Project 3 Notes

Aparna Gupte and Liu Zhang

December 20, 2022

1 Week 1. June 14-18

This week, we read the paper by Basri et al. [BJKK19], completed Task 1 and started work on Task 2. In our report for Week 1, we summarize our understanding of Basri et al. [BJKK19], and then describe our observations for Task 1, and our next steps for Task 2.

1.1 Paper Summary: The convergence rate of neural networks for learned functions of different frequencies, Basri et al. [BJKK19]

This paper serves as the main reference for tasks 1 and 2. The paper studies how the convergence rate of two-layered shallow neural networks (henceforth 2-Layer NN) is related to the frequency of the input function. There are two main results:

1. The convergence rate is faster for low frequency functions and slower for high frequency functions.
2. The bias terms are necessary to learn odd frequencies.

This paper also gives the close-form formula for computing the eigenvalues and eigenvectors for the neural tangent kernel of the NN, which we will investigate in task 2.

Definition 1 (2-Layer NN without bias terms)**.**

$$f(\mathbf{x}_i; \mathbf{W}, \mathbf{a}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(\mathbf{w}_r^T \mathbf{x}_i) \tag{1}$$

Definition 2 (Loss function: the L_2 loss)**.**

$$\Phi(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \mathbf{W}, \mathbf{a}))^2. \tag{2}$$

Definition 3 (The dynamic model of the NN)**.**

$$Z = \frac{1}{\sqrt{m}} \begin{pmatrix} a_1 \mathbb{I}_{11} \mathbf{x}_1 & a_1 \mathbb{I}_{12} \mathbf{x}_2 & \cdots & a_1 \mathbb{I}_{1n} \mathbf{x}_n \\ a_2 \mathbb{I}_{21} \mathbf{x}_1 & a_2 \mathbb{I}_{22} \mathbf{x}_2 & \cdots & a_2 \mathbb{I}_{2n} \mathbf{x}_n \\ \vdots & \vdots & & \vdots \\ a_n \mathbb{I}_{n1} \mathbf{x}_1 & a_n \mathbb{I}_{n2} \mathbf{x}_2 & \cdots & a_n \mathbb{I}_{nn} \mathbf{x}_n \end{pmatrix}, \tag{3}$$

where $\mathbb{I}_{ij} = \begin{cases} 1 \ \textit{(if } \mathbf{w}_i^T x_j \geqslant 0\textit{)} \\ 0 \ \textit{(otherwise)} \end{cases}$.

Using this definition, we can rewrite the output of the NN as $\mathbf{u}(t) = Z^T \mathbf{w} \in \mathbb{R}^n$ *where* $\mathbf{w} = (\mathbf{w}_1^T, \ldots, \mathbf{w}_m^T)^T$.

Definition 4 (H matrix)**.**

$$H = H(t) = Z^T Z, \quad H_{ij} = \frac{1}{m} \mathbf{x}_i^T \mathbf{x}_j \sum_{r=1}^{m} \mathbb{I}_{ri} \mathbb{I}_{rj}. \tag{4}$$

Definition 5 ($\mathbf{H}^\infty$ matrix)**.**

$$\begin{aligned}\mathbf{H}_{ij}^\infty &= \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(0,\kappa^2 I)} H_{ij} \\ &= \frac{1}{2\pi}\mathbf{x}_i^T\mathbf{x}_j(\pi - \arccos(\mathbf{x}_i^T\mathbf{x}_j)).\end{aligned} \tag{5}$$

Arora et al. [ADH+19] give a guarantee on the convergence rate of a shallow 2-layer neural network in terms of the eigenvalues of $\mathbf{H}^\infty$. For a network with $m = \Omega\left(\frac{n^7}{\lambda_0^4\kappa^2\varepsilon^2\delta}\right)$ units, $\kappa = O\left(\frac{\varepsilon\delta}{\sqrt{n}}\right)$ and learning rate $\eta = O(\lambda_0/n^2)$, where λ_0 denotes the minimal eigenvalues of $\mathbf{H}^\infty$. With probability $1-\delta$ over the random initializations,

$$\|\mathbf{y} - \mathbf{u}(t)\|_2 = \left(\sum_{i=1}^n (1-\eta\lambda_i)^{2t}(\mathbf{v}_i^\intercal\mathbf{y})^2\right) \pm \varepsilon \tag{6}$$

Definition 6 (Convolution on a hyper-sphere)**.**

$$K * f(u) = \int_{\mathbb{S}^d} K(\mathbf{u}^T\mathbf{v}) f(v) dV, \tag{7}$$

where the kernel $K(\mathbf{u}^T\mathbf{v})$ is measurable and absolutely integrable.

Theorem 7 ("Theorem 1" in [1]**.** *] Suppose the training data $\{\mathbf{x}_i\}_{i=1}^n$ is uniformly distributed in $\mathbb{S}^d$. Then $\mathbf{H}^\infty$ forms a convolution matrix in $\mathbb{S}^d$.*

The above theorem essentially tells us that

- in $\mathbb{S}^1$, the eigenvectors of $\mathbf{H}^\infty$ = Fourier series in $\mathbb{S}^1$;
- in $\mathbb{S}^d$, the eigenvectors of $\mathbf{H}^\infty$ = spherical harmonics in $\mathbb{S}^d$.

Theorem 8 ("Theorem 2" in [1]**.** *] In the harmonic expansion of $f(\mathbf{x}_i; \mathbf{W}, a)$ in Definition 1, the coefficients corresponding to odd frequencies $k \geqslant 3$ are zero.*

This theorem shows that the NN without bias terms cannot represent odd frequencies $k \geqslant 3$. To introduce the bias terms, we extend the NN in Definition 1 to homogeneous coordinates by defining $\bar{x} = \frac{1}{\sqrt{2}}(\mathbf{x}^T, 1)^T$.

Definition 9 (Extension of $\mathbf{H}^\infty$ to $\overline{H}^\infty$)**.**

$$\overline{H}_{ij}^\infty(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(0,\kappa^2 I)} \overline{H}_{ij} \tag{8}$$

$$= \frac{1}{2\pi}\overline{\mathbf{x}_i^T}\,\overline{\mathbf{x}_j}(\pi - \arccos(\overline{\mathbf{x}_i^T}\,\overline{\mathbf{x}_j})) \text{ (by the definition of } H_{ij}) \tag{9}$$

$$= \frac{1}{4\pi}(\mathbf{x}_i^T\mathbf{x}_j + 1)(\pi - \arccos(\mathbf{x}_i^T\mathbf{x}_j)) \text{ (by the definition of } \bar{\mathbf{x}}). \tag{10}$$

The associated kernel, $\overline{K}^\infty(\theta) = \overline{H}_{ij}^\infty(\mathbf{x}_i, \mathbf{x}_j)$ is defined as $\overline{K}^\infty(\theta) = \frac{(\cos(\theta)+1)(\pi-\theta)}{4\pi}$, where $\theta = \mathbf{x}_i^T\mathbf{x}_j$ is the angle between two input vectors.

Eigenvalues in $\mathbb{S}^1$. K^∞ (without bias) and $\overline{K}^\infty$ (with bias) include the Fourier series as their eigenfunctions. For frequencies $k \geq 0$, the eigenvalues of K^∞ are as follows.

$$a_k^1 = \begin{cases} 1/\pi^2 & k = 0 \\ 1/4 & k = 1 \\ \dfrac{2(k^2+1)}{\pi^2(k^2-1)^2} & \text{even } k \geq 2 \\ 0 & \text{odd } k \geq 2 \end{cases} \tag{11}$$

The eigenvalues of $\overline{K}^{\infty}$ are as follows.

$$c_k^1 = \begin{cases} \frac{1}{2\pi^2} + \frac{1}{8} & k = 0 \\ 1/\pi^2 + 1/8 & k = 1 \\ \frac{k^2+1}{\pi^2(k^2-1)^2} & \text{even } k \geq 2 \\ \frac{1}{\pi^2 k^2} & \text{odd } k \geq 2 \end{cases} \tag{12}$$

For each frequency value k, there are two eigenfunctions: $\sin(k\theta)$ and $\cos(k\theta)$.

Note that with a bias, the eigenvalues of $\overline{H}^{\infty}$ corresponding to frequency k decay as $1/k^2$ for all $k \in \mathbb{Z}, k \geq 0$.

Eigenvalues in $\mathbb{S}^d, d \geq 2$. To generalize this to higher dimensional unit spheres, the Funk-Hecke theorem tells us that the eigenfunctions of $\mathbf{H}^{\infty}$ are the higher dimensional analogs of the Fourier series, the spherical harmonics.

Definition 10 (Gegenbauer Polynomial)**.**

$$P_{k,d}(t) = \frac{(-1)^k}{2^k} \frac{\Gamma(\frac{d}{2})}{\Gamma(k+\frac{d}{2})} \frac{1}{(1-t^2)^{\frac{d-2}{2}}} \frac{d^k}{dt^k}(1-t^2)^{k+\frac{d-2}{2}}. \tag{13}$$

Theorem 11 (Funk-Hecke)**.** *Given any measurable function K on $[-1,1]$, such that the integral $\int_{-1}^{1} \|K(t)\|(1-t^2)^{\frac{d-2}{2}} dt < \infty$, for every spherical harmonic $H(\sigma)$ of frequency k, we have:*

$$\int_{\mathbb{S}^d} K(\sigma \cdot \xi) H(\xi) d\xi = \left(Vol(\mathbb{S}^{d-1}) \int_{-1}^{1} K(t) P_{k,d}(t) (1-t^2)^{\frac{d-2}{2}} dt \right) H(\sigma), \tag{14}$$

where $Vol(\mathbb{S}^{d-1})$ denotes the volume of $(\mathbb{S}^{d-1})$ and $P_{k,d}(t)$ denotes the Gegenbauer polynomial.

Similar to the 1-dimensional case, when there is no bias, K^{∞} has all the odd harmonics with $k \geq 3$ in its null space, so those eigenvalues vanish.

The paper provides a calculation for the eigenvalues analytically, but roughly, the eigenvalues corresponding to frequency k decay as $1/k^d$.

1.2 Task 1: Observing the spectral bias

In this task, we ran experiments to get an empirical understanding of the spectral bias of neural networks during training. We first describe our set-up. The target function we were trying to learn is

$$f(x) = \sin(2\pi x) + \sin(4 \cdot 2\pi x) + \sin(8 \cdot 2\pi x) \tag{15}$$

where $x \in [0,1]$.

First, we trained an over-parameterized 2-layer neural network N with a ReLU activation function, defined as follows:

$$N(x) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(w_r^T x), \quad \sigma(x) = \text{ReLU}(x, 0) = \max\{x, 0\}. \tag{16}$$

The weights were initialized as follows: $W, b \sim \mathcal{N}(0,1)$, $a_r \sim \text{Unif}[\{-1,1\}]$.

To observe the spectral bias phenomenon, we made two plots at fixed intervals during training. The first is the plot of the target function ("exact") and the function that the NN has learned at the point of plotting ("learned"). This plot will inform us the current stage of training. The second is a plot of the Discrete Fourier Transform (DFT) of the residue $f - N$ at the point of plotting. This will inform us on the specific

frequency components that the NN has learned and those that the NN has yet to learn. We observed that the peaks in the DFT plot corresponding to the lower frequencies decreased first, confirming the spectral bias of the training procedure of neural networks. To illustrate this observation, we will include the function plot and DFT plot before training, after 20000 iterations, and after 40000 iterations, respectively:

Figure 1: Function plot before training

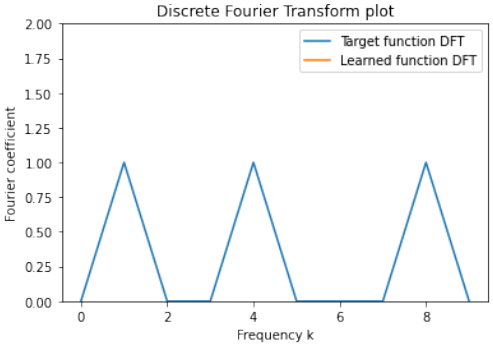


Figure 2: DFT plot before training

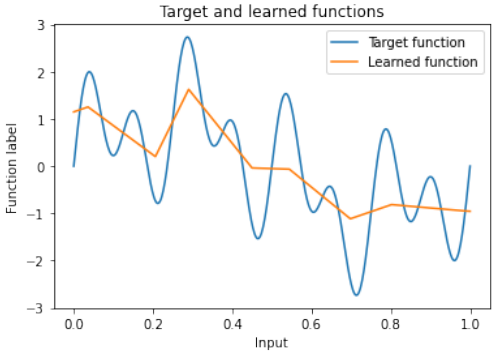


Figure 3: Function plot after 20000 iterations

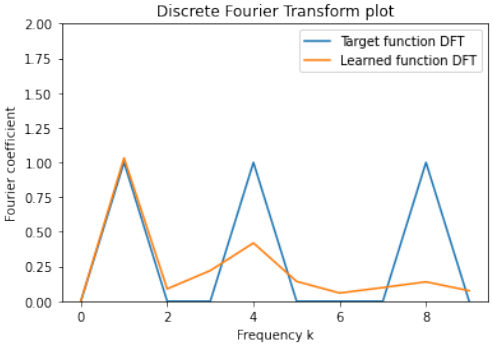


Figure 4: DFT plot after 20000 iterations

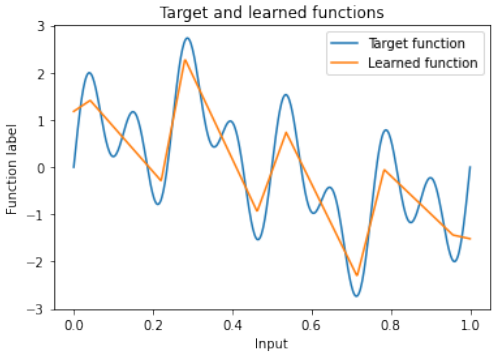


Figure 5: Function plot after 40000 iterations

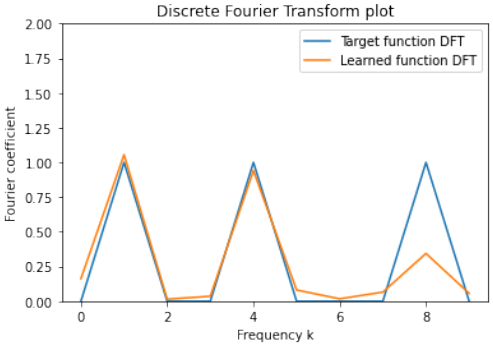


Figure 6: DFT plot after 40000 iterations

We performed further experiments by varying the activation functions, the optimization methods, and the width and will report our results on each:

1. **Varying the activation functions.** The preference for learning low frequencies first was also observed when using other non-linear activation functions including sigmoid, tanh, and leaky ReLU. We have observed that the leaky ReLU activation function gave the best learning performance.

2. **Varying the optimization methods.** Further, we observed that this spectral bias is robust to differences in the training procedure. For example, we observed the frequency bias when training with both the simple vanilla gradient descent optimizer, as well as the Adam optimizer. However, the simple gradient descent optimizer quickly got stuck in a local minimum, which did not allow us to observe the bias to the extent we saw with Adam. This showed us that we might not always be able to observe empirically what we have shown theoretically, due to issues such as unexpected local minimum issues.

3. **Increasing the width of the NN.** We noticed that when the 2-layer neural network is highly over-parameterized, the training process becomes much easier. Specifically, we see that with smaller number of parameters, it gets stuck in local minima, and this problem is resolved with larger number of parameters (with increased width). This phenomenon is explained in [DZPS19]. As the width becomes large enough, the over-parameterized NN behaves similarly to a linear system, i.e., the Neural Tangent Kernel (NTK), which we will investigate further in Task 2.

Our code and results for Task 1 can be found in the iPython notebooks in the folder named "code".

1.3 Task 2: Investigating the Neural Tangent Kernel (NTK)

For task 2, we started by doing a literature review to find previous work that computes the eigenvalues of the NTK for activation functions other than ReLU. Our goal for the next week is to either analytically or numerically estimate the eigenvalues for the other activation functions, to get a generalization of the theory that explains the low frequency bias. We would like to see whether we can generalize further by relaxing the assumptions on the input data as well as the assumption of the weights for different parameters.

2 Week 2. June 21-25

2.1 Motivation of NTK

Suppose $N(\mathbf{x};\theta)$ is neural network function with parameters θ and input x. Jacot et al. [JGH] showed that during gradient descent on the parameters of the NN, the dynamics of the network function f follows the (negative) kernel gradient

$$\partial_t N_t = -\nabla_{\Theta^{(L)}} C|_{N_t} \tag{17}$$

in the function space (of the loss function) with respect to the NTK (formally defined below).

Definition 12 (Neural Tangent Kernel (NTK))**.**

$$\Theta(\mathbf{x},\overline{\mathbf{x}}) = \langle \nabla_\theta N(\mathbf{x};\theta), \nabla_\theta N(\overline{x};\theta)\rangle \tag{18}$$

As the width of the hidden layer tends to infinity, the neural network can be described by the limit of the NTK, an constant kernel, $\Theta(\mathbf{x},\overline{\mathbf{x}})$. The explicit formula for the limiting NTK for ReLU is just the $\mathbf{H}^\infty$ matrix that we have previously defined in Definition 5.

The positive-definiteness of the infinite-width limiting NTK can allow us to describe the values of f outside the training set, thus making it crucial in understanding the generalization features of the NN.

We highlight several properties of the NTK which will be important to our analysis:

1. As the width tends to infinity, the NTK converges to an explicit limiting kernel and it stays constant during training.
2. The limiting kernel only depends on the depth of the network, the choice of non-linearity and the initialization variance.
3. The limiting NTK is positive-definite when the data is supported on the sphere and the non-linearity (based on the chosen activation function) is non-polynomial.

In this week, our aim is to use the NTK regime to understand the frequency bias of an over-parameterized NN (as previously defined) that we have observed in our experiments in Week 1. Specifically, we are interested in analytically deriving and numerically approximating the eigenvalues and eigenvectors of the limiting NTK.

2.2 Literature review

We surveyed the existing analytical methods relevant to computing the eigenvalues and eigenvectors of the $\mathbf{H}^\infty$ matrix corresponding to different activation functions. We now summarize the most relevant approaches as follows.

1. **Vempala et al. [VW19]**
   - Vempala et al. [VW19] studies a general class of activation functions, most notably the sigmoid and the ReLU. This work gives an exponentially decaying lower bound on the eigenvalues. Concretely, define the sigmoid function as $\sigma(x) = 1/(1+e^{-x})$. Then, the eigenvalues c_k^d corresponding to degree/frequency k for the corresponding kernel $\mathbf{H}^\infty$, in the d dimensional problem, have the following lower bound
   $$c_k^d \leq d^{-k-O(1)}. \tag{19}$$
   - Their assumption is that the data points are independently drawn from the uniform distribution D on the sphere $\mathbb{S}^{n-1}$.
   - A similar lower bound holds for the case where the activation function is ReLU $\sigma(x) = \max\{0, x\}$. Although this gives us an upper bound on the time taken to learn the k-degree component of the target function, this work does not show that this bound is tight, and so it doesn't fully explain the frequency bias.

2. **Bietti and Mairal [BM]**

   Later work by Bietti and Mairal [BM] computes more fine-grained estimates for the eigenvalues for the k-degree components, for ReLU activations, and shows that the above lower bound is not tight. Specifically, they show that the eigenvalues decay polynomially as $1/k^d$, for even k, and 0 for odd k. This implies that the 2-layer neural network model studied cannot learn the odd frequencies of the target function.

3. **Basri et al. [BJKK19]**

   - The work by Basri et al. [BJKK19] builds on this work by showing that by adding a bias term in the 2-layer network, the eigenvalues for odd k also decay as $1/k^d$, matching the result for even k.
   - The assumption in this work is that the data is uniformly on the a hypersphere $\mathbb{S}^d$.

4. **Luo et al. [BJKK19]** This paper derived the Low-frequency Principle (LFP) dynamics model and the explicit formula for the corresponding LFP operator. With this LFP operator, they extend to two corollaries with applications to ReLU and tanh, respectively. Finally, they showed the equivalence between long term gradient descent dynamics with an optimization problem that shows the spectral bias.

2.3 Gegenbauer Polynomials, Spherical Harmonics and the Funk-Hecke Theorem

Definition 13 (Spherical harmonics)**.** *The spherical harmonic functions $Y(\theta, \phi)$ is defined as satisfying the equation*

$$\left[\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial}{\partial\theta}\right) + \frac{1}{\sin^2(\theta)}\frac{\partial^2}{\partial\phi^2}\right] Y(\theta,\phi) \equiv L^2 y(\theta,\phi) = -\lambda Y(\theta,\phi), \tag{20}$$

where L^2 denotes the angular part of the Laplacian operator. The spherical harmonic functions $Y(\theta, \phi)$ are thus the eigenfuntions of L^2.

Spherical harmonics are useful for systems that process spherical symmetry (rotational invariance). Intuitively, the spherical harmonics can be thought of as the closest analogue of the orthogonal vectors and orthogonal polynomials for functions defined on the sphere.

2.4 Analytic derivations for the limiting NTK for ReLU and Leaky ReLU activation functions

Recall from Definition **??** that the network function is defined as

$$f(\mathbf{x}_i; W, \mathbf{a}) = \frac{1}{\sqrt{m}}\sum_{r=1}^{m} a_r \sigma(\mathbf{w}_r^T \mathbf{x}_i). \tag{21}$$

Differentiating $f_i = f(\mathbf{x}_i)$ with respect to w_r, we get

$$\frac{\partial f_i}{\partial \mathbf{w}_r} = \frac{1}{\sqrt{m}} a_r \sigma'(\mathbf{w}_r^T \mathbf{x}_i)\mathbf{x}_i \in \mathbb{R}^d. \tag{22}$$

The training dynamics is described by the matrix,

$$Z = \begin{pmatrix} \frac{\partial f_1}{\partial \mathbf{w}_1} & \cdots & \frac{\partial f_n}{\partial \mathbf{w}_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial \mathbf{w}_m} & \cdots & \frac{\partial f_n}{\partial \mathbf{w}_m} \end{pmatrix}. \tag{23}$$

Since the Gram matrix $H = Z^T Z$, we can write its entry:

$$H_{ij} = \sum_{r=1}^{m} \left(\frac{\partial f_i}{\partial \mathbf{w}_r}\right)^T \left(\frac{\partial f_i}{\partial \mathbf{w}_r}\right) \tag{24}$$

$$= \frac{1}{m} \sum_{r=1}^{m} a_r^2 \sigma'(\mathbf{w}_r^T \mathbf{x}_i)\sigma'(\mathbf{w}_r^T \mathbf{x}_j)\mathbf{x}_i^T \mathbf{x}_j. \tag{25}$$

The limiting NTK, which we have previously defined as the $\mathbf{H}^\infty$ matrix in Definition 5, is the expectation of the Gram matrix H over all the possible initializations. Thus, given H_{ij} above, we can write the entry of $\mathbf{H}^\infty$:

$$\begin{aligned}
\mathbf{H}_{ij}^\infty &= \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(0,\mathbb{I}_\times)}[a_r^2 \sigma'(\mathbf{w}_r^T \mathbf{x}_i)\sigma'(\mathbf{w}_r^T \mathbf{x}_j)\mathbf{x}_i^T \mathbf{x}_j] \\
&= \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(0,\mathbb{I}_\times)}[\sigma'(\mathbf{w}_r^T \mathbf{x}_i)\sigma'(\mathbf{w}_r^T \mathbf{x}_j)\mathbf{x}_i^T \mathbf{x}_j] \text{ (by the given assumption that } a \sim \{-1, 1\}) \\
&= \mathbf{x}_i^T \mathbf{x}_j \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(0,\mathbb{I}_\times)}[\sigma'(\mathbf{w}_r^T \mathbf{x}_i)\sigma'(\mathbf{w}_r^T \mathbf{x}_j)] \\
&= \mathbf{x}_i^T \mathbf{x}_j \int_{\mathbb{R}^d} \sigma'(\mathbf{w}_r^T \mathbf{x}_i)\sigma'(\mathbf{w}_r^T \mathbf{x}_j) \frac{e^{-\frac{\mathbf{w}^T\mathbf{w}}{2}}}{\sqrt{(2\pi)^d}} d\mathbf{w}.
\end{aligned} \tag{26}$$

Note that Equation 26 holds for any activation function $\sigma(\cdot)$. Further, since the integral in this expression depends on the unit vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ only through their inner products with $\mathbf{w}$, and the Gaussian distribution considered is spherically symmetric, by rotational symmetry, we can conclude that $\mathbf{H}_{ij}^\infty$ depends on $\mathbf{x}_i, \mathbf{x}_j$ only through the inner product $\mathbf{x}_i^\intercal \mathbf{x}_j$. This allows us to use the Funk-Hecke theorem, for any activation function.

Allowing a bias is equivalent to considering inputs of the form $\overline{\mathbf{x}} = \frac{1}{\sqrt{2}}(\mathbf{x}, 1)$. Note that $\|\overline{\mathbf{x}}\| = \|\mathbf{x}\| = 1$. Assuming that the bias weights are initialized to 0, and then the corresponding $\overline{H}^\infty$ matrix takes the form

$$\overline{H}_{ij}^\infty = \frac{1}{2}(\mathbf{x}_i^\intercal \mathbf{x}_j + 1) \int_{\mathbb{R}^d} \sigma'(\mathbf{w}_r^\intercal \mathbf{x}_i)\sigma'(\mathbf{w}_r^\intercal \mathbf{x}_j) \frac{e^{-\frac{\mathbf{w}^T\mathbf{w}}{2}}}{\sqrt{(2\pi)^d}} d\mathbf{w}. \tag{27}$$

Now, suppose the activation function is the ReLU function, that is, $\sigma(x) = \max\{x, 0\}$. Then $\sigma'(\mathbf{w}_r^T \mathbf{x}_i)\sigma'(\mathbf{w}_r^T \mathbf{x}_j) = 1$ when $\mathbf{w}^T \mathbf{x}_i, \mathbf{w}^T \mathbf{x}_i \geqslant 0$, and 0 otherwise. So we get

$$\mathbf{H}_{ij}^\infty = \mathbf{x}_i^T \mathbf{x}_j \int_{\mathbb{R}^d} \mathbb{1}_{\mathbf{w}^T\mathbf{x}_i \geq 0} \mathbb{1}_{\mathbf{w}^T\mathbf{x}_i \geqslant 0} \frac{e^{-\frac{\mathbf{w}^T\mathbf{w}}{2}}}{\sqrt{(2\pi)^d}} d\mathbf{w} \tag{28}$$

$$= \frac{1}{\sqrt{(2\pi)^d}} \mathbf{x}_i^T \mathbf{x}_j (\pi - \arccos(\mathbf{x}_i^T \mathbf{x}_j)). \tag{29}$$

Including the bias term, the individual term in the limiting NTK for ReLU function, $\overline{H}_{ij}$ is as follows:

$$\overline{H}_{ij} \frac{1}{4\pi}(\mathbf{x}_i^T \mathbf{x}_j + 1)(\pi - \arccos(\mathbf{x}_i^T \mathbf{x}_j)). \tag{30}$$

Similarly, if the activation function is the Leaky ReLU function, that is, $\sigma(x) = 1(x < 0)(\alpha x) + 1(x \geqslant 0)(x)$. Then we get

$$\mathbf{H}_{ij}^\infty = \alpha[\pi - \arccos(\mathbf{x}_i^T \mathbf{x}_j)](1 + \alpha^2) - 2\alpha \arccos(\mathbf{x}_i^T \mathbf{x}_j). \tag{31}$$

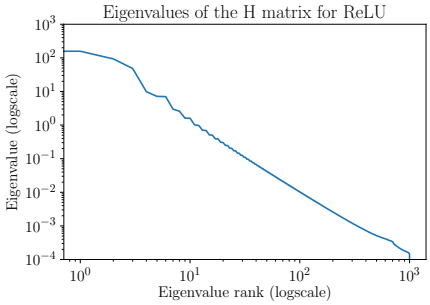
2.5 Numerical computation via estimations of $\mathbf{H}^\infty$

In this series of experiments, we estimated the spectrum of the $\mathbf{H}^\infty$ matrix (or equivalently, the limiting NTK) when different activation functions were chosen. We computed the eigenvalues and eigenvectors of the H matrix as an estimate of $\mathbf{H}^\infty$ matrix. We then made the log-log plot to show the rate of decay for the eigenvalues of the H matrix when the activation function is chosen to be ReLU, Leaky ReLU,

sigmoid, and hyperbolic tangent (tanh), respectively. In addition, we obtained the dominant frequency of the corresponding eigenvectors using Discrete Fourier Transform.

The code can be found on the GitHub repository and we now highlight our findings.

We started with the over-parameterized regime where $m = 100000, n = 1000$. The first finding was that for the over-parameterized case, the eigenvalues of H matrix decay polynomially for non-smooth activation functions (ReLU and Leaky ReLU) and exponentially for smooth activation functions (sigmoid and tanh). This was shown in Figure 7 and 8: the log-log plot for eigenvalues showed a linear trend for both ReLU and Leaky ReLU (implying polynomial decay); the semi-log plot for eigenvalues showed a linear trend for both sigmoid and tanh (implying exponential decay). This finding verified the results in [BM]) that eigenvalues of $\mathbf{H}^{\infty}$ matrix decay polynomially as $1/k^d$, for even k, and 0 for odd k. It also verified the results in [LMXZ20] that the eigenvalues of the limiting NTK decay polynomially for non-smooth activation functions and exponentially for smooth activation functions. (We suspected that the discontinuity in figure (a) below was due to rounding error and was negligible.)



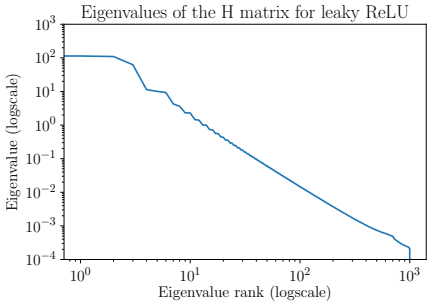
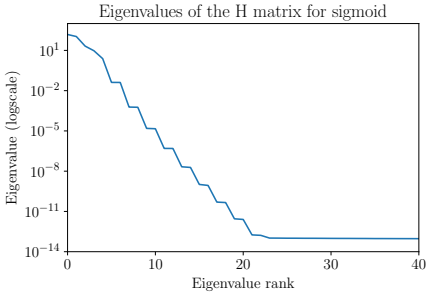


Figure 7: Both of the log-log plots show an approximately linear trend, indicating that the eigenvalues of H matrix for ReLU and Leaky ReLU decay polynomially.



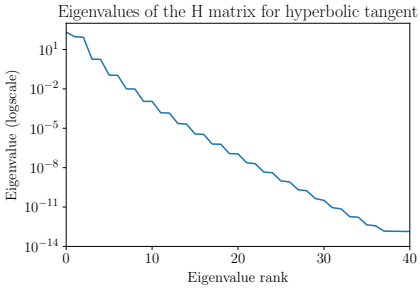


Figure 8: Both of the semi-log plots show an approximately linear trend, indicating that the eigenvalues of H matrix for sigmoid and hyperbolic tangent decay exponentially.

The second finding was that eigenvectors corresponding to larger eigenvalues have lower dominant frequencies and eigenvectors corresponding to smaller eigenvalues have higher dominant frequencies. This was observed by visualizing the eigenvectors and their DFT plot. We included the plots for ReLU activation function for illustration.

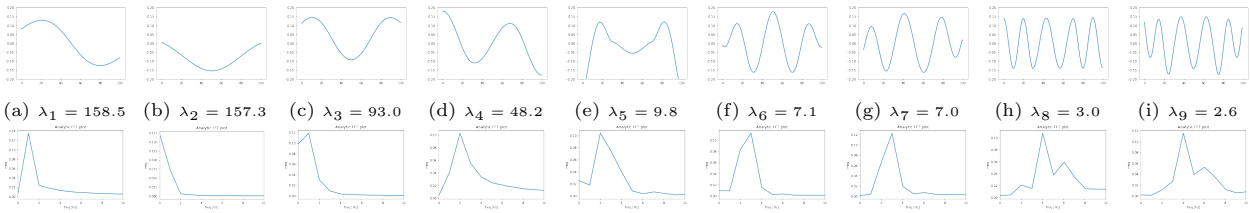


Figure 9: Eigenvectors of the H matrix (in descending order based on corresponding eigenvalues) and their frequencies when $m \gg n$ and the bias term is included. The leading eigenvector has the lowest frequency and the smaller the corresponding eigenvalue, the higher the frequency of the eigenvector.

We further plotted the eigenvetors and corresponding frequencies when there was no bias term below. Our finding aligned with Basri et al. [BJKK19]: when the bias term was not included, the odd frequencies greater than 1 could not be learned, even though the same frequency bias was preserved.

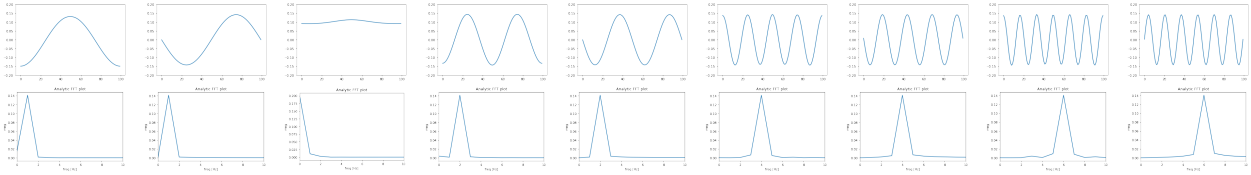


Figure 10: Eigenvectors of the H matrix (in descending order based on corresponding eigenvalues) and their frequencies when $m \gg n$ and the bias term is not included. The leading eigenvector has the lowest frequency and the smaller the corresponding eigenvalue, the higher the frequency of the eigenvector. However, when the bias is not included the odd frequencies greater than three are not learned.

We repeated this experiments for the case where $m = n = 1000$ and $m(= 10) \ll n(= 1000)$ and obtained the following plots:

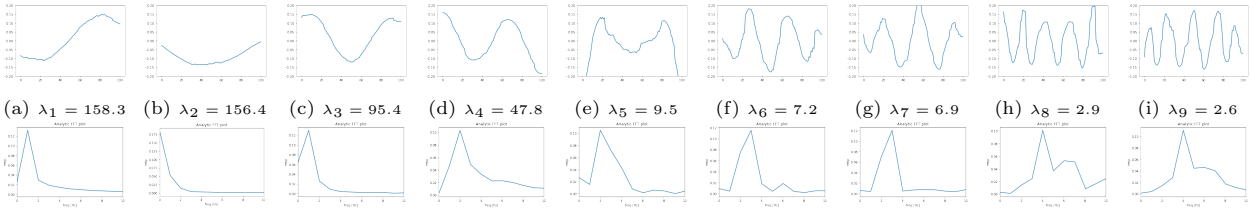


Figure 11: Eigenvectors of the H matrix (in descending order based on corresponding eigenvalues) and their frequencies when $m = n$ and the bias term is included. The leading eigenvector has the lowest frequency and the smaller the corresponding eigenvalue, the higher the frequency of the eigenvector.

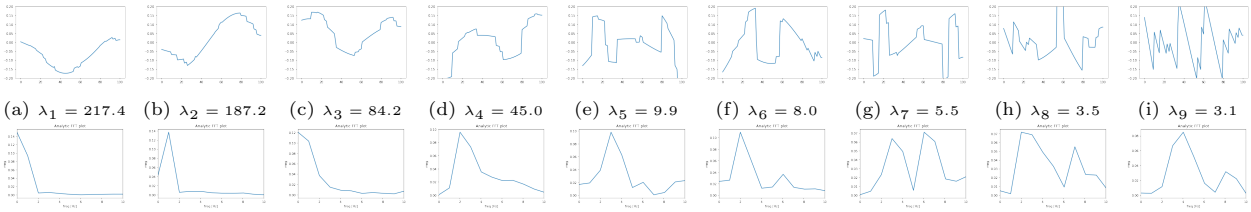


Figure 12: Eigenvectors of the H matrix (in descending order based on corresponding eigenvalues) and their frequencies when $m \ll n$ and the bias term is included. The frequency bias is not preserved in this case. For instance, the 6th leading eigenvector has lower dominant frequency than the 5th leading eigenvector.

To verify if the eigenvalues decay polynomially when the ReLU neural network is not over-parameterized, we plotted the eigenvalues of the H-matrix when $m = n = 100$ and $m(= 10) \ll n(= 1000)$.

Figure 13: The log-log plot shows a linear trend for when $m = n$, indicating polynomial decay. However, the log-log plot for under-parameterized neural network $m \ll n$ is not linear.

To summarize our findings:

1. Frequency bias was observed when $m \gg n$ and $m \approx n$, but not when $m \ll n$.
2. When the bias term is included, the odd frequencies can be learned; without the bias term, the odd frequencies cannot be learned.

2.6 Numerical computation via the Funk-Hecke Theorem

To calculate the eigenvalues more accurately, we applied the Funk-Hecke Theorem, which is an essential tool used in [CFW+20], [VW19], [BJKK19], [GMMM20], and [BM]. Our code can be found on the GitHub repository.

We give our calculations here.

By the Funk-Hecke theorem, with kernel K (which changes depending on the activation function), the eigenvalue corresponding to the kth degree zonal harmonic, when the inputs are d-dimensional, are given by

$$\lambda_k^d = \text{Vol}(\mathbb{S}^d) \int_{-1}^{1} K(t) P_{k,d}(t) (1-t^2)^{\frac{d-2}{2}} dt \tag{32}$$

where $P_{k,d}(t)$ denotes the Gegenbauer polynomial, given by the formula:

$$P_{k,d}(t) = \frac{(-1)^k}{2^k} \frac{\Gamma\left(\frac{d}{2}\right)}{\Gamma\left(k + \frac{d}{2}\right)} (1-t^2)^{\frac{d-2}{2}} \frac{d^k}{dt^k} (1-t^2)^{k+\frac{d-2}{2}}. \tag{33}$$

Here we denote the gamma function as $\Gamma(\cdot)$. For ReLU and Leaky ReLU activation functions, we have an analytic expression for the kernel K, but this is not the case for the sigmoid and tanh activation function. We use numerical integration methods in scipy to compute the first few eigenvalues of the kernel for these four different activation funcitons, and present our results here.

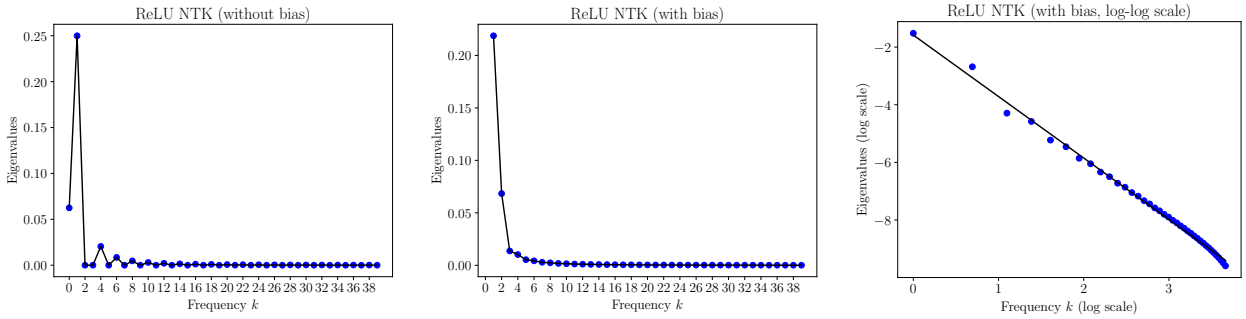


Figure 14: Eigenvalues of the ReLU Neural Tangent Kernel decay polynomially.

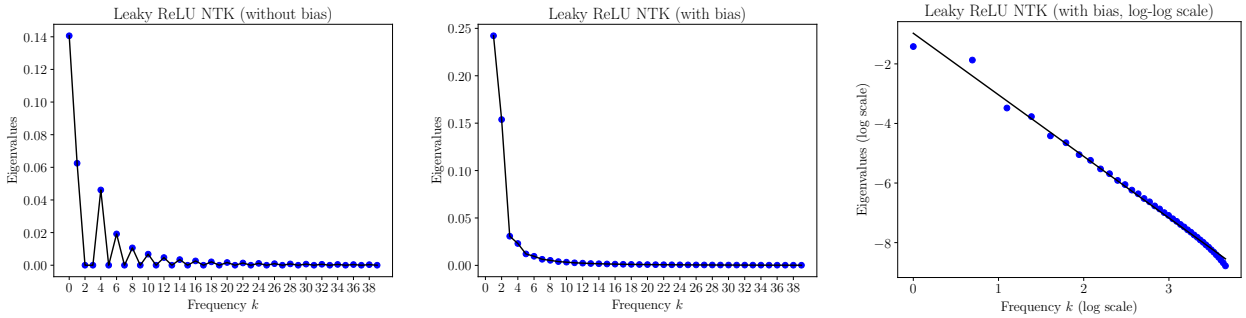


Figure 15: Eigenvalues of the Leaky ReLU Neural Tangent Kernel decay polynomially.

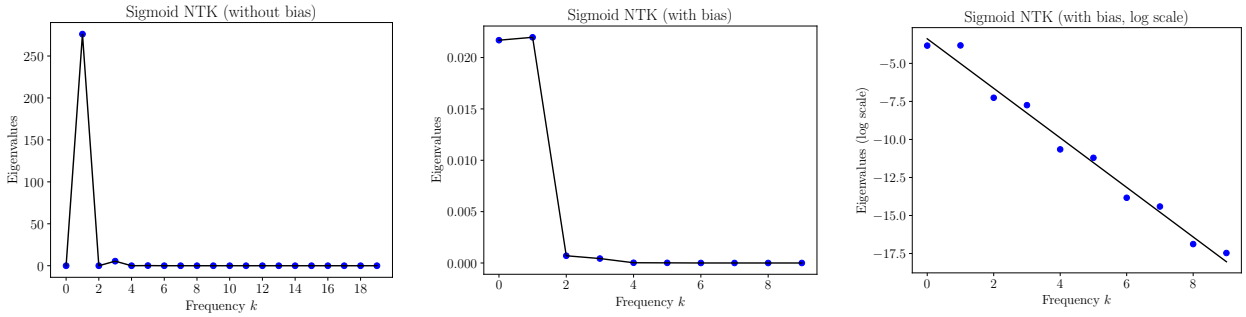


Figure 16: Eigenvalues of the Sigmoid Neural Tangent Kernel decay exponentially.

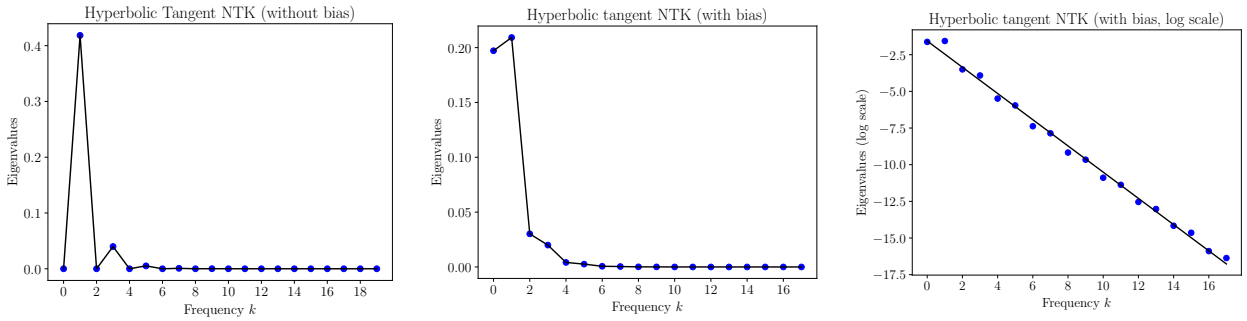


Figure 17: Eigenvalues of the NTK with hyperbolic tangent activation function decay exponentially.

The code can be found on the GitHub (gegenbauer.py) repository and we now describe our results.

We see that in all the cases, bias is critical for ensuring that all eigenvalues are non-zero. Our plot for ReLU agrees with the result by Basri et al. [BJKK19] that states that odd frequencies have 0 eigenvalues. Our results show that the same holds for the Leaky ReLU activation function (with parameter 0.2). For the sigmoid and tanh activation functions, on the other hand, the even eigenvalues are 0.

Further, we see that for both ReLU and Leaky ReLU, the decay is polynomial, but for sigmoid and tanh, the eigenvalues decay much faster, exponentially.

These plots hint at the fact that kernels with smooth versus non-smooth activation functions have different spectral properties, as has been studied by previous work [SH21], [PSG19].

Main ideas in Scetbon:

- NTK is a dot product kernel on the sphere, K.
- For any dot product kernel, there is an associated integral operator, T_k.
- Based on the coefficients $(b_m)_{m\geqslant 0}$ in the definition of the dot product kernel, we can determine whether the eigenvalue decay is in the polynomial, geometric, or super-geometric regime. (Both geometric and super-geometric decay are specific instances of exponential decay.)
- This paper goes one step further to derive the approximation error and statistical prediction of regularized least-squares estimator of the dot product kernel in each regime.

As proved in the section on analytic derivation, ReLU and leaky ReLU activation functions give kernels of the form $\pi - \arccos(\langle x_i, x_j\rangle)$. Based on the results in Table 1 of [SH21], the coefficients $(b_m) \in \mathcal{O}(m^{-3/2})$. As a result, the eigenvalues of the integral operator associated with ReLU NTK are given by $\lambda_m = m^{-d/2}$, implying a polynomial eigenvalue decay.

Similarly, sigmoid and hyperbolic tangent activation functions give kernels of the form $e^{-b\|x_i - x_j\|_2^2}$. Based on the results in Table 1 of [SH21], the coefficients $|b_m/b_{m-1}| \in \mathcal{O}(m^{-1}) \in \mathcal{O}(m^{-3/2})$. As a result, the eigenvalues of the integral operator associated with sigmoid NTK are given by $\lambda_m = (eb)^m m^{-m+(d-1)/2}$, implying an exponential eigenvalue decay.

Another parallel approach is to derive a kernel on the frequency domain [need to verify this] and thus obtain a linear frequency principle model. In Luo et al. showed that the higher frequencies evolve polynomially slower than lower frequencies for non-smooth activation functions and exponentially slower than lower frequencies for non-smooth activation functions.

Some hypothesis;

2.7 Next Steps

Our numerical computations of the eigenvalues show an exponential decay for eigenvalues of the NTK with smooth activation functions(tanh and sigmoid), but a polynomial decay for non-smooth functions like ReLU and Leaky ReLU. We can further see the effect of this exponential versus polynomial decay when training, if we quantitatively understand how the DFT of the residual changes with time. After this, we will get started on Task 3.

3 Week 3. June 28 - July 2

In this week, we first refined the report and the plots and started working on Task 3. We first spent some time understanding the theoretical basis for the H^s norm, and explained the main definitions and properties that we will use. These properties will allow us to make predictions about what we will observe if we train the neural network using an H^s-based loss function. Then, using the same set-up as in task 1, we train a neural network with an H^s norm-based loss function, and present our observations. We see that these results are consistent with the predictions made by the theory.

In the previous tasks, we studied the frequency bias of over-parameterized neural networks during training using both empirical and theoretical methods. In this task, we focused on using the Sobolev norm to improve training.

3.1 Sobolev Spaces and the H^s norm

Intuition: As emphasized in Terry Tao's blog [Tao], sometimes we are not only interested in the "width" and "height" of a function, but also its frequency, and "regularity" or smoothness. A natural way to reason about a function's smoothness is using its derivative. Sobolov spaces are vector spaces of functions with norms equal to the L^p norm of the function and its derivatives, upto a specified order.

Definition 14 (Sobolov Spaces $W^{k,p}(\Omega)$)**.** *Let $k \in \mathbb{N}, 1 \leq p \leq \infty$. The Sobolev space $W^{k,p}(\Omega)$ is defined to be the set of all functions f on Ω such that for every multi-index α with $|\alpha| \leq k$, the mixed partial derivative*

$$f^{(\alpha)} = \frac{\partial^{|\alpha|} f}{\partial x_1 \alpha_1 \dots \partial x_n^{\alpha_n}} \tag{34}$$

exists in the weak sense and is in $L^p(\Omega)$. The natural number k is called the order of the Sobolev space.

There are several choices for a norm for the $W^{k,p}$ space. We give three definitions, which are equivalent in the sense of equivalent norms, but mainly use the first definition.

Definition 15 ($W^{k,p}(\Omega)$ norm)**.** *Let $u \in W^{k,p}(\Omega)$ be a function. The following choices for the $W^{k,p}(\Omega)$ norm are equivalent in the sense of equivalent norms. That is, $\|u\|_{W^{k,p}(\Omega)} < \infty$ if and only if $\|u\|'_{W^{k,p}(\Omega)} < \infty$, which by definition indicates that function $u \in W^{k,p}(\Omega)$.*

$$\|u\|_{W^{k,p}(\Omega)} := \left\| (I - \Delta)^{k/2} u \right\|^p_{L^p(\Omega)} \tag{35}$$

$$\|u\|'_{W^{k,p}(\Omega)} := \begin{cases} \left(\sum_{|\alpha| \leq k} \|D^\alpha u\|^p_{L^p(\Omega)} \right)^{1/p} & 1 \leq p \leq \infty \\ \max_{|\alpha| \leq k} \|D^\alpha u\|_{L^\infty(\Omega)} & p = \infty \end{cases} \tag{36}$$

For the rest of our analysis, we will use the first definition above:

$$\|u\|_{W^{k,p}(\Omega)} := \left\| (I - \Delta)^{k/2} u \right\|^p_{L^p(\Omega)}. \tag{37}$$

3.1.1 Bessel Potential Sobolev Spaces

Bessel potential spaces generalize the notion of Sobolev spaces, where the order k is a natural number, to spaces with any real order s.

Definition 16 (Bessel Potential Spaces $H^{s,p}(\mathbb{R}^d)$)**.** *Let $s \in \mathbb{R}$, $1 \leq p < \infty$. Then the Bessel potential space over $\mathbb{R}^d$ associated with parameters s, p is given by*

$$H^{s,p}(\mathbb{R}^d) := \left\{ f \in \mathcal{S}'(\mathbb{R}^d) : \mathcal{F}^{-1} \left[\left(1 + |\xi|^2\right)^{s/2} \mathcal{F} f \right] \in L^p(\mathbb{R}^d) \right\}, \tag{38}$$

with norm

$$\|f\|_{H^{s,p}(\mathbb{R}^d)} := \left\| \mathcal{F}^{-1}\left[\left(1+|\xi|^2\right)^{s/2} \mathcal{F}f \right] \right\|_{L^p(\mathbb{R}^d)} = \left\| (I-\Delta)^{s/2} f \right\|_{L^p(\mathbb{R}^d)}. \tag{39}$$

Here $\mathcal{S}'(\mathbb{R}^d)$ is the space of tempered distributions, $\mathcal{F}f$ denotes the Fourier transform of f and $\mathcal{F}^{-1}f$ denotes the inverse Fourier transform of f.

In the special case where $p = 2$, we drop the superscript p in the notation. The resulting Bessel space is then a Hilbert space, and since $\mathcal{F}$ is unitary, we get the expression for the norm

$$\|f\|_{H^{s}(\mathbb{R}^d)} := \left\| \left(1+|\xi|^2\right)^{s/2} \mathcal{F}f \right\|_{L^2(\mathbb{R}^d)} = \left\| (I-\Delta)^{s/2} f \right\|_{L^2(\mathbb{R}^d)}. \tag{40}$$

3.1.2 Fourier formulation of the H^s norm

In this section, we derive the expression of the H^s norm of a function in terms of its Fourier expansion. This expression allows us to reason about the effect of a H^s-based loss function on the neural network training process, and make predictions that we confirm in our experiments in later sections of the report.

We can use Parseval's identity to obtain an expression for the H^s norm of a function and the H^s inner product in terms of its Fourier transform, giving us more intuition for how this norm behaves in the Fourier domain.

Geometrically, the Parseval's theorem is a generalization of the Pythagorean theorem, which states that the sum of the squares of the components of a vector in any orthonormal basis is equal to the squared length of the vector.

Theorem 17 (Parseval's Identity)**.** *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a square-integrable function, then its 2-norm is equal to the 2-norm of its Fourier transform*

$$\int_{\mathbb{R}^d} \left| \widehat{f}(\xi) \right|^2 d\xi = \int_{\mathbb{R}^d} |f(x)|^2 dx. \tag{41}$$

Here $\widehat{f} : \mathbb{R}^d \to \mathbb{R}$ denotes the Fourier transform of f.

Proposition 18 (Fourier formula for the H^s norm)**.** *The H^s norm of a function $f : \mathbb{R}^d \to \mathbb{R}$ can be expressed in terms of its Fourier transform as follows*

$$\|f\|_{H^s}^2 = \|(1+|\xi|^2)^{s/2} \widehat{f}(\xi)\|_2^2. \tag{42}$$

Proof. By Parsevals' Identity we know that

$$\|f\|_2^2 = \int_{\mathbb{R}^d} |f(x)|^2 dx = \int_{\mathbb{R}^d} |\widehat{f}(\xi)|^2 d\xi = \|\widehat{f}\|_2^2. \tag{43}$$

By the definition of the H^s norm,

$$\begin{aligned}
\|f\|_{H^s}^2 &= \int_{\mathbb{R}^d} f(x)(I-\Delta)^s f(x) dx && (44)\\
&= \int_{\mathbb{R}^d} (I-\Delta)^{s/2} f(x) (I-\Delta)^{s/2} f(x) dx && (45)\\
&= \int_{\mathbb{R}^d} \widehat{(I-\Delta)^{s/2} f}(x) \widehat{(I-\Delta)^{s/2} f}(x) dx && (46)\\
&= \int_{\mathbb{R}^d} (1+|\xi|^2)^{s/2} \widehat{f}(\xi) (1+|\xi|^2)^{s/2} \widehat{f}(\xi) d\xi && (47)\\
&= \|(1+|\xi|^2)^{s/2} \widehat{f}(\xi)\|_2^2, && (48)
\end{aligned}$$

where Equation 47 is obtained by considering the Fourier transform of the Laplacian, which can be computed with integration by parts:

$$\widehat{\Delta f}(\xi) = -|\xi|^2 \widehat{f}(\xi). \tag{49}$$

□

This property, stated in Proposition 18, allows us to make predictions about the frequency bias behavior of a neural network trained on a H^s-norm based loss function. When $s = 0$, we see that the loss function simply reduces to the L^2 loss.

When $s > 0$, frequencies ξ are weighted by the increasing function $(1 + |\xi|^2)^{s/2}$, putting larger weights on higher frequencies. From this, we can guess that the $H^s (s > 0)$ loss might be able to counterbalance the intrinsic low frequency bias of neural nets. As a result, higher frequency components should be learnt faster than in the L^2 loss scenario. This could potentially be used to speed up convergence rates of neural nets.

On the other hand, since low frequency bias is believed to be one of the reasons why overparameterized neural nets are able to generalize well, using the H^s norm could make the neural net training process more prone to picking up (higher frequency) noise in the data, and overfitting.

Conversely, when $s < 0$, the frequency ξ component is weighted by a decreasing function of ξ. This can be used to enhance the low frequency bias, and its associated good generalization properties in the underparameterized regime.

3.1.3 The $\dot{H}^s$ seminorm

We are also interested in the seminorm $\dot{H}^s$ of functions. Here we give some relevant definitions and properties.

Definition 19 ($\dot{H}^s$ seminorm)**.** *The $\dot{H}^s$ seminorm of a function $f : \mathbb{R} \to \mathbb{R}$ is given by*

$$\|f\|^2_{\dot{H}^s} = \int f(x)(-\Delta)^s f(x)dx. \tag{50}$$

Definition 20 ($\dot{H}^s$ inner product)**.** *Given two function $f, g : \mathbb{R} \to \mathbb{R}$, their $\dot{H}^s$ inner product is defined as*

$$\langle f, g\rangle_{\dot{H}^s} = \langle f, (-\Delta)^s g\rangle_{L^2} \tag{51}$$

$$= \int f(x)(-\Delta)^s g(x)dx. \tag{52}$$

Proposition 21 (Fourier formula for the $\dot{H}^s$ seminorm)**.** *The $\dot{H}^s$ seminorm of a function f is given by*

$$\|f\|^2_{\dot{H}^s} = \left\| |\xi|^s \widehat{f}(\xi) \right\|^2_2 \tag{53}$$

3.1.4 Implement H^s norm for $s \in \mathbb{Z}$ via the discretized Laplacian operator

The goal of this task is to use the H^s norm for the loss function, in place of the L^2 norm, which is the standard mean squared loss function. To empirically understand how this affects the training procedure, we need to first obtain a method to compute the H^s norm of functions, when the functions are available in the form of discretized vectors, evaluated on uniformly spaced inputs.

To recall our notation, $N(\cdot)$ is the function computed by the neural network at a particular iteration during training, and $f(\cdot)$ is the target function to be learnt. Denote by $\mathbf{u}$ the discretization of the residual $r = f - N$. Then, the L^2-based loss is computed as the L^2 norm of this residual, $\mathbf{u}^\intercal\mathbf{u}$. This is the discretization of the L^2 norm.

We follow the method proposed in [YTA20] and discretize functions and operators using finite difference schemes. To discretize the H^s norm, we must first discretize the Laplacian, let us denote the result by P. In one dimension, the Laplacian is the second derivative, which can be approximated with the following discretization,

$$\frac{d^2}{dx^2} r(x_i) \approx \frac{r(x_{i-1}) + r(x_{i+1}) - 2r(x_i)}{(\Delta x)^2}, \tag{54}$$

where $\Delta x = x_j - x_{j-1}$ for all j.

This gives us the following matrix approximation for the Laplacian operator:

$$K_n = \frac{1}{(\Delta x)^2} \begin{pmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -1 \end{pmatrix}, \tag{55}$$

where we use the Neumann boundary conditions to choose the top left and bottom right entries. The Neumann boundary condition in this case is $r'(x_{\text{boundary}}) = 0$.

Given the discretization of the Laplacian operator, we can derive the discretized version of the H^{-1} norm and H^{-2} norm. Let v be the discretization of f and w the discretization of g, and f, g are defined on the domain $(0, 1)$. Then $\|f - g\|_{H^{-1}}$ is given by

$$d(v, w) = \sqrt{h}\|(I_n - K_n)^{-1/2}(v - w)\|_2, \tag{56}$$

where I_n is the n-by-n identity matrix.

Similarly, the discretization of $\|f - g\|_{H^{-2}}$ is given by

$$d(v, w) = \sqrt{h}\|(I_n - K_n + K_n^2)^{-1/2}(v - w)\|_2. \tag{57}$$

3.1.5 Implement H^s norm for $s \in \mathbb{R}$ via Fourier transform

To extend our implementation of H^s norm from $s \in \mathbb{Z}$ to $s \in \mathbb{R}$, we make use of the equality in

$$\|f\|_{H^{s,p}(\mathbb{R}^d)} := \left\| \mathcal{F}^{-1}\left[\left(1 + |\xi|^2\right)^{s/2} \mathcal{F} f \right] \right\|_{L^p(\mathbb{R}^d)} = \left\| (I - \Delta)^{s/2} f \right\|_{L^p(\mathbb{R}^d)}.$$

This equality shows that the fractional Laplacian $(-\Delta)^s$ can be viewed as a psedo-differential operator $|\xi|^{2s}$.

3.2 Task 3: Preliminary Results

In this section, we describe our initial experiments with a H^s-based loss function, and describe how our findings are consistent with our predictions in Section 3.1.2.

The set up for our preliminary experiments is identical to that of Task 1: We use the Tensorflow Adam optimizer to train a two-layer neural net with ReLU activations, on the same target function

$$f(x) = \sin(2\pi x) + \sin(4 \cdot 2\pi x) + \sin(8 \cdot 2\pi x). \tag{58}$$

As a first step, we try to train the network with three different loss functions and compare the results (H^1, H^0, H^{-1}). Figures 18, 19 and 20 show the target and learned functions, and the DFTs of the target function and the residual, when trained with these three loss functions.

We see that there is a clear dependence of the frequency bias properties of the neural networks on the loss function, in a way that is consistent with what we predicted in Section 3.1.2.

Specifically, we see that with $s > 0$, the spectral bias is almost neutralized, with all frequencies being learnt at roughly the same rate, speeding up the convergence. On the other hand, when $s < 0$, we see that the spectral bias is enhanced, and the model learns only the lowest frequency even after several iterations, reducing the convergence speed.

In further investigations, we want to understand how we can fine-tune the choice of the parameter s, to get better convergence rates, or to provide better generalization in the underparameterized regime.

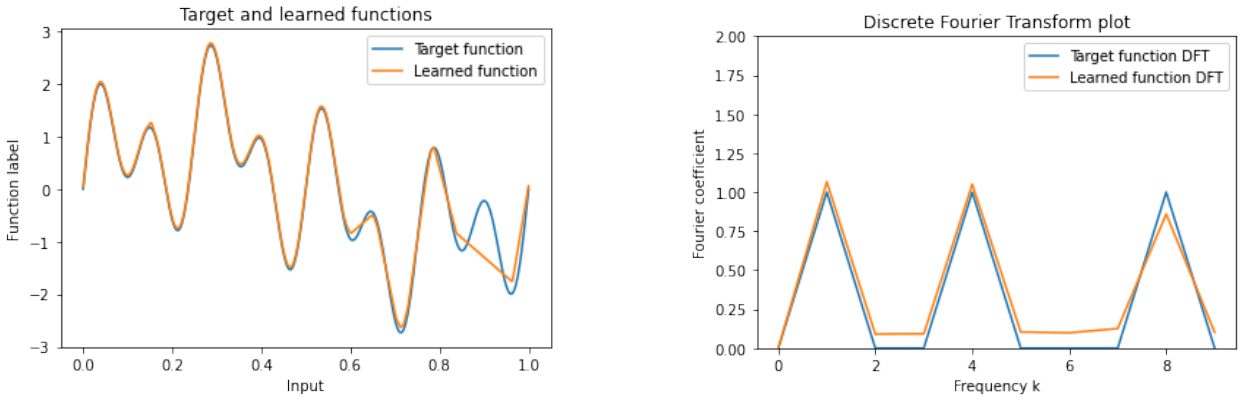


Figure 18: Training with the H^1-based loss function ($s = 1$): We see a near-perfect fit between the target and learned functions (left). We observe faster convergence, with all frequencies being learnt simultaneuously (see DFT, right), demonstrating that the H^1 loss counterbalances the "intrinsic" low frequency bias of the network (with an L^2 loss).

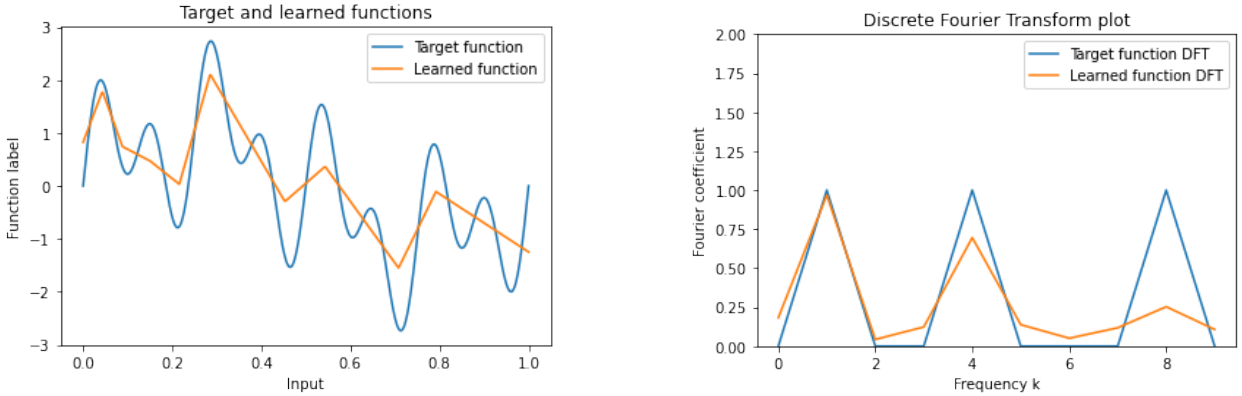


Figure 19: Training with L^2-based loss function ($s = 0$): This experiment is the control, demonstrating the "intrinsic/natural" low frequency bias of neural networks, usually visible when training with the standard L^2 loss.

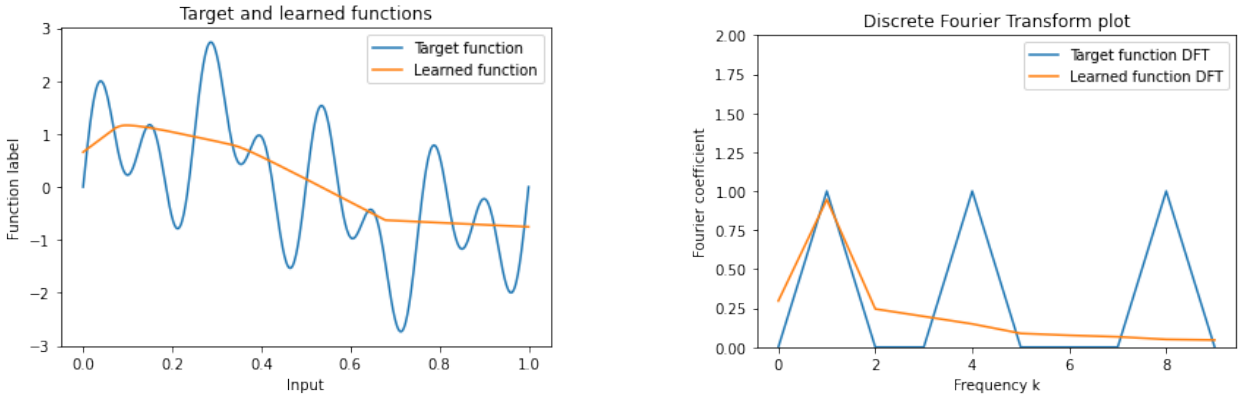


Figure 20: Training with H^{-1}-based loss function ($s = -1$): We see very slow convergence, and the network essentially learns only the lowest frequency, even after 15000 iterations.

3.3 Next steps

Next week we plan to finish up Task 3, which involves experimenting with both the underparameterized and overparameterized (NTK) regime and ask the following questions, for different values of the parameter s in the H^s norm loss.

1. Does the training get stuck at local minima?
2. How many iterations are needed until the training loss is 0.001?
3. How fast does the optimization procedure converge?
4. What is the generalization ability (measured by the test loss)?

After this, we hope to get started on Task 4, where we try to theoretically explain our results with H^s norm using the NTK model.

Thoughts:

- does generalization/test loss makes sense only in the presence of noise/with sparser training data?
- we need to repeat the experiments with simple gradient descent to understand the effects on local minima/loss landscape. so far we have been using adam.
- we also want to overparameterize (and reduce data) further till we don't need to train the outer weights
- quantifying learning rate for different frequencies empirically is still a good task we need to do.
- what about the other frequencies that get learned in between? Gibbs phenomenon? Worth mentioning/explaining somewhere in the report.

4 Week 4. July 5 - July 9

This week, we will investigate how the choice of H^s norm changes the training dynamics. The aim is to come up with a regime for designing a loss function that improves training performance given a specific task. To this end, our approach will consist of the following:

1. Empirical observations during training
2. Spectral analysis using the NTK regime
3. Analytic derivations

The main reference for this week is [ADH+19].

4.1 Paper Summary: [ADH+19]

Here, we give the statement of the main results of the paper by Arora et al. [ADH+19], and give quick proof sketches. The two main results of this paper can be summarized as follows.

1. A fine-grained analysis of the convergence of gradient descent on 2-layer over-parameterized neural networks (NTK regime), in terms of the spectrum of a certain Gram matrix ($\mathbf{H}^\infty$ in Definition 5).
2. A generalization bound that depends only on the data $\mathbf{y}$. This result is again, with respect to gradient descent and a 2-layer over-parameterized neural network.

4.1.1 Convergence rate

Theorem 22. *Suppose $\lambda_0 > 0$ is the smallest eigenvalues of the $\mathbf{H}^\infty$ matrix defined earlier, $\kappa = O\left(\frac{\varepsilon\delta}{\sqrt{n}}\right), m = \Omega\left(\frac{n^7}{\lambda_0^4\kappa^2\delta^4\varepsilon^2}\right)$ and $\eta = O\left(\frac{\lambda_0}{n^2}\right)$. Then with probability at least $1-\delta$ over the random initialization, for all time steps $t = 0, 1, 2, \ldots$ we have*

$$\|\mathbf{y} - \mathbf{u}(t)\|_2 = \sqrt{\sum_{i=1}^{n}(1-\eta\lambda_i)^{2t}(\mathbf{v}_i^\intercal \mathbf{y})^2} + \varepsilon. \tag{59}$$

Here $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$ are orthonormal eigenvectors $\mathbf{H}^\infty$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$.

Proof Sketch. The proof proceeds by showing that when the size of the initialization κ is small and the network width m is large, the sequence $\{\mathbf{u}(t)\}_{t=0}^\infty$ stays close to another sequence $\{\tilde{\mathbf{u}}(t)\}_{t=0}^\infty$, which has a linear update rule:

$$\tilde{\mathbf{u}}(0) = \mathbf{0}), \tag{60}$$
$$\tilde{\mathbf{u}}(t+1) = \tilde{\mathbf{u}}(t) - \eta\mathbf{H}^\infty(\tilde{\mathbf{u}}(t) - \mathbf{y}). \tag{61}$$

Define $\mathbf{r}(t) = \mathbf{y} - \tilde{\mathbf{u}}(t)$ as the approximation of the residue $\mathbf{y} - \mathbf{u}(t)$ at time step t.

From the initial conditions and update rule in Equation 60, we get the following recursive update rule for $\mathbf{r}$.

$$\mathbf{r}(0) = \mathbf{y} \tag{62}$$
$$\mathbf{r}(t+1) = (I - \eta\mathbf{H}^\infty)\mathbf{r}(t). \tag{63}$$

Writing the decomposition of $\mathbf{y} = \sum_{i=1}^n (\mathbf{v}_i^\intercal y)\mathbf{v}_i$ in terms of the eigenvectors $\{\mathbf{v}_i\}_{i=1}^n$, and noticing that $(I - \eta\mathbf{H}^\infty)$ has the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, with corresponding eigenvalues $1-\eta\lambda_1, \ldots, 1-\eta\lambda_n$, we get that

$$\mathbf{r}(t) = (I - \eta \mathbf{H}^{\infty})^t \mathbf{r}(0) \tag{64}$$
$$= (I - \eta \mathbf{H}^{\infty})^t \mathbf{y} \tag{65}$$
$$= \sum_{i=1}^{n} (I - \eta \mathbf{H}^{\infty})^t (\mathbf{v}_i^\intercal y) \mathbf{v}_i \tag{66}$$
$$= \sum_{i=1}^{n} (1 - \eta \lambda_i)^t (\mathbf{v}_i^\intercal y) \mathbf{v}_i. \tag{67}$$

The orthonormality of the eigenbasis $\{\mathbf{v}_i\}_{i=1}^n$ gives an estimate for the magnitude of the residual,

$$\|\mathbf{r}(t)\|_2^2 = \|\mathbf{y} - \tilde{\mathbf{u}}(t)\|_2^2 = \sum_{i=1}^{n} (1 - \eta \lambda_i)^{2t} (\mathbf{v}_i^\intercal y)^2. \tag{68}$$

□

The main idea of Theorem 22 is as follows: if $(\mathbf{v}_i^T \mathbf{y})^2$ is large for large λ_i, then gradient descent converges quickly; if $(\mathbf{v}_i^T \mathbf{y})^2$ is uniform or is large for small λ_i, then gradient descent converges slowly.

4.1.2 Generalization error bound

The next main theorem proved in the paper allows us to study the generalization ability of a 2-layer neural network trained by Gradient Descent (GD). The theorem gives an upper bound on the population loss Before stating the theorem, we requires the assumption that the data is non-degenerate in order for zero training loss to be achieved.

Definition 23. *A distribution $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{R}$ is (λ_0, δ, n)-non-degenerate if for n independent and identically distributed (i.i.d.) samples $\{(x_i, y_i)\}_{i=1}^n$ from $\mathcal{D}$, with probability at least $1 - \delta$ we have $\lambda_{\min}(\mathbf{H}^{\infty}) \geqslant \lambda_0 > 0$.*

Note that in most real-world distributions, no two input x_i and x_j are parallel to each other, which guarantees that $\lambda_{\min}(\mathbf{H}^{\infty}) > 0$.

Theorem 24. *Fix a failure probability $\delta \in (0, 1)$. Suppose the data $\{(x_i, y_i)\}$ are i.i.d. samples from some $(\lambda_0, \delta/3, n)$-non-degenerate distribution $\mathcal{D}$, and $\kappa = O\left(\frac{\lambda_0 \delta}{n}\right), m \geqslant \kappa^{-2} poly(n, \lambda_0^{-1}, \delta^{-1})$.*

Consider any loss function $l : \mathbb{R} \times \mathbb{R} \mapsto [0, 1]$ that is 1-Lipschitz in the first argument such that $l(y, y) = 0$. Then with probability at least $1 - \delta$ over the random initialization and the training samples, the 2-layer neural network trained by GD $N(x_i; W, a)$ for $k \geqslant \Omega(\frac{1}{\eta \lambda_0} \log \frac{n}{\delta})$ iterations has population loss

$$L_{\mathcal{D}}(N) \leqslant \sqrt{\frac{2 \mathbf{y}^T (\mathbf{H}^{\infty})^{-1} \mathbf{y}}{n}} + O\left(\sqrt{\frac{\log \frac{n}{\lambda_0 \delta}}{n}}\right). \tag{69}$$

The main idea of Theorem 24 is that the term $\sqrt{\frac{2 \mathbf{y}^T (\mathbf{H}^{\infty})^{-1} \mathbf{y}}{n}}$ determines the generalization error.

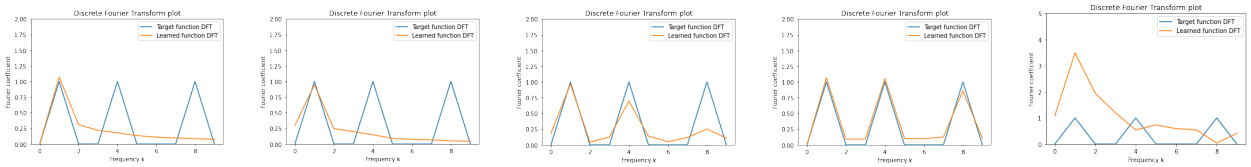
4.2 Experimental results

Liu: We will be changing the framework from the domain of the real line to the circle in the following week. This is to make sure that the experiments are align with the analysis.

4.2.1 Changing the coefficient of terms in the ground truth function

(a) H^{-1} (b) H^{-1} (c) H^{0} (d) H^{1} (e) H^{1}

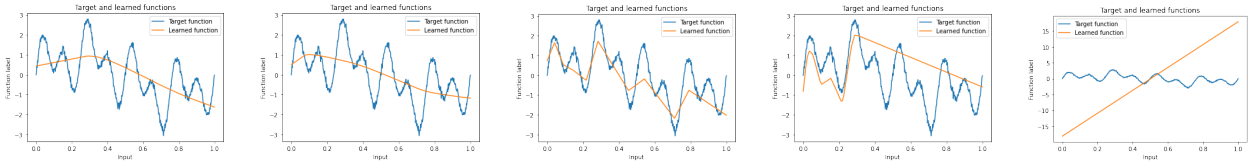
Figure 21: Effect of $\mathcal{H}^s$ norm on convergence rate.



(a) H^{-1} (b) H^{-1} (c) H^{0} (d) H^{1} (e) H^{1}

Figure 22: Effect of $\mathcal{H}^s$ norm on frequency bias. $\mathcal{H}^s(s > 0)$ norm counter-balances the inherent low frequency bias of the neural network. $\mathcal{H}^s(s < 0)$ norm reinforces the inherent low frequency bias of the neural network.

4.2.2 Adding noise to the training data



(a) H^{-1} (b) H^{-1} (c) H^{0} (d) H^{1} (e) H^{-1}

Figure 23: Effect of $\mathcal{H}^s$ norm on generalization ability.

4.2.3 Summary of results

- $\mathcal{H}^s(s > 0)$ norm gives faster convergence rate.
- $\mathcal{H}^s(s > 0)$ norm counters the low frequency bias of the neural network whereas $\mathcal{H}^s(s < 0)$ norm reinforces the low frequency bias of the neural network.
- In the presence of noise in the input data, $\mathcal{H}^s(s > 0)$ shows worse generalization ability as s gets larger.
- It is more likely to be stuck in local minima in the case of $\mathcal{H}^2$ norm.

4.3 Spectral Analysis

4.3.1 Spherical Harmonics in Dimensions d

In this section, we define the spherical harmonics in higher dimensions and give some of the properties. Our reference for the material in this section is the expository tutorial on spherical harmonics in higher dimensions by Frye and Efthimiou [FE12].

Let Δ_d be the Laplacian in d dimensions.

$$\Delta_d = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}. \tag{70}$$

In spherical coordinates, we can write the Laplacian as follows

$$\Delta_d = \frac{\partial^2}{\partial r^2} + \frac{d-1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\Delta_{\mathbb{S}^{d-1}}. \tag{71}$$

Aparna: Laplace-Beltrami The spherical harmonics on the unit sphere $\mathbb{S}^{d-1}$, which we will denotes as $Y_{d,k}$ (of degree k), are eigenfunctions on the $\Delta_{\mathbb{S}^{d-1}}$ operator

$$\Delta_{\mathbb{S}^{d-1}} Y_{d,k} = -k(k+d-2)Y_{d,k}. \tag{72}$$

4.3.2 Problem set up

In this section, we recall the problem set up and the notation we use. The training data is $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with the inputs $\mathbf{x}_i \in \mathbb{S}^{d-1} \subset \mathbb{R}^d$ uniformly spaced on the unit sphere, and $y_i \in \mathbb{R}$. We write $\mathbf{y} = (y_1, \ldots, y_n)^\intercal$ as the vector representing the discretization of the target function to be learnt on the unit sphere.

We consider a two layer network, with inner weights $\mathbf{W}$ and outer weights $\mathbf{a}$,

$$f(\mathbf{x}_i; \mathbf{W}, \mathbf{a}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(\mathbf{w}_r^\intercal \mathbf{x}_i). \tag{73}$$

We will omit $\mathbf{W}$ or $\mathbf{a}$ or both from the notation when it is clear. We write $\mathbf{f}(\mathbf{W}) = (f(\mathbf{x}_1; \mathbf{W}), \ldots, f(\mathbf{x}_n; \mathbf{W}))^\intercal$ to denote the vector of the predictions of the neural network on the training data, for a given set of weights $\mathbf{W}, \mathbf{a}$.

The inner weights are randomly initialized from a narrow spherical Gaussian $\mathbf{w}_r \sim \mathcal{N}(\mathbf{0}, \kappa^2 I_{d\times d})$, and are trained by the process of gradient descent with learning rate η. Here κ is a small constant. The outer weights are randomly sampled as $a_r \in \{-1, 1\}$, and kept constant throughout the training process.

The loss function we consider is a general quadratic form

$$\Phi(\mathbf{W}) = \frac{1}{2}\mathbf{r}(\mathbf{W})^\intercal \mathbf{P} \mathbf{r}(\mathbf{W}), \tag{74}$$

where we define $\mathbf{r}(\mathbf{W}) = \mathbf{y} - \mathbf{f}(\mathbf{W})$ to be the residual. To obtain a particular loss function, such as the H^s loss function, we choose a suitable $\mathbf{P}$ operator.

4.3.3 Spectral Analysis of the H^s norm operator

In this section, we give the eigenvectors and eigenvalues of the $\mathbf{P}$ operator corresponding to the H^s loss function, specialized to the one dimensional case,

$$\Phi_{H^s}(\mathbf{W}) = \|\mathbf{r}(\mathbf{W})\|_{H^s}^2 = \mathbf{r}^\intercal \mathbf{P}_{H^s} \mathbf{r}, \tag{75}$$

$$\Phi_{H^{-s}}(\mathbf{W}) = \|\mathbf{r}(\mathbf{W})\|_{H^{-s}}^2 = \mathbf{r}^\intercal \mathbf{P}_{H^{-s}} \mathbf{r} \tag{76}$$

where $s > 0$, and

$$\mathbf{P}_{H^s} = \sum_{r=0}^{s} (-\Delta)^r, \tag{77}$$

$$\mathbf{P}_{H^{-s}} = \left(\sum_{r=0}^{s} (-\Delta)^r \right)^{-1}. \tag{78}$$

Denote by $\lambda_k(-\Delta)$ the eigenvalue of $-\Delta$ indexed by k. Here we use the shorthand Δ to denote the spherical Laplacian operator $\Delta_{\mathbb{S}^1}$ on $\mathbb{S}^1$. Then we see that

$$\lambda_k(-\Delta) = k^2. \tag{79}$$

Therefore the eigenvalues of of $\mathbf{P}_{H^s}$ and $\mathbf{P}_{H^{-s}}$ are as follows

$$\lambda_k(\mathbf{P}_{H^s}) = \sum_{r=0}^{s} \left(\lambda_k(-\Delta)\right)^r, \tag{80}$$

$$\lambda_k(\mathbf{P}_{H^{-s}}) = \frac{1}{\sum_{r=0}^{s} \left(\lambda_k(-\Delta)\right)^r}. \tag{81}$$

From this formula, we see that setting $s = 1$ has the effect of making the eigenvalues grow quadratically in k,

$$\lambda_k(\mathbf{P}_{H^1}) = 1 + k^2. \tag{82}$$

Aparna: Need to specify that a bunch of matrices are positive definite and cite sources.

4.3.4 The Neural Tangent Kernel

In this section, we derive the Neural Tangent Kernel for the gradient flow dynamics. The familiar (discrete) gradient descent updates, with learning rate η are

$$\mathbf{w}_r(t+1) - \mathbf{w}_r(t) = -\eta \frac{\partial \Phi(\mathbf{W}(t))}{\partial \mathbf{w}_r} \tag{83}$$

$$= -\eta \frac{a_r}{\sqrt{m}} \left(\sum_{i=1}^{n} (\mathbf{Pr})_i \sigma'(\mathbf{w}_r(t)^\intercal \mathbf{x}_i) \mathbf{x}_i \right) \tag{84}$$

$$= -\eta \mathbf{Z}_r \mathbf{Pr}, \tag{85}$$

$$\mathbf{W}(t+1) - \mathbf{W}(t) = -\eta \mathbf{ZPr}, \tag{86}$$

where we define the submatrix

$$\mathbf{Z}_r = \frac{1}{\sqrt{m}} \begin{pmatrix} a_r \sigma'(\mathbf{w}_r^\intercal \mathbf{x}_1) x_1 & \dots & a_r \sigma'(\mathbf{w}_r^\intercal \mathbf{x}_n) x_n \end{pmatrix} \in \mathbb{R}^{d \times n} \tag{87}$$

of the matrix

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_m \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial \mathbf{w}_1} & \dots & \frac{\partial f_n}{\partial \mathbf{w}_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial \mathbf{w}_m} & \dots & \frac{\partial f_n}{\partial \mathbf{w}_m} \end{pmatrix} \in \mathbb{R}^{md \times n}. \tag{88}$$

The corresponding gradient flow equations are

$$\frac{\partial \mathbf{w}_r}{\partial t} = -\mathbf{Z}_r \mathbf{Pr} \tag{89}$$

$$\frac{\partial \mathbf{W}}{\partial t} = -\mathbf{ZPr} \tag{90}$$

Now we can compute how the residue/prediction error for each training data point changes with time,

$$\frac{dr_i}{dt} = \frac{df_i}{dt} \tag{91}$$

$$= \sum_{r=1}^{m} \left(\frac{\partial f_i}{\partial \mathbf{w}_r}\right)^\intercal \frac{\partial \mathbf{w}_r}{\partial t} \tag{92}$$

$$= \begin{pmatrix} \left(\frac{\partial f_i}{\partial \mathbf{w}_1}\right)^\intercal & \cdots & \left(\frac{\partial f_i}{\partial \mathbf{w}_m}\right)^\intercal \end{pmatrix} \begin{pmatrix} \frac{\partial \mathbf{w}_1}{\partial t} \\ \vdots \frac{\partial \mathbf{w}_m}{\partial t} \end{pmatrix} \tag{93}$$

$$= \begin{pmatrix} \left(\frac{\partial f_i}{\partial \mathbf{w}_1}\right)^\intercal & \cdots & \left(\frac{\partial f_i}{\partial \mathbf{w}_m}\right)^\intercal \end{pmatrix} \frac{\partial \mathbf{W}}{\partial t}. \tag{94}$$

Collecting these derivatives into a vector, we get that

$$\frac{\partial \mathbf{r}}{\partial t} = \begin{pmatrix} \frac{dr_1}{dt} \\ \vdots \\ \frac{dr_m}{dt} \end{pmatrix} \tag{95}$$

$$= \mathbf{Z}^\intercal \frac{\partial \mathbf{W}}{\partial t} = -\mathbf{Z}^\intercal \mathbf{Z} \mathbf{P} \mathbf{r} = -\mathbf{H} \mathbf{P} \mathbf{r}, \tag{96}$$

where

$$H_{ij} = \frac{1}{\sqrt{m}} \mathbf{x}_i^\intercal \mathbf{x}_j \sum_{r=1}^{m} \sigma'(\mathbf{w}_r^\intercal \mathbf{x}_i) \sigma'(\mathbf{w}_r^\intercal \mathbf{x}_j). \tag{97}$$

Note that $\mathbf{Z}$ and by extension $\mathbf{H}$ actually evolve with time, so $\mathbf{Z} = \mathbf{Z}(t), \mathbf{H} = \mathbf{H}(t)$ is better notation. However, in the infinite width limit, where $m \to \infty$, the matrix $\mathbf{H}$ becomes an expectation,

$$H_{ij}^{\infty} = \mathop{\mathbb{E}}_{\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \kappa^2 I_{d \times d})} \left[\mathbf{x}_i^\intercal \mathbf{x}_j \sigma'(\mathbf{w}^\intercal \mathbf{x}_i) \sigma'(\mathbf{w}^\intercal \mathbf{x}_j)\right]. \tag{98}$$

This is the so-called Neural Tangent Kernel, and actually stays constant with time, so that the residues are subjected to a linear update rule. In the overparameterized regime, this is a reasonable approximation to make, and allows us to approximate the residue easily, using this linear update rule,

$$\mathbf{r}(0) = \mathbf{y} \tag{99}$$

$$\mathbf{r}(t+1) = (\mathbf{I} - \eta \mathbf{H}^{\infty} \mathbf{P}) \mathbf{r}(t) = (\mathbf{I} - \eta \mathbf{K}) \mathbf{r}(t) \tag{100}$$

$$\mathbf{r}(t) = (\mathbf{I} - \eta \mathbf{K})^t \mathbf{r}(0) = (\mathbf{I} - \eta \mathbf{K})^t \mathbf{y}. \tag{101}$$

Here we write $\mathbf{K} = \mathbf{H}^{\infty} \mathbf{P}$ to denote the modified neural tangent kernel when using the more general loss function $\Phi(\mathbf{W}) - \mathbf{r}^\intercal \mathbf{P} \mathbf{r}$. Writing the residual as a linear combination of the eigenvectors of $\mathbf{K}$,

$$\mathbf{r}(t) = \sum_{k=1}^{n} c_k(t) \mathbf{v}_k. \tag{102}$$

Let $\mathbf{K}$ have eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and corresponding eigenvalues $\lambda_1(\mathbf{K}), \ldots, \lambda_n(\mathbf{K})$. Then the operator $(\mathbf{I} - \eta \mathbf{K})$ has the same eigenvectors $\mathbf{v}_k$, and eigenvalues $\lambda_k(\mathbf{I} - \eta \mathbf{K}) = 1 - \eta \lambda_k(\mathbf{K})$, for $k \in \{1, \ldots, n\}$.

Aparna: We are using both i, k to index eigenvalues, maybe better to choose one only. I'm going to go with k from now on. we can use i to index the inputs (though same set of indices).

4.3.5 Spectral analysis of the NTK $\mathbf{H}^{\infty}\mathbf{P}$

When $\mathbf{x}_i$ are uniformly spaced on the unit sphere, by the Funk Hecke theorem, we know that the eigenfunctions of $\mathbf{H}^{\infty}$ are the spherical harmonics, and the eigenvalue decay rate differs with different activation functions. When the eigenvectors of $\mathbf{P}$ are also exactly the spherical harmonics, we get that

$$\lambda_k(\mathbf{H}^{\infty}\mathbf{P}) = \lambda_k(\mathbf{H}^{\infty})\lambda_k(\mathbf{P}). \tag{103}$$

4.4 Notation

We denote vectors with boldface lowercase letters and matrices with boldface uppercase letters.

5 Week 5. July 10 - July 16

Previous works studying the frequency bias of neural networks theoretically have used a framework where the inputs are drawn from the unit sphere $\mathbb{S}^{d-1}$ (spherical domains), rather than $[-1,1]^d$ (rectangular domains). Our theoretical analysis of the H^s-based loss function in the NTK regime follows this set up, but our experiments do not.

In this week, we will set up the experimental framework for observing the frequency bias on the domain of $\mathbb{S}^1$, come up with a regime for choosing the $\mathcal{H}^s$ norm, and finally investigate if such regime can be further generalized to non-uniform data.

5.1 Defining the H^s norm on the spherical domain

In our problem, we are considering inputs to the neural network to be drawn from the unit sphere $\mathbb{S}^{d-1}$. To implement and analyze the H^s-based norm with inputs from a spherical domain, we need to define the H^s norm on the sphere. Several different but equivalent definitions of the Sobolev spaces have been studied [GSW10], [BLPE20].

Definition 25 (Sobolev space $\mathcal{H}^s(\mathbb{S}^{d-1})$)**.** *The Sobolev space $\mathcal{H}^s(\mathbb{S}^{d-1}), s>0$ is defined as the space of all functions $f \in L^2(\mathbb{S}^{d-1})$ such that $(-\Delta_{\mathbb{S}^{d-1}})^{s/2} f \in L^2(\mathbb{S}^{d-1})$. We provide this Hilbert space with the norm*

$$\|f\|^2_{\mathcal{H}^s(\mathbb{S}^{d-1})} = \|u\|^2_{L^2(\mathbb{S}^{d-1})}, \tag{104}$$

where u is the solution to the following Laplace problem: $(-\Delta_{\mathbb{S}^{d-1}})^{-s/2} u = f$.

(The fact that the Laplacian commutes with the spehrical harmonics is a little bit like saying that if you have the eigenvectors of "A", i.e., $Av_i = \lambda_i v_i$. Then, $A = V\Lambda V^\top$. This means this that $A^{-s/2} = V\Lambda^{-s/2}V^\top$.)

Let $\{Y^j_{\ell,d} \mid \ell = 0, 1, \ldots, 1 \le j \ne (\ell)\}$ denote an orthonormal basis of spherical harmonics for $L(\mathbb{S}^{d-1})$

Definition 26 (Spherical Harmonic decomposition)**.**

Definition 27 (H^s norm on the unit sphere)**.**

5.1.1 The unit circle $\mathbb{S}^1$

5.2 Effect of the H^s norm on different activation functions

5.3 Our proposed regime for Sobolev-norm-based weighted optimization

Recent works ([RBA+19], [BJKK19], [DZPS19]) have shown that there is a low frequency bias during the gradient-based training process of over-parameterized NNs, where low frequency components are learned first before the high frequency components. The ground truth, with different frequency components, also imposes a frequency bias. Liu: need to fact-check this again Based on the frequency properties of the given ground truth, we might want to reinforce or counter the inherent frequency bias of the NNs. This motivates us to use the Sobolev norm to improve the training process in terms of convergence rate, generalization capability, and optimization landscape.

1. **Convergence rate.**

   We first quantify the inherent frequency bias of the NN using the NTK regime, i.e., spectral analysis of H^∞, the limiting NTK at the infinite width limit of the NN. The frequency bias of the NNs depends on the activation functions: smooth activations give polynomial eigenvalue decay (at a rate of $1/k^d$ by results by Basri et al. [BJKK19]) and non-smooth activations give exponential eigenvalue decay.

   When the loss function is changed from the standard L^2 norm to our proposed $\mathcal{H}^s$ norm, the NTK for the gradient flow dynamics becomes $H^\infty P$, where P is obtained from the discretization of the $\mathcal{H}^s$ norm.

   Recall from the spectral analysis in last week's report that

   $$\lambda_k(\mathbf{H}^\infty \mathbf{P}) = \lambda_k(\mathbf{H}^\infty)\lambda_k(\mathbf{P}).$$

   Liu: also add in the experiment results

2. **Generalization capability.**

3. **Optimization landscape.**

5.4 Analysis for non-uniform data

The aim of this section is to generalize our regime to non-uniform data. We first need to verify that the same frequency bias is observed when the input data is not uniformly distributed on the domain of $\mathbb{S}^1$. We now introduce some circular distributions that we use for generating non-uniform data on $\mathbb{S}^1$:

1. Piecewise constant distribution.

$$p(x) = \begin{cases} \frac{1}{10} & x \in [-\pi, -\frac{1}{3}\pi) \\ \frac{7}{10} & x \in [-\frac{1}{3}\pi, \frac{1}{3}\pi) \\ \frac{2}{10} & x \in [\frac{1}{3}\pi, \pi) \end{cases} \tag{105}$$

2. Wrapped normal distribution.

$$p(\theta) = \frac{1}{2\pi}\left(1 + 2\sum_{p=1}^{\infty} \rho^{p^2} \cos p(\theta - \mu)\right), \tag{106}$$

where $\rho = e^{-\frac{\sigma^2}{2}}$.

When $\mu = 0, \sigma = 1$, we have

$$p(\theta) = \frac{1}{2\pi}(1 + 2\sum_{p=1}^{\infty} e^{-\frac{p^2}{2}} \cos p(\theta)) \tag{107}$$

$$= \frac{4.5 + 3\cos(2x + \pi)}{9\pi}. \tag{108}$$

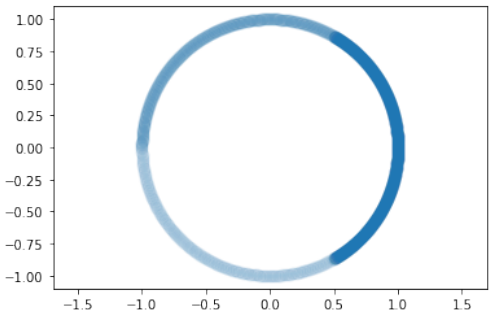


Figure 24: Piecewise constant distribution on the circle.

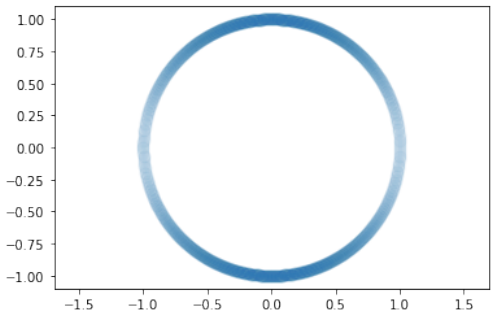


Figure 25: Wrapped normal distribution on the circle.

3. Asymmetric circular distribution.

$$p(\theta; \mu_1, \mu_2, \kappa_1, \kappa_2) = C \exp\{\kappa_1 cos(\theta - \mu_1) + \kappa_2 \cos 2(\theta - \mu_2)\} \tag{109}$$

$$(0 \leqslant \theta, \mu_1, \mu_2 < 2\pi, \kappa_1, \kappa_2 \geqslant 0). \tag{110}$$

In our experiment, $mu_1 = pi, \mu_2 = 0, \kappa_1 = \kappa_2 = 1, C = \frac{1}{5}$.

4. Antipodal distribution.

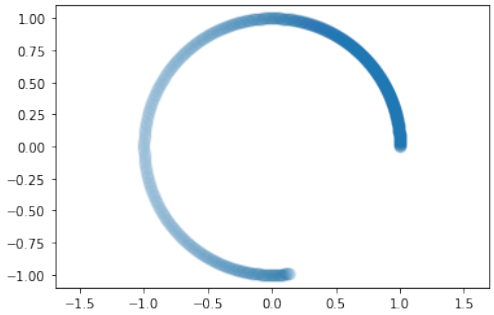


Figure 26: Asymmetric distribution on the circle.

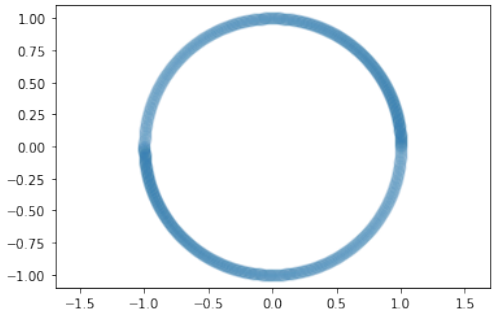
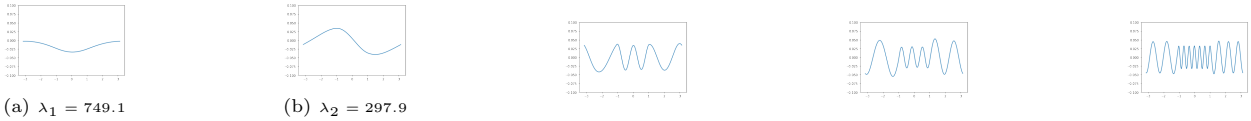


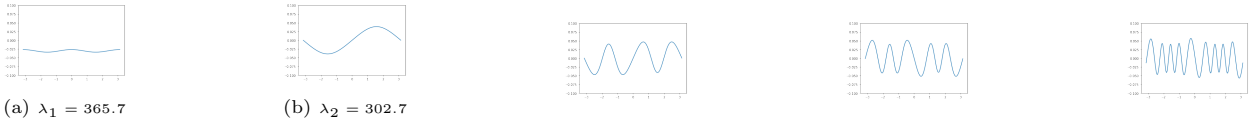
Figure 27: Antipodal distribution on the circle.

5.4.1 Eigenvectors of NTK for different distributions



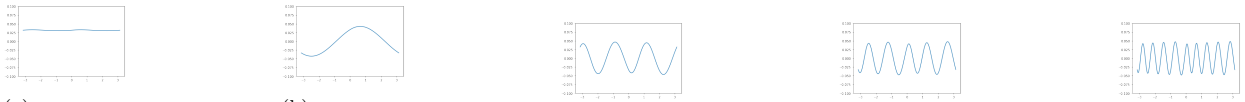
(a) $\lambda_1 = 749.1$ (b) $\lambda_2 = 297.9$

Figure 28: Eigenvectors of NTK for piecewise constant distribution.



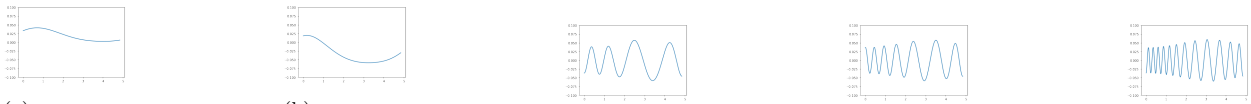
(a) $\lambda_1 = 365.7$ (b) $\lambda_2 = 302.7$

Figure 29: Eigenvectors of NTK for continuous distribution (wrapped normal).



(a) $\lambda_1 = 353.0$ (b) $\lambda_2 = 251.3$

Figure 30: Eigenvectors of NTK for antipodal distribution.



(a) $\lambda_1 = 522.3$ (b) $\lambda_2 = 180.5$

Figure 31: Eigenvectors of NTK for asymmetric distribution.

5.4.2 Fourier analysis of circular distributions

This section is based on the book [MJ09].

Given $p(\theta)$ a distribution function defined on the unit circle. Analogous with distributions on the real line, we can define the characteristic function of a random angle θ as

$$\phi_k = \alpha_k + i\beta_k, \tag{111}$$

where

$$\alpha_k = \mathbb{E}[\cos k\theta] = \int_0^{2\pi} \cos k\theta dp(\theta) \tag{112}$$

and

$$\beta_k = \mathbb{E}[\sin k\theta] = \int_0^{2\pi} \sin k\theta dp(\theta). \tag{113}$$

The complex numbers $\{\phi_k : k = 0, \pm 1, \dots\}$ are the Fourier coefficients of the distribution function p. If $\sum_{k=1}^{\infty}(\alpha_k^2 + \beta_k^2)$ is convergent, then the random variable θ has a density p which is defined almost everywhere by

$$p(\theta) = \frac{1}{2\pi} \sum_{p=-\infty}^{\infty} \phi_k e^{-ik\theta}, \tag{114}$$

or equivalently

$$p(\theta) = \frac{1}{2\pi} \left(1 + 2 \sum_{p=1}^{\infty} (\alpha_k cosk\theta + \beta_k sinp\theta) \right). \tag{115}$$

The above result is analogous to the inversion theorem for continuous random variables on the real line.

5.5 Training Experiments on non-uniform data

6 Further directions

- Generalization behavior and convergence rate for different activation functions, the choice of $\mathcal{H}^s$ should be affected by the activation function because that will change the H^∞ matrix Liu: intuitively smooth activations might benefit from $\mathcal{H}^2$ norm?
- Come up with a regime for choosing the $\mathcal{H}^s$ norm in the domain of $\mathbb{S}^1$.
- Set up the experiments in the domain of $\mathbb{S}^1$.
- Preliminary experiments and literature review on whether the same frequency bias is observed when the input data is not uniformly distributed on $\mathbb{S}^1$.

7 Week 6. July 19 - July 23

The first task this week is to check whether the Funk-Hecke formula still applies for non-uniform data, and if so, in which types of non-uniform distributions.

7.1 Frames and Riesz Bases

Definition 28 (Frames)**.**

Definition 29 (Riesz Basis)**.**

7.1.1 Properties

7.1.2 Relation between Frames and Riesz Bases

7.2 Stability of Riesz Bases – The Paley-Wiener Theorem and the Kadec-1/4 Theorem

8 Week 7. July 26 - July 29

8.1 Applying Kadec-1/4 theorem to non-uniform data: specialize to only real perturbations

Let $A \in \mathbb{R}^{N\times N}$, where N is odd, be an inverse non-uniform discrete Fourier transform (INDFT) matrix with entries $A_{jk} = e^{i\lambda_k t_j}$, with $\lambda_k \in \mathbb{C}$, $k = 0 \pm 1, \pm 2, \ldots, \pm(N-1)/2$. Let $t_j = -\pi + 2\pi k/N$ for $j = 0, 1, \ldots, N-1$ so that $\{t_j\}_{j=0}^{N-1}$ is a collection of equally spaced samples on the interval $[-\pi, \pi)$. Let F be an unnormalized IDFT (inverse DFT) matrix, with entries $F_{jk} = e^{ip_k t_j}$, where $p_k = -(N-1)/2 + k$ for $k = 0, \ldots, N-1$. Note that the singular values of F are all $\sqrt{N}$. We would like to put a condition on λ_k such that λ_k is not too far perturbed from p_k. In this case, we restrict the perturbations to be real numbers, writing $\lambda_k = \eta_k$, with $\eta_k \in \mathbb{R}$. We now require that $|\eta_k - p_k| \leqslant L < 1/4$, which is the condition used in proving Kadec-1/4 theorem [Vel15]. Under this condition, we have the following bound on the condition number of matrix A, $\kappa(A)$:

$$\kappa(A) = \|A\|_2\|A^{-1}\|_2 \leqslant \frac{1+\phi_L}{1-\phi_L}, \qquad \phi_L = 1 - \cos\pi L + \sin\pi L. \tag{116}$$

To bound $\|A^{-1}\|_2$ and $\|A\|_2$, we consider the matrix $E = F - A$. We have from Weyl's inequality on singular values that

$$|\sigma_{\min}(F) - \sigma_{\min}(A)| \leqslant \|E\|_2,$$

so that $\sigma_{\min}(A) \geqslant \sqrt{N} - \|E\|_2$.

The entries of $F - A$ can be expressed as

$$\begin{aligned}(F-A)_{jk} &= e^{ip_k t_j} - e^{i\lambda_k t_j} \\ &= e^{ip_k t_j}\left(1 - e^{i\eta_k t_j - ip_k t_j}\right) \\ &= e^{ip_k t_j}\left(1 - e^{i\delta_k t_j}\right) (\delta \coloneqq \eta_k - p_k).\end{aligned}$$

Let $\{c_k\}_{k=0}^{N-1}$ be any collection of scalars such that $\sum_{k=0}^{N-1}|c_k|^2 = 1$. We seek a bound on $\|(F-A)\vec{c}\|_2$. We can write this using the column vectors of $F - A$ as follows:

$$\|(F-A)\vec{c}\|_2 = \left\|\sum_{k=0}^{N-1} e^{ip_k\vec{t}} \circ \left(\vec{1} - -e^{i\delta_k\vec{t}}\right) c_k\right\|_2,$$

where $\vec{t} = (t_0, \ldots, t_{N-1})$, $\vec{1}$ is a vector of ones, the exponential is applied componentwise, and '$\circ$' is the Hadamard matrix product.

Using the expansion of entries $1 - e^{i\delta_k t_j}$ given in [You01, Ch 1.10], we have for each k that

$$\vec{1} - e^{i\delta_k\vec{t}} = \underbrace{\left(1 - \frac{\sin\pi\delta_k}{\pi\delta_k}\right)\vec{1}}_{A_k} + \underbrace{\sum_{\ell=1}^{\infty}\frac{(-1)^\ell 2\delta_k\sin\pi\delta_k}{\pi(\ell^2-\delta_k^2)}\cos\ell\vec{t}}_{B_k} + \underbrace{i\sum_{\ell=1}^{\infty}\frac{(-1)^\ell 2\delta_k\cos\pi\delta_k}{\pi(\ell-\frac{1}{2})^2-\pi\delta_k^2}\sin\left(\ell-\frac{1}{2}\right)\vec{t}}_{C_k}, \tag{117}$$

So

$$\|(F-A)\vec{c}\|_2 = \left\|\sum_{k=0}^{N-1}\left(\vec{1} - e^{i\delta_k\vec{t}}\right)\circ e^{ip_k\vec{t}}c_k\right\|_2 \leqslant \left\|\sum_{k=0}^{N-1}A_k\circ e^{ip_k\vec{t}}c_k\right\|_2 + \left\|\sum_{k=0}^{N-1}B_k\circ e^{ip_k\vec{t}}c_k\right\|_2 + \left\|\sum_{k=0}^{N-1}C_k\circ e^{ip_k\vec{t}}c_k\right\|_2. \tag{118}$$

Now we bound each term in (118). Since $\max_k \left(1 - \frac{\sin \pi\delta_k}{\pi\delta_k}\right) \leqslant 1 - \frac{\sin \pi L}{\pi L}$, we have that

$$\left\| \sum_{k=0}^{N-1} A_k \circ e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \left(1 - \frac{\sin \pi L}{\pi L}\right) \sqrt{N}. \tag{119}$$

Writing out B_k in the second term in (118) explicitly, we switch the order of summation and apply the triangle inequality to find that

$$\left\| \sum_{k=0}^{N-1} B_k \circ e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \sum_{\ell=1}^{\infty} \left\| \sum_{k=0}^{N-1} \frac{(-1)^\ell 2\delta_k \sin \pi\delta_k}{\pi(\ell^2 - \delta_k^2)} \cos \ell\vec{t} \circ e^{ip_k \vec{t}} c_k \right\|_2 .$$

Using the orthogonality of the vectors $\{e^{ip_k \vec{t}}\}_{k=0}^{N-1}$ and the fact that $2x \sin \pi x/(\pi\ell^2 - \pi x^2)$ is an increasing function on $x \in [0, 1/4]$ and a decreasing function on $x \in [-1/4, 0]$, we have that

$$\left\| \sum_{k=0}^{N-1} B_k \circ e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \sum_{\ell=1}^{\infty} \max_{j,k} \left| \frac{2\delta_k \sin \pi\delta_k}{\pi(\ell^2 - \delta_k^2)} \cos \ell t_j \right| \left\| \sum_{k=0}^{N-1} e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \sum_{\ell=1}^{\infty} \frac{2L \sin \pi L}{\pi(\ell^2 - L^2)} \sqrt{N}.$$

We note that the series $\sum_{\ell=1}^{\infty} 2L/(\pi\ell^2 - \pi L^2)$ is the partial fraction expansion of the function $1/(\pi L) - \cot \pi L$, so that

$$\left\| \sum_{k=0}^{N-1} B_k \circ e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \sqrt{N} \left(\frac{\sin \pi L}{\pi L} - \cos \pi L \right). \tag{120}$$

By an argument similar to this one, we find for the third term in (118) that

$$\left\| \sum_{k=0}^{N-1} C_k \circ e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \sum_{\ell=1}^{\infty} \frac{2L \cos \pi L}{\pi(\ell - \frac{1}{2})^2 - \pi L^2} \sqrt{N}.$$

The series $\sum_{\ell=1}^{\infty} 2L/(\pi(\ell - 1/2)^2 - \pi L^2)$ is the partial fraction expansion of $\tan \pi L$, so we have that

$$\left\| \sum_{k=0}^{N-1} C_k \circ e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \sqrt{N} \sin \pi L. \tag{121}$$

It follows from applying (119), (120) and (121) to (118) that

$$\left\| \sum_{k=0}^{N-1} \left(\vec{1} - e^{i\delta_k \vec{t}} \right) \circ e^{ip_k \vec{t}} c_k \right\|_2 \leqslant \sqrt{N} \left(1 - \cos \pi L + \sin \pi L\right). \tag{122}$$

Applying this to (118), we find that

$$\|(F - A)\vec{c}\|_2 \leqslant (1 - \cos \pi L + \sin \pi L) \sqrt{N}, \tag{123}$$

Therefore, under the condition that $L < \frac{1}{4}$, $1 - \cos \pi L + \sin \pi L < 1$ and we have the following bound on $\|E\|_2$:

$$\|E\|_2 = \|(F - A)\|_2 \leqslant (1 - \cos \pi L + \sin \pi L) \sqrt{N} < \sqrt{N}. \tag{124}$$

9 AT's thoughts

We suppose that we have a finite set $\mathcal{C}$ comprising of K distinct points that are located at angles $-\pi = \xi_0 < \xi_1 < \cdots < \xi_{K-1} < \xi_K = \pi$. Set $R_j = \left[\frac{\xi_{j-1} + \xi_j}{2}, \frac{\xi_j + \xi_{j+1}}{2}\right]$ for $j = 0, \ldots, K-1$ (be careful about going over π). We take $\mu(R_j) = \xi_{j+1} - \xi_j$, which is the arc length of the interval. The mesh norm δ_C and $\|R\|$ are given by

$$\delta_C = \frac{1}{2} \max_{0 \leqslant j \leqslant K-1} |\xi_{j+1} - \xi_j|, \qquad \|R\| = \max \operatorname{diam}(R_j) = 2\delta_C.$$

Let

$$E_C(P) = \left| \sup_{\theta \in \mathbb{S}^1} |P(\theta)| - \max_{\xi \in C} |P(\xi)| \right|.$$

It is easy to verify that for any trigonometric polynomial, P, we have

$$\sup_{\theta \in R_\xi} |P(\theta) - P(\xi)| \leqslant \frac{1}{2} \|R\| \|P'\|_{\mathbb{S}^1,\infty}.$$

By the Bernstein inequality in L^∞, we have $\|P'\|_{\mathbb{S}^1,\infty} \leqslant n\|P\|_{\mathbb{S}^1,\infty}$. Thus, we have the following lemma:

Lemma 1.

$$E_C(P) \leqslant \max_{\xi \in C} \sup_{\theta \in R_\xi} |P(\theta) - P(\xi)| \leqslant \frac{n}{2} \|R\| \|P\|_{\mathbb{S}^1,\infty}.$$

If $\|R\| \leqslant 2c\eta/n$ for some $\eta > 0$, then

$$(1 - c\eta) \sup_{\theta \in \mathbb{S}^1} |P(\theta)| \leqslant \max_{\xi \in C} |P(\xi)| \leqslant (1 + c\eta) \sup_{\theta \in \mathbb{S}^1} |P(\theta)|.$$

We take $X = \Pi_n^1$, $\|P\|_X = \|P\|_{\mathbb{S}^1,\infty}$, and Z to be the set of point evaluation functionals $\{\delta_\xi\}_{\xi \in C}$. The operator T_Z is then the restriction map $P \mapsto P|_C$, equipped with the norm $\|P|_C\| = \max_{\xi \in C} |P(\xi)|$. We now take y to be the functional

$$y : P \mapsto \frac{1}{2\pi} \int_{\mathbb{S}^1} P(\theta) d\theta$$

By Hölder's inequality, we have $\|y\|_{X^*} \leqslant 1$. Therefore, we find that

$$\frac{1}{2\pi} \int_{\mathbb{S}^1} P(\theta) d\theta = \sum_{\xi \in C} a_\xi P(\xi)$$

provided that $\sum_{\xi \in C} |a_\xi| \leqslant (1 - \eta)^{-1}$. To see if the weights are positive,

References

[ADH+19] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

[BJKK19] Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. *arXiv preprint arXiv:1906.00425*, 2019.

[BLPE20] Juan Antonio Barceló, Teresa Luque, and Salvador Pérez-Esteva. Characterization of sobolev spaces on the sphere. *Journal of Mathematical Analysis and Applications*, 491(1):124240, 2020.

[BM] Alberto Bietti and Julien Mairal. On the Inductive Bias of Neural Tangent Kernels. page 12.

[CFW+20] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards Understanding the Spectral Bias of Deep Learning. *arXiv:1912.01198 [cs, stat]*, October 2020. arXiv: 1912.01198.

[DZPS19] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. 2019.

[FE12] Christopher Frye and Costas J Efthimiou. Spherical harmonics in p dimensions. *arXiv preprint arXiv:1205.3548*, 2012.

[GMMM20] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *arXiv:1904.12191 [cs, math, stat]*, February 2020. arXiv: 1904.12191.

[GSW10] QT Le Gia, Ian H Sloan, and Holger Wendland. Multiscale analysis in sobolev spaces on the sphere. *SIAM journal on numerical analysis*, 48(6):2065–2090, 2010.

[JGH] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. page 19.

[LMXZ20] Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang. On the exact computation of linear frequency principle dynamics and its generalization. *arXiv:2010.08153 [cs]*, October 2020. arXiv: 2010.08153.

[MJ09] Kanti V Mardia and Peter E Jupp. *Directional statistics*, volume 494. John Wiley & Sons, 2009.

[PSG19] Abhishek Panigrahi, Abhishek Shetty, and Navin Goyal. Effect of activation functions on the training of overparametrized neural nets. *arXiv preprint arXiv:1908.05660*, 2019.

[RBA+19] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[SH21] Meyer Scetbon and Zaid Harchaoui. A spectral analysis of dot-product kernels. In *International Conference on Artificial Intelligence and Statistics*, pages 3394–3402. PMLR, 2021.

[Tao] Terrence Tao. 245c notes 4: Sobolev spaces.

[Vel15] Pierluigi Vellucci. A simple pointview for kadec-1/4 theorem in the complex case. *Ricerche di Matematica*, 64(1):87–92, 2015. Communicated by Salvatore Rionero.

[VW19] Santosh Vempala and John Wilmes. Gradient Descent for One-Hidden-Layer Neural Networks: Polynomial Convergence and SQ Lower Bounds. *arXiv:1805.02677 [cs, stat]*, May 2019. arXiv: 1805.02677.

[You01] Robert M. Young. *An Introduction to nonharmonic Fourier series*. Academic Press, 2001.

[YTA20] Yunan Yang, Alex Townsend, and D. Appelo. Anderson acceleration using the $\mathcal{H}^{-s}$ norm. *arXiv: Numerical Analysis*, 2020.