Approximation Theory for Neural Networks

Jonathan W. Siegel

Texas A&M University

jwsiegel@tamu.edu

Workshop and Summer School in Applied Analysis
Chemnitz University of Technology
Sept 22-26, 2025

Outline

1. Neural Networks
2. Shallow Network Approximation
   - Smooth activation functions and polynomials
   - General activation functions and Barron's space
   - Approximation rates for convex hulls
   - Lower Bounds
3. Deep ReLU Network Approximation
   - Upper Bounds
   - Approximating Multiplication
   - Bit Extraction
   - Lower Bounds
   - Stability
   - Symmetry-Preserving Neural Networks

Neural Networks

- Recently, neural networks have been widely applied to science and scientific computing:
  - Solving PDEs
  - Learning operators from data
  - Inverse Problem/Inverse Design
  - e.g. protein folding, modeling quantum systems, predicting materials properties, etc.

Neural Networks

- Recently, neural networks have been widely applied to science and scientific computing:
  - Solving PDEs
  - Learning operators from data
  - Inverse Problem/Inverse Design
  - e.g. protein folding, modeling quantum systems, predicting materials properties, etc.
- At its heart, a neural network learns a function $f : X \to Y$ from data
  - For example $X = \mathbb{R}^d$ and $Y = \mathbb{R}$
  - X and Y are Banach spaces in operator learning

Neural Networks

- Recently, neural networks have been widely applied to science and scientific computing:
  - Solving PDEs
  - Learning operators from data
  - Inverse Problem/Inverse Design
  - e.g. protein folding, modeling quantum systems, predicting materials properties, etc.
- At its heart, a neural network learns a function $f : X \to Y$ from data
  - For example $X = \mathbb{R}^d$ and $Y = \mathbb{R}$
  - X and Y are Banach spaces in operator learning
- Fundamental problems:
  - How efficient are neural networks?
  - How do neural networks compare with classical methods?

Neural Networks

- Consider an affine map $A_{\mathbf{W},b} : \mathbb{R}^n \to \mathbb{R}^k$

$$A_{\mathbf{W},b}(x) = \mathbf{W}x + b. \tag{1}$$

Neural Networks

- Consider an affine map $A_{\mathbf{W},b} : \mathbb{R}^n \to \mathbb{R}^k$

$$A_{\mathbf{W},b}(x) = \mathbf{W}x + b. \quad (1)$$

- Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an *activation function*
    - When applied to a vector, σ is applied component-wise

Neural Networks

- Consider an affine map $A_{\mathbf{W},b} : \mathbb{R}^n \to \mathbb{R}^k$

$$A_{\mathbf{W},b}(x) = \mathbf{W}x + b. \tag{1}$$

- Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an *activation function*
  - When applied to a vector, σ is applied component-wise
- A deep neural network with width W, depth L, and activation function σ mapping $\mathbb{R}^d$ to $\mathbb{R}^k$ is a composition

$$A_{\mathbf{W}_L,b_L} \circ \sigma \circ A_{\mathbf{W}_{L-1},b_{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ A_{\mathbf{W}_1,b_1} \circ \sigma \circ A_{\mathbf{W}_0,b_0} \tag{2}$$

  - Here $A_{\mathbf{W}_1,b_1}, ..., A_{\mathbf{W}_{L-1},b_{L-1}} : \mathbb{R}^W \to \mathbb{R}^W$

Neural Networks

- Consider an affine map $A_{\mathbf{W},b} : \mathbb{R}^n \to \mathbb{R}^k$

$$A_{\mathbf{W},b}(x) = \mathbf{W}x + b. \tag{1}$$

- Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an *activation function*
  - When applied to a vector, σ is applied component-wise
- A deep neural network with width W, depth L, and activation function σ mapping $\mathbb{R}^d$ to $\mathbb{R}^k$ is a composition

$$A_{\mathbf{W}_L,b_L} \circ \sigma \circ A_{\mathbf{W}_{L-1},b_{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ A_{\mathbf{W}_1,b_1} \circ \sigma \circ A_{\mathbf{W}_0,b_0} \tag{2}$$

  - Here $A_{\mathbf{W}_1,b_1}, ..., A_{\mathbf{W}_{L-1},b_{L-1}} : \mathbb{R}^W \to \mathbb{R}^W$
  - We denote the set of these by $\Upsilon^{W,L}_\sigma(\mathbb{R}^d, \mathbb{R}^k)$
    - $\Upsilon^{W,L}_\sigma(\mathbb{R}^d)$ if $k = 1$

Shallow Neural Networks

- Shallow neural networks with width n and activation function σ:

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

Shallow Neural Networks

- Shallow neural networks with width n and activation function σ:

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

- Examples of activation functions:
  - Sigmoidal: $\sigma(x) = 1/(1+e^{-x})$
  - ReLU: $\sigma(x) = \max(0,x)$
  - ReLU^k: $\sigma(x) = \max(0,x)^k$

Universal Approximation

- Let $\Omega \subset \mathbb{R}^d$ be a compact set
  - Are neural networks dense in $C(\Omega)$?

[1] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257, Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072.

[2] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

[3] Boris Hanin. "Universal function approximation by deep neural nets with bounded width and relu activations". In: *Mathematics* 7.10 (2019), p. 992.

Universal Approximation

- Let $\Omega \subset \mathbb{R}^d$ be a compact set
  - Are neural networks dense in $C(\Omega)$?
- Yes, $\bigcup_{n\geq 1} \Sigma_n^\sigma(\mathbb{R}^d)$ is dense if $\sigma \in C(\mathbb{R})$ is not a polynomial[1]

[1] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257, Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072.

[2] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

[3] Boris Hanin. "Universal function approximation by deep neural nets with bounded width and relu activations". In: *Mathematics* 7.10 (2019), p. 992.

Universal Approximation

- Let $\Omega \subset \mathbb{R}^d$ be a compact set
  - Are neural networks dense in $C(\Omega)$?
- Yes, $\bigcup_{n\geq 1} \Sigma_n^\sigma(\mathbb{R}^d)$ is dense if $\sigma \in C(\mathbb{R})$ is not a polynomial[1]
- Yes, $\bigcup_{W\geq 1} \Upsilon_\sigma^{W,L}(\mathbb{R}^d)$ is dense for any $L \geq 1$ if $\sigma \in C(\mathbb{R})$ is not a polynomial[2]

[1] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257, Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072.

[2] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

[3] Boris Hanin. "Universal function approximation by deep neural nets with bounded width and relu activations". In: *Mathematics* 7.10 (2019), p. 992.

Universal Approximation

- Let $\Omega \subset \mathbb{R}^d$ be a compact set
  - Are neural networks dense in $C(\Omega)$?
- Yes, $\bigcup_{n\geq 1} \Sigma_n^\sigma(\mathbb{R}^d)$ is dense if $\sigma \in C(\mathbb{R})$ is not a polynomial[1]
- Yes, $\bigcup_{W\geq 1} \Upsilon_\sigma^{W,L}(\mathbb{R}^d)$ is dense for any $L \geq 1$ if $\sigma \in C(\mathbb{R})$ is not a polynomial[2]
- Yes, $\bigcup_{L\geq 1} \Upsilon_\sigma^{W,L}(\mathbb{R}^d)$ is dense[3] if $W \geq d+1$ and σ is the ReLU

[1] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257, Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072.

[2] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

[3] Boris Hanin. "Universal function approximation by deep neural nets with bounded width and relu activations". In: *Mathematics* 7.10 (2019), p. 992.

Universal Approximation

- Let $\Omega \subset \mathbb{R}^d$ be a compact set
  - Are neural networks dense in $C(\Omega)$?
- Yes, $\bigcup_{n\geq 1} \Sigma_n^\sigma(\mathbb{R}^d)$ is dense if $\sigma \in C(\mathbb{R})$ is not a polynomial[1]
- Yes, $\bigcup_{W\geq 1} \Upsilon_\sigma^{W,L}(\mathbb{R}^d)$ is dense for any $L \geq 1$ if $\sigma \in C(\mathbb{R})$ is not a polynomial[2]
- Yes, $\bigcup_{L\geq 1} \Upsilon_\sigma^{W,L}(\mathbb{R}^d)$ is dense[3] if $W \geq d+1$ and σ is the ReLU
- What about approximation rates?
  - Need assumptions on the target function

[1] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257, Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072.

[2] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

[3] Boris Hanin. "Universal function approximation by deep neural nets with bounded width and relu activations". In: *Mathematics* 7.10 (2019), p. 992.

Sobolev and Besov Spaces[6]

- We consider the Sobolev spaces $W^s(L_q(\Omega))$, defined (for integer s) by

$$\|f\|_{W^s(L_q(\Omega))} = \|f\|_{L_q(\Omega)} + \|f^{(s)}\|_{L_q(\Omega)} \tag{4}$$

  - The L_q-norm is

$$\|f\|_{L_q(\Omega)} = \left(\int_\Omega |f(x)|^q dx\right)^{1/q} \tag{5}$$

  - Can also be defined[4] for non-integer s
  - Can also consider more general spaces like Besov, Triebel-Lizorkin, etc[5]

---

[4] Eleonora Di Nezza, Giampiero Palatucci, and Enrico Valdinoci. "Hitchhiker's guide to the fractional Sobolev spaces". In: *Bulletin des sciences mathématiques* 136.5 (2012), pp. 521–573.

[5] Ronald A DeVore and Robert C Sharpley. "Besov spaces on domains in $\mathbb{R}^d$". In: *Transactions of the American Mathematical Society* 335.2 (1993), pp. 843–864, Hans Triebel. *Theory of function spaces III*. Springer, 2006.

[6] Lawrence C Evans. *Partial differential equations*. Vol. 19. American Mathematical Soc., 2010.

Neural Network Approximation

- How efficiently can neural networks approximate functions from Sobolev and Besov spaces?

Neural Network Approximation

- How efficiently can neural networks approximate functions from Sobolev and Besov spaces?
  - Minimax rates for deep networks:

$$\sup_{\|f\|_{W^s(L_q)}\leq 1} \inf_{f_{W,L}\in\Upsilon_\sigma^{W,L}(\mathbb{R}^d)} \|f - f_{W,L}\|_{L_p} \quad (6)$$

- Number of parameters $P = O(W^2L)$

Neural Network Approximation

- How efficiently can neural networks approximate functions from Sobolev and Besov spaces?
  - Minimax rates for deep networks:

$$\sup_{\|f\|_{W^s(L_q)}\leq 1} \inf_{f_{W,L}\in \Upsilon_\sigma^{W,L}(\mathbb{R}^d)} \|f - f_{W,L}\|_{L_p} \tag{6}$$

    - Number of parameters $P = O(W^2L)$
  - Minimax rates for shallow networks:

$$\sup_{\|f\|_{W^s(L_q)}\leq 1} \inf_{f_n\in \Sigma_n^\sigma(\mathbb{R}^d)} \|f - f_n\|_{L_p} \tag{7}$$

Neural Network Approximation

- How efficiently can neural networks approximate functions from Sobolev and Besov spaces?
  - Minimax rates for deep networks:

$$\sup_{\|f\|_{W^s(L_q)}\leq 1} \inf_{f_{W,L}\in \Upsilon^{W,L}_\sigma(\mathbb{R}^d)} \|f - f_{W,L}\|_{L_p} \tag{6}$$

    - Number of parameters $P = O(W^2L)$
  - Minimax rates for shallow networks:

$$\sup_{\|f\|_{W^s(L_q)}\leq 1} \inf_{f_n\in \Sigma^{\sigma}_n(\mathbb{R}^d)} \|f - f_n\|_{L_p} \tag{7}$$

  - Need the compact embedding condition: $s/d > 1/q - 1/p$.

Neural Network Approximation

- How efficiently can neural networks approximate functions from Sobolev and Besov spaces?
  - Minimax rates for deep networks:

$$\sup_{\|f\|_{W^s(L_q)}\leq 1} \inf_{f_{W,L}\in\Upsilon^{W,L}_\sigma(\mathbb{R}^d)} \|f - f_{W,L}\|_{L_p} \tag{6}$$

    - Number of parameters $P = O(W^2L)$
  - Minimax rates for shallow networks:

$$\sup_{\|f\|_{W^s(L_q)}\leq 1} \inf_{f_n\in\Sigma^{\sigma}_n(\mathbb{R}^d)} \|f - f_n\|_{L_p} \tag{7}$$

  - Need the compact embedding condition: $s/d > 1/q - 1/p$.
  - The *non-linear* regime $q < p$ is of particular interest

The non-linear regime

- Suppose that $d = 1$, $s = 1$, and $q = \infty$
  - Then $W^s(L_q)$ is the class of Lipschitz functions

$$|f(x) - f(y)| \leq C|x - y| \tag{8}$$

The non-linear regime

- Suppose that $d = 1$, $s = 1$, and $q = \infty$
  - Then $W^s(L_q)$ is the class of Lipschitz functions

$$|f(x) - f(y)| \leq C|x - y| \tag{8}$$

- If instead $q = 1$, $W^s(L_q)$ is (almost) the class of BV functions

$$\sum_{i=1}^{n} |f(x_i) - f(x_{i-1})| \leq C \tag{9}$$

for $x_0 < x_1 < \cdots < x_n$.
  - Allows jump discontinuities!

The non-linear regime

- Suppose that $d = 1$, $s = 1$, and $q = \infty$
  - Then $W^s(L_q)$ is the class of Lipschitz functions

$$|f(x) - f(y)| \leq C|x - y| \tag{8}$$

- If instead $q = 1$, $W^s(L_q)$ is (almost) the class of BV functions

$$\sum_{i=1}^{n} |f(x_i) - f(x_{i-1})| \leq C \tag{9}$$

  for $x_0 < x_1 < \cdots < x_n$.
  - Allows jump discontinuities!
- Approximate in L_p to error ϵ:
  - each jump must be captured to resolution ϵ^p

The non-linear regime

- Suppose that $d = 1$, $s = 1$, and $q = \infty$
  - Then $W^s(L_q)$ is the class of Lipschitz functions

$$|f(x) - f(y)| \leq C|x - y| \tag{8}$$

- If instead $q = 1$, $W^s(L_q)$ is (almost) the class of BV functions

$$\sum_{i=1}^{n} |f(x_i) - f(x_{i-1})| \leq C \tag{9}$$

  for $x_0 < x_1 < \cdots < x_n$.
  - Allows jump discontinuities!
- Approximate in L_p to error ϵ:
  - each jump must be captured to resolution ϵ^p
- Approximation in L_p for $p > q$ requires *sharper resolution of discontinuities*
  - Requires non-linear methods of approximation

Smooth activation functions

- Suppose that $\sigma \in C^\infty(\mathbb{R})$
- Then we have

$$x\sigma'(b) = \lim_{h\to 0} \frac{\sigma(hx+b) - \sigma(b)}{h} \in \Sigma_2^\sigma(\mathbb{R}) \tag{10}$$

Smooth activation functions

- Suppose that $\sigma \in C^\infty(\mathbb{R})$
- Then we have

$$x\sigma'(b) = \lim_{h\to 0} \frac{\sigma(hx+b)-\sigma(b)}{h} \in \Sigma_2^\sigma(\mathbb{R}) \quad (10)$$

$$x^n\sigma^{(n)}(b) = \lim_{h\to 0} \frac{\sum_{j=0}^n (-1)^{n-j}\binom{n}{j}\sigma(jhx+b)}{h} \in \Sigma_{n+1}^\sigma(\mathbb{R}) \quad (11)$$

  - These limits are uniform for $x \in \Omega$ (compact)

Smooth activation functions

- Suppose that $\sigma \in C^\infty(\mathbb{R})$
- Then we have

$$x\sigma'(b) = \lim_{h\to 0} \frac{\sigma(hx + b) - \sigma(b)}{h} \in \Sigma_2^\sigma(\mathbb{R}) \quad (10)$$

$$x^n\sigma^{(n)}(b) = \lim_{h\to 0} \frac{\sum_{j=0}^n (-1)^{n-j}\binom{n}{j}\sigma(jhx + b)}{h} \in \Sigma_{n+1}^\sigma(\mathbb{R}) \quad (11)$$

  - These limits are uniform for $x \in \Omega$ (compact)
- If σ is not a polynomial, then $\sigma^{(n)}(b) \neq 0$ for some b for any $n \geq 0$
  - We conclude that $\mathcal{P}_n(\mathbb{R}) \subset \overline{\Sigma_{n+1}^\sigma(\mathbb{R})}$

Smooth activation functions ($d > 1$)

Lemma

There exist $\omega_1, \ldots, \omega_N$ with $N = O(n^{d-1})$ directions, such that every polynomial $p \in \mathcal{P}_n(\mathbb{R}^d)$ can be written

$$p(x) = \sum_{i=1}^{N} p_i(\omega_i \cdot x) \tag{12}$$

for some $p_i \in \mathcal{P}_n(\mathbb{R})$.

[7] Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072, Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

Smooth activation functions ($d > 1$)

Lemma

There exist $\omega_1, \ldots, \omega_N$ with $N = O(n^{d-1})$ directions, such that every polynomial $p \in \mathcal{P}_n(\mathbb{R}^d)$ can be written

$$p(x) = \sum_{i=1}^{N} p_i(\omega_i \cdot x) \tag{12}$$

for some $p_i \in \mathcal{P}_n(\mathbb{R})$.

- Hence, we conclude that $\mathcal{P}_n(\mathbb{R}^d) \subset \overline{\Sigma_N^\sigma(\mathbb{R}^d)}$ with $N = O(n^d)$

[7] Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072, Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

Smooth activation functions ($d > 1$)

Lemma

There exist $\omega_1, ..., \omega_N$ with $N = O(n^{d-1})$ directions, such that every polynomial $p \in \mathcal{P}_n(\mathbb{R}^d)$ can be written

$$p(x) = \sum_{i=1}^{N} p_i(\omega_i \cdot x) \tag{12}$$

for some $p_i \in \mathcal{P}_n(\mathbb{R})$.

- Hence, we conclude that $\mathcal{P}_n(\mathbb{R}^d) \subset \overline{\Sigma_N^\sigma(\mathbb{R}^d)}$ with $N = O(n^d)$
- Thus, approximation by $\Sigma_N^\sigma(\mathbb{R}^d)$ is at least as good as with $\mathcal{P}_n(\mathbb{R}^d)$
- Same technique used to prove density[7]

[7] Kurt Hornik. "Some new results on neural network approximation". In: *Neural networks* 6.8 (1993), pp. 1069–1072, Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867.

Rates for smooth activation functions

- This argument gives the approximation rate[8]:

$$\inf_{f_n \in \Sigma_n^\sigma(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{W^s(L_q)} n^{-\frac{s}{d}+(\frac{1}{q}-\frac{1}{p})_+}. \tag{13}$$

[8] Hrushikesh N Mhaskar. "Neural networks for optimal approximation of smooth and analytic functions". In: *Neural Computation* 8.1 (1996), pp. 164–177.

[9] Vitaly E Maiorov. "On best approximation by ridge functions". In: *Journal of Approximation Theory* 99.1 (1999), pp. 68–94, Vitaly Maiorov and Allan Pinkus. "Lower bounds for approximation by MLP neural networks". In: *Neurocomputing* 25.1-3 (1999), pp. 81–91.

[10] Vitaly E Maiorov and Ron Meir. "On the near optimality of the stochastic approximation of smooth functions by neural networks". In: *Advances in Computational Mathematics* 13.1

Rates for smooth activation functions

- This argument gives the approximation rate[8]:

$$\inf_{f_n \in \Sigma_n^\sigma(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{W^s(L_q)} n^{-\frac{s}{d} + (\frac{1}{q} - \frac{1}{p})_+}. \tag{13}$$

- Lower bounds:
  - For general smooth activation functions[9]:

$$\sup_{\|f\|_{W^s(L_q)} \leq 1} \inf_{f_n \in \Sigma_n^\sigma(\mathbb{R}^d)} \|f - f_n\|_{L_p} \geq Cn^{-\frac{s}{d-1}} \tag{14}$$

  - For the sigmoid activation function[10]:

$$\sup_{\|f\|_{W^s(L_q)} \leq 1} \inf_{f_n \in \Sigma_n^\sigma(\mathbb{R}^d)} \|f - f_n\|_{L_p} \geq C(n \log n)^{-\frac{s}{d}} \tag{15}$$

---

[8] Hrushikesh N Mhaskar. "Neural networks for optimal approximation of smooth and analytic functions". In: *Neural Computation* 8.1 (1996), pp. 164–177.

[9] Vitaly E Maiorov. "On best approximation by ridge functions". In: *Journal of Approximation Theory* 99.1 (1999), pp. 68–94, Vitaly Maiorov and Allan Pinkus. "Lower bounds for approximation by MLP neural networks". In: *Neurocomputing* 25.1-3 (1999), pp. 81–91.

[10] Vitaly E Maiorov and Ron Meir. "On the near optimality of the stochastic approximation of smooth functions by neural networks". In: *Advances in Computational Mathematics* 13.1

More general activation functions

- Consider the Heaviside activation:

$$\sigma_0(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0. \end{cases} \tag{16}$$

More general activation functions

- Consider the Heaviside activation:

$$\sigma_0(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0. \end{cases} \tag{16}$$

- Observe that for $r \in [0, 1]$

$$e^{irx} = 1 + i\int_0^x re^{irt}dt = 1 + i\int_0^1 re^{irt}\sigma_0(x - t)dt. \tag{17}$$

  - The complex exponential can be written as an integral in terms of σ_0!

Barron's space

- Suppose that f satisfies Barron's condition[11]:

$$|f|_{\mathcal{B}} := \int_{\mathbb{R}^d} |\xi||\hat{f}(\xi)|d\xi < \infty. \tag{18}$$

[11] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945.

Barron's space

- Suppose that f satisfies Barron's condition[11]:

$$|f|_{\mathcal{B}} := \int_{\mathbb{R}^d} |\xi||\hat{f}(\xi)|d\xi < \infty. \tag{18}$$

- Then, using Fourier inversion we get for $|x| \leq 1$

$$\begin{aligned} f(x) &= \frac{1}{2\pi}\int_{\mathbb{R}^d} \hat{f}(\xi)e^{i\xi\cdot x}d\xi \\ &= \frac{1}{2\pi}\int_{\mathbb{R}^d} \hat{f}(\xi)d\xi + i\int_{\mathbb{R}^d} \hat{f}(\xi)\int_0^1 |\xi|e^{i|\xi|t}\sigma_0\left(\frac{\xi}{|\xi|}\cdot x - t\right)dtd\xi \\ &= C(f) + i\int_{\mathbb{R}^d} \hat{f}(\xi)\int_0^1 |\xi|e^{i|\xi|t}\sigma_0\left(\frac{\xi}{|\xi|}\cdot x - t\right)dtd\xi \end{aligned} \tag{19}$$

- Integral representation of f or *continuous shallow network*

[11] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945.

Barron's argument

- Total mass of the integral:

$$\int_{\mathbb{R}^d} |\hat{f}(\xi)| \int_0^{-1} |\xi||e^{i|\xi|t}| dt d\xi \leq \int_{\mathbb{R}^d} |\xi||\hat{f}(\xi)| d\xi = |f|_{\mathcal{B}}. \qquad (20)$$

[12] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

Barron's argument

- Total mass of the integral:

$$\int_{\mathbb{R}^d} |\hat{f}(\xi)| \int_0^{-1} |\xi||e^{i|\xi|t}|dtd\xi \leq \int_{\mathbb{R}^d} |\xi||\hat{f}(\xi)|d\xi = |f|_{\mathcal{B}}. \qquad (20)$$

- Next step[12]: approximate the *continuous shallow network* by an element of $\Sigma_n^{\sigma_0}(\mathbb{R}^d)$

[12] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

Convex Hulls of Dictionaries

- Let X be a Banach space

Convex Hulls of Dictionaries

- Let X be a Banach space
- Let $\mathbb{D} \subset X$ be collection of functions (called a dictionary)
  - Assume that $\mathbb{D}$ is bounded, i.e. $|\mathbb{D}| := \sup_{d\in\mathbb{D}} \|d\|_X < \infty$

Convex Hulls of Dictionaries

- Let X be a Banach space
- Let $\mathbb{D} \subset X$ be collection of functions (called a dictionary)
  - Assume that $\mathbb{D}$ is bounded, i.e. $|\mathbb{D}| := \sup_{d\in\mathbb{D}} \|d\|_X < \infty$
- Let $B = B_1(\mathbb{D})$ be the symmetric closed convex hull of $\mathbb{D}$, i.e.

$$B_1(\mathbb{D}) := \overline{\left\{\sum_{j=1}^n a_j h_j : \ n \in \mathbb{N},\ h_j \in \mathbb{D},\ \sum_{i=1}^n |a_i| \leq 1\right\}} \tag{21}$$

  - Here the closure is taken with respect to the norm on X

Convex Dictionary Spaces

- Since $B_1(\mathbb{D})$ is convex, it is the unit ball of the norm

$$\|f\|_{\mathcal{K}_1(\mathbb{D})} = \inf\{r > 0 : f \in rB_1(\mathbb{D})\} \tag{22}$$

  - This is called the *guage* of the set $B_1(\mathbb{D})$
- If $\mathbb{D}$ is bounded, the associated space

$$\mathcal{K}_1(\mathbb{D}) := \{f \in L^2(\Omega) : \|f\|_{\mathcal{K}_1(\mathbb{D})} < \infty\}$$

  is a Banach space[13]
  - Also called *variation space*[14] with respect to $\mathbb{D}$

[13] Jonathan W Siegel and Jinchao Xu. "Characterization of the variation spaces corresponding to shallow neural networks". In: *Constructive Approximation* 57.3 (2023), pp. 1109–1132.

[14] Vera Kurková and Marcello Sanguineti. "Bounds on rates of variable-basis and neural-network approximation". In: *IEEE Transactions on Information Theory* 47.6 (2001), pp. 2659–2665.

Non-linear Dictionary Approximation

- Suppose we want to approximate $f \in \mathcal{K}_1(\mathbb{D})$

Non-linear Dictionary Approximation

- Suppose we want to approximate $f \in \mathcal{K}_1(\mathbb{D})$
- Consider approximation from the set

$$\Sigma_n(\mathbb{D}) := \left\{ \sum_{i=1}^n a_i d_i,\ d_i \in \mathbb{D} \right\} \tag{23}$$

  - This corresponds to non-linear dictionary approximation (note the elements d_i in general depend upon the element f to be approximated)

Non-linear Dictionary Approximation

- Suppose we want to approximate $f \in \mathcal{K}_1(\mathbb{D})$
- Consider approximation from the set

$$\Sigma_n(\mathbb{D}) := \left\{\sum_{i=1}^n a_i d_i,\ d_i \in \mathbb{D}\right\} \tag{23}$$

  - This corresponds to non-linear dictionary approximation (note the elements d_i in general depend upon the element f to be approximated)
- Key question: How efficiently can this be done?
  - Where error is measured in the norm of X

Neural Network Dictionaries

- Consider the dictionary

$$\mathbb{D}_0 := \{\sigma_0(\omega \cdot x + b) : \ \omega \in \mathbb{R}^d, \ b \in \mathbb{R}\} \subset L_p \tag{24}$$

Neural Network Dictionaries

- Consider the dictionary

$$\mathbb{D}_0 := \{\sigma_0(\omega \cdot x + b) : \ \omega \in \mathbb{R}^d, \ b \in \mathbb{R}\} \subset L_p \tag{24}$$

- The integral representation implies that

$$\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} \leq C\|f\|_{\mathcal{B}} \tag{25}$$

- Further, we have

$$\Sigma_n(\mathbb{D}_0) = \Sigma_n^{\sigma_0}(\mathbb{R}^d) \tag{26}$$

- This special case is exactly our shallow network approximation problem

Non-linear Dictionary Approximation of Convex Hulls

- What approximation rates can be achieved for $\Sigma_n(\mathbb{D})$ on $B_1(\mathbb{D})$ with respect to X?

Non-linear Dictionary Approximation of Convex Hulls

- What approximation rates can be achieved for $\Sigma_n(\mathbb{D})$ on $B_1(\mathbb{D})$ with respect to X?
- Suppose that X is a Hilbert space, $\mathbb{D} \subset X$ is a bounded dictionary

Non-linear Dictionary Approximation of Convex Hulls

- What approximation rates can be achieved for $\Sigma_n(\mathbb{D})$ on $B_1(\mathbb{D})$ with respect to X?
- Suppose that X is a Hilbert space, $\mathbb{D} \subset X$ is a bounded dictionary
- If $f \in B_1(\mathbb{D})$, then $f = \sum_{i=1}^N a_i d_i$ with $a_i \geq 0$ and $\sum a_i = 1$.

Non-linear Dictionary Approximation of Convex Hulls

- What approximation rates can be achieved for $\Sigma_n(\mathbb{D})$ on $B_1(\mathbb{D})$ with respect to X?
- Suppose that X is a Hilbert space, $\mathbb{D} \subset X$ is a bounded dictionary
- If $f \in B_1(\mathbb{D})$, then $f = \sum_{i=1}^N a_i d_i$ with $a_i \geq 0$ and $\sum a_i = 1$.
- Define a random variable F with values in X by

$$\mathbb{P}(F = d_i) = a_i.$$

- Note that we have $\mathbb{E}(F) = f$.

Non-linear Dictionary Approximation of Convex Hulls

- What approximation rates can be achieved for $\Sigma_n(\mathbb{D})$ on $B_1(\mathbb{D})$ with respect to X?
- Suppose that X is a Hilbert space, $\mathbb{D} \subset X$ is a bounded dictionary
- If $f \in B_1(\mathbb{D})$, then $f = \sum_{i=1}^N a_i d_i$ with $a_i \geq 0$ and $\sum a_i = 1$.
- Define a random variable F with values in X by

$$\mathbb{P}(F = d_i) = a_i.$$

- Note that we have $\mathbb{E}(F) = f$.
- Construct an approximant f_n by sampling:
  - Let $F_1, ..., F_n$ be independent copies of F and consider the random variable

$$\tilde{F}_n = \frac{1}{n}\sum_{i=1}^n F_i.$$

Non-linear Dictionary Approximation of Convex Hulls

- We clearly have $\mathbb{E}(\tilde{F}_n) = f$ and

$$\mathbb{E}(\|\tilde{F}_n - f\|_X^2) \leq \mathbb{E}(\|\tilde{F}_n - f\|_X^2) \leq \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}(\|F_i\|_X^2), \qquad (27)$$

since we are in a Hilbert space.

  - This argument also works in more general type-2 Banach spaces X, e.g. in $L^p(\Omega)$ for $2 \leq p < \infty$

[15] Gilles Pisier. "Remarques sur un résultat non publié de B. Maurey". In: *Séminaire Analyse fonctionnelle (dit "Maurey-Schwartz")* (1981), pp. 1–12.

Non-linear Dictionary Approximation of Convex Hulls

- We clearly have $\mathbb{E}(\tilde{F}_n) = f$ and

$$\mathbb{E}(\|\tilde{F}_n - f\|_X^2) \leq \mathbb{E}(\|\tilde{F}_n - f\|_X^2) \leq \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}(\|F_i\|_X^2), \tag{27}$$

  since we are in a Hilbert space.
  - This argument also works in more general type-2 Banach spaces X, e.g. in $L^p(\Omega)$ for $2 \leq p < \infty$
- Since $\|F_i\|_X$ is bounded by $\sup_{d\in\mathbb{D}} \|d\|_X$, there must exist a realization $f_n \in \Sigma_n(\mathbb{D})$ such that[15]

$$\|f_n - f\|_X \leq \frac{1}{\sqrt{n}} \sup_{d\in\mathbb{D}} \|d\|_X. \tag{28}$$

[15] Gilles Pisier. “Remarques sur un résultat non publié de B. Maurey”. In: *Séminaire Analyse fonctionnelle (dit “Maurey-Schwartz”)* (1981), pp. 1–12.

Consequences for Neural Networks

- Applying this to neural network approximation[16]:

$$\inf_{f_n \in \Sigma_n^{\sigma_0}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} n^{-\frac{1}{2}} \leq C|f|_{\mathcal{B}} n^{-\frac{1}{2}}. \tag{29}$$

for $2 \leq p < \infty$.

[16] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

[17] Jonathan W Siegel and Jinchao Xu. "Approximation rates for neural networks with general activation functions". In: *Neural Networks* 128 (2020), pp. 313–321.

Consequences for Neural Networks

- Applying this to neural network approximation[16]:

$$\inf_{f_n\in\Sigma_n^{\sigma_0}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} n^{-\frac{1}{2}} \leq C|f|_{\mathcal{B}} n^{-\frac{1}{2}}. \tag{29}$$

for $2 \leq p < \infty$.

- Dimension independent approximation rate!

[16] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

[17] Jonathan W Siegel and Jinchao Xu. "Approximation rates for neural networks with general activation functions". In: *Neural Networks* 128 (2020), pp. 313–321.

Consequences for Neural Networks

- Applying this to neural network approximation[16]:

$$\inf_{f_n \in \Sigma_n^{\sigma_0}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} n^{-\frac{1}{2}} \leq C|f|_{\mathcal{B}} n^{-\frac{1}{2}}. \qquad (29)$$

  for $2 \leq p < \infty$.
  - Dimension independent approximation rate!
- General sigmoidal activation function σ
  - Sigmoidal means $\lim_{t\to-\infty} \sigma(t) = 0$ and $\lim_{t\to\infty} \sigma(t) = 1$

[16] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

[17] Jonathan W Siegel and Jinchao Xu. "Approximation rates for neural networks with general activation functions". In: *Neural Networks* 128 (2020), pp. 313–321.

Consequences for Neural Networks

- Applying this to neural network approximation[16]:

$$\inf_{f_n \in \Sigma_n^{\sigma_0}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} n^{-\frac{1}{2}} \leq C|f|_{\mathcal{B}} n^{-\frac{1}{2}}. \quad (29)$$

  for $2 \leq p < \infty$.
  - Dimension independent approximation rate!
- General sigmoidal activation function σ
  - Sigmoidal means $\lim_{t\to-\infty} \sigma(t) = 0$ and $\lim_{t\to\infty} \sigma(t) = 1$
  - Thus $\sigma(Rt) \to \sigma_0(t)$ as $R \to \infty$

[16] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

[17] Jonathan W Siegel and Jinchao Xu. "Approximation rates for neural networks with general activation functions". In: *Neural Networks* 128 (2020), pp. 313–321.

Consequences for Neural Networks

- Applying this to neural network approximation[16]:

$$\inf_{f_n \in \Sigma_n^{\sigma_0}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} n^{-\frac{1}{2}} \leq C|f|_{\mathcal{B}} n^{-\frac{1}{2}}. \tag{29}$$

  for $2 \leq p < \infty$.
  - Dimension independent approximation rate!
- General sigmoidal activation function σ
  - Sigmoidal means $\lim_{t\to-\infty} \sigma(t) = 0$ and $\lim_{t\to\infty} \sigma(t) = 1$
  - Thus $\sigma(Rt) \to \sigma_0(t)$ as $R \to \infty$
  - Get same approximation rates with σ

[16] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

[17] Jonathan W Siegel and Jinchao Xu. "Approximation rates for neural networks with general activation functions". In: *Neural Networks* 128 (2020), pp. 313–321.

Consequences for Neural Networks

- Applying this to neural network approximation[16]:

$$\inf_{f_n \in \Sigma_n^{\sigma_0}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} n^{-\frac{1}{2}} \leq C|f|_{\mathcal{B}} n^{-\frac{1}{2}}. \tag{29}$$

  for $2 \leq p < \infty$.
  - Dimension independent approximation rate!
- General sigmoidal activation function σ
  - Sigmoidal means $\lim_{t\to-\infty} \sigma(t) = 0$ and $\lim_{t\to\infty} \sigma(t) = 1$
  - Thus $\sigma(Rt) \to \sigma_0(t)$ as $R \to \infty$
  - Get same approximation rates with σ
  - Same result holds for even more general activation functions[17]

[16] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945, Lee K Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". In: *The Annals of Statistics* 20.1 (1992), pp. 608–613.

[17] Jonathan W Siegel and Jinchao Xu. "Approximation rates for neural networks with general activation functions". In: *Neural Networks* 128 (2020), pp. 313–321.

Consequences for Sobolev space approximation

- Using the Cauchy-Schwartz inequality, it follows that

$$
\begin{aligned}
|f|_{\mathcal{B}} &= \int_{\mathbb{R}^d} \frac{|\xi|}{(1+|\xi|)^s}(1+|\xi|)^s|\hat{f}(\xi)|d\xi \\
&\leq C(d,\epsilon)\|f\|_{W^s(L_2)}
\end{aligned}
\tag{30}
$$

for $s = d/2 + 1 + \epsilon$.

Consequences for Sobolev space approximation

- Using the Cauchy-Schwartz inequality, it follows that

$$
\begin{aligned}
|f|_{\mathcal{B}} &= \int_{\mathbb{R}^d} \frac{|\xi|}{(1+|\xi|)^s}(1+|\xi|)^s|\hat{f}(\xi)|d\xi \\
&\leq C(d,\epsilon)\|f\|_{W^s(L_2)}
\end{aligned}
\tag{30}
$$

  for $s = d/2 + 1 + \epsilon$.

- This gives

$$
\inf_{f_n \in \Sigma_n^\sigma(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{W^s(L_2)} n^{-\frac{1}{2}} \tag{31}
$$

  - for any $s > d/2 + 1$
  - for $2 \leq p < \infty$
  - for any sigmoidal activation function

Further improvements

- Embedding $W^s(L_2) \subset \mathcal{B} \subset \mathcal{K}_1(\mathbb{D}_0)$ ($s = d/2 + 1 + \epsilon$) can be improved to
$$W^s(L_2) \subset \mathcal{K}_1(\mathbb{D}_0) \tag{32}$$
for $s = (d+1)/2$
    - Proved using the Radon transform[18]

[18] Tong Mao, Jonathan W Siegel, and Jinchao Xu. "Approximation Rates for Shallow ReLU^k Neural Networks on Sobolev Spaces via the Radon Transform". In: *arXiv preprint arXiv:2408.10996* (2024), Rahul Parhi and Robert D Nowak. "Banach space representer theorems for neural networks and ridge splines". In: *Journal of Machine Learning Research* 22.43 (2021), pp. 1–40.

Further improvements

- Barron's approximation rates can be improved to[19]

$$\inf_{f_n \in \Sigma_n^{\sigma_0}(\mathbb{R}^d)} \|f - f_n\|_{L_\infty} \leq C\|f\|_{\mathcal{K}_1(\mathbb{D}_0)} n^{-\frac{1}{2}-\frac{1}{2d}} \quad (33)$$

  - Can also be extended to ReLU^k activation function[20]

[19] Limin Ma, Jonathan W Siegel, and Jinchao Xu. "Uniform approximation rates and metric entropy of shallow neural networks". In: *Research in the Mathematical Sciences* 9.3 (2022), p. 46, Yuly Makovoz. "Random approximants and neural networks". In: *Journal of Approximation Theory* 85.1 (1996), pp. 98–109.

[20] Jonathan W Siegel and Jinchao Xu. "Sharp bounds on the approximation rates, metric entropy, and n-widths of shallow neural networks". In: *Foundations of Computational Mathematics* 24.2 (2024), pp. 481–537, Jonathan W Siegel. "Optimal approximation of zonoids and uniform approximation by shallow neural networks". In: *Constructive Approximation* (2025), pp. 1–29.

Shallow ReLU^k network approximation

- Putting these results together, we get

$$\inf_{f_n \in \Sigma_n^{\sigma_k}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \leq C\|f\|_{W^s(L_p)} n^{-\frac{s}{d}} \tag{34}$$

  - Here $\sigma_k(x) = \max(0,x)^k$ (when $k = 0$ any sigmoidal activation function if $p < \infty$)
  - $s \leq \frac{d}{2} + k + \frac{1}{2}$
  - $2 \leq p \leq \infty$
  - Extends and improves a variety of existing results[21]

[21] Ronald A DeVore, Konstantin I Oskolkov, and Pencho P Petrushev. "Approximation by feed-forward neural networks". In: *Annals of Numerical Mathematics* 4 (1996), pp. 261–288, Pencho P Petrushev. "Approximation by ridge functions and neural networks". In: *SIAM Journal on Mathematical Analysis* 30.1 (1998), pp. 155–189, Francis Bach. "Breaking the curse of dimensionality with convex neural networks". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 629–681, Yunfei Yang and Ding-Xuan Zhou. "Nonparametric regression using over-parameterized shallow ReLU neural networks". In: *Journal of Machine Learning Research* 25 (2024), pp. 1–35, Yunfei Yang and Ding-Xuan Zhou. "Optimal rates of approximation by shallow ReLU^{k}neural networks and applications to nonparametric regression". In: *Constructive Approximation* (2024), pp. 1–32.

Lower Bounds

- We can also prove nearly matching lower bounds[22]:

$$\sup_{\|f\|_{W^s(L_p)}\leq 1} \inf_{f_n\in\Sigma_n^{\sigma_k}(\mathbb{R}^d)} \|f - f_n\|_{L_p} \geq C(n\log(n))^{-\frac{s}{d}} \tag{35}$$

[22] Tong Mao, Jonathan W Siegel, and Jinchao Xu. "Approximation Rates for Shallow ReLU^k Neural Networks on Sobolev Spaces via the Radon Transform". In: *arXiv preprint arXiv:2408.10996* (2024).

VC-dimension

- Let $\mathcal{F}$ be a class of functions

VC-dimension

- Let $\mathcal{F}$ be a class of functions
- A set of points $x_1, ..., x_N$ is shattered by $\mathcal{F}$ if for any $\epsilon_1, ..., \epsilon_N \in \{\pm 1\}$ there exists an $f \in \mathcal{F}$ such that

$$\text{sign}(f(x_i)) = \epsilon_i \tag{36}$$

VC-dimension

- Let $\mathcal{F}$ be a class of functions
- A set of points $x_1, ..., x_N$ is shattered by $\mathcal{F}$ if for any $\epsilon_1, ..., \epsilon_N \in \{\pm 1\}$ there exists an $f \in \mathcal{F}$ such that

$$\text{sign}(f(x_i)) = \epsilon_i \tag{36}$$

- The VC-dimension of $\mathcal{F}$ is the largest N such that $\mathcal{F}$ shatters a set of N points
  - Degree d polynomials have VC-dimension $d+1$
  - Linear functions on $\mathbb{R}^d$ have VC-dimension $d+1$

Lower Bounds in terms of VC-dim

- Suppose that $\mathcal{F}$ has VC-dimension less than N

[23] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

Lower Bounds in terms of VC-dim

- Suppose that $\mathcal{F}$ has VC-dimension less than N
- Consider a grid of N points $\{0, 1/n, 2/n, ..., (n-1)/n\}^d$ ($n = N^{1/d}$)

[23] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

Lower Bounds in terms of VC-dim

- Suppose that $\mathcal{F}$ has VC-dimension less than N
- Consider a grid of N points $\{0, 1/n, 2/n, ..., (n-1)/n\}^d$ $(n = N^{1/d})$
- We can interpolate the values $c\epsilon_i N^{-s/d}$ by a function $\|f\|_{W^s(L_\infty(\Omega))} \leq 1$
    - Here ϵ_i represent arbitrary signs at the grid points

[23] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

Lower Bounds in terms of VC-dim

- Suppose that $\mathcal{F}$ has VC-dimension less than N
- Consider a grid of N points $\{0, 1/n, 2/n, ..., (n-1)/n\}^d$ ($n = N^{1/d}$)
- We can interpolate the values $c\epsilon_i N^{-s/d}$ by a function $\|f\|_{W^s(L_\infty(\Omega))} \leq 1$
  - Here ϵ_i represent arbitrary signs at the grid points
- VC-dimension bound implies that there exist ϵ_i which cannot be matched

[23] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

Lower Bounds in terms of VC-dim

- Suppose that $\mathcal{F}$ has VC-dimension less than N
- Consider a grid of N points $\{0, 1/n, 2/n, ..., (n-1)/n\}^d$ $(n = N^{1/d})$
- We can interpolate the values $c\epsilon_i N^{-s/d}$ by a function $\|f\|_{W^s(L_\infty(\Omega))} \leq 1$
  - Here ϵ_i represent arbitrary signs at the grid points
- VC-dimension bound implies that there exist ϵ_i which cannot be matched
- So we get

$$\sup_{\|f\|_{W^s(L_\infty(\Omega))}\leq 1} \inf_{g\in\mathcal{F}} \|f-g\|_{L_\infty(\Omega)} \geq cN^{-s/d} \tag{37}$$

  - Can also derive lower bounds[23] in L_p

[23] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

VC-dimension of shallow networks

- VC-dimension of $\Sigma_n^{\sigma_k}(\mathbb{R}^d)$ is[24]

$$\begin{cases} d_{VC}(\Sigma_n^{\sigma_k}(\mathbb{R}^d)) \eqsim n & d = 1 \\ n \lesssim d_{VC}(\Sigma_n^{\sigma_k}(\mathbb{R}^d)) \lesssim n\log(n) & d = 2,3 \\ d_{VC}(\Sigma_n^{\sigma_k}(\mathbb{R}^d)) \eqsim n\log(n) & n \geq 4. \end{cases} \tag{38}$$

  - Lower bounds for shallow ReLU^k networks follow

[24] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation". In: *Acta Numerica* 30 (2021), pp. 327–444.

Open Problems

- What happens with $p < 2$ and σ is ReLU^k?
- What happens for larger values of s?
- What happens in the non-linear regime $q < p$?
- Can we obtain sharp (or nearly sharp) rates for other classes of activation functions?
- What are the right logarithmic factors in the lower bound?
- Can we determine approximation spaces for shallow networks:

$$|f|_{\mathcal{A}(\sigma,\alpha,p)} := \sup_{n\geq 1} n^{\alpha} \left(\inf_{f_n \in \Sigma_n^{\sigma}(\mathbb{R}^d)} \|f - f_n\|_{L_p}\right) \tag{39}$$

  - For $d = 1$ and σ the ReLU^k this is variable knot spline approximation

Deep Neural Network Approximation of Sobolev Functions

- Given a Sobolev class $W^s(L_q(\Omega))$ and an error norm $L^p(\Omega)$, what are the optimal rates of approximation by deep networks:

$$\sup_{\|f\|_{W^s(L_q(\Omega))}\leq 1} \inf_{f_L\in\Upsilon_\sigma^{W,L}(\mathbb{R}^d)} \|f - f_L\|_{L_p(\Omega)} \tag{40}$$

[25] Dmitry Yarotsky. "Elementary superexpressive activations". In: *International conference on machine learning*. PMLR. 2021, pp. 11932–11940, Shijun Zhang, Zuowei Shen, and Haizhao Yang. "Deep network approximation: Achieving arbitrary accuracy with fixed number of neurons". In: *Journal of Machine Learning Research* 23.276 (2022), pp. 1–60.

Deep Neural Network Approximation of Sobolev Functions

- Given a Sobolev class $W^s(L_q(\Omega))$ and an error norm $L^p(\Omega)$, what are the optimal rates of approximation by deep networks:

$$\sup_{\|f\|_{W^s(L_q(\Omega))}\leq 1} \inf_{f_L\in \Upsilon_\sigma^{W,L}(\mathbb{R}^d)} \|f - f_L\|_{L_p(\Omega)} \tag{40}$$

- General activation function σ:
  - There exist finite size neural networks which are dense in $C(\Omega)$![25]

[25] Dmitry Yarotsky. "Elementary superexpressive activations". In: *International conference on machine learning*. PMLR. 2021, pp. 11932–11940, Shijun Zhang, Zuowei Shen, and Haizhao Yang. "Deep network approximation: Achieving arbitrary accuracy with fixed number of neurons". In: *Journal of Machine Learning Research* 23.276 (2022), pp. 1–60.

Deep Neural Network Approximation of Sobolev Functions

- Given a Sobolev class $W^s(L_q(\Omega))$ and an error norm $L^p(\Omega)$, what are the optimal rates of approximation by deep networks:

$$\sup_{\|f\|_{W^s(L_q(\Omega))}\leq 1} \inf_{f_L\in \Upsilon_\sigma^{W,L}(\mathbb{R}^d)} \|f - f_L\|_{L_p(\Omega)} \tag{40}$$

- General activation function σ:
  - There exist finite size neural networks which are dense in $C(\Omega)$![25]
- What about the ReLU activation function?

[25] Dmitry Yarotsky. "Elementary superexpressive activations". In: *International conference on machine learning*. PMLR. 2021, pp. 11932–11940, Shijun Zhang, Zuowei Shen, and Haizhao Yang. "Deep network approximation: Achieving arbitrary accuracy with fixed number of neurons". In: *Journal of Machine Learning Research* 23.276 (2022), pp. 1–60.

What types of functions are in $\Upsilon^{W,L}(\mathbb{R}^d)$

- All functions $f \in \Upsilon^{W,L}(\mathbb{R}^d)$ are continuous and piecewise linear

[26] Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. "ReLU Deep Neural Networks and Linear Finite Elements". In: *Journal of Computational Mathematics* 38.3 (2020), pp. 502–527.

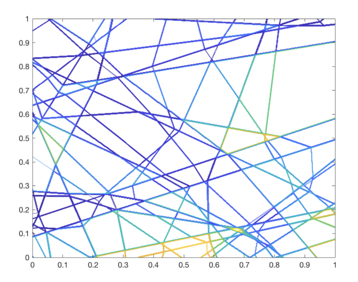
What types of functions are in $\Upsilon^{W,L}(\mathbb{R}^d)$

- All functions $f \in \Upsilon^{W,L}(\mathbb{R}^d)$ are continuous and piecewise linear
- The number of pieces can be exponential in the depth L
  - Number of parameters scales like W^2L

[26] Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. "ReLU Deep Neural Networks and Linear Finite Elements". In: *Journal of Computational Mathematics* 38.3 (2020), pp. 502–527.

What types of functions are in $\Upsilon^{W,L}(\mathbb{R}^d)$

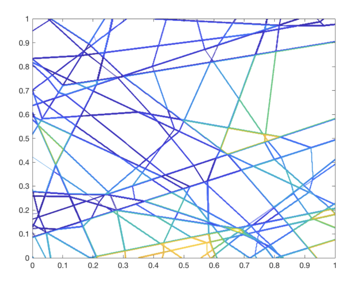
- All functions $f \in \Upsilon^{W,L}(\mathbb{R}^d)$ are continuous and piecewise linear
- The number of pieces can be exponential in the depth L
  - Number of parameters scales like $W^2 L$
- Classical piecewise linear finite element functions can be represented[26]



[26] Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. "ReLU Deep Neural Networks and Linear Finite Elements". In: *Journal of Computational Mathematics* 38.3 (2020), pp. 502–527.

What types of functions are in $\Upsilon^{W,L}(\mathbb{R}^d)$

- All functions $f \in \Upsilon^{W,L}(\mathbb{R}^d)$ are continuous and piecewise linear
- The number of pieces can be exponential in the depth L
  - Number of parameters scales like W^2L
- Classical piecewise linear finite element functions can be represented[26]



- If $L \geq \log_2(d+1)$, get all continuous piecewise linear functions

[26] Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. "ReLU Deep Neural Networks and Linear Finite Elements". In: *Journal of Computational Mathematics* 38.3 (2020), pp. 502–527.

Superconvergence

- A fascinating result discovered by Yarotsky and Shen, Yang, Zhang[27]:

Theorem

Suppose that $p = q = \infty$ and $0 < s \leq 1$. Then $W^s(L_\infty(\Omega))$ is the class of s-Hölder continuous functions. Then for sufficiently large W (depending upon d)

$$\inf_{f_L \in \Upsilon^{W,L}(\mathbb{R}^d)} \|f - f_L\|_{L_\infty(\Omega)} \leq C\|f\|_{W^s(L_\infty(\Omega))} L^{-2s/d}. \tag{41}$$

- This is sharp for deep ReLU networks

[27] Dmitry Yarotsky. "Optimal approximation of continuous functions by very deep ReLU networks". In: *arXiv preprint arXiv:1802.03620* (2018), Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth". In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135.

Superconvergence

- A fascinating result discovered by Yarotsky and Shen, Yang, Zhang[27]:

Theorem

Suppose that $p = q = \infty$ and $0 < s \leq 1$. Then $W^s(L_\infty(\Omega))$ is the class of s-Hölder continuous functions. Then for sufficiently large W (depending upon d)

$$\inf_{f_L \in \Upsilon^{W,L}(\mathbb{R}^d)} \|f - f_L\|_{L_\infty(\Omega)} \leq C\|f\|_{W^s(L_\infty(\Omega))} L^{-2s/d}. \tag{41}$$

- This is sharp for deep ReLU networks
- Classical methods (even nonlinear) can only get a rate of convergence $N^{-s/d}$
  - N is the number of parameters

[27] Dmitry Yarotsky. "Optimal approximation of continuous functions by very deep ReLU networks". In: *arXiv preprint arXiv:1802.03620* (2018), Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth". In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135.

Extensions

- Yarotsky's superconvergence result has been generalized[28] to $s > 1$

[28] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep network approximation for smooth functions". In: *SIAM Journal on Mathematical Analysis* 53.5 (2021), pp. 5465–5506.

[29] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth". In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135.

[30] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation". In: *Acta Numerica* 30 (2021), pp. 327–444.

Extensions

- Yarotsky's superconvergence result has been generalized[28] to $s > 1$
- Optimal approximation rates when both depth and width vary[29]

[28] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep network approximation for smooth functions". In: *SIAM Journal on Mathematical Analysis* 53.5 (2021), pp. 5465–5506.

[29] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth". In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135.

[30] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation". In: *Acta Numerica* 30 (2021), pp. 327–444.

Extensions

- Yarotsky's superconvergence result has been generalized[28] to $s > 1$
- Optimal approximation rates when both depth and width vary[29]
- Results for Sobolev spaces $W^s(L_q)$ with $q < \infty$ have been obtained using interpolation[30]

[28] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep network approximation for smooth functions". In: *SIAM Journal on Mathematical Analysis* 53.5 (2021), pp. 5465–5506.

[29] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth". In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135.

[30] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation". In: *Acta Numerica* 30 (2021), pp. 327–444.

Extensions

- Yarotsky's superconvergence result has been generalized[28] to $s > 1$
- Optimal approximation rates when both depth and width vary[29]
- Results for Sobolev spaces $W^s(L_q)$ with $q < \infty$ have been obtained using interpolation[30]
- What is the optimal rate for all pairs s, p, q for which we have a (compact) embedding?
  - Do we get superconvergence when $q < p \leq \infty$?
  - Are these rates optimal for all s, q, p?

[28] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep network approximation for smooth functions". In: *SIAM Journal on Mathematical Analysis* 53.5 (2021), pp. 5465–5506.

[29] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth". In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135.

[30] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation". In: *Acta Numerica* 30 (2021), pp. 327–444.

Approximating Multiplication[32]

- How can we approximate a product $(x, y) \to xy$?

[31] Matus Telgarsky. "Representation benefits of deep feedforward networks". In: *arXiv preprint arXiv:1509.08101* (2015).

[32] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Approximating Multiplication[32]

- How can we approximate a product $(x, y) \to xy$?
- Consider the hat function:

$$\phi(x) = \max(0, 1 - |2x - 1|) \tag{42}$$

[31] Matus Telgarsky. "Representation benefits of deep feedforward networks". In: *arXiv preprint arXiv:1509.08101* (2015).

[32] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Approximating Multiplication[32]

- How can we approximate a product $(x, y) \to xy$?
- Consider the hat function:

$$\phi(x) = \max(0, 1 - |2x - 1|) \tag{42}$$

- Let $\phi^{\circ k} := \underbrace{\phi \circ \phi \circ \phi \circ \cdots \circ \phi}_{k \ \ times}$

[31] Matus Telgarsky. "Representation benefits of deep feedforward networks". In: *arXiv preprint arXiv:1509.08101* (2015).

[32] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Approximating Multiplication[32]

- How can we approximate a product $(x, y) \to xy$?
- Consider the hat function:

$$\phi(x) = \max(0, 1 - |2x - 1|) \tag{42}$$

- Let $\phi^{\circ k} := \underbrace{\phi \circ \phi \circ \phi \circ \cdots \circ \phi}_{k \ \ times}$
- We have the formula[31]:

$$x^2 = x - \sum_{k=1}^{\infty} \frac{1}{2^{2k}} \phi^{\circ k} \quad x \in [0, 1] \tag{43}$$

[31] Matus Telgarsky. "Representation benefits of deep feedforward networks". In: *arXiv preprint arXiv:1509.08101* (2015).

[32] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Approximating Multiplication[33]

- Truncate this expansion at level k to approximate x^2

[33] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Approximating Multiplication[33]

- Truncate this expansion at level k to approximate x^2
- Use polarization identity

$$xy = \frac{1}{4}[(x+y)^2 - (x-y)^2] \tag{44}$$

[33] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Approximating Multiplication[33]

- Truncate this expansion at level k to approximate x^2
- Use polarization identity

$$xy = \frac{1}{4}[(x+y)^2 - (x-y)^2] \tag{44}$$

Proposition

Let $k \geq 1$. Then there exists a network $f_k \in \Upsilon^{13,6k+3}(\mathbb{R}^2)$ such that for all $x, y \in [-1,1]$ we have

$$|f_k(x,y) - xy| \leq 6 \cdot 4^{-k}. \tag{45}$$

[33] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Bit Extraction

- The key to superconvergence is the *bit-extraction* technique[34]

[34] Peter Bartlett, Vitaly Maiorov, and Ron Meir. "Almost linear VC dimension bounds for piecewise polynomial networks". In: *Advances in neural information processing systems* 11 (1998).

[35] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Bit Extraction

- The key to superconvergence is the *bit-extraction* technique[34]
- Suppose that $\mathbf{x} \in \{0,1\}^N$

[34] Peter Bartlett, Vitaly Maiorov, and Ron Meir. "Almost linear VC dimension bounds for piecewise polynomial networks". In: *Advances in neural information processing systems* 11 (1998).

[35] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Bit Extraction

- The key to superconvergence is the *bit-extraction* technique[34]
- Suppose that $\mathbf{x} \in \{0,1\}^N$
- How many parameters do we need to represent $\mathbf{x}$?
  - i.e. want a network f, s.t. $f(i) = \mathbf{x}_i$ for $i = 0, ..., N-1$.

[34] Peter Bartlett, Vitaly Maiorov, and Ron Meir. "Almost linear VC dimension bounds for piecewise polynomial networks". In: *Advances in neural information processing systems* 11 (1998).

[35] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Bit Extraction

- The key to superconvergence is the *bit-extraction* technique[34]
- Suppose that $\mathbf{x} \in \{0,1\}^N$
- How many parameters do we need to represent $\mathbf{x}$?
  - i.e. want a network f, s.t. $f(i) = \mathbf{x}_i$ for $i = 0, ..., N-1$.
- Naively, we would need $O(N)$ parameters
  - Say use a piecewise linear function

[34] Peter Bartlett, Vitaly Maiorov, and Ron Meir. "Almost linear VC dimension bounds for piecewise polynomial networks". In: *Advances in neural information processing systems* 11 (1998).

[35] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Bit Extraction

- The key to superconvergence is the *bit-extraction* technique[34]
- Suppose that $\mathbf{x} \in \{0,1\}^N$
- How many parameters do we need to represent $\mathbf{x}$?
  - i.e. want a network f, s.t. $f(i) = \mathbf{x}_i$ for $i = 0, ..., N-1$.
- Naively, we would need $O(N)$ parameters
  - Say use a piecewise linear function
- Remarkably, we only need $O(\sqrt{N})$!

[34] Peter Bartlett, Vitaly Maiorov, and Ron Meir. "Almost linear VC dimension bounds for piecewise polynomial networks". In: *Advances in neural information processing systems* 11 (1998).

[35] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Bit Extraction

- The key to superconvergence is the *bit-extraction* technique[34]
- Suppose that $\mathbf{x} \in \{0,1\}^N$
- How many parameters do we need to represent $\mathbf{x}$?
  - i.e. want a network f, s.t. $f(i) = \mathbf{x}_i$ for $i = 0,...,N-1$.
- Naively, we would need $O(N)$ parameters
  - Say use a piecewise linear function
- Remarkably, we only need $O(\sqrt{N})$!
- Superconvergence proved by combining bit-extraction with a piecewise polynomial approximation[35] on a *regular* grid

[34] Peter Bartlett, Vitaly Maiorov, and Ron Meir. "Almost linear VC dimension bounds for piecewise polynomial networks". In: *Advances in neural information processing systems* 11 (1998).

[35] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

Bit Extraction (cont.)

- Divide $\{0, 1, ..., N-1\}$ into $O(\sqrt{N})$ sub-intervals of $I_1, ..., I_n$ of length $O(\sqrt{N})$
  - $I_j = \{k_j, k_j + 1, ..., k_{j+1} - 1\}$

Bit Extraction (cont.)

- Divide $\{0, 1, ..., N-1\}$ into $O(\sqrt{N})$ sub-intervals of $I_1, ..., I_n$ of length $O(\sqrt{N})$
  - $I_j = \{k_j, k_j + 1, ..., k_{j+1} - 1\}$
- Two piecewise linear functions:
  - Map I_j to k_j
  - Map I_j to $b_j = 0.\mathbf{x}_{k_j} ... \mathbf{x}_{k_{j+1}-1}$
  - Requires $O(\sqrt{N})$ layers

Bit Extraction (cont.)

- Divide $\{0,1,...,N-1\}$ into $O(\sqrt{N})$ sub-intervals of $I_1,...,I_n$ of length $O(\sqrt{N})$
  - $I_j = \{k_j, k_j+1, ..., k_{j+1}-1\}$
- Two piecewise linear functions:
  - Map I_j to k_j
  - Map I_j to $b_j = 0.\mathbf{x}_{k_j}...\mathbf{x}_{k_{j+1}-1}$
  - Requires $O(\sqrt{N})$ layers
- Construct network which maps

$$\begin{pmatrix} i \\ k \\ 0.x_1x_2\cdots x_n \\ z \end{pmatrix} \to \begin{pmatrix} i-1 \\ k \\ 0.x_2\cdots x_n \\ z + x_1\chi(i=k) \end{pmatrix} \tag{46}$$

  - Can be done with a constant size network
  - Compose this $O(\sqrt{N})$ times

Efficient Representation of Sparse Vectors[36]

- Approximation in non-linear regime ($q < p$) requires *adaptivity* or *sparsity*

[36] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

Efficient Representation of Sparse Vectors[36]

- Approximation in non-linear regime ($q < p$) requires *adaptivity* or *sparsity*

Proposition

Let $M \geq 1$ and $N \geq 1$ and $\mathbf{x} \in \mathbb{Z}^N$ be an N-dimensional vector satisfying

$$\|\mathbf{x}\|_{\ell^1} \leq M. \tag{47}$$

- *Then if $N \geq M$, there exists a neural network $g \in \Upsilon^{17,L}(\mathbb{R},\mathbb{R})$ with depth $L \leq C\sqrt{M(1+\log(N/M))}$ which satisfies $g(i) = \mathbf{x}_i$ for $i = 1, ..., N$.*
- *Further, if $N < M$, then there exists a neural network $g \in \Upsilon^{17,L}(\mathbb{R},\mathbb{R})$ with depth $L \leq C\sqrt{N(1+\log(M/N))}$ which satisfies $g(i) = \mathbf{x}_i$ for $i = 1, ..., N$.*

[36] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

Main Result: Upper Bounds[37]

Theorem

Let $\Omega = [0,1]^d$ *be the unit cube and let* $0 < s < \infty$ *and* $1 \leq q \leq p \leq \infty$. *Assume that* $1/q - 1/p < s/d$, *which guarantees that we have the compact Sobolev embedding*

$$W^s(L_q(\Omega)) \subset\subset L_p(\Omega). \tag{48}$$

Then there exists an absolute constant $K < \infty$ *and such that*

$$\inf_{f_L \in \Upsilon^{Kd,L}(\mathbb{R}^d)} \|f - f_L\|_{L_p(\Omega)} \lesssim \|f\|_{W^s(L_q(\Omega))} L^{-2s/d}. \tag{49}$$

- Same super-convergence phenomenon for all Sobolev spaces and all error norms if we have compact embedding

[37] Jonathan W Siegel. "Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces". In: *Journal of Machine Learning Research* 24.357 (2023), pp. 1–52.

VC-dimension of deep ReLU networks

- The VC-dimension of $\Upsilon^{W,L}(\mathbb{R}^d)$ is bounded by[38]

$$C\min(W^2\log(WL)L^2, P^2) \leq CP^2 \tag{50}$$

  - Bound is attained for deep narrow networks
- Implies that superconvergence is optimal
  - approximation rate is lower bounded by $P^{-2s/d}$

[38] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. "Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks". In: *Journal of Machine Learning Research* 20.63 (2019), pp. 1–17, Paul Goldberg and Mark Jerrum. "Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers". In: *Proceedings of the sixth annual conference on Computational learning theory*. 1993, pp. 361–369.

Fundamental Lower Bound: 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

$$\epsilon_n(B)_X = \inf\{\epsilon : \ B \text{ is covered by } 2^n \text{ balls of radius } \epsilon\}. \tag{51}$$

- Roughly speaking, $\epsilon_n(B)_K$ measures how accurately elements of B can be specified with n bits.

[39] Albert Cohen, Ronald Devore, Guergana Petrova, and Przemyslaw Wojtaszczyk. "Optimal stable nonlinear approximation". In: *Foundations of Computational Mathematics* (2021), pp. 1–42.

[40] M Š Birman and MZ Solomjak. "Piecewise-polynomial approximations of functions of the classes W_p^α". In: *Mathematics of the USSR-Sbornik* 2.3 (1967), p. 295.

Fundamental Lower Bound: 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

$$\epsilon_n(B)_X = \inf\{\epsilon : \ B \text{ is covered by } 2^n \text{ balls of radius } \epsilon\}. \tag{51}$$

- Roughly speaking, $\epsilon_n(B)_K$ measures how accurately elements of B can be specified with n bits.
- Gives a fundamental lower bound on the rates of stable approximation[39]

[39] Albert Cohen, Ronald Devore, Guergana Petrova, and Przemyslaw Wojtaszczyk. "Optimal stable nonlinear approximation". In: *Foundations of Computational Mathematics* (2021), pp. 1–42.

[40] M Š Birman and MZ Solomjak. "Piecewise-polynomial approximations of functions of the classes W_p^{α}". In: *Mathematics of the USSR-Sbornik* 2.3 (1967), p. 295.

Fundamental Lower Bound: 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

$$\epsilon_n(B)_X = \inf\{\epsilon : \ B \text{ is covered by } 2^n \text{ balls of radius } \epsilon\}. \tag{51}$$

- Roughly speaking, $\epsilon_n(B)_K$ measures how accurately elements of B can be specified with n bits.
- Gives a fundamental lower bound on the rates of stable approximation[39]
- If compact Sobolev embedding holds, then[40]

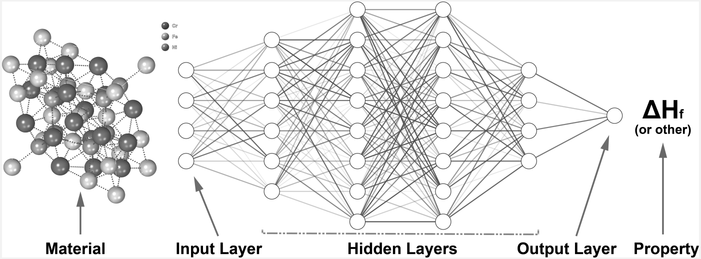
$$\epsilon_n(B^s(L_q(\Omega)))_{L^p(\Omega)} \eqsim n^{-s/d} \tag{52}$$

[39] Albert Cohen, Ronald Devore, Guergana Petrova, and Przemyslaw Wojtaszczyk. "Optimal stable nonlinear approximation". In: *Foundations of Computational Mathematics* (2021), pp. 1–42.

[40] M Š Birman and MZ Solomjak. "Piecewise-polynomial approximations of functions of the classes W_p^α". In: *Mathematics of the USSR-Sbornik* 2.3 (1967), p. 295.

Deep Learning for Science

- Recently, neural networks have been widely applied to scientific problems
  - e.g. protein folding[41], modeling quantum systems[42], predicting materials properties, etc.



[41] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[42] Giuseppe Carleo and Matthias Troyer. "Solving the quantum many-body problem with artificial neural networks". In: *Science* 355.6325 (2017), pp. 602–606.

Symmetries and Continuity

- Nature's laws typically satisfy known symmetries:
  - Translations (left action of $\mathbb{R}^d$)
  - Rotations (left action of $SO(d)$)
  - Orthogonal transformations (left action of $O(d)$)
  - Identical particles (right action of S_n)
  - Lorentz transformations (left action of $O(3,1)$)
  - etc.

Symmetries and Continuity

- Nature's laws typically satisfy known symmetries:
  - Translations (left action of $\mathbb{R}^d$)
  - Rotations (left action of $SO(d)$)
  - Orthogonal transformations (left action of $O(d)$)
  - Identical particles (right action of S_n)
  - Lorentz transformations (left action of $O(3,1)$)
  - etc.
- We would like to build these symmetries into the neural network
  - Field of geometric deep learning

Action of Permutations

- Consider point cloud inputs

$$X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{d\times n}$$

  - The points x_i are the positions of indistinguishable particles

Action of Permutations

- Consider point cloud inputs

$$X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{d \times n}$$

  - The points x_i are the positions of indistinguishable particles
  - The order of the x_i's doesn't matter
  - Can view the input as the set $\{x_1, ..., x_n\}$ of positions

Action of Permutations

- Consider point cloud inputs

$$X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{d\times n}$$

  - The points x_i are the positions of indistinguishable particles
  - The order of the x_i's doesn't matter
  - Can view the input as the set $\{x_1, ..., x_n\}$ of positions
- The permutation action of $\sigma \in S_n$ on $\mathbb{R}^{d\times n}$ given by

$$\sigma \cdot X = [x_{\sigma^{-1}(1)}, ..., x_{\sigma^{-1}(n)}] \tag{53}$$

  - We want our neural network function f to be invariant:

$$f(\sigma \cdot X) = f(X) \tag{54}$$

Invariant Neural Networks

- There are numerous ways to obtain invariant neural networks:
  - Methods based on transforming the input and averaging
    - Canonicalization or Weighted Frames[43]
  - Specialized architectures which parameterize invariant functions
    - Deep Sets[44] or Transformers[45] for permutations

---

[43] Omri Puny, Matan Atzmon, Edward J Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. "Frame Averaging for Invariant and Equivariant Network Design". In: *International Conference on Learning Representations*, Nadav Dym, Hannah Lawrence, and Jonathan W. Siegel. "Equivariant Frames and the Impossibility of Continuous Canonicalization". In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, 21–27 Jul 2024, pp. 12228–12267.

[44] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in neural information processing systems* 30 (2017).

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

Deep Sets

- The Deep Sets architecture is given by[46]:

$$f_\theta(X) = \rho\left(\sum_{i=1}^{n} \Phi(x_i)\right) \tag{55}$$

  - $X = (x_1, ..., x_n) \in \mathbb{R}^{d\times n}$ is an input point cloud
  - $\Phi : \mathbb{R}^d \to \mathbb{R}^N$ and $\rho : \mathbb{R}^N \to \mathbb{R}$ are multilayer-perceptrons
  - θ are the parameters of both Φ and ρ
- Deep Sets parameterizes permutation invariant functions

[46] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in neural information processing systems* 30 (2017).

Deep Sets

- The Deep Sets architecture is given by[46]:

$$f_\theta(X) = \rho\left(\sum_{i=1}^{n} \Phi(x_i)\right) \tag{55}$$

  - $X = (x_1, ..., x_n) \in \mathbb{R}^{d\times n}$ is an input point cloud
  - $\Phi : \mathbb{R}^d \to \mathbb{R}^N$ and $\rho : \mathbb{R}^N \to \mathbb{R}$ are multilayer-perceptrons
  - θ are the parameters of both Φ and ρ
- Deep Sets parameterizes permutation invariant functions
- Key Questions:
  - Universality: Can all (continuous) permutation invariant functions be approximated?
  - Approximation Rates: How efficiently can they be approximated?

[46] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in neural information processing systems* 30 (2017).

Universality of Deep Sets[47]

Theorem

Let $\Omega = [0,1]^d$. Then for sufficiently large N the following holds. For any permutation invariant continuous function $f : \Omega^n \to \mathbb{R}$ and $\epsilon > 0$, there are continuous functions $\rho : \mathbb{R}^N \to \mathbb{R}$ and $\Phi : \mathbb{R}^d \to \mathbb{R}^N$ such that

$$\left| f(X) - \rho\left(\sum_{i=1}^{n} \Phi(x_i)\right)\right| < \epsilon \tag{56}$$

for all $X = (x_1, ..., x_n) \in \Omega$.

[47] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in neural information processing systems* 30 (2017), Nadav Dym and Steven J Gortler. "Low-dimensional invariant embeddings for universal geometric learning". In: *Foundations of Computational Mathematics* 25.2 (2025), pp. 375–415, Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Michael A Osborne, and Ingmar Posner. "Universal approximation of functions on sets". In: *Journal of Machine Learning Research* 23.151 (2022), pp. 1–56.

Embedding Dimension Bounds

- How large does the embedding dimension N need to be for universality?

[48] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in neural information processing systems* 30 (2017).

[49] Nadav Dym and Steven J Gortler. "Low-dimensional invariant embeddings for universal geometric learning". In: *Foundations of Computational Mathematics* (2024), pp. 1–41.

[50] Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Michael A Osborne, and Ingmar Posner. "Universal approximation of functions on sets". In: *Journal of Machine Learning Research* 23.151 (2022), pp. 1–56.

Embedding Dimension Bounds

- How large does the embedding dimension N need to be for universality?
- Upper bounds:
  - $N = n$ when $d = 1$[48]
  - $N = 2nd + 1$ for $d > 1$[49]
- Lower Bounds:
  - When $d = 1$, $N = n$ is necessary[50]

[48] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in neural information processing systems* 30 (2017).

[49] Nadav Dym and Steven J Gortler. "Low-dimensional invariant embeddings for universal geometric learning". In: *Foundations of Computational Mathematics* (2024), pp. 1–41.

[50] Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Michael A Osborne, and Ingmar Posner. "Universal approximation of functions on sets". In: *Journal of Machine Learning Research* 23.151 (2022), pp. 1–56.

Approximation Rates for Deep Sets

- How large do the networks ρ and Φ have to be?
  - Can we get quantitative *approximation rates*?
  - How do these rates compare with non-invariant architectures, i.e., standard MLPs?

Approximation Rates for Deep Sets

- How large do the networks ρ and Φ have to be?
  - Can we get quantitative *approximation rates*?
  - How do these rates compare with non-invariant architectures, i.e., standard MLPs?
- Need additional assumptions on the target function f:
  - Assume that f is Lipschitz, i.e.,

$$|f(x) - f(y)| \leq |x - y| \text{ or } |\nabla f(x)| \leq 1 \tag{57}$$

for $x \in \Omega^n$.

Approximation Rates for Deep Sets

- How large do the networks ρ and Φ have to be?
  - Can we get quantitative *approximation rates*?
  - How do these rates compare with non-invariant architectures, i.e., standard MLPs?
- Need additional assumptions on the target function f:
  - Assume that f is Lipschitz, i.e.,

$$|f(x) - f(y)| \leq |x - y| \text{ or } |\nabla f(x)| \leq 1 \tag{57}$$

    for $x \in \Omega^n$.
- Existing methods for universal approximation:
  - Lack control on the functions ρ and Φ in terms of f and thus do not lead to rates

Approximation Rates for Deep Sets

- Using a different method, we can prove:

Theorem

Let $d, n \geq 2$, $\Omega = [0,1]^d$, $f : \Omega^n \rightarrow \mathbb{R}$ *a Lipschitz permutation invariant function, and* $0 < \epsilon \leq 1$*. Then for* $N = 2nd + 1$ *there exist ReLU neural networks* $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^N$ *and* $\rho : \mathbb{R}^N \rightarrow \mathbb{R}$ *with a total number of parameters* $P \leq C\epsilon^{-dn/2}(1 + |\log \epsilon|)$*, such that*

$$\left| f(X) - \rho\left(\sum_{i=1}^{n} \Phi(x_i)\right) \right| \leq \epsilon \tag{58}$$

for every $X = (x_1, ..., x_n) \in \Omega^n$.

Comparison between Deep Sets and MLP

- For Deep Sets the problem dimension is $D = nd$
- For a Lipschitz function f to achieve accuracy ϵ we need
  - $P = O(\epsilon^{-D/2})$ parameters with a general MLP
  - $P = O(\epsilon^{-D/2}(1 + |\log \epsilon|))$ parameters with Deep Sets if f is permutation invariant

Comparison between Deep Sets and MLP

- For Deep Sets the problem dimension is $D = nd$
- For a Lipschitz function f to achieve accuracy ϵ we need
  - $P = O(\epsilon^{-D/2})$ parameters with a general MLP
  - $P = O(\epsilon^{-D/2}(1 + |\log \epsilon|))$ parameters with Deep Sets if f is permutation invariant
- Up to a logarithmic factor, Deep Sets requires the same number of parameters
  - There is no loss of expressivity when using Deep Sets!

Open Problems

- Determine minimal embedding dimension for universality
- Analyze invariant architectures for rotations and orthogonal transformations
- How can we properly analyze transformers?
- etc.