ETH zürich

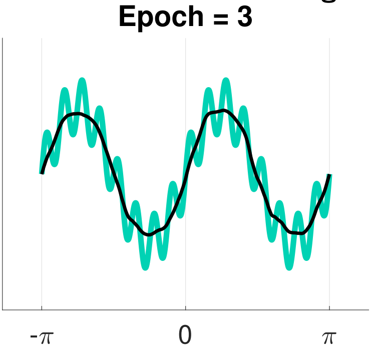
Frequency Bias in Neural Networks for Input of Non-Uniform Density

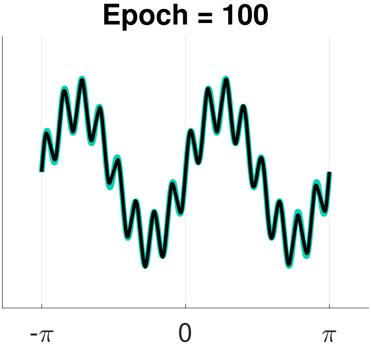
Aurelien Lucchi

1. Motivation

Motivation: Frequency Bias

Observation when training with SGD on uniform data

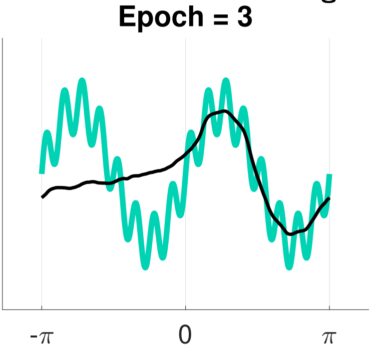


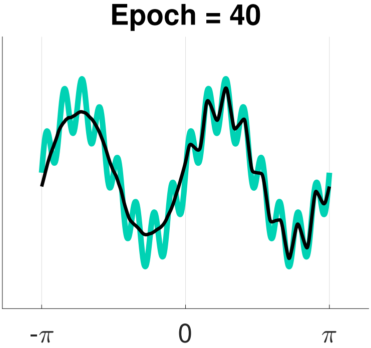


The light cyan line represents the target function which is composed of the sum of a low and high frequency functions. The thin black line represents the network output

Motivation: Frequency Bias

Observation when training with SGD on non-uniform data





Motivation: Is there a link between generalization and the tendency of overparametrized networks to exhibit frequency bias?

Motivation: Eigenfunctions of NTK

- The NTK for the training data is summarized in an $n \times n$ matrix H^{∞}, whose entries are set to $H^{\infty}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.
- Let $\mathbf{v}_i$ and λ_i respectively denote the eigenvectors of H^{∞} and their corresponding eigenvalues.

Motivation: Eigenfunctions of NTK

- The NTK for the training data is summarized in an $n \times n$ matrix H^∞, whose entries are set to $H^\infty_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.
- Let $\mathbf{v}_i$ and λ_i respectively denote the eigenvectors of H^∞ and their corresponding eigenvalues.
- Arora et al. (2019) showed the role of the eigenfunctions on the speed of convergence:

$$\|\mathbf{y} - \mathbf{u}^{(t)}\| = \sqrt{\sum_{i=1}^{n} (1 - \eta\lambda_i)^{2t} (\mathbf{v}_i^T \mathbf{y})^2} \pm \epsilon. \qquad (1)$$

i.e. rate depends on projection of target $\mathbf{y}$ onto eigenvectors $\mathbf{v}_i$

2. Recall NTK

NTK

- Notation: Denote by $f(\mathbf{w}, \boldsymbol{x}) \in \mathbb{R}$ the output of a neural network
  - $\mathbf{w} \in \mathbb{R}^N$ is all the parameters in the network
  - $\boldsymbol{x} \in \mathbb{R}^d$ is the input

NTK

- Notation: Denote by $f(\mathbf{w}, \boldsymbol{x}) \in \mathbb{R}$ the output of a neural network
  - $\mathbf{w} \in \mathbb{R}^N$ is all the parameters in the network
  - $\boldsymbol{x} \in \mathbb{R}^d$ is the input
- NTK is defined using the *gradient* of the output of the randomly initialized net with respect to its parameters, i.e.,

$$k\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathop{\mathbb{E}}_{\mathbf{w} \sim \mathcal{W}} \left\langle \frac{\partial f(\mathbf{w}, \boldsymbol{x})}{\partial \mathbf{w}}, \frac{\partial f(\mathbf{w}, \boldsymbol{x}')}{\partial \mathbf{w}} \right\rangle . \tag{2}$$

Network

- Define L-hidden-layer fully-connected neural network:

$$\begin{aligned} \boldsymbol{f}^{(h)}(\boldsymbol{x}) &= \boldsymbol{W}^{(h)}\boldsymbol{g}^{(h-1)}(\boldsymbol{x}) \in \mathbb{R}^{d_h} \\ \boldsymbol{g}^{(h)}(\boldsymbol{x}) &= \sqrt{\frac{c_\sigma}{d_h}}\sigma\left(\boldsymbol{f}^{(h)}(\boldsymbol{x})\right) \in \mathbb{R}^{d_h}, \qquad h = 1, 2, \ldots, L, \end{aligned} \tag{3}$$

where $\boldsymbol{W}^{(h)} \in \mathbb{R}^{d_h \times d_{h-1}}$, $\sigma : \mathbb{R} \to \mathbb{R}$,
$c_\sigma = \left(\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\sigma\left(z\right)^2\right]\right)^{-1}$.

Network

- Last layer of the neural network is

$$f^{(L+1)}(\boldsymbol{x}) = \boldsymbol{W}^{(L+1)} \cdot \boldsymbol{g}^{(L)}(\boldsymbol{x})$$

$$= \boldsymbol{W}^{(L+1)} \cdot \sqrt{\frac{c_\sigma}{d_L}}\sigma\left(\boldsymbol{W}^{(L)} \cdot \sqrt{\frac{c_\sigma}{d_{L-1}}}\sigma\left(\boldsymbol{W}^{(L-1)} \cdots \sqrt{\frac{c_\sigma}{d_1}}\sigma\left(\boldsymbol{W}^{(1)}\boldsymbol{x}\right)\right)\right),$$

where $\mathbf{w} = \left(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(L+1)}\right)$

NTK (Lee et al. (2018))

- $\boldsymbol{f}^{(h)}(\boldsymbol{x}) = \boldsymbol{W}^{(h)}\boldsymbol{g}^{(h-1)}(\boldsymbol{x}) \in \mathbb{R}^{d_h}$
- Recall from Lee et al. (2018) that in the infinite width limit, the pre-activations $\boldsymbol{f}^{(h)}(\boldsymbol{x})$ at every hidden layer $h \in [L]$ has all its coordinates tending to i.i.d. centered Gaussian processes of covariance $\Sigma^{(h-1)} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined recursively as:

$$\begin{aligned}
\Sigma^{(0)}(\boldsymbol{x}, \boldsymbol{x}') &= \boldsymbol{x}^\top \boldsymbol{x}', \\
\boldsymbol{\Lambda}^{(h)}(\boldsymbol{x}, \boldsymbol{x}') &= \begin{pmatrix} \Sigma^{(h-1)}(\boldsymbol{x}, \boldsymbol{x}) & \Sigma^{(h-1)}(\boldsymbol{x}, \boldsymbol{x}') \\ \Sigma^{(h-1)}(\boldsymbol{x}', \boldsymbol{x}) & \Sigma^{(h-1)}(\boldsymbol{x}', \boldsymbol{x}') \end{pmatrix} \in \mathbb{R}^{2\times 2}, \qquad (4) \\
\Sigma^{(h)}(\boldsymbol{x}, \boldsymbol{x}') &= c_\sigma \underset{(u,v)\sim\mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}^{(h)}\right)}{\mathbb{E}} \left[\sigma(u)\,\sigma(v)\right].
\end{aligned}$$

3. Two-layer ReLU network

Setting

- Training by minimizing the squared loss:

$$\Phi(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i, \mathbf{w}))^2. \tag{5}$$

- Assume that n training points are sampled i.i.d. from an arbitrary distributions $p(x)$ on the hypersphere and $y_i = g(x_i)$ where g is unknown.
- First consider a two-layer ReLU network where only the first layer is trained (Arora et al. 2019)

Two-layer ReLU network

We first consider a two layer network with bias:

$$f(\mathbf{x}, \mathbf{w}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(\mathbf{w}_r^T \mathbf{x} + b_r), \tag{6}$$

where $\|\mathbf{x}\| = 1$ (denoted $\mathbf{x} \in \mathbb{S}^{d-1}$) is the input, $\mathbf{w}$ includes the weights and bias terms of the first layer, denoted respectively $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_m] \in \mathbb{R}^{d \times m}$ and $\mathbf{b} = [b_1, ..., b_m]^T \in \mathbb{R}^m$, as well as the weights of the second layer, denoted $\mathbf{a} = [a_1, ..., a_m]^T \in \mathbb{R}^m$.

NTK Derivation

Kernel matrix Recalling $w(0) \sim \mathbf{N}(0, \mathbf{I}))$, we get

$$\begin{aligned} k(x_i, x_j) &= \mathbb{E}_{\mathbf{w} \sim \mathbf{N}(0,\mathbf{I})} \left[\mathbf{x}_i^\top \mathbf{x}_j \mathbb{I} \left\{ \mathbf{w}^\top \mathbf{x}_i \geq 0, \mathbf{w}^\top \mathbf{x}_j \geq 0 \right\} \right] \\ &= \frac{\mathbf{x}_i^\top \mathbf{x}_j \left(\pi - \arccos(\mathbf{x}_i^\top \mathbf{x}_j) \right)}{4\pi}, \quad \forall i, j \in [n]. \end{aligned} \tag{7}$$

NTK Derivation

Kernel matrix Recalling $w(0) \sim \mathbf{N}(0, \mathbf{I}))$, we get

$$\begin{aligned} k(x_i, x_j) &= \mathbb{E}_{\mathbf{w} \sim \mathbf{N}(0,\mathbf{I})} \left[\mathbf{x}_i^\top \mathbf{x}_j \mathbb{I} \left\{ \mathbf{w}^\top \mathbf{x}_i \geq 0, \mathbf{w}^\top \mathbf{x}_j \geq 0 \right\} \right] \\ &= \frac{\mathbf{x}_i^\top \mathbf{x}_j \left(\pi - \arccos(\mathbf{x}_i^\top \mathbf{x}_j) \right)}{4\pi}, \quad \forall i, j \in [n]. \end{aligned} \tag{7}$$

- First formula: $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\mathbf{w} \sim \mathcal{W}} \left\langle \frac{\partial f(\mathbf{w}, \boldsymbol{x})}{\partial \mathbf{w}}, \frac{\partial f(\mathbf{w}, \boldsymbol{x}')}{\partial \mathbf{w}} \right\rangle$
- Second formula: comes from the rotation invariance property of the Gaussian distribution.

Contribution

- **Prior work:** When the training data is distributed uniformly, eigenfunctions of kernel are the spherical harmonics on the hypersphere.

Contribution

- **Prior work:** When the training data is distributed uniformly, eigenfunctions of kernel are the spherical harmonics on the hypersphere.
- **This paper:** Focus on 1-dimensional target $y(x) : \mathbb{S}^1 \to \mathbb{R}$ and a piecewise constant data distribution $p(\mathbf{x})$
  - Derive explicit expressions for the eigenfunctions and eigenvalues of NTK
  - Prove that learning a one-dimensional function of frequency κ requires $O(\kappa^2/p^*)$ iterations, where p^* denotes the minimal density in $p(x)$.
- Empirical results for higher dimensions

Eigen-system of H^p

- Suppose n training data points are sampled from a non-uniform, piecewise constant distribution $p(\mathbf{x})$ on the circle, $\mathbf{x} \in \mathbb{S}^1$.
- Form an $n \times n$ matrix H^p whose entries for samples $\mathbf{x}_i$ and $\mathbf{x}_j$ consist of $H^p_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

Eigen-system of H^p

- Suppose n training data points are sampled from a non-uniform, piecewise constant distribution $p(\mathbf{x})$ on the circle, $\mathbf{x} \in \mathbb{S}^1$.
- Form an $n \times n$ matrix H^p whose entries for samples $\mathbf{x}_i$ and $\mathbf{x}_j$ consist of $H^p_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

Notations (expect some typos, sorry!):

- Denote by $\mathbf{u}^t(\mathbf{x}) = f(\mathbf{W}^{(t)}, \mathbf{x})$ the prediction of the network at iteration t of GD
- $$H^\infty_{ij} = \mathbb{E}_{\mathbf{W} \sim \mathbf{N}(0,\mathbf{I})} \left\langle \frac{\partial \mathbf{u}^{(0)}(\mathbf{x}_i)}{\partial \mathbf{W}}, \frac{\partial \mathbf{u}^{(0)}(\mathbf{x}_j)}{\partial \mathbf{W}} \right\rangle.$$

Eigen-system of H^p

- Following Arora et al. 2019, convergence rates of GD for such a network will depend on the eigen-system of H^p

$$\|\mathbf{y} - \mathbf{u}^{(t)}\| = \sqrt{\sum_{i=1}^{n}(1-\eta\lambda_i)^{2t}(\mathbf{v}_i^T\mathbf{y})^2} \pm \epsilon. \qquad (8)$$

i.e. rate depends on projection of target **y** onto eigenvectors $\mathbf{v}_i$

Theorem (Arora et al. 2019)

With high probability we have:

$$\Phi(\mathbf{W}) \approx \frac{1}{2}\left\|(\boldsymbol{I} - \eta\boldsymbol{H}^{\infty})^k\boldsymbol{y}\right\|_2^2, \quad \forall k \geq 0.$$

Eigen-system of H^p

- Consider the limit of H^p as $n \to \infty$.
- Williams et al 2000: Eigen-system of H^p approaches the eigen-system of the kernel $k(\mathbf{x}_i, \mathbf{x}_j)p(\mathbf{x}_j)$, where the eigenfunctions $f(x)$ satisfy the following equation:

$$\boxed{\int_{\mathbb{S}^1} k(\mathbf{x}_i, \mathbf{x}_j)p(\mathbf{x}_j)f(\mathbf{x}_j)d\mathbf{x}_j = \lambda f(\mathbf{x}_i). \qquad (9)}$$

(homogeneous Fredholm Equation of the second kind with the non-symmetric polar kernel $k(\mathbf{x}_i, \mathbf{x}_j)p(\mathbf{x}_j)$)

Reparamatrization

- Parameterize the unit circle by angles, and denote by x, z any two angles. Eq. (9) becomes

$$\int_{x-\pi}^{x+\pi} k(x,z)p(z)f(z)dz = \lambda f(x), \tag{10}$$

where the kernel expressed in terms of angles reads

$$\boxed{k(x,z) = \frac{1}{4\pi}(\cos(x-z)+1)(\pi - |x-z|). \tag{11}}$$

- Both $p(x)$ and $f(x)$ are periodic with a period of 2π since x lies on the unit circle.

Proposition: Eigenfunctions

Proposition

Let $p(x)$ be a piecewise constant density function on $\mathbb{S}^1$. Then the eigenfunctions in (10) take the general form

$$f(x) = a(p(x)) \cos\left(\frac{q}{Z}\Psi(x) + b(p(x))\right), \qquad (12)$$

where q is integer, $\Psi(x) = \int_{-\pi}^{x} \sqrt{p(\tilde{x})} d\tilde{x}$ and $Z = \frac{1}{2\pi}\Psi(\pi)$.

Proposition: Eigenfunctions

If $p(x) = p_j$ is constant in a connected region $R_j \subseteq \mathbb{S}^1$, then

$$\boxed{f(x) = a_j \cos\left(\frac{q\sqrt{p_j}x}{Z} + b_j\right), \forall x \in R_j. \qquad (13)}$$

I.e., over the region R_j, this is a cosine function with frequency proportional to $\sqrt{p_j}$.

Proof Sketch Proposition

- The proof of the proposition relies on a lemma, stating that the solution to (10) satisfies the following second order ordinary differential equation (ODE):

$$f''(x) = -\frac{p(x)}{\pi\lambda} f(x). \tag{14}$$

Proof Sketch Proposition

- The proof of the proposition relies on a lemma, stating that the solution to (10) satisfies the following second order ordinary differential equation (ODE):

$$f''(x) = -\frac{p(x)}{\pi\lambda} f(x). \tag{14}$$

- In a nutshell, the lemma proved by applying a sequence of six derivatives to (10) with respect to x, along with some algebraic manipulations, yielding a sixth order ODE for $f(x)$. Assuming that $p(x)$ is piecewise constant simplifies the ODE.

Proof Sketch Proposition

- Eq. (14) has the following general solutions

$$f(x) = Ae^{i\frac{\Psi(x)}{\sqrt{\pi\lambda}}x} + Be^{-i\frac{\Psi(x)}{\sqrt{\pi\lambda}}x}, \tag{15}$$

- Resulting real eigenfunctions are

$$f(x) = a(p(x))\cos\left(\frac{\Psi(x)}{\sqrt{\pi\lambda}}x + b(p(x))\right). \tag{16}$$

Proof Sketch Proposition

- Eq. (14) has the following general solutions

$$f(x) = Ae^{i\frac{\Psi(x)}{\sqrt{\pi\lambda}}x} + Be^{-i\frac{\Psi(x)}{\sqrt{\pi\lambda}}x}, \tag{15}$$

- Resulting real eigenfunctions are

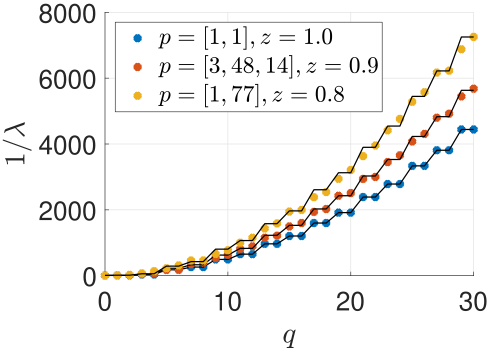
$$f(x) = a(p(x))\cos\left(\frac{\Psi(x)}{\sqrt{\pi\lambda}}x + b(p(x))\right). \tag{16}$$

- Eigenvalues:

$$\lambda = \begin{cases} Z^2\left(\frac{1}{2\pi^2} + \frac{1}{8}\right) & q = 0 \\ Z^2\left(\frac{1}{\pi^2} + \frac{1}{8}\right) & q = 1 \\ \frac{Z^2(q^2+1)}{\pi^2(q^2-1)^2} & q \geq 2 \text{ even} \\ \frac{Z^2}{\pi^2 q^2} & q \geq 2 \text{ odd.} \end{cases} \tag{17}$$

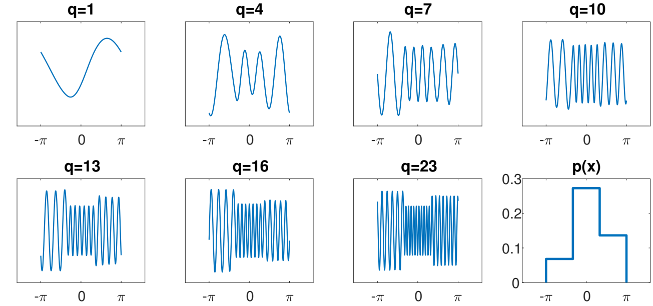
  q is integer, and there is one eigenfunction for $q = 0$ and two eigenfunctions for every $q > 0$.

Plot eigenvalues



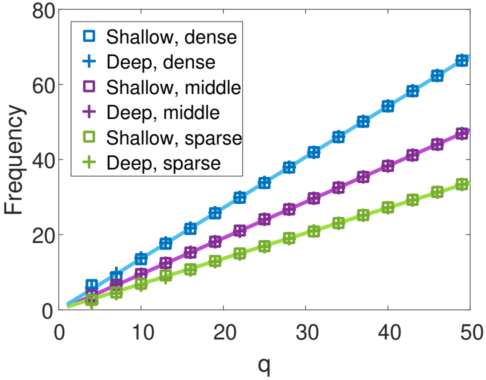
The kernel eigenvalues for several distributions. The formula (marked by the solid lines) closely matches the eigenvalues H^p computed numerically using $50K$ points.

Eigenfunctions



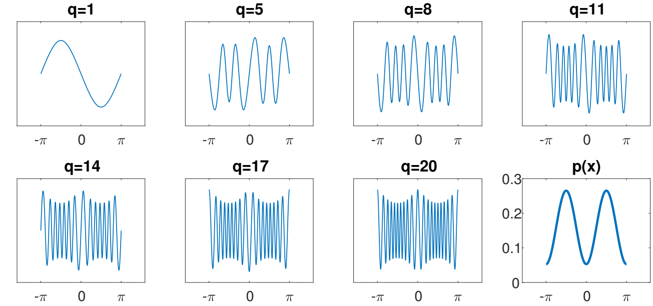
For the NTK of a two-layer network with bias we plot its eigenfunctions (in a decreasing order of eigenvalues) under a non-uniform data distribution in $\mathbb{S}^1$. Here we used a density composed of three constant regions with $p(x) \in 3/(2\pi)\{1/7, 2/7, 4/7\}$ (bottom right plot).

Eigenfunctions



The local frequency in the eigenfunctions within each of the three constant region densities in Figure 2, plotted for both a two-layer and deep (depth=10) networks (marked respectively by squares and plus signs). Measurements are obtained by applying FFT to each region. The measurements are in close match to our formula (13) (solid line).

Eigenfunctions (continuous $p(x)$)



For the NTK of a two-layer network we plot the eigenvectors of H^p for a continuous distribution, $p(x) = \frac{3\cos(2x+\pi)+4.5}{9\pi}$ (bottom right).

Convergence rate

- Consider 1d target functions of the form $g(x) = \cos(\kappa x)$ where x is drawn from a piecewise constant distribution $p(x)$ on $\mathbb{S}^1$

Theorem

Let $p(x)$ be a piecewise constant distribution on $\mathbb{S}^1$. Denote by $u^{(t)}(x)$ the prediction of the network at iteration t of GD. For any $\delta > 0$ the number of iterations t needed to achieve $\|g(x) - u^{(t)}(x)\| < \delta$ is $\tilde{O}(\kappa^2/p^*)$,
where p^* denotes the minimal density of $p(x)$ in $\mathbb{S}^1$ and $\tilde{O}(.)$ hides logarithmic terms.

Proof convergence rate

- Eigen-decomposition of H^∞:

$$H^\infty = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T, \tag{18}$$

  where $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are the eigenvectors of H^∞ and $\lambda_1, \ldots, \lambda_n$ are their corresponding eigenvalues.

- Minimal eigenvalue denoted by $\lambda_0 = \min(\lambda(H^\infty))$.
- **Assumption:** $p(x)$ is piecewise constant (PCD) with l fixed pieces, $p(x) = p_j$ in R_j, $1 \leq j \leq l$

Proof convergence rate

- Next Lemma: not too many eigenfunctions dictate the convergence rate

Lemma

Let $p(x)$ be PCD. For any $\epsilon > 0$, there exist n_k such that $\sum_{j=n_k+1}^{\infty} g_i^2 < \epsilon^2$, where $g_i = \int_{-\pi}^{\pi} v_i(x)g(x)p(x)dx$ and n_k is bounded below.

Proof Lemma

- Given a target function $g(x) = \cos(kx)$ and a basis function $v_i(x) = a(x)\cos(\frac{q_i\sqrt{p(x)}x}{Z} + b(x))$ where $q_i = \lfloor i/2 \rfloor$, their inner product is

$$g_i = \sum_{j=1}^{l} a_j p_j \int_{R_j} \cos(kx)\cos(q_{ij}x + b_j)dx \tag{19}$$

where $q_{ij} = q_i\sqrt{p_j}/Z$ denotes the local frequency of $v_i(x)$ at R_j.

- Next, to derive a bound we will restrict our treatment to $q_{ij} \geq 2k$

- Bound $|g_i|$:

$$|g_i| \leq \frac{8}{3} \sum_{j=1}^{l} \frac{a_j p_j}{q_{ij}} \leq \frac{8}{3q_i^*} \sum_{j=1}^{l} a_j p_j = \frac{8B}{3q_i^*} = \frac{8BZ}{3q_i\sqrt{p^*}},$$

where we denote by $B = \sum_{j=1}^{l} a_j p_j$.

- Bound $|g_i|$:

$$|g_i| \leq \frac{8}{3} \sum_{j=1}^{l} \frac{a_j p_j}{q_{ij}} \leq \frac{8}{3q_i^*} \sum_{j=1}^{l} a_j p_j = \frac{8B}{3q_i^*} = \frac{8BZ}{3q_i\sqrt{p^*}},$$

  where we denote by $B = \sum_{j=1}^{l} a_j p_j$.

- Next, for a given $\epsilon > 0$ we wish to bound the sum $\sum_{i=n_k}^{\infty} g_i^2$ by starting from a sufficiently high index n_k, i.e.,

$$\sum_{i=n_k+1}^{\infty} g_i^2 \leq \left(\frac{8BZ}{3\sqrt{p^*}}\right)^2 \sum_{i=n_k+1}^{\infty} \frac{1}{q_i^2} < \frac{1}{q_{n_k}} \left(\frac{8BZ}{3\sqrt{p^*}}\right)^2 \overset{!}{<} \epsilon^2$$

- Find value of n_k such that above inequality is satisfied (also using q_i, $n_k \geq 2q_{n_k}$)

Theorem

Let $p(x)$ be a PCD, for any $\delta > 0$ the number of iterations t needed to achieve $\|g(x) - u^{(t)}(x)\| < \delta$ is $\tilde{O}(k^2/p^*)$, where $\tilde{O}$ hides logarithmic terms.

Proof Theorem

- Let $\hat{g}(x) = \sum_{i=1}^{n_k} g_i v(i)$
- Then,

$$\|g(x) - \hat{g}(x)\|^2 = \sum_{i=n_k+1}^{\infty} g_i^2 < \left(\frac{\delta}{2}\right)^2$$

Proof Theorem

- Let $\hat{g}(x) = \sum_{i=1}^{n_k} g_i v(i)$
- Then,

$$\|g(x) - \hat{g}(x)\|^2 = \sum_{i=n_k+1}^{\infty} g_i^2 < \left(\frac{\delta}{2}\right)^2$$

- Due to triangle inequality

$$\|g(x) - u^{(t)}(x)\| \leq \|g(x) - \hat{g}(x)\| + \|\hat{g}(x) - u^{(t)}(x)\|$$

$\implies$ suffices to find t such that

$$\|\hat{g}(x) - u^{(t)}(x)\| < \frac{\delta}{2} = \tilde{\delta}$$

Proof Theorem

- Using (Arora et al., 2019b)'s Theorem 4.1 adapted to continuous operators

$$\Delta^2 = \|\hat{g} - u^{(t)}\|^2 \approx \sum_{i=1}^{n_k} (1 - \eta\lambda_i)^{2t} g_i^2 \leq \pi \sum_{i=1}^{n_k} (1 - \eta\lambda_i)^{2t}$$

$$\leq \pi n_k (1 - \eta\lambda_{n_k})^{2t}$$

where the left inequality is due to $|g_i|^2 \leq \| \cos^2(kx) \| = \pi$

- Lower bound t to get required inequality.

4. Deep Network

Assumptions

- Network is trained with n pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$.
- Denote the vector of target values by $\mathbf{y} = (y_1, ..., y_n)$ and the network predictions for these values at time t by $\mathbf{u}^{(t)}$.
- Assume that the first and last layers are initialized and then held fixed throughout training, and the last layer is initialized randomly $\sim \mathcal{N}(0, \tau^2 I)$

Theorem

Theorem

For any $\epsilon \in (0,1]$ and $\delta \in (0, O(\frac{1}{L})]$, let $\tau = \Theta(\frac{\epsilon\hat{\delta}}{n})$, $m \geq \Omega\left(\frac{n^{24}L^{12}\log^5 m}{\delta^8\tau^6}\right)$, $\eta = \Theta\left(\frac{\delta}{n^4L^2m\tau^2}\right)$. Then, with probability of at least $1-\hat{\delta}$ over the random initialization after t GD iterations we have that

$$\|\mathbf{y} - \mathbf{u}^{(t)}\| = \sqrt{\sum_{i=1}^{n}(1-\eta\lambda_i)^{2t}(\mathbf{v}_i^T\mathbf{y})^2} \pm \epsilon. \qquad (20)$$

Theorem: proof sketch

- Show that for any number of layers and at any iteration t the following relation holds

$$\mathbf{u}^{(t+1)} - \mathbf{y} = (I - \eta H(t))(\mathbf{u}^{(t)} - \mathbf{y}) + \epsilon(t), \tag{21}$$

  where $H_{ij}(t) = \left\langle \frac{\partial f(\mathbf{x}_i, \mathbf{w}(t))}{\partial \mathbf{w}}, \frac{\partial f(\mathbf{x}_j, \mathbf{w}(t))}{\partial \mathbf{w}} \right\rangle$, and the residual $\epsilon(t)$ due to the GD steps is relatively small.

- Based on results due to Allen-Zhu et al. (2019) and Arora et al. (2019), show that $H(t)$ can be approximated by H^{∞}, i.e.

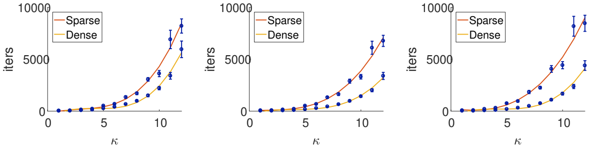
$$\mathbf{u}^{(t)} - \mathbf{y} = (I - \eta H^{\infty})^{t}(\mathbf{u}^{(0)} - \mathbf{y}) + \xi(t). \tag{22}$$

- Finally, apply spectral decomposition to H^{∞}

- **Next goal:** compute the eigenvectors and eigenvalues of NTK matrices for deep network

- **Next goal:** compute the eigenvectors and eigenvalues of NTK matrices for deep network
- Show that for uniformly distributed data, eigenfunctions are zonal spherical harmonics (spherical harmonics that are invariant under rotation).
- Show empirically:
  - In the non-uniform case, the eigenfunctions look similar to the uniform case.
  - Eigenvalues decay as $O(\kappa^{-d})$ for frequency κ
  - Convergence speed: $O(\frac{\kappa^{-d}}{p(x^*)})$

Empirical results



Convergence times as a function of the target harmonic frequency κ for a two-layer network trained with data drawn from a non-uniform distribution in $\mathbb{S}^2$. In each plot the sphere was divided into 2 halves, with density ratios (from left to right) of 1:2, 1:3, 1:4. The plot shows a cubic fit to the measurements. The median ratios between our measurements for the three subplots are 1.76, 2.45 and 2.99, undershooting our conjectured ratios. We believe this is due to sensitivity of experiments on $\mathbb{S}^2$ to sampling.

The end

Backup slides

NTK Derivation

- **Goal:** Compute the limit of $\left\langle \frac{\partial f(\mathbf{w},\boldsymbol{x})}{\partial \mathbf{w}}, \frac{\partial f(\mathbf{w},\boldsymbol{x}')}{\partial \mathbf{w}} \right\rangle$ at random initialization in the infinite width limit.
- Recall: $\boldsymbol{f}^{(h)}(\boldsymbol{x}) = \boldsymbol{W}^{(h)}\boldsymbol{g}^{(h-1)}(\boldsymbol{x}) \in \mathbb{R}^{d_h}$

NTK Derivation

- **Goal:** Compute the limit of $\left\langle \frac{\partial f(\mathbf{w},\boldsymbol{x})}{\partial \mathbf{w}}, \frac{\partial f(\mathbf{w},\boldsymbol{x}')}{\partial \mathbf{w}} \right\rangle$ at random initialization in the infinite width limit.
- Recall: $\boldsymbol{f}^{(h)}(\boldsymbol{x}) = \boldsymbol{W}^{(h)}\boldsymbol{g}^{(h-1)}(\boldsymbol{x}) \in \mathbb{R}^{d_h}$
- Partial derivative w.r.t. particular weight $\boldsymbol{W}^{(h)}$:

$$\frac{\partial f(\mathbf{w},\boldsymbol{x})}{\partial \boldsymbol{W}^{(h)}} = \mathbf{b}^{(h)}(\boldsymbol{x}) \cdot \left(\boldsymbol{g}^{(h-1)}(\boldsymbol{x})\right)^{\top}, \qquad h = 1, 2, \ldots, L+1,$$

where

$$\mathbf{b}^{(h)}(\boldsymbol{x}) = \begin{cases} 1 \in \mathbb{R}, & h = L+1, \\ \sqrt{\frac{c_\sigma}{d_h}}\boldsymbol{D}^{(h)}(\boldsymbol{x})\left(\boldsymbol{W}^{(h+1)}\right)^{\top}\mathbf{b}^{(h+1)}(\boldsymbol{x}) \in \mathbb{R}^{d_h}, & h = 1, \ldots, L, \end{cases}$$

$$\boldsymbol{D}^{(h)}(\boldsymbol{x}) = \mathrm{diag}\left(\dot{\sigma}\left(\boldsymbol{f}^{(h)}(\boldsymbol{x})\right)\right) \in \mathbb{R}^{d_h \times d_h}, \qquad h = 1, \ldots, L.$$

NTK Derivation

Then, for any $h \in [L+1]$, we can compute

$$\left\langle \frac{\partial f(\mathbf{w}, \boldsymbol{x})}{\partial \boldsymbol{W}^{(h)}}, \frac{\partial f(\mathbf{w}, \boldsymbol{x}')}{\partial \boldsymbol{W}^{(h)}} \right\rangle = \Big\langle \mathbf{b}^{(h)}(\boldsymbol{x}) \cdot \left(\boldsymbol{g}^{(h-1)}(\boldsymbol{x})\right)^{\top}, \mathbf{b}^{(h)}(\boldsymbol{x}') \cdot \Big(\boldsymbol{g}^{(h-1)}(\boldsymbol{x}'$$

$$= \left\langle \boldsymbol{g}^{(h-1)}(\boldsymbol{x}), \boldsymbol{g}^{(h-1)}(\boldsymbol{x}') \right\rangle \cdot \left\langle \mathbf{b}^{(h)}(\boldsymbol{x}), \mathbf{b}^{(h)}(\boldsymbol{x}') \right\rangle .$$

- Using CLT,

$$\left\langle \boldsymbol{g}^{(h-1)}(\boldsymbol{x}), \boldsymbol{g}^{(h-1)}(\boldsymbol{x}') \right\rangle \to \Sigma^{(h-1)}\left(\boldsymbol{x}, \boldsymbol{x}'\right).$$

- Inductively, can show that

$$\left\langle \mathbf{b}^{(h)}(\boldsymbol{x}), \mathbf{b}^{(h)}(\boldsymbol{x}') \right\rangle \to \prod_{h'=h}^{L} \dot{\Sigma}^{(h')}(\boldsymbol{x}, \boldsymbol{x}').$$