

A function space analysis of finite neural networks with insights from sampling theory

Raja Giryes

Abstract—This work suggests using sampling theory to analyze the function space represented by neural networks. First, it shows, under the assumption of a finite input domain, which is the common case in training neural networks, that the function space generated by multi-layer networks with non-expansive activation functions is smooth. This extends over previous works that show results for the case of infinite width ReLU networks. Then, under the assumption that the input is band-limited, we provide novel error bounds for univariate neural networks. We analyze both deterministic uniform and random sampling showing the advantage of the former.

I. INTRODUCTION

Recently, it has been shown that neural networks with a univariate output and bounded weights perform a smooth interpolation between their training data [1], [2], [3]. These works provide an extension to many recent results that have studied the approximation power of neural networks. While in the general universal approximation theory, either in the infinite width case [4], [5] or the finite width case [6], it is shown that virtually any function may be approximated, the new results demonstrate that by adding constraints on the network weights, we get a smaller function space although the width of the network is infinite.

An interesting question that may arise as a follow up to these works that focused on the *approximation power* of the network is whether we may use their results to get new *estimation error* bounds for networks trained on n data samples. One intriguing phenomenon of neural networks is that for “natural good data” they both overfit the training data and generalize well at the same time, while for random “bad” data they just perform memorization with no generalization [7]. This phenomenon hints that the generalization of the network depends also on the structure of the input data and not only on the network parameters.

In this work, we focus on the case of data that is generated by band-limited functions. We show for a univariate neural network that when it overfits the training data, its error scales as $O(1/n^3)$, where n is the size of the training data. Our result provides a concrete example where the memorization of the network helps its generalization. Note that it naturally excludes the case of random data, which have an infinite bandwidth. As we shall see hereafter in the proof of our results, the fact that the network fits all the training examples is a key element in its ability to get low error for all the other points of the function that generated the data.

The contribution of our work is twofold. First, we show that the function represented by a finite width network with bounded weights have a bounded total variation of its first

derivative, i.e., $\int_{-\pi}^{\pi} f''(x)dx < \infty$, where $[-\pi, \pi]$ is assumed to be the input domain. This shows that finite multi-layer networks perform a smooth interpolation of their training data. This extends over previous works that have been limited to a single layer and infinite width. The second is providing generalization results both for infinite width networks and finite width ones. We use tools from sampling theory to analyze the error of the network both in the case of deterministic uniform sampling (Theorem 4) and the more realistic case of random sampling (Theorem 5).

II. RELATED WORK

A relationship between network representation and a given function space was shown in [8], [9]. In particular, these works focused on the ridgelet transform. The first studied the approximation power of networks with some special activation function using ridgelets. The second presented a connection between neural networks with ReLU activation and the ridgelet transform. They demonstrated that such networks satisfy the universal approximation property. Another line of works showed that networks learn first lower frequencies in the data [10], [11].

The works in [1], [2], [3] have shown that shallow infinite width networks with bounded weights perform a smooth (spline) interpolation of the training data. Another connection between neural networks and splines was exhibited in [12]. It focused on the specific case of max affine splines and used them to show a relationship between template matching and networks.

A connection between adding a regularization on the weights of the network and their generalization was shown in various works. While classic generalization error bounds for neural networks presented a dependency on the number of parameters in the network [13], Rademacher complexity (RC) based analysis showed that by bounding the norm of the weights, the generalization error is independent of the network width [14], [15]. Yet, the deficiency of these bounds is their independence of the input data; thus, they do not capture cases such as overfitting of random data [7].

Margin based approaches, which take into account also the input distribution, mitigate this issue [16], [17]. Note that “ ℓ_2 regularization does not significantly impact margins or generalization” [17], where the analysis here depends on the consequence of this regularization. Thus, these approaches are complementary to our analysis. Bounding the weights is also shown useful under the kernel (RKHS) assumption [18], which is not required in our work.

III. NEURAL NETWORKS AND SAMPLING THEORY PRELIMINARIES

This section surveys some preliminaries of neural networks and sampling theory. Readers that are familiar with these topics may skip to the next section.

Any neural network training relies on a given input dataset $\{(x_i, y_i)\}_{i=0}^{n-1}$ with n pairs of data sample x_i and label y_i . In general, the input space of a neural network is limited, i.e., x is sampled just from a specific interval of interest (for example, in images the pixel values are only in the range $[0, 255]$). Without loss of generality, we will assume for the simplicity of the presentation that $x \in [-\pi, \pi]$. In this case, we can arbitrarily define the values of $f(x)$ outside this interval (we do not sample the function there and therefore it does not affect the data generation and the network trained). We specifically select a periodic continuation of f such that $f(x) = f(x+2\pi)$.

Since we assume that f is bandlimited, then f must be also smooth and thus this assumption implies that $f(-\pi) = f(\pi)$. Notice that this assumption does not limit us in any way as if this is not the case, there are various ways to mitigate this issue. For example, in the case that $f(x) - f(x+2\pi)$ is not too large, we may extend the function a bit beyond $x = 2\pi$ in a smooth way such that it will remain band limited and satisfy the periodicity assumption. Another popular alternative is using a symmetric expansion of f (copying a mirrored version of f in the interval $[-\pi, \pi]$ to the interval $[\pi, 3\pi]$, which enforces having $f(-\pi) = f(3\pi)$ due to the mirroring) before applying the periodic extension. This just changes the integral limits when calculating the Fourier coefficients of f and requires replacing the DFT (which we use hereafter) with DCT [19].

Since f is periodic, we may calculate its Fourier coefficients

$$c_k = \frac{1}{(2\pi)^d} \int_{\|x\|_\infty \leq \pi} f(x) e^{-jx^T k} dx, \quad (1)$$

where $k \in \mathbb{Z}^d$. If f is bandlimited (also known as trigonometric polynomial [20]) then $c_k = 0$ if $\exists i$ such that $k[i] > K$. Thus,

$$f(x) = \sum_{-K \leq k[i] \leq K} c_k e^{jx^T k}. \quad (2)$$

Note that we sum over all the combinations in which $|k[i]| \leq K$.

Using the sampling theorem for bandlimited periodic signals, we may recover $f(x)$ using just $n \geq N \triangleq (2K+1)^d$ samples $\{f(x_i)\}_{i=0}^{n-1}$. For completeness, and as it will help us later in the derivations, we briefly describe here this result.

Uniform sampling. We start with reconstruction using uniform sampling. Assume that our sample points are on the grid $[\frac{2\pi i_1}{2K+1}, \frac{2\pi i_2}{2K+1}, \dots, \frac{2\pi i_d}{2K+1}]$, where $i_l = 0, \dots, 2K$ for $l = 1, \dots, d$. In the one dimensional case ($d = 1$), we have

$$f(x_i) = \sum_{k=-K}^K c_k e^{\frac{j2\pi k i}{2K+1}}. \quad (3)$$

Denoting by c the vector that contains the Fourier coefficients in it and by y the vector that contains the values of $f(x_i)$, we may rewrite (3) as (see [21])

$$y = F^* c, \quad (4)$$

where $F \in \mathbb{C}^{N \times N}$ is the DFT matrix, whose columns (in 1D) are of the form $\{e^{j2\pi k \frac{i}{2K+1}}\}_{k=-K}^K$, and F^* is its conjugate transpose, which is also its inverse (up to a scale factor $1/N$) because the rows of F are orthogonal to each other. Notice that the same holds true for the multi-dimensional case ($d > 1$) and then F is simply the d -dimensional DFT (in this case, we can also cast c in a vector representation). Having this relationship, we can recover the vector c , and thus the whole function f , from y by computing $c = \frac{1}{N} F y$.

Oversampling. Notice that if the number of measurements that we have are $n > N$, then we still have the relationship in (4) but in this case $F \in \mathbb{C}^{N \times n}$ is a DFT (tight) frame, whose columns are of the form $\{e^{j2\pi k \frac{i}{n}}\}_{k=-K}^K$ (in 1d). Since the rows of F are orthogonal in this case as well (also for $d > 1$), we still have that $\frac{1}{n} F F^* = I$ and thus we can reconstruct the function f using $c = \frac{1}{n} F y$ as before.

Notice that due to the redundancy that we have in the measurements, we may use other DFT operators to reconstruct c . In particular, for any \tilde{N} and $n \geq \tilde{N} \geq (2K+1)^d$, we can simply pad c with zeros, which yields the relationship $y = F^* c$ for the DFT frame $F \in \mathbb{C}^{\tilde{N} \times n}$ (which is the standard DFT transform if $n = \tilde{N}$). As before, we can reconstruct the Fourier coefficients by $c = \frac{1}{n} F y$. We abuse notation here and elsewhere denoting by c also the padded representation. The use will be clear from the context.

Non-uniform sampling. In many cases, we get just a random (non-uniform) set of samples of the space. In this case, the set of input points $\{x_i\}_{i=0}^{n-1}$ do not lie on the grid but are randomly spread in $[-\pi, \pi]^d$. The sampled points obey

$$f(x_i) = \sum_{k=-K}^K c_k e^{k^T x_i}. \quad (5)$$

Writing (5) in a matrix form yields $f = Dc$, where the rows of D are $\{e^{jkx_i}\}_{k=-K}^K$ (in the 1D case). Notice that $D \in \mathbb{C}^{n \times N}$ is very similar to the DFT inverse transform (F^*) but with the difference that its rows correspond to random frequencies (that leads to the orthogonality property). Notice that also here we may pad c with zeros and thus have $D \in \mathbb{C}^{n \times \tilde{N}}$ in a similar way to the oversampling case. If $\tilde{N} = 2\tilde{K} + 1$ for some $\tilde{K} \geq K$ then the rows of D are $\{e^{jkx_i}\}_{k=-\tilde{K}}^{\tilde{K}}$ (in the 1D case).

If D has a full column rank (which is the case of many random sampling schemes [22], [23], [21]), i.e., invertible, then we may again reconstruct the function f by computing $c = D^\dagger y$, where $D^\dagger = (D^* D)^{-1} D^*$ is the pseudo-inverse of D . Although we can get perfect reconstruction also with random sampling, its disadvantage is the noisy case, where we get noise amplification that depends on the ratio κ between the largest and smallest non-zero singular value of D (the condition number of $D^* D$). This ratio is dependent also on the ratio between n and \tilde{N} [24], [25]. We use it hereafter to provide error bounds for neural networks with randomly sampled training data.

IV. THE FUNCTION SPACE OF BOUNDED FINITE NEURAL NETWORKS

The work in [1] proved that any function ϕ represented by a two layer overparameterized (with number of parameters going to infinity) ReLU network with univariate input and output has a bounded total variation in their first derivative as the bound on the network norm imposes a constraint on

$$\max \left(\int_x \phi''(x) dx, \phi'(\infty) + \phi'(-\infty) \right). \quad (6)$$

They have shown that this implies a spline interpolation (of at least order one, i.e., linear) between the training data, which the network overfitted (which is possible due to its overparameterization). The work in [2] have extended their results showing that the network performs a second order (cubic) spline interpolation between the data points under some assumption on the initial weights and the optimization process. The result of [1] have been extended in [3] to the case of multi-dimensional input. They have shown that in this case, the functions represented by the network have a bounded \mathcal{R} -norm, which is related to the Radon transform of the represented function.

Notice that the existing works [1], [2], [3] assume shallow networks with infinite width. We show here that under the assumption that the input domain is bounded (as is the common case with neural networks training), then neural networks with bounded norm approximate functions that have a bounded derivative and thus also total variation in the second derivative. Our proof technique much simpler than the ones presented in the earlier works.

Denote by σ_i the non-linearity in the network at the i th layer and by W_i and b_i the weights and biases there. Then, we may write a feed-forward network with L layers as

$$\phi(x) = \sigma_L(b_L + W_L \sigma_{L-1}(\dots \sigma_2(b_2 + W_2 \sigma_1(b_1 + W_1 x))).$$

If we denote by z_i the output of the i th layer, then we can write the above recursively as

$$z_i = \sigma_i(b_i + W_i z_{i-1}), \quad (7)$$

where $z_0 = x$ and $z_L = \phi(x)$. For such a network we prove the following proposition, which is an extension of the result in [16]

We rely on a result from [16] that shows the relationship $\left\| \frac{d\phi}{dx} \right\| \leq \prod_i \|W_i\|_F$. Yet, that work presents this result only for networks with ReLU, Sigmoid or hyperbolic tangent as non-linearity and without biases. The following proposition presents this result also for networks with biases and other non-expansive activation functions.

Proposition 1: Let $\phi(x)$ be a feed-forward network with an input x , non-expansive non-linear function σ_i and weights and biases $\{W_i\}_{i=1}^L$ and $\{b_i\}_{i=1}^L$. Then, we have

$$\left\| \frac{d\phi}{dx} \right\| \leq \prod_{i=1}^L \|W_i\|_F. \quad (8)$$

Proof. For calculating the Jacobian $\frac{d\phi}{dx}$, we may use the chain rule (as used in back-propagation), getting

$$\frac{d\phi}{dx} = \frac{d\phi}{dz_{L-1}} \frac{dz_{L-1}}{dz_{L-2}} \dots \frac{dz_2}{dz_1} \frac{dz_1}{dx}. \quad (9)$$

Thus, using matrix norm inequalities we have

$$\left\| \frac{d\phi}{dx} \right\| = \left\| \prod_{i=1}^L \frac{dz_i}{dz_{i-1}} \right\| \leq \prod_{i=1}^L \left\| \frac{dz_i}{dz_{i-1}} \right\|. \quad (10)$$

Now, notice that

$$\frac{dz_i}{dz_{i-1}} = \text{diag}(\sigma'_i(b_i + W_i z_{i-1})) W_i^T. \quad (11)$$

Since the spectral norm of the diagonal matrix $\text{diag}(\sigma'_i(b_i + W_i z_{i-1}))$ is its maximal value and as this value is smaller or equal to 1 (as we assume σ is non-expansive), we have that

$$\left\| \frac{dz_i}{dz_{i-1}} \right\| = \left\| \text{diag}(\sigma'_i(b_i + W_i z_{i-1})) W_i^T \right\| \leq \|W_i\|. \quad (12)$$

Plugging this inequality in (10) and then using the known relationship between the spectral and the Frobenius norms, we get the desired result. \square

To get a bound on the total variation of the second derivative we make the following simple observation: The discontinuities in the function approximated by the network are only due to the non-linear function in the network. Since the first derivative is bounded the ‘‘jumps’’ that occur in it are finite. Since we are dealing with a finite domain and a finite network, the number of such discontinuities is finite and therefore the integral over the second derivative is also finite (also known as the total variation of the first derivative).

Notice that in the case of infinite network and infinite domain, we cannot make the above assumptions and therefore a more sophisticated approach as the one in [1] is required to give a bound on the total variation of the first derivative. Yet, their work does not apply to the finite network case as does our result here. Notice that for shallow networks, which is the case studied in [1], [2], [3], the number of discontinuities in the network grows linearly with the width. In the deeper case, it grows faster (see analysis in [12], [26]) but is still bounded.

This provides us with the following corollary for finite neural networks with a univariate output.

Corollary 2: Let ϕ be a finite multi-layer neural network with non-expansive non-linearities that have a finite amount of discontinuities in their first derivative. Assume the training data is in the interval $[-\pi, \pi]^d$. Then the total variation of the derivative of this function, $\int_{x \in [-\pi, \pi]^d} \Delta \phi(x) dx$, is finite, where $\Delta \phi(x) = \nabla^2 \phi(x)$ is the Laplacian of $\phi(x)$.

Proof. Using Proposition 1 we have that all the partial derivatives of $\phi(x)$ are bounded in the domain $[-\pi, \pi]^d$. Since the network is finite and the discontinuities in the network derivative emerges from the non-linearities that have a finite amount of discontinuities in their first derivative, we have a finite amount of ‘‘jumps’’ in the interval $[-\pi, \pi]$ and all of them. The integral over the second derivative can be bounded by the difference between the largest and smallest first derivative of ϕ times the interval size plus the sum of the sizes of the jumps (as each is a delta function in the second derivative). As the first derivative and the amount of ‘‘jumps’’ are bounded, we have that $\int_{x \in [-\pi, \pi]^d} \nabla^2 \phi(x) dx$ is finite. \square

V. SAMPLING THEORY BASED ERROR BOUNDS

We turn now to use the above findings to prove that a neural network with bounded norms can recover band-limited functions with very high precision both with uniform and non-uniform sampling, where the latter is the more common case when getting a training data for a neural network. The underlying assumption in the analysis here is that the labels y_i are generated by a band limited function $f(x)$. We focus on univariate functions $f : \mathbb{R} \rightarrow \mathbb{R}$.

A periodic representation of the neural network function

Denote by $\tilde{\phi}_n : \mathbb{R} \rightarrow \mathbb{R}$ a function represented by a neural network that has bounded weights and is trained with n training samples. While this function is defined for all \mathbb{R} , for our data we are only interested in the output of the network in the domain $[-\pi, \pi]$. Therefore, for analyzing the network estimation error compared to the function $f(x)$ in this domain, we can change $\tilde{\phi}_n$ arbitrarily as we wish outside of this domain.

To be able to calculate a Fourier series, we define the function ϕ_n , which is equal to $\tilde{\phi}_n$ in the domain $[-\pi, \pi]$ and is periodic outside of it with a period 2π . Clearly, also in this case we may have that $\tilde{\phi}_n(x + 2\pi) \neq \tilde{\phi}_n(x)$. Yet, as we discussed in the preliminaries section, this can be leveraged, for example, by using a symmetric extension and then the same analysis that we present below will remain the same but with a DCT replacing the DFT used in the analysis. Since both are orthogonal, the derived results remain the same. Thus, for simplicity we just assume a regular periodic extension.

Given that ϕ_n is periodic we may calculate its Fourier series

$$\phi_n(x) = \sum_{k \in \mathbb{Z}} \zeta_k e^{jxk}, \quad (13)$$

where ζ_k is calculated as in (1) (with ϕ_n instead of f).

Now, assume that the network has overfitted the data, i.e., $f(x_i) = \phi_n(x_i)$, then if ϕ_n is band-limited as f , then we get from sampling theory that $f = \phi_n$. In the case of uniform sampling, if the network function ϕ_n was exactly a spline, we could have used the result in [27] to calculate the network error as a function of n . Yet, ϕ_n is not guaranteed to be band-limited and as shown in [1], [2], the connection between the points may be beyond "linear". Figure 1 provides an example of a trained network output, where we get different types of interpolations between the training points that are generated from a band-limited function. Therefore a more general error analysis is required.

To this end, we take the following strategy. First we show that since the network approximates smooth functions, then its spectrum decay fast. Then we use this to bound the error of the network for data that is generated from bandlimited mappings.

Spectral decay rate of networks with bounded weights

We introduce the following lemma that provide a bound on the decay rate of finite neural networks with bounded weights.

Lemma 3: Let $\phi_n(x)$ be a finite multi-layer neural network with non-expansive non-linearities that have a finite amount

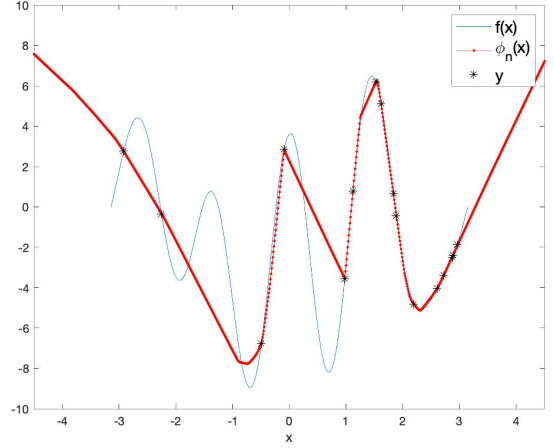


Fig. 1. Approximation of a band-limited function $f(x)$ using a neural network ϕ_n trained using only 16 training examples (y).

of discontinuities in their first derivative. Assume the training data is in the interval $[-\pi, \pi]$. Then the Fourier coefficients of $\phi_n(x)$, obeys

$$\zeta_k = O(|k|^{-2}). \quad (14)$$

Proof. According to Corollary 2, if the network has bounded weights then $\phi_n(x)$ has a finite total variation of the derivative of this function, i.e., $\int_{x \in [-\pi, \pi]} \phi_n''(x) dx$. Clearly, in this case also $\tilde{\phi}_n(x) = \phi(x)I_{[-\pi, \pi]}$ has a finite total variation. The indicator function $I_{[-\pi, \pi]}$ is one inside the domain $[-\pi, \pi]$ and zero outside of it.

Notice that $\phi_n(x)I_{[-\pi, \pi]^d} \in L^1$. One way to see this is using the fact that $\phi_n(x)$ is Lipschitz (as it has a bounded first derivative as shown in Lemma 1) and $[-\pi, \pi]$ is a finite domain. Thus, using a standard known result, the finite total variation in the first derivative implies that the Fourier transform $\hat{\phi}_n(w)$ of $\phi_n(x)I_{[-\pi, \pi]^d}$ obeys $\hat{\phi}_n(w) = O(|w|^{-2})$. Using the known relationship that the Fourier coefficients of ϕ_n (ζ_k) are equal (up to a constant) to the "sampled Fourier transform" $\hat{\phi}_n(k)$ yields the desired result. \square

Having the above decay rate for the Fourier coefficients of $\phi_n(x)$, we turn to bound the error between $\phi_n(x)$ and $f(x)$. We start with the case of uniform sampling and then move to the case of non-uniform sampling. We present both results for the univariate case. One may extend them to the multi-dimensional input case using a similar technique. We defer this to a future work.

Network error with uniform univariate samples

The next theorem shows that the network error in the uniform sampling case decreases as a function of $\frac{1}{n^3}$.

Theorem 4: If a univariate network has bounded weights, the training data of size $n \geq 2K + 1$ is fitted by the network

and it is uniformly sampled from a band-limited function with $2K + 1$ non-zero Fourier coefficients, then we have

$$\begin{aligned} \|f(x) - \phi_n(x)\|_{L^2[-\pi, \pi]}^2 &= \\ \int_{x=-\pi}^{\pi} (f(x) - \phi_n(x))^2 dx &= O(1/n^3), \end{aligned} \quad (15)$$

i.e., the error of the network scales as $O(1/n^3)$

The proof of this theorem is a special case of the one of Theorem 5 for non-uniform sampling, which is presented next.

Network error with random univariate samples

Having the result for the uniform sampling case, we move to study the random sampling case. Analyzing this case is more important as it resembles in a closer way the case of real data, where we get labels for randomly sampled inputs. We show in this case the rate of convergence is of the order of $\frac{1}{\tilde{N}^3}$ ($\tilde{N} \leq n$), where we assume that the random sampling pattern generates an operator $D \in \mathbb{C}^{n \times \tilde{N}}$ that is invertible with a condition number κ . Notice that this enables us to tradeoff the network error decay rate and the condition number of D . If $\tilde{N} = n$ we get the fastest decay rate but the condition number is very bad. Reducing \tilde{N} improves the condition number but slows down the decay rate. We discuss the case, which is equivalent to $x_i \sim U[-\pi, \pi]$ (i.e., sampling from a uniform distribution in the domain $[-\pi, \pi]$), after the proof of the theorem. We claim that in that random sampling case, the network error scales as $\frac{1}{n^3}$, like in the deterministic uniform sampling case.

Theorem 5: If a univariate network has bounded weights, the training data (x_i, y_i) of size $n \geq 2K + 1$ is randomly sampled from a band-limited function f with $2K + 1$ non-zero Fourier coefficients (i.e., $y_i = f(x_i)$), an operator $D \in \mathbb{C}^{n \times \tilde{N}}$ ($\tilde{N} \leq n$) that corresponds to the sampling pattern is invertible with a condition number κ , and the network ϕ_n fits the data, then with high probability

$$\|f(x) - \phi_n(x)\|_{L^2[-\pi, \pi]}^2 = O(\kappa^2/\tilde{N}^3). \quad (16)$$

Proof. Let $\tilde{N} = 2\tilde{K} + 1$ for $\tilde{K} \in \mathbb{Z}$.¹ From the Parseval identity and the fact that f is band-limited, we have

$$\begin{aligned} \|f(x) - \phi_n(x)\|_{L^2[-\pi, \pi]}^2 &= \sum_{k=-\infty}^{\infty} |c_k - \zeta_k|^2 \\ &= \sum_{k \leq \tilde{K}} |c_k - \zeta_k|^2 + \sum_{|k| > \tilde{K}} |\zeta_k|^2. \end{aligned} \quad (17)$$

To bound the network error, we need to bound the two terms in the rhs (right hand side) of the (17).

We start with the second term. Using Lemma 3, we have that $|\zeta_k| \leq a/|k|^2$ for some constant a . Thus,

$$\sum_{|k| > \tilde{K}} |\zeta_k|^2 \leq a \sum_{|k| > \tilde{K}} \frac{1}{|k|^4} = O\left(\frac{1}{\tilde{K}^3}\right) = O\left(\frac{1}{\tilde{N}^3}\right) \quad (18)$$

¹This assumption is used just for the simplicity of the presentation to perform a symmetric expansion of c . If n is even we can just perform a non-symmetric expansion.

where the first equality follows from the decay rate of the sum $\sum_{|k| > \tilde{K}} \frac{1}{|k|^4}$. Plugging (18) in (17) leads to

$$\|f(x) - \phi_n(x)\|_{L^2[-\pi, \pi]}^2 \leq \sum_{k \leq \tilde{K}} |c_k - \zeta_k|^2 + O\left(\frac{1}{\tilde{N}^3}\right). \quad (19)$$

Turning to bound the first term in the rhs of (19), notice that from the assumption that the network fitted the training data, we have $f(x_i) = \phi_n(x_i)$ for $1 \leq i \leq n$. Using the Fourier series expansion of $\phi_n(x)$, we have that

$$\phi_n(x_i) = \sum_{k \in \mathbb{Z}} \zeta_k e^{jkx_i} = \sum_{|k| \leq \tilde{K}} \zeta_k e^{jkx_i} + \sum_{|k| > \tilde{K}} \zeta_k e^{jkx_i}. \quad (20)$$

Denote by y the vector whose i th entry is $\phi_n(x_i)$, D the operator that contains $\{e^{jkx_i}\}_{k=-\tilde{K}}^{\tilde{K}}$ in its rows, ζ the vector containing the coefficients ζ_k , $k \leq \tilde{K}$, and $y_{\setminus \tilde{K}} \in \mathbb{C}^n$ the vector whose i th entry is equal to $\sum_{|k| > \tilde{K}} \zeta_k e^{jkx_i}$. With this notation, we may write (20) in a vector form

$$y = D\zeta + y_{\setminus \tilde{K}}, \quad (21)$$

Denote by ζ^l the vector that contains the set of coefficients $\zeta_{-\tilde{K}+l\tilde{N}}, \dots, \zeta_{\tilde{K}+l\tilde{N}}$. Notice that each coefficient in ζ^l is multiplied in $y_{\setminus \tilde{K}}$ by the same complex exponent as in the multiplication between D and ζ but with a factor $e^{jl\tilde{N}x_i}$. Thus, by denoting $L_l = \text{diag}(e^{j\tilde{N}lx_1}, \dots, e^{j\tilde{N}lx_n})$, the diagonal matrix that contains these exponent factors, we may write $y_{\setminus \tilde{K}} = \sum_{l \neq 0} L_l D \zeta^l$. Using the assumption that D is invertible and $y = Dc$, we get from (21) that

$$c = \zeta + \sum_{l \neq 0} D^\dagger L_l D \zeta^l. \quad (22)$$

Notice that $\|c - \zeta\|_2^2 = \sum_{k \leq \tilde{K}} |c_k - \zeta_k|^2$, which is exactly the term we want to bound in (19). From (22), we have

$$\begin{aligned} \|c - \zeta\|_2^2 &= \left\| \sum_{l \neq 0} D^\dagger L_l D \zeta^l \right\|_2^2 \\ &= \sum_{l \neq 0} \|D^\dagger L_l D \zeta^l\|_2^2 + \sum_{q \neq l, 0} \sum_{l \neq 0} (D^\dagger L_l D \zeta^l)^* D^\dagger L_l D \zeta^q \\ &\leq \sum_{l \neq 0} \|D^\dagger L_l D \zeta^l\|_2^2 + \sum_{q \neq l, 0} \sum_{l \neq 0} \|D^\dagger L_l D \zeta^l\| \|D^\dagger L_l D \zeta^q\|_2 \\ &\leq \kappa^2 \sum_{l \neq 0} \|\zeta^l\|_2^2 + \kappa^2 \sum_{q \neq l, 0} \sum_{l \neq 0} \|\zeta^l\| \|\zeta^q\|_2, \end{aligned} \quad (23)$$

where we use the Cauchy Schwartz inequality in the second step, and matrix norm inequalities in the last step, namely,

$$\|D^\dagger L_l D \zeta^l\|_2 \leq \|D^\dagger\|_2 \|L_l\|_2 \|D\|_2 \|\zeta^l\|_2 \quad (24)$$

with the fact that $\|D^\dagger\|_2 = 1/\sigma_{\min}(D)$, $\|D\|_2 = \sigma_{\max}(D)$, $\|L_l\|_2 = 1$ and $\kappa = \sigma_{\max}(D)/\sigma_{\min}(D)$.

We turn to bound the terms at the rhs of (23). For the first, we have that $\sum_{l \neq 0} \|\zeta^l\|_2^2 = O\left(\frac{1}{\tilde{N}^3}\right)$ as in (18). For the second term, from Lemma 3, we have that $\|\zeta^l\|_2$ and $\|\zeta^q\|_2$ behave as $\sqrt{\frac{n}{(nl)^4}} = \frac{1}{\sqrt{nnl^2}}$ and $\frac{1}{\sqrt{nnq^2}}$ respectively. Thus,

$$\sum_{q \neq l, 0} \sum_{l \neq 0} \|\zeta^l\|_2 \|\zeta^q\|_2 \leq \frac{1}{n^3} \sum_{q \neq 0} \frac{1}{q^2} \sum_{l \neq q} \frac{1}{l^2} = O\left(\frac{1}{n^3}\right), \quad (25)$$

where in the last equality we use the fact that $\sum_{l \neq q} \frac{1}{l^2} = \text{constant}$ and thus $\sum_{q \neq 0} \frac{1}{q^2} \sum_{l \neq q} \frac{1}{l^2} = \text{constant}$ as well. Thus, we get from (19) that $\|c - \zeta\|_2^2 = O(\kappa^2/\tilde{N}^3)$. (23). Combining this with (19) leads to the desired result. \square

One may inquire what can be said on κ in Theorem 5. To this end, we employ the empirical analysis performed in [25]. In that work, it was conjectured that the eigenvalues of a randomly subsampled frame obey a Manova distribution. To employ their result in our case, we may treat D as a matrix sampled from a significantly larger Fourier basis. In their work, they have two parameters. The first is γ , which is the fraction between the large basis and the size of the rows, namely \tilde{N} in our case. This selection of subset of the rows creates a frame (the selection can be deterministic in this step of the selection as is our case). We set $\gamma = \epsilon\tilde{N}$, where ϵ is a very small number as the large basis should represent the whole space we are sampling from and we scale ϵ with \tilde{N} as we get closer to the whole space when we add more samples. The second parameter is $\beta = \frac{n}{\tilde{N}}$, which is the redundancy factor in D . Given these two parameters, the support of the MANOVA distribution that characterize the singular values of D is $[r_-, r_+]$, where

$$\begin{aligned} r_{\pm} &= \left(\sqrt{\beta(1-\gamma)} \pm \sqrt{1-\beta\gamma} \right)^2 \\ &= \left(\sqrt{\frac{n}{\tilde{N}}} - \epsilon n \pm \sqrt{1-\epsilon n} \right)^2. \end{aligned} \quad (26)$$

Note that r_-/r_+ provides a bound to the condition number (as the minimal/maximal singular value may be greater/smaller than r_-/r_+). Assuming ϵn is negligible, we have that

$$\kappa \leq \frac{(\sqrt{\beta} + 1)^2}{(\sqrt{\beta} - 1)^2}. \quad (27)$$

Notice that in Theorem 5, the ratio β of D is a free parameter that we may adjust to optimize the bound. This leads us to the following conjecture

Conjecture 6: If a univariate network has bounded weights, the training data (x_i, y_i) of size $n \geq 2K + 1$ is randomly sampled from a band-limited function f with $2K + 1$ non-zero Fourier coefficients (i.e., $y_i = f(x_i)$), $x_i \sim U[-\pi, \pi]$, and the network ϕ_n fits the data, then

$$\|f(x) - \phi_n(x)\|_{L^2[-\pi, \pi]}^2 = O(1/n^3). \quad (28)$$

It is conjecture as it relies on empirical analysis [25] (with no rigorous proof) and on our assumptions above. If all of these are correct, then we get this result by simply plugging (27) and $\tilde{N} = \frac{n}{\beta}$ in the bound of Theorem 5, which yields

$$\|f(x) - \phi_n(x)\|_{L^2[-\pi, \pi]}^2 = O\left(\frac{(\sqrt{\beta} + 1)^4 \beta^3}{(\sqrt{\beta} - 1)^4} / n^3\right). \quad (29)$$

Since β is an arbitrary constant, the nominator can be also considered as such and thus we get that the error scales as $O(1/n^3)$. Notice that this bound is not tight and thus we cannot use it to approximate the ratio between the number of samples required in the deterministic and random cases in order to get the same error. Next we present a numerical simulation that demonstrates that this ratio is not so high and

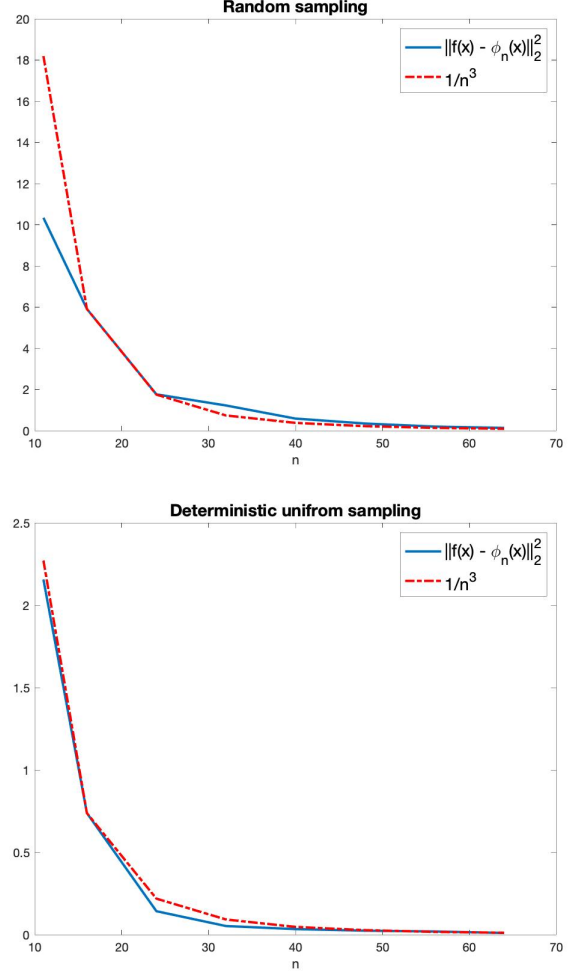


Fig. 2. Network error as a function of the number of training samples n . Top: Training with random samples. Bottom: Training with uniform (equispaced) samples. Notice that in both cases, the network error scales as $1/n^3$.

that both uniform deterministic and random sampling indeed obey a decay rate of $1/n^3$ for band-limited signals.

Empirical demonstration

We have generated a bandlimited signal with 11 Fourier coefficients ($K = 5$). The signal is presented in Figure 1. We sampled both uniformly (equispaced) and randomly the function f , generating n pairs of $(x_i, y_i = f(x_i))$, where $n \in \{11, 16, 24, 32, 40, 48, 56, 64\}$. Then we trained a neural network with two hidden layers of size 1000. We trained the network with weight decay and a SGD with momentum (with parameter 0.5). Once the network converged, we calculated its error compared to the generating function. Figure 2 shows that in both cases the error scales as $1/n^3$. The very larger error at $n = 11$ in the random case may be explained by the fact that in this case, we can just have $\beta = 1$ and then the condition number is relatively large, which increases the error. An additional experiment appears in the appendix.

Notice that in the random sampling case, we need roughly twice the number of points to get to the same error as in the equispaced sampling case. This shows the great advantage of

the latter. This observation may serve as a motivation for the farthest point sampling technique used in active learning when searching for new examples to annotate.

VI. CONCLUSION

This work used sampling theory tools to analyze the error of neural networks. We showed that when the input data is band-limited, the network error scales as $O(1/n^3)$ in the univariate case both with uniformly sampled and randomly sampled data. To the best of our knowledge, no such decay rate was demonstrated in the literature of neural network generalization (see for example the survey [28]). As we assume that the network fits the data, the total network error studied in this work is the same as its generalization error.

While this work provides a generalization error of over-parameterized networks with bounded weights, our analysis does not take into account the implicit bias on the margin of these networks implied by the optimization [29], [30], which is also important for network generalization [16], [17]. We believe that a combination of these tools may further improve the understanding of neural networks.

Notice that while the discussion in this paper was on band-limited functions, our results may be easily extended to other types of functions such as ones that have compact support in wavelets or splines. In this case, one may use tools from generalized sampling theory [22], [31], [32], [33], [34] to represent the signal in a similar way as we have done in (4) and then perform a similar analysis to the one performed in this paper.

Acknowledgment. This work is supported by the European research council starting grant (ERC-StG 757497 PI Giryes).

REFERENCES

- [1] P. Savarese, I. Evron, D. Soudry, and N. Srebro, "How do infinite width bounded norm networks look in function space?" in *Conference on Learning Theory*, 2019, pp. 2667–2690.
- [2] F. Williams, M. Trager, D. Panozzo, C. Silva, D. Zorin, and J. Bruna, "Gradient dynamics of shallow univariate relu networks," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8376–8385.
- [3] G. Ongie, R. Willett, D. Soudry, and N. Srebro, "A function space view of bounded norm infinite width re{lu} nets: The multivariate case," in *International Conference on Learning Representations*, 2020.
- [4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Systems*, vol. 2, pp. 303–314, 1989.
- [5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jul. 1989.
- [6] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6231–6239.
- [7] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations*, 2017.
- [8] E. J. Cands, "Harmonic analysis of neural networks," *Applied and Computational Harmonic Analysis*, vol. 6, no. 2, pp. 197 – 218, 1999.
- [9] S. Sonoda and N. Murata, "Neural network with unbounded activation functions is universal approximator," *Applied and Computational Harmonic Analysis*, vol. 43, no. 2, pp. 233 – 268, 2017.
- [10] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5301–5310.

- [11] Z.-Q. J. Xu, Y. Zhang, and Y. Xiao, "Training behavior of deep neural network in frequency domain," in *Neural Information Processing*, 2019, pp. 264–274.
- [12] R. Balestrierio and richard baraniuk, "A spline theory of deep learning," in *International Conference on Machine Learning (ICML)*, 10–15 Jul 2018, pp. 374–383.
- [13] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," *Machine Learning*, vol. 14, no. 1, pp. 115–133, Jan 1994.
- [14] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Proceedings of The 28th Conference on Learning Theory*, 03–06 Jul 2015, pp. 1376–1401.
- [15] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Proceedings of the 31st Conference On Learning Theory*, vol. 75, 06–09 Jul 2018, pp. 297–299.
- [16] J. Sokoli, R. Giryes, G. Sapiro, and M. R. D. Rodrigues, "Robust large margin deep neural networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4265–4280, Aug 2017.
- [17] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 6240–6249.
- [18] C. Wei, J. D. Lee, Q. Liu, and T. Ma, "Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 9709–9721.
- [19] D. Grishin and T. Strohmer, "Fast scattered data approximation with neumann and other boundary conditions," *Linear Algebra and its Applications*, vol. 391, pp. 99 – 123, 2004.
- [20] W. Rudin, *Real and Complex Analysis, 3rd Ed.* USA: McGraw-Hill, Inc., 1987.
- [21] E. Margolis and Y. C. Eldar, "Nonuniform sampling of periodic bandlimited signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 2728–2745, July 2008.
- [22] A. J. Jerri, "The shannon sampling theoremits various extensions and applications: A tutorial review," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, Nov 1977.
- [23] M. Unser and J. Zerubia, "Generalized sampling: stability and performance analysis," *IEEE Transactions on Signal Processing*, vol. 45, no. 12, pp. 2941–2950, Dec 1997.
- [24] B. Farrell, "Limiting empirical singular value distribution of restrictions of discrete fourier transform matrices," *Journal of Fourier Analysis and Applications*, vol. 17, no. 4, pp. 733–753, 2011.
- [25] M. Haikin, R. Zamir, and M. Gavish, "Random subsets of structured deterministic frames have manova spectra," *Proceedings of the National Academy of Sciences*, vol. 114, no. 26, pp. E5024–E5033, 2017.
- [26] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 2924–2932.
- [27] M. Jacob, T. Blu, and M. Unser, "Sampling of periodic signals: A quantitative error analysis," *Signal Processing, IEEE Transactions on*, vol. 50, pp. 1153 – 1159, 06 2002.
- [28] D. Jakubovitz, R. Giryes, and M. R. D. Rodrigues, *Generalization Error in Deep Learning*. Springer International Publishing, 2019, pp. 153–193.
- [29] K. Lyu and J. Li, "Gradient descent maximizes the margin of homogeneous neural networks," in *International Conference on Learning Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=SJeLIgBKPS>
- [30] M. S. Nacson, S. Gunasekar, J. Lee, N. Srebro, and D. Soudry, "Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models," in *ICML*, 2019, pp. 4683–4692.
- [31] M. Unser, "Sampling-50 years after Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.
- [32] A. Aldroubi and K. Grchenig, "Nonuniform sampling and reconstruction in shift-invariant spaces," *SIAM Review*, vol. 43, no. 4, pp. 585–620, 2001.
- [33] Y. C. Eldar, "Sampling with arbitrary sampling andreconstruction spaces and oblique dual frame vectors," *Journal of Fourier Analysis and Applications*, vol. 9, no. 1, pp. 77–96, Jan 2003.
- [34] —, *Sampling Theory: Beyond Bandlimited Systems*. Cambridge University Press, 2015.

APPENDIX A

AN ADDITIONAL EMPIRICAL EXPERIMENT

We generate a bandlimited signal with 9 Fourier coefficients ($K = 4$). The signal is presented in Figure 3.

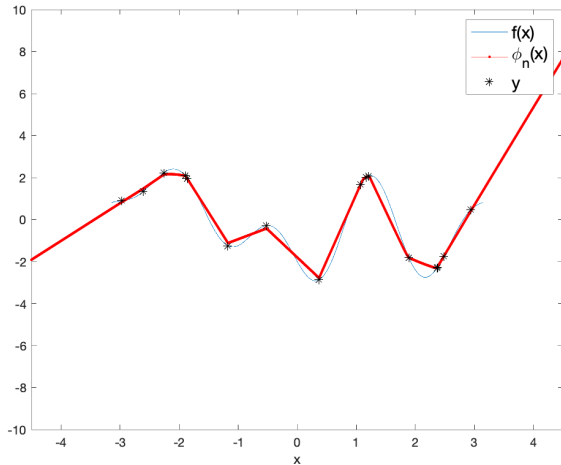


Fig. 3. Approximation of a band-limited ($K = 4$) function $f(x)$ using a network ϕ_n trained using only 16 training examples (y).

We sampled both uniformly (equispaced) and randomly the function f , generating n pairs of $(x_i, y_i = f(x_i))$, where $n \in \{9, 16, 24, 32, 40, 48, 56, 64\}$. Then we trained a neural network with two hidden layers of size 1000. We trained the network with weight decay and a SGD with momentum (with parameter 0.5). Once the network converged, we calculated its error compared to the generating function. Figure 4 shows that in both cases the error scales as $1/n^3$. As before, the larger error at $n = 9$ in the random case may be explained by the fact that in this case, we can just have $\beta = 1$ and then the condition number is relatively large, which increases the error.

Notice that also here, in the case of a small number of samples, we get better error with deterministic uniform sampling compared to random sampling.

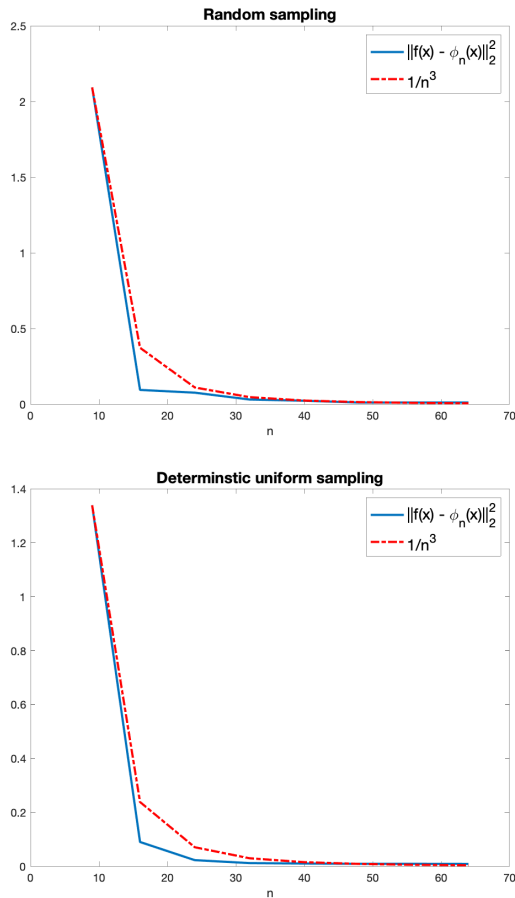


Fig. 4. Network error as a function of the number of training samples n for $K = 4$. Top: Training with random samples. Bottom: Training with uniform (equispaced) samples. Notice that in both cases, the network error scales as $1/n^3$.