Finite Neuron Method

Jinchao Xu

KAUST and Penn State

xu@multigrid.org

CIRM, July 18th, 2023

CEMRACS 2023 Summer School

KAUST Baseline Research Fund

Course outline

Finite Neuron Method

1. Finite Element, ReLU DNN and Approximation
   - Finite elements versus ReLU and ReLU^k neural networks
   - Approximation of shallow neural networks
   - Metric entropy
   - Deep ReLU NNs and hierarchical basis
2. Iterative Methods and Frequency Principle
   - Linear systems and basic iterative methods
   - Frequency principle
   - Multigrid methods
   - Subspace correction and federated learning
3. Finite Neuron Method
   - Model problems
   - Shallow NN and stable NN approximation
   - Error estimates for numerical quadrature of stable NN approximation
   - Optimization algorithms
   - Numerical examples
4. Computer session

Finite Neuron Method

Finite Element, ReLU DNN and Approximation

Jinchao Xu

KAUST and Penn State

xu@multigrid.org

CIRM, July 18th, 2023

CEMRACS 2023 Summer School

KAUST Baseline Research Fund

Outline

Finite element: Piecewise linear functions

- Uniform grid $\mathcal{T}_h$

$$0 = x_0 < x_1 < \cdots < x_{N+1} = 1, \quad x_j = \frac{j}{N+1} \ (j = 0 : N+1).$$

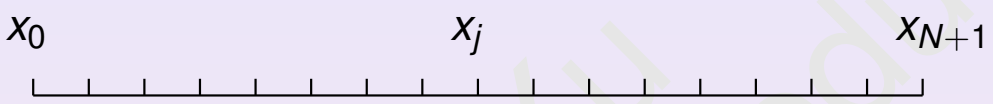


Figure: 1D uniform grid

- Linear finite element space

$$V_h = \{v_h : \ v \text{ is continuous and piecewise linear w.r.t. } \mathcal{T}_h \ , v_h(0) = 0\}.$$

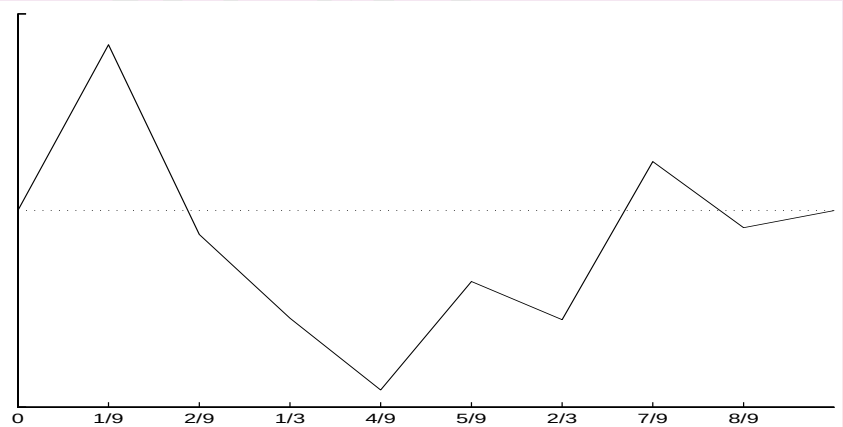
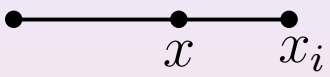


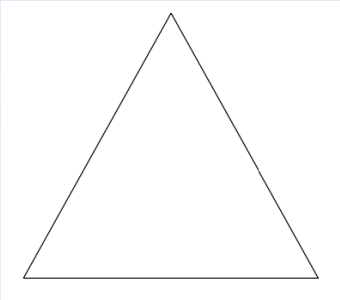
Figure: Linear finite element functions.

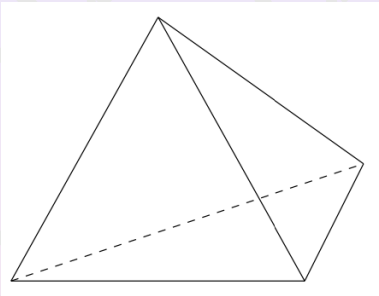
Finite element in multi-dimensions

$(k = 1)$

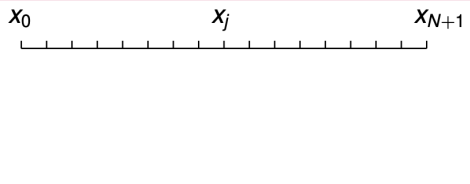
$w_1 x + b$ $\quad$ $w_1 x_1 + w_2 x_2 + b$ $\quad$ $w_1 x_1 + w_2 x_2 + w_3 x_3 + b$ $\quad \cdots$

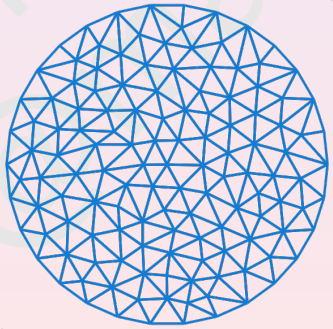


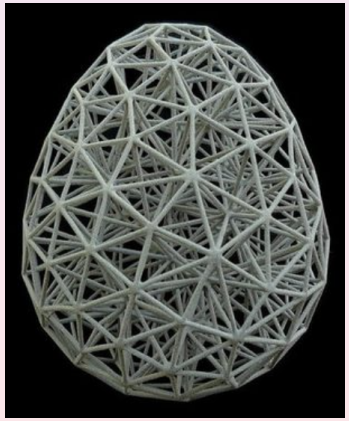




$\cdots$







$\cdots$

Two basis of the finite element space V_h

- Hat basis:

$$\varphi(x) = \begin{cases} x & x \in [0,1] \\ 2-x & x \in [1,2] \\ 0, & \text{others} \end{cases}.$$

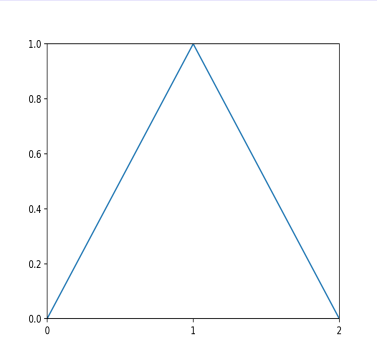
$$\varphi_i(x) = \varphi(\frac{x - x_{i-1}}{h}) = \varphi(w_h x + b_i).$$

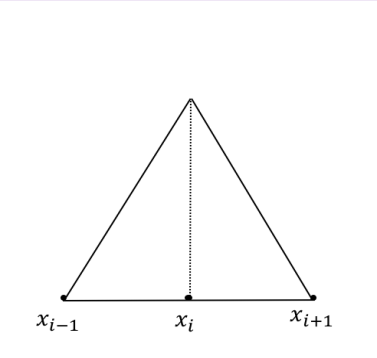
with $w_h = \frac{1}{h}, \quad b_i = \frac{-x_{i-1}}{h}$.

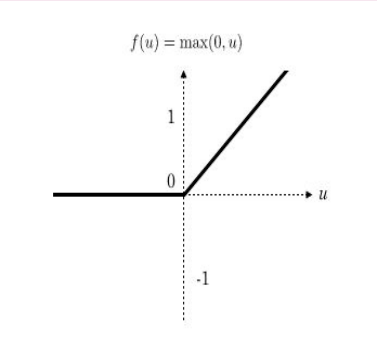
- ReLU basis: $\mathrm{ReLU}(x) = \max(0, x)$ and

$$r_i(x) = \mathrm{ReLU}(\frac{x - x_{i-1}}{h}) = \mathrm{ReLU}(w_h x + b_i)$$

- $V_h = \mathrm{span}\{\varphi(w_h x + b_i)\} = \mathrm{span}\{\mathrm{ReLU}(w_h x + b_i)\}$







Hat and ReLU bases on a uniform grid

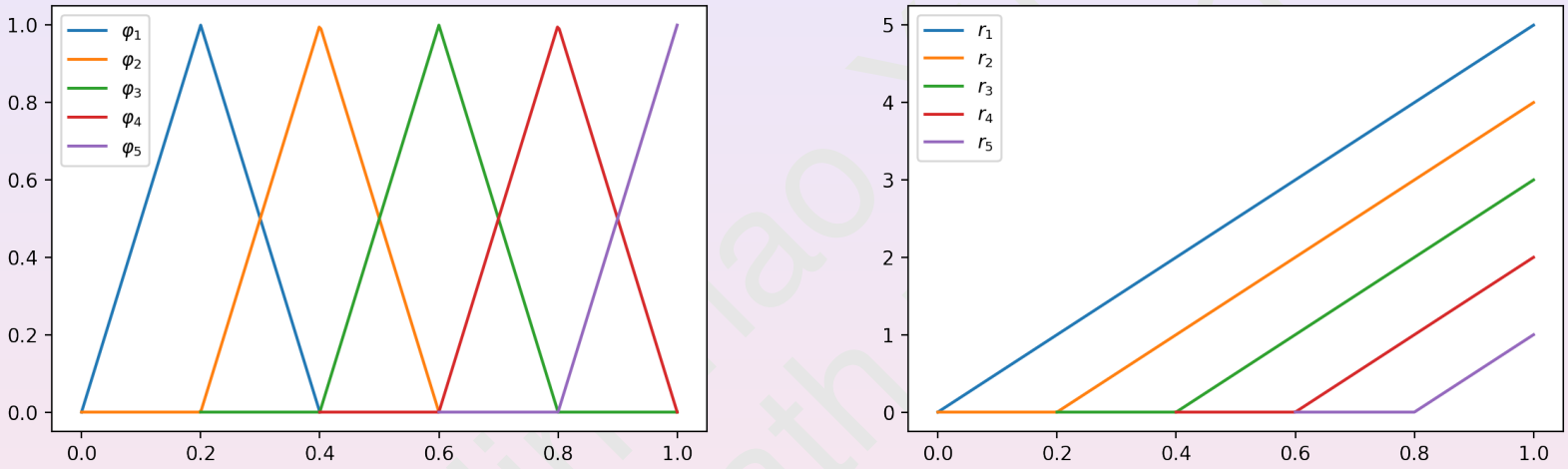


Figure: Left: Hat bases. Right: ReLU bases.

Finite element space and neural network functions

- Finite element function space on a uniform grid

$$V_h = \left\{ \sum_{i=1}^{n} a_i \varphi(w_h x + b_i), i = 1:n \right\} = \left\{ \sum_{i=1}^{n} a_i \mathrm{ReLU}(w_h x + b_i), i = 1:n \right\} \tag{1}$$

with $w_h = \frac{1}{h}, \quad b_i = \frac{-x_{i-1}}{h}$.

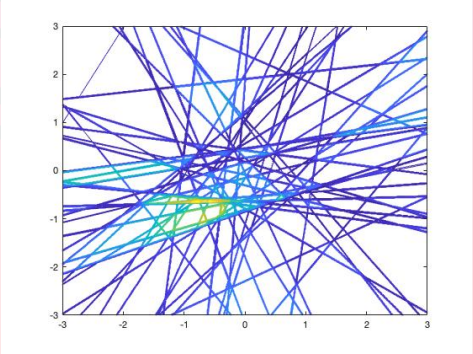
- Finite element function space on an arbitrary grid

$$\Sigma_n^{\mathrm{ReLU}} = \left\{ \sum_{i=1}^{n} a_i \mathrm{ReLU}(w_i x + b_i), w_i, b_i \in \mathbb{R}, i = 1:n \right\}, \quad x_i = -\frac{b_i}{w_i}, \tag{2}$$

is contained in the ReLU neural network function space.

- ReLU neural network (shallow) functions in any dimension d

$$\Sigma_n^{\mathrm{ReLU}} = \left\{ \sum_{i=1}^{n} a_i \mathrm{ReLU}(w_i \cdot x + b_i), w_i \in \mathbb{R}^d, b_i \in \mathbb{R}, i = 1:n \right\} \tag{3}$$



Shallow NNs with general activation functions

Change ReLU to general activation function $\sigma : \mathbb{R} \mapsto \mathbb{R}$:

$$\Sigma_n^\sigma = \left\{ \sum_{i=1}^{n} a_i \sigma(w_i \cdot x + b_i), w_i \in \mathbb{R}^d, b_i \in \mathbb{R}, i = 1 : n \right\} \tag{4}$$

Popular activation functions:

- Heaviside $\sigma = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$
- Sigmoidal $\sigma = (1 + e^{-x})^{-1}$
- Rectified Linear $\sigma = \max(0, x)$
- ReLU^k: Power of a ReLU $\sigma = [\max(0, x)]^k$
- Cosine $\sigma = \cos(x)$
- $\ldots$

Deep neural network (DNN)

1. Start from a linear function of $x \in \mathbb{R}^{n_0}$
$$W^0 x + b^0$$
2. Compose with the activation function:
$$x^{(1)} = \sigma(W^0 x + b^0) \in \mathbb{R}^{n_1}$$
3. Compose with another linear function:
$$W^1 x^{(1)} + b^1$$
4. Compose with the activation function:
$$x^{(2)} = \sigma(W^1 x^{(1)} + b^1) \in \mathbb{R}^{n_2}$$
5. ...
6. $$x^{(L)} = \sigma(W^{L-1} x^{(L-1)} + b^{L-1}) \in \mathbb{R}^{n_L}$$
7. Compose with another linear function (final output layer)
$$f(x; \Theta) = W^L x^{(L)} + b^L$$

Outline

ReLU-DNN and FEM

Deep neural network functions with ℓ-hidden layers

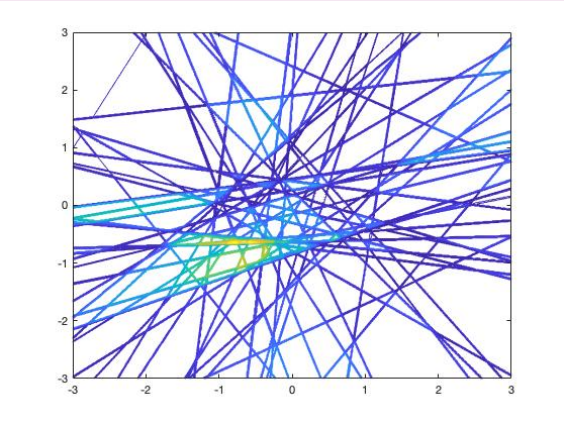
$$\Sigma^{\sigma}_{n_{1:\ell}} = \{ W^{\ell} x^{(\ell)} + b^{\ell},\ W^{\ell} \in \mathbb{R}^{n_{\ell+1} \times n_{\ell}},\ b^{i} \in \mathbb{R}^{n_{\ell+1}} \}$$

ReLU-DNN $\Leftrightarrow$ FEM

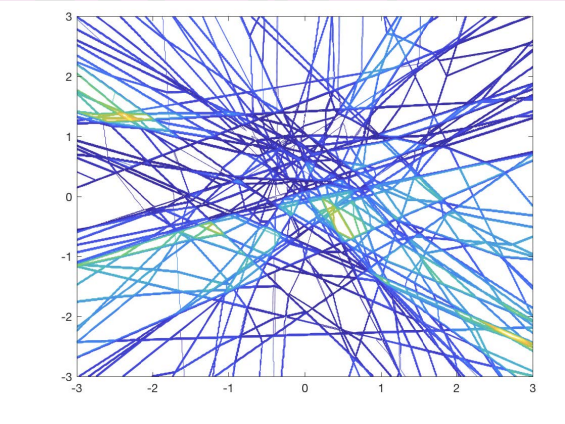
ReLU^k-DNN

$$\Sigma^{k}_{n_{1:\ell}} := \Sigma^{ReLU^k}_{n_{1:\ell}}. \tag{5}$$

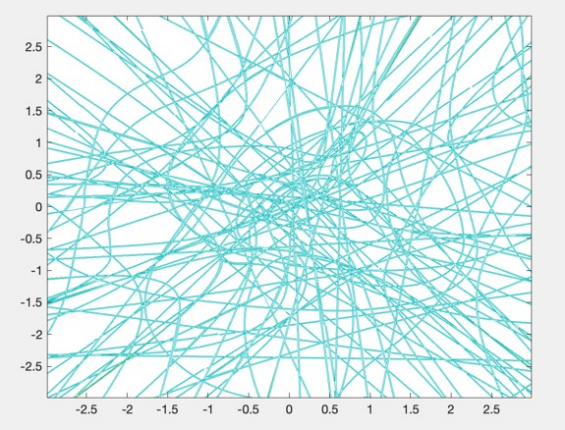
ReLU^k-DNN $= \Sigma^{k}_{n_{1:\ell}} =$ *piecewise polynomials* $\subset H^k(\Omega)$



(1) $\ell = 1, k \geq 1$



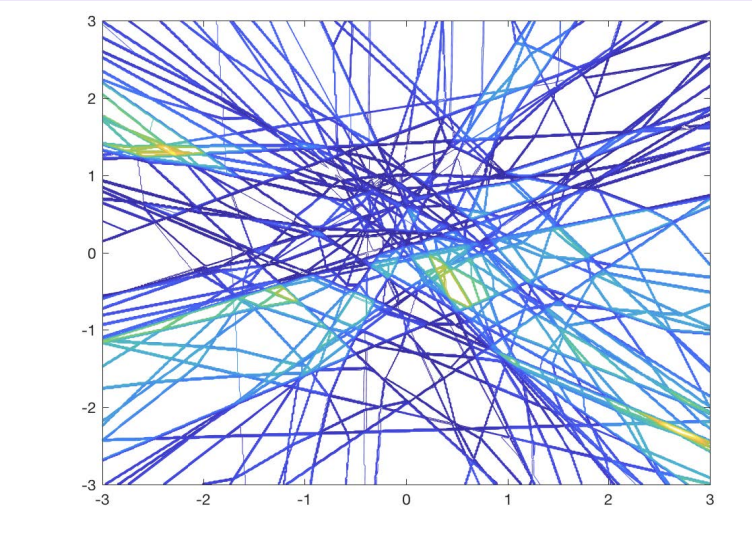
(2) $\ell = 2, k = 1$



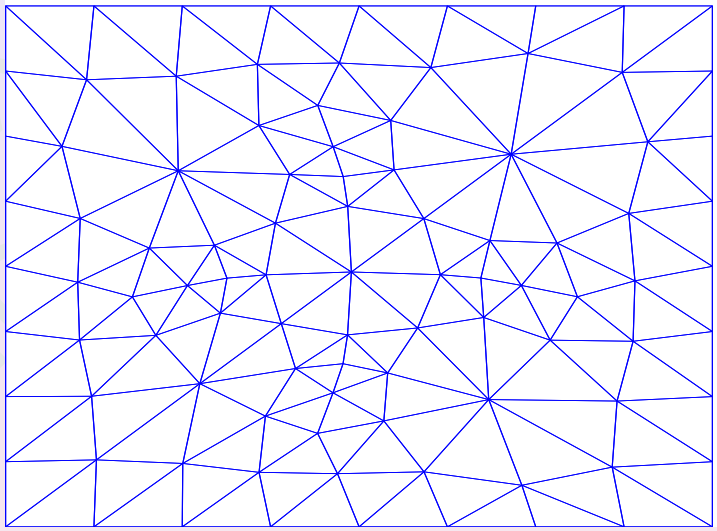
(3) $\ell = 2, k = 2$

ReLU-DNN = FEM

ReLU-DNN $= \Sigma^1_{n_{1:\ell}} \subset$ Linear FEM $\subset H^1(\Omega)$



$\Longrightarrow$



- Ref: Arora, R., Basu, A., Mianjy, P. & Mukherjee, A. (2016)., He, J., Li, L., Xu, J., & Zheng, C. (2020), He, J., Li, L., & Xu, J. (2022).

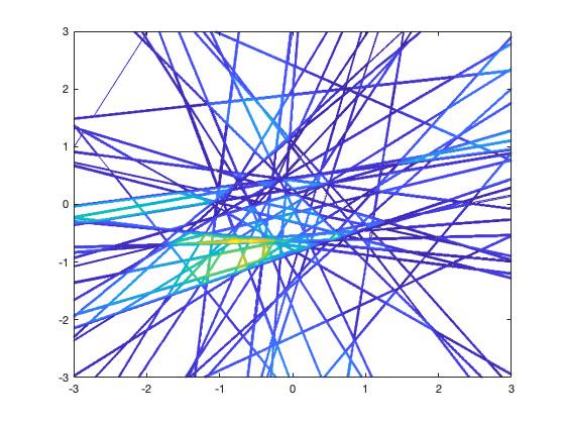
Connection of ReLU DNN and linear FEM

Theorem (He, Li, X and Zheng 2018)

For $d \geq 2$, not all linear FE functions can be represented by a shallow neural network, namely

$$\mathrm{LFE} \not\subset \Sigma_{n_1}^1 .$$

- Proof: $\Sigma_{n_1}^1$ can not represent locally supported functions;



- Deep is necessary for high dimensional spaces;

Connection of ReLU DNN and linear FEM

Theorem (He, Li, X and Zheng 2018)

Given a locally convex finite element grid $\mathcal{T}_h$, any linear finite element function with N degrees of freedom, it can be written as a function in $\Sigma^1_{n_{1:J}}$ with

$$J = 2 + \lceil \log_2 d_h \rceil \text{ and } N_{\text{neurons}} = \mathcal{O}(d_h N).$$

where d_h is the largest number of elements that share one vertex.

- Theoretically, $J \leq \lceil \log_2(d+1) \rceil$, however

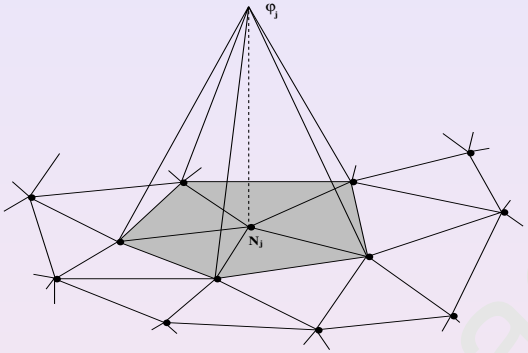
$$N_{\text{neurons}} = \mathcal{O}\left(d 2^{N!+dN}\right)$$

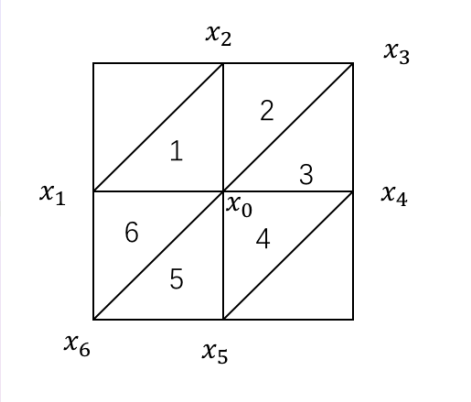
  - Ref: R. Arora, A. Basu, P. Mianjy and A. Mukherjee, 2016

- One key idea: represent basis functions of LFE in terms of DNN more effectively
  - See the following example.

A 2D example: FE basis function

$\phi(x)$:





Here g_i is linear in Domain i, and $x_7 = x_1$, satisfying

$$g_i(x_0) = 1 \qquad g_i(x_i) = 0 \qquad g_i(x_{i+1}) = 0$$

$$\phi(x) = \begin{cases} g_i(x), & x \in \text{Domain } i \\ 0. & x \in \mathbb{R}^2 - \overline{x_1 x_2 x_3 x_4 x_5 x_6} \end{cases} \tag{6}$$

Then we have

$$\phi(x) = \mathrm{ReLU}(\min(g_1, g_2, g_3, g_4, g_5, g_6)), \tag{7}$$

if the support of $\phi(x)$ is convex.

Properties of ReLU

Important identities:

$$x = \mathrm{ReLU}(x) - \mathrm{ReLU}(-x), |x| = \mathrm{ReLU}(x) + \mathrm{ReLU}(-x)$$

and

$$\min(a, b) = \frac{a+b}{2} - \frac{|a-b|}{2} = v \cdot \mathrm{ReLU}(W \cdot [a, b]^T) \tag{8}$$

where

$$v = \frac{1}{2}[1, -1, -1, -1] \qquad W = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Properties of ReLU

Important identities:

$$x = \mathrm{ReLU}(x) - \mathrm{ReLU}(-x), |x| = \mathrm{ReLU}(x) + \mathrm{ReLU}(-x)$$

and

$$\min(a, b) = \frac{a+b}{2} - \frac{|a-b|}{2} = v \cdot \mathrm{ReLU}(W \cdot [a, b]^T) \tag{9}$$

where

$$v = \frac{1}{2}[1, -1, -1, -1] \qquad W = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Example of DNN and FEM: 2D-FEM basis function

$$
\begin{aligned}
\min(a,b,c) &= \min(\min(a,b),c) \\
&= v\cdot \mathrm{ReLU}\left(W\cdot\begin{pmatrix}\min(a,b)\\ c\end{pmatrix}\right) \\
&= v\cdot \mathrm{ReLU}\left(W\cdot\begin{pmatrix}v\cdot \mathrm{ReLU}(W\cdot[a,b]^T)\\ \mathrm{ReLU}(c)-\mathrm{ReLU}(-c)\end{pmatrix}\right) \\
&= v\cdot \mathrm{ReLU}\left(W\cdot\begin{pmatrix}v\cdot \mathrm{ReLU}(W\cdot[a,b]^T)\\ [1,-1]\cdot \mathrm{ReLU}([1,-1]^T c)\end{pmatrix}\right) \\
&= v\cdot \mathrm{ReLU}\left(W_2\cdot \mathrm{ReLU}(W_1\cdot[a,b,c]^T)\right)
\end{aligned}
$$

where

$$
W_2 = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -1 & 1 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -1 & 1 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & -1 \end{bmatrix}. \qquad W_1 = \begin{bmatrix} 1 & 1 & 0 \\ -1 & -1 & 0 \\ 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}.
$$

Example of DNN and FEM: 2D-FEM basis function

It follows that

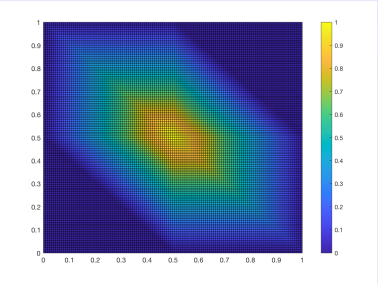
$$
\begin{aligned}
g \equiv & \min(g_1, g_2, g_3, g_4, g_5, g_6) = \min(\min(g_1, g_2, g_3), \min(g_4, g_5, g_6)) \\
= & v \cdot \mathrm{ReLU}(W \cdot \begin{bmatrix} \min(g_1, g_2, g_3) \\ \min(g_4, g_5, g_6) \end{bmatrix}) \\
= & v \cdot \mathrm{ReLU}(W \cdot \begin{bmatrix} v \cdot \mathrm{ReLU}(W_2 \cdot \mathrm{ReLU}(W_1 \cdot [g_1, g_2, g_3]^T) \\ v \cdot \mathrm{ReLU}(W_2 \cdot \mathrm{ReLU}(W_1 \cdot [g_4, g_5, g_6]^T) \end{bmatrix})
\end{aligned}
$$

Thus

$$
\phi = \mathrm{ReLU}\left(v \cdot \mathrm{ReLU}\left(W \cdot \begin{bmatrix} v \cdot \mathrm{ReLU}(W_2 \cdot \mathrm{ReLU}(W_1 \cdot [g_1, g_2, g_3]^T)) \\ v \cdot \mathrm{ReLU}(W_2 \cdot \mathrm{ReLU}(W_1 \cdot [g_4, g_5, g_6]^T)) \end{bmatrix} \right) \right) \in \Sigma^1_{n_{1:4}}
$$

A more compact result for 2D LFE on the uniform grid

The following identity holds on $\mathbb{R}^2$,



$$= \frac{1}{2}\left(g_2\left(\frac{\mathrm{ReLU1}(x)}{2}\right) + g_2\left(\frac{\mathrm{ReLU1}(y)}{2}\right) - g_2\left(\frac{\mathrm{ReLU1}(x)+\mathrm{ReLU1}(y)}{2}\right)\right),$$

where

$$g_2(x) = \quad = \sum_{i=0}^{4} \alpha_i \mathrm{ReLU}\left(x - \frac{i}{4}\right) \in \Sigma_5^1, \tag{10}$$

and

$$\mathrm{ReLU1}(x) := \mathrm{ReLU}(x) - \mathrm{ReLU}(x-1) \in \Sigma_2^1. \tag{11}$$

Question: Other approaches for this representation?

Theorem (He, Li and X 2022)

Let V_h be the LFE space on a uniform mesh on 2D, then

$$V_h \subset \Sigma^1_{n_{1:2}}, \tag{12}$$

and it can be represented explicitly and concisely.

- J. He, L. Li, J. Xu and C. Zheng, 2018: $u(x, y)$ can not be reproduced by one hidden layer.
- R. Arora, A. Basu, P. Mianjy and A. Mukherjee, 2016: $u(x, y) \in \Sigma^1_{n_{1:2}}$, but it is almost impossible to represent $u(x, y)$ explicitly.

ReLU-DNN = LFE

Conclusion:

Any linear finite element function can be represented by a deep ReLU neural network with a relatively deeper and narrower structure.

Theorem

$$\textit{ReLU-DNN} = \mathrm{LFE}$$

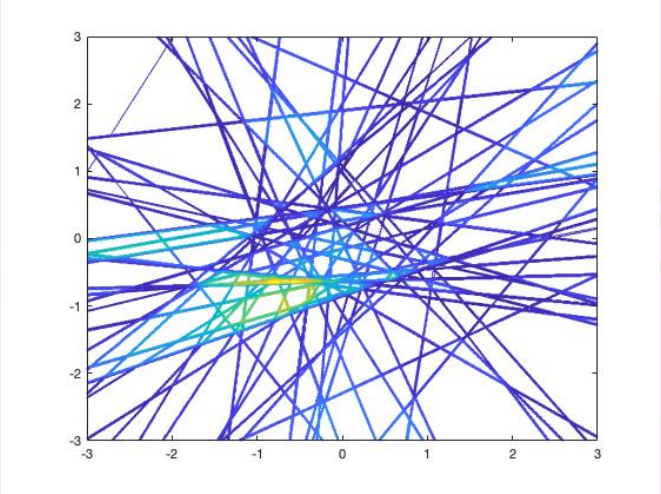
Outline

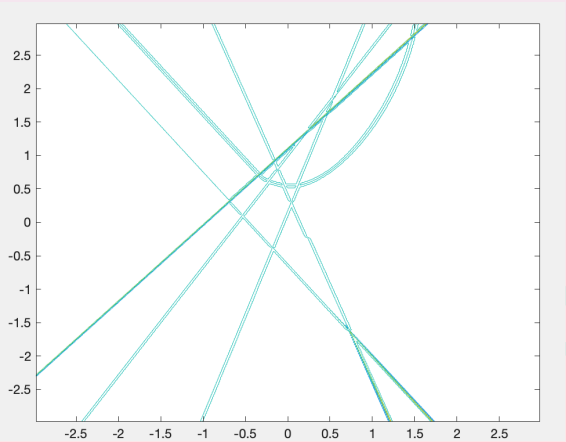
Shallow ReLU^k neural networks and polynomials

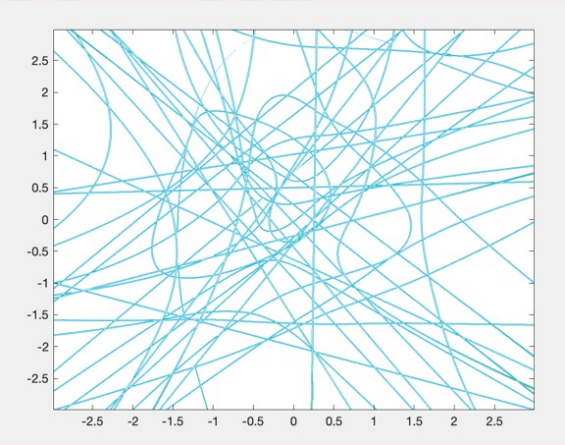
Theorem (X 2020)

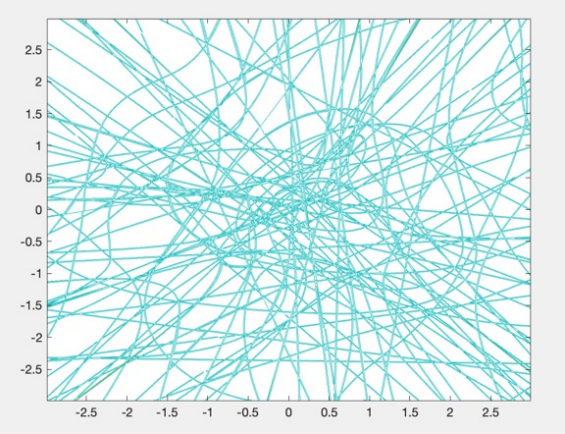
Polynomials of order k in $\mathbb{R}^d$ $\subset \Sigma_{n_1}^k$.

- $$u^k = \mathrm{ReLU}^k(u) + (-1)^k \mathrm{ReLU}^k(-u) \quad \forall u \in \mathbb{R}$$
- All polynomials with degree $\leq k$ is a linear combination of polynomials with form $(\alpha \cdot x + b)^k$.









Deep ReLU^k neural networks

$\Sigma^k_{n_{1:\ell}}$ for $k \geq 2$: Piecewise polynomials :

- "Simplex" (global) elements: any $k \geq 2$ but $\ell = 1$

- "Curved" elements: any $k \geq 2$ and $\ell \geq 2$:

Deep ReLU^k neural networks

1. $k = 2$, recover all polynomials as ℓ increase:
   - For $k = 2$, $x^2 \in \Sigma_2^2$;
   - $xy = \frac{1}{4}\left((x+y)^2 - (x-y)^2\right) \in \Sigma_{n_1}^2$;
   - Polynomial with order n on $R^d \subset \Sigma_{n_{1:\ell}}^2$ with $\ell \leq \log_2(n)$;
   - Spectral accuracy for smooth functions.

2. Possible multi-scale adaptivity features (?):
   - local singularity.
   - global smoothness

- Ref: J. Siegel and J. Xu 2019; B. Li, S. Tang, and H. Yu 2020 ($k = 2$).

Outline

Basic approximation properties

Can shallow networks approximate arbitrary functions?

- Recall $\Sigma_n^\sigma := \{\sum_{i=1}^n a_i \sigma(\omega_i \cdot x + b_i),\ a_i \in \mathbb{R},\ \omega_i \in \mathbb{R}^d,\ b_i \in \mathbb{R}\}$
- Is

$$\Sigma^\sigma = \bigcup_{n=1}^{\infty} \Sigma_n^\sigma \tag{13}$$

  dense in $C^0(\Omega)$ for bounded $\Omega \subset \mathbb{R}^d$?
- No! if σ is a polynomial.
- What σ?
- Yes! As long as σ is not a polynomial.

- Ref: M. Leshno, V. Lin, A. Pinkus and S. Schocken, 1993.

Activation function: non-polynomial

Lemma

Σ^σ is dense in $C^0(\Omega) \iff \sigma$ is not a polynomial (and almost everywhere continuous).

Proof.

1. Assume $\sigma \in C^\infty$
   - $[\sigma((\omega + he_j)\cdot x + b) - \sigma(\omega \cdot x + b)]/h \in \Sigma^\sigma$,
   - $\frac{\partial}{\partial \omega_j}\sigma(\omega \cdot x + b)|_{\omega=0} = x_j \sigma'(b) \in \overline{\Sigma^\sigma}$
     $\Rightarrow x_j \in \overline{\Sigma^\sigma}$ if $\sigma'(b) \neq 0$ for some b.
   - $D^\alpha_\omega \sigma(\omega \cdot x + b) = x^\alpha \sigma^{(|\alpha|)}(\omega \cdot x + b) \in \overline{\Sigma^\sigma}$
     $\Rightarrow x^\alpha = x_1^{\alpha_1} \cdots x_d^{\alpha_d} \in \overline{\Sigma^\sigma}$ if $\sigma^{(|\alpha|)}(b) \neq 0$ for some b
   - $\overline{\Sigma^\sigma}$ contains all polynomials.
2. If σ is not smooth ...

□

Our interest: What about approximation rates?

- Ref: M. Leshno, V. Lin, A. Pinkus and S. Schocken, 1993.

Convergence rate: approximation of Cosine networks

- Cosine networks

$$\Sigma_n^{\cos} = \left\{ u_n : u_n = \sum_{i=1}^{n} a_i \cos(\omega_i \cdot x + b_i), \quad \forall a_i, w_i, b_i \right\}$$

- Integral representation of u in terms of cosine functions

$$\begin{aligned} u(x) &= Re \int_{\mathbb{R}^d} e^{2\pi i \omega \cdot x} \hat{u}(\omega) d\omega \\ &= \int_{\mathbb{R}^d} \cos(2\pi\omega \cdot x + b(\omega)) |\hat{u}(\omega)| d\omega \qquad (\text{let } \hat{u}(\omega) = |\hat{u}(\omega)| e^{ib(\omega)}) \\ &= \|\hat{u}\|_{L^1} \int_{\mathbb{R}^d} \cos(2\pi\omega \cdot x + b(\omega)) \lambda(\omega) d\omega \qquad (\text{let } \lambda(\omega) = \tfrac{|\hat{u}(\omega)|}{\|\hat{u}\|_{L^1}}) \\ &= \|\hat{u}\|_{L^1} \mathbb{E}\left(g(\omega, x)\right) \qquad (\text{let } g(\omega, x) = \cos(2\pi\omega \cdot x + b(\omega))) \end{aligned}$$

- Sampling:

$$\mathbb{E}(g(\omega, x)) \approx \frac{1}{n} \sum_{j=1}^{n} \cos(2\pi\omega_j \cdot x + b(\omega_j)) \in \Sigma_n^{cos}$$

Sampling argument

1. Let $v = \mathbb{E}\left(g(\omega, x)\right) = \int_G g(\omega, x)\lambda(x)dx$ and the sampling $v_n = \frac{1}{n}\sum_{i=1}^{n} g(\omega_i, x)$

$$v(x) - v_n(x) = \mathbb{E}g(x) - \frac{1}{n}\sum_{i=1}^{n} g(\omega_i, x); \tag{14}$$

2. By a direct calculation

$$\begin{aligned}
&\mathbb{E}_n\left(\left\|\mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(\omega_i, \cdot)\right\|^2\right) \\
&= \int_{\mathbb{R}^{d\times n}}\left(\left\|\mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(\omega_i, \cdot)\right\|^2\right)\lambda(\omega_1)\cdots\lambda(\omega_n)d\omega_1\cdots d\omega_n \\
&= \frac{1}{n}(\mathbb{E}(\|g\|^2) - [E(g)]^2) \le \frac{1}{n};
\end{aligned} \tag{15}$$

3. There exist $\{\omega_i^*\}_{i=1}^n$ such that

$$\|\mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(\omega_i^*, \cdot)\|_0^2 \le n^{-1}. \tag{16}$$

Approximation rates for Cosine networks

The preceding *sampling argument* gives the *approximation rate*:

Theorem

There exists $u_n \in \Sigma_{n,M}^{\cos} = \left\{ \sum_{i=1}^{n} a_i \cos(\omega_i \cdot x + b_i), \sum_{i=1}^{n} |a_i| \leq M \right\}$ *such that*

$$\|u - u_n\| \leq n^{-\frac{1}{2}} \|\hat{u}\|_{L^1(\mathbb{R}^d)}. \tag{17}$$

where $M = \|\hat{u}\|_{L^1(\mathbb{R}^d)}$.

From Cosine networks to ReLU^k networks

$$u(x) = \frac{1}{4}\left(\frac{\pi}{2}\right)^k \int_{\mathbb{R}^d}\int_{\mathbb{R}} \mathrm{ReLU}^k\left(\pi^{-1}\omega\cdot x + b\right)|\hat{u}(\omega)|e^{-i(\pi b + \frac{\pi(k+1)}{2} + \theta(\omega))}dbd\omega \tag{18}$$

Sampling arguments $\Rightarrow$ approximation result

Theorem (X 2020)

$$\inf_{u_n\in\Sigma_n^k}\|u-u_n\|_{H^m(\Omega)} \lesssim \begin{cases} n^{-\frac{1}{2}-\frac{1}{d}}\|u\|_{B^{k+1}(\Omega)}, & k>m, \\ n^{-\frac{1}{2}}\|u\|_{B^{k+1}(\Omega)}, & k=m, \end{cases} \tag{19}$$

where spectral Barron norm

$$\|u\|_{B^s(\Omega)} = \inf_{u_e|\Omega=u}\int_{\mathbb{R}^d}(1+|\omega|)^s|\hat{u}_e(\omega)|d\omega. \tag{20}$$

Outline

Perspective: Dictionary Approximation

$\mathbb{D} \subset X$ for a Banach space X is a dictionary if

- $\mathbb{D}$ is bounded, i.e. $|\mathbb{D}| = \sup_{d \in \mathbb{D}} \|d\|_X < \infty$
- $\mathbb{D}$ is symmetric, i.e. $d \in \mathbb{D} \to -d \in \mathbb{D}$

Non-linear dictionary approximation:

$$\Sigma_n(\mathbb{D}) := \left\{ \sum_{i=1}^{n} a_i d_i, \ d_i \in \mathbb{D} \right\} \tag{21}$$

Stable dictionary approximation:

$$\Sigma_{n,M}(\mathbb{D}) := \left\{ \sum_{i=1}^{n} a_i d_i, \ d_i \in \mathbb{D}, \ \sum_{i=1}^{n} |a_i| \leq M \right\} \tag{22}$$

Ref: Siegel, J. W. & Xu, J. (2023)

Variation spaces

- Take

$$B_1(\mathbb{D}) := \overline{\mathrm{conv}(\mathbb{D})} = \overline{\left\{ \sum_{i=1}^{n} a_i d_i : \ \sum_{i=1}^{n} |a_i| \le 1, \ n \in \mathbb{N} \right\}} \tag{23}$$

- Define $\mathcal{V}_1(\mathbb{D})$-norm by

$$\|f\|_{\mathcal{V}_1(\mathbb{D})} := \inf\{r > 0 : \ f \in B_1(\mathbb{D})\} = \inf\left\{ \sum_{i=1}^{n} |a_i| : f = \sum_{i=1}^{n} a_i h_i \right\}.$$

  Clearly, the unit ball of $\mathcal{V}_1(\mathbb{D})$ is $B_1(\mathbb{D})$.
- $\left\{ f \in X : \ \|f\|_{\mathcal{V}_1(\mathbb{D})} < \infty \right\}$ is a Banach space
- Example: Suppose $X = \ell_2$, $\mathbb{D} = \{e_1, e_2, \dots\}$ be the set of classical basis. Then

$$\mathcal{V}_1(\mathbb{D}) = \overline{\left\{ (a_1, a_2, \dots, a_n, 0, \dots) : \ \sum_{j=1}^{n} |a_j| \le 1, \quad n \in \mathbb{N} \right\}} = \ell_1$$

Ref: DeVore (1998), Siegel & Xu (2023)

Neural Network Dictionaries with Activation Function

- What is the relationship with shallow neural networks?
- Given an activation function σ and domain $\Omega \subset \mathbb{R}^d$, consider the dictionary

$$\mathbb{D}_\sigma^d = \{\sigma(\omega \cdot x + b),\ \omega \in \mathbb{R}^d,\ b \in \mathbb{R}\} \subset L^p(\Omega) \tag{24}$$

  - For some σ, may need to restrict ω and b to ensure boundedness
- In this case

$$\Sigma_n(\mathbb{D}_\sigma^d) = \left\{ \sum_{i=1}^n a_i \sigma(\omega_i \cdot x + b_i) \right\} \tag{25}$$

  is exactly the set of shallow neural networks with width n
- Typical σ: ReLU^k activation functions.

ReLU^k Activation Function

- Consider the ReLU^k activation function

$$\sigma_k(x) = \begin{cases} 0 & x \leq 0 \\ x^k & x > 0. \end{cases} \tag{26}$$

- In this case, $\sigma_k(\omega \cdot x + b)$ is not uniformly bounded in $L^p(\Omega)$!
- Must restrict ω and b, so consider the dictionary

$$\mathbb{P}_k^d = \{\sigma_k(\omega \cdot x + b),\ \omega \in S^{d-1},\ b \in [-2, 2]\} \subset L^2(B_1^d)\}, \tag{27}$$

where B_1^d is the unit ball in $\mathbb{R}^d$.

$\mathcal{V}_1(\mathbb{P}_k^d)$ is the variation space corresponding to shallow ReLU^k networks

Integral Representations of $\|f\|_{\mathcal{V}_1(\mathbb{D})}$

- If $\mathbb{D} \subset X$ is dense, we have

$$\|f\|_{\mathcal{V}_1(\mathbb{D})} = \inf\left\{\sum_{i=1}^{n} |a_i| : f = \sum_{i=1}^{n} a_i h_i\right\}$$
$$= \inf\left\{\int_{\mathbb{D}} d|\mu| : \ f = \int_{\mathbb{D}} h d\mu\right\}.$$

- For ReLU^k neural network dictionaries, we can write

$$\|f\|_{\mathcal{V}_1(\mathbb{D}_\sigma^d)} = \inf_{\mu \in \mathcal{B}(\mathbb{S}^d \times [-2,2])} \left\{\int_{\mathbb{S}^d \times [-2,2]} d|\mu| : \ f = \int_{\mathbb{S}^d \times [-2,2]} \sigma(w \cdot x + b) d\mu(w, b)\right\},$$

where $\mathcal{B}(\mathbb{S}^d \times [-2, 2])$ is the set of Borel measures on $\mathbb{S}^d \times [-2, 2]$.

Ref: E, W. 2017 ($k = 1$) , Siegel, J. W. & Xu, J. 2023

Stable neural network and approximation rate

$$\Sigma_{n,M}^{\sigma} := \{\sum_{i=1}^{n} a_i h_i,\ h_i \in \mathbb{D}_\sigma, \sum_{i=1}^{n} |a_i| \leq M\}$$

Theorem (Siegel & X, 2021-2022)

A function $f \in L^2(\Omega)$ can be approximated at all, i.e.

$$\lim_{n\to\infty} \inf_{f_n \in \Sigma_{n,M}^{\sigma}} \|f - f_n\|_{L^2(\Omega)} = 0, \tag{28}$$

for some $M > 0$ with $\sigma \in L^\infty(\mathbb{R})$, if and only if $u \in \mathcal{V}_1(\mathbb{D}_\sigma)$.
Furthermore, with $M = C\|f\|_{\mathcal{V}_1(\mathbb{D}_\sigma)}$,

$$\inf_{f_n \in \Sigma_{n,M}^{\sigma}} \|f - f_n\|_{L^2(\Omega)} \lesssim n^{-\frac{1}{2}} \|f\|_{\mathcal{V}_1(\mathbb{D}_\sigma)}. \tag{29}$$

For $\sigma = \mathrm{ReLU}^k$ and $M = C\|f\|_{\mathcal{V}_1(\mathbb{P}_k^d)}$, we have

$$\inf_{f_n \in \Sigma_{n,M}^{\sigma}} \|f - f_n\|_{L^2(\Omega)} \lesssim n^{-\frac{1}{2} - \frac{2k+1}{2d}} \|f\|_{\mathcal{V}_1(\mathbb{P}_k^d)}. \tag{30}$$

- Rate is optimal (up to log factors) for *stable* approximation
- Holds more generally for any smoothly parameterizable dictionary $\mathbb{D}$
- Earlier results: Barron, A. R. (1993), Makovoz, Y.(1996), Klusowski, J. M. & Barron, A. R. (2018), E, W., Ma, C. & Wu, L. (2019), Xu, J. (2021), Siegel, J. W. & Xu J. (2021)

Proof of (29): sampling argument

(similar to Cosine-NN)

1. Let $f \in B_1(D)$, for any $\epsilon > 0$, there exist ρ_i, h_i with $i = 1, \ldots, N$, such that

$$\|f - g\|_H \le \epsilon, \quad \text{with} \quad g = \sum_{i=1}^{N} a_i h_i, \text{ and } \sum_{i=1}^{N} a_i = 1. \tag{31}$$

   Without loss of generality, assume $a_i \ge 0$.

2. For any $g_{i_1,\cdots,i_n}$, define

$$\mathbb{E}_n g_{i_1,\cdots,i_n} := \sum_{i_1,\cdots,i_n=1}^{N} g_{i_1,\cdots,i_n} \prod_{j=1}^{n} a_{i_j}$$

3. For $g_{i_1,\cdots,i_n} = \frac{1}{n}\sum_{j=1}^{n} h_{i_j}$,

$$\mathbb{E}_n \|g - g_{i_1,\cdots,i_n}\|_H^2 = \frac{1}{n}\left(\mathbb{E}(\|h\|_H^2) - (\mathbb{E}\|h\|_H)^2\right) \le \frac{1}{n}\mathbb{E}(\|h\|_H^2) \le \frac{1}{n}\|\mathbb{D}\|^2.$$

4. There exist $\{i_j^*\}$ such that

$$\|g - g_{i_1^*,\cdots,i_n^*}\|_H \le n^{-\frac{1}{2}}\|\mathbb{D}\|.$$

5. Let $g_n = \frac{1}{n}\sum_{j=1}^{n} h_{i_j^*}$. Then,

$$\|f - g_n\|_H \le \|f - g\|_H + \|g - g_n\|_H \le \epsilon + n^{-\frac{1}{2}}\|\mathbb{D}\|.$$

Proof of (30): smoothly parameterized dictionaries

- Let $U \subset \mathbb{R}^d$ be an open set and $f : U \to \mathbb{R}$. Let $s = k + \alpha$ ($k \geq 0$, $\alpha \in (0, 1]$). Recall

$$|f|_{Lip(s,L^\infty(U))} := \sup_{x \neq y \in U} \frac{|D^k f(x) - D^k f(y)|}{|x - y|^\alpha}. \tag{32}$$

Definition

The map $\mathcal{P} : U \to X$ is of smoothness class s if for any $\xi \in X^*$ we have, letting $f_\xi(x) = \langle \mathcal{P}(x), \xi \rangle$,

$$|f_\xi|_{Lip(s,L^\infty(U))} \leq C \|\xi\|_{X^*}. \tag{33}$$

- Extended to smooth manifolds via charts

Theorem (Siegel & X, 2022)

Let X be a type-2 Banach space. Suppose that $\mathbb{D}$ is a parameterized by a smooth compact d-dimensional manifold $\mathcal{M}$ with smoothness order s. Then there exists M and C such that for any $f \in B_1(\mathbb{D})$,

$$\inf_{f_n \in \Sigma_{n,M}(\mathbb{D})} \|f - f_n\|_X \leq C n^{-\frac{1}{2} - \frac{s}{d}}, \tag{34}$$

where M and C depend only upon $s, d, \mathcal{P}$ and the type-2 constant of X.

- For ReLU^k networks, i.e. $\mathbb{D} = \mathbb{P}_k^d$, we get the rate $n^{-\frac{1}{2} - \frac{2k+1}{2d}}$ in $L^2(\Omega)$.
- Previous best rate was $n^{-\frac{1}{2} - \frac{1}{d}}$ in $L^2(\Omega)$ when $k > 1$.

Outline

Approximation properties of classic FEM

Theorem

Assume that V_h^k is a finite element of degree k on quasi-uniform mesh $\{\mathcal{T}_h\}$ of $\mathcal{O}(N)$ elements. Assume u is sufficiently smooth and not piecewise polynomials, then we have

$$c(u)N^{-\frac{k}{d}} \leq \inf_{v_h \in V_h^k} \|u - v_h\|_{H^1(\Omega)} \leq C(u)N^{-\frac{k}{d}} = \mathcal{O}(h^k). \tag{35}$$

In general

$$\inf_{v_h \in V_h^k} \|u - v_h\|_{H_h^m(\Omega)} \eqsim N^{\frac{m-(k+1)}{d}} = \mathcal{O}(h^{k+1-m}). \tag{36}$$

Ref: Q. Lin, H. Xie and J. Xu (2014)

Proof ($k = 1$)

Let Π_2 be quadratic interpolation. For any linear FE $v_h \in V_h$,

$$\begin{aligned} |u|_2 = |u - v_h|_{2,h} &\le |u - \Pi_2 u|_{2,h} + |\Pi_2 u - v_h|_{2,h} \\ &\le C_1 h |u|_3 + C_2 h^{-1} \|\Pi_2 u - v_h\|_1 \\ &\le C_1 h |u|_3 + C_2 h^{-1} \|\Pi_2 u - u\|_1 + C_2 h^{-1} \|u - v_h\|_1 \\ &\le C_1 h |u|_3 + C_3 h^{-1} h^2 |u|_3 + C_2 h^{-1} \|u - v_h\|_1 \end{aligned} \tag{37}$$

namely

$$|u|_2 - (C_1 + C_3) h |u|_3 \le C_2 h^{-1} \|u - v_h\|_1 . \tag{38}$$

Noting that $v_h \in V_h$ is arbitrary, then for sufficiently small h, we have

$$\inf_{v_h \in V_h} \|u - v_h\|_1 \ge Ch = CN^{-\frac{1}{d}} \tag{39}$$

since $h = CN^{-\frac{1}{d}}$ for quasi-uniform mesh.

Better approximation rates of NNs

Theorem (Siegel & X, 2022)

For $k > m$,

$$\inf_{u_n \in \Sigma_n^k} \|u - u_n\|_{H^m} \lesssim n^{m-(k+1)} \log(n) \|u\|_{B^\alpha(\Omega)} \tag{40}$$

if $\alpha > (d+1)(k-m+1/2) + m + 1/2$ and spectral Barron norm

$$\|u\|_{B^\alpha(\Omega)} = \inf_{u_e|\Omega = u} \int_{\mathbb{R}^d} (1+|\omega|)^\alpha |\hat{u}_e(\omega)| d\omega. \tag{41}$$

- This estimate improves result of Barron ($\alpha = 2$)
- Shows that very high order approximation rates can be attained with sufficient smoothness

- Ref: Siegel and Xu 2022, Lin, Xie and Xu 2014.

Approximation properties: DNN versus AFEM

Adaptive FEM (L^2 error): [similar for N-term wavelets approximation]

$$\inf_{\dim V_N = N} \inf_{u_N \in V_N} |u - u_N|_{0,\Omega} \lesssim N^{-\frac{2}{d}} \|u\|_*$$

Shallow ReLU NN (L^2 error, i.e. $m = 1, k = 1$):

$$\inf_{u_N \in \Sigma_N^1} |u - u_N|_{0,\Omega} \lesssim N^{-2} \log(N) \|u\|_{B^\alpha(\Omega)}$$

Observation:

$$\inf_{v \in V_N^{\mathrm{FEM}}} \|u - v\| = \mathcal{O}(N^{-\frac{2}{d}}) \Longrightarrow \inf_{w \in \Sigma_N^1} \|u - w\| \approx \left\{ \inf_{v \in V_n^{\mathrm{FEM}}} \|u - v\| \right\}^d .$$

Should we be excited?

1. NN has SUPER-approximation property!
2. NN breaks curse-of-dimensionality?

Caution:
We should not get too excited by such a “dimension-independent” result!

Example: a network of 3 parameters

$$\Sigma_3^{coscos} = \left\{ C\cos(t\cos(\lfloor Kx \rfloor)),\ C, t, K \in \mathbb{R} \right\}, \tag{42}$$

$$\lfloor x \rfloor = \text{largest integer that is} \leq x. \tag{43}$$

Theorem

For any continuous function g on $[0,1]$ and any $\epsilon > 0$, there exist $C, t, K \in \mathbb{R}$ such that

$$\|g - f(\circ; C, t, K)\|_{L^\infty([0,1])} < \epsilon. \tag{44}$$

- This theorem means

$$\inf_{u_3 \in \Sigma_3^{coscos}} \|u - u_3\| = 0 = \mathcal{O}(3^{-\infty}). \tag{45}$$

- Three parameters suffice to capture any function arbitrarily accurately!
- Parameters must be large to obtain high accuracy
  - Number of parameters is not a priori useful notion
  - Cannot be specified with a fixed number of bits
  - Not *encodable*!
- Ref: Shen, Z., Yang, H. & Zhang, S. (2021)

Proof

1. Choose $C = \|g\|_{L^\infty([0,1])}$. We assume next that $\|g\|_{L^\infty([0,1])} \le 1$.
2. Choose $K \in \mathbb{N}$ sufficiently large such that

$$\max_{x \in \left[\frac{j}{K}, \frac{j+1}{K}\right)} \left| g(x) - g\left(\frac{j}{K}\right) \right| < \frac{\epsilon}{2}, \quad j = 0, 1, \ldots, K.$$

   We have $\cos(\lfloor Kx \rfloor) = \cos(j)$ for $x \in \left[\frac{j}{K}, \frac{j+1}{K}\right)$
3. The set $\{\cos 0, \cos 1, \ldots, \cos(K)\}$ is linearly independent over $\mathbb{Q}$ since $\cos 1$ is transcendental.
4. $\{t(\cos 0, \ldots, \cos(K)) : \ t \in \mathbb{R}\}$ is dense in $\mathbb{R}^{K+1}/(2\pi\mathbb{Z})^{K+1}$. Namely there exists some $t \in \mathbb{R}$ and $\boldsymbol{m} \in \mathbb{Z}^{K+1}$ such that

$$\|(t\cos 0, \ldots, t\cos(K)) + 2\pi\boldsymbol{m} - \boldsymbol{y}\|_{L^\infty([0,2\pi]^{K+1})} < \frac{\epsilon}{2}.$$

   for $\boldsymbol{y} = \left(\arccos\left(g\left(\frac{0}{K}\right)\right), \ldots, \arccos\left(g\left(\frac{K}{K}\right)\right)\right)$.
5. Now for any $x \in [0,1]$, there exists some $0 \le j \le K$ such that $x \in \left[\frac{j}{K}, \frac{j+1}{K}\right)$. Thus

$$\begin{aligned}
|f(x;1,t,K) - g(x)| &= \left| f(x;1,t,K) - g\left(\frac{j}{K}\right) \right| + \left| g\left(\frac{j}{K}\right) - g(x) \right| \\
&\le \left| \cos(t\cos(j)) - g\left(\frac{j}{K}\right) \right| = \left| \cos(t\cos(j) + 2\pi m_j) - \cos\left(\arccos\left(g\left(\frac{j}{K}\right)\right)\right) \right| + \frac{\epsilon}{2} \\
&\le \left| t\cos(j) + 2\pi m_j - \arccos\left(g\left(\frac{j}{K}\right)\right) \right| + \frac{\epsilon}{2} < \epsilon.
\end{aligned}$$

6. $$\|f(\circ;1,t,K) - g\|_{L^\infty([0,1])} < \epsilon.$$

Outline

Encodability: metric entropy

Definition (Kolmogorov)

Let X be a Banach space and $B \subset X$. The metric entropy numbers of B, $\epsilon_n(B)_X$ are given by

$$\epsilon_n(B)_X = \inf\{\epsilon : \ B \text{ is covered by } 2^n \text{ balls of radius } \epsilon\}. \tag{46}$$

Example: The cube $[0,1]^d$ can be covered by the class of 2nd ℓ_∞-balls of radius $2^{-(n+1)}$:

$$\left\{ \bigotimes_{j=1}^{d} \left[\frac{k_j}{2^n}, \frac{k_j+1}{2^n} \right] : \ k_j = 0, 1, \ldots, 2^n - 1, \ j = 1, \ldots, d \right\},$$

but it cannot be covered by 2nd balls with any smaller radius. So the metric entropy of $[0,1]^d$ with respect to ℓ_∞-norm is $\epsilon_{nd}([0,1]^d) = 2^{-(n+1)}$. This gives

$$\epsilon_n([0,1]^d) \simeq 2^{-\frac{n}{d}}.$$

Metric entropy and dimension

High-dimensional problems are always harder to solve?

- High-dimensional sets does not always have larger entropy. From definition, we can observe $\epsilon_n(rB)_X = r\epsilon_n(B)_X$. So if we take $r = 2^{\frac{n}{d}-(n+1)}$, then

$$\epsilon_n([0,r]^d) \lesssim 2^{\frac{n}{d}-(n+1)} 2^{-\frac{n}{d}} = 2^{-(n+1)} = \epsilon_n([0,1]).$$

- The scale of the set should also be considered.

Some classical properties of metric entropy

- $\epsilon_n(B)_X$ measures how accurately elements of B can be specified with n bits
- Gives fundamental limit for any (digital) numerical algorithm
- Curse of dimensionality: for unit ball $B_1(W^{s,p}(\Omega))$ in Sobolev space $W^{s,p}(\Omega)$,

$$\epsilon_n(B_1(W^{s,p}(\Omega)))_{L^p(\Omega)} \simeq n^{-\frac{s}{d}}$$

- In high dimensions, we need novel function classes with small metric entropy!

- Ref: Birman and Solomyak (1967), Narcowich and Ward (2004), Cohen, Devore, Petrova, and Wojtaszczyk (2021)

No curse of dimensionality: polynomial & kernel

Theorem

$$\inf_{u_n \in P_n} \|u - u_n\| \eqsim n^{-\frac{s}{d}} \|u\|_{H^s(\Omega)}, \tag{47}$$

where $\Omega = [0,1]^d$, $u \in H^s(\Omega)$, P_n *is the space of polynomials on* Ω *with* n *degree of freedom.*

Theorem

Let Q *be a Guassian kernel and* $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$ *be appropriately distributed, for any* $s > \frac{d}{2}$ *we have*

$$\inf_{u_n \in Q_n} \|u - u_n\| \lesssim n^{-\frac{s}{d}} \|u\|_{H^s}, \text{ where } Q_n = span\{Q(x, x_i)\}_{i=1}^n \tag{48}$$

No curse of dimensionality in both cases for sufficiently smooth functions:

$$\inf_{u_n} \|u - u_n\| \lesssim n^{-\frac{1}{2}} \|u\|_{H^{d/2}}. \tag{49}$$

- Ref: DeVore, R. A., & Lorentz, G. G. (1993), Arcangéli, R., López de Silanes, M. C., & Torrens, J. J. (2007), Narcowich. F. J, Ward. J. D., and Wendland. H (2006); Batlle, P., Chen, Y., Hosseini, B., Owhadi, H., & Stuart, A. M. (2023).

Metric Entropy of Dictionary Spaces

What are the metric entropies of the unit ball in $\mathcal{V}_1(\mathbb{P}_k^d)$?

Theorem (Siegel & X 2022)

The metric entropies of $\mathbb{P}_k^d$ and $\mathbb{F}_s^d$ satisfy

$$\epsilon_n(B_1(\mathbb{P}_k^d)) \eqsim n^{-\frac{1}{2}-\frac{2k+1}{2d}}, \ \epsilon_n(B_1(\mathbb{F}_s^d)) \eqsim n^{-\frac{1}{2}-\frac{s}{d}} \tag{50}$$

No curse of dimensionality (in terms of metric entropy)!

Ref: J. Siegel & J. Xu 2022

Outline

Sharpness of AFEM estimate

Example: $u = |x|^2$

$$N^{-\frac{1}{d}} \lesssim \inf_{\dim V_N = N} \inf_{u_N \in V_N} |u - u_N|_{1,2,\Omega} \lesssim N^{-\frac{1}{d}}$$

$$N^{-\frac{2}{d}} \lesssim \inf_{\dim V_N = N} \inf_{u_N \in V_N} \|u - u_N\|_{0,\infty,\Omega} \lesssim N^{-\frac{2}{d}}$$

optimal grid: uniform grid.

d=1

- Grid $\mathcal{T} := \{0 = x_0 < x_1 < \cdots < x_{N+1} = 1\}$
- Local basis

$$\phi_i = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in (x_{i-1}, x_i], \\ \frac{x - x_{i+1}}{x_i - x_{i+1}}, & x \in (x_i, x_{i+1}], \\ 0, & \text{others}. \end{cases}.$$

- Interpolation: $I_{\mathcal{T}} u = \sum_i u_{x_i} \phi_i$.

Approximation of x^2 by multilevel FEM

Let $f(x) = x^2$ on $[0,1] \subset \mathbb{R}$. Recall

$$(I - I_J)f = \mathcal{O}(N^{-2})$$

with

$$I_J f = \sum_{i=0}^{2^J} f(x_i)\phi_i^J(x).$$

We write

$$I_J f = I_0 f + \sum_{k=1}^{J} (I_k - I_{k-1})f$$

Notice that $I_0 f = x$ and

$$(I_k - I_{k-1})f = -4^{-k} \sum_{j=1}^{2^{k-1}} \phi_{2j-1}^k(x).$$

Non-local basis: $g_i = \sum_{j=1}^{2^{i-1}} \phi_{2j-1}^i(x)$

$$\|x^2 - (x - \sum_{i=1}^{L} 4^{-i} g_i)\|_{0,\infty,\Omega} = \mathcal{O}(N^{-2}), \tag{51}$$

with $L = \log_2 N$!

- Ref: D.Yarotsky 2017, W. E and Q. Wang 2018, J. He and J. Xu 2019.

Diagram for basis functions g_i

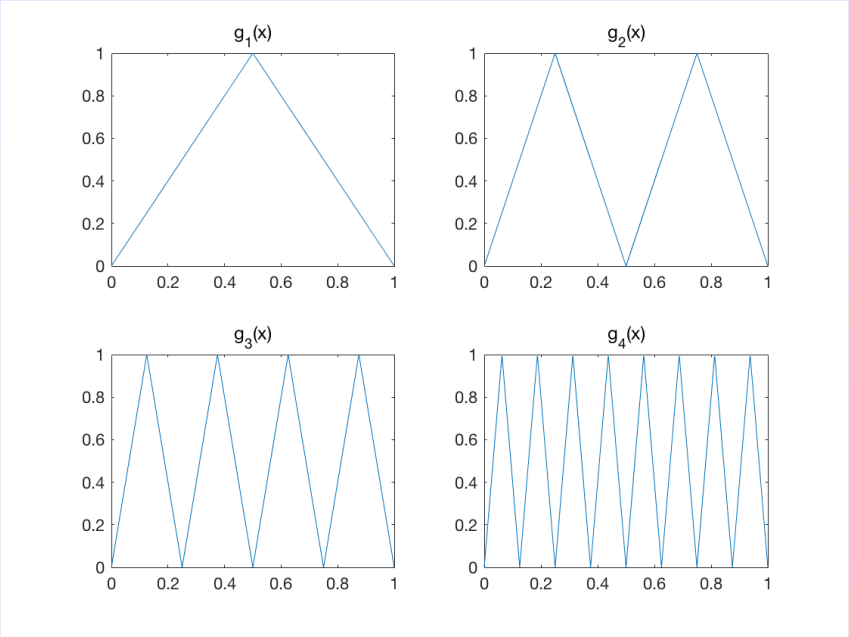


Figure: The figure of $g_i(x)$

Important observation: composition property:

$$g_i(x) = \underbrace{g_1 \circ g_1 \circ \cdots \circ g_1}_{i}(x) \in \Sigma^1_{n_{1:i}}. \tag{52}$$

Comparison between DNN and FEM on x^2

Recall: Approximating x^2 by AFEM

Theorem

$$\inf_{v(x)\in V_L} \max_{x\in[0,1]} |x^2 - v(x)| = \mathcal{O}(L^{-2}), \tag{53}$$

where V_L denotes the AFEM space with L elements on $[0,1]$.

Approximating x^2 by ReLU DNNs

Theorem

There exists $v(x) \in \Sigma^1_{n_{1:L}}$, such that

$$\max_{x\in[0,1]} |x^2 - v(x)| = \mathcal{O}(4^{-L}), \tag{54}$$

where $n_\ell \leq 4$ for $\ell = 1 : L$.

Ref: D.Yarotsky 2017, W. E and Q. Wang 2018, J. He, Lin Li and J. Xu 2022.

From x^2 to polynomials and smooth functions

We note that

$$g_i \in \Sigma^1_{n_{1:i}}, \quad \text{where} \quad n_j \le 4, j = 1:i.$$

Using the following identity

$$xy = 2\left(\left(\frac{x+y}{2}\right)^2 - \left(\frac{x}{2}\right)^2 - \left(\frac{y}{2}\right)^2\right)$$

Theorem

For any polynomial $p(x) = \sum_{|\boldsymbol{k}|\le p} a_{\boldsymbol{k}} \boldsymbol{x}^{\boldsymbol{k}}$ with order p on $[0,1]^d$, there exists $v(x) \in \Sigma^1_{n_{1:CL}}$, such that

$$\max_{\boldsymbol{x}\in[0,1]^d} |p(\boldsymbol{x}) - v(\boldsymbol{x})| \le 4(p-1)4^{-L} \sum_{\boldsymbol{k}} |a_{\boldsymbol{k}}|, \tag{55}$$

where $n_\ell \le 2d+4$ and $C = 3(p-1)\binom{p+d}{d}$ that is independent from L.

Corollary

ReLU DNN can approximate smooth functions as good as polynomials !

Ref: D.Yarotsky 2017, W. E and Q. Wang 2018, J. He, L. Li and J. Xu 2022, J. He and J. Xu 2023.

DNN versus FEM

FEM

1. Local basis functions
2. Multilevel basis functions give more "global" basis functions:
   - Sparse grid
   - Combination of hierarchical basis functions

DNN

1. It contains all AFEM
2. It generates "global" basis functions (from composition)
   - "Features" are mostly "global"?
3. Nonlinearity function generates redundant number of basis functions:

Lemma (He, Lin, X and Zheng 2018, Siegel and X 2019)

$\{\sigma(w_i \cdot x + b_i)\}_{i=1}^N$ are linearly independent if σ is not polynomials and $\begin{pmatrix} w_i \\ b_i \end{pmatrix}, \begin{pmatrix} w_j \\ b_j \end{pmatrix} \in \mathbb{R}^{d+1}$ are linear independent for any $i \neq j$.

Outline

Concluding remarks

Summary

- Finite elements $\Rightarrow$ neural networks
- ReLU and ReLU^k networks
  - ReLU DNN $\equiv$ linear finite elements (local adaptivity)
  - ReLU^k-DNN can recover all polynomials (global)
- Approximation properties
  - Variation spaces
  - Optimal rates
- Metric entropy
- Deep ReLU NNs and hierarchical basis

Thank You !