Spherical Harmonics and Neural Tangent Kernel

- A bit an unusual presentation, more a **tutorial** on spherical harmonics in the context of NTK

- A bit an unusual presentation, more a **tutorial** on spherical harmonics in the context of NTK

- Based on several papers on NTK (and kernels in general):

- A bit an unusual presentation, more a **tutorial** on spherical harmonics in the context of NTK

- Based on several papers on NTK (and kernels in general):

- *On the inductive bias of neural tangent kernels* (Alberto Bietti and Julien Mairal)

- A bit an unusual presentation, more a **tutorial** on spherical harmonics in the context of NTK

- Based on several papers on NTK (and kernels in general):

- *On the inductive bias of neural tangent kernels* (Alberto Bietti and Julien Mairal)

- *The convergence rate of neural networks for learned functions of different frequencies* (Ronen Basri et al.)

- A bit an unusual presentation, more a **tutorial** on spherical harmonics in the context of NTK
- Based on several papers on NTK (and kernels in general):
- *On the inductive bias of neural tangent kernels* (Alberto Bietti and Julien Mairal)
- *The convergence rate of neural networks for learned functions of different frequencies* (Ronen Basri et al.)
- *Regularization with Dot-Product Kernels* (Williamson et al)

- A bit an unusual presentation, more a **tutorial** on spherical harmonics in the context of NTK
- Based on several papers on NTK (and kernels in general):
- *On the inductive bias of neural tangent kernels* (Alberto Bietti and Julien Mairal)
- *The convergence rate of neural networks for learned functions of different frequencies* (Ronen Basri et al.)
- *Regularization with Dot-Product Kernels* (Williamson et al)
- The goal is to understand the eigenspectrum of the NTK if the data is supported **uniformly on the sphere**

Setup

Setup

- Assume we have a **regression** problem with a true underlying function:
$$y = f^*(\boldsymbol{x})$$

Setup

- Assume we have a **regression** problem with a true underlying function:
$$y = f^*(\boldsymbol{x})$$
- **Input** data is generated uniformly on the unit sphere $\mathbb{S}^{d-1}$:
$$p(\boldsymbol{x}) \sim \mathcal{U}(\mathbb{S}^{d-1})$$

Setup

- Assume we have a **regression** problem with a true underlying function:
$$y = f^*(\boldsymbol{x})$$
- **Input** data is generated uniformly on the unit sphere $\mathbb{S}^{d-1}$:
$$p(\boldsymbol{x}) \sim \mathcal{U}(\mathbb{S}^{d-1})$$
- More interesting setting of **adversarial spheres** is very related

Setup

- Assume we have a **regression** problem with a true underlying function:

$$y = f^*(\boldsymbol{x})$$

- **Input** data is generated uniformly on the unit sphere $\mathbb{S}^{d-1}$:

$$p(\boldsymbol{x}) \sim \mathcal{U}(\mathbb{S}^{d-1})$$

- More interesting setting of **adversarial spheres** is very related

- **Notice**: No assumption on the target distribution made

Mercer Decomposition (I)

- **Recall** the Helmholz operator T_K associated to the kernel K:

$$(T_K\phi)(\boldsymbol{x}) = \int_{\mathcal{X}} K(\boldsymbol{x}, \boldsymbol{y})\phi(\boldsymbol{y})p(\boldsymbol{y})d\boldsymbol{y}$$

  where $\mathcal{X}$ denotes the support of the input data distribution p.

- We are interested in the **eigenfunctions** of this operator:

$$(T_K\phi)(\boldsymbol{x}) = \lambda\phi(\boldsymbol{x})\ \forall \boldsymbol{x} \in \mathcal{X}$$

- Why? Because we get the decomposition

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=0}^{\infty} \lambda_i\phi_i(\boldsymbol{x})\phi_i(\boldsymbol{y})$$

Mercer Decomposition (II)

- Usually p is of course **unknown** for real datasets
- Moreover, performing the integral is usually **hard** even for known p
- Here we choose $p(\boldsymbol{x}) \sim \mathcal{U}(\mathbb{S}^{d-1})$ to make life easy
- Moreover, NTK (and NNGP) is a **dot-product kernel**:

$$\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\Theta}(\boldsymbol{x}^T \boldsymbol{y})$$

- Turns out that the **spherical harmonics** play a crucial role

Spherical Harmonics

Spherical Harmonics

- Used a lot in many different fields (PDEs, Quantum Mechanics etc)

Spherical Harmonics

- Used a lot in many different fields (PDEs, Quantum Mechanics etc)
- Lots of **different** conventions regarding constants...

Spherical Harmonics

- Used a lot in many different fields (PDEs, Quantum Mechanics etc)
- Lots of **different** conventions regarding constants...
- Higher-dimensional **analog** to Fourier basis (more on that later)

Spherical Harmonics

- Used a lot in many different fields (PDEs, Quantum Mechanics etc)
- Lots of **different** conventions regarding constants...
- Higher-dimensional **analog** to Fourier basis (more on that later)
- Key ingredient in finding the **eigenfunctions** of the Helmholz operator associated with the NTK

Spherical Harmonics

- Used a lot in many different fields (PDEs, Quantum Mechanics etc)
- Lots of **different** conventions regarding constants...
- Higher-dimensional **analog** to Fourier basis (more on that later)
- Key ingredient in finding the **eigenfunctions** of the Helmholz operator associated with the NTK
- We will mostly look at the **three dimensional** case but extensions are analogous

Everything starts with the Laplacian

Everything starts with the Laplacian

- **Recall:** $\Delta f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$

Everything starts with the Laplacian

- **Recall:** $\Delta f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$
- Laplace equation: $\Delta f = 0$

Everything starts with the Laplacian

- **Recall:** $\Delta f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$
- Laplace equation: $\Delta f = 0$
- Focus first on the **two-dimensional** case:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

Everything starts with the Laplacian

- **Recall:** $\Delta f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$

- Laplace equation: $\Delta f = 0$

- Focus first on the **two-dimensional** case:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

- Rewrite it in **polar** coordinates $x = r\cos(\theta), y = r\sin(\theta)$:

$$\Delta f = \frac{\partial^2 f}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r}\frac{\partial f}{\partial r}$$

Solving the 2d Laplacian

Solving the 2d Laplacian

- $\frac{\partial^2 f}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r}\frac{\partial f}{\partial r} = 0$

Solving the 2d Laplacian

- $\frac{\partial^2 f}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r}\frac{\partial f}{\partial r} = 0$

- Method of **separation of variables**: $f(\theta, r) = R(r)\Theta(\theta)$

Solving the 2d Laplacian

- $\frac{\partial^2 f}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r}\frac{\partial f}{\partial r} = 0$
- Method of **separation of variables**: $f(\theta, r) = R(r)\Theta(\theta)$
- Leads to two ordinary differential equations:
  - $-r^2\ddot{R}(r) - r\dot{R}(r) = \lambda$
  - $\Theta''(\theta) = \lambda\Theta(\theta)$

Solving the 2d Laplacian

- $\frac{\partial^2 f}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r}\frac{\partial f}{\partial r} = 0$
- Method of **separation of variables**: $f(\theta, r) = R(r)\Theta(\theta)$
- Leads to two ordinary differential equations:
  - $-r^2\ddot{R}(r) - r\dot{R}(r) = \lambda$
  - $\Theta''(\theta) = \lambda\Theta(\theta)$
- The solution for the **angular part**, due to required periodicity (forcing $\lambda = -k^2$) is given by

$$\Theta_k(\theta) = C_k e^{ik\theta} \ \forall k \in \mathbb{Z}$$

Solving the 2d Laplacian

- $\frac{\partial^2 f}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r}\frac{\partial f}{\partial r} = 0$
- Method of **separation of variables**: $f(\theta, r) = R(r)\Theta(\theta)$
- Leads to two ordinary differential equations:
    - $-r^2\ddot{R}(r) - r\dot{R}(r) = \lambda$
    - $\Theta''(\theta) = \lambda\Theta(\theta)$
- The solution for the **angular part**, due to required periodicity (forcing $\lambda = -k^2$) is given by

$$\Theta_k(\theta) = C_k e^{ik\theta} \; \forall k \in \mathbb{Z}$$

- This is exactly the **Fourier basis**!

3d Laplacian

3d Laplacian

- How to generalize the Fourier basis to **higher dimensions**?

3d Laplacian

- How to generalize the Fourier basis to **higher dimensions**?
- Let's solve for the **angular part** of the higher dimensional Laplace equation!

3d Laplacian

- How to generalize the Fourier basis to **higher dimensions**?
- Let's solve for the **angular part** of the higher dimensional Laplace equation!
- **First step:** Translate Laplacian to **spherical** coordinates:

$$\Delta f = \frac{\partial^2 f}{\partial r^2} + \frac{2}{r}\frac{\partial f}{\partial r} + \frac{1}{r^2 \sin(\theta)}\left(\cos(\theta)\frac{\partial f}{\partial \theta} + \sin(\theta)\frac{\partial^2 f}{\partial \theta^2}\right) + \frac{1}{r^2 \sin^2(\theta)}\frac{\partial^2 f}{\partial \phi^2}$$

3d Laplacian

- How to generalize the Fourier basis to **higher dimensions**?
- Let's solve for the **angular part** of the higher dimensional Laplace equation!
- **First step:** Translate Laplacian to **spherical** coordinates:

$$\Delta f = \frac{\partial^2 f}{\partial r^2} + \frac{2}{r}\frac{\partial f}{\partial r} + \frac{1}{r^2 \sin(\theta)}\left(\cos(\theta)\frac{\partial f}{\partial \theta} + \sin(\theta)\frac{\partial^2 f}{\partial \theta^2}\right) + \frac{1}{r^2 \sin^2(\theta)}\frac{\partial^2 f}{\partial \phi^2}$$

- Again a **separation ansatz**: $f(r, \theta, \phi) = R(r)Y(\theta, \phi)$

Radial Part of Laplacian

Radial Part of Laplacian

- We get the following ODE for the radius:

$$\left(\ddot{R}(r) + \frac{2}{r}\dot{R}(r)\right)\frac{r^2}{R(r)} = \lambda \iff r^2\ddot{R}(r) + 2r\dot{R}(r) - \lambda R(r) = 0$$

Radial Part of Laplacian

- We get the following ODE for the radius:

$$\left(\ddot{R}(r) + \frac{2}{r}\dot{R}(r)\right)\frac{r^2}{R(r)} = \lambda \iff r^2\ddot{R}(r) + 2r\dot{R}(r) - \lambda R(r) = 0$$

- Second order **Cauchy-Euler** differential equation

Radial Part of Laplacian

- We get the following ODE for the radius:

$$\left(\ddot{R}(r) + \frac{2}{r}\dot{R}(r)\right)\frac{r^2}{R(r)} = \lambda \iff r^2\ddot{R}(r) + 2r\dot{R}(r) - \lambda R(r) = 0$$

- Second order **Cauchy-Euler** differential equation

- The solutions are given by $R_l(r) = r^l$ where l satisfies:

$$l(l-1) + 2l - \lambda = 0 \iff \lambda = l(l+1)$$

Angular Part of Laplacian

Angular Part of Laplacian

- We get the partial differential equation for the **angle**:

$$\cos(\theta)\,Y'(\theta,\phi)+\sin(\theta)\,Y''(\theta,\phi)+\frac{1}{r^2\sin^2(\theta)}\,Y^{**}(\theta,\phi)=-\lambda\sin(\theta)\,Y(\theta,\phi)$$

where $f^{'}(\theta,\phi)=\frac{\partial f}{\partial\theta}$ and $f^{*}(\theta,\phi)=\frac{\partial f}{\partial\phi}$

Angular Part of Laplacian

- We get the partial differential equation for the **angle**:

$$\cos(\theta)Y'(\theta,\phi)+\sin(\theta)Y''(\theta,\phi)+\frac{1}{r^2\sin^2(\theta)}Y^{**}(\theta,\phi)=-\lambda\sin(\theta)Y(\theta,\phi)$$

where $f^{'}(\theta,\phi)=\frac{\partial f}{\partial\theta}$ and $f^{*}(\theta,\phi)=\frac{\partial f}{\partial\phi}$

- Again make the **separation ansatz**: $Y(\theta,\phi)=\Theta(\theta)\Phi(\phi)$ splitting into the polar and azimuthal angles.

Azimuthal Angle Part of Laplacian

Azimuthal Angle Part of Laplacian

- This leads to the following equation for Φ:

$$-\frac{\Phi^{**}(\phi)}{\Phi(\phi)} = \omega$$

Azimuthal Angle Part of Laplacian

- This leads to the following equation for Φ:

$$-\frac{\Phi^{**}(\phi)}{\Phi(\phi)} = \omega$$

- We directly get the solution (again due periodicity, $\omega = m^2$)

$$\Phi(\phi) = A_m e^{i\phi m} \; \forall m \in \mathbb{Z}$$

Polar Angle Part of Laplacian

Polar Angle Part of Laplacian

- For the **polar angle** we have

$$\frac{\sin(\theta)}{\Theta(\theta)}\left(\cos(\theta)\Theta'(\theta)+\sin(\theta)\Theta''(\theta)+\lambda\Theta(\theta)\sin(\theta)\right)=\omega$$

Polar Angle Part of Laplacian

- For the **polar angle** we have

$$\frac{\sin(\theta)}{\Theta(\theta)}\left(\cos(\theta)\Theta'(\theta)+\sin(\theta)\Theta''(\theta)+\lambda\Theta(\theta)\sin(\theta)\right)=\omega$$

- Pugging in $w=m^2$, $\lambda=l(l+1)$ and can reformulating:

$$\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin(\theta)\Theta'(\theta)\right)+\left(l(l+1)-\frac{m^2}{\sin^2(\theta)}\right)=0$$

Polar Angle Part of Laplacian

- For the **polar angle** we have

$$\frac{\sin(\theta)}{\Theta(\theta)}\left(\cos(\theta)\Theta'(\theta)+\sin(\theta)\Theta''(\theta)+\lambda\Theta(\theta)\sin(\theta)\right)=\omega$$

- Pugging in $w = m^2$, $\lambda = l(l+1)$ and can reformulating:

$$\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\Big(\sin(\theta)\Theta'(\theta)\Big)+\left(l(l+1)-\frac{m^2}{\sin^2(\theta)}\right)=0$$

- Making a **change of variable** $x = \cos(\theta)$ leads to

$$\frac{\partial}{\partial x}\left((1-x^2)\frac{\partial\Theta(x)}{\partial x}\right)+\left(l(l+1)-\frac{m^2}{1-x^2}\right)\Theta(x)=0$$

Polar Angle Part of Laplacian

- For the **polar angle** we have

$$\frac{\sin(\theta)}{\Theta(\theta)}\left(\cos(\theta)\Theta'(\theta)+\sin(\theta)\Theta''(\theta)+\lambda\Theta(\theta)\sin(\theta)\right)=\omega$$

- Pugging in $w = m^2$, $\lambda = l(l+1)$ and can reformulating:

$$\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\Big(\sin(\theta)\Theta'(\theta)\Big)+\left(l(l+1)-\frac{m^2}{\sin^2(\theta)}\right)=0$$

- Making a **change of variable** $x = \cos(\theta)$ leads to

$$\frac{\partial}{\partial x}\left((1-x^2)\frac{\partial\Theta(x)}{\partial x}\right)+\left(l(l+1)-\frac{m^2}{1-x^2}\right)\Theta(x)=0$$

- This is the **general Legendre equation**

Associated Legendre Polynomials

Associated Legendre Polynomials

- The solution to this equation are **associated Legendre polynomials** P_l^m (not a polynomial for odd m though)

Associated Legendre Polynomials

- The solution to this equation are **associated Legendre polynomials** P_l^m (not a polynomial for odd m though)

- They are defined via:

$$P_l^m(x) = \frac{(-1)^l}{2^l l!}(1-x^2)^{\frac{m}{2}} \frac{d^{m+l}}{dx^{m+l}}(x^2-1)^l$$

Associated Legendre Polynomials

- The solution to this equation are **associated Legendre polynomials** P_l^m (not a polynomial for odd m though)

- They are defined via:

$$P_l^m(x) = \frac{(-1)^l}{2^l l!}(1-x^2)^{\frac{m}{2}}\frac{d^{m+l}}{dx^{m+l}}(x^2-1)^l$$

- They form a **basis** of continuous functions mapping from $[-1,1]$ to $\mathbb{R}$ and are orthogonal:

$$\int_{-1}^{1} P_k^m(x)P_l^m(x)dx = \frac{2(l+m)!}{(2l+1)!(l-m)!}\delta_{k,l}$$

$$\int_{-1}^{1} P_l^m(x)P_l^n(x)\frac{1}{1-x^2}dx = \frac{(l+m)!}{m(l-m)!}\delta_{m,n}$$

Spherical Harmonics

Spherical Harmonics

- For every $l \in \mathbb{N}$ and $-l \leq m \leq l$ the angular part is called **spherical harmonic** and is given by

$$Y^l_m(\theta, \phi) = Ce^{im\phi} P^m_l(\cos(\theta))$$

Spherical Harmonics

- For every $l \in \mathbb{N}$ and $-l \leq m \leq l$ the angular part is called **spherical harmonic** and is given by

$$Y_m^l(\theta, \phi) = Ce^{im\phi} P_l^m(\cos(\theta))$$

- They form an **orthonormal** basis of $\{f : \mathbb{S}^2 \to \mathbb{R}\}$:

$$\int_{\mathbb{S}^2} Y_m^l(\boldsymbol{x}) Y_{m'}^{l'}(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \delta_{m,m'} \delta_{l,l'}$$

Spherical Harmonics

- For every $l \in \mathbb{N}$ and $-l \leq m \leq l$ the angular part is called **spherical harmonic** and is given by

$$Y_m^l(\theta, \phi) = Ce^{im\phi} P_l^m(\cos(\theta))$$

- They form an **orthonormal** basis of $\{f : \mathbb{S}^2 \to \mathbb{R}\}$:

$$\int_{\mathbb{S}^2} Y_m^l(\boldsymbol{x}) Y_{m'}^{l'}(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \delta_{m,m'} \delta_{l,l'}$$

- Exact analog of Fourier basis which forms a basis for **periodic** functions, hence $\mathbb{S}^1$

Spherical Harmonics

- For every $l \in \mathbb{N}$ and $-l \leq m \leq l$ the angular part is called **spherical harmonic** and is given by

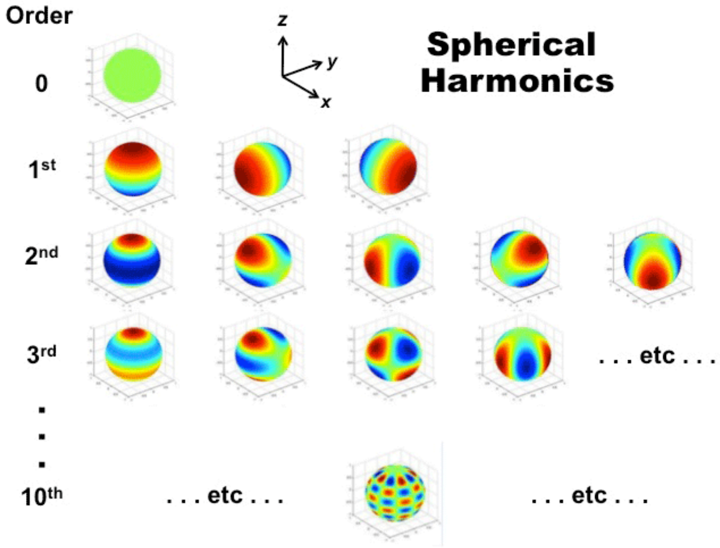
$$Y_m^l(\theta, \phi) = Ce^{im\phi} P_l^m(\cos(\theta))$$

- They form an **orthonormal** basis of $\{f : \mathbb{S}^2 \to \mathbb{R}\}$:

$$\int_{\mathbb{S}^2} Y_m^l(\boldsymbol{x}) Y_{m'}^{l'}(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \delta_{m,m'} \delta_{l,l'}$$

- Exact analog of Fourier basis which forms a basis for **periodic** functions, hence $\mathbb{S}^1$

- Both stem from solving the **Laplacian equation**


Order
0
1st
2nd
3rd
10th
z
y
x
Spherical
Harmonics
. . . etc . . .
. . . etc . . .
. . . etc . . .

Back to NTK

Back to NTK

- **Observe:** $\Theta(\boldsymbol{x}, \boldsymbol{y})$ is just a function of $\boldsymbol{x}^T \boldsymbol{y}$

Back to NTK

- **Observe:** $\Theta(\boldsymbol{x}, \boldsymbol{y})$ is just a function of $\boldsymbol{x}^T \boldsymbol{y}$
- Since we're on the sphere: $-1 \leq \boldsymbol{x}^T \boldsymbol{y} \leq 1$

Back to NTK

- **Observe:** $\Theta(\boldsymbol{x}, \boldsymbol{y})$ is just a function of $\boldsymbol{x}^T \boldsymbol{y}$

- Since we're on the sphere: $-1 \leq \boldsymbol{x}^T \boldsymbol{y} \leq 1$

- Now we can leverage a theorem by **Funk-Hecke**: For any continuous f defined on $[-1, 1]$ we have that

$$\int_{\mathbb{S}^2} f(\boldsymbol{x}^T \boldsymbol{y}) Y_m^l(\boldsymbol{y}) d\boldsymbol{y} = \left(\mathrm{vol}(\mathbb{S}^1) \int_{-1}^{1} f(t) G_k(t) dt\right) Y_m^l(\boldsymbol{x})$$

where G_k is the **Gegenbauer polynomial**.

Back to NTK

- **Observe:** $\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{y})$ is just a function of $\boldsymbol{x}^T \boldsymbol{y}$

- Since we're on the sphere: $-1 \leq \boldsymbol{x}^T \boldsymbol{y} \leq 1$

- Now we can leverage a theorem by **Funk-Hecke**: For any continuous f defined on $[-1, 1]$ we have that

$$\int_{\mathbb{S}^2} f(\boldsymbol{x}^T \boldsymbol{y}) Y_m^l(\boldsymbol{y}) d\boldsymbol{y} = \left(\text{vol}(\mathbb{S}^1) \int_{-1}^{1} f(t) G_k(t) dt \right) Y_m^l(\boldsymbol{x})$$

where G_k is the **Gegenbauer polynomial**.

- We thus have the following **Mercer decomposition**:

$$\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{l=0}^{\infty} \lambda_l \sum_{m=1}^{N(p,k)} Y_m^l(\boldsymbol{x}) Y_m^l(\boldsymbol{y})$$

Eigenvalues λ_l

Eigenvalues λ_l

- Here **specific form** of kernel becomes important!

Eigenvalues λ_l

- Here **specific form** of kernel becomes important!

- In the **general network** case, not much is known about the exact values of λ_l

Eigenvalues λ_l

- Here **specific form** of kernel becomes important!
- In the **general network** case, not much is known about the exact values of λ_l
- One can prove that $\lambda_{2k+1} = 0$ for $k \geq 2$ and $\lambda_k \sim C(p)k^{-p}$ for **one hidden** layer networks.
  If only top layer trainable: $\lambda_k \sim B(p)k^{-(p+2)}$

Eigenvalues λ_l

- Here **specific form** of kernel becomes important!

- In the **general network** case, not much is known about the exact values of λ_l

- One can prove that $\lambda_{2k+1} = 0$ for $k \geq 2$ and $\lambda_k \sim C(p)k^{-p}$ for **one hidden** layer networks.
  If only top layer trainable: $\lambda_k \sim B(p)k^{-(p+2)}$

- If we only have one hidden layer with a **non-trainable** top layer and data on $\mathbb{S}^1$, then we find

$$\lambda_0 = \frac{1}{\pi^2} \text{ , } \lambda_1 = \frac{1}{4} \text{ and } \lambda_k = \frac{2(k^2+1)}{\pi^2(k^2-1)^2} \text{ for k even}$$

Eigenvalues λ_l

- Here **specific form** of kernel becomes important!

- In the **general network** case, not much is known about the exact values of λ_l

- One can prove that $\lambda_{2k+1} = 0$ for $k \geq 2$ and $\lambda_k \sim C(p)k^{-p}$ for **one hidden** layer networks.
  If only top layer trainable: $\lambda_k \sim B(p)k^{-(p+2)}$

- If we only have one hidden layer with a **non-trainable** top layer and data on $\mathbb{S}^1$, then we find

$$\lambda_0 = \frac{1}{\pi^2}\ ,\ \lambda_1 = \frac{1}{4} \text{ and } \lambda_k = \frac{2(k^2+1)}{\pi^2(k^2-1)^2} \text{ for k even}$$

- Similar but more complicated **analytical** formulas are available for $\mathbb{S}^{d-1}$

Corresponding RKHS

Corresponding RKHS

- Using the decomposition we can describe the functions lying in the RKHS induced by the NTK:

$$\mathcal{H} = \left\{ f = \sum_{l:\lambda_l \neq 0} \sum_{m=1}^{N(l,m)} a_{lm} Y_m^l(\cdot) \text{ s.t. } ||f||_{\mathcal{H}}^2 = \sum_{l:\lambda_l \neq 0} \sum_{m=1}^{N(l,m)} \frac{a_{lm}}{\lambda_l} < \infty \right\}$$

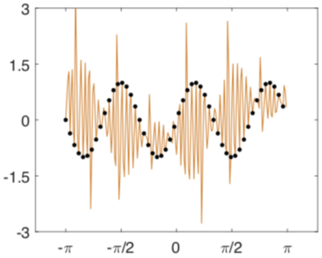
Corresponding RKHS

- Using the decomposition we can describe the functions lying in the RKHS induced by the NTK:

$$\mathcal{H} = \left\{ f = \sum_{l:\lambda_l \neq 0} \sum_{m=1}^{N(l,m)} a_{lm} Y_m^l(\cdot) \text{ s.t. } ||f||_{\mathcal{H}}^2 = \sum_{l:\lambda_l \neq 0} \sum_{m=1}^{N(l,m)} \frac{a_{lm}}{\lambda_l} < \infty \right\}$$

- This means that odd frequencies should be harder to learn for one hidden layer networks because $\lambda_{2k+1} = 0$.

Odd frequencies are hard to learn



Bias-free wide one hidden layer network learning $f(x) = \cos(3\theta)$ on left and $f(x) = \cos(4\theta)$ on right

One hidden layer case

One hidden layer case

- We use a **one hidden layer** network:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sqrt{\frac{2}{m}} \sum_{i=1}^{m} v_j \sigma \left(\boldsymbol{w}_j^T \boldsymbol{x} \right)$$

  where $\sigma(\cdot)$ denotes the ReLU activation

One hidden layer case

- We use a **one hidden layer** network:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sqrt{\frac{2}{m}} \sum_{i=1}^{m} v_j \sigma \left(\boldsymbol{w}_j^T \boldsymbol{x} \right)$$

  where $\sigma(\cdot)$ denotes the ReLU activation

- We can write the NTK $\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{x}')$ as:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}^T \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}' \kappa_0(\boldsymbol{x}^T \boldsymbol{x}') + \kappa_1(\boldsymbol{x}^T \boldsymbol{x}')$$

One hidden layer case

- We use a **one hidden layer** network:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sqrt{\frac{2}{m}} \sum_{i=1}^{m} v_j \sigma \left(\boldsymbol{w}_j^T \boldsymbol{x} \right)$$

  where $\sigma(\cdot)$ denotes the ReLU activation

- We can write the NTK $\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{x}')$ as:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}^T \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}' \kappa_0(\boldsymbol{x}^T \boldsymbol{x}') + \kappa_1(\boldsymbol{x}^T \boldsymbol{x}')$$

- $\kappa_0(u) = \frac{1}{\pi} (\pi - \arccos(u))$

One hidden layer case

- We use a **one hidden layer** network:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sqrt{\frac{2}{m}} \sum_{i=1}^{m} v_j \sigma \left(\boldsymbol{w}_j^T \boldsymbol{x} \right)$$

  where $\sigma(\cdot)$ denotes the ReLU activation

- We can write the NTK $\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{x}')$ as:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}^T \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}' \kappa_0(\boldsymbol{x}^T \boldsymbol{x}') + \kappa_1(\boldsymbol{x}^T \boldsymbol{x}')$$

- $\kappa_0(u) = \frac{1}{\pi} (\pi - \arccos(u))$

- $\kappa_1(u) = \frac{1}{\pi} \left(u(\pi - \arccos(u)) + \sqrt{1 - u^2} \right)$

Feature Representation

Feature Representation

- We can derive a **dot product** representation for $\boldsymbol{\Theta}$

Feature Representation

- We can derive a **dot product** representation for $\boldsymbol{\Theta}$
- Denote by ϕ_1 the feature representation of κ_1, namely $\kappa_1(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_1(\boldsymbol{z}), \phi_1(\boldsymbol{z}') \rangle$ and the same with ϕ_0 and κ_0, namely $\kappa_0(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_0(\boldsymbol{z}), \phi_0(\boldsymbol{z}') \rangle$

Feature Representation

- We can derive a **dot product** representation for $\boldsymbol{\Theta}$
- Denote by ϕ_1 the feature representation of κ_1, namely $\kappa_1(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_1(\boldsymbol{z}), \phi_1(\boldsymbol{z}') \rangle$ and the same with ϕ_0 and κ_0, namely $\kappa_0(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_0(\boldsymbol{z}), \phi_0(\boldsymbol{z}') \rangle$

Lemma 1:
We can write $\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi_n(\boldsymbol{x}), \Phi_n(\boldsymbol{x}') \rangle$ *where*

1. $\Phi_0(\boldsymbol{x}) = \Psi_0(\boldsymbol{x}) = \boldsymbol{x}$
2. $\Psi_k(\boldsymbol{x}) = \phi_1(\Phi_{k-1}(\boldsymbol{x}))$
3. $\Phi_k(\boldsymbol{x}) = \begin{pmatrix} \phi_0(\Psi_{k-1}(\boldsymbol{x})) \odot \Phi_{k-1}(\boldsymbol{x}) \\ \phi_1(\Psi_{k-1}(\boldsymbol{x})) \end{pmatrix}$

Feature Representation

- We can derive a **dot product** representation for $\boldsymbol{\Theta}$

- Denote by ϕ_1 the feature representation of κ_1, namely $\kappa_1(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_1(\boldsymbol{z}), \phi_1(\boldsymbol{z}') \rangle$ and the same with ϕ_0 and κ_0, namely $\kappa_0(\boldsymbol{z}, \boldsymbol{z}') = \langle \phi_0(\boldsymbol{z}), \phi_0(\boldsymbol{z}') \rangle$

Lemma 1:
We can write $\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi_n(\boldsymbol{x}), \Phi_n(\boldsymbol{x}') \rangle$ *where*

1. $\Phi_0(\boldsymbol{x}) = \Psi_0(\boldsymbol{x}) = \boldsymbol{x}$

2. $\Psi_k(\boldsymbol{x}) = \phi_1(\Phi_{k-1}(\boldsymbol{x}))$

3. $\Phi_k(\boldsymbol{x}) = \begin{pmatrix} \phi_0(\Psi_{k-1}(\boldsymbol{x})) \odot \Phi_{k-1}(\boldsymbol{x}) \\ \phi_1(\Psi_{k-1}(\boldsymbol{x})) \end{pmatrix}$

- This holds for **arbitrary** depth networks, proof via induction

Lipschitzness of Kernel

Lipschitzness of Kernel

- The feature representation $\Phi(\cdot)$ is **not** Lipschitz:

$$\sup_{\boldsymbol{x},\boldsymbol{y}} \frac{||\Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y})||_{\mathcal{H}}}{||\boldsymbol{x} - \boldsymbol{y}||} \to \infty$$

Lipschitzness of Kernel

- The feature representation $\Phi(\cdot)$ is **not** Lipschitz:

$$\sup_{\boldsymbol{x},\boldsymbol{y}} \frac{||\Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y})||_{\mathcal{H}}}{||\boldsymbol{x} - \boldsymbol{y}||} \to \infty$$

- If however, the weights in first layer are fixed, and we **only** train top layer (NNGP), then $\kappa = \kappa_1$ and the feature map is Lipschitz

Lipschitzness of Kernel

- The feature representation $\Phi(\cdot)$ is **not** Lipschitz:

$$\sup_{\boldsymbol{x},\boldsymbol{y}} \frac{||\Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y})||_{\mathcal{H}}}{||\boldsymbol{x} - \boldsymbol{y}||} \to \infty$$

- If however, the weights in first layer are fixed, and we **only** train top layer (NNGP), then $\kappa = \kappa_1$ and the feature map is Lipschitz

- We can get something similar to **Hölder-smoothness**:

$$||\Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y})||_{\mathcal{H}} \leq \sqrt{||\boldsymbol{x} - \boldsymbol{y}||} + 2||\boldsymbol{x} - \boldsymbol{y}||$$

Why train all layers?

- Take any $f : \mathbb{S}^{p-1} \to \mathbb{R}$ even function such that $f(\boldsymbol{x}) \leq \eta$ and $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \eta ||\boldsymbol{x} - \boldsymbol{y}||_2 \; \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^{d-1}$. Then there is $g \in \mathcal{H}$ with $||g||_{\mathcal{H}} \leq \delta$ such that

$$\sup_{\boldsymbol{x} \in \mathbb{S}^{p-1}} |f(\boldsymbol{x}) - g(\boldsymbol{x})| \leq C(p)\eta \left(\frac{\delta}{\eta}\right)^{-\frac{1}{0.5p-1}} \log\left(\frac{\delta}{\eta}\right)$$

- We can basically approximate any even Lipschitz function over $\mathbb{S}^{d-1}$ well.

- For training only top layer, we can weaker rate $\frac{\delta}{\eta}^{-\frac{1}{0.5p}}$

One Hidden Layer with First Layer Frozen

One Hidden Layer with First Layer Frozen

- Simpler NTK expression given by:

$$\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{2\pi} \boldsymbol{x}^T \boldsymbol{x}' (\pi - \arccos(\boldsymbol{x}^T \boldsymbol{x}'))$$

One Hidden Layer with First Layer Frozen

- Simpler NTK expression given by:

$$\boldsymbol{\Theta}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{2\pi} \boldsymbol{x}^T \boldsymbol{x}' (\pi - \text{arccos}(\boldsymbol{x}^T \boldsymbol{x}'))$$

- Again spherical harmonics **decompose** the kernel

One Hidden Layer with First Layer Frozen

- Simpler NTK expression given by:

$$\Theta(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{2\pi} \boldsymbol{x}^T \boldsymbol{x}' (\pi - \arccos(\boldsymbol{x}^T \boldsymbol{x}'))$$

- Again spherical harmonics **decompose** the kernel

- For $d = 1$, easy formulas for eigenvalues (as said before) with odd harmonics vanishing:

$$\lambda_0 = \frac{1}{\pi^2} \ , \ \lambda_1 = \frac{1}{4} \text{ and } \lambda_l = \frac{2(l^2+1)}{\pi^2 (k^2-1)^2} \text{ for l even}$$

Generalization Bound for Bandlimited Target

Generalization Bound for Bandlimited Target

- Assume the target function is given by some **bandlimited**, one-dimensional function:

$$f^*(x) = \sum_{k=0}^{\bar{k}} \alpha_k e^{2\pi i k x}$$

Generalization Bound for Bandlimited Target

- Assume the target function is given by some **bandlimited**, one-dimensional function:

$$f^*(x) = \sum_{k=0}^{\bar{k}} \alpha_k e^{2\pi i k x}$$

- **Recall:** Arora derived generalization bound for one hidden layer NNGP:

$$L_{\text{gen}} \leq \sqrt{\frac{2\boldsymbol{y}^T \boldsymbol{\Theta}^{-1} \boldsymbol{y}}{n}} + \mathcal{O}(\text{stuff})$$

Generalization Bound for Bandlimited Target

- Assume the target function is given by some **bandlimited**, one-dimensional function:

$$f^*(x) = \sum_{k=0}^{\bar{k}} \alpha_k e^{2\pi i k x}$$

- **Recall:** Arora derived generalization bound for one hidden layer NNGP:

$$L_{\text{gen}} \leq \sqrt{\frac{2\boldsymbol{y}^T \boldsymbol{\Theta}^{-1} \boldsymbol{y}}{n}} + \mathcal{O}(\text{stuff})$$

- Approximating the eigenvalues of $\boldsymbol{\Theta}$ with above formulas leads to

$$L_{\text{gen}} \approx \sqrt{\frac{2\pi \sum_{k=1}^{\bar{k}} \alpha_k^2 k^2}{n}}$$

Generalization Bound for Bandlimited Target

- Assume the target function is given by some **bandlimited**, one-dimensional function:

$$f^*(x) = \sum_{k=0}^{\bar{k}} \alpha_k e^{2\pi i k x}$$

- **Recall:** Arora derived generalization bound for one hidden layer NNGP:

$$L_{\text{gen}} \leq \sqrt{\frac{2\boldsymbol{y}^T \boldsymbol{\Theta}^{-1} \boldsymbol{y}}{n}} + \mathcal{O}(\text{stuff})$$

- Approximating the eigenvalues of $\boldsymbol{\Theta}$ with above formulas leads to

$$L_{\text{gen}} \approx \sqrt{\frac{2\pi \sum_{k=1}^{\bar{k}} \alpha_k^2 k^2}{n}}$$

- Hence bound increases **linearly** with frequency

Discussion

Discussion

-