such as (stochastic) gradient descent and its variants [77, 75, 43, 35, 57] – routinely achieve near-zero training loss on complex tasks, the theoretical principles governing their convergence remain poorly understood, particularly in realistic, non-convex settings. Even for the mathematically more tractable gradient flow, which can be viewed as the continuous time limit of gradient descent, a fundamental question persists:

Under what conditions does the gradient flow converge to a good local minimum?[1]

[1] Note that one cannot always expect convergence to *global* minima [50, 29, 30], but rather *local* minima with low risk level [46, 40, 41].

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Recent years have seen significant progress in the theory of deep learning, with convergence analyses often relying on strong assumptions, such as convexity [11, 22], overparameterization [34, 2], specific initialization schemes [48], or Łojasiewicz inequalities [25, 24, 1]. Moreover, most of these works either assume or imply that the iterates of the optimization scheme or the trajectory of the gradient flow stay within a bounded region. At the same time, an increasing body of work shows that this assumption is not always satisfied, with the simplest example being a linear classifier with logistic activation [70, 67, 78, 39].

1.1 Contributions

In this work, we bridge the gap outlined above by observing a general **dichotomy**: For a large class of smooth activation functions (such as the logistic, hyperbolic tangent, softplus or GELU function [42] which has been used in BERT [27] and GPT-3 [12]) and loss functions (such as mean squared error or binary cross entropy), the gradient flow either

- converges to a critical point, or
- diverges to infinity with the loss converging to a *generalized critical value* [76].

Using finiteness of the set of generalized critical values [17], we deduce the existence of a threshold $\varepsilon > 0$ such that any gradient flow initialized within ε-distance to the optimal level necessarily converges to it. We refer to Theorem 2.8 for a mathematically rigorous statement of our dichotomy result. We emphasize that convergence of gradient methods to critical points under boundedness assumptions is a well-understood phenomenon [10, 3, 4, 19, 25, 8, 52, 45]. Our main goal in stating the above dichotomy is to make the results from [17] accessible to the deep learning community as the existence of the threshold $\varepsilon > 0$ seems to be unnoticed in the deep learning literature.

As a concrete application of this dichotomy, we complement the divergence results obtained in [70, 67, 78, 39] by a general **divergence result for polynomial target functions**: For a polynomial target function $p\colon \mathbb{R}^n \to \mathbb{R}^m$ of degree at least two, a sufficiently big neural network architecture and a sufficiently big training dataset, we prove that the loss has no global minimum, but infimum zero. In particular, the optimal neural network parameters can only be achieved asymptotically. Thus, by the dichotomy result, the gradient flow diverges whenever the parameters are initialized at a loss level below ε. We refer to Theorem 3.4, Theorem 3.5 and Corollary 3.6 for a mathematically rigorous statement of our divergence result. The "infimum zero" part of our result is essentially well known and a standard ingredient in proofs of the universal approximation theorem [74, 9, 28, 21, 70]. Our novel contribution here is the "no global minimum" part and the resulting divergence phenomenon for the gradient flow.

Our analysis applies to fully connected feedforward deep neural networks with multiple input and output dimensions and a general class of activation functions including the logistic, hyperbolic tangent, softplus and GELU function. Moreover, we can treat both the average loss on a finite dataset and the true loss with respect to a sufficiently smooth probability density. This level of generality arises because our analysis builds on bootstrapping abstract properties rather than concrete computations:

- The functions appearing in our analysis are **definable** in an o-minimal structure in the sense of mathematical model theory [31], which intuitively means that they can be defined using first order statements involving well-behaved operations such as $+, -, \cdot, \div$, exponentials, logarithms, derivatives and anti-derivatives. Definable functions have strong rigidity and finiteness properties such as the uniform finiteness theorem [15], and definable dimension theory [31]. Our proofs use these techniques in a central and novel way.
- We isolate the general class of *Sublinear Analytic Definable* (**SAD**) functions (see Definition 3.1 and Example 3.2) for which our results hold. SAD functions have good permanence properties ensuring that neural networks with SAD activation function are themselves SAD. The main idea of proving the absence of global minima in our divergence result is that the sublinearity of a SAD neural network prevents it from being globally identical to a polynomial of degree at least two, while the analyticity and definability ensure enough rigidity to detect this property on a finite dataset (or compactly supported distribution). On the other hand, analyticity guarantees a sufficient supply of non-trivial Taylor coefficients, allowing polynomials to be approximated arbitrarily well. We believe that our definition of

SAD activation functions provides a convenient theoretical framework for future research on smooth neural networks.

Building on the theoretical framework established earlier, we validate our findings by training neural networks on polynomial target functions using optimization methods such as gradient descent and Adam [57]. The results closely align with the behavior predicted by our gradient flow analysis. We then extend our investigation to more complex and more practically relevant scenarios: numerical solution of PDEs using the Deep Kolmogorov Method [5] and image classification of the MNIST dataset [63]. Across all these experiments, we consistently observe a growth in the norm of the neural network parameters as the loss diminishes. These findings suggest that the divergence phenomenon we analyze actually holds in much broader generality. We include the source code for our numerical simulations at https://github.com/deeplearningmethods/sad.

The paper is structured as follows: We end this introduction by briefly summarizing related work and limitations of ours below. We introduce the necessary mathematical background and formally state the dichotomy result in Section 2. In Section 3, we develop the divergence result for polynomial target functions. Our numerical experiments are reviewed in Section 4. We conclude in Section 5. The proofs of our main results are given in the Appendix.

1.2 Related work

The general theory of o-minimal structures originates from [72, 58, 73] and is introduced accessibly in [31, 15, 81]. We specifically use o-minimality of Pfaffian functions [80], Kurdyka's Łojasiewicz inequality [60], definable dimension theory [31], the uniform finiteness theorem [15] and finiteness of the set of generalized critical values [17]. Applications of o-minimal structures to optimization are surveyed in [47]. There is an increasing body of work on the application of o-minimal structures to convergence results for gradient flows and stochastic gradient descent (SGD) under stability assumptions (assuming that the parameters stay within a bounded region) [10, 3, 4, 19, 25, 8, 52, 45]. Conditions ensuring stability are given in [53, 62]. In the unstable regime, there are divergence results which consider homogeneous networks [67, 78], very specific regression problems [39], or use convergence of the loss and absence of global minimizers as an assumption [70]. Other applications of o-minimality to neural networks are given in [55, 56, 54]. Neural network approximation results such as the universal approximation theorem [16, 37, 44, 64] are surveyed in [28, 7]. Our Theorem 3.5 can be extracted from existing literature. The proof in [21] for tanh for instance generalizes to our setting. Further neural network approximation results are given in [82, 71, 9, 20, 69, 70].

1.3 Limitations and outlook

While we prove powerful results for gradient flows, our methods are currently not capable of proving analogous results for (stochastic) gradient descent. Although results exist under stability assumptions (see Subsection 1.2 above), the main limitation in the unstable regime is our lack of control over the Hessian. To mitigate this, one could try to prove that the loss function satisfies a generalized smoothness condition in the sense of [83, 65] (at least along the gradient trajectory) and use this to employ a non-convex convergence proof for SGD. In light of the empirical evidence provided in [83] this approach seems quite promising and will thus be pursued in future work.

Although we do provide empirical evidence for similar results for gradient descent and other optimizers in Section 4, our numerical results are fairly limited by the variety and size of the considered datasets. Since the main focus of our paper are the mathematical theorems, these experiments are intended as a proof of concept rather than an in-depth experimental analysis.

Another limitation of our analysis is that the considered activation functions are required to be at least C^1 or, in some cases, analytic. In particular, our analysis does not apply to the ReLU activation function. In fact our divergence result for polynomial target functions does **not** hold for the ReLU function since in this case, global minima exist for shallow networks [49, 26, 23]. Moreover, global minima always exist if one adds an L^2-regularization term to the loss function. For the same reason, divergence does not occur if one uses an optimizer with weight decay such as AdamW [66].

In several neural network training scenarios with diverging model parameters, one can prove that the normalized parameters converge, see [51, 78, 59]. This phenomenon known as *directional convergence* is closely related to the *gradient conjecture at infinity* [18] and can be deduced for

C^2-functions definable in a *polynomially bounded* o-minimal structure as a consequence of [61].[2] However, the case of o-minimal structures containing the exponential function which is relevant to our setting is still open [18] and remains to be explored.

2 A dichotomy for definable gradient flows

In this section, we present our first result (see Theorem 2.8 below) which states that for arbitrary deep neural networks and most of the commonly used loss functions and activation functions (see Example 2.4 below), any gradient flow associated to the training loss landscape either converges to a critical point or diverges to a *generalized critical value* at infinity. The main conceptual reason for this phenomenon is that all the occuring functions are definable in an o-minimal structure in the sense of model theory. We recall the necessary mathematical background below:

Definition 2.1 (O-minimal structure)**.** A *structure* $\mathcal{S}$ (expanding the real closed field $\mathbb{R}$) is a collection of subsets $\mathcal{S}_n \subseteq \mathcal{P}(\mathbb{R}^n)$ for all $n \in \mathbb{N}$ satisfying the following properties:

(i) For $A, B \in \mathcal{S}_n$, we have $A \cup B$, $A \cap B$, $\mathbb{R}^n \setminus A \in \mathcal{S}^n$.

(ii) For $A \in \mathcal{S}_n$ and $B \in \mathcal{S}_m$, we have $A \times B \in \mathcal{S}^{n+m}$.

(iii) For $A \in \mathcal{S}^n$ and $\pi\colon \mathbb{R}^n \to \mathbb{R}^m$ a coordinate projection with $m \leq n$, we have $\pi(A) \in \mathcal{S}_m$.

(iv) For each polynomial $p \in \mathbb{R}[X_1, \dots, X_n]$, we have $\{x \in \mathbb{R}^n \mid p(x) > 0\} \in \mathcal{S}^n$ and $\{x \in \mathbb{R}^n \mid p(x) = 0\} \in \mathcal{S}_n$.

A structure $\mathcal{S}$ is called *o-minimal* if all the elements of $\mathcal{S}_1$ are finite unions of points and intervals.

If $\mathcal{S}$ is an o-minimal structure, we call the sets $A \in \mathcal{S}_n$ the *definable* sets. A function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is called *definable*, if its graph $\{(x, f(x)) \mid x \in \mathbb{R}^n\}$ is definable. If no o-minimal structure is specified, we refer to a set or function as *definable* if it is definable in some o-minimal structure. An instructive (and in fact universal) example of a structure is the one *generated* by a set of functions $\{f_i\colon \mathbb{R}^{n_i} \to \mathbb{R} \mid i \in I\}$ which precisely consists of all sets that can be defined using first order statements involving real numbers and the symbols $+, -, \cdot, \div, <, (f_i)_{i \in I}$.

Standard examples of o-minimal structures are the structure of semialgebraic sets (generated by polynomials), the structure of subanalytic sets (generated by restricted analytic functions) [38], the structure generated by the exponential function [79], and the combination of the latter two [32]. The popular GELU activation function [42] is definable in none of these structures [33, Theorem 5.11], but it is definable in the structure generated by *Pfaffian functions*:

Definition 2.2 (Pfaffian functions)**.** A C^1-function $f\colon \mathbb{R}^n \to \mathbb{R}$ is called *Pfaffian* if there exist C^1-functions $f_1, \dots, f_r\colon \mathbb{R}^n \to \mathbb{R}$ with $f_r = f$ such that for any $i = 1, \dots, r$ and $j = 1, \dots, n$, the partial derivative $\frac{\partial f_i}{\partial x_j}(x)$ is a polynomial in $x, f_1(x), \dots, f_i(x)$.

Theorem 2.3 ([80])**.** *The structure $\mathbb{R}_{\mathrm{Pfaff}}$ generated by all Pfaffian functions is o-minimal.*

Pfaffian functions include all functions that can be defined iteratively using polynomials, exponentials, logarithms, derivatives and antiderivatives.

Example 2.4. The following activation functions $f\colon \mathbb{R} \to \mathbb{R}$ are C^1 and definable.

(i) The *logistic* function $f(x) = \frac{1}{1+e^{-x}}$,

(ii) The *hyperbolic tangent* function $f(x) = \tanh(x)$,

(iii) The *softplus* function $f(x) = \log(1 + e^x)$ [36],

(iv) The *swish* function $f(x) = \frac{x}{1+e^{-\beta x}}$ with $\beta > 0$,

(v) The *Gaussian error linear unit (GELU)* function $f(x) = x \cdot \frac{1}{2}(1 + \operatorname{erf}(\frac{x}{\sqrt{2}}))$, where $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ is the Gaussian error function [42],

[2] We would like to thank Vincent Grandjean for pointing this out.

(vi) The *Mish* function $f(x) = x\tanh(\log(1+e^x))$ [68],

(vii) The *exponential linear unit (ELU)* function $f(x) = \begin{cases} e^x - 1, & x \leq 0 \\ x, & x > 0 \end{cases}$ [14],

(viii) The *softsign* function $f(x) = \frac{x}{|x|+1}$.

The following loss functions $\ell\colon \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ are C^1 and definable.

(ix) The *squared error* $\ell(x, y) = (x-y)^2$,

(x) The *binary cross-entropy* $\ell(x, y) = -(x\log y + (1-x)\log(1-y))$, for $y \in (0, 1)$,

(xi) The *Huber loss* $\ell(x, y) = \begin{cases} \frac{1}{2}(x-y)^2, & |x-y| \leq \delta \\ \delta(|x-y| - \frac{1}{2}\delta), & |x-y| > \delta \end{cases}$ with $\delta \in (0, \infty)$.

The following functions are definable but not C^1:

(xiii) The *ReLU* function $f\colon \mathbb{R} \to \mathbb{R}, \quad f(x) = \max(0, x)$,

(xiv) The *absolute error* $\ell\colon \mathbb{R}\times\mathbb{R} \to \mathbb{R}, \quad \ell(x, y) = |x-y|$.

We proceed with a definition of neural network architectures. For a function $\psi\colon \mathbb{R} \to \mathbb{R}$ and an integer $d \in \mathbb{N}$, we denote by $\psi^{(d)}\colon \mathbb{R}^d \to \mathbb{R}^d, \quad \psi^{(d)}(x_1, \ldots, x_d) \coloneqq (\psi(x_1), \ldots, \psi(x_d))$ its *amplification*.

Definition 2.5 (Neural network architectures)**.** A *neural network architecture*[3] is a tuple $\mathcal{A} = (d_0, \ldots, d_k, \psi)$ where $d_0, \ldots, d_k \in \mathbb{N}$ and $\psi\colon \mathbb{R} \to \mathbb{R}$ is a function called the *activation function*. The *dimension* of $\mathcal{A}$ is given by $d(\mathcal{A}) = \sum_{i=1}^k d_i(d_{i-1}+1)$. Fix an isomorphism $\mathbb{R}^{d(\mathcal{A})} \mathrel{\hat{=}} \prod_{i=1}^k \mathbb{R}^{d_i \times d_{i-1}} \times \mathbb{R}^{d_i}$. The *response* $\mathcal{N}_\theta^{\mathcal{A}}\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_k}$ of the architecture $\mathcal{A}$ and parameters $\theta \mathrel{\hat{=}} (W_1, b_1, \ldots, W_k, b_k) \in \mathbb{R}^{d(\mathcal{A})}$ with $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$ and $b_i \in \mathbb{R}^{d_i}$ is given by $\mathcal{N}_\theta^{\mathcal{A}}(x) \coloneqq f_k \circ \ldots \circ f_1(x)$ where $f_1, \ldots, f_k$ are defined as

$$f_i(x) = \begin{cases} \psi^{(d_i)}(W_i x + b_i), & i < k \\ W_k x + b_k, & i = k. \end{cases}$$

Definition 2.6 (Generalized critical values)**.** Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a differentiable function.

(i) The set of *critical points* of f is the set $\{\theta \in \mathbb{R}^d \mid \nabla f(\theta) = 0\}$.

(ii) The set of *critical values* of f is the set $K_0(f) = \{f(\theta) \in \mathbb{R} \mid \theta \in \mathbb{R}^d, \nabla f(\theta) = 0\}$.

(iii) The set of *asymptotic critical values* of f is the set

$$K_\infty(f) = \{c \in \mathbb{R} \mid \exists (\theta_n)_{n\in\mathbb{N}} \subseteq \mathbb{R}^d \text{ s. t. } \|\theta_n\| \to \infty,\ f(\theta_n) \to c,\ \|\theta_n\|\|\nabla f(\theta_n)\| \to 0\}.$$

(iv) The set of *generalized critical values* of f is the set $K(f) = K_0(f) \cup K_\infty(f)$.

Definition 2.7 (Locally Lipschitz derivative)**.** A C^1-function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is said to have a *locally Lipschitz derivative*, if for each $x \in \mathbb{R}^n$, there is an open neighbourhood $U \subseteq \mathbb{R}^n$ of x and a constant $L > 0$ such that for all $x_1, x_2 \in U$, we have $\|Df(x_1) - Df(x_2)\| \leq L\|x_1 - x_2\|$.

Below, we characterize the gradient flows of neural network loss landscapes in the C^1 definable setting. Our result is mostly a well-known consequence of the available literature. We consider both the empirical loss with respect to a finite dataset (see item (i) below) and the true loss with respect to a continuous data distribution (see item (ii) below).

For a point $x \in \mathbb{R}^n$, we denote by δ_x the Dirac measure at x. For a measure μ and a measurable function $p\colon \mathbb{R}^n \to [0, \infty)$, we write $d\mu(x) = p(x)dx$ if μ is given by $\mu(A) = \int_A p(x)dx$ for all measurable sets A. The *support* of a function $p\colon \mathbb{R}^n \to [0, \infty)$ is the set $\overline{\{x \in \mathbb{R}^n \mid p(x) \neq 0\}}$.

[3] or more precisely a *fully connected feedforward* neural network

Theorem 2.8 (Dichotomy for gradient flows)**.** *Let $\mathcal{A} = (d_0, \ldots, d_k, \psi)$ be a neural network architecture with a C^1 definable activation function ψ (see Example 2.4 (i) - (xi)). Let $f\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_k}$ and $\ell\colon \mathbb{R}^{d_k} \times \mathbb{R}^{d_k} \to [0, \infty)$ be C^1 definable functions (the target and loss functions). Let μ be a probability measure on $\mathbb{R}^{d_0}$ with expected loss $\mathcal{L}\colon \mathbb{R}^{d(\mathcal{A})} \to \mathbb{R}$ defined by $\mathcal{L}(\theta) \coloneqq \mathbb{E}_{x\sim\mu}[\ell(\mathcal{N}^{\mathcal{A}}_{\theta}(x), f(x))]$, such that*

(i) $\mu = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$ for some $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathbb{R}^{d_0}$, or

(ii) $d\mu(x) = p(x)dx$ for a C^1 function $p\colon \mathbb{R}^{d_0} \to [0, \infty)$ such that $\mathcal{L}$ is definable, or

that μ is a convex combination of cases (i) and (ii). Assume moreover that ℓ, f and ψ have locally Lipschitz derivatives (see Example 2.4 (i) - (xi)). Then the following hold:

(iii) For every $\theta_0 \in \mathbb{R}^{d(\mathcal{A})}$, there exists a unique C^1 function $\Theta\colon [0, \infty) \to \mathbb{R}^{d(\mathcal{A})}$ satisfying

$$\Theta(0) = \theta_0, \quad \Theta'(t) = -\nabla\mathcal{L}(\Theta(t)), \quad \forall t \in [0, \infty).$$

(iv) For every Θ as in (iii), exactly one of the following holds:

(a) either $\lim_{t\to\infty} \Theta(t)$ exists and is a critical point of $\mathcal{L}$, or
(b) $\lim_{t\to\infty} \|\Theta(t)\| = \infty$ and $\lim_{t\to\infty} \mathcal{L}(\Theta(t))$ is an asymptotic critical value of $\mathcal{L}$.

(v) There exists an $\varepsilon > 0$ such that for every Θ as in (iii) with $\mathcal{L}(\Theta(t_0)) < \inf_{\theta\in\mathbb{R}^{d(\mathcal{A})}} \mathcal{L}(\theta) + \varepsilon$ for some $t_0 \in [0, \infty)$, it holds that $\lim_{t\to\infty} \mathcal{L}(\Theta(t)) = \inf_{\theta\in\mathbb{R}^{d(\mathcal{A})}} \mathcal{L}(\theta)$.

Proof. The proof of Theorem 2.8 is given on page 27 in Appendix B. □

Remark 2.9. Note that Theorem 2.8 applies to all the activation and loss functions listed in Example 2.4 (i) - (xi). In particular, Theorem 2.8 applies to loss functions $\mathcal{L}\colon \mathbb{R}^d \to \mathbb{R}$ of the form $\mathcal{L}(\theta) \coloneqq \frac{1}{n}\sum_{i=1}^{n}(\mathcal{N}^{\mathcal{A}}_{\theta}(x_i) - y_i)^2$ for pairwise distinct $x_1, \ldots, x_n \in \mathbb{R}^{d_0}$ and $y_1, \ldots, y_n \in \mathbb{R}^{d_k}$.

3 Divergence of gradient flows for polynomial target functions

In this section, we state the main result of this paper (see Corollary 3.6) saying that for most of the common activation functions and any non-linear polynomial target function, the gradient flow almost always diverges to infinity while the loss converges to zero. Our conceptual starting point to this result is to isolate the key properties of common activation functions into the following definition.

Definition 3.1 (SAD activation function)**.** A function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is called *sublinear analytic definable* (**SAD**) if it satisfies the following properties:

(S) $\limsup_{t\to\infty} \frac{\|f(tx)\|}{t} < \infty$, for all $x \in \mathbb{R}^n$, *(sublinear)*

(A) f is analytic, *(analytic)*

(D) f is definable in some o-minimal structure (see Definition 2.1). *(definable)*

Practically all activation functions used in Machine Learning applications satisfy conditions (S) and (D) above. Moreover, many of them satisfy condition (A) too.

Example 3.2 (SAD activation functions)**.** The activation functions (i) - (vi) in Example 2.4 are SAD.

Definition 3.3 (Loss function)**.** We call a function $\ell\colon \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$ a *loss function* if and only if it satisfies for all $x, y \in \mathbb{R}^n$ that $x = y \Leftrightarrow \ell(x, y) = 0$.

The main technical contribution of this work states that on a sufficiently big training dataset (or data distribution), a given SAD neural network can never fit any non-linear polynomial perfectly.

We call a Borel measurable set $A \subseteq \mathbb{R}^n$ *conull* if its complement has Lebesgue measure zero.

Theorem 3.4 (Non-representability of polynomials)**.** *Let $\mathcal{A} = (d_0, \ldots, d_k, \psi)$ be a neural network architecture with SAD activation function ψ (see Example 2.4 (i) - (vi)). Let $f\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_k}$ be a polynomial of degree at least 2. Then there exist an integer $N \in \mathbb{N}$ and a conull dense open subset $\mathcal{D}_N \subseteq (\mathbb{R}^{d_0})^N$ such that the following holds: For any measurable loss function $\ell\colon \mathbb{R}^{d_0} \times \mathbb{R}^{d_0} \to [0, \infty)$ and any probability measure μ on $\mathbb{R}^{d_0}$ such that* *(i)* $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ *for some* $n \geq N$ *and* $x_1, \dots, x_n \in \mathbb{R}^{d_0}$ *with* $(x_1, \dots, x_N) \in \mathcal{D}_N$*, or*

(ii) $d\mu(x) = p(x)dx$ *for a measurable function* $p \colon \mathbb{R}^{d_0} \to [0, \infty)$*, or*

such that μ *is a convex combination of (i) and (ii), the expected loss* $\mathcal{L} \colon \mathbb{R}^{d(\mathcal{A})} \to [0, \infty)$ *defined by* $\mathcal{L}(\theta) := \mathbb{E}_{x \sim \mu}[\ell(\mathcal{N}_\theta^{\mathcal{A}}(x), f(x))]$ *satisfies* $\mathcal{L}(\theta) > 0$ *for all* $\theta \in \mathbb{R}^{d(\mathcal{A})}$*. Moreover, in the case* $d_0 = 1$*, the set* $\mathcal{D}_N$ *can be chosen as the set of all tuples* $(x_1, \dots, x_N)$ *for which* $x_1, \dots, x_N$ *are pairwise distinct.*

Proof. The proof of Theorem 3.4 is given on page 29 in Appendix C. □

While Theorem 3.4 prevents the loss $\mathcal{L}(\theta)$ from being exactly zero, we prove that a fixed sufficiently big architecture $\mathcal{A}$ still achieves arbitrarily good approximations. This is a standard result (see [21, 28, 7]) which we include for convenience of the reader.

Theorem 3.5 (Neural network approximation for polynomials)**.** *Let* $f \colon \mathbb{R}^m \to \mathbb{R}^r$ *be a polynomial of degree* $n \in \mathbb{N}$*. Let* $\mathcal{A} = (d_0, \dots, d_k, \psi)$ *be a neural network architecture with* $d_0 = m$ *input neurons,* $d_k = r$ *output neurons and a non-polynomial analytic activation function* ψ*. Assume that* $\mathcal{A}$ *is sufficiently big in the sense that*

(i) there is a hidden layer $0 < i < k$ *of size* $d_i \geq r\left(\binom{n+m}{m} - m\right)$*,*

(ii) all previous hidden layers have size $\min(d_0, \dots, d_{i-1}) \geq d_0$*,*

(iii) all subsequent hidden layers have size $\min(d_{i+1}, \dots, d_k) \geq d_k$*.*

Then there exists a sequence of parameters $(\theta_j)_{j \in \mathbb{N}} \subseteq \mathbb{R}^{d(\mathcal{A})}$ *such that* $(\mathcal{N}_{\theta_j}^{\mathcal{A}})_{j \in \mathbb{N}}$ *converges to* f *uniformly on compact sets. In particular, we have* $\inf_{\theta \in \mathbb{R}^{d(\mathcal{A})}} \mathcal{L}(\theta) = 0$ *where* $\mathcal{L}$ *is as in Theorem 3.4.*

Proof. The proof of Theorem 3.5 is given on page 35 in Appendix D. □

We can now phrase our main result. It states that for a sufficiently big architecture, a sufficiently big dataset (or distribution) and a sufficiently good initialization, the gradient flow diverges to infinity while the loss converges to zero.

Corollary 3.6 (Divergence for polynomial target functions)**.** *Let* $f \colon \mathbb{R}^m \to \mathbb{R}^r$ *be a polynomial of degree* $\deg(f) \geq 2$*. Let* $\mathcal{A} = (d_0, \dots, d_k, \psi)$ *be a neural network architecture with* $d_0 = m$ *input neurons,* $d_k = r$ *output neurons and a non-polynomial SAD activation function* ψ *(see Example 2.4 (i) - (vi)). Assume that* $\mathcal{A}$ *is sufficiently big in the sense that*

(i) there is a hidden layer $0 < i < k$ *of size* $d_i \geq r\left(\binom{\deg(f)+m}{m} - m\right)$*,*

(ii) all previous hidden layers have size $\min(d_0, \dots, d_{i-1}) \geq d_0$*,*

(iii) all subsequent hidden layers have size $\min(d_{i+1}, \dots, d_k) \geq d_k$*.*

Then there is an $N \in \mathbb{N}$ *and a conull dense open subset* $\mathcal{D}_N \subseteq (\mathbb{R}^{d_0})^N$ *with the following property: For any* C^1 *definable loss function* $\ell \colon \mathbb{R}^{d_0} \times \mathbb{R}^{d_0} \to [0, \infty)$ *and any probability measure* μ *on* $\mathbb{R}^{d_0}$ *with expected loss* $\mathcal{L} \colon \mathbb{R}^{d(\mathcal{A})} \to [0, \infty)$ *defined by* $\mathcal{L}(\theta) := \mathbb{E}_{x \sim \mu}[\ell(\mathcal{N}_\theta^{\mathcal{A}}(x), f(x))]$*, such that*

(iv) $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ *for some* $n \geq N$ *and* $x_1, \dots, x_n \in \mathbb{R}^{d_0}$ *with* $(x_1, \dots, x_N) \in \mathcal{D}_N$*, or*

(v) $d\mu(x) = p(x)dx$ *for a* C^1 *function* $p \colon \mathbb{R}^{d_0} \to [0, \infty)$ *such that* $\mathcal{L}$ *is definable, or*

such that μ *is a convex combination of (iv) and (v), there exists an* $\varepsilon > 0$ *such that for every* C^1 *function* $\Theta \colon [0, \infty) \to \mathbb{R}^d$ *satisfying* $\mathcal{L}(\Theta(t_0)) < \varepsilon$ *for some* $t_0 \in [0, \infty)$ *and* $\Theta'(t) = -\nabla\mathcal{L}(\Theta(t))$ *for all* $t \in [0, \infty)$*, we have* $\lim_{t \to \infty} \mathcal{L}(\Theta(t)) = 0$ *and* $\lim_{t \to \infty} \|\Theta(t)\| = \infty$*.*

Proof. The corollary is a direct consequence of Theorem 2.8, Theorem 3.4 and Theorem 3.5. □

4 Numerical experiments

In this section, we empirically validate the theoretical results established in the previous sections (see Corollary 3.6), demonstrating the divergence phenomena of the neural network parameters while the loss converges to zero. Interestingly, we observe that despite a sufficiently good initialization is required in our theorem, in practice it is not required for this phenomenon to occur. Our analysis begins with experiments on polynomial target functions and is then extended to more complex learning tasks, showing that this phenomenon persists beyond our settings and suggesting broader implications for neural network training. Further implementation details are provided in Appendix E.

While our theoretical results hold in the continuous-time gradient flow regime, practical training is typically performed using discrete-time (stochastic) gradient descent. Since gradient descent can be viewed as an explicit discretization of gradient flow, we expect qualitatively similar behavior, with potential numerical differences arising due to discretization.

The gradient flow dynamics of a differentiable loss function $\mathcal{L}\colon \mathbb{R}^d \to \mathbb{R}$ are defined by the ordinary differential equation

$$\theta(0) = \theta_0, \quad \theta'(t) = -\nabla\mathcal{L}(\theta(t)), \quad \forall t \in [0, \infty),$$

which describes the evolution of parameters under infinitesimal gradient updates. Gradient descent corresponds to an explicit (forward) Euler discretization of this flow:

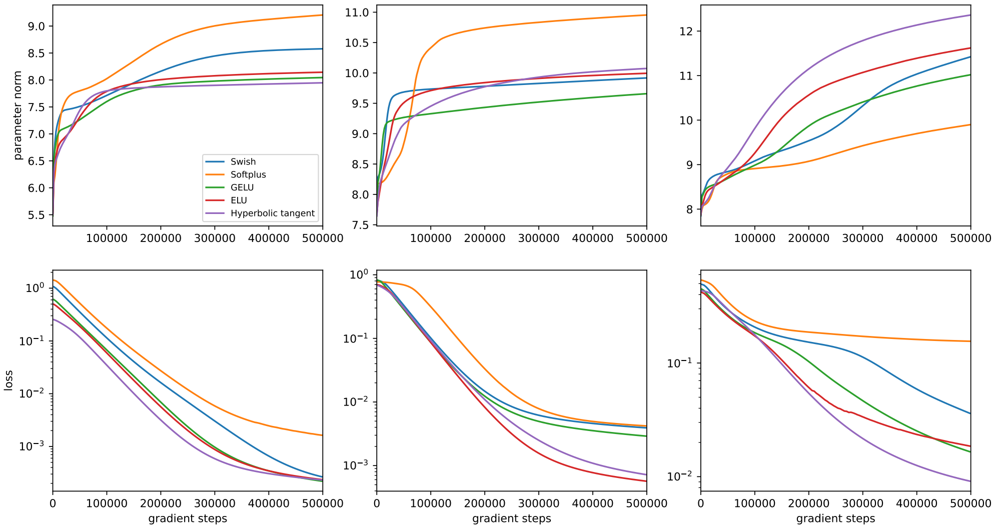
$$\theta_{k+1} = \theta_k + \eta\theta'(t)|_{t=k\eta} = \theta_k - \eta\nabla\mathcal{L}(\theta_k).$$

Here $\theta_k \in \mathbb{R}^d$ denotes the parameter vector at iteration k, and $\eta > 0$ denotes the learning rate.

4.1 Validation on polynomial target functions

We train fully connected neural networks with three hidden layers to approximate polynomial target functions of varying input dimension. To exhibit the generality of our findings, we consider five different SAD activation functions (see Definition 3.1) and for each of them we plot averages of the loss and the parameter norm (using exponential moving averages and Monte Carlo averaging over 20 independent random initializations).

To most closely resemble the gradient flow dynamics, we begin our analysis with gradient descent, see Figure 1. The slow divergence of parameters is theoretically justified by the fact that the growth of parameter norms along the gradient flow is $\mathcal{O}(\sqrt{t})$ (see (8) in Appendix B). In our setting, the growth seems to be logarithmic as illustrated in the exponentially rescaled Figure 6 in Appendix E. This is consistent with a similar logarithmic behavior proved in [67, Theorem 4.3].



(a)

(b)

(c)

Figure 1: Approximation of polynomial target functions using different activation functions and GD algorithm. From left to right: 1-dimensional input, 2-dimensional input, 4-dimensional input.

Motivated by this limitation, we also employ the Adam optimizer [57], whose adaptive learning rate mitigates the vanishing gradients problem by dynamically rescaling the effective step size. In contrast to gradient descent, where the update magnitude decreases as $\mathcal{O}(1/\sqrt{t})$ with time t, Adam maintains substantially larger updates during training, enabling faster convergence in scenarios where gradient descent or SGD would require more iterations to achieve comparable results.

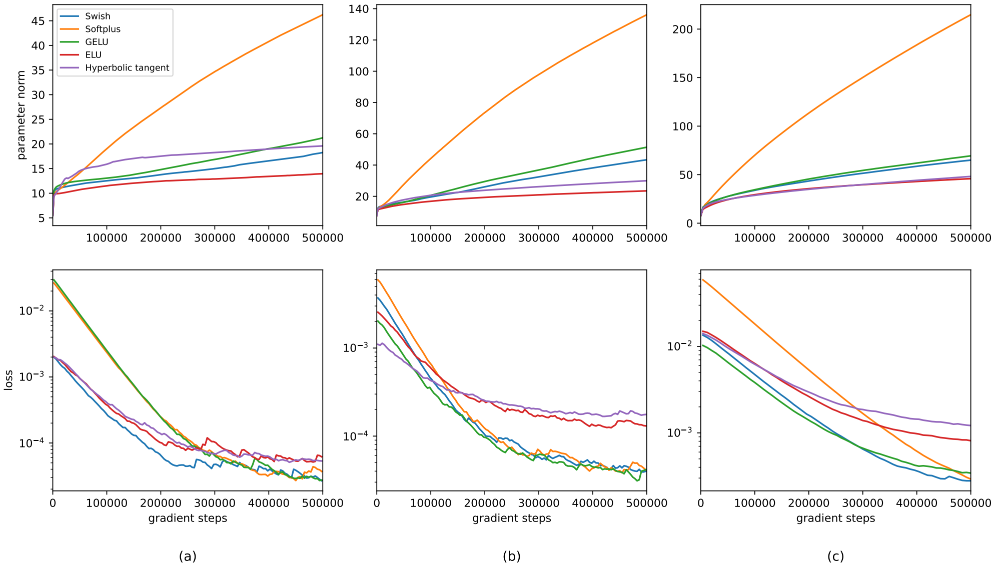


Figure 2: Approximation of polynomial target functions using different activation functions and Adam algorithm. From left to right: 1-dimensional, 2-dimensional, and 4-dimensional input.

In Figure 1 and Figure 2, we consider target functions with 1-, 2-, and 4-dimensional input. In the multi-dimensional cases, we also scale up the network size, increasing the number of hidden neurons from $(d_1, d_2, d_3) = (10, 20, 10)$ to $(20, 40, 20)$. Despite the simplicity of the task and the relatively small network size, we observe a significant growth in the norm of the neural network parameters when the mean squared error loss converges to zero, confirming our theoretical results.

4.2 Extension to more complex learning tasks

We investigate whether the divergence behavior observed in the theoretical settings extends to more complex, real-world learning tasks. Interestingly, we find that the same phenomena persist. In particular, the growth of the norm of neural network parameters becomes even more evident in practical scenarios involving high-dimensional data.

In Figure 3 (a), we train deep neural networks to approximate the solutions of two partial differential equations (PDEs): the Heat PDE and the Black–Scholes PDE. These equations are defined as follows:

Heat equation:

$$\begin{aligned} \partial_t u(t,x) &= \tfrac{1}{2}\Delta u(t,x), && (t,x) \in [0,T] \times \mathbb{R}^d, \\ u(0,x) &= \|x\|^2, && x \in \mathbb{R}^d. \end{aligned}$$

Black–Scholes equation:

$$\begin{aligned} &\partial_t u(t,x) + \sum_{i=1}^{d} \big((r-c)x_i \partial_{x_i} u(t,x) + \tfrac{1}{2}\sigma_i^2 x_i^2 \partial_{x_i x_i} u(t,x) \big) = 0, && (t,x) \in [0,T] \times (0,\infty)^d, \\ &u(0,x) = e^{-rT} \max\{\max\{x_1,\ldots,x_d\} - K, 0\}, && x \in (0,\infty)^d, \end{aligned}$$

where r is the risk-free interest rate, c is the cost of carry, K is the strike price and σ is the volatility. To solve these PDEs we employ the deep Kolmogorov method [5] which reformulates the PDEs as stochastic optimization problems using the Feynman–Kac representation. Additionally, in Figure 3 (b) we present results from a standard supervised learning task: image classification on the MNIST dataset [63]. This allows us to observe a similar blow-up phenomenon in a classification setting with discrete labels.

In all cases we observe a significant increase in the parameter norm, consistent with the behavior analyzed in the previous section. These results show that our theoretical findings can be generalized to realistic examples and are indicative of a general phenomenon. For the approximation of the PDEs we employ a neural network with three hidden layers and the GELU activation function. The loss function displayed is the relative mean squared error between the model prediction and either the true solution or a Monte Carlo approximation of it. In the MNIST simulations we flatten the input images and train a neural network with GELU activation that gradually decrease the dimension to 10, using the cross-entropy loss to predict the correct input class. To generalize our results, every line represents the exponential moving average of the results, averaged over five independent random initializations.

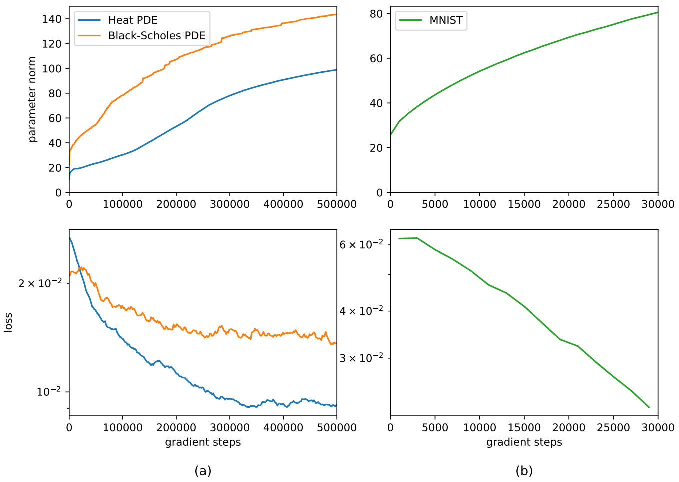


Figure 3: Left: approximation of Heat PDE and Black-Scholes PDE solutions using the deep Kolmogorov method. Right: image classification on the MNIST dataset.

5 Conclusion

We proved that the gradient flow of definable neural networks either converges to a critical point or diverges to infinity – and that the loss values converge to one of finitely many generalized critical values (see Theorem 2.8). We examined the special case of nonlinear polynomial target functions in detail. More specifically, assuming a sufficiently big training dataset and neural network architecture, we showed that if the activation function satisfies a mild condition (see Definition 3.1), then the loss function cannot achieve zero loss (see Theorem 3.4) but it does achieve arbitrarily low values (see Theorem 3.5). From this we deduced our main result stating that in such a situation, the gradient flow diverges to infinity under mild assumptions (see Corollary 3.6). We validated our findings by training neural networks on polynomial target functions using different optimization methods. We then extended our investigation to more realistic scenarios and consistently observed a growth in the norm of the neural network parameters as the loss diminishes, suggesting that the divergence phenomenon we analyzed actually holds in much broader generality (see Section 4).

Acknowledgements

This work has been supported by the Ministry of Culture and Science NRW as part of the Lamarr Fellow Network. In addition, this work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure. This work is partially supported via the AI4Forest project, which is funded by the German Federal Ministry of Education and Research (BMBF; grant number 01IS23025A), and the French National Research Agency (ANR). We gratefully acknowledge the substantial computational resources that were made available to us by the PALMA II cluster at the University of Münster (subsidized by the DFG; INST 211/667-1). The authors are indebted to Floris Vermeulen and Mariana Vicaria for their help in improving the statement of Theorem 3.4. Helpful conversations with Christopher Deninger, Ksenia Fedosova, Allen Gehret, Fabian Gieseke, Vincent Grandjean, and Dennis Wulle are gratefully acknowledged.

References

[1] Pierre-Antoine Absil, Robert Mahony, and Ben Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005. doi:10.1137/040605266.

[2] 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, 2019. URL: https://proceedings.mlr.press/v97/allen-zhu19a.html.

[3] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine 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. doi:10.1287/moor.1100.0449.

[4] Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1):91–129, 2013. doi:10.1007/s10107-011-0484-9.

[5] Christian Beck, Sebastian Becker, Philipp Grohs, Nor Jaafari, and Arnulf Jentzen. Solving the Kolmogorov PDE by means of deep learning. *Journal of Scientific Computing*, 88:1–28, 2021. doi:10.1007/s10915-021-01590-0.

[6] Alessandro Berarducci and Margarita Otero. An additive measure in o-minimal expansions of fields. *Quarterly Journal of Mathematics*, 55(4):411–419, 2004. doi:10.1093/qmath/hah010.

[7] Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. *Mathematical Aspects of Deep Learning*, chapter The Modern Mathematics of Deep Learning, pages 1–111. Cambridge University press, 2022. doi:10.1017/9781009025096.002.

[8] Pascal Bianchi, Walid Hachem, and Sholom Schechtman. Convergence of constant step stochastic gradient descent for non-smooth non-convex functions. *Set-Valued and Variational Analysis*, 30(3):1117–1147, 2022. doi:10.1007/s11228-022-00638-z.

[9] Helmut Bolcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematics of Data Science*, 1(1):8–45, 2019. doi:10.1137/18M118709X.

[10] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007. doi:10.1137/050644641.

[11] Léon Bottou. On-line learning and stochastic approximations. In *On-Line Learning in Neural Networks*, pages 9–42. Cambridge University Press, 1999. doi:10.1017/CBO9780511569920.003.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh,

Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[13] Marshall W. Buck, Raymond A. Coley, and David P. Robbins. A generalized Vandermonde determinant. *Journal of Algebraic Combinatorics*, 1(2):105–109, 1992. doi:10.1023/A:1022468019197.

[14] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *International Conference on Learning Representations*, 2016. arXiv:1511.07289.

[15] M. Coste. *An Introduction to O-minimal Geometry*. Dottorato di ricerca in matematica / Università di Pisa, Dipartimento di Matematica. Istituti Editoriali e Poligrafici Internazionali, 2000. URL: https://perso.univ-rennes1.fr/michel.coste/polyens/OMIN.pdf.

[16] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989. doi:10.1007/BF02551274.

[17] Didier d'Acunto. Valeurs critiques asymptotiques d'une fonction définissable dans une structure o-minimale. *Annales Polonici Mathematici*, 75(1):35–45, 2000. doi:10.4064/ap-75-1-35-45.

[18] Aris Daniilidis, Mounir Haddou, and Olivier Ley. A convex function satisfying the Łojasiewicz inequality but failing the gradient conjecture both at zero and infinity. *Bulletin of the London Mathematical Society*, 54(2):590–608, 2022. doi:10.1112/blms.12586.

[19] Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1):119–154, 2020. doi:10.1007/s10208-018-09409-5.

[20] Joseph Daws and Clayton Webster. Analysis of deep neural networks with quasi-optimal polynomial approximation rates. 2019. arXiv:1912.02302.

[21] Tim De Ryck, Samuel Lanthaler, and Siddhartha Mishra. On the approximation of functions by tanh neural networks. *Neural Networks*, 143:732–750, 2021. doi:10.1016/j.neunet.2021.08.015.

[22] Steffen Dereich and Arnulf Jentzen. Convergence rates for the Adam optimizer. 2024. arXiv:2407.21078.

[23] Steffen Dereich, Arnulf Jentzen, and Sebastian Kassing. On the existence of minimizers in shallow residual ReLU neural network optimization landscapes. *SIAM Journal on Numerical Analysis*, 62(6):2640–2666, 2024. doi:10.1137/23M1556241.

[24] Steffen Dereich and Sebastian Kassing. Cooling down stochastic differential equations: Almost sure convergence. *Stochastic Processes and their Applications*, 152:289–311, October 2022. doi:10.1016/j.spa.2022.06.020.

[25] Steffen Dereich and Sebastian Kassing. Convergence of stochastic gradient descent schemes for Łojasiewicz-landscapes. *Journal of Machine Learning*, 3(3):245–281, 2024. doi:10.4208/jml.240109.

[26] Steffen Dereich and Sebastian Kassing. On the existence of optimal shallow feedforward networks with ReLU activation. *Journal of Machine Learning*, 3(1):1–22, 2024. doi:10.4208/jml.230903.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, pages 4171–4186, 2019. arXiv:1810.04805.

[28] Ronald DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021. doi:10.1017/S0962492921000052.

[29] Thang Do, Sonja Hannibal, and Arnulf Jentzen. Non-convergence to global minimizers in data driven supervised deep learning: Adam and stochastic gradient descent optimization provably fail to converge to global minimizers in the training of deep neural networks with ReLU activation. 2025. arXiv:2410.10533.

[30] Thang Do, Arnulf Jentzen, and Adrian Riekert. Non-convergence to the optimal risk for Adam and stochastic gradient descent optimization in the training of deep neural networks. 2025. arXiv:2503.01660.

[31] Lou van den Dries. *Tame Topology and O-minimal structures*, volume 248. Cambridge University Press, 1998. doi:10.1017/CBO9780511525919.

[32] Lou van den Dries, Angus Macintyre, and David Marker. The elementary theory of restricted analytic fields with exponentiation. *Annals of Mathematics*, 140(1):183–205, 1994. doi:10.2307/2118545.

[33] Lou van den Dries, Angus Macintyre, and David Marker. Logarithmic-exponential power series. *Journal of the London Mathematical Society*, 56(3):417–434, 1997. doi:10.1112/S0024610797005437.

[34] Simon S. Du, Jason D. Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *International Conference on Machine Learning*, 2019. URL: https://proceedings.mlr.press/v97/du19c/du19c.pdf.

[35] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61), 2011. URL: http://jmlr.org/papers/v12/duchi11a.html.

[36] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. *Advances in Neural Information Processing Systems*, 13, 2000. URL: https://proceedings.neurips.cc/paper_files/paper/2000/file/44968aece94f667e4095002d140b5896-Paper.pdf.

[37] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989. doi:10.1016/0893-6080(89)90003-8.

[38] Andrei M. Gabrielov. Projections of semi-analytic sets. *Functional Analysis and its Applications*, 2(4):282–291, 1968. doi:10.1007/BF01075680.

[39] Davide Gallon, Arnulf Jentzen, and Felix Lindner. Blow up phenomena for gradient descent optimization methods in the training of artificial neural networks. 2022. arXiv:2211.15641.

[40] Russell Gentile and Gerrit Welper. Approximation results for gradient descent trained shallow neural networks in $1d$. 2022. arXiv:2209.08399.

[41] Russell Gentile and Gerrit Welper. Approximation results for gradient flow trained shallow neural networks in $1d$. *Constructive Approximation*, 60(3):547–594, 2024. doi:10.1007/s00365-024-09694-0.

[42] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). 2023. arXiv:1606.08415.

[43] G. Hinton. Lecture 6e: RMSprop: Divide the gradient by a running average ofits recent magnitude., 2014. URL: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[44] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. doi:10.1016/0893-6080(89)90020-8.

[45] Xiaoyin Hu, Nachuan Xiao, Xin Liu, and Kim-Chuan Toh. Learning-rate-free momentum SGD with reshuffling converges in nonsmooth nonconvex optimization. *Journal of Scientific Computing*, 102(3):85, 2025. doi:10.1007/s10915-025-02798-0.

[46] Shokhrukh Ibragimov, Arnulf Jentzen, and Adrian Riekert. Convergence to good non-optimal critical points in the training of neural networks: Gradient descent optimization with one random initialization overcomes all bad non-global local minima with high probability. 2025. arXiv:2212.13111.

[47] Alexander D Ioffe. An invitation to tame optimization. *SIAM Journal on Optimization*, 19(4):1894–1917, 2009. doi:10.1137/080722059.

[48] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

[49] Arnulf Jentzen and Adrian Riekert. On the existence of global minima and convergence analyses for gradient descent methods in the training of deep neural networks. *Journal of Machine Learning*, 1(2):141–246, 2022. doi:10.1137/23M1556241.

[50] Arnulf Jentzen and Adrian Riekert. Non-convergence to global minimizers for Adam and stochastic gradient descent optimization and constructions of local minimizers in the training of artificial neural networks. *SIAM/ASA Journal on Uncertainty Quantification*, 13(3):1294–1333, 2025. doi:10.1137/24M1639464.

[51] Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. 33:17176–17186, 2020. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/c76e4b2fa54f8506719a5c0dc14c2eb9-Paper.pdf.

[52] Cédric Josz. Global convergence of the gradient method for functions definable in o-minimal structures. *Mathematical Programming*, 202(1):355–383, 2023. doi:10.1007/s10107-023-01937-5.

[53] Cédric Josz and Lexiao Lai. Global stability of first-order methods for coercive tame functions. *Mathematical Programming*, 207(1):551–576, 2024. doi:10.1007/s10107-023-02020-9.

[54] Cédric Josz and Xiaopeng Li. Certifying the absence of spurious local minima at infinity. *SIAM Journal on Optimization*, 33(3):1416–1439, 2023. doi:10.1137/22M1479531.

[55] Marek Karpinski and Angus Macintyre. Polynomial bounds for VC dimension of sigmoidal neural networks. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 200–208, 1995. doi:10.1145/225058.225118.

[56] Marek Karpinski and Angus Macintyre. Polynomial bounds for VC dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, 1997. doi:10.1006/jcss.1997.1477.

[57] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014. arXiv:1412.6980.

[58] Julia F Knight, Anand Pillay, and Charles Steinhorn. Definable sets in ordered structures. II. *Transactions of the American Mathematical Society*, 295(2):593–605, 1986. doi:10.1090/S0002-9947-1986-0833698-1.

[59] Akshay Kumar and Jarvis Haupt. Directional convergence near small initializations and saddles in two-homogeneous neural networks, 2024. URL: https://openreview.net/forum?id=hfrPag75Y0.

[60] Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'institut Fourier*, 48(3):769–783, 1998. doi:10.5802/aif.1638.

[61] Krzysztof Kurdyka and Adam Parusinski. Quasi-convex decomposition in o-minimal structures. application to the gradient conjecture. *Advanced Studies in Pure Mathematics.*, 43:137–177, 2006. doi:10.2969/aspm/04310137.

[62] Lexiao Lai. *Stability of first-order methods in tame optimization*. PhD thesis, Columbia University, 2024. arXiv:2412.00640.

[63] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi:10.1109/5.726791.

[64] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993. doi:10.1016/S0893-6080(05)80131-5.

[65] Haochuan Li, Jian Qian, Yi Tian, Alexander Rakhlin, and Ali Jadbabaie. Convex and non-convex optimization under generalized smoothness. In *Advances in Neural Information Processing Systems*, volume 36, pages 40238–40271, 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/7e8bb8d17bb1cb24dfe972a2f8ff2500-Paper-Conference.pdf.

[66] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. arXiv:1711.05101.

[67] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020. arXiv:1906.05890.

[68] Diganta Misra. Mish: A self regularized non-monotonic activation function. In *British Machine Vision Conference*, 2020. arXiv:1908.08681.

[69] Joost AA Opschoor, Philipp C Petersen, and Christoph Schwab. Deep ReLU networks and high-order finite element methods. *Analysis and Applications*, 18(05):715–770, 2020. doi:10.1142/S0219530519410136.

[70] Philipp Petersen, Mones Raslan, and Felix Voigtlaender. Topological properties of the set of functions generated by neural networks of fixed size. *Found. Comput. Math.*, 21(2):375–444, 2021. doi:10.1007/s10208-020-09461-0.

[71] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018. doi:10.1016/j.neunet.2018.08.019.

[72] Anand Pillay and Charles Steinhorn. Definable sets in ordered structures. I. *Transactions of the American Mathematical Society*, 295(2):565–592, 1986. doi:10.1090/S0002-9947-1986-0833697-X.

[73] Anand Pillay and Charles Steinhorn. Definable sets in ordered structures. III. *Transactions of the American Mathematical Society*, pages 469–476, 1988. doi:10.2307/2000920.

[74] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8:143–195, 1999. doi:10.1017/S0962492900002919.

[75] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. doi:10.1016/0041-5553(64)90137-5.

[76] Patrick J. Rabier. Ehresmann fibrations and Palais-Smale conditions for morphisms of Finsler manifolds. *Annals of Mathematics*, pages 647–691, 1997. doi:10.2307/2952457.

[77] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951. doi:10.1214/aoms/1177729586.

[78] Gal Vardi, Ohad Shamir, and Nati Srebro. On margin maximization in linear and ReLU networks. In *Advances in Neural Information Processing Systems*, volume 35, pages 37024–37036, 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/f062da1973ac9ac61fc6d44dd7fa309f-Paper-Conference.pdf.

[79] Alex J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094, 1996. doi:10.1090/S0894-0347-96-00216-0.

[80] Alex J. Wilkie. A theorem of the complement and some new o-minimal structures. *Selecta Mathematica*, 5:397–421, 1999. doi:10.1007/s000290050052.

[81] Alex J. Wilkie. o-minimal structures. *Astérisque*, (326):Exp. No. 985, vii, 131–142, 2009. Séminaire Bourbaki. Vol. 2007/2008. URL: https://www.numdam.org/article/AST_2009__326__131_0.pdf.

[82] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Metworks*, 94:103–114, 2017. doi:10.1016/j.neunet.2017.07.002.

[83] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *International Conference on Learning Representations*, 2020. arXiv:1905.11881.

NeurIPS Paper Checklist

(i) **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Both the abstract and the introduction informally state our main theorems including their assumtions: The dichotomy for definable gradient flows and the divergence result for polynomial target functions. The numerical experiments confirming our theoretical anylsis are outlined as well.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

(ii) **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss to what extent our theoretical analysis resembles real-world learning tasks, and point out our inability of our mathematical tools to tackle the ReLU function or stochastic gradient descent rather than gradient flows. We also acknowledge the limited size of our datasets and emphasized our focus on the mathematical theorems.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

(iii) **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main paper provides mathematically complete and rigorous statements (theorems and corollaries) of the main claims including all assumptions. All mathematical claims are either referenced to appropriate sources or proved in full detail in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

(iv) **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes] .

Justification: Our models are quite simple, the main details are in Section 4 and Appendix E. Moreover, detailed instruction for reproducing our experiments are provided in the README file included with the supplementary code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

(v) **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to our code, including configuration files and detailed instructions to faithfully reproduce all main experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

(vi) **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify the key training and test details in Section 4 and Appendix E. All remaining parameters are included in the configuration files available in the code repository.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

(vii) **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We did not report error bars because our main focus is to illustrate the growth of network parameters as the loss vanishes. To reduce variance, we average results over multiple independent runs, which consistently show the same qualitative behavior.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

(viii) **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We specify the resources in Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

(ix) **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conforms the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

(x) **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our work contributes towards a theoretical mathematical understanding of why gradient based optimization works the way it does. Our work does not have an immediate impact on society since it does not develop any novel algorithm and does not use any data apart from the MNIST dataset.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

(xi) **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Being a mathematical work focusing on general theory, our paper does not release any relevant models or data. There are no risks for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

(xii) **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cited the publicly available MNIST dataset.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

(xiii) **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The only asset released in the paper is the code for producing the numerical experiments. The code is published along with sufficient documentation and license information in an anonymous GitHub repository.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

(xiv) **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

(xv) **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

Appendix

A Locally Lipschitz derivatives

The aim of this section is to establish that the expected loss appearing in Theorem 2.8 has a locally Lipschitz derivative whenever the activation function and the loss function have a locally Lipschitz derivative. This is needed to ensure uniqueness of the gradient flow for the ELU function, the softsign function and the Huber loss in Example 2.4. This section can safely be skipped if one is willing to assume that the activation function and the loss function are C^2.

Lemma A.1. *The class functions with locally Lipschitz derivatives (see Definition 2.7) is closed under the following operations.*

(i) Every C^2-function has a locally Lipschitz derivative.

(ii) A function $f = (f_1, \dots, f_m)\colon \mathbb{R}^n \to \mathbb{R}^m$ has a locally Lipschitz derivative if and only if all of its components $f_1, \dots, f_m\colon \mathbb{R}^n \to \mathbb{R}$ have locally Lipschitz derivatives.

(iii) If $f\colon \mathbb{R}^{n_1} \to \mathbb{R}$ and $g\colon \mathbb{R}^{n_2} \to \mathbb{R}$ have locally Lipschitz derivatives, then

$$f \oplus g\colon \mathbb{R}^{n_1+n_2} \to \mathbb{R}, \quad f \oplus g(x, y) = f(x) + g(y)$$

and

$$f \odot g\colon \mathbb{R}^{n_1+n_2} \to \mathbb{R}, \quad f \odot g(x, y) = f(x)g(y)$$

have locally Lipschitz derivatives.

(iv) If $f\colon \mathbb{R}^m \to \mathbb{R}^k$ and $g\colon \mathbb{R}^n \to \mathbb{R}^m$ have locally Lipschitz derivatives, then

$$f \circ g\colon \mathbb{R}^n \to \mathbb{R}^k, \quad f \circ g(x) = f(g(x))$$

has a locally Lipschitz derivative.

(v) Let $a, b \in \mathbb{R}$ with $a < b$ and suppose that $f\colon \mathbb{R}^n \to \mathbb{R}$ has a locally Lipschitz derivative. Then the map

$$F\colon \mathbb{R}^{n-1} \to \mathbb{R}, \quad F(x) = \int_a^b f(t, x)dt$$

has a locally Lipschitz derivative.

Proof. (i) This is a well-known consequence of the fundamental theorem of calculus.

(ii) This follows from the inequality

$$\max_{1\le i\le m} \|Df_i(x)\| \le \|Df(x)\| \le \sqrt{m} \max_{1\le i\le m} \|Df_i(x)\|$$

for all $x \in \mathbb{R}^n$.

(iii) We only prove that $f \odot g$ has a locally Lipschitz derivative since the proof for $f \oplus g$ is similar. Fix $(x_0, y_0) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} = \mathbb{R}^{n_1+n_2}$. Fix $L_f, L_g > 0$ and relatively compact open neighborhoods $x_0 \in U_f \subseteq \mathbb{R}^{n_1}$ and $y_0 \in U_g \subseteq \mathbb{R}^{n_2}$ satisfying

$$\|Df(x_1) - Df(x_2)\| \le L_f \|x_1 - x_2\|, \quad \forall x_1, x_2 \in U_f$$

and

$$\|Dg(y_1) - Dg(y_2)\| \le L_g \|y_1 - y_2\|, \quad \forall y_1, y_2 \in U_g.$$

Since f and g are C^1 and thus locally Lipschitz, there are constants $L'_f, L'_g > 0$ such that

$$\|f(x_1) - f(x_2)\| \le L'_f \|x_1 - x_2\|, \quad \forall x_1, x_2 \in U_f$$

and

$$\|g(y_1) - g(y_2)\| \le L'_g \|y_1 - y_2\|, \quad \forall y_1, y_2 \in U_g.$$

By continuity and relative compactness, the quantities

$$K_f := \sup_{x\in U_f} |D_f(x)|, \quad K'_f := \sup_{x\in U_f} |f(x)|, \quad K_g := \sup_{y\in U_g} |D_g(y)|, \quad K'_g := \sup_{y\in U_g} |g(y)|$$

are all finite. Defining

$$L \coloneqq K_f L_g' + K_f' L_g + K_g' L_f + K_g L_f' < \infty,$$

we have for all $(x_1, y_1), (x_2, y_2) \in U_f \times U_g$ that

$$\begin{aligned}
&\|D(f \odot g)(x_1, y_1) - D(f \odot g)(x_2, y_2)\| \\
=&\left\| \begin{pmatrix} Df(x_1)g(y_1) \\ f(x_1)Dg(y_1) \end{pmatrix} - \begin{pmatrix} Df(x_2)g(y_2) \\ f(x_2)Dg(y_2) \end{pmatrix} \right\| \\
\leq&\|Df(x_1)g(y_1) - Df(x_2)g(y_2)\| + \|f(x_1)Dg(y_1) - f(x_2)Dg(y_2)\| \\
\leq&\|Df(x_1)\|\|g(y_1) - g(y_2)\| + \|Df(x_1) - Df(x_2)\|\|g(y_2)\| \\
&+ \|f(x_1) - f(x_2)\|\|Dg(y_1)\| + \|f(x_2)\|\|Dg(y_1) - Dg(y_2)\| \\
\leq&(K_f L_g' + K_f' L_g)\|y_1 - y_2\| + (K_g' L_f + K_g L_f')\|x_1 - x_2\| \\
\leq&L\|(x_1, y_1) - (x_2, y_2)\|.
\end{aligned}$$

This proves that $f \odot g$ has a locally Lipschitz derivative.

(iv) Fix $x_0 \in \mathbb{R}^n$. By assumption, there are $L_f, L_g > 0$ and relatively compact open neighbourhoods $g(x_0) \in U_f \subseteq \mathbb{R}^m$ and $x_0 \in U_g \subseteq \mathbb{R}^n$ satisfying

$$\|Df(y_1) - Df(y_2)\| \leq L_f \|y_1 - y_2\|, \quad \forall y_1, y_2 \in U_f$$

and

$$\|Dg(x_1) - Dg(x_2)\| \leq L_g \|x_1 - x_2\|, \quad \forall x_1, x_2 \in U_g.$$

By continuity of g, we may assume that $g(U_g) \subseteq U_f$. Since g is C^1 and thus itself locally Lipschitz, we may moreover assume that there is $L_g' > 0$ such that

$$\|g(x_1) - g(x_2)\| \leq L_g' \|x_1 - x_2\|, \quad \forall x_1, x_2 \in U_g.$$

By continuity of Df and Dg and relative compactness of U_f and U_g, we have

$$L \coloneqq L_f L_g' \sup_{x \in U_g} \|Dg(x)\| + L_g \sup_{y \in U_f} \|Df(y)\| < \infty.$$

Moreover, for all $x_1, x_2 \in U_f$, we have

$$\begin{aligned}
\|D(f \circ g)(x_1) - D(f \circ g)(x_2)\| =&\|Df(g(x_1))Dg(x_1) - Df(g(x_2))Dg(x_2)\| \\
\leq&\|Df(g(x_1)) - Df(g(x_2))\|\|Dg(x_1)\| \\
&+ \|Df(g(x_2))\|\|Dg(x_1) - Dg(x_2)\| \\
\leq&L\|x_1 - x_2\|.
\end{aligned}$$

This proves that $f \circ g$ has a locally Lipschitz derivative.

(v) Fix $x_0 \in \mathbb{R}^n$. Since f has a locally Lipschitz derivative, there is for every $t \in [a, b]$ a constant $L_t > 0$ and an open neighbourhood $(t, x_0) \in U_t \subseteq \mathbb{R}^n$ such that for all $(t_1, x_1), (t_2, x_2) \in U_t$, we have

$$\|Df(t_1, x_1) - Df(t_2, x_2)\| < L_t \|(t_1, x_1) - (t_2, x_2)\|.$$

By compactness, we can find $t_1, \ldots, t_k \in [a, b]$ such that $[a, b] \times \{x_0\} \subseteq \bigcup_{i=1}^k U_{t_i}$. Without loss of generality, we may assume that each U_{t_i} for $i = 1, \ldots, k$ is of the form $(a_i, b_i) \times V$ for an open neighbourhood $x_0 \in V \subseteq \mathbb{R}^{n-1}$ and $a_i, b_i \in \mathbb{R}$. Now define $L \coloneqq (b - a) \max_{1 \leq i \leq k} L_{t_i}$ and let $x_1, x_2 \in V$ be arbitrary. Using the Leibniz integral rule, we get

$$\begin{aligned}
\|DF(x_1) - DF(x_2)\| &= \left\| \int_a^b (D_x f(t, x_1) - D_x f(t, x_2)) dt \right\| \\
&\leq \int_a^b \|Df(t, x_1) - Df(t, x_2)\| dt \\
&\leq \int_a^b \max_{1 \leq i \leq k} L_{t_i} \|x_1 - x_2\| dt \\
&\leq L\|x_1 - x_2\|,
\end{aligned}$$

where $D_x f(t, x_i)$ denotes the differential of the map $x \mapsto f(t, x)$ at $x = x_i$. This proves that F has a locally Lipschitz derivative.

□

Corollary A.2. *Let $\mathcal{A} = (d_0, \dots, d_k, \psi)$ be a neural network architecture with a C^1 activation function $\psi\colon \mathbb{R} \to \mathbb{R}$ with a locally Lipschitz derivative. Let $\ell\colon \mathbb{R}^{d_k} \times \mathbb{R}^{d_k} \to [0, \infty)$ be a C^1 loss function with a locally Lipschitz derivative. Let $f\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_k}$ be a C^1 target function with a locally Lipschitz derivative. Let μ be a probability measure on $\mathbb{R}^{d_0}$ such that either*

(i) $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ *for some $n \in \mathbb{N}$ and $x_1, \dots, x_n \in \mathbb{R}^{d_0}$, or*

(ii) $d\mu(x) = p(x)dx$ *for a C^1 function $p\colon \mathbb{R}^{d_0} \to [0, \infty)$ of compact support.*

Then the function

$$\mathcal{L}\colon \mathbb{R}^{d(\mathcal{A})} \to \mathbb{R}, \quad \mathcal{L}(\theta) = \mathbb{E}_{x \sim \mu}[\ell(\mathcal{N}_\theta^{\mathcal{A}}(x), f(x))]$$

has a locally Lipschitz derivative.

Proof. This immediately follows from Lemma A.1 since $\mathcal{L}$ can be constructed iteratively from ψ and linear functions using the operations (i) - (v). □

B Proof of Theorem 2.8

This section gives a proof of Theorem 2.8 which is probably well-known to experts and follows standard techniques. The proof is mostly given for lack of a precise reference and for convenience of the reader. Before giving the technical details, we outline the main ideas below.

Idea of proof

Existence and uniqueness of the gradient flow for $\mathcal{L}$ follow from the fact that $\mathcal{L}$ is C^1 and has a locally Lipschitz derivative. The latter assumption is checked in Appendix A. Now one can distinguish two cases: Either the gradient flow visits some bounded set infinitely often or it does not. In the first case, we use Kurdyka's definable Łojasiewicz inequality [60] and follow an argument from [1, Theorem 2.2] to prove convergence to a critical point (see Lemma B.1). In the second case, the parameters diverge by assumption. We check that the limit is a generalized critical value using the fact that $\mathcal{L}$ is bounded below and therefore has square-integrable gradients along any gradient flow trajectory (see Lemma B.2).

For a point $x_0 \in \mathbb{R}^n$ and $R > 0$, we denote by

$$B_R(x_0) \coloneqq \{x \in \mathbb{R}^n \mid \|x - x_0\| < R\} \subseteq \mathbb{R}^n$$

the open ball of radius R centered at x_0.

Lemma B.1. *Let $\mathcal{L}\colon \mathbb{R}^d \to [0, \infty)$ be a C^1 definable function and let $\Theta\colon [0, \infty) \to \mathbb{R}^d$ a C^1-function satisfying the differential equation $\Theta'(t) = -\nabla\mathcal{L}(\Theta(t))$ for all $t \in [0, \infty)$. Suppose that there is a point $\theta_* \in \mathbb{R}^d$ satisfying*

$$\liminf_{t \to \infty} \|\Theta(t) - \theta_*\| = 0. \tag{1}$$

Then we have $\lim_{t\to\infty} \Theta(t) = \theta_*$.

Proof. Without loss of generality, we may assume that $\mathcal{L}(\theta_*) = 0$. Note that the function $\mathcal{L} \circ \Theta\colon [0, \infty) \to \mathbb{R}$ is non-increasing since we have $\frac{d}{dt}\mathcal{L}(\Theta(t)) = -\|\nabla\mathcal{L}(\Theta(t))\|^2$ for all $t \in [0, \infty)$. Since $\mathcal{L}$ is continuous, (1) and the fact that $\mathcal{L}$ is non-increasing imply that

$$\lim_{t \to \infty} \mathcal{L}(\Theta(t)) = \mathcal{L}(\theta_*) = 0. \tag{2}$$

Now fix $\varepsilon > 0$. By [60, Theorem 2 (b)], there is a continuous strictly increasing definable function $\sigma\colon [0, \infty) \to [0, \infty)$ with $\sigma(0) = 0$ such that whenever $t_1, t_2 \in [0, \infty)$ with $t_1 < t_2$ satisfy $\theta([t_1, t_2]) \subseteq B_{2\varepsilon}(\theta_*)$, we have

$$\int_{t_1}^{t_2} \|\Theta'(t)\| dt \leq \sigma(|\mathcal{L}(\Theta(t_2)) - \mathcal{L}(\Theta(t_1))|). \tag{3}$$

Since $t \mapsto \mathcal{L}(\Theta(t))$ is nonincreasing, (3) implies that

$$\int_{t_1}^{t_2} \|\Theta'(t)\| dt \leq \sigma(|\mathcal{L}(\Theta(t_2)) - \mathcal{L}(\Theta(t_1))|) \leq \sigma(\mathcal{L}(\Theta(t_1))) \tag{4}$$

for all $t_1, t_2 \in [0, \infty)$ satisfying $\Theta([t_1, t_2]) \subseteq B_{2\varepsilon}(\theta_*)$.

By continuity of σ and (2), we can find a large enough $t_1 > 0$ such that $\sigma(\mathcal{L}(\Theta(t_1))) < \frac{\varepsilon}{2}$. By (1), we may moreover assume that $\Theta(t_1) \in B_{\frac{\varepsilon}{2}}(\theta_*)$. We claim that we have $\Theta(t) \in B_\varepsilon(\theta_*)$ for all $t > t_1$.

Assuming the contrary, there is a smallest $t_2 > t_1$ satisfying $\|\Theta(t_2) - \Theta_*\| = \varepsilon$. But then we have $\Theta([t_1, t_2]) \subseteq B_{2\varepsilon}(\theta_*)$ and by an application of (4), we get

$$\begin{aligned} \|\Theta(t_2) - \theta_*\| &\leq \|\Theta(t_2) - \Theta(t_1)\| + \|\Theta(t_1) - \theta_*\| \\ &< \int_{t_1}^{t_2} \|\Theta'(t)\| dt + \frac{\varepsilon}{2} \\ &\leq \sigma(\mathcal{L}(\Theta(t_1))) + \frac{\varepsilon}{2} \\ &< \varepsilon, \end{aligned}$$

a contradiction. Thus, we have $\Theta([t_1, \infty)) \subseteq B_\varepsilon(\theta_*)$. Since $\varepsilon > 0$ was arbitrary, the lemma is proven. □

Lemma B.2. *Let $\mathcal{L}\colon \mathbb{R}^d \to [0, \infty)$ be a C^1 function. Then for any $\theta_0 \in \mathbb{R}^d$, there is a C^1-function $\Theta\colon [0, \infty) \to \mathbb{R}^d$ satisfying*

$$\Theta'(t) = -\nabla\mathcal{L}(\Theta(t)), \quad \Theta(0) = \theta_0 \tag{5}$$

for all $t \in [0, \infty)$. Moreover, the following statements are true:

(i) If $\mathcal{L}$ moreover has a locally Lipschitz derivative, then the solution Θ to (5) is unique.

(ii) If $\mathcal{L}$ is moreover definable, then exactly one of the following two statements holds true.

(a) $\lim_{t\to\infty} \Theta(t)$ exists and is a critical point of f, or

(b) $\lim_{t\to\infty} \|\Theta(t)\| = \infty$ and $\lim_{t\to\infty} \mathcal{L}(\Theta(t))$ is a generalized critical value of $\mathcal{L}$ (see Definition 2.6).

Proof. The existence of a C^1-map $\Theta\colon [0, \infty) \to \mathbb{R}^d$ satisfying (5) follows for instance from [54, Proposition 2.11]. The uniqueness part in item (i) follows from the Picard–Lindelöf theorem.

Now assume that $\mathcal{L}$ is moreover definable. We consider two cases:

Case 1: We have $\liminf_{t\to\infty} \|\Theta(t)\| < \infty$.

In this case, there is a constant $R > 0$ and a sequence $(t_n)_{n\in\mathbb{N}}$ in $[0, \infty)$ satisfying $\Theta(t_n) \in B_R(0)$ for all $n \in \mathbb{N}$ and $\lim_{n\to\infty} t_n = \infty$. By compactness, a subsequence of $(\Theta(t_n))_{n\in\mathbb{N}}$ converges to some point $\theta_* \in B_R(0)$. In particular, we have $\liminf_{t\to\infty} \|\Theta(t) - \theta_*\| = 0$. Therefore, Lemma B.1 implies that

$$\lim_{t\to\infty} \Theta(t) = \theta_*. \tag{6}$$

To check that θ_* is a critical point, note that since $\mathcal{L}$ takes only non-negative values, we have

$$\int_0^t \|\Theta'(s)\|^2 ds = -\int_0^t \langle\Theta'(s), \nabla\mathcal{L}(\Theta(s))\rangle ds = \mathcal{L}(\Theta(0)) - \mathcal{L}(\Theta(t)) \leq \mathcal{L}(\Theta(0)) \tag{7}$$

for all $t \in [0, \infty)$. In particular, $t \mapsto \|\Theta'(t)\|$ is square-integrable. In combination with (6), this proves

$$\|\nabla\mathcal{L}(\theta_*)\| = \liminf_{t\to\infty} \|\nabla\mathcal{L}(\Theta(t))\| = \liminf_{t\to\infty} \|\Theta'(t)\| = 0,$$

so that θ_* is a critical point of $\mathcal{L}$.

Case 2: We have $\liminf_{t\to\infty} \|\Theta(t)\| = \infty$.

In this case, the limit $c := \lim_{t\to\infty} \mathcal{L}(\Theta(t))$ exists since the map $t \mapsto \mathcal{L}(\Theta(t))$ is decreasing and bounded below. It remains to be shown that $\liminf_{t\to\infty} \|\Theta(t)\|\|\nabla\mathcal{L}(\Theta(t))\| = 0$. By [39, Proposition 2.54], we have

$$\|\Theta(t)\| \leq \|\Theta(0)\| + \sqrt{t\mathcal{L}(\Theta(0))}, \tag{8}$$

and thus

$$\|\Theta(t)\|\|\nabla\mathcal{L}(\Theta(t))\| = \|\Theta(t)\|\|\Theta'(t)\| \leq \left(\|\Theta(0)\| + \sqrt{t\mathcal{L}(\Theta(0))}\right)\|\Theta'(t)\|, \tag{9}$$

for every $t \in [0, \infty)$.

Assume by contradiction that we have

$$\varepsilon := \liminf_{t\to\infty} \left(\|\Theta(0)\| + \sqrt{t\mathcal{L}(\Theta(0))}\right)\|\Theta'(t)\| > 0.$$

Then there exists $T \in (0, \infty)$ such that $\inf_{t\geq T} \left(\|\Theta(0)\| + \sqrt{t\mathcal{L}(\Theta(0))}\right)\|\Theta'(t)\| > \frac{\varepsilon}{2}$. Thus, we get

$$\infty = \int_T^\infty \left(\|\Theta(0)\| + \sqrt{t\mathcal{L}(\Theta(0))}\right)^{-2} dt \leq \frac{4}{\varepsilon^2}\int_T^\infty \|\Theta'(t)\|^2 dt,$$

contradicting (7). Thus we have $\liminf_{t\to\infty} \left(\|\Theta(0)\| + \sqrt{t\mathcal{L}(\Theta(0))}\right)\|\Theta'(t)\| = 0$. Together with (9), this proves that $\liminf_{t\to\infty} \|\Theta(t)\|\|\nabla\mathcal{L}(\Theta(t))\| = 0$ and concludes the proof of the lemma. □

Proof of Theorem 2.8. It is either easy to see or stated as an assumption that $\mathcal{L}$ is definable. Moreover, it is an easy consequence of the Leibniz integral rule that $\mathcal{L}$ is C^1. Thus, [17] is applicable and establishes that the set $K(\mathcal{L})$ of generalized critical values of $\mathcal{L}$ (see Definition 2.6) is finite. Moreover, $\mathcal{L}$ has a locally Lipschitz derivative by Corollary A.2. Now the theorem follows from Lemma B.2 if we choose $\varepsilon > 0$ small enough so that the open interval $(\inf_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta), \inf_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta) + \varepsilon)$ does not contain any generalized critical value. □

C Proof of Theorem 3.4

This section is dedicated to the proof of our main technical contribution Theorem 3.4. This is the main part where o-minimal structures are used in a new way which cannot be found in the neural network literature. We outline the main ideas of the proof below.

Idea of proof

The starting point is that the sublinearity condition (S) in Definition 3.1 also implies that the neural network responses $\mathcal{N}_\theta^\mathcal{A}$ are sublinear (see Lemma C.2). In particular, it cannot be equal as a function to a polynomial f of degree at least 2 (see Lemma C.3). This does not explain why $\mathcal{L}$ does not have any zeros yet since $\mathcal{N}_\theta^\mathcal{A}$ and f could be equal on the dataset or data distribution for some θ. The main bulk of the proof now exploits the remarkable rigidity properties of definable analytic functions to show that this cannot happen (see Lemma C.4). We treat the one-dimensional and multi-dimensional input cases separately. In the one-dimensional case, the combination of the identity theorem for analytic functions and the uniform finiteness theorem from o-minimal geometry imply that for any θ, $\mathcal{N}_\theta^\mathcal{A}$ and f cannot be equal on a sufficiently large finite dataset where the required size can be made independent of θ. In the multidimensional case, the conclusion can fail for very degenerate datasets (for example if all datapoints lie on a line along which the polynomial is linear). But by combining the ideas from the one-dimensional case with dimension theory for definable sets, we show that the space of such "bad datasets" has strictly smaller dimension than the space of all datasets. Therefore its complement contains an open dense subset of full Lebesgue measure. The use of definable dimension theory was kindly suggested to us by Floris Vermeulen and Mariana Vicaria.

Lemma C.1. *Let $f\colon \mathbb{R} \to \mathbb{R}$ be a C^1 definable function. Then f satisfies condition (S) of Definition 3.1 if and only if its derivative is a bounded function.*

Proof. Assume first that $C := \sup_{t\in\mathbb{R}} |f'(t)| < \infty$ and fix $x \in \mathbb{R}$. Then the fundamental theorem of calculus implies for any $t \in (0, \infty)$ that

$$\frac{|f(tx)|}{t} \leq \frac{1}{t}\left(|f(0)| + \int_0^1 |f'(stx)||tx|ds\right) \leq \frac{|f(0)|}{t} + C|x| \xrightarrow{t\to\infty} C|x| < \infty,$$

proving condition (S). We prove the converse and assume f' is unbounded. Since f' is definable [15, Lemma 6.1], it is monotone outside a compact interval $[-R, R]$ by the monotonicity theorem [15, Theorem 2.1]. Without loss of generality, we may assume that f' is increasing and unbounded on $[R, \infty)$. The cases that f is decreasing and unbounded on $[R, \infty)$, increasing and unbounded on $(-\infty, R]$, or decreasing and unbounded on $(-\infty, R]$ are analogous. Fix a constant $K > 0$. Since f' is increasing and unbounded on $[R, \infty)$, there is a constant $R' > R$ such that $f'(t) > K$ for all $t > R'$. Thus for all $t \in [R', \infty)$, we have

$$\begin{aligned}\limsup_{t\to\infty} \frac{f(t\cdot 1)}{t} &= \limsup_{t\to\infty} \left| \frac{f(R')}{t} + \frac{1}{t}\int_{R'}^{t} f'(s)ds \right| \\ &\geq \limsup_{t\to\infty} \left(\frac{-|f(R')|}{t} + \frac{1}{t}\int_{R'}^{t} f'(s)ds \right) \\ &\geq \limsup_{t\to\infty} \frac{-|f(R')| + (t-R')K}{t} \\ &= K.\end{aligned}$$

Since $K > 0$ was arbitrary, we conclude that $\limsup_{t\to\infty} \frac{f(t\cdot 1)}{t} = \infty$, disproving condition (S). $\square$

Lemma C.2. *Let $\mathcal{A} = (d_0, \ldots, d_k, \psi)$ be a neural network architecture and $\theta \in \mathbb{R}^{d(\mathcal{A})}$. If the activation $\psi\colon \mathbb{R} \to \mathbb{R}$ is SAD, then the response $\mathcal{N}_\theta^{\mathcal{A}}\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_k}$ is SAD, too.*

Proof. It is easy to see that $\mathcal{N}_\theta^{\mathcal{A}}$ satisfies conditions (A) and (D) of Definition 3.1 for all $\theta \in \mathbb{R}^d$. It remains to verify (S). Since condition (S) can be checked on each component of $\mathcal{N}_\theta^{\mathcal{A}}$ we may assume without loss of generality that $\mathcal{A} = (d_0, \ldots, d_k, \psi)$ has only $d_k = 1$ output neuron.

We prove that $\mathcal{N}_\theta^{\mathcal{A}}$ satisfies condition (S) by induction on k. Assume first that $k = 1$. Then $\mathcal{N}_\theta^{\mathcal{A}}$ is of the form

$$\mathcal{N}_\theta^{\mathcal{A}}(x) = w_1 x_1 + \cdots + w_{d_0} x_{d_0} + b$$

for $\theta = (w_1, \ldots, w_n, b)$. Then, $\mathcal{N}_\theta^{\mathcal{A}}$ is affine linear and thus satisfies condition (S).

Assume now that we have proven the statement for $k-1$ and fix $\theta \in \mathbb{R}^d$ and $x \in \mathbb{R}^{d_0}$. We define a smaller architecture $\mathcal{A}' = (d_0, \ldots, d_{k-2}, 1, \psi)$ with only $k-1$ layers by deleting the $(k-1)$-th layer. Note that by rearranging the components of $\theta \in \mathbb{R}^d$ as $\theta = (\theta_1, \ldots, \theta_{d_{k-1}}, w_1, \ldots, w_{d_{k-1}}, b)$, for appropriate $w_1, \ldots, w_{d_{k-1}}, b \in \mathbb{R}$ and $\theta_1, \ldots, \theta_{d_{k-1}} \in \mathbb{R}^{(d-1)/d_{k-1}-1}$, we can write

$$\mathcal{N}_\theta^{\mathcal{A}}(tx) = \sum_{i=1}^{d_{k-1}} w_i \psi(\mathcal{N}_{\theta_i}^{\mathcal{A}'}(tx)) + b \tag{10}$$

for all $t \in \mathbb{R}$. By the induction hypothesis, the functions $t \mapsto \mathcal{N}_{\theta_i}^{\mathcal{A}'}(tx)$ satisfiy condition (S) for every $i = 0, \ldots, d_{k-1}$. In particular, they have bounded derivatives by Lemma C.1. By another application of Lemma C.1 and the chain rule, it follows that the functions $t \mapsto w_i \psi(\mathcal{N}_{\theta_i}^{\mathcal{A}'}(tx))$ for $i = 0, \ldots, d_{k-1}$ satisfy condition (S), too. By linearity and Lemma C.1 applied to (10), it follows that the function $t \mapsto \mathcal{N}_\theta^{\mathcal{A}}(tx)$ satisfies condition (S). Since $x \in \mathbb{R}^{d_0}$ was arbitrary, we conclude that $\mathcal{N}_\theta^{\mathcal{A}}$ satisfies condition (S). $\square$

Lemma C.3. *Let $f\colon \mathbb{R}^m \to \mathbb{R}^r$ be a polynomial. Then f is SAD if and only if f is affine linear.*

Proof. Note that by definition, polynomials are analytic and definable. Moreover, every linear function $p\colon \mathbb{R}^m \to \mathbb{R}^r$ satisfies condition (S) in Definition 3.1 since we have $\frac{\|p(tx)\|}{t} = \|p(x)\|$ for all $x \in \mathbb{R}^m$ and $t \in (0, \infty)$. Conversely, let $p\colon \mathbb{R}^m \to \mathbb{R}^r$ be polynomial of degree at least 2 and write $p(x) = (p_1(x), \ldots, p_r(x))$ for $x \in \mathbb{R}^m$ and polynomials $p_1, \ldots, p_r\colon \mathbb{R}^m \to \mathbb{R}$. To prove that p does not satisfy condition (S), it suffices to prove that one of the components $p_1, \ldots, p_r$ does not satisfy condition (S). Thus, we may without loss of generality assume that $r = 1$. We write p as $p(x) = p_H(x) + p_L(x)$ where p_H is a homogeneous polynomial of degree $k \geq 2$ and p_L is a polynomial of degree strictly less than k. Fix $x_0 \in \mathbb{R}^m$ such that $\|x_0\| = 1$ and $p_H(x_0) \neq 0$. Since p_L has degree strictly less than k, the limit $\alpha \coloneqq \lim_{t\to\infty} \frac{|p_L(tx_0)|}{t^{k-1}} \in \mathbb{R}$ exists. We have

$$\begin{aligned}
\limsup_{t\to\infty} \frac{|p(tx_0)|}{t} &\geq \limsup_{t\to\infty} \frac{|p(tx_0)|}{t^{k-1}} \\
&\geq \limsup_{t\to\infty} \left(\frac{|p_H(tx_0)|}{t^{k-1}} - \frac{|p_L(tx_0)|}{t^{k-1}} \right) \\
&\geq \limsup_{t\to\infty} \frac{t^k |p_H(x_0)|}{t^{k-1}} - \alpha \\
&= \infty.
\end{aligned}$$

□

The next lemma helps us to detect non-representability of polynomials on a finite dataset. We are indebted to Floris Vermeulen and Mariana Vicaria for explaining to us the proof in the case $m > 1$.

Lemma C.4. *Let $f\colon \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}$ be a definable analytic function such that for each $\theta \in \mathbb{R}^d$, the function $x \mapsto f(\theta, x)$ is not identically zero. Then there is an integer $N \in \mathbb{N}$ such that the interior of the set*

$$\mathcal{D}_N \coloneqq \{(x_1, \dots, x_N) \mid \forall \theta \in \mathbb{R}^d \colon \exists i = 1, \dots, N \colon f(\theta, x_i) \neq 0\} \subseteq (\mathbb{R}^m)^N$$

is conull and dense in $(\mathbb{R}^m)^N$. If we moreover assume that $m = 1$, then N can be chosen such that $\mathcal{D}_N$ contains all tuples $(x_1, \dots, x_N)$ for which $x_1, \dots, x_N$ are pairwise distinct.

Proof. Fix $N \in \mathbb{N}$ such that $d + (m-1)N < mN$. Recall from [31, Chapter 4] that every definable set $X \subseteq \mathbb{R}^k$ for $k \in \mathbb{N}$ has a well defined *dimension* $\dim(X) \in \{-\infty, 0, 1, \dots, k\}$ and that $\dim(X) = k$ if and only if X has non-empty interior if and only if X has non-zero Lebesgue measure [6, Theorem 2.5]. Note that the set

$$X \coloneqq \{(\theta, x) \in \mathbb{R}^d \times \mathbb{R}^m \mid f(\theta, x) = 0\} \subseteq \mathbb{R}^d \times \mathbb{R}^m$$

as well as its slices

$$X_\theta \coloneqq \{x \in \mathbb{R}^m \mid f(\theta, x) = 0\} \subseteq \mathbb{R}^m$$

for all $\theta \in \mathbb{R}^d$ are definable. Since each of the functions $f(\theta, -)$ for $\theta \in \mathbb{R}^d$ is analytic and not identically zero, the identity theorem implies that each X_θ for $\theta \in \mathbb{R}^d$ has empty interior. In particular, we have $\dim(X_\theta) \leq m - 1$ for each $\theta \in \mathbb{R}^d$. Note moreover that we have

$$(\mathbb{R}^m)^N \setminus \mathcal{D}_N = \bigcup_{\theta \in \mathbb{R}^d} X_\theta^N \subseteq (\mathbb{R}^m)^N. \tag{11}$$

The combination of Proposition 1.3 (iii), Proposition 1.5 and Corollary 1.6 (iii) in [31, Chapter 4] applied to (11) thus imply the inequality

$$\dim((\mathbb{R}^m)^N \setminus \mathcal{D}_N) \leq d + (m-1)N < mN = \dim((\mathbb{R}^m)^N).$$

Using Theorem 1.8 of [31, Chapter 4], we even get $\dim\left(\overline{(\mathbb{R}^m)^N \setminus \mathcal{D}_N}\right) = \dim((\mathbb{R}^m)^N \setminus \mathcal{D}_N) < \dim((\mathbb{R}^m)^N)$. In particular, $\overline{(\mathbb{R}^m)^N \setminus \mathcal{D}_N} \subseteq (\mathbb{R}^m)^N$ is a null set with empty interior. In other words: $\mathcal{D}_N^\circ \subseteq (\mathbb{R}^n)^N$ is conull and dense.

For the second part of the lemma, assume moreover that $m = 1$. Then each of the sets X_θ for $\theta \in \mathbb{R}^d$ is finite. The uniform finiteness theorem [15, Theorem 4.9] implies that the sets $(X_\theta)_{\theta \in \mathbb{R}^d}$ have uniformly finite cardinality, i.e. $\sup_{\theta \in \mathbb{R}^d} |X_\theta| < \infty$. The lemma then follows by picking $N = \sup_{\theta \in \mathbb{R}^d} |X_\theta| + 1$. □

Proof of Theorem 3.4. By Lemma C.2 and Lemma C.3, for each $\theta \in \mathbb{R}^{d(\mathcal{A})}$, the analytic definable function

$$\mathbb{R}^{d_0} \to \mathbb{R}, \quad x \mapsto \mathcal{N}_\theta(x) - f(x) \tag{12}$$

is not identically zero. If μ is as in item (i) of Theorem 3.4, the theorem now follows from an application of Lemma C.4. Note that the identity theorem implies that (12) is non-zero outside of a null set. Since ℓ is a loss function, the function

$$\mathbb{R}^{d_0} \to \mathbb{R}, \quad x \mapsto \ell(\mathcal{N}_\theta(x), f(x))$$

is non-zero outside of a null set as well. This proves the theorem in the case that μ is as in item (ii) of Theorem 3.4. If μ is a convex combination of the two, the proof follows from the combination of the two arguments. $\square$

D Proof of Theorem 3.5

This section is devoted to the proof of Theorem 3.5 which follows classical proofs of the unviersal approximation theorem [16, 37, 44, 64]. For the particular case of $\tanh$ activations, the proof can be found in [21]. We do not claim originality and include the proof mainly for convenience of the reader. Before going into details, we outline the main ideas of the proof below.

Idea of proof

The proof starts from the special case of fitting linear polynomials in one variable with one hidden layer and then successively builds up generality (see Theorem D.4 and Figure 4). The starting point is that if ψ is an analytic activation function and $x_0 \in \mathbb{R}$, then one can approximate a linear function with slope $\psi'(x_0)$ by neural networks with one hidden neuron using the formula $x \mapsto n\psi(x/n + x_0)$. Similarly, one can inductively approximate the k-th Taylor polynomial of ψ using the formula $x \mapsto n^k\psi(x/n + x_0)$. By adding hidden neurons and compensating the lower order terms of the Taylor polynomial with previous approximations, one can thus approximate polynomials of degree k using neural networks with one hidden layer and k neurons. In order upgrade this to the multivariate case, we employ a linear algebra argument (see Lemma D.1) that allows us to express arbitrary multivariant polynomials as linear combinations of polynomials of the form $p(x_1, \dots, x_n) = (a_1x_1 + \dots + a_nx_n)^k$. To extend our result to multiple hidden layers, we let the other layers approximate the identity and apply the Arzela-Ascoli theorem (see Lemma D.3) to achieve compatibility of the various approximations. Along the way we carefully keep track of the required number of hidden neurons.

Lemma D.1. *Fix integers $n \geq 1$ and $m \geq 0$. Denote by $\mathbb{R}[X_0, \dots, X_m]_n \subseteq \mathbb{R}[X_0, \dots, X_m]$ the real vector space of homogeneous polynomials of degree n in $m+1$ variables. Then, we have $\dim \mathbb{R}[X_0, \dots, X_m]_n = \binom{n+m}{m}$ and moreover*

$$\mathbb{R}[X_0, \dots, X_m]_n = \operatorname{span}\{(X_0 + a_1X_1 + \dots + a_mX_m)^n \mid a_1, \dots, a_m \in \mathbb{R}\}.$$

To **avoid confusion about the dimensions**, we emphasize that this lemma consideres polynomials in $m+1$ variables and not in m variables.

Proof of Lemma D.1. We define the following index set.

$$\mathcal{S}(n,m) := \{\lambda = (\lambda_1, \dots, \lambda_m) \in \mathbb{N}^m \mid 0 \leq \lambda_i \leq n, \lambda_1 + \dots + \lambda_m \leq n\} \subseteq \{0, \dots, n\}^m$$

For $\lambda \in \mathcal{S}(n,m)$, we furthermore write $\lambda_0 = n - \sum_{i=1}^m \lambda_i$. A basis of $\mathbb{R}[X_0, \dots, X_m]_n$ is given by

$$\left(\binom{n}{\lambda_0, \dots, \lambda_m} X_0^{\lambda_0} \cdots X_m^{\lambda_m}\right)_{\lambda \in \mathcal{S}(n,m)}, \tag{13}$$

where $\binom{n}{\lambda_0, \dots, \lambda_m} = \frac{n!}{\lambda_0! \cdots \lambda_m!}$ denotes the multinomial coefficient. It is easy to see from a stars and bars argument that $\dim \mathbb{R}[X_0, \dots, X_m]_n = |\mathcal{S}(n,m)| = \binom{n+m}{m}$.

To prove the second claim of the lemma, we define a linear map $P\colon \mathbb{R}[X_0, \dots, X_m]_n \to \mathbb{R}[X_0, \dots, X_m]_n$ on basis elements by the formula

$$\begin{aligned} P\left(\binom{n}{\lambda_0, \dots, \lambda_m} X_0^{\lambda_0} \cdots X_m^{\lambda_m}\right) &:= (X_0 + \lambda_1X_1 + \ldots + \lambda_mX^m)^n \\ &= \sum_{\mu \in \mathcal{S}(n,m)} \binom{n}{\mu_0, \dots, \mu_m} \lambda_1^{\mu_1} \cdots \lambda_m^{\mu_m} X_0^{\mu_0} \cdots X_m^{\mu_m}. \end{aligned}$$

We introduce a relation $\prec$ on $\mathcal{S}(n,m)$ by defining that $\lambda \prec \mu$ if and only if $\lambda_r < \mu_r$ for some $r \in \{1, \dots, m\}$ and $\lambda_i = \mu_i$ for all $i \in \{1, \dots, m\} \setminus \{r\}$. In this case, we write $\Delta(\mu, \lambda) = \mu_r - \lambda_r$.

Using the basis (13) and the generalized Vandermonde determinant formula [13], we can calculate the determinant of P as

$$\det(P) = \det\Big((\lambda_1^{\mu_1} \cdots \lambda_m^{\mu_m})_{\lambda,\mu \in \mathcal{S}(n,m)} \Big) = \prod_{\substack{\lambda,\mu \in \mathcal{S}(n,m) \\ \lambda \prec \mu}} \Delta(\mu, \lambda) \neq 0.$$

In particular, P is surjective and we have

$$\mathbb{R}[X_0, \ldots, X_m]_n \subseteq P(\mathbb{R}[X_0, \ldots, X_m]_n) \subseteq \operatorname{span}\{(X_0 + a_1 X_1 + \cdots + a_m X_m)^n \mid a_1, \ldots, a_m \in \mathbb{R}\}.$$

□

Definition D.2 (Uniform Norm)**.** For a subset $X \subseteq \mathbb{R}^n$ and a function $f \colon \mathbb{R}^n \to \mathbb{R}$, we write

$$\|f\|_X := \sup_{x \in X} |f(x)|.$$

For a sequence $(f_j)_{j \in \mathbb{N}}$ of functions $\mathbb{R}^n \to \mathbb{R}$, we say that $(f_j)_{j \in \mathbb{N}}$ *converges to f uniformly on compact sets*, if for every compact set $K \subseteq \mathbb{R}^n$, we have

$$\|f_j - f\|_K \xrightarrow{j \to \infty} 0.$$

For approximating the identity with redundant hidden layers, we will also need the following elementary application of the Arzelà–Ascoli theorem.

Lemma D.3. *Let $n, m, k \in \mathbb{N}$ and let $f_j, f_\infty \colon \mathbb{R}^m \to \mathbb{R}^k$ and $g_j, g_\infty \colon \mathbb{R}^n \to \mathbb{R}^m$ for all $j \in \mathbb{N}$ be continuous functions such that $f_j \xrightarrow{j \to \infty} f_\infty$ and $g_j \xrightarrow{j \to \infty} g_\infty$ uniformly on compact sets. Then we have $f_j \circ g_j \xrightarrow{j \to \infty} f_\infty \circ g_\infty$ uniformly on compact sets.*

Proof. Fix a compact subset $K \subseteq \mathbb{R}^n$ and $\varepsilon > 0$. By the Arzelà–Ascoli theorem, there is an $R > 0$ satisfying

$$\bigcup_{j \in \mathbb{N} \cup \{\infty\}} g_j(K) \subseteq B_R(0).$$

Again by the Arzelà–Ascoli theorem, there is $\delta > 0$ such that for all $j \in \mathbb{N} \cup \{\infty\}$ and $x, y \in B_R(0)$ with $|x - y| < \delta$, we have $|f_j(x) - f_j(y)| < \frac{\varepsilon}{2}$. Now choose a large enough $N \in \mathbb{N}$ such that for all $j \in \mathbb{N}$ with $j \geq N$, we have

$$\|f_j - f_\infty\|_{B_R(0)} < \frac{\varepsilon}{2}$$

and

$$\|g_j - g_\infty\|_K < \delta.$$

Then we have

$$|f_j(g_j(x)) - f_\infty(g_\infty(x))| \leq |f_j(g_j(x)) - f_j(g_\infty(x))| + |f_j(g_\infty(x)) - f_\infty(g_\infty(x))| < \varepsilon$$

for all $x \in K$ and $j \in \mathbb{N}$ with $j \geq N$. □

Theorem D.4. *Let $n \in \mathbb{N}$ and $\psi \colon \mathbb{R} \to \mathbb{R}$ be a C^n-function with $\psi^{(n)} \not\equiv 0$ and let $p \colon \mathbb{R}^m \to \mathbb{R}^r$ be a polynomial of degree at most n. Let $\mathcal{A} = (d_0, \ldots, d_k, \psi)$ be a neural network architecture with $d_0 = m$ input neurons, $d_k = r$ output neurons and activation function ψ. Suppose that $\mathcal{A}$ is sufficiently big in the sense that*

(i) there is a hidden layer $0 < i < k$ of size $d_i \geq r\left(\binom{n+m}{m} - m\right)$,

(ii) all previous hidden layers have size $\min(d_0, \ldots, d_{i-1}) \geq d_0$,

(iii) all subsequent hidden layers have size $\min(d_{i+1}, \ldots, d_k) \geq d_k$.

Then p can be approximated uniformly on compact sets by realization functions $\mathcal{N}_\theta^{\mathcal{A}} \colon \mathbb{R}^m \to \mathbb{R}^r$ for $\theta \in \mathbb{R}^d$.

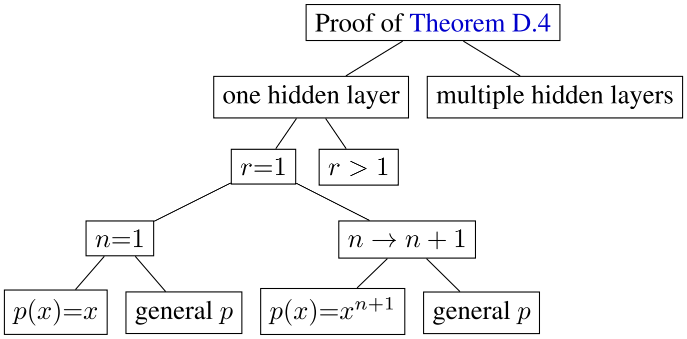


Figure 4: Structure of the special cases in the proof of Theorem D.4.

Proof. We prove the theorem by considering several successively more general cases. The structure of the proof is visualized in Figure 4.

Throughout the proof, we denote for $j \in \mathbb{N}$ by

$$\mathfrak{R}^{\psi}_{m,j,r} \coloneqq \{\mathcal{N}^{(m,j,r,\psi)}_{\theta} \mid \theta \in \mathbb{R}^{(m+1)j+(j+1)r}\} \subseteq C^n(\mathbb{R}^m, \mathbb{R}^r)$$

the set of all realization functions of shallow neural networks with m input neurons, j hidden neurons, r output neurons and activation ψ. Note that

$$\mathfrak{R}^{\psi}_{m,i,r} + \mathfrak{R}^{\psi}_{m,j,r} = \mathfrak{R}^{\psi}_{m,i+j,r} \tag{14}$$

for all $i, j \in \mathbb{N}$. We moreover denote by $\overline{\mathfrak{R}}^{\psi}_{m,j,r} \subseteq C(\mathbb{R}^m, \mathbb{R}^r)$ the set of all functions $f \colon \mathbb{R}^m \to \mathbb{R}^r$, which can be approximated uniformly on compact subsets by elements of $\mathfrak{R}^{\psi}_{m,j,r}$. For $a, x \in \mathbb{R}^m$, we denote by $\langle x, y \rangle = \sum_{i=1}^{m} a_i x_i$ their inner product.

Case 1: We have $r = 1$ and $\mathcal{A} = (m, \binom{n+m}{m} - m, 1, \psi)$.

We prove Case 1 by induction on n.

We start with the case $n = 1$. We first consider the case that p is equal to the identity function $\mathrm{id}_{\mathbb{R}} \colon \mathbb{R} \to \mathbb{R}, \quad \mathrm{id}_{\mathbb{R}}(x) = x$. By the assumptions on ψ, there exists $x_0 \in \mathbb{R}$ with $\alpha \coloneqq \psi'(x_0) \neq 0$. Observe that the functions $(f_j)_{j \in \mathbb{N}} \subseteq \mathfrak{R}^{\psi}_{1,1,1}$ defined by

$$f_j(x) = \frac{j}{\alpha}(\psi(x/j + x_0) - \psi(x_0)), \quad x \in \mathbb{R}$$

converge along $j \to \infty$ to $\mathrm{id}_{\mathbb{R}}$ uniformly on every compact subset $I \subseteq \mathbb{R}$. Indeed, using continuity of ψ', we have for every $j \in \mathbb{N}$ that

$$\begin{aligned}
\sup_{x \in I} |f_j(x) - x| &= \sup_{x \in I} \left| \frac{j}{\alpha} \int_{x_0}^{x_0 + \frac{x}{j}} \psi'(t) dt - x \right| \\
&= \sup_{x \in I} \left| \int_{x_0}^{x_0 + \frac{x}{j}} \left(\frac{j}{\alpha} \psi'(t) - j \right) dt \right| \\
&\leq \frac{j}{|\alpha|} \sup_{x \in I} \int_{x_0}^{x_0 + \frac{x}{j}} |\psi'(t) - \alpha| \, dt \\
&\leq \underbrace{\frac{1}{|\alpha|} \sup_{x \in I} |x|}_{< \infty} \underbrace{\sup_{t \in [x_0, x_0 + \frac{x}{j}]} |\psi'(t) - \alpha|}_{\to 0} \\
&\xrightarrow{j \to \infty} 0.
\end{aligned}$$

This proves that $\mathrm{id}_{\mathbb{R}} \in \overline{\mathfrak{R}}^{\psi}_{1,1,1}$.

Now let

$$p\colon \mathbb{R}^m \to \mathbb{R}, \quad p(x) = \langle a, x\rangle + a_0 = a_1x_1 + \cdots + a_mx_m + a_0$$

be an arbitrary polynomial of degree at most one with $a \in \mathbb{R}^m$ and $a_0 \in \mathbb{R}$. Choose a sequence $(f_j)_{j\in\mathbb{N}} \subseteq \mathfrak{R}^{\psi}_{1,1,1}$ converging to $\mathrm{id}_{\mathbb{R}}$ uniformly on compact subsets. Since p is continuous and $f_j \xrightarrow{j\to\infty} \mathrm{id}_{\mathbb{R}}$ uniformly on compact sets, the sequence $(\tilde{f}_j)_{j\in\mathbb{N}} \subseteq \mathfrak{R}^{\psi}_{m,1,1}$ defined by

$$\tilde{f}_j(x) \coloneqq f_j(\langle a, x\rangle + a_0)$$

converges to p uniformly on compact sets. This proves that $p \in \overline{\mathfrak{R}}^{\psi}_{m,1,1}$. Noting that $\binom{1+m}{m} - m = 1$, this concludes the case $n = 1$.

We now prove Case 1 for $n + 1$, assuming that we have proved Case 1 for n. We first consider the case that p is given by the $(n + 1)$-th power function $\mathrm{id}^{n+1}_{\mathbb{R}}\colon \mathbb{R} \to \mathbb{R}, \quad \mathrm{id}^{n+1}_{\mathbb{R}}(x) = x^{n+1}$. By the assumptions on ψ, there exists $x_0 \in \mathbb{R}$ with

$$\alpha \coloneqq \frac{\psi^{(n+1)}(x_0)}{(n+1)!} \neq 0. \tag{15}$$

We denote by

$$T_n\psi\colon \mathbb{R} \to \mathbb{R}, \quad T_n\psi(x) \coloneqq \sum_{j=0}^{n} \frac{\psi^{(j)}(x_0)}{j!}(x - x_0)^j \tag{16}$$

the Taylor approximation of ψ at x_0 of order n. We define functions $(g_j)_{j\in\mathbb{N}} \subseteq C^{n+1}(\mathbb{R}, \mathbb{R})$ by

$$g_j(x) = \frac{j^{n+1}}{\alpha}(\psi(x_0 + x/j) - T_n\psi(x_0 + x/j)), \quad x \in \mathbb{R} \tag{17}$$

and claim that

$$\sup_{x\in I} |g_j(x) - x^{n+1}| \xrightarrow{j\to\infty} 0 \tag{18}$$

for every compact subset $I \subseteq \mathbb{R}$.

Indeed, by the Lagrange form of the remainder in Taylor's formula, for every $j \in \mathbb{N}$ there is a $\xi_j \in [x_0, x_0 + x/j]$ satisfying

$$\begin{aligned}\sup_{x\in I} |g_j(x) - x^{n+1}| &= \sup_{x\in I} \left| \frac{j^{n+1}}{\alpha} \frac{\psi^{(n+1)}(\xi_j)}{(n+1)!} \left(\frac{x}{j}\right)^{n+1} - x^{n+1} \right| \\ &= \sup_{x\in I} \left| \left(\frac{\psi^{(n+1)}(\xi_j)}{\psi^{(n+1)}(x_0)} - 1 \right) x^{n+1} \right| \\ &\xrightarrow{j\to\infty} 0,\end{aligned}$$

where at the last step we have used continuity of $\psi^{(n+1)}$ and compactness of I.

Now fix an arbitrary polynomial $p\colon \mathbb{R}^m \to \mathbb{R}$ of degree at most $n + 1$. We write $p = p_0 + p_1$ where p_0 is a homogeneous polynomial of degree $n + 1$ and p_1 is a polynomial of degree at most n. By Lemma D.1 (applied to $n \leftarrow n + 1$ and $m \leftarrow m - 1$), we can find $a^{(i)} \in \mathbb{R}^m$ for $i = 1, \ldots, \binom{n+m}{m-1}$ such that

$$p_0(x) = \sum_{i=1}^{\binom{n+m}{m-1}} (\langle a^{(i)}, x\rangle)^{n+1}, \quad \forall x \in \mathbb{R}^m. \tag{19}$$

Fix an increasing sequence $(K_j)_{j\in\mathbb{N}}$ of compact subsets of $\mathbb{R}^m$ such that $\mathbb{R}^m = \bigcup_{j\in\mathbb{N}} K_j$. Since ψ satisfies the assumptions of the theorem for $n + 1$, then it does for n. In particular, for every $j \in \mathbb{N}$, we can find a function $q_j \in \mathfrak{R}^{\psi}_{m,\binom{n+m}{m}-m,1}$ such that

$$\sup_{x\in K_j} \left\| q_j(x) - p_1(x) + \sum_{i=1}^{\binom{n+m}{m-1}} \frac{j^{n+1}}{\alpha} T_n\psi(\langle a^{(i)}, x\rangle) \right\| < \frac{1}{j}, \tag{20}$$

where $T^n\psi$ is defined in (16).

We now define a sequence of functions $(f_j)_{j\in\mathbb{N}} \subseteq C^{n+1}(\mathbb{R}^m, \mathbb{R})$ by

$$f_j(x) = q_j(x) + \sum_{i=1}^{\binom{n+m}{m-1}} \frac{j^{n+1}}{\alpha}\psi\left(x_0 + \frac{1}{j}(\langle a^{(i)}, x\rangle)\right),$$

where $\alpha, x_0 \in \mathbb{R}$ are defined as in (15).

Note that by definition, the first summand of f_j defines an element of $\mathfrak{R}^{\psi}_{m,\binom{n+m}{m}-m,1}$ while the second summand defines an element of $\mathfrak{R}^{\psi}_{m,\binom{n+m}{m-1},1}$. Thus, by (14), f_j defines an element of $\mathfrak{R}^{\psi}_{m,\binom{n+1+m}{m}-m,1}$.

Now let $K \subseteq \mathbb{R}^m$ be a compact subset and $j \in \mathbb{N}$. Using (17) and (19) at the second step, and the combination of (18) and (20) at the last step, we get

$$\begin{aligned}
&\sup_{x\in K} |f_j(x) - p(x)| \\
&= \sup_{x\in K} \left| q_j(x) + \sum_{i=1}^{\binom{n+m}{m-1}} \frac{j^{n+1}}{\alpha}\psi\left(x_0 + \frac{1}{j}(\langle a^{(i)}, x\rangle)\right) - p_0(x) - p_1(x)\right| \\
&= \sup_{x\in K} \left| q_j(x) - p_1(x) + \sum_{i=1}^{\binom{n+m}{m-1}} \frac{j^{n+1}}{\alpha} T^n\psi(\langle a^{(i)}, x\rangle) + \sum_{i=1}^{\binom{n+m}{m-1}} \left(g_j(\langle a^{(i)}, x\rangle) - (\langle a^{(i)}, x\rangle)^{n+1}\right)\right| \\
&\leq \sup_{x\in K} \left| q_j(x) - p_1(x) + \sum_{i=1}^{\binom{n+m}{m-1}} \frac{j^{n+1}}{\alpha} T^n\psi(\langle a^{(i)}, x\rangle)\right| + \sup_{x\in K} \sum_{i=1}^{\binom{n+m}{m-1}} \left|g_j(\langle a^{(i)}, x\rangle) - (\langle a^{(i)}, x\rangle)^{n+1}\right| \\
&\xrightarrow{j\to\infty} 0.
\end{aligned}$$

This proves that $p \in \overline{\mathfrak{R}}^{\psi}_{m,\binom{n+1-m}{m}-m,1}$ and finishes the proof of Case 1.

Case 2: We have $\mathcal{A} = \left(m, r\left(\binom{n+m}{m} - m\right), r, \psi\right)$. This case follows directly from Case 1 by applying Case 1 to each of the components $p_1, \dots, p_r$ of $p = (p_1, \dots, p_r)\colon \mathbb{R}^m \to \mathbb{R}^r$ and stacking the neural network architectures used in the approximation on top of each other.

Case 3: We have a general architecture $\mathcal{A} = (m, d_1, \dots, d_{k-1}, r, \psi)$ as in the statement of the theorem.

We fix $0 < i < k$ such that

(i) $d_i \geq r\left(\binom{n+m}{m} - m\right)$,

(ii) $\min(d_0, \dots, d_{i-1}) \geq d_0$,

(iii) $\min(d_{i+1}, \dots, d_k) \geq d_k$.

By Case 2, there is a sequence $(f_j^{(i)})_{j\in\mathbb{N}} \subseteq \mathfrak{R}_{m,r\left(\binom{n+m}{m}-m\right),r}$ converging to p uniformly on compact subsets of $\mathbb{R}^m$. By a componentwise application of Case 1, for any $0 \leq l < i$, there is a sequence $(f_j^{(l)})_{j\in\mathbb{N}} \subseteq \mathfrak{R}^{\psi}_{m,m,m}$ converging to $\mathrm{id}_{\mathbb{R}^m}$ uniformly on compact sets. Similarly, for any $i < l \leq k$, there is a sequence $(f_j^{(l)})_{j\in\mathbb{N}} \subseteq \mathfrak{R}^{\psi}_{r,r,r}$ converging to $\mathrm{id}_{\mathbb{R}^r}$ uniformly on compact sets. By Lemma D.3, we have

$$f_j^{(k)} \circ \dots \circ f_j^{(0)} \xrightarrow{j\to\infty} p$$

uniformly on compact subsets of $\mathbb{R}^m$. Moreover, by the assumptions on $d_0, \dots, d_k$ we have

$$f_j^{(k)} \circ \dots \circ f_j^{(0)} \in \mathfrak{R}^{\psi}_{r,r,r} \circ \dots \circ \mathfrak{R}^{\psi}_{m,r\left(\binom{n+m}{m}-m\right),r} \circ \dots \circ \mathfrak{R}^{\psi}_{m,m,m} \subseteq \{\mathcal{N}^{\mathcal{A}}_{\theta} \mid \theta \in \mathbb{R}^{d(\mathcal{A})}\},$$

see Figure 5 for an illustration. This finishes the proof of the theorem. □

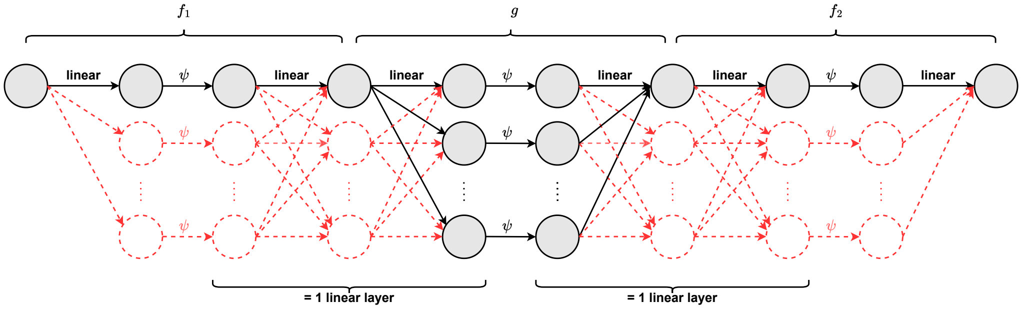


Figure 5: An illustration how a neural network with 3 hidden layers can realize functions of the form $f_2 \circ g \circ f_1$ where g is the realization of a shallow neural network with n neurons and f_i are realizations of shallow neural networks with on hidden neuron. Dotted lines correspond to weights that are set to zero.

Proof of Theorem 3.5. Theorem 3.5 follows immediately from Theorem D.4 and the fact that μ is a probability measure since uniform convergence on compact sets implies L^1-convergence with respect to probability measures. □

Remark D.5. The bounds on the required number of neurons in Theorem 3.5 are not sharp. In many cases, one can approximate polynomials with far fewer neurons. This can be done by following the steps in the proof of Theorem D.4 in detail and depends on several factors such as

(i) the number of nonzero homogeneous parts of the polynomial,

(ii) how many nonzero coefficients are needed to express the homogenous polynomials in the basis from Lemma C.4,

(iii) the zeros of the Taylor coefficients of the activation function.

Moreover, the number of neurons can be reduced drastically if one allows for deep neural networks and rewrites the polynomial as a composition of easier polynomials. The polynomial $p(x, y) = x^7y^9$ for example can be rewritten as the composition of the functions $(x, y) \mapsto (x^7, y^9)$ and $(x, y) \mapsto xy$ and can therefore be approximated by realization functions of the architecture $\mathcal{A} = (2, 16, 4, 1, \psi)$ for any SAD activation function ψ.

E Experimental Details

In this section, we detail the experimental setup. All experiments were conducted on a single Tesla V100 GPU, with each training run requiring between 10 and 20 minutes. Overall, the computational demands of our study remain modest.

Polynomial target function

We train our neural networks to approxomate the polynomial target function $p\colon [a, b]^{d_0} \to \mathbb{R}$ where $[a, b]^{d_0} = [-1, 1]^{d_0}$. The target functions differ based on the input dimension d_0:

(i) for $d_0 = 1$,
$$p(x_1) = x_1^{10} - 2x_1^8 + 2x_1^5 + 3x_1^3 - 2x_1^2 + 5,$$

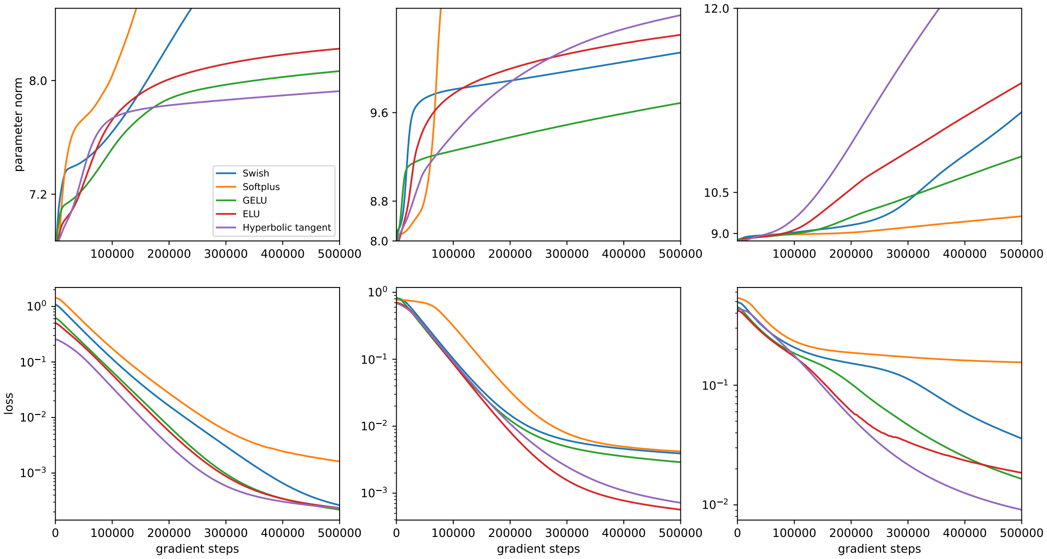
(ii) for $d_0 = 2$,
$$p(x_1, x_2) = x_2^5 - x_1^3x_2^2 - 4x_1^2x_2 + 3x_1^3 - x_2^2 + x_1 + 2,$$

and

(iii) for $d_0 = 4$,

$$p(x_1, x_2, x_3, x_4) = x_1^6 x_4^5 + x_2^6 - x_1^3 x_2^2 x_3 + x_4^2 - 4x_3^4 x_2^4 + 3x_4^3 x_2^3 - x_3^2 x_1 + x_3 + 3.$$

In the one-dimensional case, the neural network architecture consists of three hidden layers with widths $(d_1, d_2, d_3) = (10, 20, 10)$. For the higher-dimensional settings we increase the dimensions to $(d_1, d_2, d_3) = (20, 40, 20)$.[4] For gradient descent, we use the full training dataset of 10,000 randomly sampled points at each step. For Adam, batches of 100 samples are drawn from the same dataset. We use a learning rate of 0.001 for gradient descent and 0.005 for Adam. Note that the plotted loss corresponds to the training loss. Our theoretical results, indeed, show that the norm of the neural network parameters diverges while the training loss converges to zero. To reduce noise, we plot the exponential moving average of the training loss with smoothing factor $\alpha = 0.95$, averaged over 20 independent random initializations. We also construct a test set as a uniform grid over the domain. The test loss exhibits a similar qualitative behavior. We include the plots of the test losses in the GitHub repository https://github.com/deeplearningmethods/sad.



(a)

(b)

(c)

Figure 6: Approximation of polynomial target functions using different activation functions and GD algorithm. From left to right: 1-dimensional input case, 2-dimensional input case, 4-dimensional input case. The y-axis is rescaled exponentially in order to illustrate the logarithmic growth of the parameter norms.

Real world learning task

To solve the PDE problems we use the Deep Kolmogorov method. This approach approximates the solution using a neural network, leveraging the Feynman–Kac representation of the PDE solution. Specifically, the neural network is trained to match the expected terminal value of the underlying stochastic differential equation (SDE), estimated via Monte Carlo sampling. The method provides a generalizable solution across input space, enabling instant evaluations of the PDE once the network is trained. This is particularly advantageous in high-dimensional settings, where traditional solvers suffer the curse of dimensionality. We refer to [5] for a comprehensive description of the method.

In the case of the Heat PDE we consider a neural network with architecture $(d_0, d_1, d_2, d_3, d_4) = (10, 50, 50, 50, 1)$ while for the Black-Scholes PDE $(d_0, d_1, d_2, d_3, d_4) = (10, 200, 300, 200, 1)$. For the latter, we set the interest rate $r = 0.05$, the cost

[4] The reader might notice that the architecture in the four dimensional input case does not satisfy the crude bounds of (i) from Theorem 3.5. However, as pointed out in Remark D.5, these are only worst case bounds. In specific examples like these, one can approximate the given polynomials with significantly smaller architectures.

of carry $c = 0.01$, the strike price $K = 100$ and define the volatility vector as $\sigma = (0.1000, 0.1444, 0.1889, 0.2333, 0.2778, 0.3222, 0.3667, 0.4111, 0.4556, 0.5000)$. The loss plots in Figure 3 report the relative mean squared error between the neural network prediction and the reference solutions. For the Heat PDE, the solution is available in closed form, while for the Black–Scholes PDE we use a high-precision Monte Carlo approximation by averaging over 1000 independents rounds, each consisting of 1024 sample paths. Importantly, the reference solutions are not used during training. At each training step, a single Monte Carlo sample is employed as the target solution. We train the neural networks using the Adam optimizer with a learning rate of 0.005.

For the MNIST classification task, we use a neural network with architecture $(d_0, d_1, d_2, d_3) = (784, 256, 256, 10)$ trained with a standard cross-entropy loss. Optimization is performed with the Adam optimizer and a learning rate of 0.001. As in the previous experiments, we report the training loss in the figure. Training is stopped after 30000 steps to prevent overfitting, resulting in a final test accuracy of 97.93%.

In both the MNIST and PDE examples, we smooth the plots by plotting an exponential moving average of the loss with smoothing factor $\alpha = 0.95$, averaged over 5 independent random initializations.