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

Xialiang Dou\textsuperscript{1} and Tengyuan Liang\textsuperscript{*1}

\textsuperscript{1}University of Chicago

January 23, 2019

Abstract

Consider the problem: given data pair \((x, y)\) drawn from a population with \(f_\ast(x) = \mathbb{E}[y|x = x]\), specify a neural network 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 \(f_\ast\), 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. 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. In addition, we prove that as the RKHS is data-adaptive and task-specific, the residual for \(f_\ast\) lies in a subspace that is smaller than the orthogonal complement of the RKHS, formalizing the representation and approximation benefits of neural networks.

1 Introduction

Consider i.i.d. data pairs drawn from a joint distribution \((x, y) \sim P = P_x \times P_{y|x}\) on the space \(\mathcal{X} \times \mathcal{Y}\). Lies at the heart 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 \(f \in \mathcal{F} : \mathcal{X} \to \mathcal{Y}\) such that the population squared loss is small,

\[
L(f) = \mathbb{E} \left[ \frac{1}{2} (y - f(x))^2 \right] = \mathbb{E}_{x \sim P_x} \frac{1}{2} [f_\ast(x) - f(x)]^2 + \mathbb{E}_{(x,y) \sim P} \frac{1}{2} [y - f_\ast(x)]^2,
\]

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

Theoretically, researchers approach the above question mainly in two ways. The first is by assuming the conditional expectation \(f_\ast\) lies in the correct model class \(\mathcal{F}\). For example, say \(\mathcal{F}\) consists of

\*tengyuan.liang@chicagobooth.edu. The author gratefully acknowledges the support of the George C. Tiao faculty fellowship.
linear functions, or splines with certain order of smoothness, or more broadly functions lie in a reproducing kernel Hilbert space (RKHS). Conceptually, this “well-specification” assumption requires strong knowledge about what model class $\mathcal{F}$ may be good for the regression task at hand, which is rarely met in practice. Within each framework, minimax optimal rates and extensive study have been established (Stone, 1980; Wahba, 1990). The second, 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 $\mathcal{F}_\epsilon$ called sieves with $\epsilon$ changing (Geman and Hwang, 1982), such that the class $\mathcal{F}_\epsilon$ contains an $\epsilon$-approximation to any $f_*$ under some metric. A final result usually requires a careful balancing of the approximation and stochastic error by tuning $\epsilon$. Concrete cases for the latter approach include: polynomials (Stone-Weierstrass, Bernstein), radial-basis (Park and Sandberg, 1991; Niyogi and Girosi, 1996), 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; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Farrell et al., 2018; Koehler and Risteski, 2018; Poggio et al., 2017), to name a few.

However, the following major drawbacks of the above current theory make it inadequate to present an adaptive and realistic explanation of the practical success of neural networks:

- The function computed in practice could be very different from that claimed in the approximation theory, either by existence or by construction. To see this, consider the one hidden layer neural networks. It will be hard to conceive that the function computed in practice using now-standard stochastic gradient descent (SGD) training procedure is anywhere close to the one asserted by the universal approximation theorem in analysis.
- In practice, what researchers usually do is to try out different model classes $\mathcal{F}$ to learn from the data on which representation suits the best. For example, using different kernels machines, random forests, or specify certain architectures then run SGD on neural networks. In this case, strictly speaking, the model class choice depends on the data in an adaptive way that is very distinct from the theoretical approach we undertake in the literature.

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

In addition, 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 neural networks compared to the classical nonparametric pre-specified fixed basis representation.

### 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), with the time index $t$ corresponding to
the evolution of parameters driven by the gradient flow/descent (GD) training dynamics,

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

(1.2)

Here each individual pair \((w_j \in \mathbb{R}, u_j \in \mathbb{R}^d)\) in the summation is called 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 \(z := (x, y)\), the parameters \((w_j, u_j)\) evolve with time \(t\) as follows

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

(1.3)

More generally, we can rewrite the function computed by NN at time \(t\) as

\[ f_t(x) := \int \sigma(x^T u) \tau_t(du), \]  

(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 notion of re-scaling of \(\tau_t\) (denoted as \(\rho_t\)), then derive the corresponding distribution dynamic driven by the gradient flow later in Section 2.2.

The intimate connection between two-layer neural networks and reproducing kernel Hilbert spaces (RKHS) has been studied in the literature, see Cho and Saul (2009); Daniely et al. (2016); Bach (2017). However, to the best of our knowledge, all known results are based on a fixed RKHS (in our notation \(K_0\) in Section 2.1). In that sense, random features for kernel learning (Rahimi and Recht, 2008; Rudi and Rosasco, 2017) can be viewed as NN with fixed random sampled first layer weights, and tunable second layer weights. From neural networks side, Mei et al. (2018); Du et al. (2018b) study when the second layer weights are fixed, but the first layer weights tunable. In contrast, we will establish a theory with the dynamic and data-adaptive RKHS obtained via training neural networks, with standard GD on weights of both layers.

In this paper, when the joint distribution \(P\) has finite support, and the minimizer of Eqn. (1.1) has loss zero (i.e., \(f(x_i) = y_i\) with \((x_i, y_i)\)'s scan the support of \(P\)), we refer to the problem as 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). Otherwise, we call it the approximation problem.

1.2 Main Results: Informal

We state the main results of the paper, in an informal way. The formal statements are deferred to Section 2 and 3. One can view that the gradient flow training dynamics — on the parameters of NN — induce a sequence of functions \(\{f_t : t \geq 0\}\) and dynamic RKHS \(\{\mathcal{H}_t : t \geq 0\}\), indexed by the time \(t\).

Here we denote \(\mu := P_x\), and consider all \(f_s \in L^2_\mu\).

**GD on NN is projection to data-adaptive RKHS** The first theorem characterizes \(f_t\) — the function computed by GD on the parameters of NN — when reaches any stationarity as \(t \to \infty\).

Define \(f_x = \lim_{t \to \infty} f_t\) as the function computed by ReLU networks (defined in 1.2, or more generally in 2.7) until any stationarity of the gradient flow dynamics (defined in 1.3, with the squared loss, for the population distribution \((x, y) \sim P\)). Define the corresponding stationary RKHS \(\mathcal{H}_x = \lim_{t \to \infty} \mathcal{H}_t\) (defined in 3.1). Recall \(f_s(x) = \mathbb{E}[y|x = x]\).
**Theorem 1** (Informal version of Theorem 4). Consider \( f_\star \in L^2_\mu \), 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 \( f_\star \) satisfies

\[
 f_\star = \arg\min_{g \in \mathcal{H}_\star} \| f_\star - g \|_{L^2_\mu}^2 .
\]  

(1.5)

Training NN is learning a dynamic representation/basis (quantified by \( \mathcal{H}_t \)), at the same time updating the predicted function \( f_t \). A qualitative pictorial illustration can be found in Fig. 1. Curiously, the above theorem unveils the fact that \( \lim_{t \to \infty} f_t \) obtained by training on the weights over time until any stationarity, is the same as the projection of \( f_\star \) onto the stationary RKHS \( \mathcal{H}_\star \). The projection is the solution to the classic nonparametric least squares, had one known the adaptive representation \( \mathcal{H}_\star \) beforehand. 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.

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

**Theorem 2** (Informal version of Theorem 6). Consider \( f_\star \in L^2_\mu \). As in the setup for Theorem 1, decompose \( f_\star \) into the function \( f_\star \) computed by the neural network and the residual \( \Delta_\star \)

\[
 f_\star = f_\star + \Delta_\star ,
\]  

(1.6)

then there is another RKHS (defined in 3.7) \( \mathcal{K}_\star \supset \mathcal{H}_\star \), such that

\[
 f_\star \in \mathcal{H}_\star , \quad \Delta_\star \in \text{Ker}(\mathcal{K}_\star) \subset \text{Ker}(\mathcal{H}_\star) ,
\]  

(1.7)

with \( \mathcal{H}_\star \oplus \text{Ker}(\mathcal{K}_\star) \neq L^2_\mu \).

Figure 1: Illustration of Theorem 1 and 4. Red dotted line denotes the function \( f_t \) computed along the gradient flow dynamics on the weights of NN. Along training, one is also learning a sequence of dynamic RKHS representation \( \mathcal{H}_t \). Finally, \( f_t \) converges to the projection of \( f_\star \) onto \( \mathcal{H}_\star \). We emphasize that the initial function \( f_0 \) computed by NN is very different from the projection of \( f_\star \) onto the initial RKHS \( \mathcal{H}_0 \).
The above theorem formalizes the widely-acknowledged intuition in practice (Wenliang et al., 2018) that neural networks has the advantage of learning task-specific, data-dependent adaptive basis, compared to the conventional nonparametric fixed basis approach. In the above decomposition, there are two data-adaptive functional spaces $K_\mathcal{X}$ and $H_\mathcal{X}$, with distinct kernels $\text{Ker}(K_\mathcal{X}) \subset \text{Ker}(H_\mathcal{X})$. Therefore, the decomposition $f_\mathcal{X} + \Delta_\mathcal{X}$ is not a trivial orthogonal decomposition of the RKHS $H_\mathcal{X}$ and its complement $\text{Ker}(K_\mathcal{X})$. In other words, as the learned adaptive basis $H_\mathcal{X}$ (from GD) depends on the data distribution and the task $f_*$ implicitly, it has advantage of representing $f_*$ by squeezing the residual into a smaller subspace of the kernel $H_\mathcal{X}$. A pictorial illustration can be found in Fig. 2.

Figure 2: Illustration of Theorem 2 and 6: 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 $H_\mathcal{X}$ that depends on $f_*$. Therefore, the residual $\Delta_\mathcal{X}$ lies in a subspace of $\text{Ker}(H_\mathcal{X})$.

1.3 Notations

We use boldface lower case $x$ to denote a random variable or vector. The normal letter $x$ can either be a scaler or a vector when there is no confusion. We use boldface capital letter $K$ denote a finite dimensional matrix. The transpose of a matrix $A$, resp. vector $u$ is denoted by $A^T$, resp. $u^T$. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. We use $A[i, j]$ to denote the $i, j$-th entry of a matrix. We denote $\mathbb{1}_D$ as the indicator function of set $D$. 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 a Hilbert space. We use $\langle f, g \rangle_\mu = \int f(x)g(x)\mu(dx)$ to denote the inner product in $L^2_\mu$. Notation $E_x$ is the expectation w.r.t random variable $x$, and $E_{x, \bar{x}} h(x, \bar{x}) = \int h(x, \bar{x})\mu(dx)\mu(d\bar{x})$. For a signed measure $\rho = \rho_+ - \rho_-$ with the positive and negative parts, define $|\rho| = \rho_+ + \rho_-.$

2 Time-varying Kernel

In this section, we inspect the training process through the lens of a time-varying kernel (2.1) defined by the gradient flow (GD) on the weights of a two-layer neural network.

Lemma 2.1 (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
\[ \Delta_t(x) = f_t(x) - f_t(x) \] be the residual. Define the time-varying kernel \( K_t(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \),

\[
K_t(x, \tilde{x}) = \sum_{j=1}^{m} \left[ \sigma(x^T u_j(t)) \sigma(\tilde{x}^T y_j(t)) + w_j(t)^2 \mathbb{1}_{x^T \tilde{u}_j(t) \geq 0} \mathbb{1}_{\tilde{x}^T \tilde{u}_j(t) \geq 0} x^T \tilde{x} \right].
\] (2.1)

Then the ODE of the residual \( \Delta_t \) driven by the GD dynamics satisfies,

\[
\frac{dE_{x}}{dt} \left[ \frac{1}{2} (\Delta_t(x))^2 \right] = -E_{x, \tilde{x}} \left[ \Delta_t(x) K_t(x, \tilde{x}) \Delta_t(\tilde{x}) \right].
\] (2.2)

As a corollary, consider using GD to solve the empirical risk minimization (ERM). In this in-sample case, inspecting the dynamics of the sample residual \( \| \Delta_t \|_n \) has been considered in Du et al. (2018b), where the optimization is done only on the first layer weights, with fixed second layer. In contrast, here we are optimizing the weights of both layers, resulting in a different kernel. Plugging in the empirical distribution over the \( n \) data points \((x_1, y_1), \ldots, (x_n, y_n)\), we can define the finite dimensional time-varying kernel matrix \( K \in \mathbb{R}^{n \times n} \) to be \( K[i, j] = K_t(x_i, x_j) \).

**Corollary 2.1.** Consider the GD training process (1.3) with empirical distribution on finite samples \( \{(x_i, y_i)\}_{i=1}^{n} \). Let \( \Delta_t = (y_1 - f_t(x_1), \ldots, y_n - f_t(x_n))^T \) be the residual vector, with kernel matrix defined above. Then we have,

\[
\frac{d}{dt} \| \Delta_t \|_n^2 = -\Delta_t^T K_t \Delta_t.
\]

**Remark 2.1.** Note the rate of change for the approximation error \( E_x[\| \Delta_t(x) \|^2] \), or the interpolation error \( \| \Delta_t \|_n^2 \) depends on the time-varying kernel \( K_t \). We would like to make two remarks: first, one can see that whole spectrum of \( K_t \) matters to the dynamics of the error; second, the error will keep decreasing until \( \Delta_t \) lies in the null space of \( K_t \).

For a general loss function \( \ell(y, f) \) with curvature (say, logistic loss), similar results hold.

**Corollary 2.2.** Consider a general loss function \( \ell(y, f) \) that is \( \alpha \)-strongly convex in the second argument \( f \), with \( K_t \) defined in (2.1). Assume in addition \( K_t \) has smallest eigenvalue \( \lambda > 0 \). Define \( \Delta_t(x) := E_{y \sim P_{y|x}} \left[ \frac{\partial \ell(y, f_t(x))}{\partial f} \right] \), then we have

\[
\frac{dE[\ell(y, f_t(x))]}{dt} = -E_{x, \tilde{x}} \left[ \Delta_t(x) K_t(x, \tilde{x}) \Delta_t(\tilde{x}) \right] \leq -2\alpha \lambda E \left[ \ell(y, f_t(x)) - \ell(y, f_*(x)) \right].
\]

for all \( f_* : \mathbb{R}^d \rightarrow \mathbb{R} \).

### 2.1 Initialization and \( K_0 \)

In this section, we analyze the initial kernel matrix \( K_0 \) under infinitesimal random initialization, as done in practice. Specifically, consider the initialization with \( w_j \) being \( \pm 1/\sqrt{m} \) with equal chance and \( u_j \sim N(0, 1/m \cdot I_d) \) i.i.d. sampled. We shall call this infinitesimal random initialization as the initial weights vanishes to 0 as \( m \rightarrow \infty \). We characterize the initial state of the kernel \( K_0 \) in the next lemma.
Lemma 2.2 (Fixed Kernel). With initialization specified above, consider w.l.o.g. \( \|x\| = \|\tilde{x}\| = 1 \), with \( u \sim \pi \) is the isotropic Gaussian \( N(0, I_d) \), by law of large number, we have,

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

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 are based on some fixed kernel, as in the above lemma. However, to instantiate useful statistical rates, one requires \( f_* \) to lie in the corresponding pre-specified RKHS \( K_0 \), which is non-verifiable in practice. In fact, as discussed in details (Bach, 2017, Section 2.3), there is a subtle yet important difference between a function representable by a two-layer NN and lying in the corresponding RKHS. In contrast, we will establish a dynamic and adaptive kernel theory defined by GD, without making any structural assumptions on \( f_* \) other than \( f_* \in L^2_{\mu} \).

2.2 Evolution Underlying \( K_t \)

In this section we derive the evolution of the signed measure defined by the neurons during the training process, which in turn determines the dynamic kernel \( K_t \) defined in (2.1). To generalize the result to the infinite neurons case, we follow and borrow tools from the mean-field characterization (Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Jordan et al., 1998). However, we spell out an interesting notion of re-scaling to capture the practical concerns: (1) weights on both layers are optimized following the gradient flow; (2) infinitesimal random initialization is used in practice.

Rewrite (1.1) according to the signs of the second layer weights

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

We show later that following gradient descent flow (1.3) with particular initialization, \( w_{+,j} \) will stay non-negative and \( w_{-,j} \) non-positive. To be concrete, we consider infinitesimal initialization with all \( u_{+,j}(0) \) and \( u_{-,j}(0) \) drawn i.i.d. from probability measures \( \rho_{+,0} \) and \( \rho_{-,0} \) (that do not depend on \( m \)) and then scaled by \( 1/\sqrt{m} \). Set \( w_{+,j}(0) = \| u_{+,j}(0) \| \geq 0 \) and \( w_{-,j}(0) = -\| u_{-,j}(0) \| \leq 0 \).

To layout the distribution dynamic theory, we introduce a parameter re-scaling with a \( \sqrt{m} \) 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,

\[
f_t(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_{+,j}(t) \sigma(x^T \Theta_{+,j}(t)) + \frac{1}{m} \sum_{j=1}^{m} \theta_{-,j}(t) \sigma(x^T \Theta_{-,j}(t)). \tag{2.3}
\]

By the positive homogeneity of ReLU, we have the corresponding dynamics on the re-scaled
parameters,
\[
\frac{d\theta^{-j}}{dt} = \sqrt{m} \frac{dw^{-j}}{dt} = -\sqrt{m} E_{\mathbf{z}} \left[ \frac{\partial \ell(y, f(x))}{\partial f} \sigma(x^T \theta^{-j}) \right] = -E_{\mathbf{z}} \left[ \frac{\partial \ell(y, f(x))}{\partial f} \sigma(x^T \Theta^{-j}) \right], \tag{2.4}
\]
\[
\frac{d\Theta^{-j}}{dt} = \sqrt{m} \frac{dw_{-j}}{dt} = -\sqrt{m} E_{\mathbf{z}} \left[ \frac{\partial \ell(y, f(x))}{\partial f} w_{-j} 1_{x^T \theta_{-j} \geq 0} \mathbf{x} \right] = -E_{\mathbf{z}} \left[ \frac{\partial \ell(y, f(x))}{\partial f} \theta_{-j} 1_{x^T \theta_{-j} \geq 0} \mathbf{x} \right]. \tag{2.5}
\]

Note that this is different from directly performing gradient descent with the re-scaled parameter formulation. In order words, we are representing \( f_t \) via new parameters for convenience, while still studying the dynamics driven by training on the original parameters. Define
\[
\rho_{+,t} := \frac{1}{m} \sum_{j=1}^{m} \delta_{\Theta_{+,j}(t)}, \quad \rho_{-,t} := \frac{1}{m} \sum_{j=1}^{m} \delta_{\Theta_{-,j}(t)} \tag{2.6}
\]
as the empirical distribution of neurons on the parameter space \( \Theta \). This notations \( \rho_{+,t} \) and \( \rho_{-,t} \) converge to proper distributions in the infinite neurons limit \( m \to \infty \), see e.g. Bach (2017); Mei et al. (2018). Define the signed measure as \( \rho_t = \rho_{+,t} - \rho_{-,t} \), we will establish a simple balancing lemma and then show that
\[
f_t(x) = \int \|\Theta\| \sigma(x^T \Theta) \rho_t(d\Theta). \tag{2.7}
\]

We define the following terms which correspond to the velocity field driven by the regression task and the interaction among particles,
\[
V(\Theta) = E[y \sigma(x^T \Theta)], \quad U(\Theta, \tilde{\Theta}) = -E[\sigma(x^T \Theta) \sigma(x^T \tilde{\Theta})]. \tag{2.8}
\]
The following theorem casts the training process as distribution dynamics on \( \rho_{+,t}, \rho_{-,t} \).

**Theorem 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 \( \rho_t \) defined in (2.6) with possibly infinite neurons, we have the following PDE characterization on distribution dynamics of \( \rho_{+,t}, \rho_{-,t} \)
\[
\dot{\rho}_{+,t}(\Theta) = -\nabla_{\Theta} \cdot \left[ \rho_{+,t}(\Theta) \cdot \|\Theta\| \left( \nabla_{\Theta} V(\Theta) + \nabla_{\Theta} \int U(\Theta, \tilde{\Theta}) \|\tilde{\Theta}\| \rho_t(d\tilde{\Theta}) \right) \right],
\]
\[
\dot{\rho}_{-,t}(\Theta) = \nabla_{\Theta} \cdot \left[ \rho_{-,t}(\Theta) \cdot \|\Theta\| \left( \nabla_{\Theta} V(\Theta) + \nabla_{\Theta} \int U(\Theta, \tilde{\Theta}) \|\tilde{\Theta}\| \rho_t(d\tilde{\Theta}) \right) \right]. \tag{2.9}
\]

Moreover, the kernel \( K_t \) is defined as
\[
K_t(x, \tilde{x}) = \int \left( \|\Theta\|^2 1_{x^T \Theta \geq 0} 1_{\tilde{x}^T \Theta \geq 0} x^T \tilde{x} + \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \right) |\rho_t|(d\Theta). \tag{2.10}
\]

**Remark 2.2.** As in Mei et al. (2018); Rotskoff and Vanden-Eijnden (2018), let’s first show that in the infinite neurons limit \( m \to \infty \), \( \rho_{+,t}, \rho_{-,t} \) are properly defined, with Eqn. (2.9) also characterizing the distribution dynamics, induced by the gradient flow training. 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 (2.6) to
indicate their dependence on $m$. Consider $\nabla_\Theta V(\Theta), \nabla_\Theta U(\Theta, \tilde{\Theta})$ in (2.8) are bounded and uniform Lipchitz continuous as in (Mei et al., 2018, Theorem 3). With the same proof as in (Mei et al., 2018, A3). With the same proof as in (Mei et al., 2018, Theorem 3), one can show that with $m \to \infty$, the initial distribution $\rho^m_0 \Rightarrow \tilde{\rho}_0 = \rho_{+,0} - \rho_{-,0}$ by law of large number. And by the solution’s continuity depending on the initial value, we have $\rho^m_t \Rightarrow \rho_t$ as $m \to \infty$ well defined, for any fixed $t$.

Note that our problem setting is different from that in Mei et al. (2018), where the authors consider the NN with fixed second layer weights to be $1/m$. To reiterate, the re-parameterization via $\theta$ and $\Theta$ is crucial to connect to practical concerns: (1) weights on both layers are optimized following the gradient flow; (2) infinitesimal random initialization is used in practice. In the setting of (Mei et al., 2018, Eqn. (3)), the training process not the same as the vanilla GD on weights, with an additional $m$ factor in the velocity term. This subtlety is also addressed in Rotskoff and Vanden-Eijnden (2018). In short, the re-scaling looks at the dynamics where $\Theta$’s are on the right scale as $m \to \infty$, and the analogy from vanilla GD to distribution dynamics transits naturally. 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 (2.8) compared to that in Mei et al. (2018).

We introduce two simple propositions useful to Theorem 3. First, we present a simple version of the result firstly proved in Maennel et al. (2018); Du et al. (2018a). It shows that with the initialization and the GD training process, we have a balanced condition which indicates that the degree of freedom of the system is determined by $\Theta$’s ($u$’s). Note that our initialization of $w_j$ is only for the ease of presentation.

**Proposition 2.1** (Balanced condition). For $\theta_+ (t), \theta_- (t), \Theta_+ (t)$ and $\Theta_- (t)$ defined in (2.3)-(2.5), and the initialization specified above, at any time $t$, we have

$$
\theta_+ (t) = \|\Theta_+ (t)\|, \quad \theta_- (t) = -\|\Theta_- (t)\|.
$$

**Proposition 2.2** (Absorbing state). For the training process (1.3) for problem (1.1) with NN (1.2), once $w_j (t)$ and $u_j (t)$ hit zero at $t_0$, for $t > t_0$ at least there exists a solution that can be viewed as training without the $j$-th neuron.

**Remark 2.3.** The balanced condition inspires us to represent infinite NN (1.4) in the form of (2.7). This can be done by substituting $\theta_j$ by $\|\Theta_j\|$ and taking $m$ goes to infinity to obtain the representation by law of large number. Theorem 3 derives the evolution of $\rho_t$ by re-writing (1.3) via the transportation equation, and extends to the infinite neurons case. Here we briefly demonstrate that the no-sign-change condition on $w_+$ and $w_-$ is valid. Viewing (1.3) as an ODE, it is natural to assume that this ODE satisfies the existence condition, since we can train this NN from arbitrary initialization. Once a $w_j (t)$ hits zero, the corresponding $u_j (t)$ hits zero as well by the balanced condition, and the velocity for $u_j$ and $w_j$ also vanishes. Intuitively, the gradient flow for this neuron will stick to zero afterwards. Moreover, if the ODE system satisfies the uniqueness condition, with the same proof, one can show that the sign of $w_j$’s will never change.

### 3 Benefits of Adaptive Representation: Time-varying RKHS

We are now ready to state two main results of the paper, Theorem 4 and Theorem 6.
3.1 Gradient Flow, Projection and Adaptive RKHS

We answer the question on how the function \( f_t \) computed from gradient flow on NN represents \( f_\star(x) = \mathbb{E}[y|x = x] \), under the squared loss. Suppose the gradient flow dynamics (2.9) (or equivalently (1.3)) reach any stationarity \((\rho_+, \rho_-)\), and assume that \( \text{TV}(\rho_\star) < \infty \) with a compact support. We employ the notation \( \rho_\star \) because reaching a stationary point can be viewed as \( t \to \infty \). We would like to emphasize that this stationary signed measure \( \rho_\star \) is task adaptive: it implicitly depends on the regression task \( f_\star \) and the data distribution \( P \), instead of being pre-specified by the researcher (see for instance in Bach (2017); Daniely et al. (2016); Cho and Saul (2009), a uniform measure on the sphere).

We use this stationary signed measure \( \rho_\star \) to construct a stationary RKHS. For completeness we walk through the construction of the kernel and the RKHS with \( \rho_\star \). Define the linear operator \( \mathcal{T} : L^2_\mu(x) \to L^2_{\rho_\star}(\Theta) \), such that for any \( f(x) \in L^2_\mu(x) \)

\[
(\mathcal{T} f)(\Theta) := \int f(x) \| \Theta \| \sigma(x^T \Theta) \mu(dx).
\]

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

\[
(\mathcal{T}^* p)(x) := \int p(\Theta) \| \Theta \| \sigma(x^T \Theta) |\rho_\star|(d\Theta).
\]

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

\[
H_{\rho_\star}(x, \tilde{x}) = \int \| \Theta \|^2 \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) |\rho_\star|(d\Theta), \quad \text{and} \quad (\mathcal{T}^* \mathcal{T} f)(x) := \int H_{\rho_\star}(x, \tilde{x}) f(\tilde{x}) \mu(d\tilde{x}). \tag{3.1}
\]

We construct the RKHS \( \mathcal{H}_{\rho_\star} \) via \( H_{\rho_\star} \). 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 \( \infty \), and \( \lambda_i > 0 \), and \( e_i \) 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, \( t_i \) can represent a function in \( L^2_{\rho_\star}(\Theta) \) or a linear functional. The RKHS can be specified as follows.

\[
\mathcal{H}_{\rho_\star} = \left\{ h \ | \ h(x) = \sum_i h_i e_i(x), \ \sum_i \frac{h_i^2}{\lambda_i} < \infty \right\}.
\]

We refer to \( H_{\rho_\star} \) defined in (3.1) as the stationary RKHS kernel. With the RKHS established above, we are ready to state the following theorem.

Theorem 4. Consider the approximation problem (1.1), with finite or infinite neurons in the NN. Consider the training dynamics (2.9) reaching any stationarity \((\rho_+, \rho_-)\) with compact support and finite total variation, and define the corresponding stationary RKHS \( \mathcal{H}_{\rho_\star} \) with kernel in (3.1). Then the function computed by neural network at the stationarity

\[
f_{\rho_\star}(x) = \int \| \Theta \| \sigma(x^T \Theta) \rho_\star(d\Theta), \tag{3.2}
\]

10
is a global minimizer of the approximation to the conditional mean \( f_\ast \) within the RKHS \( \mathcal{H}_\infty \)

\[
f_\infty(x) \in \arg\min_{g \in \mathcal{H}_\infty} \| f_\ast(x) - g(x) \|_{L_\mu^2}^2.
\]

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

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.

Note in the infinite neuron case, for any fixed time \( t \), with the proper random initialization, setting \( m \to \infty \) defines a proper distribution dynamics on \( \rho_t \) as in Theorem 3. Then let \( t \to \infty \) to reach the stationarity RKHS \( \mathcal{H}_\infty \).

From the above, we have the following natural decomposition,

\[
\Delta_\infty(x) = f_\ast(x) - f_\infty(x) \in \text{Ker}(\mathcal{H}_\infty).
\]

Surprisingly, as we shall show in the next section, \( \Delta_\infty \) 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.

### 3.2 Representation Benefits of Adaptive RKHS

We now need to define another adaptive RKHS \( \mathcal{K}_\infty \), which turns out to be different from \( \mathcal{H}_\infty \) in (3.1). Interestingly, the difference in these two kernels sheds light on the representation benefits of adaptive RKHS. We start with generalizing Lemma 2.1 with possibly infinite neurons (2.9), to characterize the residual \( \Delta_t \).

**Corollary 3.1.** Consider the approximation problem (1.1) with possibly infinite neurons NN (2.7), and the training process (2.9). Let \( \Delta_t(x) = f_\ast(x) - f_t(x) \) be the residual. Define the time-varying kernel matrix \( K_t(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \)

\[
K_t(x, \tilde{x}) = \int \left( \| \Theta \|_{L_\mu^2}^2 1_{x^T \Theta \geq 0} 1_{\tilde{x}^T \Theta \geq 0} x^T \tilde{x} + \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \right) |\rho_t| (d\Theta)
\]

\[
=: K_t^{(0)}(x, \tilde{x}) + K_t^{(1)}(x, \tilde{x}).
\]

Then we still have

\[
\frac{d}{dt} \mathbb{E}_x \left[ \frac{1}{2} \Delta_t(x)^2 \right] = -\mathbb{E}_x,\tilde{x} \left[ \Delta_t(x) K_t(x, \tilde{x}) \Delta_t(\tilde{x}) \right].
\]
We refer to (3.5) as GD kernel. Remark that when the training dynamics reaches the stationarity \( \rho_8 \), then the kernel defined by GD is different than the stationary RKHS kernel \( H_8 \) defined in (3.1)

\[
K_8(x, \tilde{x}) = \int \left( \|\Theta\|^2 1_{x^T \Theta \geq 0} 1_{\tilde{x}^T \Theta \geq 0} x^T \tilde{x} + \sigma(x^T \Theta)\sigma(\tilde{x}^T \Theta) \right) |\rho_8||d\Theta| \neq H_8(x, \tilde{x}) \tag{3.7}
\]

We use \( K_t : L^2_\mu(x) \rightarrow L^2_\mu(x) \) to denote the integral operator associated with \( K_t \), \( pK_t f \equiv \int K_t(x, \tilde{x})f(\tilde{x})\mu(d\tilde{x}) \).

With a slight abuse of notation, we denote the corresponding RKHS to be \( K_t \) as well.

In the next theorem, we compare the kernels \( K_8 \) and \( H_8 \), defined in (3.5) and (3.1) respectively.

**Theorem 5.** Consider the training process reaches any stationarity \( \rho_8 = \rho_{+, \infty} - \rho_{-, \infty} \) with compact support within radius \( D \) and finite total variation. We have

\[
K_8 \preceq K_8^{(0)} \preceq K_8^{(1)} \geq \frac{1}{D^2} H_8, \tag{3.8}
\]

with \( K_8^{(0)}, K_8^{(1)} \) defined in (3.5). Combining with the fact that \( H_8 \neq K_8 \) implies \( \text{Ker}(K_8) \subset \text{Ker}(H_8) \).

Once reached stationarity, due to the ODE defined by GD, the residual must satisfy

\[
\Delta_8(x) = f_8(x) - f_8(x) \in \text{Ker}(K_8). \tag{3.9}
\]

To sum up, we have the following theorem.

**Theorem 6.** Consider the problem (1.1) with possibly infinite neurons, and the gradient flow dynamics (2.9) (or equivalently (1.3)) training on data pair \( (x, y) \sim P \). When reaching any stationarity \( \rho_8 \), we have

\[
f_8 = f_\infty + \Delta_\infty.
\]

Recall the operator \( T \), RKHS \( H_8 \) in (3.1), and the GD RKHS \( K_8 \) in (3.5), all learned based on \( \rho_8 \) that depends on the data \( (x, y) \sim P \) and \( f_8 \). We have

\[
f_\infty = \int \|\Theta\|^2 \sigma(x^T \Theta)\rho_\infty(d\Theta) = T^{*} \frac{d\rho_\infty}{|\rho_\infty|} \in H_8, \quad \Delta_\infty = f_8 - f_\infty \in \text{Ker}(K_8) \subset \text{Ker}(H_8).
\]

Therefore GD on NN decomposes \( f_8 \) into two parts, each lies in a space that is NOT the orthogonal complement to the other.

**Remark 3.2.** As we can see \( \text{Ker}(K_8) \) and \( \text{Ker}(H_8) \) are not the same. Therefore, the decomposition \( f_\infty + \Delta_\infty \) is not a trivial orthogonal decomposition of the RKHS \( H_8 \) and its complement.

Recall Theorem 4, projecting \( f_8 \) to the RKHS \( H_8 \) with the data-adaptive kernel

\[
H_8(x, \tilde{x}) = \int \|\Theta\|^2 \sigma(x^T \Theta)\sigma(\tilde{x}^T \Theta) |\rho_\infty|(d\Theta)
\]
associated with $|\rho_x|$ is the same as the function constructed by neural networks (GD limit as $t \to \infty$). However, the residual lies in a possibly much smaller space due to Theorem 6, which is the null space of the RKHS $\mathcal{K}_\infty$

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

In other words, as the learned adaptive basis $\mathcal{H}_\infty$ (from GD) depends on the data distribution and the task $f_\pi$ implicitly, it has advantage of representing $f_\pi$ by squeezing the residual into a smaller subspace of the null space of $\mathcal{H}_\infty$. A pictorial illustration can be found in Fig. 2. This representation and approximation benefits help 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 $f_\pi$, thus the residual/interpolation error lies in a smaller space; (2) in view of the ODE in Corollary 3.1, the second layer of NN adds implicit regularization to the smallest eigenvalues of $K_t$, thus improving the converging speed of $\Delta_t$ to zero.

4 Experiments

We run experiments to illustrate the spectral decay of the dynamic kernels defined in $K_t$ over time $t$. The exercise is to showcase that during neural network training, one does learn the data-adaptive representation, that is task-specific depending on the true complexity of $f_\pi$. The training process is the same as what we theoretically prove: 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 $y_i = f_\pi(x_i) = \sum_{j=1}^{J} w_{ij}^* \sigma(x_i^T u_j^*)$ with different $J$. In other words, we choose different target $f_\pi$ (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 the index of eigenvalues in descending order, and the $y$-axis is the logarithmic values of corresponding eigenvalues. Different color indicates the spectral decay of the $K_t$ at different training step $t$. The eigenvalue-decays stabilize over time $t$ means that the training process approaches stationarity. As we can see with $f_\pi$ belongs to the NN family, the eigenvalues of kernel matrix in general become larger during the training process. For more complicated target function, it takes longer to reach the stationarity.
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 \( \pm 1 \) with equal chance. We select \( m = 200, 500 \), and \( n = 50, 200 \) to investigate those parameters’ influence on the kernel \( K_t \). We would like 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 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.

5 Extensions

In this section, we extend the definition of the dynamic kernel in Section 2 to the multi-layer neural networks case. We construct a recursive expression for the kernel defined by the multi-layer perceptron (MLP). Let $\Theta_{l,j}^l, l = 0, \cdots, h - 1$ denote the coefficient from the $i$-th node on the $l$-th layer to the $j$-th node on the $(l+1)$-th layer. Let the input (before activation) of the $i$-th node on $l$-th layer be $v_i^l(x) = \sum_j \Theta_{j,i}^{l-1} o_j^{l-1}(x)$ and let the output at that node be $o_i^l = \sigma(v_i^l)$, for $l \neq \{0, h\}$,
and \( d_l = x_i \), for \( l = 0 \). The final output \( g(x) = (v_1^h(x), v_2^h(x), \ldots, v_L^h(x))^T \). Let \( L_0 = d \) and \( L_i \) is the number of nodes at the \( i \)-th layer. Denote \( K_t^h(x, \tilde{x}; \{\Theta^i\}_{l=0,\ldots,h}) \) the kernel of \( h \) layers NN. The training dynamic is still the gradient flow, for all \( \Theta \)

\[
\frac{d\Theta}{dt} = -\mathbb{E}_x \left[ \frac{\partial f(y, g(x))}{\partial y} \frac{\partial g(x)}{\partial \Theta} \right].
\]

**Proposition 5.1.** For a \((h + 1)\)-layer NN function denoted by \( g(x) \), for simplicity, let

\[
\begin{align*}
K_t^{h+1}(x, \tilde{x}) &= K_t^{h+1}(x, \tilde{x}; \{\Theta^i\}_{l=0,\ldots,h+1}), \quad (5.1) \\
K_t^h(z, \tilde{z}) &= K_t^h(z, \tilde{z}; \{\Theta^i\}_{l=1,\ldots,h+1}). \quad (5.2)
\end{align*}
\]

With gradient flow training process, we have the following recursive representation of the corresponding kernel matrix

\[
K_t^{h+1}(x, \tilde{x}) = K_t^h(o^1(x), o^1(\tilde{x})) + \sum_{i=1,j=1}^{L_0,L_1} \frac{\partial g(x)}{\partial \Theta_{i,j}^0} \frac{\partial g(\tilde{x})}{\partial \Theta_{i,j}^0}.
\]

Here the kernel matrix is always positive semidefinite.

6 Proofs

6.1 Main Results

**Proof of Theorem 4.** 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 \((3.3)\), then we claim that for any \( p \in L^2_{[\rho, \infty]} \), one must have

\[
\langle f_\epsilon - \hat{g}, \mathcal{T}^*p \rangle_\mu = 0, \quad \forall p \in L^2_{[\rho, \infty]}. \tag{6.1}
\]

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

\[
\|f_\epsilon - \hat{g}_\epsilon\|_\mu^2 = \|f_\epsilon - \hat{g}\|_\mu^2 - 2\epsilon \langle f_\epsilon - \hat{g}, \mathcal{T}^*p \rangle_\mu + \epsilon^2 \|\mathcal{T}^*p\|_\mu^2. \tag{6.2}
\]

For \( \epsilon \) with the same sign as \( \langle f_\epsilon - \hat{g}, \mathcal{T}^*p \rangle_\mu \neq 0 \) and small enough, one can see that \( \|f_\epsilon - \hat{g}_\epsilon\|_\mu^2 < \|f_\epsilon - \hat{g}\|_\mu^2 \) 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 (6.1) holds, in other words,

\[
\langle \mathcal{T}^*g, p \rangle_{[\rho, \infty]} = \langle f_\epsilon - \hat{g}, \mathcal{T}^*p \rangle_\mu = 0 \tag{6.3}
\]

From PDE characterization \((2.9)\) with ReLU activation, one knows that

\[
\begin{align*}
V(\Theta) &= \mathbb{E}[y\sigma(x^T\Theta)] = \mathbb{E}[f_\epsilon(x)\sigma(x^T\Theta)] \\
U(\Theta, \hat{\Theta}) &= -\mathbb{E}[\sigma(x^T\Theta)\sigma(x^T\hat{\Theta})],
\end{align*}
\]

16
and the expression for the velocity field
\[
 \|\Theta\| \left( \nabla_\Theta V(\Theta) + \nabla_\Theta \int U(\Theta, \hat{\Theta})\|\rho_t(d\hat{\Theta})\| \right) \\
= \|\Theta\| \left( \int f_*(x) x1_{xT\Theta>0} \mu(dx) - \int \int x1_{xT\Theta>0} \sigma(x^T\hat{\Theta})\|\rho_x(d\hat{\Theta})\| \mu(dx) \right).
\]

We know that any stationary point \((\rho_+, \rho_-)\) has the following property (Mei et al., 2018):
\[
\text{supp}(\rho_x) \subseteq \left\{ \Theta : \int f_*(x) x1_{xT\Theta>0} \mu(dx) = \int \int \|\sigma(x^T\Theta)\|\|\rho_x(d\hat{\Theta})\| \mu(dx) \right\}. \tag{6.4}
\]

Multiplying both sides by \(\|\Theta\|\Theta^T\) and recall the property of ReLU, the above condition implies that for all \(\Theta \in \text{supp}(\rho_x)\), we have
\[
\int f_*(x)\|\Theta\|\sigma(x^T\Theta) \mu(dx) = \int \int \|\Theta\|\sigma(x^T\Theta)\|\rho_x(d\hat{\Theta})\| \mu(dx).
\] (6.5)

One can see the stationary condition on \(\rho_x\) (fixed points of the dynamics) (6.5) translates to
\[
\mathcal{T} f_*(\Theta) = \left( \mathcal{T} \mathcal{T}^* \frac{d\rho_x}{d|\rho_x|} \right)(\Theta), \quad \forall \Theta \in \text{supp}(\rho_x). \tag{6.6}
\]

Here the function \(\frac{d\rho_x}{d|\rho_x|}\) is the Radon-Nikodym derivative. In addition, one can easily verify that, as \(\rho_x\) has bounded total variation
\[
\frac{d\rho_x}{d|\rho_x|} \in L^2_{|\rho_x|}.
\]

Therefore, combining all above, one knows that
\[
f_x(x) = \int \|\Theta\|\sigma(x^T\Theta) \rho_x(d\Theta) = \mathcal{T}^* \frac{d\rho_x}{d|\rho_x|} \in \mathcal{H}_{\mathcal{X}}
\]
and that for any \(p \in L^2_{|\rho_x|}\)
\[
\langle f_* - f_x, \mathcal{T}^* p \rangle_\mu = \langle \mathcal{T} (f_* - f_x), p \rangle_{|\rho_x|} = \langle \mathcal{T} f_* - \mathcal{T} \mathcal{T}^* \frac{d\rho_x}{d|\rho_x|}, p \rangle_{|\rho_x|} = \int \left( \mathcal{T} f_* - \mathcal{T} \mathcal{T}^* \frac{d\rho_x}{d|\rho_x|} \right)(\Theta) |\rho_x|(d\Theta) = 0 \text{ due to (6.6)} \tag{6.7}
\]

We have proved that \(f_x = \mathcal{T}^* \frac{d\rho_x}{d|\rho_x|}\) satisfies normal condition for being a minimizer to (3.3). \(\square\)

**Proof of Theorem 3.** Let’s first show that in the infinite neuron limit \(m \to \infty\), \(\rho_{+,t}, \rho_{-,t}\) are properly defined. Therefore Eqn. (2.9) in the above theorem also characterize the distribution dynamics for infinite neurons NN, induced by gradient flow training. For simplicity, we assume the initialization \(\rho_{+,0}, \rho_{-,0}\) with bounded support. We add the superscript \(m, \rho_{+,t}^m, \rho_{-,t}^m, \rho_t^m\) to (2.6) to indicate their
dependence on \(m\). Consider \(\nabla_\Theta V, \nabla_\Theta U(\Theta, \tilde{\Theta})\) in (2.8) 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 \(m \to \infty\), the initial distribution \(\rho_0^{m} \xrightarrow{d} \tilde{\rho}_0 = \rho_{+,0} - \rho_{-,0}\) by law of large number, and by the solution’s continuity depending on the initial value. Therefore we have \(\rho_t^{m} \xrightarrow{d} \rho_t\) as \(m \to \infty\) well defined.

The velocity of a particle \(\Theta\) in the positive part as a rewrite of (2.4)-(2.5) is

\[
\mathcal{V}(\Theta, \rho_t) = \|\Theta\| \left( \nabla_\Theta V(\Theta) + \nabla_\Theta \int U(\Theta, \tilde{\Theta}) \|\rho_t(d\tilde{\Theta})\| \right),
\]

resp. for the negative part and (2.5), we have

\[
-\mathcal{V}(\Theta, \rho_t) = -\|\Theta\| \left( \nabla_\Theta V(\Theta) + \nabla_\Theta \int U(\Theta, \tilde{\Theta}) \|\rho_t(d\tilde{\Theta})\| \right).
\]

Given the velocity of particle, we have the transport equation for gradient flow,

\[
\partial_t \rho_{+,t} = -\nabla_\Theta \cdot (\rho_{+,t} \cdot \mathcal{V}(\Theta, \rho_t)),
\]

\[
\partial_t \rho_{-,t} = -\nabla_\Theta \cdot (-\rho_{-,t} \cdot \mathcal{V}(\Theta, \rho_t)).
\]

To see this, recall the definition of weak derivative \(\partial_t \rho_t\): for any bounded smooth function \(g\), \(\partial_t \rho_t\) is defined in the following sense

\[
d \cdot \int g \rho_t = -\int g \partial_t \rho_t \cdot dt.
\]

We take any bounded smooth function \(g(\Theta)\), given the velocity of \(\Theta\)’s, then we have

\[
-\int g \partial_t \rho_t \cdot dt = d \cdot \int g(\Theta) \rho_{+,t}(\Theta) = \int \nabla g(\Theta) \cdot \mathcal{V}(\Theta, \rho_t) \rho_{+,t}(\Theta) \cdot dt,
\]

and \(\rho_{-,t}\) correspondingly. By the weak derivative, we get the above PDE. We use the above dynamic description as the training process for infinite neuron NN. Plug above equation into \(\rho_t = \rho_{+,t} - \rho_{-,t}\) and \(|\rho_t| = \rho_{+,t} + \rho_{-,t}\), we get

\[
\partial_t \rho_{t}(\Theta) = -\nabla_\Theta \cdot (|\rho_t| (\Theta) \mathcal{V}(\Theta, \rho_t)),
\]

\[
\partial_t |\rho_t| (\Theta) = -\nabla_\Theta \cdot (\rho_t(\Theta) \mathcal{V}(\Theta, \rho_t)).
\]

Proof of Theorem 5. The first inequality in (3.8) is trivial. For the second inequality, it suffices to show for any \(c = (c_1, \ldots, c_p)^T, x_1, \ldots, x_p, \Theta\), we have

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

The RHS equals

\[
\sum_{i,j} c_i c_j 1_{x_i^T \Theta > 0} 1_{x_j^T \Theta > 0} = \left( \sum_{i} c_i x_i^T \Theta \right)^2 \geq \left( \sum_{i} c_i 1_{x_i^T \Theta > 0} \right)^2 = LHS.
\]
For the last inequality, with compactness condition on \( \rho_x \), we have
\[
\sum_{i,j} c_ic_j \int \| \theta \|^2 \sigma(x_j^T \theta) \sigma(x_j^T \theta) |\rho_x|(\theta) \leq D^2 \sum_{i,j} c_ic_j \int \sigma(x_i^T \theta) \sigma(x_j^T \theta) |\rho_x|(\theta).
\]
(6.19)

Therefore, \( D^2 K_x^{(1)} \geq H_x \).

\[ \square \]

**Proof of Theorem 6.** Let us rewrite Corollary 3.1 into
\[
\frac{d}{dt} \| \Delta_t \|^2_{\mu} = -2\langle \Delta_t, K_t \Delta_t \rangle_{\mu} = -2\| K_t^{1/2} \Delta_t \|^2_{\mu},
\]
(6.20)

here \( K_t : L^2_{\mu}(x) \to L^2_{\mu}(x) \) denotes the integral operator associated with \( K_t \),
\[
\langle K_t f, g \rangle(x) := \int K_t(x, \tilde{x}) f(\tilde{x}) \mu(\tilde{x}).
\]
(6.21)

From (6.20)
\[
\frac{d}{dt} \| \Delta_x \|^2_{\mu} = -2\| K_x^{1/2} \Delta_x \|^2_{\mu},
\]
(6.22)

we know that the RHS equals zero implies
\[
\| K_x^{1/2} \Delta_x \|^2_{\mu} = 0 \quad \langle K_x^{1/2} g, \Delta_x \rangle_{\mu} = \langle g, K_x^{1/2} \Delta_x \rangle_{\mu} = 0, \quad \forall g \in L^2_{\mu}.
\]

This further implies \( \Delta_x \) lies in the kernel of RKHS \( K_x \) as \( K_x = \{ K_x^{1/2} g : g \in L^2_{\mu} \} \).
\[ \square \]

### 6.2 Supporting Results

**Proof of Lemma 2.1.** First we write down the dynamic of prediction \( f(\tilde{x}) \) at each point \( \tilde{x} \) based on Eqn. (1.3). For notational simplicity, let \( u_j, w_j \) be \( u_j(t), w_j(t) \), and let \( \sigma_j(\tilde{x}) = \sigma(u_j^T \tilde{x}) \), and with the square loss \( \ell(y, f) = \frac{1}{2}(y - f)^2 \), we have
\[
\frac{df_j(\tilde{x})}{dt} = \sum_{j=1}^m \left[ \frac{dw_j}{dt} \sigma_j(\tilde{x}) + w_j \frac{d\sigma_j(\tilde{x})}{dt} \right]
\]
(6.23)

\[
= \sum_{j=1}^m \left\{ \mathbb{E}_x \left[ (y - f_t(x)) \sigma(x^T u_j) \right] \sigma_j(\tilde{x}) + w_j \mathbb{1}_{\tilde{x}^T u_j \geq 0} \tilde{x}^T \mathbb{E}_x \left[ (y - f_t(x)) w_j \mathbb{1}_{x^T u_j \geq 0} \right] \right\}
\]
(6.24)

\[
= \sum_{j=1}^m \left\{ \mathbb{E}_x \left[ (f_s(x) - f_t(x)) \left( \sigma(\tilde{x}^T u_j) \sigma(x^T u_j) + w_j^2 \mathbb{1}_{\tilde{x}^T u_j \geq 0} \tilde{x}^T \mathbb{1}_{x^T u_j \geq 0} \tilde{x} \right) \right] \right\}
\]
(6.25)

\[
= \mathbb{E}_x \left\{ \sum_{j=1}^m \left[ \sigma(\tilde{x}^T u_j) \sigma(x^T u_j) + w_j^2 \mathbb{1}_{\tilde{x}^T u_j \geq 0} \mathbb{1}_{x^T u_j \geq 0} \tilde{x}^T \mathbb{1}_{x^T u_j \geq 0} \tilde{x} \right] (f_s(x) - f_t(x)) \right\}
\]
(6.26)

\[
= \mathbb{E}_x \left[ K_t(\tilde{x}, x) \Delta_t(x) \right].
\]
(6.27)
Therefore, we have
\[
\frac{dE_x}{dt} \left[ \frac{1}{2} \Delta_t(x)^2 \right] = -E_x \left[ (f_u(x) - f_i(x)) \frac{df_i(x)}{dt} \right] \tag{6.28}
\]
\[
= -E_x \left[ \Delta_t(x) E_x \left[ K_t(x, \bar{x}) \Delta_t(\bar{x}) \right] \right] \tag{6.29}
\]
\[
= -E_{x, \bar{x}} \left[ \Delta_t(x) K_t(x, \bar{x}) \Delta_t(\bar{x}) \right]. \tag{6.30}
\]

\textbf{Proof of Lemma 2.2.} We know
\[
E_{u \sim \pi} \left[ \sigma(u^T x) \sigma(u^T \bar{x}) \right] = E_{u \sim \pi} \left[ \bar{x}^T uu^T \mathbb{1}_{u^T x > 0} \mathbb{1}_{u^T \bar{x} > 0} \right] x \tag{6.31}
\]
Consider the coordinate system $e_1, e_2, \ldots e_d$ such that $e_1, e_2$ spans the space of $x, \bar{x}$, with
\[
x = e_1, \quad \bar{x} = \cos \theta \cdot e_1 + \sin \theta \cdot e_2, \tag{6.32}
\]
where $\theta = \arccos(x^T \bar{x})$. Note $u = [v_1, v_2, \ldots v_d]$ is still an isotropic Gaussian under this coordinate system. The constraint reads
\[
\mathbb{1}_{u^T x > 0} \mathbb{1}_{u^T \bar{x} > 0}, \tag{6.33}
\]
equivalent to $v_1 > 0, v_1 \cos \theta + v_2 \sin \theta > 0, \tag{6.34}$
and one can see that $v_2, \ldots v_d$ integrate out.
Let’s focus on the spherical coordinates of $v_1 = r \cos \phi, v_2 = r \sin \phi$, then $r^2 \sim \chi^2(2)$ and $\phi \sim U[-\pi, \pi]$. W.l.o.g., we can consider the case when $\theta \in [0, \pi]$.
\[
E_{u \sim \pi} \left[ uu^T \mathbb{1}_{u^T x > 0} \mathbb{1}_{u^T \bar{x} > 0} \right] x
\]
\[
= E[r^2] \left( e_1 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \cos^2 \phi \mathbb{1}_{\phi \in [\theta - \pi/2, \pi/2]} d\phi + e_2 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \cos \phi \sin \phi \mathbb{1}_{\phi \in [\theta - \pi/2, \pi/2]} d\phi \right)
\]
because the above are equivalent to $e_1 E[v_1^2 \mathbb{1}_{u^T x > 0} \mathbb{1}_{u^T \bar{x} > 0}] + e_2 E[v_1 v_2 \mathbb{1}_{u^T x > 0} \mathbb{1}_{u^T \bar{x} > 0}]$
\[
= 2 \cdot \frac{1}{2\pi} \left[ e_1 \cdot \frac{\pi - \theta}{2} + (e_1 \cos \theta + e_2 \sin \theta) \cdot \frac{\sin \theta}{2} \right]
\]
just evaluate $\int_{-\pi/2}^{\pi/2} \cos^2 \phi d\phi, \int_{-\pi/2}^{\pi/2} \cos \phi \sin \phi d\phi$
\[
= \frac{\pi - \theta}{2\pi}, \quad \frac{\sin \theta}{2\pi}.
\]
Therefore, we get
\[
E_{u \sim \pi} \left[ uu^T \mathbb{1}_{u^T x > 0} \mathbb{1}_{u^T \bar{x} > 0} \right] x
\]
\[
= \frac{\pi - \theta}{2\pi} \cos \theta + \frac{\sin \theta}{2\pi}.
\]
Similarly, we have
\[
 E_{\tilde{x}} \tilde{x}^T [I_{u^T} \geq 0 I_{\tilde{u}^T} \geq 0] x = \frac{\pi - \theta}{2\pi} \cos \theta. \tag{6.35}
\]

Summing them up, we get the result.

**Proof of Proposition 2.1.** It suffices to show \( \theta^2_{+,i}(t) = \|\Theta_{+,i}(t)\|^2 \) and resp. \( \theta^2_{-,i}(t) = \|\Theta_{-,i}(t)\|^2 \). By our path dynamics, we have
\[
\frac{d\theta^2_{+,i}}{dt} = 2\theta_{+,i} \frac{d\theta_{+,i}}{dt} = -2E_x \left[ \frac{\partial \ell(y, f(x))}{\partial f} \theta_{+,i} \sigma(x^T \Theta_{+,i}) \right], \tag{6.36}
\]
\[
\frac{d\left\|\Theta_{+,i}\right\|^2}{dt} = 2\Theta^T_{+,i} \frac{d\Theta_{+,i}}{dt} = -2E_x \left[ \frac{\partial \ell(y, f(x))}{\partial f} \theta_{+,i} 1_{x^T \Theta_{+,i} \geq 0} x^T \Theta_{+,i} \right] = \frac{d\theta^2_{+,i}}{dt}. \tag{6.37}
\]
Thus, by the initialization, we have \( \theta_{+,i}(t) = \|\Theta_{+,i}(t)\| \), and resp. \( \theta_{-,i}(t) = -\|\Theta_{-,i}(t)\| \).

**Proof of Proposition 2.2.** Using \( w_j(t_0), u_j(t_0) \), for \( j \neq i \), as an initial value for ODE (1.3) without the \( i \)-th node. By assumption, we have a solution of this \( 2 \cdot (2m - 1) \)-dimensional initial value problem. Then padding the solution with \( u_i \equiv 0 \) and \( w_i \equiv 0 \), which can be a solution for ODE (1.3) with \( i \)-th neuron included.

**Proof of Corollary 3.1.** Our proof essentially follows the same steps for (2.1). First, we write down the dynamic of \( f_t(x) \),
\[
\frac{df_t(x)}{dt} = \int \|\Theta\| \sigma(x^T \Theta) \rho_t(d\Theta). \tag{6.38}
\]
Plug-in the training dynamic (6.15), we get
\[
\frac{df_t(x)}{dt} = -\int -\nabla_\Theta \left[ \|\Theta\| \sigma(x^T \Theta) \right] \cdot V(\Theta, \rho_t)(d\Theta) \tag{6.39}
\]
\[
= \int \nabla_\Theta \left[ \|\Theta\| \sigma(x^T \Theta) \right] \cdot \|\Theta\| \left\{ E_{\tilde{x}} \left[ f_\tilde{x}(\tilde{x}) 1_{\tilde{x}^T \Theta \geq 0} \tilde{x} \right] - E_{\tilde{x}} \left[ \left( \|\Theta\| \sigma(\tilde{x}^T \tilde{\Theta}) 1_{\tilde{x}^T \tilde{\Theta} \geq 0} \tilde{x} \right) \rho_t(d\Theta) \right] \right\} \rho_t(d\Theta)
\]
\[
= E_{\tilde{x}} \left\{ \int \nabla_\Theta \left[ \|\Theta\| \sigma(x^T \Theta) \right] \cdot \|\Theta\| \left[ \Delta_t(\tilde{x}) 1_{\tilde{x}^T \Theta \geq 0} \tilde{x} \right] \rho_t(d\Theta) \right\}
\]
\[
= E_{\tilde{x}} \left\{ \Delta_t(\tilde{x}) \cdot \int \|\Theta\|^2 1_{x^T \Theta \geq 0} 1_{\tilde{x}^T \Theta \geq 0} \tilde{x} \sigma(x^T \Theta) \sigma(\tilde{x}^T \Theta) \rho_t(d\Theta) \right\}. \tag{6.40}
\]
Therefore, we have
\[
\frac{dE_{\tilde{x}} \left[ \frac{1}{2} \Delta_t(x)^2 \right]}{dt} = -E_{\tilde{x}, \tilde{x}} [\Delta_t(x) K_t(x, \tilde{x}) \Delta_t(\tilde{x})]. \tag{6.42}
\]
Proof of Proposition 5.1. For notational simplicity, let $K_{t}^{h+1}(x,\tilde{x}) = K_{t}^{h+1}(x,\tilde{x};\{\Theta^{l}\}_{l=0,\ldots,h+1})$, and $K_{t}^{h}(z,\tilde{z}) = K_{t}^{h}(z,\tilde{z};\{\Theta^{l}\}_{l=1,\ldots,h+1})$.

For the proof, we calculate the dynamic of prediction $g(x)$, by elementary calculus, we have

$$\frac{dg(x)}{dt} = -\mathbb{E}_{x}[f_{*}(x) - g(x)] \left[ \sum_{\Theta} \frac{\partial g(x)}{\partial \Theta} \cdot \frac{\partial g(x)}{\partial \Theta} \right]. \tag{6.43}$$

With same calculation for the dynamic of $\Delta_{t}$ as in (6.28), we get

$$K_{t}^{h+1}(x, x') = \sum_{\Theta \in \Theta^{0}} \frac{\partial g(x)}{\partial \Theta} \cdot \frac{\partial g(x')}{\partial \Theta} + \sum_{\text{other } \Theta} \frac{\partial g(x)}{\partial \Theta} \cdot \frac{\partial g(x')}{\partial \Theta}. \tag{6.44}$$

By induction, we get

$$K_{t}^{h+1}(x, \tilde{x}) = K_{t}^{h}(o^{l}(x), o^{l}(\tilde{x})) + \sum_{i=1, j=1}^{L_{0}L_{1}} \frac{\partial g(x)}{\partial \Theta_{i,j}^{0}} \cdot \frac{\partial g(\tilde{x})}{\partial \Theta_{i,j}^{0}}. \tag{6.45}$$

Now, we prove the positive semi-definiteness of the kernel. By induction, we only need to prove that the second term above is non-negative. We construct a canonical mapping $\phi_{h+1}(x) := v(x), \mathbb{R}^{d} \rightarrow \mathbb{R}^{L_{0} \times L_{1}}$, whereas the $i, j$-th coordinate $v(x)_{i,j} = \frac{\partial g(x)}{\partial \Theta_{i,j}^{0}}$. Then the second term can be seen as an inner product $\langle \phi_{h+1}(x), \phi_{h+1}(\tilde{x}) \rangle$, which implies the non-negativity. \qed

Acknowledgement

We thank Maxim Raginsky for pointing out relevant references, and for providing helpful discussion.

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.

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.
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, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. arXiv preprint arXiv:1806.00900, 2018a.

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

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.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

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. arXiv preprint arXiv:1808.00387, 2018.

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.

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.

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.

Li Wenliang, Dougal Sutherland, Heiko Strathmann, and Arthur Gretton. Learning deep kernels for exponential family densities. *arXiv preprint arXiv:1811.08357*, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.