

Finite Neuron Method

Finite Neuron Method

Jinchao Xu

KAUST and Penn State

xu@multigrid.org

CIRM, July 18th, 2023

CEMRACS 2023 Summer School

KAUST Baseline Research Fund

- 1 Model problem
- 2 Stable shallow neural network functions and approximation
- 3 Error estimates for numerical quadrature of stable NN approximation
- 4 Optimization algorithms
- 5 Numerical examples
- 6 Summary

A Model PDE

Given $\Omega \subset \mathbb{R}^d$, consider a $2m$ -th order elliptic problems

$$\sum_{|\alpha|=m} (-1)^m \partial^\alpha (a_\alpha(x) \partial^\alpha u) + u = f \quad \text{in } \Omega. \quad (1)$$

Special cases:

$$-\Delta u = f \quad (m=1), \quad \Delta^2 u = f \quad (m=2).$$

The Dirichlet boundary condition for the $2m$ -th order elliptic problems are

$$B_D^k(u) = 0 \text{ on } \partial\Omega, \quad (0 \leq k \leq m-1),$$

where B_D^k are given by the following Dirichlet type trace operators

$$B_D^k(u) := \left. \frac{\partial^k u}{\partial \nu^k} \right|_{\partial\Omega} \quad (0 \leq k \leq m-1). \quad (2)$$

Dirichlet vs Neumann boundary condition

Lemma

For each $k = 0, 1, \dots, m - 1$, there exists a bounded linear differential operator of order $2m - k - 1$:

$$B_N^k : H^{2m}(\Omega) \mapsto L^2(\partial\Omega) \quad (3)$$

such that the following identity holds

$$\sum_{|\alpha|=m} (-1)^m (\partial^\alpha (a_\alpha \partial^\alpha u), v)_{0,\Omega} = \sum_{|\alpha|=m} (a_\alpha \partial^\alpha u, \partial^\alpha v)_{0,\Omega} - \sum_{k=0}^{m-1} \langle B_N^k(u), B_D^k(v) \rangle_{0,\partial\Omega} \quad (4)$$

for all $u \in H^{2m}(\Omega)$, $v \in H^m(\Omega)$. Furthermore,

$$\sum_{k=0}^{m-1} \|B_D^k(u)\|_{L^2(\partial\Omega)} + \sum_{k=0}^{m-1} \|B_N^k(u)\|_{L^2(\partial\Omega)} \lesssim \|u\|_{2m,\Omega}. \quad (5)$$

Neumann problem: Consider

$$\begin{cases} \sum_{|\alpha|=m} (-1)^m \partial^\alpha (a_\alpha(x) \partial^\alpha u) + u = f & \text{in } \Omega, \\ B_N^k(u) = 0 \quad \text{on } \partial\Omega, \quad 0 \leq k \leq m-1. \end{cases} \quad (6)$$

Dirichlet and Neumann problems

- Find $u \in V$ such that

$$J(u) = \min_{v \in V} J(v) \quad (7)$$

where

$$J(v) = \frac{1}{2} \int_{\Omega} \sum_{|\alpha|=m} a_{\alpha} |\partial^{\alpha} v|^2 + v^2 dx - \int_{\Omega} fv dx. \quad (8)$$

Then

$$V = \begin{cases} H^m(\Omega) & \iff \text{Neumann problem} \\ H_0^m(\Omega) & \iff \text{Dirichlet problem} \end{cases}$$

On the construction of smooth FEM

Question: For any $m, d \geq 1$, how to construct conforming finite element space

$$V_h \subset H^m(\Omega) \iff V_h \subset C^{m-1}(\Omega)?$$

Refs: Argyris et al., (1968); Bramble & Zlámal, (1970); Zhang (2009); Hu & Zhang (2015); Fu, Guzmán & Neilan (2020).

Answer: mostly **open**, especially when $m \geq 3, d \geq 3$ until recently (2021).

Theorem (Hu, Lin, & Wu 2021, ArXiv: 2103.14924))

For any $d \geq 1, r \geq 0$, a globally C^r finite element of degree $k \geq 2^d r + 1$ can be constructed on any simplicial mesh with locally defined DOF.

- 1 Model problem
- 2 Stable shallow neural network functions and approximation
- 3 Error estimates for numerical quadrature of stable NN approximation
- 4 Optimization algorithms
- 5 Numerical examples
- 6 Summary

Shallow Neural Networks

- A shallow neural network function: with n neurons

$$f(x) = \sum_{i=1}^n a_i \sigma(\omega_i \cdot x + b_i) \quad (9)$$

- Common activation functions:

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

Approximation properties

General Shallow neural network

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

Preliminary approximation results

$$\inf_{u_n \in \Sigma_n^\sigma} \|u - u_n\|_{L^2(\Omega)} = \mathcal{O}\left(n^{-\frac{1}{2}}\right)$$

(Jones 1992, Barron 1993, Hornik et al 1994, ...)

Optimal rate for stable shallow ReLU^k NNs (Siegel & Xu 2021):

$$\inf_{u_n \in \Sigma_{n,M}^k} \|u - u_n\|_{H^m} = \mathcal{O}\left(n^{-\frac{1}{2} - \frac{2(k-m)+1}{2d}}\right) \quad (10)$$

Application of ReLU^k-DNN for high order PDE in any dimension

Finite neuron method: We take

$$V_n = \Sigma_n^\sigma \subset H^m(\Omega)$$

Neumann Problem: We define DNN-Galerkin approximation $u_n \in V_n$ as follows:

$$J(u_n) = \min_{v \in V_n} J(v). \quad (11)$$

Theorem (X 2020)

Let $u \in V$ and $u_n \in V_n$ be solutions to (7) with $V = H^m(\Omega)$ and (11) respectively. Then

$$\|u - u_n\|_a = \inf_{v_n \in V_n} \|u - v_n\|_a. \quad (12)$$

Error estimates for NN-PDE

For Neumann boundary condition,

$$\|u - u_n\|_{m,\Omega} \lesssim \inf_{v_n \in V_n} \|u - v_n\|_{m,\Omega} = \mathcal{O}(n^{-\frac{1}{2}-\alpha})$$

For Dirichlet problem, similar but less sharp estimates hold.

Issues:

- Discretization of the integral in $J(v)$, i.e. how do we evaluate

$$\frac{1}{2} \int_{\Omega} \sum_{|\alpha|=m} a_{\alpha} |\partial^{\alpha} v|^2 + v^2 dx - \int_{\Omega} f v dx? \quad (13)$$

- How to analyze the convergence when numerical quadratures are used?

Convergence analysis when numerical quadratures are used? Existing works:

- M. Hutzenthaler, A. Jentzen, T. Kruse, T. A. Nguyen, and P. Wurstemberger, 2020;
- T. Luo and H. Yang, 2020;
- S. Mishra and T. K. Rusch, 2020; S. Mishra and R. Molinaro, 2020;
- J. Müller and M. Zeinhofer, 2020;
- Y. Shin, Z. Zhang, and G.E. Karniadakis, 2020;
- S. Lanthaler, S. Mishra, G.E. Karniadakis, 2021;
- J. Lu, Y. Lu and M. Wang, 2021;
- H. Son, J. Jang, W. Han, and H. J. Hwang, 2021;

Numerical Quadratures

Given NN:

$$\Sigma_n^k = \left\{ \sum_{i=1}^n a_i \sigma_k(\omega_i \cdot x + b_i) : \omega_i \in \mathbb{R}^d, a_i, b_i \in \mathbb{R} \right\}.$$

Numerical Quadrature:

$$\int_{\Omega} g(x) dx \approx |\Omega| \sum_{i=1}^N w_i g(x_i) \quad (14)$$

Consider

$$J(u) = \min J(v), \quad J_N(u_{n,N}) = \min J_N(v_n),$$

where $J_N(v)$ is computed by deterministic numerical quadrature

$$J_N(v) = \sum_{i=1}^N w_i \left(\frac{1}{2} \sum_{|\alpha|=m} a_{\alpha}(x_i) (\partial^{\alpha} v(x_i))^2 + \frac{1}{2} a_0(x_i) (v(x_i))^2 - f(x_i) v(x_i) \right) \quad (15)$$

or by sampling

$$J_N(v) = \frac{1}{2N} \sum_{i=1}^N \sum_{|\alpha|=m} a_{\alpha}(x_i) (\partial^{\alpha} v(x_i))^2 + \frac{1}{2N} \sum_{i=1}^N a_0(x_i) v(x_i)^2 - \frac{1}{N} \sum_{i=1}^N f(x_i) v(x_i). \quad (16)$$

Our objective

Estimate the error $u - u_{n,N}$

Error analysis

Numerical quadrature: for any $g(x)$,

$$\left| \int_{\Omega} g(x) dx - |\Omega| \sum_{i=1}^N w_i g(x_i) \right| \lesssim N^{-\frac{r+1}{d}} \|g\|_{r,1}.$$

Challenges: how to bound

$$\|g\|_{r,1} \leq ?, \quad \text{for } g \in \Sigma_n^\sigma.$$

OK if the following Bernstein or inverse inequality holds for $r > s$

$$\|v_n\|_r \lesssim n^\beta \|v_n\|_s, \quad \forall v_n \in \Sigma_n^k. \tag{17}$$

Bad news: Bernstein inequality does not hold for NN

Given any $\epsilon > 0$, consider an NN function with 3 neurons:

$$u_3(x) = \text{ReLU}(x - \frac{1}{2} + \epsilon) - 2\text{ReLU}(x - \frac{1}{2}) + \text{ReLU}(x - \frac{1}{2} - \epsilon), \quad \forall x \in (0, 1).$$

A direct calculation shows that

$$\int_0^1 |u'_3(x)|^2 dx = 2\epsilon \quad \text{and} \quad \int_0^1 |u_3(x)|^2 dx = \epsilon^2.$$

Therefore

$$|u_3|_{H^1} = \sqrt{\frac{2}{\epsilon}} \|u_3\|_{L^2}, \quad \forall \epsilon > 0$$

As a result, the following Bernstein inequality **can not hold** for any constant $C(n)$

$$|\nu_n|_{H^1} \leq C(n) \|\nu_n\|_{L^2}, \quad \forall \nu_n \in \Sigma_n^\sigma$$

Our approach: Stable neural network

State neural networks:

$$\Sigma_{n,M}^\sigma := \left\{ \sum_{i=1}^n a_i \sigma(\omega_i \cdot x + b_i), \omega_i \in \mathbb{S}^{d-1}, |b_i| \leq B, \sum_{i=1}^n |a_i| \leq M \right\} \quad (18)$$

Theorem (Siegel & X 2021)

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

$$\lim_{n \rightarrow \infty} \inf_{f_n \in \Sigma_{n,M}(\mathbb{D})} \|f - f_n\|_H = 0, \quad (19)$$

for a fixed $M < \infty$ if and only if

$$f \in \mathcal{V}_1(\mathbb{D}) \text{ with } \|f\|_{\mathcal{V}_1(\mathbb{D})} \leq M.$$

Furthermore, if $\mathbb{D} = \mathbb{P}_k^d$ (i.e. ReLU^k) and $M \leq C \|u\|_{\mathcal{V}_1(\mathbb{P}_k^d)}$ we have

$$\inf_{u_n \in \Sigma_{n,M}^k} \|u - u_n\|_{H^m} \lesssim n^{-\frac{1}{2} - \frac{2(k-m)+1}{2d}} \|u\|_{\mathcal{V}_1(\mathbb{P}_k^d)}. \quad (20)$$

- 1 Model problem
- 2 Stable shallow neural network functions and approximation
- 3 Error estimates for numerical quadrature of stable NN approximation
- 4 Optimization algorithms
- 5 Numerical examples
- 6 Summary

Error estimates for deterministic numerical quadrature

Consider

$$\Sigma_{n,M}^k = \left\{ \sum_{i=1}^n a_i \sigma_k(\omega_i \cdot x + b_i) : \omega_i \in \mathbb{S}^{d-1}, |b_i| \leq B, \sum_{i=1}^n |a_i| \leq M \right\} \subset W^{k,\infty}(\Omega)$$

with

$$M \geq \|u\|_{\mathcal{V}_1(\mathbb{P}_k^d)}.$$

Theorem (Hong, Siegel & X 2021)

Let N be the number of quadrature points and

$$J_N(u_{n,N}) = \min_{v \in \Sigma_{n,M}^k} J_N(v). \quad (21)$$

If $N = \mathcal{O}(n^{\frac{d+1+2(k-m)}{k-1}})$, it holds that

$$\|u_{n,N} - u\|_a \lesssim n^{-\frac{1}{2} - \frac{2(k-m)+1}{2d}} \|u\|_{\mathcal{V}_1(\mathbb{P}_k^d)}.$$

Sketch of proof

- ① For any conforming v ,

$$|J_N(v) - J(v)| \leq N^{-\frac{k-1}{d}} \|v\|_{k,\infty}. \quad (22)$$

- ② Since Ω is bounded, $\omega_i \in \mathbb{S}^{d-1}$, $|b_i| \leq B$, $\sum_{i=1}^n |a_i| \leq M$,
- $$\|u_{n,N}\|_{k,\infty} \lesssim M.$$

- ③ For any n , there exists $v_n \in \Sigma_{n,M}^k$

$$\|u - v_n\|_a \lesssim n^{-\frac{1}{2} - \frac{2(k-m)+1}{2d}} \|u\|_{\mathcal{V}_1(\mathbb{P}_k^d)}. \quad (23)$$

- ④ Since $J_N(u_{n,N}) \leq J_N(v_n)$,

$$\begin{aligned} \frac{1}{2} \|u_{n,N} - u\|_a^2 &= J(u_{n,N}) - J(u) \leq J(u_{n,N}) - J_N(u_{n,N}) + J_N(u_{n,N}) - J(v_n) + J(v_n) - J(u) \\ &\lesssim N^{-\frac{k-1}{d}} + \|v_n - u\|_a^2 \lesssim N^{-\frac{k-1}{d}} + n^{-1 - \frac{2(k-m)+1}{d}}. \end{aligned}$$

- ⑤ Choose $N = \mathcal{O}(n^{\frac{d+1+2(k-m)}{k-1}})$, then

$$\|u_{n,N} - u\|_a \lesssim n^{-\frac{1}{2} - \frac{2(k-m)+1}{2d}}.$$

Error estimate for numerical quadrature by sampling

Quadrature by sampling: Consider

$$J_N(u_{n,N}) = \min_{v \in \Sigma_{n,M}^k} J_N(v),$$

with sampling

$$J_N(v) = \frac{1}{2N} \sum_{i=1}^N \sum_{|\alpha|=m} a_\alpha(x_i) (\partial^\alpha v(x_i))^2 + \frac{1}{2N} \sum_{i=1}^N a_0(x_i) v(x_i)^2 - \frac{1}{N} \sum_{i=1}^N f(x_i) v(x_i). \quad (24)$$

Rademacher complexity: The Rademacher complexity of function class \mathcal{F} on Ω is given by

$$R_N(\mathcal{F}) = \mathbb{E}_{x_1, \dots, x_N} \mathbb{E}_{\xi_1, \dots, \xi_N} \left(\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N \xi_i f(x_i) \right), \quad (25)$$

where x_i are drawn uniformly at random from Ω and ξ_i are uniformly random signs.

Properties of Rademacher complexity

Lemma

Let \mathcal{F}, \mathcal{S} be classes of functions on Ω . Then the following bounds hold.

- $R_N(\text{conv}(\mathcal{F})) = R_N(\mathcal{F})$.
- Define the set $\mathcal{F} + \mathcal{S} = \{h(x) + g(x) : h \in \mathcal{F}, g \in \mathcal{S}\}$. We have

$$R_N(\mathcal{F} + \mathcal{S}) = R_N(\mathcal{F}) + R_N(\mathcal{S}). \quad (26)$$

- Suppose that $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is L -Lipschitz. Let $\phi \circ \mathcal{F} = \{\phi(h(x)) : h \in \mathcal{F}\}$. Then

$$R_N(\phi \circ \mathcal{F}) \leq L R_N(\mathcal{F}). \quad (27)$$

- Suppose that $f : \Omega \rightarrow \mathbb{R}$ is a fixed function. Let $f \cdot \mathcal{F} = \{f(x)h(x) : h \in \mathcal{F}\}$. Then

$$R_N(f \cdot \mathcal{F}) \leq \|f(x)\|_{L^\infty(\Omega)} R_N(\mathcal{F}). \quad (28)$$

Bound by Rademacher complexity

Theorem (Bartlett, 2002)

Let \mathcal{F} be a set of functions. Then

$$\mathbb{E}_{x_1, \dots, x_N \sim \mu} \sup_{h \in \mathcal{F}} \left| \frac{1}{N} \sum_{i=1}^N h(x_i) - \int_{\Omega} h(x) d\mu \right| \leq 2R_N(\mathcal{F}). \quad (29)$$

Bound the error of numerical quadrature by Rademacher complexity

Our function class:

$$\mathcal{F}_{n,M} = \left\{ \frac{1}{2} \sum_{|\alpha|=m} a_\alpha(x) (\partial^\alpha u(x))^2 + \frac{1}{2} u(x)^2 - f(x)u(x) : u \in \Sigma_{n,M}(\mathbb{D}) \right\}. \quad (30)$$

Theorem (Hong, Siegel & X 2021)

Suppose that $\|a_\alpha\|_{L^\infty(\Omega)}, \|a_0\|_{L^\infty(\Omega)} \leq K$ and $\sup_{d \in \mathbb{D}} \|d\|_{W^{m,\infty}} \leq C$. Then the Rademacher complexity of the set $\mathcal{F}_{n,M}$ in (30) is bounded by

$$R_N(\mathcal{F}_{n,M}) \leq CKM \sum_{|\alpha|=m} R_N(\partial^\alpha \mathbb{D}) + CKMR_N(\mathbb{D}) + \|f\|_{L^\infty(\Omega)} MR_N(\mathbb{D}), \quad (31)$$

where $\mathbb{D} = \{\sigma_k(\omega \cdot x + b) : \omega \in \mathbb{S}^{d-1}, |b| \leq B\}$ and $\partial^\alpha \mathbb{D} = \{\partial^\alpha d : d \in \mathbb{D}\}$.

Bound the Rademacher complexity

Theorem (Hong, Siegel & X 2021)

Suppose that $\sigma \in W^{m+1, \infty}$. Then for any α with $|\alpha| \leq m$, we have

$$R_N(\mathbb{D}) \lesssim N^{-\frac{1}{2}}, \quad R_N(\partial^\alpha \mathbb{D}) \lesssim N^{-\frac{1}{2}}. \quad (32)$$

where the implied constant is independent of N .

Sketch of Proof

- 1 Noting that

$$\partial^\alpha \mathbb{D} = \{\omega^\alpha \sigma^{(\alpha)}(\omega \cdot x + b) : \omega \in \mathbb{S}^{d-1}, |b| \leq B\}. \quad (33)$$

hence $|\omega^\alpha|$ is bounded.

- 2 Since $\sigma \in W^{m+1, \infty}$, we have that $\sigma^{(\alpha)}$ is Lipschitz. We obtained

$$R_N(\partial^\alpha \mathbb{D}) \lesssim R_N(\{\omega \cdot x + b : \omega \in \mathbb{S}^{d-1}, |b| \leq B\}). \quad (34)$$

- 3 It is well-known that the Rademacher complexity of the set of linear functions is bounded by

$$R_N(\{\omega \cdot x + b : \omega \in \mathbb{S}^{d-1}, |b| \leq B\}) \lesssim N^{-\frac{1}{2}}, \quad (35)$$

for any distribution on x which is bounded almost surely.

Convergence Analysis Assuming Perfect Optimization

- Suppose we can solve perfectly the optimization problem

$$J_N(u_{n,N}) = \min_{v \in \Sigma_{n,M}^k} J_N(v) \quad (36)$$

- Combining the uniform error bound with optimal approximation rates for ReLU^k networks, we get

Theorem

Suppose that $\|a_\alpha\|_{L^\infty(\Omega)}, \|a_0\|_{L^\infty(\Omega)} \leq K$ and $\sup_{d \in \mathbb{D}} \|d\|_{W^{m,\infty}} \leq C$. Let $\mathbb{D} = \mathbb{P}_k^d$ be the dictionary of ReLU^k ridge functions. Then we have

$$\mathbb{E}_{x_1, \dots, x_N} \left[\|u_{n,N} - u\|_{H^m(\Omega)}^2 \right] \lesssim MN^{-\frac{1}{2}} + Mn^{-1 - \frac{2(k-m)}{d}}. \quad (37)$$

- 1 Model problem
- 2 Stable shallow neural network functions and approximation
- 3 Error estimates for numerical quadrature of stable NN approximation
- 4 Optimization algorithms
- 5 Numerical examples
- 6 Summary

Another Issue

How can we efficiently solve the optimization of the discrete energy, i.e.

$$J_N(u_{n,N}) = \min_{v \in \Sigma_{n,M}^k} J_N(v).$$

Challenge: SGD or Adam

For the ReLU^k shallow neural network, let

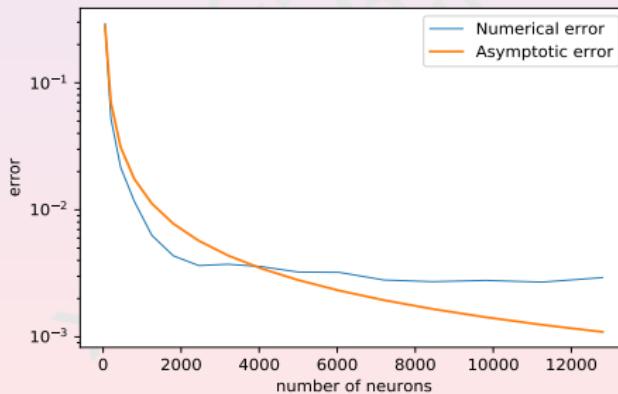
$$u_n = \arg \min_{v_n \in \Sigma_{n,M}(\mathbb{D})} \|v_n - u\| \quad (38)$$

be the solution trained by SGD or Adam, then it is extremely difficult to observe

$$\|u - u_n\| \leq cn^{-\alpha} \quad (39)$$

for any $\alpha > 0$.

Non-convergence for Adam etc for large n



Our approach: Greedy Algorithm

- Orthogonal greedy algorithm (OGA):

$$u_0 = 0, \quad g_n = \arg \max_{g \in \mathbb{D}} |\langle u - u_{n-1}, g \rangle_H|, \quad u_n = P_n u, \quad (40)$$

where P_n : orthogonal projection onto $\text{span}\{g_1, \dots, g_n\}$.

- Relaxed greedy algorithm (RGA):

$$\begin{aligned} u_0 &= 0 \\ g_n &= \arg \max_{g \in \mathbb{D}} \langle \nabla J_N(u_{n-1}), g \rangle \\ u_n &= (1 - s_n)u_{n-1} - Ms_n g_n. \end{aligned} \quad (41)$$

A Property of RGA:

Theorem (Hong, Siegel & X 2021)

The $u_n, n = 1, 2, \dots$, produced by RGA satisfy $\|u_n\|_{\mathcal{V}_1(\mathbb{D})} \leq M$ for all n and

$$J_N(u_n) - \min_{v \in \Sigma_{n,M}} J_N(v) \lesssim \frac{1}{n}. \quad (42)$$

- Ref: DeVore and Temlyakov 1996

Relaxed Greedy Algorithm (Jones 1992)

- 1 Let $\|f\|_{\mathcal{K}_1(\mathbb{D})} \leq 1$ and consider the *relaxed greedy algorithm*

$$f_1 = 0, h_n = \arg \max_{h \in \mathbb{D}} \langle f - f_{n-1}, h \rangle, f_n = \left(1 - \frac{1}{n}\right) f_{n-1} + \frac{1}{n} h_n \quad (43)$$

- ▶ Note that $f_n \in \Sigma_{n,1}(\mathbb{D})$

- 2 Claim: $\|f - f_n\| \leq 2|\mathbb{D}|n^{-\frac{1}{2}}$

- 3 Proof:

- ▶ We only need prove this for those $f \in B_1(\mathbb{D})$ that can be written as $f = \sum_{i=1}^n a_i g_i$, $a_i \geq 0$, $g_i \in \mathbb{D}$, $\sum_{i=1}^n a_i \leq 1$.
- ▶ Note that $\|f - f_n\|^2 = \left\| \left(1 - \frac{1}{n}\right) (f - f_{n-1}) + \frac{1}{n} (f - h_n) \right\|^2$, expand:

$$\|f - f_n\|^2 = \left(1 - \frac{1}{n}\right)^2 \|f - f_{n-1}\|^2 + \frac{2}{n} \left(1 - \frac{1}{n}\right)^2 \langle f - f_{n-1}, f - h_n \rangle + \frac{1}{n^2} \|f - h_n\|^2 \quad (44)$$

- ▶ By the argmax property: $\langle f - f_{n-1}, h_n \rangle \geq \sum_{i=1}^n a_i \langle f - f_{n-1}, g_i \rangle = \langle f - f_{n-1}, f \rangle$
- ▶ By boundedness of \mathbb{D} : $\|f - h_n\|^2 \leq 4|\mathbb{D}|^2$
- ▶ Get

$$\|f - f_n\|^2 \leq \left(1 - \frac{1}{n}\right)^2 \|f - f_{n-1}\|^2 + \frac{4|\mathbb{D}|^2}{n^2}$$

- ▶ Base case: $\|f - f_1\|^2 \leq |\mathbb{D}|^2 \leq 4|\mathbb{D}|^2$. Induction gives

$$\|f - f_n\|^2 \leq \left[\left(1 - \frac{1}{n}\right)^2 \frac{1}{n-1} + \frac{1}{n^2} \right] 4|\mathbb{D}|^2 = \frac{1}{n} 4|\mathbb{D}|^2.$$

Main Theorem

Theorem (Hong, Siegel & X 2021)

Assume that the true solution $u \in \mathcal{V}_1(\mathbb{D})$ satisfies $\|u\|_{\mathcal{V}_1(\mathbb{D})} \leq M$ and let the numerical solution $u_{n,M,N} \in \Sigma_{n,M}(\mathbb{D})$ be obtained by the relaxed greedy algorithm for n steps. Then we have

$$\mathbb{E}_{x_1, \dots, x_N} (J(u_{n,M,N}) - J(u)) \leq M \left[C_1 (1 + \|f\|_{L^\infty(\Omega)}) N^{-\frac{1}{2}} + C_2 M n^{-1} \right]. \quad (45)$$

and

$$\mathbb{E}_{x_1, \dots, x_N} (\|u_{n,M,N} - u\|_{H^m(\Omega)}^2) \leq M \left[C'_1 N^{-\frac{1}{2}} + C'_2 M n^{-1} \right], \quad (46)$$

where C'_1 and C'_2 depend only upon the dictionary and the differential operator.

Sketch of Proof

- 1 Recall $J(u_n) = \min_{v \in \Sigma_{n,M}} J(v)$. We write

$$\begin{aligned}\mathbb{E}_{x_1, \dots, x_N} (J(u_{n,M,N}) - J(u)) &\leq \mathbb{E}_{x_1, \dots, x_N} (J_N(u_{n,M,N}) - J_N(u_n)) \\ &\quad + \mathbb{E}_{x_1, \dots, x_N} (|J_N(u_n) - J(u_n)|) \\ &\quad + \mathbb{E}_{x_1, \dots, x_N} (|J(u_n) - J(u)|).\end{aligned}\tag{47}$$

- 2 Since $u_{n,M,N}$ is obtained by the relaxed greedy algorithm for n steps. The first term on the right hand side can be estimated by

$$\begin{aligned}J_N(u_{n,M,N}) - J_N(u_n) &= J_N(u_{n,M,N}) - \min_{v \in \Sigma_{n,M}} J_N(v) + \min_{v \in \Sigma_{n,M}} J_N(v) - J_N(u_n) \\ &\leq J_N(u_{n,M,N}) - \min_{v \in \Sigma_{n,M}} J_N(v) \lesssim \frac{1}{n}.\end{aligned}\tag{48}$$

- 3 The second term on the right is the error of numerical quadrature which is bounded via the Rademacher complexity by

$$CM(K + \|f\|_{L^\infty(\Omega)})N^{-\frac{1}{2}}.$$

- 4 The third term can be bounded by the approximation error.

- Loss function for PINN:

$$L(u) = \min_{v \in H^1(\Omega)} L(v) \text{ with } L(v) = \int_{\Omega} |\Delta v + f|^2 dx + \int_{\partial\Omega} \left| \frac{\partial v}{\partial n} \right|^2 ds. \quad (49)$$

- Discretization of the loss:

$$L_N(u_{n,N}) = \min_{v \in \Sigma_{n,M}^k} L_N(v) \text{ with } L_N(v) = \frac{1}{N} \sum_{i=1}^N |\Delta v(x_i) + f(x_i)|^2 + \frac{1}{N_b} \sum_{i=1}^{N_b} \left| \frac{\partial v}{\partial n}(x_i) \right|^2. \quad (50)$$

- Similar results can be obtained for PINN.

- Ref: Raissi, Perdikaris and Karniadakis 2019

Remarks on the relaxed greedy algorithm

Question: How to solve

$$g_k = \arg \max_{g \in \mathbb{D}} \langle \nabla J_N(u_{k-1}), g \rangle$$

- Feasible for small d ($d = 1, 2, 3$).
- Challenge if d is large, but possible if u has special structure.
- We do not know any other method.

Orthogonal greedy algorithm

- Orthogonal greedy algorithm (OGA) for function approximation in a Hilbert space H :

$$u_0 = 0, \quad g_n = \arg \max_{g \in \mathbb{D}} |\langle u - u_{n-1}, g \rangle_H|, \quad u_n = P_n u, \quad (51)$$

where P_n : orthogonal projection onto $V_n = \text{span}\{g_1, \dots, g_n\}$.

Theorem (Siegel & X 2022)

Let the iterates f_n be given by the orthogonal greedy algorithm, where $f \in \mathcal{V}_1(\mathbb{P}_k^d)$. Then we have

$$\|f_n - f\|_{L^2} \lesssim \|f\|_{\mathcal{V}_1(\mathbb{P}_k^d)} n^{-\frac{1}{2} - \frac{2k+1}{2d}}. \quad (52)$$

- Implies that the OGA trains optimal neural networks
- Open question: no stability, i.e. $\|f_n\|_{\mathcal{V}_1(\mathbb{D})}$ may be arbitrarily large!
- Ref:** Barron, Cohen, Dahmen and DeVore 2008, Siegel and Xu 2022.

- 1 Model problem
- 2 Stable shallow neural network functions and approximation
- 3 Error estimates for numerical quadrature of stable NN approximation
- 4 Optimization algorithms
- 5 Numerical examples
- 6 Summary

NN with OGA for Data fitting

Example (2D approximation)

Consider approximating the following 2D function

$$u(x, y) = \cos(2\pi x) \cos(2\pi y), \quad (x, y) \in (0, 1)^2.$$

By fixing $\|\omega\| = 1$ and $b \in [-2, 2]$, the convergence order of OGA is shown in Table below for ReLU^k neural networks. Theoretical order is shown in parenthesis.

n	$k = 1(n^{-1.25})$		$k = 2(n^{-1.75})$		$k = 3(n^{-2.25})$	
	$\ u - u_n\ _{L^2}$	order	$\ u - u_n\ _{L^2}$	order	$\ u - u_n\ _{L^2}$	order
2	4.969e-01	-	4.998e-01	-	4.976e-01	-
4	4.883e-01	0.025	4.992e-01	0.002	4.957e-01	0.006
8	2.423e-01	1.011	3.233e-01	0.627	4.193e-01	0.242
16	6.632e-02	1.869	4.911e-02	2.719	1.099e-01	1.932
32	2.206e-02	1.588	1.688e-02	1.541	8.075e-03	3.767
64	1.060e-02	1.058	4.156e-03	2.022	1.149e-03	2.813
128	4.284e-03	1.306	9.773e-04	2.088	2.185e-04	2.395
256	1.703e-03	1.331	2.622e-04	1.898	4.718e-05	2.211

Table: Convergence order of OGA with ReLU^k activation function

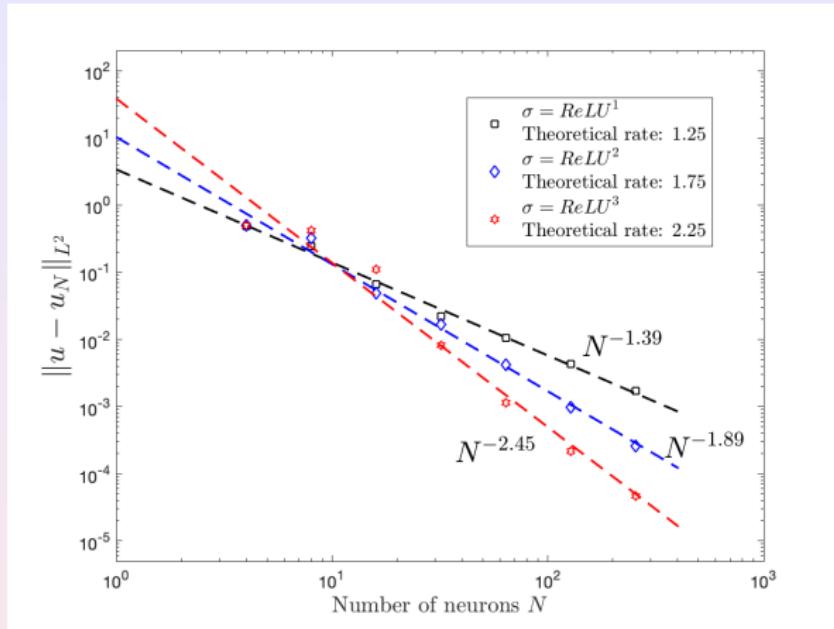


Figure: Convergence order of OGA with ReLU^k activation function

NN with OGA for Numerical PDE

Example (1D elliptic equation)

Let us consider the 1D second order elliptic equation on $\Omega = (-1, 1)$:

$$-u'' + u = f, \text{ in } \Omega \quad (53)$$

$$\frac{\partial u}{\partial n} = 0, \text{ on } \partial\Omega. \quad (54)$$

with the source term $f = (1 + \pi^2) \cos(\pi x)$, then the analytical solution is $u(x) = \cos(\pi x)$. Let $\sigma = \text{ReLU}^2$ and the convergence rates are predicted theoretically.

n	$\ u - u_n\ _{L^2}$	order(n^{-3})	$\ u - u_n\ _{H^1}$	order(n^{-2})
16	7.86e-04	-	2.79e-02	-
32	7.70e-05	3.35	5.89e-03	2.24
64	8.45e-06	3.19	1.36e-03	2.11
128	9.68e-07	3.13	3.22e-04	2.08
256	1.18e-07	3.04	7.81e-05	2.04
512	1.44e-08	3.03	1.94e-05	2.01
1024	1.83e-09	2.97	4.86e-06	1.99
2048	2.50e-10	2.88	1.28e-06	1.93

Table: L^2 and H^1 numerical error of the OGA solution

OGA v.s. SGD and Adam

We take the learning rate $\eta = 1e-3$ for SGD and Adam. It is difficult to observe any convergence order from the solutions of SGD or Adam:

n	Adam				SGD			
	$\ u - u_n\ _{L^2}$	order	$\ u - u_n\ _{H^1}$	order	$\ u - u_n\ _{L^2}$	order	$\ u - u_n\ _{H^1}$	order
16	1.61e-02	-	1.45e-01	-	1.30e-02	-	1.52e-01	-
32	3.71e-03	2.12	5.84e-02	1.32	9.35e-03	0.47	1.13e-01	0.43
64	1.80e-03	1.04	3.46e-02	0.76	7.11e-03	0.39	8.64e-02	0.38
128	5.52e-04	1.70	1.43e-02	1.27	5.91e-03	0.27	7.22e-02	0.26
256	2.26e-04	1.29	6.99e-03	1.04	5.75e-03	0.04	7.03e-02	0.04
512	1.88e-04	0.27	3.90e-03	0.84	4.41e-03	0.38	5.40e-02	0.38
1024	2.09e-04	-0.16	2.56e-03	0.61	1.52e-03	1.54	1.99e-02	1.43
2048	4.11e-04	-0.97	2.51e-03	0.03	3.22e-03	-1.09	3.56e-02	-0.84

Table: Numerical results from different training methods.

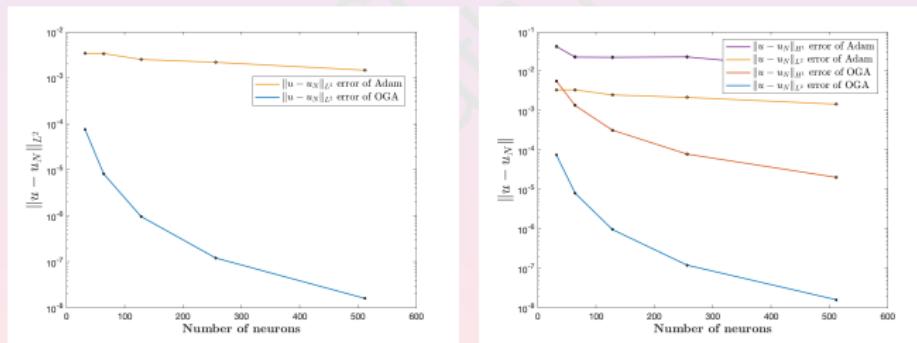


Figure: OGA vs Adam

NN with OGA for 4th order PDE

Example (1D 4th-order equation)

We consider the 4th-order equation $(-\Delta)^2 u + u = f$ on $(-1, 1)$ with

$$u(x) = (1 - x)^4(1 + x)^4.$$

Take $\sigma = \text{ReLU}^3$. The convergence orders with $\|\cdot\|_0$ and $\|\cdot\|_a$ errors are listed in the following table to confirm 4th order and 2nd order convergence, respectively.

n	$\ u - u_n\ _{L^2}$	order	$\ u - u_n\ _a$	order
2	8.762e-01	-	6.063e+00	-
4	9.892e-01	-0.17	4.122e+00	0.56
8	1.493e-02	6.05	1.069e+00	1.95
16	2.608e-04	5.84	1.431e-01	2.90
32	1.089e-05	4.58	3.120e-02	2.20
64	5.530e-07	4.30	6.639e-03	2.23
128	2.989e-08	4.21	1.679e-03	1.98
256	1.746e-09	4.10	4.092e-04	2.04

Table: The $\|\cdot\|_0$ and $\|\cdot\|_a$ errors of the OGA solution.

Numerical experiments of OGA

Example (2D 4th-order equation)

Next consider the $\|\cdot\|_0$ and $\|\cdot\|_a$ errors of OGA for 2D 4th-order equations on $\Omega = (-1, 1)^2$. Let the exact solution to be $u(x, y) = (x^2 - 1)^4(y^2 - 1)^4$ to satisfy the Neumann boundary conditions. We have

n	$\ u - u_n\ _{L^2}$	order	$\ u - u_n\ _a$	order
2	6.528e-01	-	7.927e+00	-
4	7.859e-01	-0.27	7.593e+00	0.06
8	9.906e-01	-0.33	6.295e+00	0.27
16	8.215e-01	0.27	4.003e+00	0.65
32	1.513e-01	2.44	1.446e+00	1.47
64	7.206e-02	1.07	4.747e-01	1.61
128	2.259e-02	1.67	1.809e-01	1.39
256	4.696e-03	2.27	6.970e-02	1.38

Table: The convergence order with $\|\cdot\|_0$ and $\|\cdot\|_a$ errors by OGA.

Nonlinear problem: 2D

Example (A nonlinear 2D example)

Consider the following nonlinear 2D equation $-\Delta u + u^3 + u = f$ on $(0, 1)^2$ with $\partial u / \partial n = 0$. The analytical solution is $u = \cos(2\pi x) \cos(2\pi y)$ and the dictionary for RGA is taken as

$$\mathbb{D} = \{\sigma(w_1x + w_2y + b) | (w_1, w_2, b) \in [-20, 20]^3\}$$

where $\sigma(x)$ is the sigmoid function. The convergence is considered on the approximating space where $\|u\|_{\mathcal{K}_1(\mathbb{D})} \leq M = 15$.

n	$\ u - u_n\ _2$	order	$\ Du - Du_n\ _2$	order	$J(u_n) - J(u)$	order(n^{-1})
16	7.847e-01	-	4.645e+00	-	1.805e+04	-
32	6.679e-01	0.23	2.955e+00	0.65	7.563e+03	1.25
64	2.370e-01	1.49	1.675e+00	0.82	2.328e+03	1.70
128	1.216e-01	0.96	1.087e+00	0.62	9.680e+02	1.27
256	6.184e-02	0.98	5.205e-01	1.06	2.222e+02	2.12
512	3.797e-02	0.70	3.611e-01	0.53	1.067e+02	1.06
1024	2.687e-02	0.50	2.110e-01	0.77	3.662e+01	1.54
2048	1.072e-02	1.33	1.432e-01	0.56	1.663e+01	1.14

Table: Convergence order of RGA.

Nonlinear problem: 2D

Example (2D Poisson-Boltzmann equation)

Consider the PB equation $-\Delta u + \kappa \sinh(u) = f$ on the sphere $\{(x, y) | x^2 + y^2 \leq 4\}$ with $\partial u / \partial n = 0$. The energy functional for this problem is

$$\mathcal{J}(u) = \int_{\Omega} \left(\frac{1}{2} |\nabla u|^2 + \kappa \cosh(u) - fu \right) dx,$$

which is a strictly **convex** and **coercive** energy as long as $\kappa > 0$. We set $\kappa = 1$ and consider the radially symmetric solution

$$u(x, y) = \cos\left(\frac{\pi}{2} \sqrt{x^2 + y^2}\right).$$

n	$\ u - u_n\ _2$	order	$\ Du - Du_n\ _2$	order	$J(u_n) - J(u)$	$\text{order}(n^{-1})$
16	1.103e+00	-	2.470e+00	-	1.512e+05	-
32	6.499e-01	0.76	1.845e+00	0.42	7.750e+04	0.96
64	5.440e-01	0.26	1.536e+00	0.26	5.480e+04	0.50
128	2.435e-01	1.16	7.509e-01	1.03	1.285e+04	2.09
256	1.507e-01	0.69	4.532e-01	0.73	4.690e+03	1.45
512	7.373e-02	1.03	2.659e-01	0.77	1.423e+03	1.72
1024	2.949e-02	1.32	1.647e-01	0.69	5.363e+02	1.41
2048	2.401e-02	0.30	1.156e-01	0.51	2.571e+02	1.06

Table: Convergence order of RGA for the 2D Poisson-Boltzmann equation.

10D example

Example (10D Poisson equation)

Consider a 10d elliptic equation in the following form

$$-\nabla \cdot (\alpha \nabla u) + u = f, \quad x \in (0, 1)^{10}, \quad \frac{\partial u}{\partial n} = 0, \quad x \in \partial(0, 1)^{10}. \quad (55)$$

We take

$$u = \sum_{i=1}^{10} \cos(\pi x_i), \quad \alpha = \sqrt{1 + \sum_{i=1}^{10} (x_i - \frac{1}{2})^2}. \quad (56)$$

We discretize the energy using 100 million quasi-Monte-Carlo samples in $[0, 1]^{10}$ and optimize the energy using the OGA with dictionary

$$\mathbb{P}_2^{10,r} = \{\sigma(\omega \cdot x + b), \quad \omega = \pm e_i, \quad i = 1, \dots, 10, \quad b \in [-2, 2]\}, \quad (57)$$

and we obtained

n	$\ u - u_n\ _{L^2}$	order(n^{-3})	$\ u - u_n\ _{H^1}$	order(n^{-2})
2	2.058e+00	0.00	6.778e+00	0.00
4	1.850e+00	0.15	6.139e+00	0.14
8	1.307e+00	0.50	4.604e+00	0.42
16	5.019e-01	1.38	3.184e+00	0.53
32	4.702e-02	3.42	5.988e-01	2.41
64	4.629e-03	3.34	1.104e-01	2.44
128	4.437e-04	3.38	2.272e-02	2.28
256	5.411e-05	3.04	5.194e-03	2.13

- 1 Model problem
- 2 Stable shallow neural network functions and approximation
- 3 Error estimates for numerical quadrature of stable NN approximation
- 4 Optimization algorithms
- 5 Numerical examples
- 6 Summary

Summary

- 1 Using neural network to solve $2m$ -th order PDEs.
- 2 Inverse inequality does not hold for the neural network function class
- 3 Using the good property of stable neural network
- 4 Convergence analysis for sampling
- 5 Greedy algorithm to solve the optimization problem with numerical examples.

References

- Q. Hong and J. W. Siegel, and J. Xu, A Priori Analysis of Stable Neural Network Solutions to Numerical PDEs, arXiv preprint arXiv:2104.02903 (2021).
- J. W. Siegel, Q. Hong, X. Jin, W. Hao and J. Xu, Greedy training algorithms for neural networks and applications to PDEs. Journal of Computational Physics, 2023, Vol. 484: 112084.

Thank you very much!