arXiv:1906.07842v1 [cs.LG] 18 Jun 2019

Gradient Dynamics of Shallow Univariate ReLU Networks

Francis Williams[*]
New York University
francis.williams@nyu.edu

Matthew Trager[*]
New York University
matthew.trager@cims.nyu.edu

Claudio Silva
New York University
csilva@nyu.edu

Daniele Panozzo
New York University
panozzo@nyu.edu

Denis Zorin
New York University
dzorin@cs.nyu.edu

Joan Bruna
New York University
bruna@cims.nyu.edu

Abstract

We present a theoretical and empirical study of the gradient dynamics of overparameterized shallow ReLU networks with one-dimensional input, solving least-squares interpolation. We show that the gradient dynamics of such networks are determined by the gradient flow in a non-redundant parameterization of the network function. We examine the principal qualitative features of this gradient flow. In particular, we determine conditions for two learning regimes: *kernel* and *adaptive*, which depend both on the relative magnitude of initialization of weights in different layers and the asymptotic behavior of initialization coefficients in the limit of large network widths. We show that learning in the kernel regime yields smooth interpolants, minimizing curvature, and reduces to *cubic splines* for uniform initializations. Learning in the adaptive regime favors instead *linear splines*, where knots cluster adaptively at the sample points.

1 Introduction

An important open problem in the theoretical study of neural networks is to describe the dynamical behavior of the parameters during training and, in particular, the influence of the dynamics on the generalization error. To make progress on these issues, a number of studies have focused on a tractable class of architectures, namely single hidden-layer neural networks. For a fixed number of neurons, negative results establish that, even with random initialization, gradient descent may be trapped in arbitrarily bad local minima [27, 31], which motivates an asymptotic analysis that studies the optimization and generalization properties of these models as the number of neurons m grows.

Recently, several works [13, 2, 6, 12, 23] explain the success of gradient descent at optimizing the loss in the so-called *over-parameterized* regime, *i.e.*, for the number of neurons significantly higher than the number of training samples. In parallel, another line of work also established global convergence of gradient descent in the infinite-width limit using a seemingly different scaling [8, 26, 22, 29], relying on tools from optimal transport and mean-field theory. In a remarkable effort, [7] captured an essential difference between these two methodologies, stemming from their different scaling as $m \to \infty$: in one case, the neural network model asymptotically behaves as a kernel machine, with a

---

[*]Equal contribution.

Preprint. Under review.

kernel given by the linear approximation around its initialization [19], which in turn implies that as over-parameterization increases, the first-layer parameters stay close to their initial value. In contrast, the mean-field scaling offers a radically different picture, whereby parameters evolve in the limit $m \to \infty$ following a PDE based on a continuity equation.

Although both scaling regimes explain the success of gradient descent optimization on over-parametrized networks, they paint a different picture when it comes to generalization. The generalization properties in the kernel regime borrow from the well established theory of kernel regression in Reproducing Kernel Hilbert Spaces (RKHS), which has been applied to kernels arising from neural networks in [17, 14, 20, 24, 10], and provide a somehow underwhelming answer to the benefit of neural networks compared to kernel methods. However, in practice large neural networks do not exhibit the traits of kernel/lazy learnings, since filter weights significantly deviate from their initialization despite the over-parameterization. Also, empirically, active learning provides better generalization than kernel learning [7], although the theoretical reasons for this are still poorly understood.

In order to progress in answering this important question, we consider a simplified setting, and study wide, single-hidden layer ReLU networks defined with one-dimensional inputs. We show how the kernel and active dynamics define fundamentally different function estimation models. For a fixed number of neurons, the neural network may follow either of these dynamics, depending on the initialization and the learning rates of the layers, and described by a simple condition on the initial weights. Specifically, we show that kernel dynamics corresponds to interpolation with *cubic splines*, whereas active dynamics yields *adaptive linear* splines, where neurons accumulate at the discontinuities and yield piecewise linear approximations.

1.1 Further related work

Our work lies at the intersection between two lines of research: the works, described above, that develop a theory for optimization and generalization for shallow neural networks, and the works that attempt to shed light on these properties on low-dimensional inputs. In the latter category, we mention [4] for their study of the abilities of ReLU networks to approximate low-dimensional manifolds, and [32] for their empirical study of 3D surface reconstruction using precisely the intrinsic bias of SGD in overparametrised ReLU networks. Another remarkable recent work is [11], where the approximation power of deep ReLU networks is studied in the context of univariate functions. Our analysis in the active regime (cf Section 3.1) is closely related to [21], in which the authors establish convergence of gradient descent to piece-wise linear functions under initialisations sufficiently close to zero. We provide an Eulerian perspective using Wasserstein gradient flows that simplifies the analysis, and is consistent with their conclusions. The implicit bias of SGD dynamics appears in several works, such as [30, 15], and, closest to our setup, in [28], where the authors observe a link between gradient dynamics and *linear splines*. They do not however observe the connection with *cubic splines*, although they observe experimentally that the function returned by a network is often smooth and not piecewise linear. Finally, we mention the works that attempt to study the tesselation of ReLU networks on the input space [16].

1.2 Main contributions

The goal of this paper is to describe the qualitative behavior of the dynamics or 1D shallow ReLU networks. Our main contributions can be summarized as follows.

- We investigate the gradient dynamics of shallow 1D ReLU networks using a "canonical" parameterization (Section 3.1). We use machinery from mean field theory to show that the dynamics in this case are are completely determined by the evolution of the residuals. Furthermore, neurons will always converge at certain samples points where the residual is large and of opposite sign compared to neighboring samples. This means that the dynamics in the reduced parameterization biases towards functions that are *piecewise-linear*.
- We observe that the dynamics in full parameters are related to the dynamics in canonical parameters by a change of metric which depends *only* on the "lift" at initialization (*i.e.*, on which full parameters are associated with a given function). In particular, the change of metric is expressible in terms of invariants δ_i associated with each neuron (Proposition 4). When $\delta_i \gg 0$ the dynamics in full parameters (locally) agree with the dynamics in reduced

parameters; when $\delta_i \ll 0$, the dynamics in full parameters (locally) follow *kernel dynamics*, in which only the outer layer weights change.

- We consider the idealized kernel dynamics in the limit of infinite neurons, and we show that the RKHS norm of a function f corresponds to a weighted L2 norm of the second derivative f'', *i.e.*, a form of *linearized curvature*. For appropriate initial distributions of neurons, the solution to kernel learning is a smooth *cubic spline* (Theorem 5). This illustrates the qualitative difference between the "reduced" and "kernel" regimes, which depend on parameter lift at initialization. Arbitrary initializations will locally interpolate between these two regimes, as can be seen using NTK kernel [19].
- Throughout our presentation, we also discuss the effect of a scaling parameter $\alpha(m)$ applied the network function (where m is the number of neurons), which becomes important as the number of neurons tends to infinity. As argued in [7], when $\alpha(m) = o(m)$, the variation of each neuron will asymptotically go to zero (*lazy regime*), so our local analysis translates into a global one.

2 Preliminaries

2.1 Problem Setup

The training of a two-layer ReLU neural network with m scalar inputs, a single scalar output, and least-squares loss solves the following problem:

$$\begin{aligned} &\min_{\boldsymbol{z}} L(\boldsymbol{z}) = \frac{1}{2}\sum_{j=1}^{s} |f_{\boldsymbol{z}}(x_j) - y_j|^2 \\ &\text{where} \quad f_{\boldsymbol{z}}(x) = \frac{1}{\alpha(m)}\sum_{i=1}^{m} c_i[a_i x - b_i]_+, \quad \boldsymbol{z} = (\boldsymbol{a} \in \mathbb{R}^m, \boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{c} \in \mathbb{R}^m). \end{aligned} \tag{1}$$

Here, $S = \{(x_i, y_i) \in \mathbb{R}^2,\ i = 1, \ldots, s\}$ is a given set of s samples, $\boldsymbol{z}$ is a vector of *parameters*, and $\alpha(m)$ is a normalization factor that will be important as we consider the limit $m \to \infty$.

We are interested in the minimisation of (1) performed by gradient descent over the parameters $\boldsymbol{z}$, evolving in a domain $\mathcal{D} = \mathbb{R}^{3m}$:

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \eta \nabla_{\boldsymbol{z}} L(\boldsymbol{z}^{(k)}) \,.$$

As $\eta \to 0$, this scheme may be analysed through its continuous-time counterpart, the gradient flow

$$\boldsymbol{z}(0) = \boldsymbol{z}_0, \qquad \boldsymbol{z}'(t) = -\nabla L(\boldsymbol{z}(t)). \tag{2}$$

While (2) describes the dynamics $\boldsymbol{z}(t)$ in the parameter space $\mathcal{D}$, one is ultimately interested in the learning trajectories of the function $f_{\boldsymbol{z}(t)}$ they represent. Let $\mathcal{F} := \{f : \mathbb{R} \to \mathbb{R}\}$ denote the space of square-integrable scalar functions, and $\varphi : \mathcal{D} \to \mathcal{F}$ the function-valued mapping $\varphi(\boldsymbol{z}) := f_{\boldsymbol{z}}$. Then, by observing that $L(\boldsymbol{z}) = R(\varphi(\boldsymbol{z}))$, with $R(f) = \frac{1}{2}\sum_{j \le s} |f(x_j) - y_j|^2$, the chain rule immediately shows that the dynamics of $g(t) := \varphi(\boldsymbol{z}(t)) = f_{\boldsymbol{z}(t)}$ are described by

$$g(0) = f_{\boldsymbol{z}_0}, \qquad g'(t) = -\nabla\varphi(\boldsymbol{z}(t))^\top \nabla\varphi(\boldsymbol{z}(t)) \nabla R(g(t)) \,. \tag{3}$$

The dynamics in function space are thus controlled by a time-varying *tangent kernel* $K_t := \nabla\varphi(\boldsymbol{z}(t))^\top \nabla\varphi(\boldsymbol{z}(t))$, that under appropriate assumptions on the parametrisation remains nearly constant throughout the optimization [19, 7]. This kernel may be interpreted as a change of metric resulting from a specific parametrisation of the functional space.

A simple, yet important, observation from (3) is that the evolution of the predicted function depends on two essential aspects: the initialization and the parametrisation. In particular, the coefficients of each neuron can be rescaled using a positive scale factor $k > 0$ according to $(a, b, c) \mapsto (ak, bk, c/k)$. Similarly, the normalization factor $\alpha(m)$ can be absorbed into c_i for all i.

In order to study the impact of different choices of parametrisation and initialisation, as well as the asymptotic behavior of (3) as m increases, we introduce the following *canonical parameterization*:

$$\tilde{f}_{\boldsymbol{w}}(x) = \frac{1}{m}\sum_{i=1}^{m} r_i \langle \tilde{x}, d(\theta_i)\rangle_+, \qquad \boldsymbol{w} = (\boldsymbol{r} \in \mathbb{R}^m, \boldsymbol{\theta} \in [0, 2\pi)^m), \qquad \tilde{x} = (x, 1). \tag{4}$$

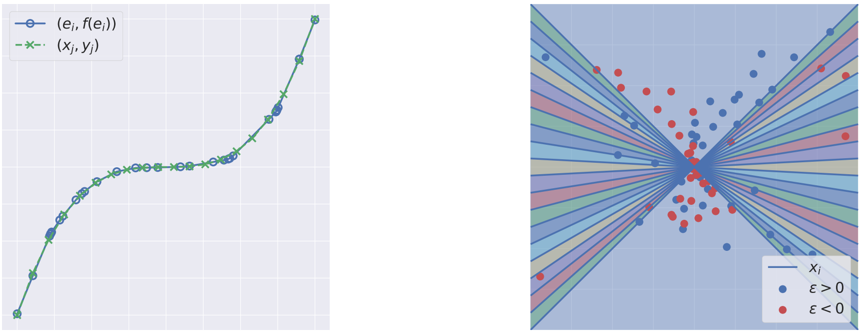


Figure 1: *Left:* Input samples (blue x's) to which we fit a neural network $f_{\boldsymbol{z}}(x)$ using the least squares loss (1). Clearly, $f_{\boldsymbol{z}}(x)$ is piecewise linear with the boundaries between pieces occurring at $(e_i, f_{\boldsymbol{z}}(e_i))_{i=1}^{m}$ (green circles). *Right:* the neurons visualized as (u_i, v_i) in (6). Each particle represents a neuron and the color indicates the sign of ϵ_i. The samples x_j correspond to the lines $ux_j + v = 0$ in this space. These sample lines divide the space into the colored regions which correspond to different activation patterns.

where $d(\theta_i) = (\cos\theta_i, \sin\theta_i) \in S^1$ and we chose $\alpha(m) = m$. As shown in Section 3.1, this choice of normalization, together with an initialisation where r_i are sampled iid from a distribution with $O(1)$ variance, provides a well-defined limit of the dynamics as $m \to \infty$, and provides a basis to compare the other normalization and parametrisation choices.

2.2 Reparametrisation and Normalization

Thanks to the fact that the canonical parametrisation admits a well-defined limit, we can study the effect of specific parametrisations and normalisations by expressing them as changes of variables relative to $\boldsymbol{w} \mapsto \tilde{f}_{\boldsymbol{w}}$. The mapping from the weights $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ to $(\boldsymbol{r}, \boldsymbol{\theta})$ in the canonical parametrisation is given by

$$\boldsymbol{\pi}(a_i, b_i, c_i) = \left(\frac{m}{\alpha(m)} c_i \sqrt{a_i^2 + b_i^2}, \arctan(-b_i/a_i)\right) = (r_i, \theta_i). \tag{5}$$

This map satisfies $\tilde{f}_{\boldsymbol{\pi}(\boldsymbol{z})} = f_{\boldsymbol{z}}$. We also define the loss with respect to this parameterization as $\tilde{L}(\boldsymbol{w}) = L(\boldsymbol{z})$ where $\boldsymbol{w} = \boldsymbol{\pi}(\boldsymbol{z})$. Section 3.2 explains how the Jacobian of $\boldsymbol{\pi}$ affects the dynamics relative to the canonical dynamics. In particular, this illustrates the drastic effects of a choice of normalisation $\alpha(m) = o(m)$.

Note that $f_{\boldsymbol{z}}(x)$ (and thus $\tilde{f}_{\boldsymbol{w}}(x)$) are *continuous piecewise linear functions* in x. The *knots* of of these functions are the points where the operand inside a ReLU activation changes sign: $e_i = \frac{b_i}{a_i}$, $a_i \neq 0$, $i = 1, \ldots, m$. See the left image in Figure 1 for an example of a function $f_{\boldsymbol{z}}$ and its knots.

2.3 Visualizing the Network State

In this paper, we visualize the state of the network in two ways. First, we plot $f_{\boldsymbol{z}}$ to visualize the piecewise-linear function encoded by the network and its knots at given time (Figure 1, Left). We use the alternative parameterization $\tilde{f}_{\boldsymbol{w}}$ to visualize the reduced state of the network in $\mathbb{R}^2$, by plotting a neuron (r_i, θ_i) as a particle with coordinates

$$(u_i, v_i) = (|r_i|\cos(\theta_i), |r_i|\sin(\theta_i)) \tag{6}$$

and coloring each particle according to $\epsilon_i = \operatorname{sign}(r_i)$. Figure 1 (right) shows an example of this visualization. In the standard parameterization, a neuron (a_i, b_i, c_i) coincides with a point x if $\frac{b_i}{a_i} = \frac{v_i}{u_i} = x$. Thus, in the visualization, a sample point x_j corresponds to the line satisfying the equation $ux_j + v = 0$. These sample lines, divide the phase space into *activation regions* where a neuron has a fixed *activation pattern* (the colored regions in the right image of Figure 1).

3 Training Dynamics

Our goal is to solve (1) using the *gradient flow* (2) of the loss $L(\boldsymbol{z})$[1]. We begin in Section 3.1 by investigating the dynamics in the canonical parametrisation:

$$\boldsymbol{w}(0) = \boldsymbol{w}_0, \qquad \boldsymbol{w}'(t) = -\nabla \tilde{L}(\boldsymbol{w}(t)). \tag{7}$$

While the relationship between flows (2) and (7) is nonlinear, we show in Section 3.2 that these are related by a simple change of metric.

3.1 Dynamics in the Canonical Parameters

We now examine the qualitative behavior of the gradient flow using the canonical parameterization (4). In this setting, when (r_i, θ_i) are initialized i.i.d. from some base distribution $\mu_0(r, \theta)$, the function $\tilde{f}_{\boldsymbol{w}}$ is well-defined pointwise as $m \to \infty$ (by the law of large numbers). We optimize the loss over both $\boldsymbol{\theta}$ and $\boldsymbol{r}$ using the reduced gradient flow (7).

Using the initialization $(r_i, \theta_i) \sim \mu$, and following the mean-field formulation of single-hidden layer neural networks of [22, 8, 26], we express the function as an expectation with respect to the probability measure μ over the cylinder $\mathcal{D} = \mathbb{R} \times S^1$:

$$\tilde{f}_{\boldsymbol{w}}(x) = \int_{\mathcal{D}} \varphi(w; x) \mu^{(m)}(\mathrm{d}w) \ , \tag{8}$$

where $\varphi(w; x) := r_i \langle \tilde{x}, d(\theta_i) \rangle_+$ and $\mu^{(m)}(w) = \frac{1}{m} \sum_{i=1}^m \delta_{w_i}(w)$ is the empirical measure determined by the m particles w_i, $i = 1 \ldots m$. The least squares loss in this case becomes

$$\begin{aligned} \tilde{L}(\boldsymbol{w}) &= \frac{1}{2} \| \tilde{f}_{\boldsymbol{w}} - y \|_{\mathcal{X}}^2 && (9) \\ &= C_y - \frac{1}{m} \sum_{i=1}^m \langle \varphi_{w_i}, y \rangle_{\mathcal{X}} + \frac{1}{2m^2} \sum_{i,i'=1}^m \langle \varphi_{w_i}, \varphi_{w_{i'}} \rangle_{\mathcal{X}} \ , && (10) \end{aligned}$$

where $\langle f, g \rangle_{\mathcal{X}} := \sum_{j=1}^s f(x_j) g(x_j)$ is the empirical dot-product. This loss may be interpreted as the Hamiltonian of a system of m-interacting particles, under external field F and interaction kernel K defined respectively by $F(w) := \langle \varphi_w, y \rangle_{\mathcal{X}}$,$K(w, w') := \langle \varphi_w, \varphi_{w'} \rangle_{\mathcal{X}}$. We may also express this Hamiltonian in terms of the empirical measure, by abusing notation

$$\tilde{L}(\mu^{(m)}) = C_y - \int_{\mathcal{D}} F(w) \mu^{(m)}(\mathrm{d}w) + \frac{1}{2} \iint_{\mathcal{D}^2} K(w, w') \mu^{(m)}(\mathrm{d}w) \mu^{(m)}(\mathrm{d}w') \ .$$

A direct calculation shows that the gradient $\nabla_{w_i} \tilde{L}(\boldsymbol{w})$ can be written as

$$\frac{m}{2} \nabla_{w_i} \tilde{L}(\boldsymbol{w}) = \nabla_w V(w_i; \mu^{(m)}) \ ,$$

where V is the potential function $V(w; \mu) := -F(w) + \int_{\mathcal{D}} K(w, w') \mu(\mathrm{d}w')$.

The gradient flow in the space of parameters $\boldsymbol{w}$ can now be interpreted in Eulerian terms as a gradient flow in the space of measures over $\mathcal{D}$, by using the notion of Wasserstein gradient flows [22, 8, 26]. Indeed, particles evolve in $\mathcal{D}$ by "feeling" a velocity field ∇V defined in $\mathcal{D}$. This formalism allows us now to describe the dynamics independently of the number of neurons m, by replacing the empirical measure $\mu^{(m)}$ by any generic probability measure μ in $\mathcal{D}$. The evolution of a measure under a generic time-varying vector field is given by the so-called continuity equation:[2]

$$\partial_t \mu_t = \mathrm{div}(\nabla V \mu_t) \ . \tag{11}$$

The global convergence of this PDE for interaction kernels arising from single-hidden layer neural networks has been established under mild assumptions in [22, 8, 25]. Although the conditions for

---

[1] To be precise, we should replace the gradient $\nabla L(\boldsymbol{z})$ with the *Clarke subdifferential* $\partial L(\boldsymbol{z})$ [9], since $L(\boldsymbol{z})$ is only piecewise smooth. At generic smooth points $\boldsymbol{z}$, the subdifferential coincides with the gradient $\partial L(\boldsymbol{z}(t)) = \{\nabla L(\boldsymbol{z})\}$.

[2] Understood in the weak sense, *i.e.*, $\partial_t \left(\int_{\mathcal{D}} \phi(w) \mu_t(\mathrm{d}w) \right) = -\int \langle \nabla \phi(w), \nabla V(w; \mu_t) \rangle \mu_t(\mathrm{d}w)$, $\forall \phi \in C_c^1(\mathcal{D})$ continuously differentiable and with compact support.

global convergence hold in the mean field limit $m \to \infty$, a propagation-of-chaos argument from statistical mechanics gives Central Limit Theorems for the behavior of finite-particle systems as fluctuations of order $1/\sqrt{m}$ around the mean-field solution; see [26, 25] for further details.

The dynamics in $\mathcal{D}$ are thus described by the velocity field $\nabla V(w; \mu_t)$, which depends on the current state of the system through the measure $\mu_t(w)$, describing the probability of encountering a particle at position w at time t. We emphasize that equation (11) is valid for any measure, including the empirical measure $\mu^{(m)}$, and is therefore an exact model for the dynamics in both the finite-particle and infinite-particle regime. Let us now describe its specific form in the case of the empirical loss given above.

Assume without loss of generality that the data points $x_j \in \mathbb{R}$, $j \leq s$ satisfy $x_j \leq x_{j'}$ whenever $j < j'$. Denote

$$\mathcal{C}_j := \{j'; j' \leq j\} \text{ for } j = 1 \dots s, \qquad \mathcal{C}_{s+j} := \{j'; j' > j\}, \text{ for } j = 1 \dots s-1 .$$

We observe that for each j, two angles $\alpha_j^{\pm} = \arctan(x_j) \pm \pi/2$ partition the circle S^1 into $2s-1$ regions $\mathcal{A}_k$ (visualized as the colored regions in Figure 1), which are in one-to-one correspondence with the sets $\mathcal{C}_k$, in the sense that

$$\theta \in \mathcal{A}_k \Longleftrightarrow \{j; \langle \tilde{x}_j, d(\theta) \rangle \geq 0\} = \mathcal{C}_k .$$

We also denote by $\mathcal{B}_j$ the half-circle where $\langle \tilde{x}_j, \theta \rangle \geq 0$. Let $t(\theta)$ be the tangent vector of S^1 at θ (so $t(\theta) = d(\theta)^{\perp}$) and $w = (r, \theta)$, where we suppose $\theta \in \mathcal{A}_k$. A straightforward calculation (presented in Appendix B) shows that the angular and radial components of the velocity field $\nabla V(w; \mu_t)$ are given by

$$\begin{aligned} \nabla_\theta V(w; \mu_t) &= r \left\langle \sum_{j \in \mathcal{C}_k} \rho_j(t) \tilde{x}_j, t(\theta) \right\rangle , \\ \nabla_r V(w; \mu_t) &= \left\langle \sum_{j \in \mathcal{C}_k} \rho_j(t) \tilde{x}_j, d(\theta) \right\rangle , \end{aligned} \tag{12}$$

where $\rho_j(t) = \int_{\mathbb{R} \times \mathcal{B}_j} r \langle \tilde{x}_j, \theta \rangle \mu_t(\mathrm{d}r, \mathrm{d}\theta) - y_j$ is the residual at point x_j at time t. These expressions show that the dynamics are entirely controlled by the s-dimensional vector of residuals $\boldsymbol{\rho}(t) = (\rho_1(t), \dots \rho_s(t))$, and that the velocity field is *piece-wise linear* on each cylindrical region $\mathbb{R} \times \mathcal{A}_k$ (e.g. Figure 9 in Appendix D). Under the assumptions that ensure global convergence of (11), we have $\lim_{t \to \infty} \tilde{L}(\mu_t) = 0$, and therefore $\|\boldsymbol{\rho}(t)\| \to 0$. The oscillations of $\boldsymbol{\rho}(t)$ as it converges to zero determine the relative orientation of the flow within each region. The exact dynamics for the vector of residuals are given by the following proposition, proved in Appendix B:

Proposition 1. *For each j,*

$$\dot{\rho}_j(t) = -\tilde{x}_j^{\top} \sum_{k; \mathcal{A}_k \subset \mathcal{B}_j} \Sigma_k(t) \left(\sum_{j' \in \mathcal{C}_k} \rho_{j'}(t) \tilde{x}_{j'} \right) , \tag{13}$$

where

$$\Sigma_k(t) = \int_{\mathbb{R} \times \mathcal{A}_k} \left(r^2 t(\theta)\, t(\theta)^{\top} + d(\theta)\, d(\theta)^{\top} \right) \mu_t(\mathrm{d}r, \mathrm{d}\theta)$$

tracks the covariance of the measure along each cylindrical region.

Equation (13) defines a system of ODEs for the residuals $\boldsymbol{\rho}(t)$, but its coefficients are time-varying, and behave roughly as quadratic terms in $\boldsymbol{\rho}(t)$ (since they are second-order moments of the measure whereas the residuals are first-order moments). It may be possible to obtain asymptotic control of the oscillations $\boldsymbol{\rho}(t)$ by applying Duhamel's principle, but this is left for future work.

Now let $w = (r, \theta)$ with θ at a boundary of two regions $\mathcal{A}_k$, $\mathcal{A}_{k+1}$. The velocity field is modified at the transition by

$$\nabla V(w)|_{\mathcal{A}_k} - \nabla V(w)|_{\mathcal{A}_{k+1}} = \rho_{j*}(t) \begin{pmatrix} r \langle \tilde{x}_{j*}, t(\theta) \rangle \\ \langle \tilde{x}_{j*}, \theta \rangle \end{pmatrix} ,$$

where j_* is such that $\langle \tilde{x}_{j*}, d(\theta) \rangle = 0$, since θ is at the boundary of $\mathcal{A}_k$. It follows that the only discontinuity is in the angular direction, of magnitude $|r \rho_{j*}(t)| \|\tilde{x}_{j^*}\|$. In particular, an interesting phenomenon arises when the angular components of $\nabla V(w)|_{\mathcal{A}_k}$ and $\nabla V(w)|_{\mathcal{A}_{k+1}}$ have opposite signs, corresponding to an "attractor" or "repulsor" that attracts/repels particles along the direction given by $\tilde{x}_{j*}$ (See Figure 9 in Appendix D). Writing $s_k = \left\langle \sum_{j\in\mathcal{C}_k} \rho_j(t)\tilde{x}_j, t(\theta)\right\rangle$, we deduce from (12) that this occurs when

$$|s_k| < |\rho_{j*}(t)|\|\tilde{x}_{j^*}\| \text{ and } \mathrm{sign}(s_k) \neq \mathrm{sign}(\rho_{j*}(t)). \tag{14}$$

We expand this condition in the following Lemma.

Lemma 2. *A data point x_k is an attractor/repulsor if and only if*

$$\sum_{i=1}^{k-1} \rho_i\rho_k\langle\tilde{x}_i,\tilde{x}_k\rangle > -\rho_k^2\|\tilde{x}_k\|^2, \textit{ or } \sum_{i=k+1}^{s} \rho_i\rho_k\langle\tilde{x}_i,\tilde{x}_k\rangle > -\rho_k^2\|\tilde{x}_k\|^2. \tag{15}$$

In words, mass will concentrate towards input points where the residual is currently large and of opposite sign from a weighted average of neighboring residuals. This is in stark contrast with the kernel dynamics (Section 3.3), where there is no adaptation to the input data points. We point out that this qualitative behavior has been established in [21] under appropriate initial conditions, sufficiently close to zero, in line with our mean-field analysis.

Regularised Objective in Functional Space: The global convergence of (11) studied in [8] includes the case where the energy functional $\tilde{L} = R(\int \Phi\mu(\mathrm{d}w))$ is augmented with a regulariser $\Gamma(\mu) = \int \gamma(w)\mu(\mathrm{d}w)$ sharing the same degree of homogeneity as Φ. If $\gamma(w) = |r|$ this corresponds to the *Total Variation* norm of the signed Radon measure $\tilde{\mu}(w) = \int \mu(\mathrm{d}r, w)$: $\Gamma(\mu) = \|\tilde{\mu}\|_{\mathrm{TV}}$. The Wasserstein gradient flow on the resulting regularised objective

$$\min_{\tilde{\mu}} \tilde{L}(\tilde{\mu}) + \lambda\|\tilde{\mu}\|_{\mathrm{TV}} \tag{16}$$

thus converges to global minimisers under appropriate initial conditions (which only apply in the infinite width regime $m = \infty$). The regularised dynamics are obtained by replacing $\nabla_r V(w;\mu_t) = \left\langle \sum_{j\in\mathcal{C}_k} \rho_j(t)\tilde{x}_j, d(\theta)\right\rangle$ in (12) by

$$\nabla_r V(w;\mu_t) = \left\langle \sum_{j\in\mathcal{C}_k} \rho_j(t)\tilde{x}_j, d(\theta)\right\rangle + \lambda \mathrm{sign}(r)\ .$$

In the case of scalar inputs, [28] recently characterised the solutions of (16) in functional space, as

$$\min_f L(f) + \lambda\int |f''(x)|\mathrm{d}x\ , \tag{17}$$

for vanishing boundary conditions. In Section 3.3 we will see that the kernel regime corresponds to a very different prior, where the L^1 norm on the second derivatives is replaced by a Hilbert L^2 norm. The distinction between kernel and active regime in terms of L^1 versus L^2 norms was already studied in [3].

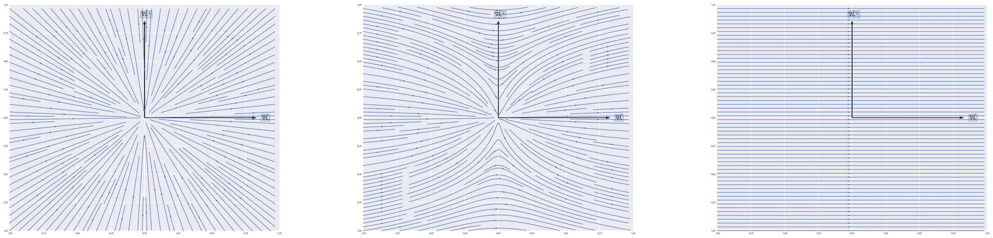
3.2 Dynamics in the Full Parameters

The dynamics of gradient flow (2) are different from the dynamics of the gradient flow (7). For the gradient flow in canonical parameters we have observed adaptive learning behavior under the assumption of iid distribution of parameter initialization. The full set of parameters $\boldsymbol{z} = (\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})$, may exhibit both kernel and adaptive behavior depending on the initialization. To characterize this behavior we rely on the following lemma:

Lemma 3. *If $\boldsymbol{z}(t) = (\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{c}(t))$ is a solution of the gradient flow (2), then the quantities*

$$\boldsymbol{\delta} = (c_i(t)^2 - a_i(t)^2 - b_i(t)^2)_{i=1}^m \tag{18}$$

remain constant for all t. In particular, given a reduced neuron (r_i,θ_i), we can uniquely recover the original neuron parameters (a_i,b_i,c_i) from δ_i computed from the initialization.



$\delta = -100$ $\delta = 0$ $\delta = 100$

Figure 2: The value of δ interpolates between different kinds of local trajectories of neurons. The plots are in the coordinate frame $(\nabla \tilde{L}, \nabla \tilde{L}^{\perp})$. Left: the neurons move radially towards and away from the origin. Middle: the trajectories containing both radial and tangential components. Right: the trajectories are parallel to the gradient $\nabla \tilde{L}$.

Lemma 3 allows us to analyze how canonical parameters evolve under *full* gradient flow in $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$. Overall, the behavior is qualitatively the same, except it is in addition dependent on the relative scale of redundant parameters.

Proposition 4. *Let $\boldsymbol{z}(t)$ be a solution of the gradient flow (2) of $L(\boldsymbol{z})$, and let $\boldsymbol{\delta} = (\delta_i) \in \mathbb{R}^m$ be the vector of invariants (18), which depend only on the initialization $\boldsymbol{z}(0)$. If $\boldsymbol{w}(t) = (\boldsymbol{r}(t), \boldsymbol{\theta}(t))$ is curve of canonical parameters corresponding to $\boldsymbol{z}(t)$, then we have that*

$$\dot{\boldsymbol{w}}_i(t) = \boldsymbol{P}_i \cdot \nabla_{\boldsymbol{w}_i} \tilde{L}(\boldsymbol{w}), \quad i = 1, \ldots, m, \tag{19}$$

where

$$\boldsymbol{P}_i = \begin{bmatrix} \frac{m^2}{\alpha(m)^2}(a_i^2 + b_i^2 + c_i^2) & 0 \\ 0 & \frac{1}{a_i^2 + b_i^2} \end{bmatrix}. \tag{20}$$

With respect to rescaled differentials $d\tau = r d\theta$, corresponding to representing the flow locally in a Cartesian system aligned with the radial direction (pointing away from $\boldsymbol{z} = \boldsymbol{0}$) and its perpendicular, the flow can be written as

$$\begin{bmatrix} dr_i \\ d\tau_i \end{bmatrix} = \begin{bmatrix} \frac{m^2}{\alpha(m)^2}(a_i^2 + b_i^2 + c_i^2) & 0 \\ 0 & c_i^2 \end{bmatrix} \cdot \begin{bmatrix} \nabla_{r_i} \tilde{L}(\boldsymbol{w}) dt \\ \nabla_{\tau_i} \tilde{L}(\boldsymbol{w}) dt \end{bmatrix}, \quad i = 1, \ldots, m, \tag{21}$$

From these equations, one can see that if $c_i^2 \ll a_i^2 + b_i^2$ for all i (*i.e.*, $\delta_i \ll 0$), then radial motion will dominate. In other words, initializing the first layer with significantly larger values than the second leads to kernel learning. On the other hand, if $c_i^2 \gg a_i^2 + b_i^2$, then a solution of the gradient flow (2) will follow the same trajectory as for the reduced gradient flow (7). Also, if $\alpha(m) = o(m)$, the radial component will dominate as m increases.

Figure 2 shows the trajectories corresponding to different values of δ_i for each neuron, with $\alpha(m) = m$.The extreme cases of $\delta = -\infty$ and $\delta = +\infty$ correspond to the "kernel" and "adaptive" regimes, respectively. Note that as δ increases, the neurons cluster at sample points, as explained in our analysis of the reduced dynamics in Section 3.1, and in accordance to [21].

3.3 Kernel Dynamics

We now consider the dynamics in the special case where $\delta \ll 0$, and we consider $m \to \infty$. To obtain the kernel regime in this case, it is sufficient to consider a normalisation $\alpha(m) = o(m)$. In particular, when $\alpha(m) = 1$, as shown in the previous section, the parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ remain mostly fixed and the parameters $\boldsymbol{c}$ change throughout training, corresponding to the so-called random-features (RF) kernel of Rahimi and Recht [24].

In the limit case where $\boldsymbol{a}$ and $\boldsymbol{b}$ are completely fixed to their initial values, if we choose $\boldsymbol{c}$ close to the zero vector, then the least squares problem (1) solved using gradient flow, is equivalent to the minimal-norm constraint problem solution:

$$\begin{aligned} &\text{minimize } \|\boldsymbol{c}\|^2 \\ &\text{subject to } f_{\boldsymbol{z}}(x_i) = y_i, \qquad i = 1, \ldots, s. \end{aligned} \tag{22}$$

Given an initial distribution μ_0 over the domain $\mathcal{D}_a \times \mathcal{D}_b$ of parameters a and b, the random-feature (RF) kernel associated with (22) is given by

$$K_{\mathrm{RF}}(x, x') = \int_{\mathcal{D}_a \times \mathcal{D}_b} [xa - b]_+ \cdot [x'a - b]_+ \mu_0(\mathrm{d}a, \mathrm{d}b) \,. \tag{23}$$

The solution of (22) can now be written in terms of this RF kernel using the representer theorem:

$$\tilde{f}_{\boldsymbol{z}}(x) = \sum_{j=1}^{s} \alpha_j K_{\mathrm{RF}}(x_j, x) \,, \tag{24}$$

where α is a vector of minimal RKHS norm that fulfills the interpolation constraints. Under appropriate assumptions, the solution to (22) is a *cubic spline*, as shown by the following Theorem proved in Appendix A.

Theorem 5. *Assume the measure μ_0 has finite second moment $\sigma^2_{\mu_0} := \mathbb{E}_{(a,b)\sim\mu_0}(a^2 + b^2) < \infty$. Let $\mu_0(a, b) = q(a)\mu_a(b)$ be the decomposition in terms of marginal and conditional, and assume μ_a is bounded for each a. Define $\nu(u) = \int |a|\mu_a(ab)\mathrm{d}q(a)$. Then the solution (22) solves*

$$\min_f \quad \|f\|^2_{\mathrm{RF}} := \int_\Omega \frac{|f''(u)|^2}{\nu(u)} \mathrm{d}u \quad \text{s.t.} \quad f(x_i) = y_i \,, i = 1 \ldots s \,, \tag{25}$$

where $\Omega := supp(\nu)$. Moreover, if μ_0 is such that $\mu_0(a, b) = q(a)\mathbf{1}(b \in I_a)$, where $I_a \subset \mathbb{R}$ is an arbitrary interval, then (22) will be a cubic spline.

Notice that the assumptions on μ_0 to obtain an exact cubic spline kernel impose that if A, B is a random vector distributed according to μ_0, then $B|A$ is uniform over an arbitrary interval I_A that can depend upon A. The proof illustrates that one may generalise the interval I_A by any countable union of intervals. In particular, independent uniform initialization yields cubic splines, but radial distributions, such as A, B being jointly Gaussian, don't. In that case, the kernel is expressed in terms of trigonometric functions [7, 3]. In our setup this becomes

$$\begin{aligned} K^r_{\mathrm{RF}}(x, x') \quad &= \quad \frac{C}{4\pi}\Big\{(\pi - \arctan(x') + \arctan(x))(xx' + 1) + \\ &\quad + \left(\frac{x'}{1 + (x')^2} - \frac{x}{1 + x^2}\right)(xx' - 1) + (x + x')\left(\frac{(x')^2}{1 + (x')^2} - \frac{x^2}{1 + x^2}\right)\Big\} \,, \end{aligned} \tag{26}$$

as verified in Appendix A.

The normalization choice $\alpha(m) = \sqrt{m}$ results in a different kernel, the Neural Tangent Kernel of [19]. In this scaling regime, the linearization of the model becomes

$$\begin{aligned} K_{\mathrm{NTK}}(x, x') \quad &= \quad \nabla_{\boldsymbol{z}} f_{\boldsymbol{z}}(x)^\top \nabla_{\boldsymbol{z}} f_{\boldsymbol{z}}(x') \\ &= \quad \frac{1}{m}\sum_{i=1}^{m} [a_i x - b_i]_+ [a_i x' - b_i]_+ + \frac{xx' + 1}{m} \sum_{i=1}^{m} c_i^2 \mathbf{1}(a_i x - b_i > 0) \cdot \mathbf{1}(a_i x' - b_i > 0) \\ &\overset{m\to\infty}{\to} \quad K_{\mathrm{RF}}(x, x') + (xx' + 1) \cdot \mathbb{E}(c^2) \int \mathbf{1}(ax > b) \cdot \mathbf{1}(ax' > b) \mu_0(\mathrm{d}a, \mathrm{d}b) \,. \end{aligned} \tag{27}$$

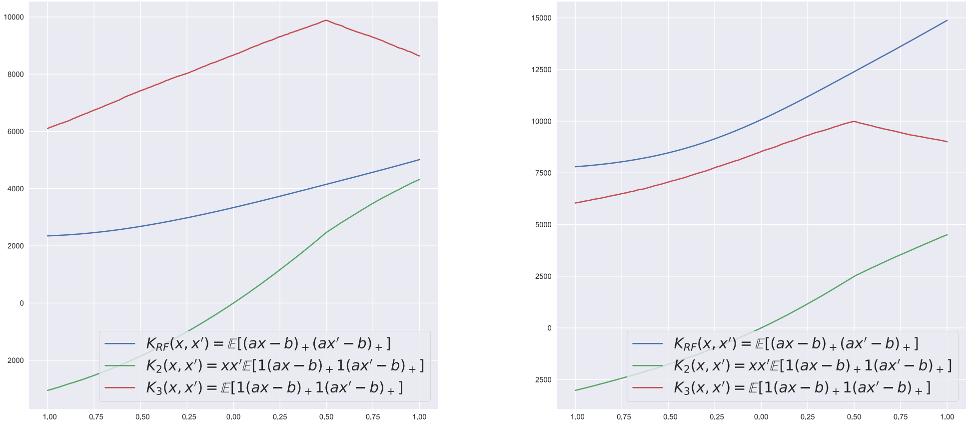
The extra term in K_{NTK} captures lower-order regularity as shown in the following corollary

Corollary 6. *Under the same assumptions as in Theorem 5, we have*

$$\|f\|^2_{\mathrm{NTK}} := \inf\left\{\int_\Omega \left(\frac{|f_1''(u)|^2}{\nu(u)} + \mathbb{E}(c^2)\left(\frac{|f_2'(u)|^2}{\nu(u)} + \frac{|uf_3'(u) - f_3(u)|^2}{u^2\nu(u)}\right)\right)\mathrm{d}u;\ f = f_1 + f_2 + f_3\right\} \,. \tag{28}$$

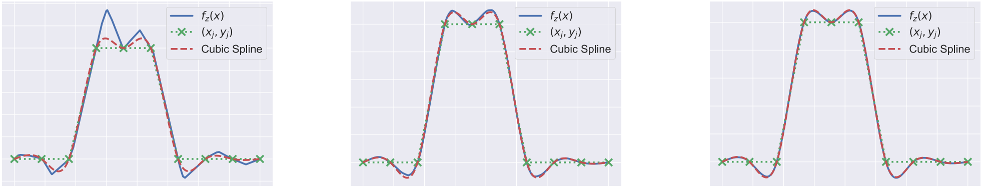
Additionally, if μ_0 is such that $\mu_0(a, b) = q(a)\mathbf{1}(b \in I_a)$, where $I_a \subset \mathbb{R}$ is an arbitrary interval, then K_{NTK} is also piece-wise cubic.

We remark that machine learning packages such as PyTorch use a uniform distribution for linear layer parameter initialization by default. We verify that indeed, solutions to (1) converge to cubic splines as m grows in Figure 4. We also point out that in Kernel Learning, early termination of gradient flow acts as a regularizer favoring smooth, non-interpolatory solutions (see [19]).



Uniform Initialization Gaussian Initialization

Figure 3: 1D slice of the plot of the different terms of the tangent kernel (28) for $x = 0.5$ with a network consisting of $m = 25000$ neurons. The parameters $\boldsymbol{a}$ and are initialized from Uniform and Gaussian distributions with $\frac{1}{\sqrt{m}}$ scaling.



$m = 10^2$ $m = 10^3$ $m = 10^4$

Figure 4: A cubic spline with vanishing second derivative at its endpoints (blue line) is approximated by a neural network ($\delta = -100$) while varying the number m of neurons.

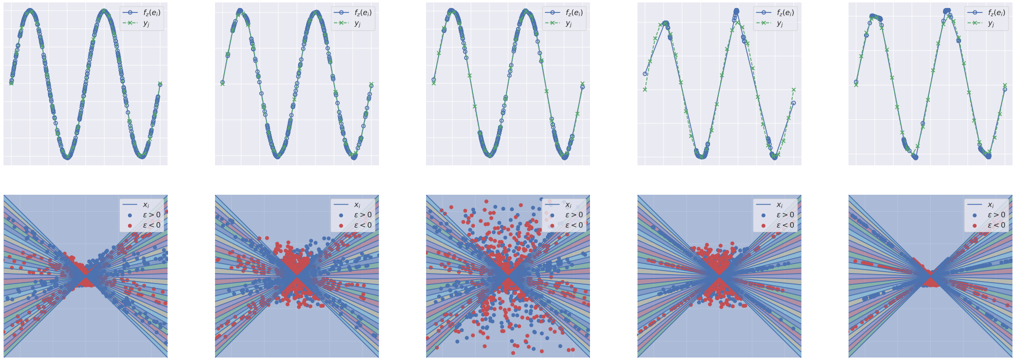
The analysis and comparison of these kernels has recently been addressed in [5, 14] in the general high-dimensional setting, by describing its spectral decay in terms of spherical harmonics. Our results complement them in the particular one-dimensional setting thanks to the explicit functional form of the resulting RKHS norms. Additionally, Savarese et al. [28] study the functional form of the minimisation in the variation norm, leading to a penality of the form $\int |f''(u)| \mathrm{d}u$. We have instead L^2 norms (RKHS) in the kernel regime. The L^2 norms do not provide any adaptivity as opposed to the L^1 norm [3]. An interesting question is to precisely describe the transition between these two regimes as a function of the initialisation.

4 Numerical Experiments

For the numerical experiments, we follow gradient descent using the parameterization (1) with $\alpha(m) = \sqrt{m}$, appropriately scaling the weights $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ to achieve different dynamical behavior. In addition to those presented here, Section D presents additional numerical experiments

Cubic Splines. We show in Figure 4 that when $-\delta \ll r^2$ (*i.e.*we are in the kernel regime), growing the number of neurons grows causes the network function $f_{\boldsymbol{z}}$ converges to a cubic spline. For this experiment, we fit 10 points sampled from a square wave, and train only the parameters c (*i.e.*$\delta_i = \infty$).

Network Dynamics as a Function of δ. We show in Figure 5 that as we vary δ, the network function goes from being smooth and non-adaptive in the kernel regime ($\delta = -\infty$, *i.e.*training only



$\delta = -\infty$ $\delta = -1$ $\delta = 0$ $\delta = 1$ $\delta = \infty$

Figure 5: Comparison of fitting the network function to a sinusoid as δ varies (10000 epochs).

the parameter $\boldsymbol{c}$) to very adaptive ($\delta = \infty$, *i.e.*training only the parameters $\boldsymbol{a}, \boldsymbol{b}$). Note how as δ gets large, clusters of knots emerge at the sample positions (collinear points in the uv diagrams).

5 Concluding Remarks

We have studied the implicit bias of gradient descent in the approximation of univariate functions with single-hidden layer ReLU networks. Despite being an extremely simplified learning setup, it provides a clear illustration that such implicit bias can be drastically different depending on how the neural architecture is parameterized, normalized, or even initialized. Building up on recent theoretical work that studies neural networks in the overparameterized regime, we show how the model can behave either as a ‘classic’ cubic spline interpolation kernel, or as an adaptive interpolation method, where neurons concentrate on sample points where the approximation most needs them. Moreover, in the one-dimensional case, we complement existing works [29] to reveal a clear transition between these two extreme training regimes, which roughly correspond to $W^{1,2}$ and $W^{2,2}$ Sobolev spaces respectively.

Although in our univariate setup there is no clear advantage of one functional space over the other, our full description of the dynamics may prove useful in the high-dimensional regime, where the curse of dimensionality hurts Hilbert spaces defined by kernels [3]. We believe that the analysis of the PDE resulting from the mean-field regime (where adaptation occurs) in the low-dimensional setting will be useful to embark in the analysis of the high-dimensional counterpart. We note however that naively extending our analysis to high-dimensions would result in an exponential increase in the number of regions that define our piecewise linear flow, thus we anticipate new tools might be needed to carry it over. Moreover, the interpretation of ReLU features in terms of Green’s functions (as first pointed out in [29]) does not directly carry over to higher dimensions. Lastly, another important limitation of the mean-field analysis is that it cannot be easily adapted to deeper neural network architectures, since the neurons within the network are no longer exchangeable as in the many-particle system described above.

Acknowledgements: This work was partially supported by the Alfred P. Sloan Foundation, NSF RI-1816753, NSF CAREER CIF 1845360, Samsung Electronics, the NSF CAREER award 1652515, the NSF grant IIS-1320635, the NSF grant DMS-1436591, the NSF grant DMS-1835712, the SNSF grant P2TIP2_175859, the Moore-Sloan Data Science Environment, the DARPA D3M program, NVIDIA, Labex DigiCosme, DOA W911NF-17-1-0438, a gift from Adobe Research, and a gift from nTopology. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

[1] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

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

[4] Ronen Basri and David Jacobs. Efficient representation of low-dimensional manifolds using deep networks. *arXiv preprint arXiv:1602.04723*, 2016.

[5] Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019.

[6] Yuan Cao and Quanquan Gu. A generalization theory of gradient descent for learning overparameterized deep relu networks. *arXiv preprint arXiv:1902.01384*, 2019.

[7] Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

[8] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for overparameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

[9] Frank H Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

[10] Amit Daniely. Sgd learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2017.

[11] I Daubechies, R DeVore, S Foucart, B Hanin, and G Petrova. Nonlinear approximation and (deep) relu networks. *arXiv preprint arXiv:1905.02199*, 2019.

[12] Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

[13] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[14] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

[15] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018.

[16] Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. *arXiv preprint arXiv:1901.09021*, 2019.

[17] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

[18] Thomas Hotz and Fabian JE Telschow. Representation by integrating reproducing kernels. *arXiv preprint arXiv:1202.4443*, 2012.

[19] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[20] Chao Ma, Lei Wu, et al. A comparative analysis of the optimization and generalization property of two-layer neural network and random feature models under gradient descent dynamics. *arXiv preprint arXiv:1904.04326*, 2019.

[21] Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

[22] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, August 2018.

[23] Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? *arXiv preprint arXiv:1812.10004*, 2018.

[24] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[25] Grant Rotskoff, Samy Jelassi, Joan Bruna, and Eric Vanden-Eijnden. Global convergence of neuron birth-death dynamics. *arXiv preprint arXiv:1902.01843*, 2019.

[26] Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915*, 2018.

[27] Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks. *arXiv preprint arXiv:1712.08968*, 2017.

[28] Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? *arXiv preprint arXiv:1902.05040*, 2019.

[29] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A central limit theorem. *arXiv preprint arXiv:1808.09372*, 2018.

[30] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[31] Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in two-layer neural network optimization landscapes. *arXiv preprint arXiv:1802.06384*, 2018.

[32] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. *arXiv preprint arXiv:1811.10943*, 2018.

Appendix

A Spline Kernels

A.1 Proof of Theorem 5

Consider the factorization of μ_0 in terms of the marginal with respect to a, $q(a)$ and its conditional $\mu_a(b)$: $\mu_0(a,b) = q(a)\mu_a(b)$. Observe that

$$\begin{aligned} K_{\mathrm{RF}}(x,x') &= \int_{\mathcal{D}_a \times \mathcal{D}_b} [ax-b]_+ \cdot [ax'-b]_+ \mathrm{d}\mu_0(a,b) \qquad (29) \\ &= \int_{\mathcal{D}_a} K_a(x,x') \mathrm{d}q(a) \text{ , with} \end{aligned}$$

$$K_a(x,x') := \int_{I_a} [ax-b]_+ [ax'-b]_+ \mathrm{d}\mu_a(b) \ .$$

We will argue that each K_a defines a RKHS $\mathcal{H}_a$ of twice-continuously differentiable functions with appropriate boundary condition, and with norm $\|f\|_a$ corresponding to a weighted squared second derivative. We will then conclude by leveraging the fact that averages of positive definite kernels also define an RKHS [1, 18].

We verify that for each $a \neq 0$ this kernel defines a natural spline. Let $I_a = \mathrm{supp}(\mu_a) \subseteq \mathcal{D}_b$ and $\tilde{I}_a = \{u; au \in I_a\}$. Let $s_a = \inf\{x \in I_a\}$. For $f \in C^2(I_a)$ with f' and f'' in $L^2(I_a)$ such that $f(s_a) = f'(s_a) = 0$, let $f_a(x) := f(ax)$, $x \in \tilde{I}_a$. Assume first that I_a is an interval (and thus $\tilde{I}_a$ is also an interval). By the Taylor integral remainder theorem, we have for $x \in \tilde{I}_a$ that

$$f_a(x) = f(ax) = \int_{s_a}^{ax} (ax-u) f^{''}(u) \mathrm{d}u = \int_{I_a} [ax-u]_+ \frac{f^{''}(u)}{\mu_a(u)} \mu_a(u) \mathrm{d}u \ . \qquad (30)$$

This shows that $g_a = \frac{f^{''}}{\mu_a}$ is the kernel representation of f_a corresponding to K_a. This kernel defines an RKHS $\mathcal{H}_a$ over the space of C^2 functions f supported in $\tilde{I}_a$ with boundary conditions $f(s_a/a) = f'(s_a/a) = 0$. Moreover, its RKHS norm is by definition

$$\|f_a\|^2_{K_a} = \int_{I_a} \left| \frac{f^{''}(u)}{\mu_a(u)} \right|^2 \mathrm{d}\mu_a(u) \ .$$

For general support I_a, consider the closure $I_a^c := (\inf I_a, \sup I_a)$. Since $I_a \subseteq \mathbb{R}$ is measurable and the support of a continuous function, we can write it as $I_a = \cup_{k=1}^{\infty} I_a^{(k)}$, where $I_a^{(k)}$ are disjoint intervals. We modify the boundary condition for f accordingly, as

$$f(\sup I_a^{(k)}) = f(\inf I_a^{(k+1)}) \text{ and } f'(\sup I_a^{(k)}) = f'(\inf I_a^{(k+1)}) \ \forall \, k. \qquad (31)$$

We extend the representation (30) to I_a^c by first extending g_a to $\mathbb{R}$ as

$$\bar{g}_a(u) = \begin{cases} \frac{f^{''}(u)}{\mu_a(u)} & u \in I_a \\ 0 & \text{otherwise.} \end{cases}$$

We verify that for f in $C^2(I_a^c)$ satisfying the boundary conditions (31), $\bar{g}_a$ satisfies

$$f_a(x) = \int_{I_a^c} [ax-u]_+ \bar{g}_a(u) \mu_a(u) \mathrm{d}u \ . \qquad (32)$$

Therefore, if $h \in C^2(\tilde{I}_a^c)$, the change of variables $h = f_a$ for some f in $C^2(I_a^c)$ yields a kernel representation

$$\hat{h}_a(u) = \begin{cases} \frac{h^{''}(a^{-1}u)}{a^2 \mu_a(u)} & u \in I_a \\ 0 & \text{otherwise.} \end{cases} \ ,$$

and thus

$$\|h\|^2_{K_a} = \frac{1}{|a|} \int_{\tilde{I}_a} \frac{|h^{''}(u)|^2}{\bar{\mu}_a(u)} \mathrm{d}u \;, \tag{33}$$

where $\bar{\mu}_a(u) := \mu_a(au)$. For $f \in C^2(\mathbb{R})$, we can thus decompose it as $f = \tilde{f} + \bar{f}$, where $\tilde{f} \in \mathcal{H}_a$ is supported in $\tilde{I}_a^c$ and satisfies the boundary conditions (31), and $\bar{f}'' \equiv 0$ in $\tilde{I}_a$. The projection of f onto $\mathcal{H}_a$ is $\tilde{f}$ and

$$\|P_{\mathcal{H}_a} f\|^2_{K_a} = \frac{1}{|a|} \int_{\tilde{I}_a} \left| \frac{f^{''}(u)}{\bar{\mu}_a(u)} \right|^2 \mathrm{d}\bar{\mu}_a(u) \;. \tag{34}$$

Let us now show that we can 'integrate' the collection of kernels K_a and define a resulting RKHS $\mathcal{H}_{\mathrm{RF}}$ whose kernel is precisely K_{RF}. For that purpose, we verify first that for all x,

$$\int_{\mathcal{D}_a} K_a(x,x) \mathrm{d}q(a) < \infty \;. \tag{35}$$

Indeed, observe that

$$\begin{aligned}
\int_{\mathcal{D}_a} K_a(x,x) \mathrm{d}q(a) &= \int_{\mathcal{D}_a \times \mathcal{D}_b} [ax - b]_+^2 \mathrm{d}\mu_0(a,b) \\
&\leq 2\max(1,x^2) \int (a^2 + b^2) \mu_0(\mathrm{d}a, \mathrm{d}b) \leq 2\max(1,x^2)\sigma^2_{\mu_0} < \infty \;,
\end{aligned}$$

which proves (35). We can thus apply Theorem 3.1 from [18], which establishes that K_{RF} defines an RKHS

$$\mathcal{H}_{\mathrm{RF}} = \left\{ f ; f = \int_{\mathcal{D}_a} f_a \mathrm{d}q(a) \;;\; f_a \in \mathcal{H}_a \right\} \;,$$

with norm

$$\|f\|^2_{K_{\mathrm{RF}}} := \inf \left\{ \int_{\mathcal{D}_a} \|f_a\|^2_{K_a} \mathrm{d}q(a) \;;\; f = \int_{\mathcal{D}_a} f_a \mathrm{d}q(a) \right\} \;,$$

which, from (34), gives

$$\|f\|^2_{K_{\mathrm{RF}}} = \inf \left\{ \int_{\mathcal{D}_a} \left(\int_{\tilde{I}_a} \frac{|f_a^{''}(u)|^2}{\bar{\mu}_a(u)} \mathrm{d}u \right) \frac{\mathrm{d}q(a)}{|a|} \;;\; f = \int_{\mathcal{D}_a} f_a \mathrm{d}q(a) \right\} \;. \tag{36}$$

Let us now show that (36) has an explicit form as a weighted curvature. Let $\Omega := \cup_a \tilde{I}_a$. From $f = \int_{\mathcal{D}_a} f_a \mathrm{d}q(a)$ we deduce that

$$\forall\, u \in \Omega \,,\, f''(u) = \int_{\mathcal{D}_a} f_a''(u) \mathrm{d}q(a) \;.$$

For each $u \in \Omega$, denote $\beta(a) = f_a''(u)$ for $a \in \mathcal{B}_u := \{a ; u \in \tilde{I}_a\}$, and $\nu(u) := \int_{\mathcal{B}_u} |a| \bar{\mu}_a(u) \mathrm{d}q(a)$. Since $\|f\|^2_{K_{\mathrm{RF}}} < \infty$ and $f'' \in C^0(\Omega)$, let us first argue that $\nu(u) = 0$ necessarily implies that $f''(u) = 0$. Indeed, if $\nu(u) = 0$, it follows that $\bar{\mu}_a(u) = 0$ for all a, and since each f_a'' is also continuous, from (36) we deduce that necessarily $f_a''(u) = 0$ for all a, which implies that $f''(u) = 0$.

Let us now assume that $\nu(u) > 0$. The constrained minimisation problem

$$\min_{\beta(a)} \int_{\mathcal{B}_u} |\beta(a)|^2 \frac{\mathrm{d}q(a)}{\bar{\mu}_a(u)|a|} \quad s.t \quad f''(u) = \int_{\mathcal{B}_u} \beta(a) \mathrm{d}q(a) \tag{37}$$

has an associated Lagrangian

$$\mathcal{L}(\beta, \lambda) = \int_{\mathcal{B}_u} |\beta(a)|^2 \frac{\mathrm{d}q(a)}{\bar{\mu}_a(u)|a|} + \lambda \left(f''(u) - \int_{\mathcal{B}_u} \beta(a) \mathrm{d}q(a) \right) \;.$$

The first-order KKT optimality conditions directly give

$$\beta(a) = \frac{|a| \bar{\mu}_a(u) f''(u)}{\int_{\mathcal{B}_u} |a'| \bar{\mu}_{a'}(u) \mathrm{d}q(a')} \;, \tag{38}$$

resulting in a minimum of (37) at

$$\frac{|f''(u)|^2}{\nu(u)} \; . \tag{39}$$

Finally, we need to show that one can make these optimal choices for each u to define $f_a \in \mathcal{H}_a$ for each a. From (38), f_a'' is defined for each $u \in \tilde{I}_a$ as $f_a''(u) = f''(u)\eta_a(u)$, where $\eta_a(u) = \frac{|a|\bar{\mu}_a}{\nu(u)}$ is the normalised density satisfying $\int \eta_a(u)\mathrm{d}q(a) = 1$ for all u. We verify that $f''\eta_a$ is in $L^2(\tilde{I}_a)$ thanks to the smoothness hypothesis of μ_0. Indeed, by assumption, the normalised density η_a is bounded for each a, $\sup_u \eta_a(u) \leq K_a$, and thus

$$\int_{\tilde{I}_a} |f_a''(u)|^2 \mathrm{d}u = \int_{\tilde{I}_a} |f''(u)|^2 \eta_a(u)^2 \mathrm{d}u \leq K_a^2 \int_\Omega \frac{|f''(u)|^2}{\nu(u)^2} \mathrm{d}u < \infty \; ,$$

since $f'' \in L^2(\Omega)$. We can therefore define f_a on each $\tilde{I}_a$ thanks the reproducing property

$$\forall \, a, \, x \in \tilde{I}_a \; , \; f_a(x) = \int f_a''(u)[x-u]_+ \mathrm{d}u \; .$$

Integrating over a yields

$$\begin{aligned} \int_{\mathcal{D}_a} f_a(x)\mathrm{d}q(a) &= \int_{\mathcal{D}_a} \int f_a''(u)[x-u]_+(\mathrm{d}u)\mathrm{d}q(a) = \int \left(\int f_a''(u)\mathrm{d}q(a) \right) [x-u]_+(\mathrm{d}u) \\ &= \int f''(u)[x-u]_+\mathrm{d}u = f(x) \, , \end{aligned}$$

which proves that $(f_a)_a$ is a feasible set to represent f in (36).

Finally, by applying the optimality conditions (37, 39) to each $u \in \Omega$ one obtains

$$\|f\|^2_{K_{\mathrm{RF}}} = \int_\Omega \frac{|f''(u)|^2}{\nu(u)} \mathrm{d}u$$

which concludes the proof of the first statement.

Finally, let us verify that in the specific case when $\mu_0(a,b) = q(a)\mathbf{1}(b \in I_a)$ the kernel is piece-wise cubic. Fix $a \in \mathcal{D}_a$ and suppose first that $a > 0$. Let $I_a = [s_a, S_a]$, and denote $\tilde{I}_a = [s_a/a, S_a/a]$. A direct calculation shows that when $x' \in \tilde{I}_a$, $K_a(x,x')$ is 0 when $x < s_a/a$, cubic in x when $x \leq x'$, and linear when $x > x'$. When $x' < s_a/a$, then $K_a(\cdot, x') \equiv 0$, and finally when $x' > S_a/a$ we have that $K_a(x,x') = 0$ for $x < s_a/a$, $K_a(x,x')$ is cubic for $x \in \tilde{I}_a$ and linear for $x > S_a/a$. We also verify that its first and second derivatives are continuous. In summary, the evaluation functions $K_a(\cdot, x')$ are 0 for $x < s_a/a$, a piece-wise cubic inside $\tilde{I}_a$, and linear for $x > S_a/a$. When $a < 0$ we verify the same holds true by flipping the sense of inequalities. □

A.2 Proof of Corollary 6

The proof follows closely the previous section.

Observe first that K_{NTK} can be written as the sum of three kernels: $K_{\mathrm{NTK}} = K_{\mathrm{RF}} + \mathbb{E}(c^2)(K_1 + K_2)$, with

$$K_1(x,x') = \mathbb{E}_{\mu_0}[\mathbf{1}(ax - b > 0)\mathbf{1}(ax' - b > 0)] \; , \; K_2(x,x') = xx'\mathbb{E}_{\mu_0}[\mathbf{1}(ax - b > 0)\mathbf{1}(ax' - b > 0)] \; .$$

We will first assume that μ_0 is concentrated at $a = 1$ and uniform $b \in (0,1)$. We can then extend to the general case by the previous proof. Observe that for $f \in C^1([0,1))$ with $f(0) = 0$ the Taylor integral remainder at order 1 now gives

$$f(x) = \int_0^1 f'(u)\mathbf{1}(x - u > 0)\mathrm{d}u \; , \; x \in (0,1) \; .$$

It follows that f' is the kernel representation associated to $K_1(x,x')$.

Similarly,

$$xf(x) = \int_0^1 f'(u)x\mathbf{1}(x - u > 0)\mathrm{d}u \; , \; x \in (0,1) \; ,$$

which shows that f' is the kernel representation of $g(x) = xf(x)$ corresponding to the kernel $K_2(x, x')$. A change of variables then implies that

$$f' = \left(\frac{g}{x}\right)' = \frac{xg' - g}{x^2}$$

is the contribution of K_2 to the RKHS norm of K_{NTK}. The proof is completed by repeating the argument used in the proof of Theorem 5 that replaces the uniform measure in the interval $(a, b) \in \{1\} \times (0, 1)$ by a smooth density μ_0. We leave the derivation of an explicit form of the mixed norm for future work. □.

A.3 Radial Kernel

We verify by direct calculation that

$$\begin{aligned}
K^r_{\mathrm{RF}}(x, x') &= \int_{\mathcal{D}_a \times \mathcal{D}_b} [ax + b]_+ \cdot [ax' + b]_+ \mu_0(da, db) \qquad (40)\\
&= \frac{1}{2\pi} \int_{\arctan(x')}^{\arctan(x)+\pi} \int_0^\infty r^3 (\cos(\eta)x + \sin(\eta))(\cos(\eta)x' + \sin(\eta))\mu_0(dr)d\eta \\
&= \frac{C}{2\pi} \int_{\arctan(x')}^{\arctan(x)+\pi} \left(xx' \cos^2(\eta) + \sin^2(\eta) + (x + x')\cos(\eta)\sin(\eta)\right) d\eta \\
&= \frac{C}{2\pi} \left[xx' \left(\frac{\eta}{2} + \frac{\sin(2\eta)}{4}\right) + \left(\frac{\eta}{2} - \frac{\sin(2\eta)}{4}\right) + \frac{(x + x')}{2} \sin^2(\eta)\right]_{\arctan(x')}^{\arctan(x)+\pi} \\
&= \frac{C}{4\pi} \Bigg\{ (\pi - \arctan(x') + \arctan(x))(xx' + 1) + \left(\frac{x'}{1 + (x')^2} - \frac{x}{1 + x^2}\right)(xx' - 1) + \\
&\qquad + (x + x')\left(\frac{(x')^2}{1 + (x')^2} - \frac{x^2}{1 + x^2}\right)\Bigg\},
\end{aligned}$$

B Mean Field Computations

Making use of notation introduced in Section 3.1, we have that if $w = (\hat{r}, \theta)$, with $\theta \in \mathcal{A}_k$, then

$$\begin{aligned}
\nabla_\theta V(w; \mu_t) &= -\nabla_\theta F(w) + \int_{\mathcal{D}} \nabla_\theta K(w, w')\mu_t(dw') \\
&= -\hat{r} \left(\sum_{j \in \mathcal{C}_k} y_j \langle \tilde{x}_j, t(\theta)\rangle - \int_{\mathcal{D}} r' \sum_{j \in \mathcal{C}_k} \langle \tilde{x}_j, t(\theta)\rangle \langle \tilde{x}_j, d(\theta')\rangle_+ \mu_t(d\hat{r}', d\theta')\right) \\
&= -\hat{r} \sum_{j \in \mathcal{C}_k} \langle \tilde{x}_j, t(\theta)\rangle \left(y_j - \int_{\mathbb{R} \times \mathcal{B}_j} \hat{r}' \langle \tilde{x}_j, d(\theta')\rangle \mu_t(d\hat{r}', d\theta')\right) \\
&= \hat{r} \left\langle \sum_{j \in \mathcal{C}_k} \rho_j(t) \tilde{x}_j, t(\theta) \right\rangle, \qquad (41)
\end{aligned}$$

where $\rho_j(t) = f_{\mu_t}(x_j) - y_j = \int_{\mathbb{R} \times \mathcal{B}_j} c\langle \tilde{x}_j, \theta\rangle \mu_t(d\hat{r}, d\theta) - y_j$ is the residual at point x_j at time t. Similarly, the field in the direction of the charges is given by

$$\nabla_{\hat{r}} V(w; \mu_t) = \left\langle \sum_{j \in \mathcal{C}_k} \rho_j(t) \tilde{x}_j, \theta \right\rangle . \tag{42}$$

B.1 Proof of Proposition 1

We observe that for each j,

$$\begin{aligned}
\dot{\rho}_j(t) &= \partial_t f_{\mu_t}(x_j) = \partial_t \left(\int_{\mathcal{D}} \varphi(w; x_j)\mu_t(dw) \right) \\
&= -\int_{\mathcal{D}} \langle \nabla_w \varphi(w; x_j), \nabla V(w; \mu_t) \rangle \mu_t(dw) \\
&= -\int_{\mathcal{D}} \left(\nabla_\theta \varphi(w; x_j) \cdot \nabla_\theta V(w; \mu_t) + \nabla_r \varphi(w; x_j) \cdot \nabla_r V(w; \mu_t) \right) \mu_t(dw) \\
&= -\sum_{k; \mathcal{A}_k \subset \mathcal{B}_j} \int_{\mathbb{R}\times\mathcal{A}_k} \left(r^2 \tilde{x}_j^\top (t(\theta)t(\theta)^\top)(\sum_{j' \in \mathcal{C}_k} \rho_{j'}(t)\tilde{x}_{j'}) + \tilde{x}_j^\top (\theta\theta^\top)(\sum_{j' \in \mathcal{C}_k} \rho_{j'}(t)\tilde{x}_{j'}) \right) \mu_t(dw) \\
&= -\tilde{x}_j^\top \sum_{k; \mathcal{A}_k \subset \mathcal{B}_j} \Sigma_k(t)(\sum_{j' \in \mathcal{C}_k} \rho_{j'}(t)\tilde{x}_{j'}) \,, \qquad (43)
\end{aligned}$$

where

$$\Sigma_k(t) = \int_{\mathbb{R}\times\mathcal{A}_k} \left(r^2 t(\theta)\, t(\theta)^\top + \theta\, \theta^\top \right) \mu_t(dr, d\theta)$$

tracks the covariance of the measure along each cylindrical region.

C Changing Metric in the Dynamics

Lemma 7. *If $\boldsymbol{z}(t) = (\boldsymbol{a}(t), \boldsymbol{b}(t), \boldsymbol{c}(t))$ is a solution of the gradient flow (2), then the quantities*

$$\boldsymbol{\delta} = (\delta_i = c_i(t)^2 - a_i(t)^2 - b_i(t)^2)_{i=1}^m \qquad (44)$$

remain constant for all t. In particular, given a reduced neuron (r_i, θ_i), we can uniquely recover the original neuron (a_i, b_i, c_i), since

$$c_i^2 = \frac{\delta_i + \sqrt{\delta_i^2 + 4r_i^2}}{2}. \qquad (45)$$

Proof. The gradient equations of the loss $L(\boldsymbol{z})$ can be written as

$$\begin{aligned}
\nabla_{a_i} L(\boldsymbol{z}) &= c_i \sum_{j=1}^s \mathbb{1}[a_i x_j - b_i \geq 0] x_j r_j, \\
\nabla_{b_i} L(\boldsymbol{z}) &= c_i \sum_{j=1}^s \mathbb{1}[a_i x_j - b_i \geq 0] r_j, \qquad (46) \\
\nabla_{c_i} L(\boldsymbol{z}) &= \sum_{j=1}^s \mathbb{1}[a_i x_j - b_i \geq 0](a_i x_j - b_i) r_j.
\end{aligned}$$

From these expressions we see that

$$\begin{aligned}
\dot{\delta}_i &= 2c_i\dot{c}_i - 2a_i\dot{a}_i - 2b_i\dot{b}_i \\
&= 2c_i \nabla_{c_i} L(\boldsymbol{z}) - 2a_i \nabla_{a_i} L(\boldsymbol{z}) - 2b_i \nabla_{b_i} L(\boldsymbol{z}) \\
&= 0.
\end{aligned}$$

Using $r_i^2 = c_i\sqrt{a_i^2 + b_i^2}$, we see that $c_i^2 - \frac{r_i^2}{c_i^2} = \delta_i$ implies $c_i^4 - \delta_i c_i^2 - r^2 = 0$, and thus (45). ☐

Proposition 8. *Let $\boldsymbol{z}(t)$ be a solution gradient flow (2) of $L(\boldsymbol{z})$, and let $\boldsymbol{\delta} = (\delta_i) \in \mathbb{R}^m$ be the vector of invariants (18), which depend only on the initialization $\boldsymbol{z}(0)$. If $\boldsymbol{w}(t) = (\boldsymbol{r}(t), \boldsymbol{\theta}(t))$ is curve of reduced parameters corresponding to $\boldsymbol{z}(t)$, then we have that*

$$\dot{\boldsymbol{w}}_i(t) = \boldsymbol{P}_i \cdot \nabla_{\boldsymbol{w}_i} \tilde{L}(\boldsymbol{w}), \quad i = 1, \dots, m,$$

where

$$\boldsymbol{P}_{\delta_i}(r_i) = \begin{bmatrix} \frac{m^2}{\alpha(m)^2}(a_i^2 + b_i^2 + c_i^2) & 0 \\ 0 & \frac{1}{a_i^2+b_i^2} \end{bmatrix} = \begin{bmatrix} \frac{m^2}{\alpha(m)^2}\left(\frac{r_i^2}{c(r_i)^2} + c(r_i)^2\right) & 0 \\ 0 & \frac{c(r_i)^2}{r_i^2} \end{bmatrix},$$

and $c(r_i)^2 = \frac{\delta_i + \sqrt{\delta_i^2 + 4r_i^2}}{2}$.

Proof. The Jacobian of the mapping $\boldsymbol{\pi}$ from parameters to reduced parameters is given by

$$\nabla(\boldsymbol{\pi})(a_i, b_i, c_i) = \begin{bmatrix} \frac{m}{\alpha(m)} \frac{ca}{\sqrt{a_i^2+b_i^2}} & \frac{m}{\alpha(m)} \frac{cb}{\sqrt{a_i^2+b_i^2}} & \frac{m}{\alpha(m)} \sqrt{a_i^2 + b_i^2} \\ -\frac{b}{a_i^2+b_i^2} & \frac{a}{a_i^2+b_i^2} & 0 \end{bmatrix}, \qquad i = 1, \dots, m.$$

This implies that the *tangent kernel* $\boldsymbol{P}_{\delta_i}(r_i) = \nabla(\boldsymbol{\pi})^\top \nabla(\boldsymbol{\pi})$ is as in (20). We emphasize that the fact that this kernel can be written only as a function of $\boldsymbol{w}$ (and, in fact, only of $\boldsymbol{r}$) relies in essential manner on Lemma 3. ☐

D Additional Numerical Experiments

In Figure 6, we plot the trajectories of neurons for $\delta = \pm\infty$ over 10000 epochs. We see that, if $\delta = -\infty$, the neurons move radially away from the origin and thus the knot positions do not change (top row). In stark contrast, if $\delta = \infty$, the neurons adapt to the input data, and the knots “stick” to input samples (bottom row).

We remark in Figure 7 that the same initial function can yield extremely different results depending on δ.

We now show the effect of varying the number of neurons during training. In this example we fit 20 samples from a sine wave using 20, 200, and 2000 neurons respectively. In PyTorch, the default initializatioon is such that $\boldsymbol{a}, \boldsymbol{b}\ U(-1, 1)$ and $\boldsymbol{c}\ U(-1/m, 1/m)$. Thus, as we scale down the numberof neurons, the value of δ grows, making the network function adapt more to the data. Figure 8 shows the results of this experiment.

D.1 Visualizing Attractor Samples

We can visualize the vector field $(\partial_t(\hat{r}, \theta)$ by considering the change of metric from $w = (\hat{r}, \theta)$ to (u, v) with the map

$$\pi_{(u,v)}(\hat{r}, \theta) = (|\hat{r}| \cos\theta, |\hat{r}| \sin\theta) = (u, v).$$

Assuming we know the sign of $\hat{r}$, the vector field

$$\begin{bmatrix} \partial_t u \\ \partial_t v \end{bmatrix} = D\pi_{(u,v)} D\pi_{(u,v)}^T \begin{bmatrix} \partial_t r \\ \partial_t \theta \end{bmatrix} \tag{47}$$

Observing that $D\pi_{(u,v)} D\pi_{(u,v)}^T = I$, we have simply that

$$\begin{bmatrix} \partial_t u \\ \partial_t v \end{bmatrix} = \begin{bmatrix} \partial_t r \\ \partial_t \theta \end{bmatrix}$$

Figure 9 shows a plot of this vector felt by a single particle in uv in the case where $\delta = \infty$. In this case, the partial derivative $\partial_t r$ remains unchanged. Furthermore, we remark that at the boundaries of samples, the vector field can change directions, causing these samples to become “attractors” or “repulsors” (see Lemma 2 in the main document).

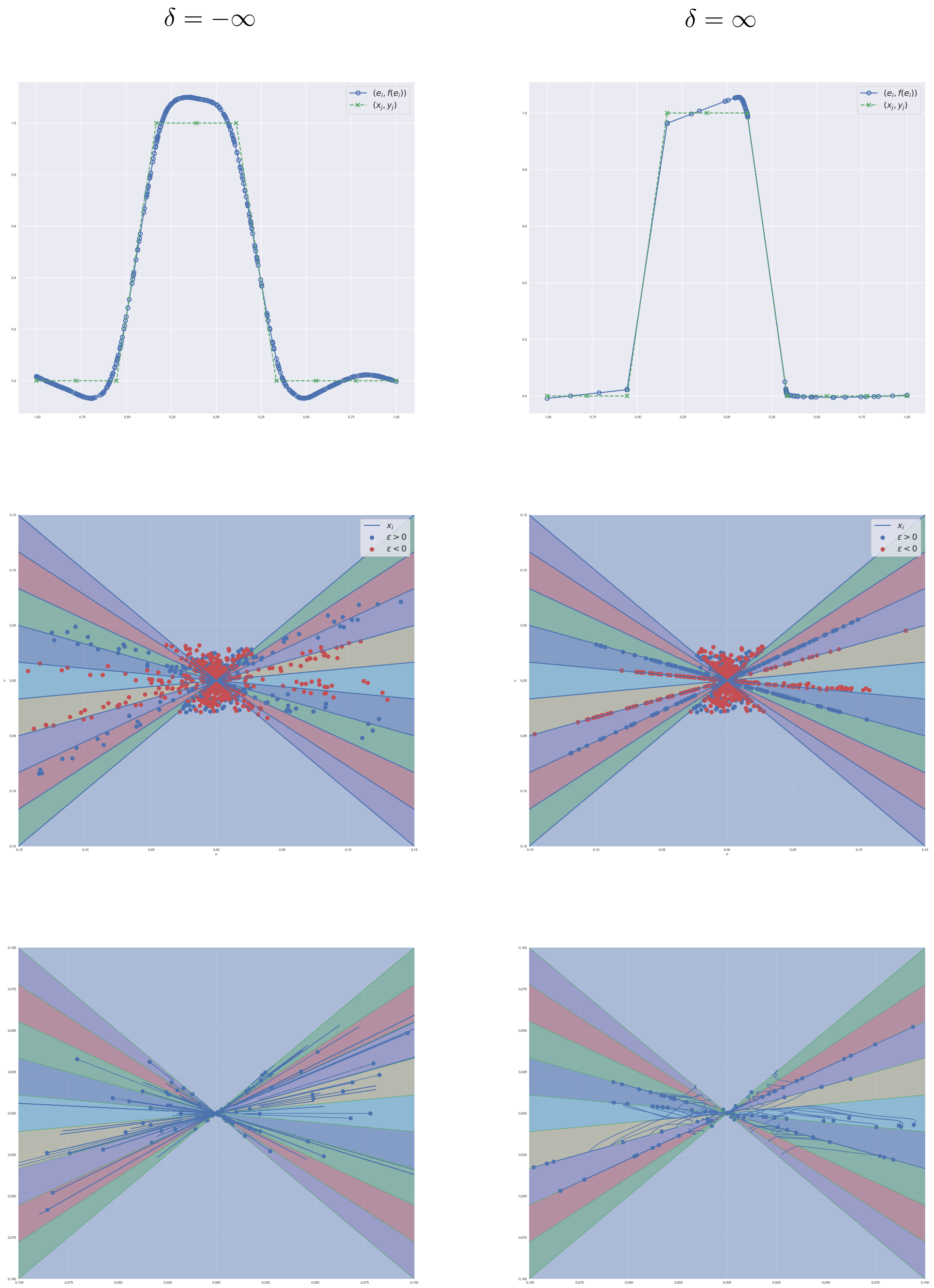
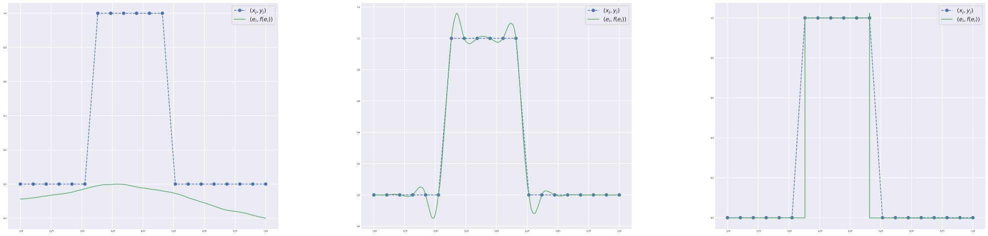


Figure 6: Evolution of 1000 neurons over 10000 epochs for $\delta = \pm\infty$ while fitting 10 points sampled from a square wave. Left: plotted network function after training. Middle: state of the network in uv space after training. Right: training trajectories of 100 random neurons.



Epoch 0
$\boldsymbol{z} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$

Epoch 10^4
$\boldsymbol{z} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$

Epoch 10^4
$\boldsymbol{z} = (10^3\boldsymbol{a}, 10^3\boldsymbol{b}, 10^{-3}\boldsymbol{c})$

Figure 7: Left: A network (green) with an initial set of parameters initial parameters $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is used to approximate a given function (blue). Scaling the initial parameters (right) leads to a very different fit (middle).

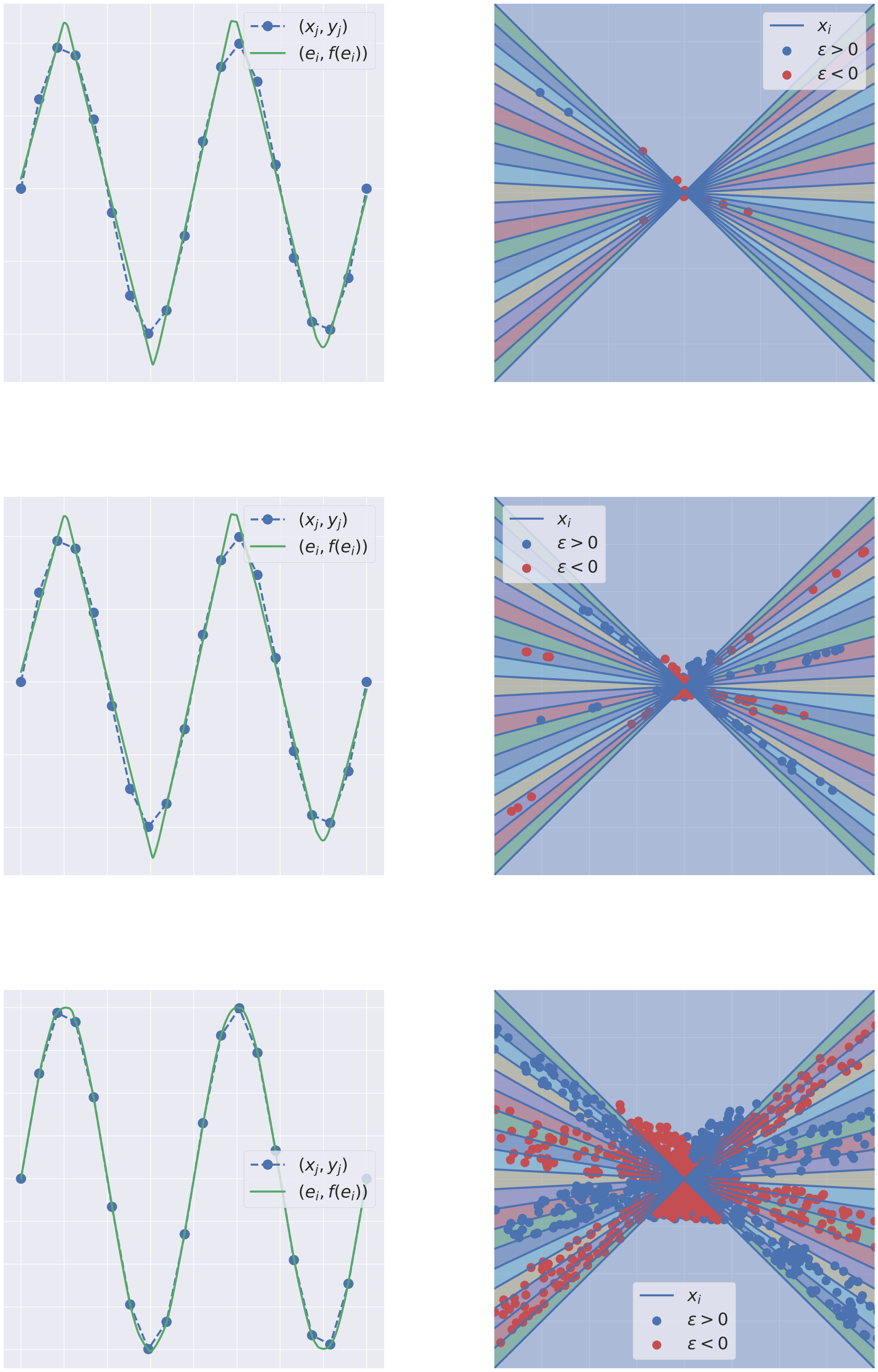


Figure 8: The effect of varying the number of neurons m the top image uses 20 neurons, the middle uses 200 and the bottom uses 2000. Observe that with fewer neurons, the function is adaptive to the data since δ gets larger.

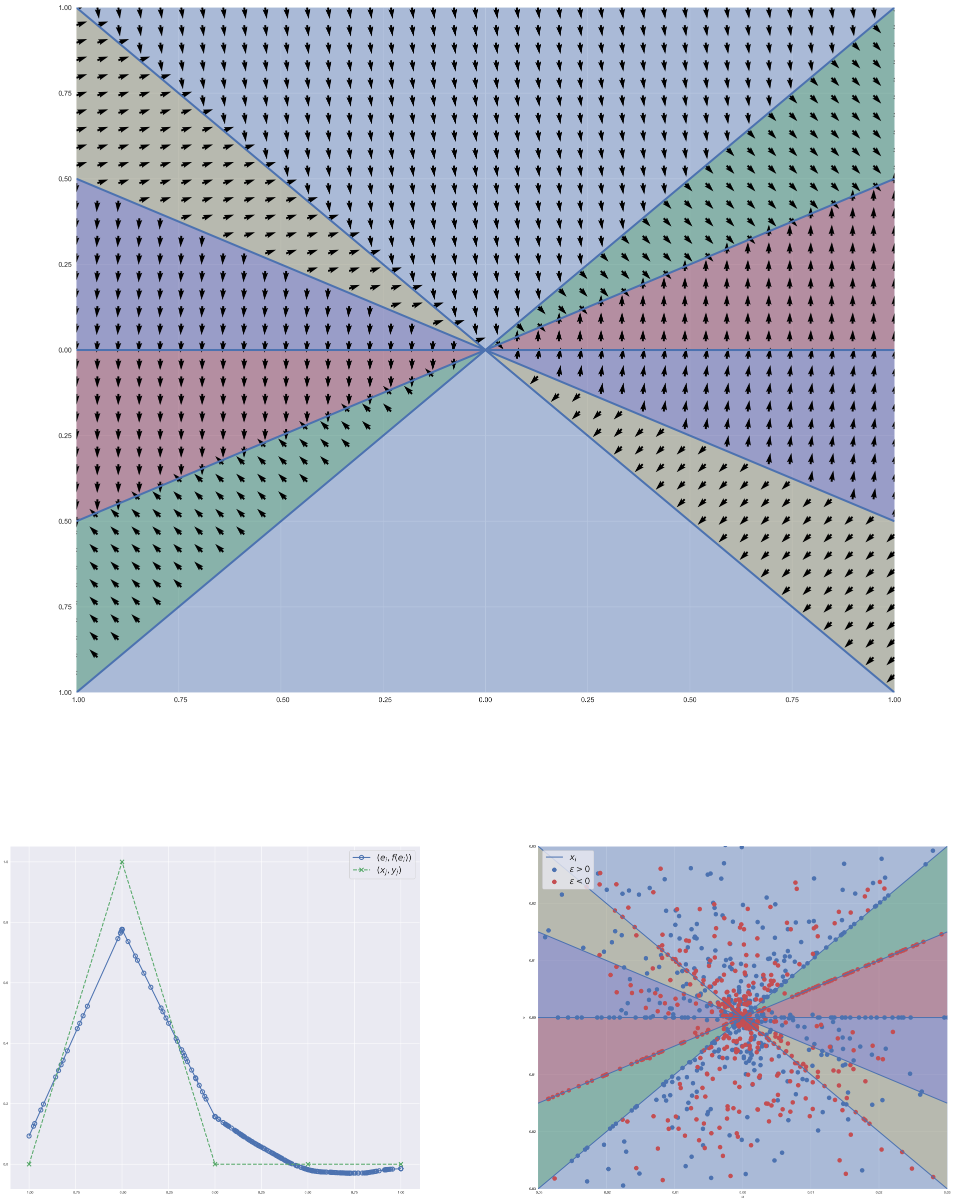


Figure 9: *Top:* The gradient field (21) felt by a particle. Note how the vectors change directions at certain samples. These samples are "attractors" or "repulsors" where particles get stuck or get pushed away from. *Bottom Left:* A plot of the network function for the gradient field in the top image. Observe how there are clusters of neurons (blue circles) aligned with certain samples. *Bottom Right:* A plot of the neurons in uv space. Observe how the red neurons cluster at "attractor" points in the top image