Numerical Analysis 101 for Neural Networks

Hongkai Zhao
Duke University

Joint work with Shijung Zhang, Haomin Zhou and Yimin Zhong

Research partially supported by NSF DMS-2012860, DMS-2309551.

Least square approximation

General form: given a target function $f(x), x \in D \subset \mathbb{R}^d$, and *a chosen* parametrized representation $h(x;\alpha)$

$$\min_{\alpha} I(\alpha) = \|h(\cdot;\alpha) - f(\cdot)\|^2_{L^2(D)}$$

Least square approximation

General form: given a target function $f(x), x \in D \subset \mathbb{R}^d$, and *a chosen* parametrized representation $h(x; \alpha)$

$$\min_{\alpha} I(\alpha) = \|h(\cdot; \alpha) - f(\cdot)\|^2_{L^2(D)}$$

Basic numerical analysis questions of practice importance:

- the best accuracy one can achieve given a finite machine precision,
- stability with respect to perturbations,
- the computation cost to achieve a given accuracy.

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x),\ \boldsymbol{\alpha} = (a_1, \dots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \| \sum_{i=1}^n a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1\rangle_D, \dots, \langle f, \psi_n\rangle_D)^T$.

- Important mathematical question:

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x),\ \boldsymbol{\alpha} = (a_1, \ldots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \|\sum_{i=1}^n a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1 \rangle_D, \ldots, \langle f, \psi_n \rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \ldots, \psi_n\}$, $\mathrm{dist}(f, V)$.

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x)$, $\boldsymbol{\alpha} = (a_1, \dots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \| \sum_{i=1}^{n} a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1 \rangle_D, \dots, \langle f, \psi_n \rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \dots, \psi_n\}$, $\mathrm{dist}(f, V)$.
- Important numerical questions:

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\alpha) = \sum_{i=1}^n a_i\psi_i(x),\ \alpha = (a_1, \dots, a_n)^T$

$$\min_{\alpha} I(\alpha) = \|\sum_{i=1}^{n} a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \alpha^* = \mathrm{argmin}_{\alpha} I(\alpha) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1 \rangle_D, \dots, \langle f, \psi_n \rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \dots, \psi_n\}$, dist$(f, V)$.
- Important numerical questions: G,

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\alpha) = \sum_{i=1}^n a_i \psi_i(x),\ \alpha = (a_1, \ldots, a_n)^T$

$$\min_{\alpha} I(\alpha) = \| \sum_{i=1}^{n} a_i \psi_i(\cdot) - f(\cdot) \|_{L^2(D)}^2$$

$$\Rightarrow \alpha^* = \text{argmin}_{\alpha} I(\alpha) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1 \rangle_D, \ldots, \langle f, \psi_n \rangle_D)^T$.

- Important mathematical question: $V = \text{span}\{\psi_1, \ldots, \psi_n\}$, dist$(f, V)$.
- Important numerical questions: G, G,

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x),\ \boldsymbol{\alpha} = (a_1, \dots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \|\sum_{i=1}^{n} a_i\psi_i(\cdot) - f(\cdot)\|_{L^2(D)}^2$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle\psi_i, \psi_j\rangle_D$, $\mathbf{f} = (\langle f, \psi_1\rangle_D, \dots, \langle f, \psi_n\rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \dots, \psi_n\}$, $\mathrm{dist}(f, V)$.
- Important numerical questions: G, G, G!

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^{n} a_i\psi_i(x),\ \boldsymbol{\alpha} = (a_1, \ldots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \| \sum_{i=1}^{n} a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1 \rangle_D, \ldots, \langle f, \psi_n \rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \ldots, \psi_n\}$, $\mathrm{dist}(f, V)$.
- Important numerical questions: G, G, G!
  - *Choice of the basis is the key!*

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x),\ \boldsymbol{\alpha} = (a_1, \ldots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \| \sum_{i=1}^{n} a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1 \rangle_D, \ldots, \langle f, \psi_n \rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \ldots, \psi_n\}$, $\mathrm{dist}(f, V)$.
- Important numerical questions: G, G, G!
  - *Choice of the basis is the key!*
  - spectral property of G

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x)$, $\boldsymbol{\alpha} = (a_1, \dots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \| \sum_{i=1}^{n} a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1\rangle_D, \dots, \langle f, \psi_n\rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \dots, \psi_n\}$, $\mathrm{dist}(f, V)$.
- Important numerical questions: G, G, G!
  - *Choice of the basis is the key!*
  - spectral property of G
  - sparsity of G

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x),\ \boldsymbol{\alpha} = (a_1, \ldots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \| \sum_{i=1}^n a_i\psi_i(\cdot) - f(\cdot)\|_{L^2(D)}^2$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle\psi_i, \psi_j\rangle_D$, $\mathbf{f} = (\langle f, \psi_1\rangle_D, \ldots, \langle f, \psi_n\rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \ldots, \psi_n\}$, $\mathrm{dist}(f, V)$.
- Important numerical questions: G, G, G!
  - *Choice of the basis is the key!*
  - spectral property of G
  - sparsity of G
  - computation cost of G, G^{-1}.

Least square approximation

Linear representation: given a target function $f(x), x \in D \subset \mathbb{R}^d$, choose a set of basis functions $\psi_i(x)$, $h(x;\boldsymbol{\alpha}) = \sum_{i=1}^n a_i\psi_i(x)$, $\boldsymbol{\alpha} = (a_1, \ldots, a_n)^T$

$$\min_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = \| \sum_{i=1}^{n} a_i\psi_i(\cdot) - f(\cdot)\|^2_{L^2(D)}$$

$$\Rightarrow \boldsymbol{\alpha}^* = \mathrm{argmin}_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = G^{-1}\mathbf{f},$$

G is Gram matrix, $G(i,j) = \langle \psi_i, \psi_j \rangle_D$, $\mathbf{f} = (\langle f, \psi_1 \rangle_D, \ldots, \langle f, \psi_n \rangle_D)^T$.

- Important mathematical question: $V = \mathrm{span}\{\psi_1, \ldots, \psi_n\}$, $\mathrm{dist}(f, V)$.
- Important numerical questions: G, G, G!
  - *Choice of the basis is the key!*
  - spectral property of G
  - sparsity of G
  - computation cost of G, G^{-1}.

For nonlinear least square problem: a non-convex optimization has to be taken into account!

Set up of neural network (NN)

Two layer NN with `reLU` activation function $\sigma(t) = \max(0, t)$:

$$h(x) = \sum_{i=1}^{n} a_i \sigma(w_i \cdot x - b_i), \quad x \in \mathbb{R}^d.$$

We study

- linear least square approximation when biases b_i are fixed,
- learning dynamics,
- Rashomon set for the parameters,

Set up of neural network (NN)

Two layer NN with `reLU` activation function $\sigma(t) = \max(0, t)$:

$$h(x) = \sum_{i=1}^{n} a_i \sigma(w_i \cdot x - b_i), \quad x \in \mathbb{R}^d.$$

We study

- linear least square approximation when biases b_i are fixed,
- learning dynamics,
- Rashomon set for the parameters,

to show

- why a two layer NN is essentially a ”low pass filter”

Set up of neural network (NN)

Two layer NN with `reLU` activation function $\sigma(t) = \max(0, t)$:

$$h(x) = \sum_{i=1}^{n} a_i \sigma(w_i \cdot x - b_i), \quad x \in \mathbb{R}^d.$$

We study

- linear least square approximation when biases b_i are fixed,
- learning dynamics,
- Rashomon set for the parameters,

to show

- why a two layer NN is essentially a "low pass filter" $\Rightarrow$ stable w.r.t. noise and over-parametrization and but may compromise for accuracy,

Set up of neural network (NN)

Two layer NN with `reLU` activation function $\sigma(t) = \max(0, t)$:

$$h(x) = \sum_{i=1}^{n} a_i \sigma(w_i \cdot x - b_i), \quad x \in \mathbb{R}^d.$$

We study

- linear least square approximation when biases b_i are fixed,
- learning dynamics,
- Rashomon set for the parameters,

to show

- why a two layer NN is essentially a "low pass filter" $\Rightarrow$ stable w.r.t. noise and over-parametrization and but may compromise for accuracy,
- the computation cost for training,

Set up of neural network (NN)

Two layer NN with `reLU` activation function $\sigma(t) = \max(0, t)$:

$$h(x) = \sum_{i=1}^{n} a_i \sigma(w_i \cdot x - b_i), \quad x \in \mathbb{R}^d.$$

We study

- linear least square approximation when biases b_i are fixed,
- learning dynamics,
- Rashomon set for the parameters,

to show

- why a two layer NN is essentially a "low pass filter" $\Rightarrow$ stable w.r.t. noise and over-parametrization and but may compromise for accuracy,
- the computation cost for training,
- what difference activation functions make,

Set up of neural network (NN)

Two layer NN with `reLU` activation function $\sigma(t) = \max(0, t)$:

$$h(x) = \sum_{i=1}^{n} a_i \sigma(w_i \cdot x - b_i), \quad x \in \mathbb{R}^d.$$

We study

- linear least square approximation when biases b_i are fixed,
- learning dynamics,
- Rashomon set for the parameters,

to show

- why a two layer NN is essentially a "low pass filter" $\Rightarrow$ stable w.r.t. noise and over-parametrization and but may compromise for accuracy,
- the computation cost for training,
- what difference activation functions make,
- why highly oscillatory functions are difficult to approximate by NN,

Set up of neural network (NN)

Two layer NN with `reLU` activation function $\sigma(t) = \max(0, t)$:

$$h(x) = \sum_{i=1}^{n} a_i \sigma(w_i \cdot x - b_i), \quad x \in \mathbb{R}^d.$$

We study

- linear least square approximation when biases b_i are fixed,
- learning dynamics,
- Rashomon set for the parameters,

to show

- why a two layer NN is essentially a "low pass filter" $\Rightarrow$ stable w.r.t. noise and over-parametrization and but may compromise for accuracy,
- the computation cost for training,
- what difference activation functions make,
- why highly oscillatory functions are difficult to approximate by NN,
- some further thoughts.

Two layer NN in 1D

$$h(x) = \sum_{i=1}^{n} a_i \sigma(x - b_i), \quad x, b_i \in D = (-1, 1),\ a_i \in \mathbb{R}$$

In theory, span$\{\sigma(x-b_1), \ldots, \sigma(x-b_n)\}$=span$\{P_1$ finite element basis$\}$.

Two layer NN in 1D

$$h(x) = \sum_{i=1}^{n} a_i \sigma(x - b_i), \quad x, b_i \in D = (-1, 1),\ a_i \in \mathbb{R}$$

In theory, span$\{\sigma(x-b_1), \ldots, \sigma(x-b_n)\}$=span$\{P_1$ finite element basis$\}$.
Numerically, the two sets of basis are very different!

Two layer NN in 1D

$$h(x) = \sum_{i=1}^{n} a_i \sigma(x - b_i), \quad x, b_i \in D = (-1, 1),\ a_i \in \mathbb{R}$$

In theory, span$\{\sigma(x-b_1), \dots, \sigma(x-b_n)\}$=span$\{P_1$ finite element basis$\}$.
Numerically, the two sets of basis are very different!

- Finite element basis is local and almost orthogonal $\Rightarrow$ the Gram matrix is sparse and the condition number is $O(1)$, ideal for least square approximation in lower dimension.

Two layer NN in 1D

$$h(x) = \sum_{i=1}^{n} a_i \sigma(x - b_i), \quad x, b_i \in D = (-1, 1), \ a_i \in \mathbb{R}$$

In theory, span$\{\sigma(x-b_1), \ldots, \sigma(x-b_n)\}$=span$\{P_1$ finite element basis$\}$. Numerically, the two sets of basis are very different!

- Finite element basis is local and almost orthogonal $\Rightarrow$ the Gram matrix is sparse and the condition number is $O(1)$, ideal for least square approximation in lower dimension.
- ReLU basis is global and can be highly correlated $\Rightarrow$ the Gram matrix is dense and has a fast spectral decay rate (ill-conditioned) $\Rightarrow$ only a certain number of leading eigen-modes are used in numerical computation $\Rightarrow$ low pass filter.

Spectral analysis for the Gram matrix of `ReLU` basis: 1D

Let $G := (G_{ij}) \in \mathbb{R}^{n\times n}$ be the Gram matrix

$$G_{ij} = \int_D \sigma(x-b_i)\sigma(x-b_j)dx = \frac{1}{12}|b_i-b_j|^3 + \frac{1}{12}(2-b_i-b_j)\left(2(1-b_i)(1-b_j)-(b_i-b_j)^2\right)$$

The corresponding continuous kernel

$$\mathcal{G}(x,y) = \int_D \sigma(z-x)\sigma(z-y)dz = \frac{1}{12}|x-y|^3 + \frac{1}{12}(2-x-y)\left(2(1-x)(1-y)-(x-y)^2\right)$$

Lemma

The eigenvalues in descending order are: $\mu_k = O(k^{-4})$. The corresponding eigenfunctions $\phi_k(x)$ satisfies

$$\phi_k^{(4)}(x) = \mu_k^{-1}\phi_k(x),\ x \in (-1,1), \quad \phi_k(1) = \phi_k^{(1)}(1) = \phi_k^{(2)}(-1) = \phi_k^{(3)}(-1) = 0$$

The first few leading eigenfunctions are a combination of exponential functions and Fourier modes, then followed by essentially Fourier modes, from low to high frequencies.

Spectral analysis for the Gram matrix of ReLU basis: 1D

Theorem

Suppose $\{b_i\}_{i=1}^n$ *are quasi-evenly spaced on* D, $b_i = -1 + \frac{2(i-1)}{n} + o\left(\frac{1}{n}\right)$. *Let* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ *be the eigenvalues of the Gram matrix* G *then* $|\lambda_k - \frac{n}{2}\mu_k| \leq C$ *for some constant* $C = O(1)$.

Spectral analysis for the Gram matrix of `ReLU` basis: 1D

Theorem

Suppose $\{b_i\}_{i=1}^n$ *are quasi-evenly spaced on* D, $b_i = -1 + \frac{2(i-1)}{n} + o\left(\frac{1}{n}\right)$. *Let* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ *be the eigenvalues of the Gram matrix* G *then* $|\lambda_k - \frac{n}{2}\mu_k| \leq C$ *for some constant* $C = O(1)$.

Corollary

Suppose b_i *are i.i.d distributed with probability density function* ρ *on* D *such that* $0 < \underline{c} \leq \rho(x) \leq \bar{c} < \infty$. $|\lambda_k - \frac{n}{2}\mu_k| \leq \frac{Cn}{k^4}\sqrt{\frac{k}{n}} \log p^{-1}$ *with probability* $1 - p$.

Spectral analysis for 1D evenly spaced biases was done in Hong et al., 2022.

Spectral analysis for the Gram matrix basis

`ReLU` basis in $\mathbb{R}^d$: $\sigma(w \cdot x - b)$, $w \in \mathbb{S}^{d-1}$, $b \in \mathbb{R}$. Use

$$\partial_b^2 \sigma(w \cdot x - b) = \Delta_x \sigma(w \cdot x - b) = \delta(w \cdot x - b),$$

and Radon transform inversion formula

Theorem

Let λ_k be the eigenvalue of the kernel $\mathcal{G}$

$$\mathcal{G}((w,b),(w',b')) = \int_D \sigma(w \cdot x - b)\sigma(w' \cdot x - b')dx.$$

There are constants $c_1, c_2 > 0$, depending on D and d, such that

$$c_1 k^{-(d+3)/d} \le \lambda_k \le c_2 k^{-(d+3)/d}.$$

- For σ^k, $\lambda_k = O(k^{-(d+k+2)/d})$.
- For analytic activation function such as `Tanh` or `Sigmoid`, the eigenvalues decays faster than any polynomial rate.

Implications to numerical computation in practice

- Low pass filter: given the machine precision ϵ, the eigenvalue threshold is $n\epsilon\lambda_1$. A two-layer neural network can use about $\epsilon^{-d/(d+3)}$ eigenmodes in d-dimensions or at most all Fourier modes up to frequency $k_d = \epsilon^{-1/(d+3)}$ can be resolved.

Implications to numerical computation in practice

- Low pass filter: given the machine precision ϵ, the eigenvalue threshold is $n\epsilon\lambda_1$. A two-layer neural network can use about $\epsilon^{-d/(d+3)}$ eigenmodes in d-dimensions or at most all Fourier modes up to frequency $k_d = \epsilon^{-1/(d+3)}$ can be resolved.
  single precision $\epsilon = 2^{-23}$: $k_1 \simeq 54, k_2 \simeq 25$, $k_3 \simeq 14$, $k_{10} = 4$.
  double precision $\epsilon = 2^{-52}$: $k_1 \simeq 8192, k_2 \simeq 1351$, $k_3 \simeq 411$, $k_{10} = 16$.

Implications to numerical computation in practice

- Low pass filter: given the machine precision ϵ, the eigenvalue threshold is $n\epsilon\lambda_1$. A two-layer neural network can use about $\epsilon^{-d/(d+3)}$ eigenmodes in d-dimensions or at most all Fourier modes up to frequency $k_d = \epsilon^{-1/(d+3)}$ can be resolved.
  single precision $\epsilon = 2^{-23}$: $k_1 \simeq 54, k_2 \simeq 25$, $k_3 \simeq 14$, $k_{10} = 4$.
  double precision $\epsilon = 2^{-52}$: $k_1 \simeq 8192, k_2 \simeq 1351$, $k_3 \simeq 411$, $k_{10} = 16$.
- Although two layer neural networks have universal approximation property when the width is increased, in practice, increasing the width does not help when b_i reaches certain density.

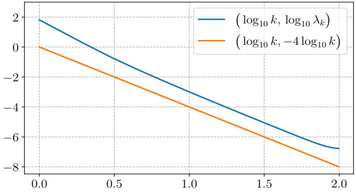
Implications to numerical computation in practice

- Low pass filter: given the machine precision ϵ, the eigenvalue threshold is $n\epsilon\lambda_1$. A two-layer neural network can use about $\epsilon^{-d/(d+3)}$ eigenmodes in d-dimensions or at most all Fourier modes up to frequency $k_d = \epsilon^{-1/(d+3)}$ can be resolved.
  single precision $\epsilon = 2^{-23}$: $k_1 \simeq 54, k_2 \simeq 25$, $k_3 \simeq 14$, $k_{10} = 4$.
  double precision $\epsilon = 2^{-52}$: $k_1 \simeq 8192, k_2 \simeq 1351$, $k_3 \simeq 411$, $k_{10} = 16$.
- Although two layer neural networks have universal approximation property when the width is increased, in practice, increasing the width does not help when b_i reaches certain density.
- Two layer neural networks can approximate smooth function well but not functions with significant high frequency components.

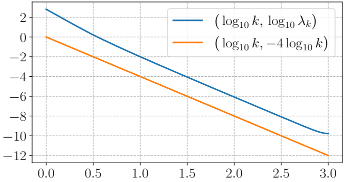
Implications to numerical computation in practice

- Low pass filter: given the machine precision ϵ, the eigenvalue threshold is $n\epsilon\lambda_1$. A two-layer neural network can use about $\epsilon^{-d/(d+3)}$ eigenmodes in d-dimensions or at most all Fourier modes up to frequency $k_d = \epsilon^{-1/(d+3)}$ can be resolved.
  single precision $\epsilon = 2^{-23}$: $k_1 \simeq 54, k_2 \simeq 25$, $k_3 \simeq 14$, $k_{10} = 4$.
  double precision $\epsilon = 2^{-52}$: $k_1 \simeq 8192, k_2 \simeq 1351$, $k_3 \simeq 411$, $k_{10} = 16$.
- Although two layer neural networks have universal approximation property when the width is increased, in practice, increasing the width does not help when b_i reaches certain density.
- Two layer neural networks can approximate smooth function well but not functions with significant high frequency components.
- Two layer neural networks is stable with respect to noise and over-parametrization.

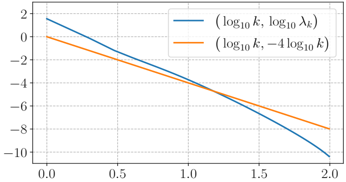
Numerical spectrum for Gram matrix (1D)



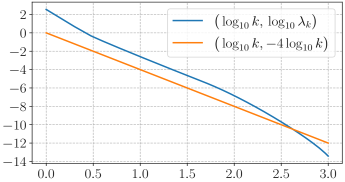
(a) n=100 (uniform bias)



(b) n=1000 (uniform bias)

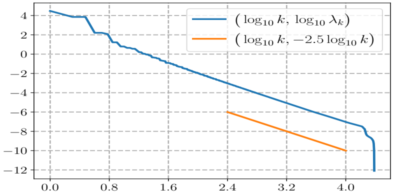


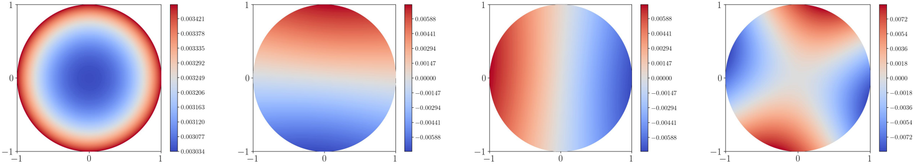
(a) n=100 (adaptive bias)



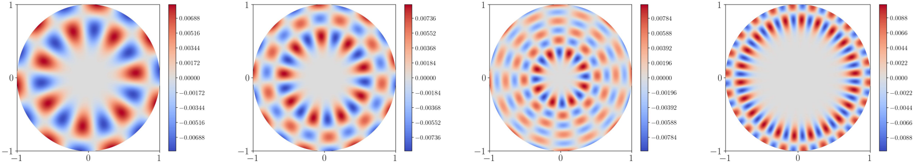
(b) n=1000 (adaptive bias)

Numerical spectrum for Gram matrix (2D)





ϕ_1 ϕ_2 ϕ_3 ϕ_4



ϕ_{50} ϕ_{100} ϕ_{200} ϕ_{250}

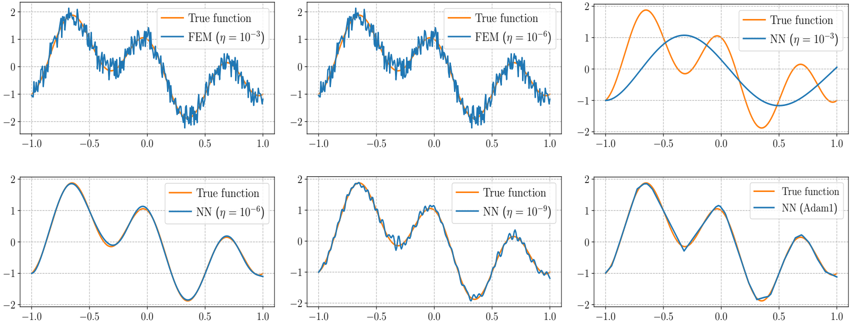
Numerical tests

Table 1: Error comparison for approximating $f(x) = \arctan(25x)$ with sufficient samples.

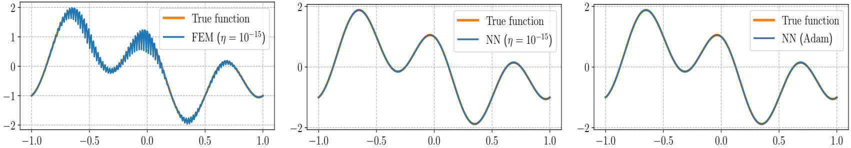
		float32				float64			
		$n = 100$		$n = 1000$		$n = 100$		$n = 1000$	
		MAX	MSE	MAX	MSE	MAX	MSE	MAX	MSE
NN	Uniform $\boldsymbol{b}$	6.09×10^{-2}	9.58×10^{-5}	7.19×10^{-2}	1.43×10^{-4}	1.37×10^{-2}	1.70×10^{-6}	1.05×10^{-4}	1.33×10^{-10}
FEM	Uniform $\boldsymbol{b}$	1.37×10^{-2}	1.70×10^{-6}	1.05×10^{-4}	1.33×10^{-10}	1.37×10^{-2}	1.70×10^{-6}	1.05×10^{-4}	1.33×10^{-10}
NN	Adaptive $\boldsymbol{b}$	6.83×10^{-2}	7.54×10^{-5}	1.89×10^{-2}	1.06×10^{-5}	3.93×10^{-3}	1.42×10^{-6}	4.74×10^{-5}	1.17×10^{-10}
FEM	Adaptive $\boldsymbol{b}$	2.92×10^{-3}	9.95×10^{-7}	3.79×10^{-5}	1.02×10^{-10}	2.92×10^{-3}	9.95×10^{-7}	3.77×10^{-5}	1.02×10^{-10}

Stability with respect to noise and over-parametrization

noise data



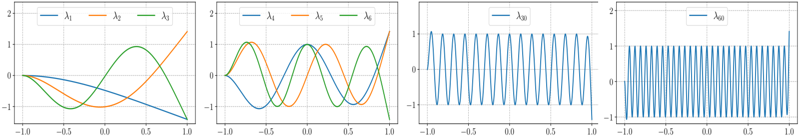
over-parametrization with 1000 samples and 1500 biases



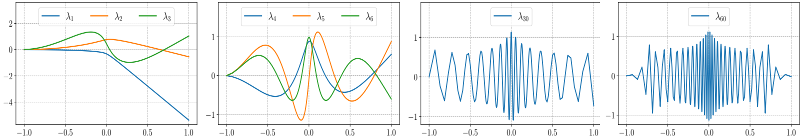
Adaptive vs uniform biases

Adaptive biases for $f(x) = \text{arctan}(25x)$. Define $F(x) = \int_{-1}^{x} |f'(t)|dt \big/ \int_{-1}^{1} |f'(t)|dt, \quad F(b_i) = (i-1)/(n-1)$.

Eigenmodes of λ_k for $k = \{1, 2, 3\}, \{4, 5, 6\}, 30, 60$ with $n = 1000$.

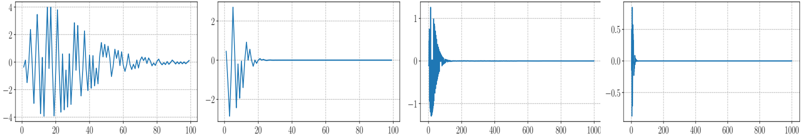


Uniformly distributed biases



adaptively distributed biases

Projection of f on the eigenmodes



uniform $n = 100$ | adaptive $n = 100$ | uniform $n = 1000$ | adaptive $n = 1000$

Training dynamics

Training is the most important process for machine learning.
Training two layer ReLU neural networks $h(x,t) = \sum_{i=1}^{n} a_i(t)\sigma(x - b_i(t))$ in 1D following the gradient flow of

$$E(t) = \frac{1}{2}\|h(x,t) - f(x)\|_D^2, \quad D = (-1,1)$$

$$\frac{da_i}{dt} = -\int_D (h(x,t) - f(x))\sigma(x - b_i)dx \quad \frac{db_i}{dt} = a_i \int_D (h(x,t) - f(x))\sigma'(x - b_i)dx.$$

Training dynamics

Training is the most important process for machine learning.
Training two layer `ReLU` neural networks $h(x,t) = \sum_{i=1}^{n} a_i(t)\sigma(x - b_i(t))$ in 1D following the gradient flow of

$$E(t) = \frac{1}{2}\|h(x,t) - f(x)\|_D^2, \quad D = (-1,1)$$

$$\frac{da_i}{dt} = -\int_D (h(x,t) - f(x))\sigma(x - b_i)dx \quad \frac{db_i}{dt} = a_i \int_D (h(x,t) - f(x))\sigma'(x - b_i)dx.$$

Basic questions:

- can the training process obtain the optimal a_i, b_i?

Training dynamics

Training is the most important process for machine learning.
Training two layer ReLU neural networks $h(x,t) = \sum_{i=1}^{n} a_i(t)\sigma(x - b_i(t))$ in 1D following the gradient flow of

$$E(t) = \frac{1}{2}\|h(x,t) - f(x)\|_D^2, \quad D = (-1,1)$$

$$\frac{da_i}{dt} = -\int_D (h(x,t) - f(x))\sigma(x - b_i)dx \quad \frac{db_i}{dt} = a_i \int_D (h(x,t) - f(x))\sigma'(x - b_i)dx.$$

Basic questions:

- can the training process obtain the optimal a_i, b_i?
- what is the computation cost of the training process?

Training dynamics

We show that training of high frequency components can be slow (even if the training process converges to the optimal solution).

Theorem

It takes at least $O(m)$ *time steps to get the initial error in Fourier mode* m *reduced by half.*

Training dynamics

We show that training of high frequency components can be slow (even if the training process converges to the optimal solution).

Theorem

It takes at least $O(m)$ time steps to get the initial error in Fourier mode m reduced by half.

Remark

1. *In practice, it can be worse!*
2. *Our result does not depend on convergence and is fully discrete (instead of letting $n \to \infty$ and using mean field formulation).*
3. *With some mild condition, the time step bound is $O(m^2)$.*
4. *With fixed biases, the time step bound is $O(m^3)$.*
5. *Smoother the activation function, the slower the training dynamics for high frequency components.*
6. *Experiments suggest Adam following a similar law at the initially.*

Key ideas in the proof I

Target function: $f(x), x \in (-1,1)$.
Two layer NN: $h(x,t) = \sum_{i=1}^{n} a_i(t)\sigma(x - b_i(t))$.

Important facts:
1. $\partial_x^2 \sigma(x-b) = \partial_b^2 \sigma(x-b) = \delta(x-b)$.
2. The eigenvalues (in descending order) and eigenfunctions of kernel

$$\mathcal{G}(x,y) = \int_D \sigma(z-x)\sigma(z-y)dz, \quad D = (-1,1)$$

are: $\lambda_k = O(k^{-4})$ and $\phi_k(x)$ (an orthonormal basis) satisfying

$$\phi_k^{(4)}(x) = \lambda_k^{-1}\phi_k(x),\ x \in (-1,1), \quad \phi_k(1) = \phi_k^{(1)}(1) = \phi_k^{(2)}(-1) = \phi_k^{(3)}(-1) = 0.$$

Key ideas in the proof II

Define

$$\theta_k(t) = \sum_{j=1}^{n} a_j(t)\phi_k(b_j(t)) - \frac{p_k}{\lambda_k},$$

where $p(b) = \int_D f(x)\sigma(x-b)dx$, $p(b) = \sum_{k\geq 1} p_k\phi_k(b)$.

$$\frac{d\theta_k(t)}{dt} = -\sum_{l=1}^{\infty} \lambda_l \left[M_{lk}(t) + S_{lk}(t)\right] \theta_l(t)$$

where $M_{lk}(t) = \sum_{i=1}^{n} \phi_l(b_i(t))\phi_k(b_i(t))$, $S_{lk}(t) = \sum_{i=1}^{n} |a_i(t)|^2 \phi'_k(b_i(t))\phi'_l(b_i(t))$.

Key ideas in the proof II

Define

$$\theta_k(t) = \sum_{j=1}^{n} a_j(t)\phi_k(b_j(t)) - \frac{p_k}{\lambda_k},$$

where $p(b) = \int_D f(x)\sigma(x-b)dx$, $p(b) = \sum_{k\geq 1} p_k \phi_k(b)$.

$$\frac{d\theta_k(t)}{dt} = -\sum_{l=1}^{\infty} \lambda_l \left[M_{lk}(t) + S_{lk}(t)\right]\theta_l(t)$$

where $M_{lk}(t) = \sum_{i=1}^{n} \phi_l(b_i(t))\phi_k(b_i(t))$, $S_{lk}(t) = \sum_{i=1}^{n} |a_i(t)|^2 \phi'_k(b_i(t))\phi'_l(b_i(t))$.

The key auxiliary function: $w(b,t) = \sum_{k=1}^{\infty} \lambda_k \theta_k(t)\phi_k(b) \in H_D^2$, satisfying

$$\partial_b^2 w(b,t) = h(b,t) - f(b).$$

Key ideas in the proof II

Define

$$\theta_k(t) = \sum_{j=1}^{n} a_j(t)\phi_k(b_j(t)) - \frac{p_k}{\lambda_k},$$

where $p(b) = \int_D f(x)\sigma(x-b)dx$, $p(b) = \sum_{k\geq 1} p_k\phi_k(b)$.

$$\frac{d\theta_k(t)}{dt} = -\sum_{l=1}^{\infty} \lambda_l \left[M_{lk}(t) + S_{lk}(t)\right] \theta_l(t)$$

where $M_{lk}(t) = \sum_{i=1}^{n} \phi_l(b_i(t))\phi_k(b_i(t))$, $S_{lk}(t) = \sum_{i=1}^{n} |a_i(t)|^2 \phi_k'(b_i(t))\phi_l'(b_i(t))$.

The key auxiliary function: $w(b,t) = \sum_{k=1}^{\infty} \lambda_k\theta_k(t)\phi_k(b) \in H_D^2$, satisfying

$$\partial_b^2 w(b,t) = h(b,t) - f(b).$$

The dynamics for the "Fourier" mode $\hat{w}(\eta,t)$ satisfy

$$\frac{d}{dt}\hat{w}(m,t) = -\frac{n}{|\pi\eta|^4}\widehat{\mu_0 w}(\eta,t) - \frac{n}{|\pi\eta|^4}(i\eta\pi)\widehat{\mu_2\partial_b w}(\eta,t),$$

where $\mu_0(b,t) = \frac{1}{n}\sum_{i=1}^{n} \delta(b - b_i(t))$, $\mu_2(b,t) = \frac{1}{n}\sum_{i=1}^{n} |a_i(t)|^2\delta(b - b_i(t))$.

Rashomon set for two layer NN

Given a target function $f(x), x \in D = B_d(1)$. Denote $\mathcal{Q}_{\mathcal{H}_n}$ to be the parameter domain for the two-layer `ReLU` neural network class

$$\mathcal{H}_n = \{h(x)|h(x) = \frac{1}{n}\sum_{j=1}^{n} a_j\sigma(w_j \cdot x - b_j), w_j \in \mathbb{S}^{d-1}, |a_j| \leq A, |b_j| \leq 1\}$$

The Rashomon set $\mathcal{R}_\epsilon(f) \subset \mathcal{Q}_{\mathcal{H}_n}$

$$\mathcal{R}_\epsilon(f) := \{(w_j, a_j, b_j) \in \mathcal{Q}_{\mathcal{H}_n}, s.t. \|h(\cdot; w_j, a_j, b_j) - f(\cdot)\|_{L^2(D)} \leq \epsilon \|f\|_{L^2(D)}\}$$

Normalize the measure on $\mathcal{Q}_{\mathcal{H}_n}$, size of $\mathcal{R}_\epsilon(f)$ characterizes the likelihood that the loss is under certain threshold of relative error or how "easy" f can be approximated by $\mathcal{H}_n$.

Rashomon set for two layer NN

Theorem

Suppose $f \in C(D)$ such that there exists $g \in C_0^2(D)$ that $\Delta g = f$, then

$$\mathbb{P}(\mathcal{R}_\epsilon) \leq \exp\left(-\frac{n(1-\epsilon)^2\|f\|_{L^2(D)}^4}{2A^2\kappa^2}\right), \quad \kappa := \sup_{(w,b)} \int_{x\in D, w\cdot x=b\}} g(x)dH_{d-1}(x).$$

Remark

- *If f oscillates with frequency ν in all directions, then $\kappa \approx \nu^{-2}$ $\Rightarrow \mathbb{P}(\mathcal{R}_\epsilon) \sim \exp(-\mathcal{O}(\nu^{-4}))$, which makes the approximation of oscillatory function difficult.*
- *Similar result holds for other bounded activation functions.*

Key observations

$$h(x) = \frac{1}{n}\sum_{j=1}^{n} a_j\sigma(w_j\cdot x - b_j) \quad \Rightarrow \; \Delta h(x) = \frac{1}{n}\sum_{j=1}^{n} a_j\delta(w_j\cdot x - b_j)$$

$$\Rightarrow \; \langle h, f\rangle \overset{\Delta g = f}{=} \langle \Delta h, g\rangle = \frac{1}{n}\sum_{j=1}^{n} X_j, \quad X_j = a_j\int_{w_j\cdot x = b_j} g(x)dH_{d-1}(x)$$

X_j are i.i.d in $[-A\kappa, A\kappa]$, $E[X_j] = 0$, $\kappa := \sup_{(w,b)}\int_{\{x\in D, w\cdot x = b\}} g(x)dH_{d-1}(x)$

$$\mathbb{P}\Big[\|h-f\|_{L^2(D)} \le \epsilon\|f\|_{L^2(D)}\Big] \le \mathbb{P}\Big[\langle h, f\rangle \ge (1-\epsilon)\|f\|^2_{L^2(D)}\Big] \le \exp\left(-\frac{n(1-\epsilon)^2\|f\|^4_{L^2(D)}}{2A^2\kappa^2}\right)$$

by Hoeffding's inequality

$$\mathbb{P}\left[\frac{1}{n}\sum_{j=1}^{n} X_j - E[X_j] \ge t\right] \le \exp\left(-\frac{nt^2}{2A^2\kappa^2}\right).$$

$\sigma(x)$ does not see oscillations well!

Key observations

$$h(x) = \frac{1}{n}\sum_{j=1}^{n} a_j\sigma(w_j \cdot x - b_j) \quad \Rightarrow \; \Delta h(x) = \frac{1}{n}\sum_{j=1}^{n} a_j\delta(w_j \cdot x - b_j)$$

$$\Rightarrow \; \langle h, f\rangle \overset{\Delta g = f}{=} \langle \Delta h, g\rangle = \frac{1}{n}\sum_{j=1}^{n} X_j, \quad X_j = a_j \int_{w_j \cdot x = b_j} g(x) dH_{d-1}(x)$$

X_j are i.i.d in $[-A\kappa, A\kappa]$, $E[X_j] = 0$, $\kappa := \sup_{(w,b)} \int_{\{x\in D, w\cdot x = b\}} g(x) dH_{d-1}(x)$

$$\mathbb{P}\Big[\|h-f\|_{L^2(D)} \leq \epsilon \|f\|_{L^2(D)}\Big] \leq \mathbb{P}\Big[\langle h, f\rangle \geq (1-\epsilon)\|f\|^2_{L^2(D)}\Big] \leq \exp\left(-\frac{n(1-\epsilon)^2\|f\|^4_{L^2(D)}}{2A^2\kappa^2}\right)$$

by Hoeffding's inequality

$$\mathbb{P}\left[\frac{1}{n}\sum_{j=1}^{n} X_j - E[X_j] \geq t\right] \leq \exp\left(-\frac{nt^2}{2A^2\kappa^2}\right).$$

$\sigma(x)$ does not see oscillations well! $\langle \sigma, f\rangle = \int_{\{x\in D, w\cdot x = b\}} \Delta^{-1} f(x) dH_{d-1}(x)$

Further discussions

- Activation functions of the form $\sigma(w \cdot x - b)$ is global and see smooth and large structure well.
- Difficult to approximate highly oscillatory functions.
- `ReLU` is the best in terms of approximation property and learning dynamics.

Further discussions

- Activation functions of the form $\sigma(w \cdot x - b)$ is global and see smooth and large structure well.
- Difficult to approximate highly oscillatory functions.
- `ReLU` is the best in terms of approximation property and learning dynamics.

Questions:

- Deep NNs, Transformers, network structure, ...
- For challenging problems, problem specific knowledge should be involved.

Reference:
Why Shallow Networks Struggle with Approximating and Learning High Frequency: A Numerical Study,
S. Zhang, H. Zhao, Y. Zhong and H. Zhou. arXiv:2306.17301, 2023.

Further discussions

- Activation functions of the form $\sigma(w \cdot x - b)$ is global and see smooth and large structure well.
- Difficult to approximate highly oscillatory functions.
- `ReLU` is the best in terms of approximation property and learning dynamics.

Questions:

- Deep NNs, Transformers, network structure, ...
- For challenging problems, problem specific knowledge should be involved.

Reference:
Why Shallow Networks Struggle with Approximating and Learning High Frequency: A Numerical Study,
S. Zhang, H. Zhao, Y. Zhong and H. Zhou. arXiv:2306.17301, 2023.

A 2D test example:

$$f(x) = \sum_{ij} a_{ij} sin(b_i x_i + c_{ij} x_i x_j) cos(b_j x_j + d_{ij} x_i^2)$$