

Training Neural Networks as Learning Data-adaptive Kernels: Provable Representation and Approximation Benefits

Xialiang Dou

Department of Statistics, University of Chicago

Tengyuan Liang*

Booth School of Business, University of Chicago

Abstract

Consider the problem: given the data pair (\mathbf{x}, \mathbf{y}) drawn from a population with $f_*(x) = \mathbf{E}[\mathbf{y}|\mathbf{x} = x]$, specify a neural network model and run gradient flow on the weights over time until reaching any stationarity. How does f_t , the function computed by the neural network at time t , relate to f_* , in terms of approximation and representation? What are the provable benefits of the adaptive representation by neural networks compared to the pre-specified fixed basis representation in the classical nonparametric literature? We answer the above questions via a dynamic reproducing kernel Hilbert space (RKHS) approach indexed by the training process of neural networks. Firstly, we show that when reaching any local stationarity, gradient flow learns an adaptive RKHS representation and performs the global least-squares projection onto the adaptive RKHS, simultaneously. Secondly, we prove that as the RKHS is data-adaptive and task-specific, the residual for f_* lies in a subspace that is potentially much smaller than the orthogonal complement of the RKHS. The result formalizes the representation and approximation benefits of neural networks. Lastly, we show that the neural network function computed by gradient flow converges to the kernel ridgeless regression with an adaptive kernel, in the limit of vanishing regularization. The adaptive kernel viewpoint provides new angles of studying the approximation, representation, generalization, and optimization advantages of neural networks.

Keywords: adaptive estimation, neural networks, reproducing kernel Hilbert space, gradient flow dynamics, representation learning, algorithmic approximation, interpolation.

*Liang gratefully acknowledges support from the George C. Tiao Fellowship.

1 Introduction

Consider i.i.d. data pairs drawn from a joint distribution $(\mathbf{x}, \mathbf{y}) \sim P = P_x \times P_{y|x}$ on the space $\mathcal{X} \times \mathcal{Y}$. At the intersection of statistical learning theory [Vapnik, 1998] and approximation theory [Cybenko, 1989], the following *approximation* problem requires to be first understood, before any further statistical results to be established. For a model class \mathcal{F} , one is interested in whether there exists $f \in \mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ such that the population squared loss is small,

$$L(f) = \mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim P} \frac{1}{2} (\mathbf{y} - f(\mathbf{x}))^2 = \mathbf{E}_{\mathbf{x} \sim P_x} \frac{1}{2} (f_*(\mathbf{x}) - f(\mathbf{x}))^2 + \mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim P} \frac{1}{2} (\mathbf{y} - f_*(\mathbf{x}))^2, \quad (1.1)$$

with the conditional expectation (or Bayes estimator) defined as $f_*(x) := \mathbf{E}[\mathbf{y}|\mathbf{x} = x]$. Eqn. (1.1) generally reads as approximating f_* in the mean squared error sense.

Statistically, researchers approach the above question mainly in two ways. The first is by assuming that the conditional expectation f_* lies in the correct model class \mathcal{F} . For example, say \mathcal{F} consists of linear models or splines with a particular order of smoothness, or more broadly functions lying in a reproducing kernel Hilbert space (RKHS). Conceptually, this “well-specification” assumption requires substantial knowledge about what model class \mathcal{F} might be suitable for the regression task at hand, which is often unavailable in practice. Within each framework, minimax optimal rates and extensive study have been established in [Stone, 1980, Wahba, 1990]. The second way, which extends the first approach further, considers all f_* under some mild conditions. Building upon certain *universal approximation theorem*, one studies a sequence of model classes \mathcal{F}_ϵ called sieves with ϵ changing [Geman and Hwang, 1982], such that the class \mathcal{F}_ϵ contains an ϵ -approximation to any f_* under some metric. A final result usually requires a careful balancing of the approximation and stochastic error by tuning ϵ . Particular cases for the latter approach include polynomials (Stone-Weierstrass, Bernstein), radial-basis [Park and Sandberg, 1991, Niyogi and Girosi, 1996], and two-layer and multi-layer neural networks [Cybenko, 1989, Hornik et al., 1989, Anthony and Bartlett, 2009, Rahimi and Recht, 2008, Daniely et al., 2016, Bach, 2017, Farrell et al., 2018, Koehler and Risteski, 2018, Poggio et al., 2017].

However, the following significant drawbacks of the above current theory make it inadequate

to present an *adaptive* and realistic explanation of the practical success of *neural networks*. Firstly, the function computed in practice could be very different from that claimed in the approximation theory, either by the existence or by constructions. To see this, consider the multi-layer neural networks. It is hard to conceive that the function, computed in practice via now-standard stochastic gradient descent (SGD) training procedure, is close to the one asserted by the universal approximation results. Secondly, in practice, researchers usually explore different model classes \mathcal{F} to learn which representation best suits the data. For example, using different kernels machines, random forests, or specify certain architectures then run SGD on neural networks. In this case, strictly speaking, the choice of the model class depends on the data in an *adaptive* way, without prior knowledge about the basis. There have been substantial advances made to address the above two concerns — for instance, Jones [1992] on the first and Huang et al. [2008], Barron et al. [2008] on the second — for \mathcal{F} being a linear span of a library of candidate functions (union of various set of basis that can be correlated), with greedy selection rules. Nevertheless, the current theory still falls short of describing the approximation and adaptivity for the non-convex and possibly non-smooth gradient descent training on all-layer weights of the neural networks, as done in practice.

We take a step to bridge the above mismatch in the current theory and practice for neural networks and to establish a theoretical framework where the model classes adapt to the data. In particular, we answer the following *algorithmic approximation* question:

Given data pair $(\mathbf{x}, \mathbf{y}) \sim P$, denote $f_*(x) = \mathbf{E}[\mathbf{y}|\mathbf{x} = x]$. Specify a neural networks model, and run gradient flow until any stationarity ($t \rightarrow \infty$). Denote the computed function to be $f_t(x)$. How does $f_t(x)$ relate to $f_*(x)$, in terms of approximation and representation?

Also, we aim to formalize and shed light on the *representation benefits* of neural networks:

What are the provable benefits of the adaptive representation learned by training neural networks compared to the classical nonparametric pre-specified fixed basis representation?

The intimate connection between two-layer neural networks and reproducing kernel Hilbert spaces (RKHS) has been studied in the literature, see [Rahimi and Recht \[2008\]](#), [Cho and Saul \[2009\]](#), [Daniely et al. \[2016\]](#), [Bach \[2017\]](#), [Jacot et al. \[2018\]](#). However, to the best of our knowledge, known results are mostly based on a *fixed* RKHS (in our notation K_0 in [Section 5.1](#)). In that sense, random features for kernel learning [[Rahimi and Recht, 2008, 2009](#), [Rudi and Rosasco, 2017](#)] can be viewed as neural networks with fixed random sampled first layer weights, and tunable second layer weights. From the neural networks side, [Rotskoff and Vanden-Eijnden \[2018\]](#), [Mei et al. \[2018\]](#), [Sirignano and Spiliopoulos \[2019\]](#) study the mean-field theory for two-layer neural networks, and [Jacot et al. \[2018\]](#), [Du et al. \[2018\]](#), [Chizat and Bach \[2018\]](#), [Ghorbani et al. \[2019\]](#) study the linearization of neural networks around the initialization and draw connections to RKHS \mathcal{K}_0 in various over-parametrized settings. In contrast, we will establish a general theory with the *dynamic* and *data-adaptive* RKHS \mathcal{K}_t obtained via training neural networks, with standard gradient flow on weights of *both* layers. Connections and distinctions to the literature that motivates our study are further discussed with details in [Section 5](#). As a distinctive feature of the adaptive theory, we emphasize that all $f_* \in L^2(P_x)$ is considered, without pre-specified structural assumptions.

1.1 Problem Formulation

In this paper, we consider the time-varying function f_t to approximate f_* , parametrized by a two-layer rectified linear unit (ReLU) neural network (NN).

$$f_t(x) := \sum_{j=1}^m w_j(t) \sigma(x^T u_j(t)). \quad (1.2)$$

The time index t corresponds to the evolution of parameters driven by the gradient flow/descent (GD) training dynamics. Here each individual pair $(w_j \in \mathbb{R}, u_j \in \mathbb{R}^d)$ in the summation is associated with a *neuron*. Consider the gradient flow as the training dynamics for the weights of the neurons: for the loss function $\ell(y, f) = (y - f)^2/2$ and the random variable $\mathbf{z} := (\mathbf{x}, \mathbf{y})$,

the parameters (w_j, u_j) evolve with time as follows

$$\frac{dw_j(t)}{dt} = -\mathbf{E}_{\mathbf{z}} \left[\frac{\partial \ell(\mathbf{y}, f_t)}{\partial f} \sigma(\mathbf{x}^T u_j(t)) \right], \quad \frac{du_j(t)}{dt} = -\mathbf{E}_{\mathbf{z}} \left[\frac{\partial \ell(\mathbf{y}, f_t)}{\partial f} w_j(t) \mathbb{1}_{\mathbf{x}^T u_j(t) \geq 0} \mathbf{x} \right]. \quad (1.3)$$

Equivalently, we can rewrite the function computed by NN at time t as

$$f_t(x) := \int \sigma(x^T u) \tau_t(du), \quad (1.4)$$

where $\tau_t = \sum_{j=1}^m w_j(t) \delta_{u_j(t)}$ is a signed combination of delta measures. We will define a careful rescaling of τ_t denoted as ρ_t (Eqn. (5.8)), then derive the corresponding distribution dynamic for ρ_t driven by the gradient flow later in Section 5.2. The rescaled formulation naturally extends to the infinite neurons case with $m \rightarrow \infty$.

In this paper, by considering various distributions of \mathbf{z} , we study two following problems: approximation and empirical risk minimization (ERM).

Function Approximation: The data pair $\mathbf{z} \sim P$ is sampled from the population joint distribution. We are going to answer how f_t approximates $f_*(x) = \mathbf{E}[\mathbf{y} | \mathbf{x} = x]$ in function spaces, induced by the gradient flow on neuron weights

$$\mathbf{E}_{\mathbf{z} \sim P} (\mathbf{y} - f_t(\mathbf{x}))^2 = \|f_t - f_*\|_{L_\mu^2}^2 + \mathbf{E}_{\mathbf{z} \sim P} (\mathbf{y} - f_*(\mathbf{x}))^2. \quad (1.5)$$

Here we denote $\mu := P_x$, and remark that all $f_* \in L_\mu^2$ are considered without additional assumptions.

ERM and Interpolation: The data pair $\mathbf{z} \sim \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}=x_i, \mathbf{y}=y_i}$ follows the empirical distribution. We will study gradient flow for the ERM

$$\frac{1}{2n} \sum_{i=1}^n (y_i - f_t(x_i))^2. \quad (1.6)$$

In this case, the target reduces to $\hat{\mathbf{E}}[\mathbf{y} | \mathbf{x} = x_i] = y_i$ with $\hat{\mathbf{E}}$ as the empirical expectation. When the minimizer of Eqn. (1.6) achieves the zero loss, we call it the *interpolation* problem [Zhang et al., 2016, Belkin et al., 2018b, Ma et al., 2017, Liang and Rakhlin, 2018, Rakhlin and Zhai, 2018, Belkin et al., 2018a]. Here we are interested in when and how $f_t(x_i)$ interpolates y_i , for $1 \leq i \leq n$.

Finally, we remark that in practice, extending the gradient flow results to the (1) positive step size GD, and (2) mini-batch stochastic GD, are standalone interesting research topics. The reasons are that the optimization is non-smooth for the ReLU activation and that the interplay between the batch size and step size is less transparent in non-convex problems.

2 Preliminaries and Summary

2.1 Notations

We use the boldface lower case \mathbf{x} to denote a random variable or vector. The normal letter x can either be a scalar or a vector when there is no confusion. The transpose of a matrix \mathbf{A} , resp. vector u is denoted by \mathbf{A}^T , resp. u^T . \mathbf{A}^+ denotes the MoorePenrose inverse. For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. We use $\mathbf{A}[i, j]$ to denote the i, j -th entry of a matrix. We denote $\mathbb{1}_{\mathcal{D}}$ as the indicator function of set \mathcal{D} . We call symmetric positive semidefinite functions $K(\cdot, \cdot), H(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ kernels, and use calligraphy letter \mathcal{K}, \mathcal{H} to denote Hilbert spaces. We use $\langle f, g \rangle_{\mu} = \int f(x)g(x)\mu(dx)$ to denote the inner product in L^2_{μ} (or $L^2(P_x)$). $\hat{\mu}$ denotes the empirical distribution for μ . Notation $\mathbf{E}_{\mathbf{x}}$ is the expectation w.r.t random variable \mathbf{x} , and $\mathbf{E}_{\mathbf{x}, \tilde{\mathbf{x}}} h(\mathbf{x}, \tilde{\mathbf{x}}) = \int \int h(x, \tilde{x})\mu(dx)\mu(d\tilde{x})$. For a signed measure $\rho = \rho_+ - \rho_-$ with the positive and negative parts, define $|\rho| = \rho_+ + \rho_-$.

2.2 Preliminaries

We use the signed measure ρ_t , defined by the neuron weights at training time t collectively, to construct a *dynamic RKHS*. The mathematical definition of ρ_t is deferred to Section 5.1 and 5.2 (specifically, Eqn. (5.8)). The stationary signed measure at $t \rightarrow \infty$ is denoted as ρ_{∞} . For completeness we walk through the construction of the dynamic kernel and RKHS with ρ_t . Define the linear operator $\mathcal{T} : L^2_{\mu}(x) \rightarrow L^2_{|\rho_t|}(\Theta)$, such that for any $f(x) \in L^2_{\mu}(x)$

$$(\mathcal{T}f)(\Theta) := \int f(x)\|\Theta\|\sigma(x^T\Theta)\mu(dx), \quad \forall \Theta \in \text{supp}(\rho_t).$$

One can define the adjoint operator $\mathcal{T}^* : L^2_{|\rho_t|}(\Theta) \rightarrow L^2_\mu(x)$, such that for $p(\Theta) \in L^2_{|\rho_t|}(\Theta)$,

$$(\mathcal{T}^*p)(x) := \int p(\Theta) \|\Theta\| \sigma(x^T \Theta) |\rho_t|(d\Theta).$$

Note that both \mathcal{T} and \mathcal{T}^* are compact operators under the finite total variation and compact support assumptions. For the finite neurons case (1.2), the operator is of finite rank. We define the compact integral operator $\mathcal{T}^*\mathcal{T}$ with the corresponding kernel

$$H_t(x, \tilde{x}) = \int \|\Theta\|^2 \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) |\rho_t|(d\Theta), \text{ and } (\mathcal{T}^*\mathcal{T}f)(x) := \int H_t(x, \tilde{x}) f(\tilde{x}) \mu(d\tilde{x}). \quad (2.1)$$

The dynamic RKHS \mathcal{H}_t can be readily constructed via H_t . Let the eigen decomposition of $\mathcal{T}^*\mathcal{T}$ be the countable sum $\mathcal{T}^*\mathcal{T} = \sum_{i=1}^E \lambda_i e_i e_i^*$. Here E can be a nonnegative integer or ∞ , and $\lambda_i > 0$. e_i without confusion can represent either an eigen function or a linear functional. Similarly, we have the singular value decomposition for $\mathcal{T} = \sum_{i=1}^E \sqrt{\lambda_i} t_i e_i^*$, and \mathcal{T}^* as well. For a detailed discussion, see e.g. [Casselman \[2014\]](#). Again, t_i is a function in $L^2_{|\rho_t|}(\Theta)$ or a linear functional. The RKHS can be specified as follows.

$$\mathcal{H}_t = \left\{ h \mid h(x) = \sum_i h_i e_i(x), \sum_i \frac{h_i^2}{\lambda_i} < \infty \right\}.$$

We refer to H_∞ as the stationary RKHS kernel, and \mathcal{H}_∞ as the stationary RKHS. One can view that the gradient flow training dynamics — on the parameters of NN — induces a sequence of functions $\{f_t : t \geq 0\}$ and dynamic RKHS $\{\mathcal{H}_t : t \geq 0\}$, indexed by the time t .

2.3 Organization and Summary

We will prove three results, which are summarized informally in this section (see also [Table 1](#)). We remark that [Theorems 3.1](#) and [3.2](#) are stated for the approximation problem. However, as done in [Corollary 3.1](#), by substituting \mathcal{P}, μ by the empirical counterparts, one can easily state the analog for the ERM problem. Recall $f_*(x) = \mathbf{E}[\mathbf{y} | \mathbf{x} = x]$.

[†]Whether the kernel is universal in the $m, n \rightarrow \infty$ case still depends on f_* and the data distribution P . See the simulations of [Maennel et al. \[2018\]](#).

Table 1: Nature of the results studied in this paper.

	finite neurons m	infinite neurons $m \rightarrow \infty$
finite sam- ples n	Interpolation (finite rank kernel, Thms. 3.1 , 3.2 & Prop. 4.1)	Interpolation (finite rank kernel, Thms. 3.1 , 3.2 & Prop. 4.1)
infinite samples $n \rightarrow \infty$	Approximation (finite rank kernel, Thms. 3.1 & 3.2)	Approximation (possibly universal kernel [†] , Thms. 3.1 & 3.2)

Gradient flow on NN converges to projection onto data-adaptive RKHS. Theorem [3.1](#) shows that as done in practice training NN with simple gradient flow, in the limit of any *local* stationarity, learns the adaptive representation, and performs the *global* least squares projection simultaneously. Define $f_\infty = \lim_{t \rightarrow \infty} f_t$ as the function computed by ReLU networks (defined in [\(1.2\)](#), or more generally in [\(5.9\)](#)) until any stationarity of the gradient flow dynamics (defined in [\(1.3\)](#), with the squared loss) for the population distribution $(\mathbf{x}, \mathbf{y}) \sim P$. Define the corresponding stationary RKHS $\mathcal{H}_\infty = \lim_{t \rightarrow \infty} \mathcal{H}_t$ (defined in [\(2.1\)](#)).

[Informal version of Thm. [3.1](#)] Consider $f_* \in L_\mu^2$, for any local stationarity of the gradient flow dynamics [\(1.3\)](#) on the weights of neural networks [\(1.2\)](#), the function computed by NN at stationarity f_∞ satisfies

$$f_\infty \in \arg \min_{g \in \mathcal{H}_\infty} \|f_* - g\|_{L_\mu^2}^2.$$

Representation benefits of data-adaptive RKHS. Theorem [3.2](#) illustrates the provable benefits of the learned data-adaptive representation/basis \mathcal{H}_∞ . We emphasize that \mathcal{H}_∞ , as obtained by training neural networks on the data $(\mathbf{x}, \mathbf{y}) \sim P$, depends on the data in an implicit way such that there are advantages of representing and approximating f_* .

[Informal version of Thm. [3.2](#)] Consider $f_* \in L_\mu^2$ and the same setup as Theorem [3.1](#). Decompose f_* into the function f_∞ computed by the neural network

and the residual Δ_∞

$$f_* = f_\infty + \Delta_\infty.$$

Then there is another RKHS (defined in (3.4)) $\mathcal{K}_\infty \supset \mathcal{H}_\infty$, such that

$$f_\infty \in \mathcal{H}_\infty, \quad \Delta_\infty \in \text{Ker}(\mathcal{K}_\infty) \subset \text{Ker}(\mathcal{H}_\infty),$$

with a gap in the spaces $\mathcal{H}_\infty \oplus \text{Ker}(\mathcal{K}_\infty) \neq L_\mu^2$.

Convergence to Ridgeless regression with adaptive kernels. Proposition 4.1 establishes that in the vanishing regularization $\lambda \rightarrow 0$ limit, the neural network function computed by gradient flow converges to the kernel ridgeless regression with an adaptive kernel (denoted as $\hat{f}_\infty^{\text{rkhs}}(x)$). Consider using the gradient flow on the weights of the neural network function $f_t(x) = \sum_{j=1}^m w_j(t) \sigma(x^T u_j(t))$, to solve the ℓ_2 -regularized ERM

$$\frac{1}{2n} \sum_{i=1}^n (y_i - f_t(x_i))^2 + \frac{\lambda}{2m} \sum_{j=1}^m [w_j(t)^2 + \|u_j(t)\|^2] \quad .$$

Denote the function computed by NN at any local stationarity of ERM as $\hat{f}^{\text{nn},\lambda}(x)$, we answer the extrapolation question at a new point x , with the generalization error discussed in Prop. 4.2. The result is extendable to the infinite neurons case.

[Informal version of Prop. 4.1] Consider only the bounded assumption on initialization that $|w_j^2(0) - \|u_j\|^2(0)| < \infty$ for all $1 \leq j \leq m$. At stationarity, denote the corresponding adaptive kernel as \hat{H}_∞^λ . The neural network function $\hat{f}_\infty^{\text{nn},\lambda}(x)$ has the following expression,

$$\lim_{\lambda \rightarrow 0} \hat{f}_\infty^{\text{nn},\lambda}(x) = \hat{H}_\infty(x, X) \hat{H}_\infty(X, X)^+ Y =: \hat{f}_\infty^{\text{rkhs}}(x) \text{ (ridgeless regression with kernel } \hat{H}_\infty).$$

3 Main Results: Benefits of Adaptive Representation

We formally state two main results of the paper, Theorem 3.1 and Theorem 3.2 below.

3.1 Gradient Flow, Projection and Adaptive RKHS

We study how the function f_t computed from gradient flow on NN represents f_* when reaching any stationarity, under the squared loss. Consider the gradient flow dynamics (5.12) reaching *any stationarity*. Assume that the corresponding signed measure in (5.8) satisfies $\text{TV}(\rho_\infty) < \infty$ with a compact support. The mathematical details about ρ_∞ are postponed to Section 5.2. We employ the notation ρ_∞ since reaching stationarity can be viewed as $t \rightarrow \infty$.

We would like to emphasize that this stationary signed measure ρ_∞ is *task adaptive*: it implicitly depends on the regression task f_* and the data distribution P , rather than being pre-specified by the researcher as in Bach [2017], Daniely et al. [2016], Cho and Saul [2009]. With the RKHS established in Section 2.2, we are ready to state the following theorem.

Theorem 3.1 (Approximation). *For any conditional mean $f_*(x) = \mathbf{E}[\mathbf{y}|\mathbf{x} = x] \in L_\mu^2$, consider solving the approximation problem (1.5), with the ReLU NN function f_t defined in (1.2) where $w_j(t)$ and $\theta_j(t)$ are the weights for $t \geq 0, 1 \leq j \leq m$. For any signed measure ρ_0 with $\text{TV}(\rho_0) < \infty$, consider the infinitesimal initialization weights $u_j(0) = \Theta_j/\sqrt{m}$, and $w_j(0) = \text{sgn}(\rho_0(\Theta_j))\|\Theta_j\|/\sqrt{m}$, with $\Theta_j \sim \rho_0$ sampled independently. When the training dynamics (1.3) reaches any stationarity, it defines a stationary signed measure $\rho_\infty^{(m)}$ (on the collective weights) with $\text{TV}(\rho_\infty^{(m)}) < \infty$, and a corresponding stationary RKHS \mathcal{H}_∞ with the kernel defined in Eqn. (2.1), such that:*

1. *the function computed by neural networks at stationarity has the form*

$$f_\infty(x) = \int \|\Theta\| \sigma(x^T \Theta) \rho_\infty^{(m)}(d\Theta) ; \quad (3.1)$$

2. *f_∞ is a global minimizer of approximating f_* within the RKHS \mathcal{H}_∞*

$$f_\infty \in \arg \min_{g \in \mathcal{H}_\infty} \|f_* - g\|_{L_\mu^2}^2 . \quad (3.2)$$

In addition, the same results extend to the infinite neurons case with $m \rightarrow \infty$ where the limit for $\rho_\infty^{(m)}$ can be defined in the weak sense.

Remark 3.1. The above theorem shows that $\lim_{t \rightarrow \infty} f_t$ obtained by training on two-layer weights over time until any stationarity, is the same as projecting f_* onto the stationary RKHS \mathcal{H}_∞ . The projection is the solution to the classic nonparametric least squares, had one known the adaptive representation \mathcal{H}_∞ beforehand. Conceptually, this is *distinct* from the theoretical framework in the current statistics and learning theory literature: we do not require the structural knowledge about f_* (say, smoothness, sparsity, reflected in \mathcal{F}). Instead, we run gradient descent on neural networks to learn an adaptive representation for f_* , and show how the computed function represents f_* in this adaptive RKHS \mathcal{H}_∞ .

In other words, as done in practice training NN with simple gradient flow, in the limit of any *local* stationarity, learns the adaptive representation, and performs the *global* least-squares projection simultaneously. Training NN is learning a dynamic representation (quantified by \mathcal{H}_t), at the same time updating the predicted function f_t , as shown in Fig. 1.

A final note on the infinite neuron case: for any fixed time t , with the proper random initialization, setting $m \rightarrow \infty$ defines a proper distribution dynamics on the weak limit ρ_t shown in Lemma 5.3. Then set $t \rightarrow \infty$ to obtain the stationarity RKHS \mathcal{H}_∞ .

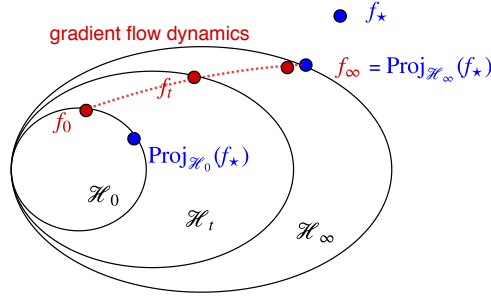


Figure 1: Illustration of Theorem 3.1. Red dotted line denotes the function f_t computed along the gradient flow dynamics on the weights of NN. Along training, one learns a sequence of dynamic RKHS representation \mathcal{H}_t 's. Over time, f_t converges to the projection of f_* onto \mathcal{H}_∞ . We emphasize that the initial function f_0 computed by NN is very different from the projection of f_* onto the initial RKHS \mathcal{H}_0 .

From the above, we have the following natural decomposition,

$$\Delta_\infty(x) = f_*(x) - f_\infty(x) \in \text{Ker}(\mathcal{H}_\infty). \quad (3.3)$$

Surprisingly, as we show in the next section, Δ_∞ actually lies in a smaller subspace of $\text{Ker}(\mathcal{H}_\infty)$, characterized by $\text{Ker}(\mathcal{K}_\infty)$. We call this the *representation and approximation benefits* of the data-adaptive RKHS learned by training neural networks.

Before moving next, we briefly discuss the above theorem when applied to the empirical measure, to solve the ERM problem. First, as a direct corollary, the following holds.

Corollary 3.1 (ERM). *Consider the ERM problem (1.6), with the other settings the same as in Theorem 3.1. One can define the finite dimensional RKHS $\hat{\mathcal{H}}_\infty$ (at most rank n) as in (2.1) with $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$ substituting μ . When reaches any stationarity, the solution satisfies*

$$\hat{f}_\infty \in \arg \min_{g \in \hat{\mathcal{H}}_\infty} \frac{1}{n} \sum_{i=1}^n (y_i - g(x_i))^2 .$$

More importantly, we will show in Proposition 4.1 that the function computed by training neural networks with gradient descent on the empirical risk objective $\hat{f}_\infty(x)$ until any stationarity (with vanishing ℓ_2 regularization), can be shown to be the **kernel ridgeless regression** with the **data-adaptive RKHS** $\hat{\mathcal{H}}_\infty$. Hence, studying the out of sample performance for GD on NN reduces to the generalization of kernel ridgeless regression with adaptive kernels.

3.2 Representation Benefits of Adaptive RKHS

We now define another adaptive RKHS \mathcal{K}_∞ named as the GD kernel, which turns out to be different from \mathcal{H}_∞ in (2.1). Interestingly, the difference in these two kernels sheds light on the representation benefits of the adaptive RKHS. The new RKHS \mathcal{K}_∞ is motivated by the gradient training dynamics. Recall the associated signed measure ρ_∞ at the stationarity, The GD kernel is defined as

$$K_\infty(x, \tilde{x}) = \int \left(\|\Theta\|^2 \mathbb{1}_{x^T \Theta \geq 0} \mathbb{1}_{\tilde{x}^T \Theta \geq 0} x^T \tilde{x} + \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \right) |\rho_\infty|(d\Theta) \neq H_\infty(x, \tilde{x}) \quad (3.4)$$

which is different than the stationary RKHS kernel H_∞ in (2.1). We use $\mathcal{K}_t : L_\mu^2(x) \rightarrow L_\mu^2(x)$ to denote the integral operator associated with K_t ,

$$(\mathcal{K}_t f)(x) := \int K_t(x, \tilde{x}) f(\tilde{x}) \mu(d\tilde{x}).$$

With a slight abuse of notation, we denote the corresponding RKHS to be \mathcal{K}_t as well. Now we are ready to state the main theorem on the representation benefits.

Theorem 3.2 (Representation Benefits). *Consider $f_* \in L_\mu^2$ and the same setting as in Theorem 3.1. Consider the approximation problem (1.5) with either finite or infinite neurons, and the gradient flow dynamics (5.12) (equivalently (1.3)) with data pair $(\mathbf{x}, \mathbf{y}) \sim P$ drawn from the population distribution. When reaching any stationary signed measure ρ_∞ , f_* is decomposed into the function f_∞ computed by the neural network and the residual Δ_∞*

$$f_* = f_\infty + \Delta_\infty.$$

Recall the RKHS \mathcal{H}_∞ in (2.1) and the GD RKHS \mathcal{K}_∞ in (3.4), all learned from the data $(\mathbf{x}, \mathbf{y}) \sim P$ and f_* adaptively. The following holds,

$$f_\infty \in \mathcal{H}_\infty, \quad \Delta_\infty \in \text{Ker}(\mathcal{K}_\infty) \subset \text{Ker}(\mathcal{H}_\infty),$$

with $\mathcal{H}_\infty \oplus \text{Ker}(\mathcal{K}_\infty) \neq L_\mu^2$. In other words, GD on NN decomposes f_* into two parts, and each lies in a space that is NOT the orthogonal complement to the other.

Remark 3.2. As we can see $\text{Ker}(\mathcal{K}_\infty)$ and $\text{Ker}(H_\infty)$ are not the same. Therefore, the decomposition $f_\infty + \Delta_\infty$ is not a trivial orthogonal decomposition to the RKHS \mathcal{H}_∞ and its complement.

Recall Theorem 3.1, projecting f_* to the RKHS \mathcal{H}_∞ with the data-adaptive kernel

$$H_\infty(x, \tilde{x}) = \int \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) |\rho_\infty|(d\Theta)$$

associated with $|\rho_\infty|$ is the same as the function constructed by neural networks (GD limit as $t \rightarrow \infty$). However, the residual lies in a possibly much smaller space due to Theorem 3.2, which is the null space of the RKHS \mathcal{K}_∞

$$K_\infty(x, \tilde{x}) = \int \left(\|\Theta\|^2 \mathbb{1}_{x^T \Theta \geq 0} \mathbb{1}_{\tilde{x}^T \Theta \geq 0} x^T \tilde{x} + \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \right) |\rho_\infty|(d\Theta).$$

In other words, as the learned adaptive basis \mathcal{H}_∞ (from GD) depends on the data distribution and the task f_* implicitly, it has the advantage of representing f_* by squeezing the residual into a smaller subspace in the null space of \mathcal{H}_∞ . A pictural illustration can be found in Fig. 2. This representation and approximation benefit helps with explaining the better interpolation results obtained by neural networks [Zhang et al., 2016, Belkin et al., 2018b, Liang and Rakhlin, 2018, Belkin et al., 2018a]: (1) the adaptive basis is tailored for the task f_* , thus the residual/interpolation error lies in a smaller space; (2) in view of the ODE in Corollary 5.2, the second layer of NN adds *implicit regularization* to the smallest eigenvalues of K_t , thus improving the converging speed of Δ_t to zero.

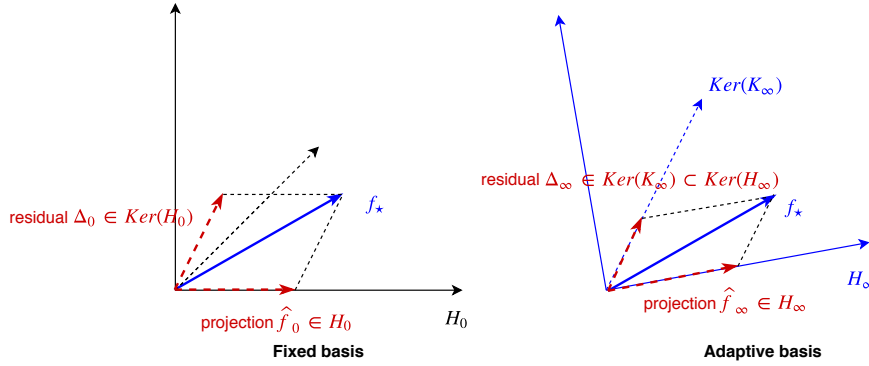


Figure 2: Illustration of 3.2: fixed basis vs. adaptive learned basis. In classic statistics, one specifies the fixed function space/basis H_0 then decompose f_* into the projection \hat{f}_0 and residual $\Delta_0 \in \text{Ker}(H_0)$. However, for GD on NN, one learns the adaptive basis H_∞ that depends on f_* . Therefore, the residual Δ_∞ lies in a subspace of $\text{Ker}(H_\infty)$.

Before concluding this section, we remark that a similar result holds for the ERM problem (1.6). As we shall discuss in the next section, the gap between \mathcal{H}_∞ and \mathcal{K}_∞ can be large, even for the ERM problem.

4 Implications of the Adaptive Theory

In this section, we will discuss some direct implications of the adaptive kernel theory for neural networks established in this paper.

Example: Gap in Spaces \mathcal{H}_∞ and \mathcal{K}_∞ . In Theorem 3.2, it is established that $\text{Ker}(\mathcal{K}_\infty) \subset \text{Ker}(\mathcal{H}_\infty)$. We now construct a concrete case to illustrate the potentially significant gap in these two spaces as follows. Consider only one neuron with $m = 1$, solving ERM problem (1.6) with n samples, and \mathbf{x} with dimension d . In this case, ρ_∞ is supported on only one point, noted as $\Theta_\infty \in \mathbb{R}^d$. Denote $X \in \mathbb{R}^{n \times d}$ as the data matrix, one can show that

$$H_\infty(X, X) = \underbrace{\sigma(X\Theta_\infty^T)}_{n \times 1} \underbrace{\sigma(X\Theta_\infty^T)^T}_{1 \times n}$$

has rank 1. In contrast,

$$K_\infty(X, X) \geq \underbrace{\text{diag}(\mathbb{1}_{X\Theta_\infty^T \geq 0})X}_{n \times d} \underbrace{X^T \text{diag}(\mathbb{1}_{X\Theta_\infty^T \geq 0})}_{d \times n}$$

can be of rank $d \wedge |\{i : x_i^T \Theta_\infty \geq 0\}|$. Hence the null space of K_∞ is much smaller than that of H_∞ . The gap can be large for many other settings of (n, m, d) .

Connections to Min-norm Interpolation. The following result establishes the connections between the solution of gradient descent on neural networks (at local stationarity), and the kernel ridgeless regression [Belkin et al., 2018b, Liang and Rakhlin, 2018, Hastie et al., 2019] with an adaptive kernel \hat{H}_∞^λ . Empirical evidence on the similarity between the interpolation with kernels and neural networks was discovered in Belkin et al. [2018b]. The following proposition provides a novel way of studying the generalization property of neural networks via adaptive kernels.

Proposition 4.1 (Interpolation: Connection to Kernel Ridgeless Regression). *Consider the gradient flow dynamics on all the weights of the neural network function $f_t(x) = \sum_{j=1}^m w_j(t)\sigma(x^T u_j(t))$,*

to solve the ℓ_2 -regularized ERM

$$\frac{1}{2n} \sum_{i=1}^n (y_i - f_t(x_i))^2 + \frac{\lambda}{2m} \sum_{j=1}^m [w_j(t)^2 + \|u_j(t)\|^2] \quad .$$

Consider only the bounded assumption on initialization that $|w_j^2(0) - \|u_j\|^2(0)| < \infty$ for all $1 \leq j \leq m$. At stationarity, denote the signed measure as $\hat{\rho}_\infty^\lambda$ and the corresponding adaptive kernel as \hat{H}_∞^λ . Then the neural network function at stationarity $\hat{f}_\infty^{\text{nn},\lambda}(x)$ satisfies,

$$\hat{f}_\infty^{\text{nn},\lambda}(x) = \hat{H}_\infty^\lambda(x, X) \left[\frac{n}{m} \lambda \cdot I_n + \hat{H}_\infty^\lambda(X, X) \right]^{-1} Y \quad .$$

In the vanishing regularization $\lambda \rightarrow 0$ limit, the neural network function converges to the kernel ridgeless regression with the adaptive kernel, when $\hat{H}_\infty(X, X) := \lim_{\lambda \rightarrow 0} \hat{H}_\infty^\lambda$ exists,

$$\lim_{\lambda \rightarrow 0} \hat{f}_\infty^{\text{nn},\lambda}(x) = \hat{H}_\infty(x, X) \hat{H}_\infty(X, X)^+ Y = \hat{f}_\infty^{\text{rkhs}}(x).$$

Note that the generalization theory for the kernel ridgeless regression has been established [Liang and Rakhlin \[2018\]](#), [Hastie et al. \[2019\]](#). Here the kernel $\hat{H}_\infty(X, X)$ is data-adaptive (that adapts to f_*) learned along training, instead of being fixed and pre-specified.

Connections to Random Kitchen Sinks. Let us introduce two function spaces, with the base measure ρ_0 (fixed representation)

$$\Gamma_2(\rho_0) := \left\{ f(x) \mid f(x) = \int \sigma(x^T \Theta) w(\Theta) \rho_0(d\Theta), w \in L_{\rho_0}^2 \right\}$$

$$\Gamma_1(\rho_0) := \left\{ f(x) \mid f(x) = \int \sigma(x^T \Theta) w(\Theta) \rho_0(d\Theta), w \in L_{\rho_0}^1 \right\}$$

In random kitchen sinks studied in [Rahimi and Recht \[2008, 2009\]](#), by assuming $f_* \in \Gamma_2(\rho_0)$ that lies in the RKHS, the approximation error can be controlled by the existence of the following function with $\theta_j, j \in [m]$ i.i.d. sampled from ρ_0

$$\hat{f}(x) = \frac{1}{m} \sum_{j=1}^m \sigma(x^T \Theta_j) w(\Theta_j) \in \Gamma_1(\rho_0), \text{ but } \hat{f}(x) \notin \Gamma_2(\rho_0) \quad .$$

Note that \hat{f} lies in a possibly much larger space $\Gamma_1(\rho_0)$ though the target only lies in $f_* \in \Gamma_2(\rho_0)$. Similarly for two-layer neural networks function $f_t(x)$ considered in [Bach, 2017, Section 2.3], the RKHS space $\Gamma_2(\rho_0)$ can be more restrictive compared to $f_t \in \Gamma_1(\rho_0)$.

In contrast, with the adaptive RKHS representation \mathcal{H}_∞ , we have shown that

$$f_\infty(x) \in \Gamma_1(|\rho_\infty|), \text{ and } f_\infty(x) \in \Gamma_2(|\rho_\infty|) .$$

The extreme case of fully adaptive function space $\Gamma_2(|\rho_*|)$ is defined with ρ_* tailored for f_* , $f_* = \int \sigma(x^T \Theta) \rho_*(d\Theta)$. The adaptive representation learned by neural networks can be viewed as in between the fixed and the fully adaptive representation.

Adaptive Generalization Theory. Now we attempt to provide a new decomposition to study the generalization of NN via adaptive kernels. Recall we have shown that $\hat{f}_\infty^{\text{rkhs}}(x) = \lim_{\lambda \rightarrow 0} \hat{f}_\infty^{\text{nn}, \lambda}(x) = \hat{H}_\infty(x, X) \hat{H}_\infty(X, X)^+ Y$, where $\hat{H}_\infty(x, \tilde{x}) := \int \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \hat{\rho}_\infty^{(n, m)}(d\Theta)$. Define the population limit $\rho_\infty^{(m)}(d\Theta) := \lim_{n \rightarrow \infty} \hat{\rho}_\infty^{(n, m)}$ and $H_\infty(x, \tilde{x}) := \int \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \rho_\infty^{(m)}(d\Theta)$. Denote the ridgeless regression with the population adaptive kernel H_∞ ,

$$f_\infty^{\text{rkhs}}(x) = H_\infty(x, X) H_\infty(X, X)^+ Y.$$

Assume $(\mathbf{y} - f_*(\mathbf{x}))^2 \leq \sigma^2$ a.s. (can be relaxed). One can derive the following decomposition for generalization.

Proposition 4.2 (Adaptive Generalization).

$$\begin{aligned} \left\| \lim_{\lambda \rightarrow 0} \hat{f}_\infty^{\text{nn}, \lambda} - f_* \right\|_\mu^2 &\lesssim \underbrace{\left\| \hat{f}_\infty^{\text{rkhs}} - f_\infty^{\text{rkhs}} \right\|_\mu^2}_{\text{adaptive representation error}} + \underbrace{\left\| f_\infty - f_* \right\|_\mu^2}_{\text{adaptive approximation error}} \\ &\quad + (n \left\| f_\infty - f_* \right\|_\mu^2 + \sigma^2) \underbrace{\mathbf{E}_{\mathbf{x} \sim \mu} \left\| H_\infty(X, X)^{-1} H_\infty(X, \mathbf{x}) \right\|^2}_{\text{adaptive variance}} \\ &\quad + \underbrace{\left\| H_\infty(x, X) H_\infty(X, X)^{-1} f_\infty(X) - f_\infty(x) \right\|_\mu^2}_{\text{adaptive bias}} \end{aligned}$$

Note this result holds without requiring global optimization guarantees. The first term is the representation error, which corresponds to the closeness of the adaptive RKHS $\hat{\mathcal{H}}_\infty$ (using

empirical distribution) and \mathcal{H}_∞ (using population distribution). The second term is the adaptive approximation error studied in the current paper. The third and fourth terms are the variance and bias expressions studied in [Liang and Rakhlin \[2018\]](#), [Hastie et al. \[2019\]](#), [Rakhlin and Zhai \[2018\]](#), [Bartlett et al. \[2019\]](#), [Liang et al. \[2020\]](#), as if assuming the actual function lies in \mathcal{H}_∞ . This decomposition suggests the possibility of studying generalization without explicit global understanding of the optimization, and providing rates that adapts to f_* without structural assumptions.

5 Time-varying Kernels and Evolution

In this section, we lay out the mathematical details on the time-varying kernels and the evolution of the signed measure ρ_t supporting the main results. In the meantime, we will discuss in depth the relevant literature motivating our proof ideas.

First, we describe the motivation behind the dynamic RKHS \mathcal{K}_t , and the GD kernel induced by the gradient descent dynamics. Extensions to the multi-layer perceptrons will be in the [Appendix 1.2](#).

Lemma 5.1 (Dynamic kernel of finite neurons GD). *Consider the approximation problem (1.1) with a neural network function (1.2), and the training process (1.3) with population distribution. Let $\Delta_t(x) = f_*(x) - f_t(x)$ be the residual. Define the time-varying kernel $K_t(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$,*

$$K_t(x, \tilde{x}) = \sum_{j=1}^m \left[\sigma(x^T u_j(t)) \sigma(\tilde{x}^T u_j(t)) + w_j(t)^2 \mathbb{1}_{x^T u_j(t) \geq 0} \mathbb{1}_{\tilde{x}^T u_j(t) \geq 0} x^T \tilde{x} \right]. \quad (5.1)$$

Then the residual Δ_t driven by the GD dynamics satisfies,

$$\frac{d\mathbf{E}_{\mathbf{x}} \left[\frac{1}{2} \Delta_t(\mathbf{x})^2 \right]}{dt} = -\mathbf{E}_{\mathbf{x}, \tilde{\mathbf{x}}} \left[\Delta_t(\mathbf{x}) K_t(\mathbf{x}, \tilde{\mathbf{x}}) \Delta_t(\tilde{\mathbf{x}}) \right]. \quad (5.2)$$

When running GD to solve the empirical risk minimization (ERM), the dynamics of the finite-dimensional sample residual $\|\Delta_t\|_{\hat{\mu}}^2$ has been established in [Jacot et al. \[2018\]](#), [Du et al. \[2018\]](#). Here we generalize the result to optimize the weights of both layers, and to solve

the infinite-dimensional population approximation problem rather than the empirical risk minimization problem. For a general loss function $\ell(y, f)$ with curvature (say, logistic loss), similar results hold under slightly stronger conditions.

Corollary 5.1. *Consider a general loss function $\ell(y, f)$ that is α -strongly convex in the second argument f , with K_t defined in (5.1). Assume in addition $\frac{1}{n}K_t(X, X) \in \mathbb{R}^{n \times n}$ has smallest eigenvalue $\lambda_t > 0$. Define $\Delta_t(x_i) := \frac{\partial \ell(y_i, f_t(x_i))}{\partial f}$, then we have for all $f_* : \mathbb{R}^d \rightarrow \mathbb{R}$,*

$$\frac{d\hat{\mathbf{E}}[\ell(\mathbf{y}, f_t(\mathbf{x}))]}{dt} = -\hat{\mathbf{E}}_{\mathbf{x}, \tilde{\mathbf{x}}} [\Delta_t(\mathbf{x})K_t(\mathbf{x}, \tilde{\mathbf{x}})\Delta_t(\tilde{\mathbf{x}})] \leq -2\alpha\lambda_t \cdot \hat{\mathbf{E}}[\ell(\mathbf{y}, f_t(\mathbf{x})) - \ell(\mathbf{y}, f_*(\mathbf{x}))] .$$

5.1 Initialization, Rescaling and K_0

Now we describe the initialization and rescaling schemes used in the main theorems. Rewrite (1.1) according to the signs of the second layer weights

$$f_t(x) := \sum_{j=1}^{m_+} w_{+,j}(t)\sigma(x^T u_{+,j}(t)) + \sum_{j=1}^{m_-} w_{-,j}(t)\sigma(x^T u_{-,j}(t)).$$

Initialization. We consider the “infinitesimal” initialization drawn *i.i.d.* from two probability measures $\rho_{+,0}$ and $\rho_{-,0}$ that do not depend on m :

$$u_{+,j}(0) = \frac{1}{\sqrt{m}}\Theta_{+,j} \text{ where } \Theta_{+,j} \sim \rho_{+,0} , \quad u_{-,j}(0) = \frac{1}{\sqrt{m}}\Theta_{-,j} \text{ where } \Theta_{-,j} \sim \rho_{-,0} . \quad (5.3)$$

Here $m = m_+ + m_-$ with $m_+ \asymp m_-$. The $1/\sqrt{m}$ rescaling factor turns out to be crucial when defining the infinite neurons limit for the evolution of signed measures. Remark that such initialization is w.l.o.g., and accounts for the infinitesimal nature used in practice when the number of neurons grows. For the second layer weights, we impose the “balanced condition” motivated by Maennel et al. [2018],

$$w_{+,j}(0) = \|u_{+,j}(0)\| \geq 0 , \quad w_{-,j}(0) = -\|u_{-,j}(0)\| \leq 0. \quad (5.4)$$

It turns out that with such initialization, the balanced condition holds throughout the training process induced by gradient flow, which is useful for the main theorems. Interestingly, in

the proof of Proposition 4.1, we show that such balanced condition always holds at stationarity when training neural networks with ℓ_2 regularization, even for unbalanced initialization.

Proposition 5.1 (Balanced condition). *For $u_{+,j}(t)$, $u_{-,j}(t)$, $w_{+,j}(t)$ and $w_{-,j}(t)$, and the initialization specified above, at any time t , we have*

$$w_{+,j}(t) = \|u_{+,j}(t)\|, \quad w_{-,j}(t) = -\|u_{-,j}(t)\|.$$

Rescaling. To prepare for the distribution dynamic theory in the next section, we introduce a parameter rescaling with the \sqrt{m} factor. Let $\theta_{+,j}(t) = \sqrt{m}w_{+,j}(t)$ and $\theta_{-,j}(t) = \sqrt{m}w_{-,j}(t)$, also define $\Theta_{+,j}(t) = \sqrt{m}u_{+,j}(t)$ and $\Theta_{-,j}(t) = \sqrt{m}u_{-,j}(t)$ sampled from $\rho_{+,0}$ and $\rho_{-,0}$ at $t = 0$. Under this representation,

$$f_t(x) = \frac{1}{m} \sum_{j=1}^{m_+} \theta_{+,j}(t) \sigma(x^T \Theta_{+,j}(t)) + \frac{1}{m} \sum_{j=1}^m \theta_{-,j}(t) \sigma(x^T \Theta_{-,j}(t)). \quad (5.5)$$

By the positive homogeneity of ReLU, we have the corresponding dynamics on the rescaled parameters,

$$\frac{d\theta_{+,j}}{dt} = \sqrt{m} \frac{dw_{+,j}}{dt} = -\sqrt{m} \mathbf{E}_{\mathbf{z}} \left[\frac{\partial \ell(\mathbf{y}, f(\mathbf{x}))}{\partial f} \sigma(\mathbf{x}^T u_{+,j}) \right] = -\mathbf{E}_{\mathbf{z}} \left[\frac{\partial \ell(\mathbf{y}, f(\mathbf{x}))}{\partial f} \sigma(\mathbf{x}^T \Theta_{+,j}) \right], \quad (5.6)$$

$$\frac{d\theta_{-,j}}{dt} = \sqrt{m} \frac{dw_{-,j}}{dt} = -\sqrt{m} \mathbf{E}_{\mathbf{z}} \left[\frac{\partial \ell(\mathbf{y}, f(\mathbf{x}))}{\partial f} w_{-,j} \mathbb{1}_{\mathbf{x}^T u_{-,j} \geq 0} \mathbf{x} \right] = -\mathbf{E}_{\mathbf{z}} \left[\frac{\partial \ell(\mathbf{y}, f(\mathbf{x}))}{\partial f} \theta_{-,j} \mathbb{1}_{\mathbf{x}^T \Theta_{-,j} \geq 0} \mathbf{x} \right]. \quad (5.7)$$

Define at time t

$$\rho_{+,t} := \frac{1}{m} \sum_{j=1}^{m_+} \delta_{\Theta_{+,j}(t)}, \quad \rho_{-,t} := \frac{1}{m} \sum_{j=1}^{m_-} \delta_{\Theta_{-,j}(t)} \quad (5.8)$$

as the empirical distribution over neurons on the parameter space Θ . The $\rho_{+,t}$ and $\rho_{-,t}$ converge weakly to proper distributions in the infinite neurons limit $m \rightarrow \infty$, see e.g. Bach [2017], Mei et al. [2018]. Through the balanced condition in Proposition 5.1 and Proposition 1.1 (see Appendix), we know by substituting θ_j with $\|\Theta_j\|$

$$f_t(x) = \int \|\Theta\| \sigma(x^T \Theta) \rho_t(d\Theta), \quad \text{where the signed measure } \rho_t := \rho_{+,t} - \rho_{-,t}. \quad (5.9)$$

The above motivates the study of the RKHS \mathcal{H}_t as in Theorem 3.1, with the kernel

$$H_t(x, \tilde{x}) = \int \|\Theta\|^2 \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) |\rho_t|(d\Theta). \quad (5.10)$$

To conclude this section, we provide the explicit formula for the initial kernel matrix K_0 under such infinitesimal random initialization. Specifically, consider the initialization with w_j being $\pm 1/\sqrt{m}$ with equal chance and $u_i \sim N(\mathbf{0}, 1/m \cdot \mathbf{I}_d)$ *i.i.d.* sampled. The initial kernel K_0 has the following expression, in the infinite neurons limit.

Lemma 5.2 (Fixed Kernel). *With initialization specified above, consider w.l.o.g. $\|x\| = \|\tilde{x}\| = 1$, and denote $\Theta \sim \pi$ as the isotropic Gaussian $N(\mathbf{0}, \mathbf{I}_d)$. By the strong law of large number, we have almost surely,*

$$\begin{aligned} \lim_{m \rightarrow \infty} K_0(x, \tilde{x}) &= \mathbf{E}_{\Theta \sim \pi} \left[\sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) + \mathbb{1}_{x^T \Theta > 0} \mathbb{1}_{\tilde{x}^T \Theta > 0} x^T \tilde{x} \right] \\ &= \left[\frac{\pi - \arccos(t)}{\pi} t + \frac{\sqrt{1 - t^2}}{2\pi} \right], \quad \text{where } t = x^T \tilde{x}. \end{aligned}$$

Much known results [Bengio et al., 2006, Rahimi and Recht, 2008, Bach, 2017, Cho and Saul, 2009, Daniely et al., 2016] on the connection between RKHS and two-layer NN focus on some fixed kernel, such as K_0 . To instantiate useful statistical rates, one requires f_* to lie in the corresponding pre-specified RKHS \mathcal{K}_0 , which is non-verifiable in practice. In contrast, the dynamic kernel is less studied. We will establish a dynamic and adaptive kernel theory defined by GD, without making any structural assumptions on f_* other than $f_* \in L_\mu^2$.

5.2 Evolution of ρ_t

In this section, we derive the evolution of the signed measure ρ_t defined by the neurons at the training t , which in turn determines the dynamic kernel K_t defined in (5.1). To generalize the result to the case of infinite neurons, we follow and borrow tools from the mean-field characterization [Mei et al., 2018, Rotskoff and Vanden-Eijnden, 2018, Jordan et al., 1998]. The rescaling described in the previous section proves handy when defining such infinite

neurons limit. We define the velocity field driven by the regression task and the interaction among neurons,

$$V(\Theta) = \mathbf{E}[\mathbf{y}\sigma(\mathbf{x}^T\Theta)], \quad U(\Theta, \tilde{\Theta}) = -\mathbf{E}[\sigma(\mathbf{x}^T\Theta)\sigma(\mathbf{x}^T\tilde{\Theta})]. \quad (5.11)$$

The following theorem casts the training process as distribution dynamics on $\rho_{+,t}, \rho_{-,t}$.

Lemma 5.3 (Dynamic Kernel and Evolution). *Consider the approximation problem (1.1), and the gradient flow as the training dynamic (1.3). For $\rho_{+,t}, \rho_{-,t}$ and ρ_t defined in (5.8) with possibly infinite neurons, we have the following PDE characterization on distribution dynamics of $\rho_{+,t}, \rho_{-,t}$*

$$\begin{aligned} \partial_t \rho_{+,t}(\Theta) &= -\nabla_{\Theta} \cdot \left[\rho_{+,t}(\Theta) \cdot \|\Theta\| \left(\nabla_{\Theta} V(\Theta) + \nabla_{\Theta} \int U(\Theta, \tilde{\Theta}) \|\tilde{\Theta}\| \rho_t(d\tilde{\Theta}) \right) \right], \\ \partial_t \rho_{-,t}(\Theta) &= \nabla_{\Theta} \cdot \left[\rho_{-,t}(\Theta) \cdot \|\Theta\| \left(\nabla_{\Theta} V(\Theta) + \nabla_{\Theta} \int U(\Theta, \tilde{\Theta}) \|\tilde{\Theta}\| \rho_t(d\tilde{\Theta}) \right) \right]. \end{aligned} \quad (5.12)$$

Moreover, the GD kernel K_t is defined as

$$K_t(x, \tilde{x}) = \int \left(\|\Theta\|^2 \mathbb{1}_{x^T\Theta \geq 0} \mathbb{1}_{\tilde{x}^T\Theta \geq 0} x^T \tilde{x} + \sigma(x^T\Theta) \sigma(\tilde{x}^T\Theta) \right) |\rho_t|(d\Theta). \quad (5.13)$$

Remark 5.1. As in Mei et al. [2018], Rotskoff and Vanden-Eijnden [2018], let's first show that in the infinite neurons limit $m \rightarrow \infty$, $\rho_{+,t}, \rho_{-,t}$ are properly defined, with Eqn. (5.12) characterizing the distribution dynamics. For simplicity, we assume the initialization $\rho_{+,0}, \rho_{-,0}$ is with bounded support. Add the superscript m , $\rho_{+,t}^{(m)}, \rho_{-,t}^{(m)}, \rho_t^{(m)}$ to (5.8) to indicate their dependence on m . Consider that $\nabla_{\Theta} V(\Theta), \nabla_{\Theta} U(\Theta, \tilde{\Theta})$ in (5.11) are bounded and uniform Lipchitz continuous as in [Mei et al., 2018, A3]. With the same proof as in [Mei et al., 2018, Theorem 3], one can show that with $m \rightarrow \infty$, the initial distribution $\rho_0^{(m)} \xrightarrow{d} \rho_0 = \rho_{+,0} - \rho_{-,0}$ by law of large number. And by the solution's continuity w.r.t. the initial value, we have $\rho_t^{(m)} \xrightarrow{d} \rho_t$ as $m \rightarrow \infty$ well defined, for any fixed t .

Note that our problem setting is slightly different from that in Mei et al. [2018], where the authors consider the NN with fixed second layer weights to be $1/m$. We reiterate that the re-parameterization via θ and Θ is crucial: (1) weights on both layers are optimized

following the gradient flow; (2) infinitesimal random initialization is employed in practice. In the setting of [Mei et al., 2018, Eqn. (3)], the training process is slightly different from the vanilla GD on weights, with an additional m factor in the velocity term. This subtlety is also mentioned in Rotskoff and Vanden-Eijnden [2018]. In short, the rescaling looks at the dynamics where Θ 's are on the invariant scale as $m \rightarrow \infty$ for any fixed effective time t (that does not depend on m). Here we analyze the exact gradient flow on the two-layer weights, with infinitesimal random initialization as in practice, resulting in a different velocity field (5.11) compared to that in Mei et al. [2018].

The proof of Theorem 3.1 makes use of (5.9)-(5.10) and the stationary condition implied by Lemma 5.3. The balanced condition is crucial in both Theorem 3.1 and Proposition 4.1. The details of the proof are deferred to Section 7.

5.3 Two RKHS: \mathcal{K}_∞ and \mathcal{H}_∞

In this section we compare the two adaptive RKHS appeared \mathcal{K}_∞ in (5.13), and \mathcal{H}_∞ in (5.10). The comparison will lead to the proof of Theorem 3.2. We start with generalizing Lemma 5.1 with the possibly infinite neurons case via the distribution dynamics in (5.12).

Corollary 5.2. *Consider the same setting as in Lemma 5.1 with possibly infinite neurons NN (5.9), and the training process (5.12). Define the time-varying kernel matrix $K_t(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, with the signed measure ρ_t follows (5.12)*

$$K_t(x, \tilde{x}) = \int \left(\|\Theta\|^2 \mathbb{1}_{x^T \Theta \geq 0} \mathbb{1}_{\tilde{x}^T \Theta \geq 0} x^T \tilde{x} + \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \right) |\rho_t|(d\Theta) \quad (5.14)$$

$$=: K_t^{(0)}(x, \tilde{x}) + K_t^{(1)}(x, \tilde{x}). \quad (5.15)$$

Then we still have $d\mathbf{E}_{\mathbf{x}} \left[\frac{1}{2} \Delta_t(\mathbf{x})^2 \right] / dt = -\mathbf{E}_{\mathbf{x}, \tilde{\mathbf{x}}} \left[\Delta_t(\mathbf{x}) K_t(\mathbf{x}, \tilde{\mathbf{x}}) \Delta_t(\tilde{\mathbf{x}}) \right]$.

It turns out that the kernels K_∞ and H_∞ , defined in (3.4) and (2.1) respectively, satisfy the following inclusion property.

Proposition 5.2. *Consider the training process reaches any stationarity $\rho_\infty = \rho_{+, \infty} - \rho_{-, \infty}$ with compact support within radius D and finite total variation. We have*

$$K_\infty \geq K_\infty^{(0)} \geq K_\infty^{(1)} \geq \frac{1}{D^2} H_\infty, \quad (5.16)$$

with $K_\infty^{(0)}, K_\infty^{(1)}$ defined in (5.14). Combining with the fact that $H_\infty \neq K_\infty$ implies

$$\text{Ker}(\mathcal{K}_\infty) \subset \text{Ker}(\mathcal{H}_\infty).$$

The proof of Theorem 3.2 uses the following fact: when reaching stationarity, due to the ODE defined by GD in Lemma 5.1, the residual must satisfy

$$\Delta_\infty(x) = f_*(x) - f_\infty(x) \in \text{Ker}(\mathcal{K}_\infty). \quad (5.17)$$

The proof of Proposition 5.2 and Theorem 3.2 are deferred to Section 7.

6 Experiments

We run experiments to illustrate the spectral decay of the dynamic kernels defined in K_t over time t . The exercise is to quantitatively showcase that during neural network training, one does learn the data-adaptive representation, which is task-specific depending on the true complexity of f_* . The training process is the same as the one we theoretically analyze: vanilla gradient descent on a two-layer NN of m neurons, with infinitesimal random initialization scales as $1/\sqrt{m}$.

The first experiment is a synthetic exercise with well-specified models. We generate $\{x_i\}_{i=1}^{50}$ from isotropic Gaussian in \mathbb{R}^5 , and $y_i = f_*(x_i) = \sum_{j=1}^J w_j^* \sigma(x_i^T u_j^*)$ with different J . In other words, we choose different target f_* (task complexity) by varying J . We select $m = 500$ in our experiment. The top 80% of the sorted eigenvalues of the kernel matrix K_t along the GD training process are shown in Fig. 3. The x -axis is the index of eigenvalues in descending order, and the y -axis is the logarithmic values of the corresponding eigenvalues. Different color indicates the spectral decay of the K_t at different training time t . The eigenvalue-decays stabilize over time t means that the training process approaches stationarity. As we

can see with f_* belongs to the NN family, the eigenvalues of the kernel matrix, in general, become larger during the training process. For a more complicated target function, it takes longer to reach stationarity.

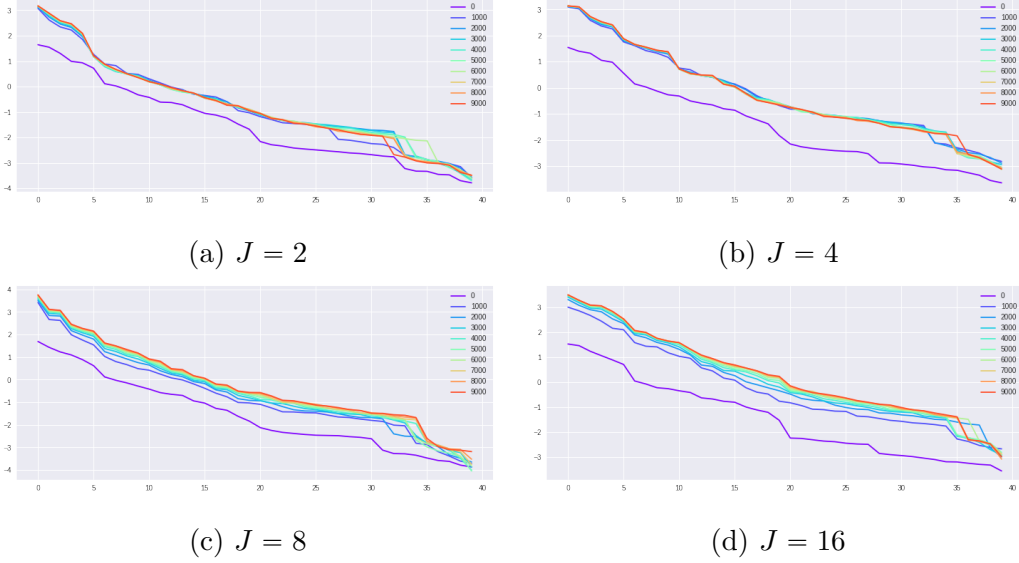


Figure 3: Log of the sorted top 80% eigenvalues of kernel matrix along training with different f_*

The second experiment is another synthetic test on fitting random labels. We generate $\{x_i\}_{i=1}^{50}$ from isotropic Gaussian in \mathbb{R}^5 , as y_i takes ± 1 with equal chance. We select $m = 200, 500$, and $n = 50, 200$ to investigate those parameters' influence on the kernel K_t . We want to point out two observations. First, fixed n , we investigate over-parametrized models ($m = 200, 500$ large). Shown from Fig. 4 along the row, the kernels for different m 's behave much alike. In other words, in the infinite neurons limit, the kernel will stabilize. Second, fixed m , we vary the number of samples n , to simulate different interpolation hardness. As seen from Fig. 4 along the column, the kernels and the convergence over time are distinct, reflecting the different difficulty of the interpolation.

The third experiment (Fig. 5) is regression using the MNIST dataset with different sample size $n = 50, 200$. We hope to investigate the influence of sample size on the kernel matrix along the training process. For a larger sample size N , it takes longer to reach stationarity.

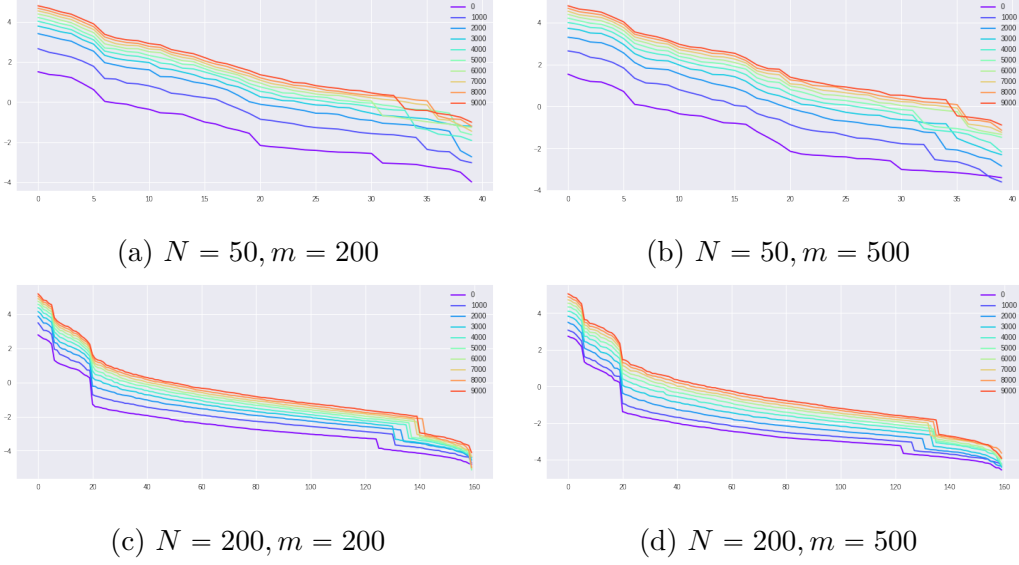


Figure 4: Log of the sorted top 80% eigenvalues of kernel matrix along training with random labels.

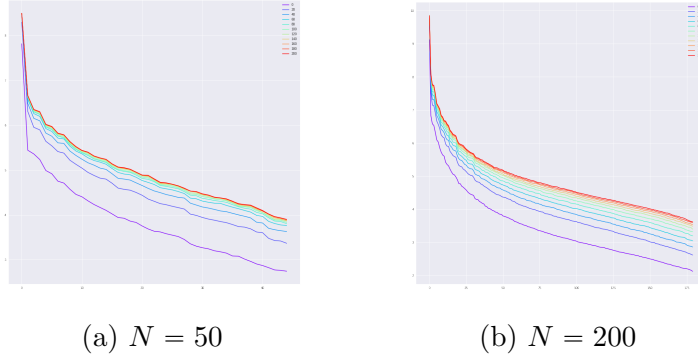


Figure 5: Log of sorted top 90% eigenvalues of kernel matrix along training process for mnist

7 Main Proofs

Proof of Theorem 3.1. From the definition, we have $\mathcal{T}^*p \in \mathcal{H}_\infty$ for any $p \in L^2_{|\rho_\infty|}$, and \mathcal{T}^* is a surjective mapping. Suppose that $\hat{g} \in \mathcal{H}_\infty$ is a minimizer of (3.2), then we claim that for any $p \in L^2_{|\rho_\infty|}$, one must have

$$\langle f_* - \hat{g}, \mathcal{T}^*p \rangle_\mu = 0, \quad \forall p \in L^2_{|\rho_\infty|}. \quad (7.1)$$

This claim can be seen from the following argument. Suppose not, then for p that violates the above, construct

$$\hat{g}_\epsilon = \hat{g} + \epsilon \mathcal{T}^* p \in \mathcal{H}_\infty,$$

we know

$$\|f_* - \hat{g}_\epsilon\|_\mu^2 = \|f_* - \hat{g}\|_\mu^2 - 2\epsilon \langle f_* - \hat{g}, \mathcal{T}^* p \rangle_\mu + \epsilon^2 \|\mathcal{T}^* p\|_\mu^2. \quad (7.2)$$

For ϵ with the same sign as $\langle f_* - \hat{g}, \mathcal{T}^* p \rangle_\mu \neq 0$ and small enough, one can see that $\|f_* - \hat{g}_\epsilon\|_\mu^2 < \|f_* - \hat{g}\|_\mu^2$ which validates that \hat{g} is a minimizer. From the same argument, one can see that \hat{g} is a minimizer if and only if (7.1) holds, in other words,

$$\langle \mathcal{T}(f_* - \hat{g}), p \rangle_{|\rho_\infty|} = \langle f_* - \hat{g}, \mathcal{T}^* p \rangle_\mu = 0 \quad (7.3)$$

From PDE characterization (5.12) with ReLU activation, one knows that

$$\begin{aligned} V(\Theta) &= \mathbf{E}[\mathbf{y} \sigma(\mathbf{x}^T \Theta)] = \mathbf{E}[f_*(\mathbf{x}) \sigma(\mathbf{x}^T \Theta)] \\ U(\Theta, \tilde{\Theta}) &= -\mathbf{E}[\sigma(\mathbf{x}^T \Theta) \sigma(\mathbf{x}^T \tilde{\Theta})], \end{aligned}$$

and the expression for the velocity field

$$\begin{aligned} &\|\Theta\| \left(\nabla_\Theta V(\Theta) + \nabla_\Theta \int U(\Theta, \tilde{\Theta}) \|\tilde{\Theta}\| \rho_t(d\tilde{\Theta}) \right) \\ &= \|\Theta\| \left(\int f_*(x) x \mathbb{1}_{x^T \Theta > 0} \mu(dx) - \int \int x \mathbb{1}_{x^T \Theta > 0} \sigma(x^T \tilde{\Theta}) \|\tilde{\Theta}\| \rho_\infty(d\tilde{\Theta}) \mu(dx) \right). \end{aligned}$$

We know that any stationary point $(\rho_{+, \infty}, \rho_{-, \infty})$ has the following property [Mei et al., 2018]:

$$\text{supp}(\rho_\infty) \subseteq \left\{ \Theta : \int f_*(x) x \mathbb{1}_{x^T \Theta > 0} \mu(dx) = \int \int x \mathbb{1}_{x^T \Theta > 0} \sigma(x^T \tilde{\Theta}) \|\tilde{\Theta}\| \rho_\infty(d\tilde{\Theta}) \mu(dx) \right\}. \quad (7.4)$$

Multiplying both sides by $\|\Theta\| \Theta^T$ and recall the property of ReLU, the above condition implies that for all $\Theta \in \text{supp}(\rho_\infty)$, we have

$$\int f_*(x) \|\Theta\| \sigma(x^T \Theta) \mu(dx) = \int \int \|\Theta\| \sigma(x^T \Theta) \sigma(x^T \tilde{\Theta}) \|\tilde{\Theta}\| \rho_\infty(d\tilde{\Theta}) \mu(dx). \quad (7.5)$$

One can see the stationary condition on ρ_∞ (fixed points of the dynamics) (7.5) translates to

$$\mathcal{T}f_*(\Theta) = \left(\mathcal{T}\mathcal{T}^* \frac{d\rho_\infty}{d|\rho_\infty|} \right) (\Theta), \quad \forall \Theta \in \text{supp}(\rho_\infty). \quad (7.6)$$

Here the function $\frac{d\rho_\infty}{d|\rho_\infty|}$ is the Radon-Nikodym derivative. In addition, one can easily verify that, as ρ_∞ has bounded total variation

$$\frac{d\rho_\infty}{d|\rho_\infty|} \in L^2_{|\rho_\infty|}.$$

Therefore, combining all the above, one knows that

$$f_\infty(x) = \int \|\Theta\| \sigma(x^T \Theta) \rho_\infty(d\Theta) = \mathcal{T}^* \frac{d\rho_\infty}{d|\rho_\infty|} \in \mathcal{H}_\infty$$

and that for any $p \in L^2_{|\rho_\infty|}$

$$\langle f_* - f_\infty, \mathcal{T}^* p \rangle_\mu = \langle \mathcal{T}(f_* - f_\infty), p \rangle_{|\rho_\infty|} \quad (7.7)$$

$$= \left\langle \mathcal{T}f_* - \mathcal{T}\mathcal{T}^* \frac{d\rho_\infty}{d|\rho_\infty|}, p \right\rangle_{|\rho_\infty|} \quad (7.8)$$

$$= \int \left(\mathcal{T}f_* - \mathcal{T}\mathcal{T}^* \frac{d\rho_\infty}{d|\rho_\infty|} \right) (\Theta) |\rho_\infty|(d\Theta) = 0 \quad \text{due to (7.6)} \quad (7.9)$$

We have proved that $f_\infty = \mathcal{T}^* \frac{d\rho_\infty}{d|\rho_\infty|}$ satisfies normal condition for being a minimizer to (3.2). \square

Proof of Proposition 5.2. The first inequality in (5.16) is trivial. For the second inequality, it suffices to show for any $c = (c_1, \dots, c_p)^T$, x_1, \dots, x_p , Θ , we have

$$\sum_{i,j} c_i c_j \|\Theta\|^2 x_i^T x_j \mathbb{1}_{x_i^T \Theta > 0} \mathbb{1}_{x_j^T \Theta > 0} \geq \sum_{i,j} c_i c_j \sigma(x_i^T \Theta) \sigma(x_j^T \Theta) \quad (7.10)$$

The RHS equals

$$\sum_{i,j} c_i c_j x_i^T \Theta x_j^T \Theta \mathbb{1}_{x_i^T \Theta > 0} \mathbb{1}_{x_j^T \Theta > 0} = \left(\sum_i c_i x_i^T \Theta \mathbb{1}_{x_i^T \Theta > 0} \right)^2 \quad (7.11)$$

$$= \langle \Theta, \sum_i c_i x_i \mathbb{1}_{x_i^T \Theta > 0} \rangle^2 \leq \|\Theta\|^2 \left\| \sum_i c_i x_i \mathbb{1}_{x_i^T \Theta > 0} \right\|^2 = \text{LHS}. \quad (7.12)$$

For the last inequality, with compactness condition on ρ_∞ , we have

$$\sum_{i,j} c_i c_j \int \|\Theta\|^2 \sigma(x_i^T \Theta) \sigma(x_j^T \Theta) |\rho_\infty|(\Theta) \leq D^2 \sum_{i,j} c_i c_j \int \sigma(x_i^T \Theta) \sigma(x_j^T \Theta) |\rho_\infty|(\Theta). \quad (7.13)$$

Therefore, $D^2 K_\infty^{(1)} \geq H_\infty$.

□

Proof of Theorem 3.2. Let us rewrite Corollary 5.2 into

$$\frac{d}{dt} \|\Delta_t\|_\mu^2 = -2 \langle \Delta_t, \mathcal{K}_t \Delta_t \rangle_\mu = -2 \|\mathcal{K}_t^{1/2} \Delta_t\|_\mu^2, \quad (7.14)$$

here $\mathcal{K}_t : L_\mu^2(x) \rightarrow L_\mu^2(x)$ denotes the integral operator associated with K_t ,

$$(\mathcal{K}_t f)(x) := \int K_t(x, \tilde{x}) f(\tilde{x}) \mu(d\tilde{x}). \quad (7.15)$$

From (7.14)

$$\frac{d}{dt} \|\Delta_\infty\|_\mu^2 = -2 \|\mathcal{K}_\infty^{1/2} \Delta_\infty\|_\mu^2, \quad (7.16)$$

we know that the RHS equals zero implies

$$\begin{aligned} \|\mathcal{K}_\infty^{1/2} \Delta_\infty\|_\mu^2 &= 0 \\ \langle \mathcal{K}_\infty^{1/2} g, \Delta_\infty \rangle_\mu &= \langle g, \mathcal{K}_\infty^{1/2} \Delta_\infty \rangle_\mu = 0, \quad \forall g \in L_\mu^2. \end{aligned}$$

This further implies Δ_∞ lies in the kernel of RKHS \mathcal{K}_∞ as $\mathcal{K}_\infty = \{\mathcal{K}_\infty^{1/2} g : g \in L_\mu^2\}$. □

Proof of Proposition 4.1. The gradients on the original parameters are,

$$\begin{aligned} \frac{dw_j(t)}{dt} &= -\hat{\mathbf{E}} \left[\frac{\partial \ell(\mathbf{y}, f_t)}{\partial f} \sigma(\mathbf{x}^T u_j(t)) \right] - \frac{1}{m} \lambda w_j(t), \\ \frac{du_j(t)}{dt} &= -\hat{\mathbf{E}} \left[\frac{\partial \ell(\mathbf{y}, f_t)}{\partial f} w_j(t) \mathbb{1}_{\mathbf{x}^T u_j(t) \geq 0} \mathbf{x} \right] - \frac{1}{m} \lambda u_j(t). \end{aligned}$$

Clearly, on the rescaled parameter, the following holds

$$\begin{aligned} \frac{d\theta_j}{dt} &= \sqrt{m} \frac{dw_j}{dt} = -\hat{\mathbf{E}} \left[(f_t(\mathbf{x}) - \mathbf{y}) \sigma(\mathbf{x}^T \Theta_j(t)) \right] - \frac{1}{m} \lambda \theta_j, \\ \frac{d\Theta_j}{dt} &= \sqrt{m} \frac{du_j}{dt} = -\hat{\mathbf{E}} \left[(f_t(\mathbf{x}) - \mathbf{y}) \theta_j \mathbb{1}_{\mathbf{x}^T \Theta_j \geq 0} \mathbf{x} \right] - \frac{1}{m} \lambda \Theta_j. \end{aligned}$$

Multiply the first equation by θ_j , and the second equation by θ_j^T , take the difference, we can verify that

$$\frac{d(\theta_j^2 - \|\Theta_j\|^2)}{dt} = -\lambda/m(\theta_j^2 - \|\Theta_j\|^2) \quad (7.17)$$

$$\theta_j(t)^2 - \|\Theta_j(t)\|^2 = (\theta_j(0)^2 - \|\Theta_j(0)\|^2) \exp(-\lambda t/m) . \quad (7.18)$$

Therefore the balanced condition still holds at stationarity for arbitrary bounded initialization,

$$\theta_j(\infty)^2 - \|\Theta_j(\infty)\|^2 = 0, \forall j.$$

Now the optimality condition for the velocity field reads the following, for any $\Theta_j(\infty) \in \text{supp}(\hat{\rho}_\infty^\lambda)$ (we abbreviate the ∞ in the following display, note $\tilde{\theta}(\infty)$ corresponds to the second layer weights w.r.t. to $\tilde{\Theta}(\infty)$)

$$\begin{aligned} \theta_j \hat{\mathbf{E}}[\mathbf{y} \mathbb{1}_{\mathbf{x}^T \Theta_j \geq 0} \mathbf{x}] &= \theta_j \int \tilde{\theta} \hat{\mathbf{E}}[\mathbb{1}_{\mathbf{x}^T \Theta_j \geq 0} \mathbf{x} \sigma(\mathbf{x}^T \tilde{\Theta})] |\hat{\rho}_\infty^\lambda|(d\tilde{\Theta}) + \frac{1}{m} \lambda \Theta_j \\ \text{Multiply by } \Theta_j^T, \quad \theta_j \hat{\mathbf{E}}[\mathbf{y} \sigma(\mathbf{x}^T \Theta_j)] &= \int \theta_j \tilde{\theta} \hat{\mathbf{E}}[\sigma(\mathbf{x}^T \Theta_j) \sigma(\mathbf{x}^T \tilde{\Theta})] |\hat{\rho}_\infty^\lambda|(d\tilde{\Theta}) + \frac{\lambda}{m} \|\Theta_j\|^2 \\ \theta_j \hat{\mathbf{E}}[\mathbf{y} \sigma(\mathbf{x}^T \Theta_j)] &= \int \theta_j \tilde{\theta} \hat{\mathbf{E}}[\sigma(\mathbf{x}^T \Theta_j) \sigma(\mathbf{x}^T \tilde{\Theta})] |\hat{\rho}_\infty^\lambda|(d\tilde{\Theta}) + \lambda \int \theta_j \tilde{\theta} \mathbb{1}_{\tilde{\Theta}=\Theta_j} |\hat{\rho}_\infty^\lambda|(d\tilde{\Theta}) \end{aligned}$$

where the last step uses the condition $\theta_j^2(\infty) = \|\Theta_j(\infty)\|^2$, and the fact that $|\hat{\rho}_\infty^\lambda| = \frac{1}{m} \sum_{j=1}^m \delta_{\Theta_j}$ and

$$\int \theta_j \tilde{\theta} \mathbb{1}_{\tilde{\Theta}=\Theta_j} |\hat{\rho}_\infty^\lambda|(d\tilde{\Theta}) = \frac{1}{m} \theta_j^2 = \frac{1}{m} \|\Theta_j\|^2.$$

In the matrix form, where $\hat{\rho}_\infty^\lambda = \frac{1}{m} \sum_{l \in [m]} \text{sgn}(\theta_l) \delta_{\Theta_l}$

$$\sum_{l \in [m]} \left[n \hat{U}(\Theta_j, \Theta_l) + n \lambda \mathbb{I}_{\Theta_l = \Theta_j} \right] \theta_l / m = \sigma(\Theta_j^T X) Y.$$

Therefore, define $\sigma(x^T \Xi) := [\sigma(x^T \Theta_1), \dots, \sigma(x^T \Theta_m)] \in \mathbb{R}^{1 \times m}$, and $\sigma(X \Xi) := [\sigma(x_1^T \Xi)^T, \dots, \sigma(x_n^T \Xi)^T] \in \mathbb{R}^{m \times n}$, we have

$$\begin{aligned} \hat{f}_\infty^{\text{nn}, \lambda}(x) &= \sum_{l \in [m]} \theta_l \sigma(x^T \Theta_l) / m = \sigma(x^T \Xi) [\sigma(X \Xi) \sigma(X \Xi)^T + n \lambda I_m]^{-1} \sigma(X \Xi) Y \\ &= \sigma(x^T \Xi) \sigma(X \Xi) [\sigma(X \Xi)^T \sigma(X \Xi) + n \lambda I_n]^{-1} Y \\ &= \hat{H}_\infty^\lambda(x, X) \left[\hat{H}_\infty^\lambda(X, X) + n/m \cdot \lambda I_n \right]^{-1} Y . \end{aligned}$$

The last line follows as $\hat{H}^\lambda(x, \tilde{x}) := \int \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) |\hat{\rho}_\infty^\lambda|(d\Theta) = 1/m \cdot \sigma(x^T \Xi) \sigma(\tilde{x}^T \Xi)^T$.

□

Acknowledgement

We thank three anonymous referees for constructive feedback. Tengyuan Liang would like to acknowledge Maxim Raginsky for pointing out relevant references.

Supplemental Material

The remaining proofs are designated to the appendix, which is included in the supplemental material.

References

- Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.
- Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.
- Andrew R Barron, Albert Cohen, Wolfgang Dahmen, Ronald A DeVore, et al. Approximation and learning by greedy algorithms. *The annals of statistics*, 36(1):64–94, 2008.
- Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300*, 2019.
- Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*, 2018a.

- Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. *arXiv preprint arXiv:1802.01396*, 2018b.
- Yoshua Bengio, Nicolas L Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In *Advances in neural information processing systems*, pages 123–130, 2006.
- Bill Casselman. *Essays in analysis*. 2014.
- Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.
- Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.
- George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.
- Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.
- Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference: Application to causal effects and other semiparametric estimands. *arXiv preprint arXiv:1809.09953*, 2018.
- Stuart Geman and Chii-Ruey Hwang. Nonparametric maximum likelihood estimation by the method of sieves. *The Annals of Statistics*, pages 401–414, 1982.
- Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

- Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- Cong Huang, Gerald HL Cheang, and Andrew R Barron. *Risk of penalized least squares, greedy selection and l1-penalization for flexible function libraries*. PhD thesis, Yale University, 2008.
- 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.
- Lee K Jones. A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training. *The annals of Statistics*, 20(1):608–613, 1992.
- Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.
- Frederic Koehler and Andrej Risteski. Representational power of relu networks and polynomial kernels: Beyond worst-case analysis. *arXiv preprint arXiv:1805.11405*, 2018.
- Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel” ridgeless” regression can generalize. *The Annals of Statistics*, to appear, 2018.
- Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. 2020.
- Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. *arXiv preprint arXiv:1712.06559*, 2017.

- Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.
- Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layers neural networks. *arXiv preprint arXiv:1804.06561*, 2018.
- Partha Niyogi and Federico Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8(4):819–842, 1996.
- Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.
- Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.
- Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.
- Alexander Rakhlin and Xiyu Zhai. Consistency of interpolation with laplace kernels is a high-dimensional phenomenon. *arXiv preprint arXiv:1812.11167*, 2018.
- 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.
- Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

- Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 2019.
- Charles J Stone. Optimal rates of convergence for nonparametric estimators. *The annals of Statistics*, pages 1348–1360, 1980.
- Vladimir Vapnik. *Statistical learning theory. 1998*, volume 3. Wiley, New York, 1998.
- Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.
- Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.