Approximation Theory

Philipp Grohs

universität wien
Faculty of Mathematics

Bedlewo, November 2019

Short Reading List

1. Helmut Bölcskei, Philipp Grohs, Gitta Kutyniok, Philipp Petersen: Optimal Approximation with Sparsely Connected Deep Neural Networks; SIAM Journal on Mathematics of Data Science, 2019
2. Philipp Grohs, Dmitry Perekreshtenko, Dennis Elbrächter, Helmut Bölcskei: Deep NN Approximation Theory; IEEE Transactions on Information Theory, 2020
3. Philipp Petersen, Felix Voigtländer: Optimal approximation of piecewise smooth functions using deep ReLU neural networks; Neural Networks, 2018
4. DmitryYarotsky: Error bounds for approximations with deep ReLU networks; Neural Networks, 2017
5. Dennis Elbrächter, Philipp Grohs, Arnulf Jentzen, Christoph Schwab: DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing; arXiv:1809.07669

Basic Notions

Basic Notions

- Let $L, N_0, \ldots, N_L \in \mathbb{N}$. A neural network (NN) Φ with L layers is a finite sequence of matrix-vector tuples
$\Phi := ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \in \times_{l=1}^{L} \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$.

Basic Notions

- Let $L, N_0, \ldots, N_L \in \mathbb{N}$. A neural network (NN) Φ with L layers is a finite sequence of matrix-vector tuples
  $\Phi := ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \in \times_{l=1}^{L} \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$.
- $\mathrm{arch}(\Phi) := (N_0, N_1, \ldots, N_L)$.

Basic Notions

- Let $L, N_0, \ldots, N_L \in \mathbb{N}$. A neural network (NN) Φ with L layers is a finite sequence of matrix-vector tuples
  $\Phi := ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \in \times_{l=1}^{L} \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$.
- $\mathrm{arch}(\Phi) := (N_0, N_1, \ldots, N_L)$.
- For $\sigma \in C(\mathbb{R}, \mathbb{R})$ we define the realization of Φ with activation function σ as the map $R_\sigma(\Phi) \in C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$ with $R_\sigma(\Phi)(x) = x_L$, where x_L is given by the following scheme:

$$\begin{aligned} x_0 := x, \quad x_l &:= \sigma(A_l x_{l-1} + b_l), \text{ for } l \in \{1, \ldots, L-1\}, \\ x_L &:= A_L x_{L-1} + b_L. \end{aligned}$$

Basic Notions

- Let $L, N_0, \ldots, N_L \in \mathbb{N}$. A neural network (NN) Φ with L layers is a finite sequence of matrix-vector tuples
  $\Phi := ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \in \times_{l=1}^{L} \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$.
- $\mathrm{arch}(\Phi) := (N_0, N_1, \ldots, N_L)$.
- For $\sigma \in C(\mathbb{R}, \mathbb{R})$ we define the realization of Φ with activation function σ as the map $R_\sigma(\Phi) \in C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$ with $R_\sigma(\Phi)(x) = x_L$, where x_L is given by the following scheme:

$$x_0 := x, \quad x_l := \sigma(A_l x_{l-1} + b_l), \text{ for } l \in \{1, \ldots, L-1\},$$
$$x_L := A_L x_{L-1} + b_L.$$

- $ReLU(x) := \max\{x, 0\}$.

Basic Notions

- Let $L, N_0, \ldots, N_L \in \mathbb{N}$. A neural network (NN) Φ with L layers is a finite sequence of matrix-vector tuples $\Phi := ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \in \times_{l=1}^{L} \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$.
- $\mathrm{arch}(\Phi) := (N_0, N_1, \ldots, N_L)$.
- For $\sigma \in C(\mathbb{R}, \mathbb{R})$ we define the realization of Φ with activation function σ as the map $R_\sigma(\Phi) \in C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$ with $R_\sigma(\Phi)(x) = x_L$, where x_L is given by the following scheme:

$$x_0 := x, \quad x_l := \sigma(A_l x_{l-1} + b_l), \text{ for } l \in \{1, \ldots, L-1\},$$
$$x_L := A_L x_{L-1} + b_L.$$

- $ReLU(x) := \max\{x, 0\}$.
- $\mathcal{H}^{\sigma}_{(N_0,\ldots,N_L)} := \{R_\sigma(\Phi) : \mathrm{arch}(\phi) = (N_0, \ldots, N_L)\} \subset C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$

Basic Notions

- Let $L, N_0, \dots, N_L \in \mathbb{N}$. A neural network (NN) Φ with L layers is a finite sequence of matrix-vector tuples $\Phi := ((A_1, b_1), (A_2, b_2), \dots, (A_L, b_L)) \in \times_{l=1}^{L} \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$.
- $\mathrm{arch}(\Phi) := (N_0, N_1, \dots, N_L)$.
- For $\sigma \in C(\mathbb{R}, \mathbb{R})$ we define the realization of Φ with activation function σ as the map $R_\sigma(\Phi) \in C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$ with $R_\sigma(\Phi)(x) = x_L$, where x_L is given by the following scheme:

$$x_0 := x, \quad x_l := \sigma(A_l x_{l-1} + b_l), \text{ for } l \in \{1, \dots, L-1\},$$
$$x_L := A_L x_{L-1} + b_L.$$

- $ReLU(x) := \max\{x, 0\}$.
- $\mathcal{H}^{\sigma}_{(N_0, \dots, N_L)} := \{R_\sigma(\Phi) : \ \mathrm{arch}(\phi) = (N_0, \dots, N_L)\} \subset C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$
- $\dim(N_0, \dots, N_L) := \sum_{l=1}^{L} (N_l \cdot N_{l-1} + N_l)$.

Basic Notions

- Let $L, N_0, \ldots, N_L \in \mathbb{N}$. A neural network (NN) Φ with L layers is a finite sequence of matrix-vector tuples $\Phi := ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \in \times_{l=1}^{L} \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$.
- $\mathrm{arch}(\Phi) := (N_0, N_1, \ldots, N_L)$.
- For $\sigma \in C(\mathbb{R}, \mathbb{R})$ we define the realization of Φ with activation function σ as the map $R_\sigma(\Phi) \in C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$ with $R_\sigma(\Phi)(x) = x_L$, where x_L is given by the following scheme:

$$
\begin{aligned}
x_0 := x, \quad x_l := \sigma(A_l x_{l-1} + b_l), \ \text{for } l \in \{1, \ldots, L-1\},\\
x_L := A_L x_{L-1} + b_L.
\end{aligned}
$$

- $ReLU(x) := \max\{x, 0\}$.
- $\mathcal{H}^{\sigma}_{(N_0,\ldots,N_L)} := \{R_\sigma(\Phi) : \ \mathrm{arch}(\phi) = (N_0, \ldots, N_L)\} \subset C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$
- $\dim(N_0, \ldots, N_L) := \sum_{l=1}^{L} (N_l \cdot N_{l-1} + N_l)$.
- $\mathrm{size}(\Phi) = \dim(\mathrm{arch}(\Phi)), \quad \mathrm{size}_0(\Phi) = \sum_{i=1}^{L} (\|A_i\|_0 + \|b_i\|_0)$.

Universal Approximation Theorem

Theorem (Cybenko (1989), Hornik (1989), Pinkus (1993))

Let $U \in C(\mathbb{R}^d, \mathbb{R}^k)$, $\sigma \in C(\mathbb{R}, \mathbb{R})$ *not a polynomial,* $L \in \mathbb{N}$ *with* $L > 1$, $K \subset \mathbb{R}^d$ *compact and* $\epsilon > 0$. *Then there is* $N_1, \ldots, N_{L-1} \in \mathbb{N}$ *and* $\Phi \in \mathcal{H}^{\sigma}_{(d, N_1, \ldots, N_{L-1}, k)}$ *with*

$$\sup_{x \in K} |U(x) - R_\sigma(\Phi)(x)| \leq \epsilon.$$

Universal Approximation Theorem

Theorem (Cybenko (1989), Hornik (1989), Pinkus (1993))

Let $U \in C(\mathbb{R}^d, \mathbb{R}^k)$, $\sigma \in C(\mathbb{R}, \mathbb{R})$ not a polynomial, $L \in \mathbb{N}$ with $L > 1$, $K \subset \mathbb{R}^d$ compact and $\epsilon > 0$. Then there is $N_1, \ldots, N_{L-1} \in \mathbb{N}$ and $\Phi \in \mathcal{H}^{\sigma}_{(d,N_1,\ldots,N_{L-1},k)}$ with

$$\sup_{x \in K} |U(x) - R_\sigma(\Phi)(x)| \leq \epsilon.$$

Quantitatively optimal results for classical smoothness spaces have been established in the 90's, for instance in [Hornik etal (1995), Barron (1993), Chui etal (1994), DeVore etal (1997), Mhaskar (1996)].

Universal Universal Approximation Theorem

Universal Universal Approximation Theorem

- Let $K \subset \mathbb{R}^d$ compact. A family of continous coefficient mappings $\mathcal{A}_N : L^2(K) \to \mathbb{R}^N$ and reconstruction mappings $\mathcal{R}_N : \mathbb{R}^N \to L^2(K)$ for $N \in \mathbb{N}$ is called *approximation method*.

Universal Universal Approximation Theorem

- Let $K \subset \mathbb{R}^d$ compact. A family of continous coefficient mappings $\mathcal{A}_N : L^2(K) \to \mathbb{R}^N$ and reconstruction mappings $\mathcal{R}_N : \mathbb{R}^N \to L^2(K)$ for $N \in \mathbb{N}$ is called *approximation method*.
- Example: $\mathcal{A}_N(U) =$ first N wavelet coefficients of U.

Universal Universal Approximation Theorem

- Let $K \subset \mathbb{R}^d$ compact. A family of continous coefficient mappings $\mathcal{A}_N : L^2(K) \to \mathbb{R}^N$ and reconstruction mappings $\mathcal{R}_N : \mathbb{R}^N \to L^2(K)$ for $N \in \mathbb{N}$ is called *approximation method*.
- Example: $\mathcal{A}_N(U) =$ first N wavelet coefficients of U.

Bold Statement

Suppose that $(\mathcal{A}_N, \mathcal{R}_N)_{N\in\mathbb{N}}$ defines any known approximation method. Let $\sigma \in C(\mathbb{R}, \mathbb{R})$ not be a polynomial. Then, for every $N \in \mathbb{N}$ and $U \in L^2(K)$ there is a NN Φ_N with
$\mathrm{size}(\Phi_N) \leq N \cdot \mathrm{polylog}(N)$ and
$\|U - R_\sigma(\Phi_N)\|_{L^2(K)} \leq \|U - \mathcal{R}_N \circ \mathcal{A}_N(U)\|_{L^2(K)}$.

ReLU Calculus: Identity

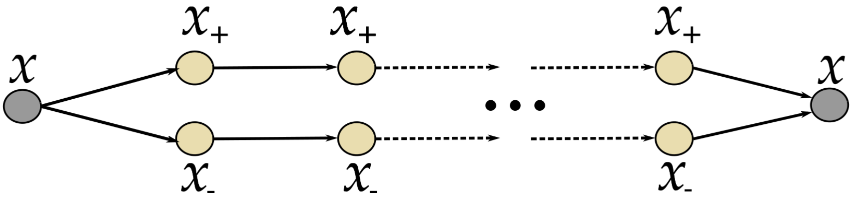
$$ReLU(x) := \max\{x, 0\},$$

$$x = ReLU(x) + ReLU(-x).$$

ReLU Calculus: Identity

$$ReLU(x) := \max\{x, 0\},$$

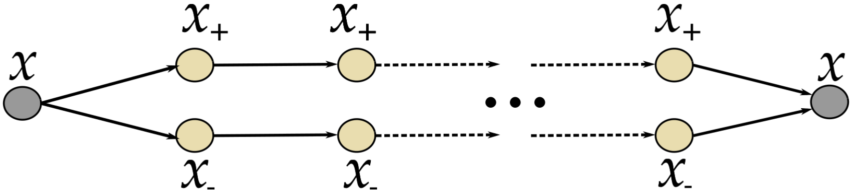
$$x = ReLU(x) + ReLU(-x).$$



ReLU Calculus: Identity

$$ReLU(x) := \max\{x, 0\},$$

$$x = ReLU(x) + ReLU(-x).$$



Reproducing Identity

The Identiy $Id_d : \mathbb{R}^d \to \mathbb{R}^d$, $Id_d(x) = x$ can be represented exactly by a NN Φ_{id} having L layers and $\mathcal{O}(Ld)$ nodes, i.e. $Id_d = R_{ReLU}(\Phi_{id})$.

ReLU Calculus: Composition

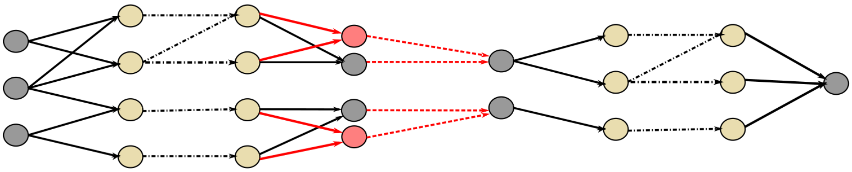
Composition

Let Φ_1, Φ_2 neural networks such that the output dimension d' of Φ_1 equals the input dimension of Φ_2 with L_1, resp. L_2 layers and M_1, resp. M_2 nodes. Then the composition $R_{ReLU}(\Phi_2) \circ R_{ReLU}(\Phi_1)$ can be representad exactly by a NN Φ with $L_1 + L_2$ layers and $M_1 + M_2 + d'$ nodes.

ReLU Calculus: Composition

Composition

Let Φ_1, Φ_2 neural networks such that the output dimension d' of Φ_1 equals the input dimension of Φ_2 with L_1, resp. L_2 layers and M_1, resp. M_2 nodes. Then the composition $R_{ReLU}(\Phi_2) \circ R_{ReLU}(\Phi_1)$ can be representad exactly by a NN Φ with $L_1 + L_2$ layers and $M_1 + M_2 + d'$ nodes.



ReLU Calculus: Increasing the Depth

Increasing the Depth

Let Φ_1 be a NN with L_1 layers, M_1 nodes and output dimension d'. Then for every $L_2 > L_1$ there is a NN Φ_2 with L_2 layers and $M_1 + \mathcal{O}((L_2 - L_1)d')$ nodes and $R_{ReLU}(\Phi_1) = R_{ReLU}(\Phi_2)$

ReLU Calculus: Parallelization

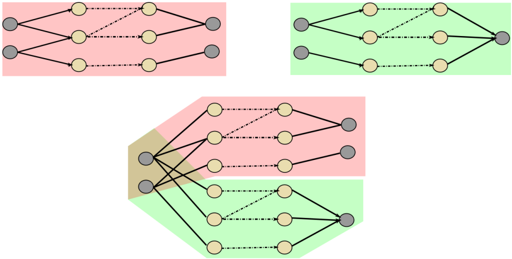
Parallelization

Let Φ_1 be a NN with L_1 layers, M_1 nodes, input dimension d, output dimension d' and Φ_2 with L_2 layers, M_2 nodes input dimension d, output dimension d''. Then there exists a NN Φ_3 with $\max\{L_1, L_2\}$ layers and $M_1 + M_2 + \mathcal{O}((d' + d'')|L_1 - L_2|)$ nodes such that $R_{ReLU}(\Phi_3)(x) = (R_{ReLU}(\Phi_1)(x), R_{ReLU}(\Phi_2)(x))$.

ReLU Calculus: Parallelization

Parallelization

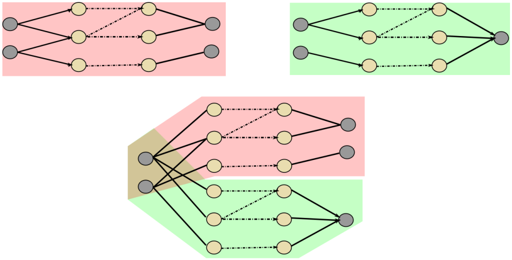
Let Φ_1 be a NN with L_1 layers, M_1 nodes, input dimension d, output dimension d' and Φ_2 with L_2 layers, M_2 nodes input dimension d, output dimension d''. Then there exists a NN Φ_3 with $\max\{L_1, L_2\}$ layers and $M_1 + M_2 + \mathcal{O}((d' + d'')|L_1 - L_2|)$ nodes such that $R_{ReLU}(\Phi_3)(x) = (R_{ReLU}(\Phi_1)(x), R_{ReLU}(\Phi_2)(x))$.



ReLU Calculus: Parallelization

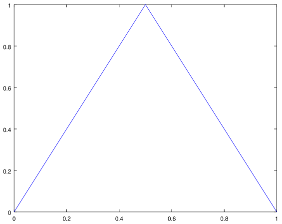
Parallelization

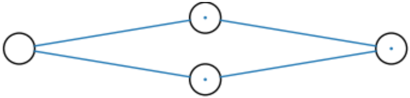
Let Φ_1 be a NN with L_1 layers, M_1 nodes, input dimension d, output dimension d' and Φ_2 with L_2 layers, M_2 nodes input dimension d, output dimension d''. Then there exists a NN Φ_3 with $\max\{L_1, L_2\}$ layers and $M_1 + M_2 + \mathcal{O}((d' + d'')|L_1 - L_2|)$ nodes such that
$R_{ReLU}(\Phi_3)(x) = (R_{ReLU}(\Phi_1)(x), R_{ReLU}(\Phi_2)(x))$.



$\rightsquigarrow$ the class of ReLU networks is closed under linear combinations (with controlled complexity)!

ReLU Calculus: Hat Function

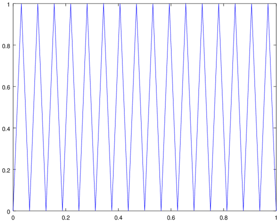




$$g(x) = \begin{cases} 2x & 0 < x < 1/2 \\ 2(1-x) & 1/2 \leq x < 1 \\ 0 & \text{else} \end{cases}$$

$$g(x) = ReLU\left(2 \cdot ReLU(x) - 4 \cdot ReLU\left(x - \frac{1}{2}\right)\right)$$

ReLU Calculus: Sawtooth Function [Telgarsky 2016]



$g_5 := g \circ g \circ g \circ g \circ g(x)$

"deep" ReLU net

$g_m = \underbrace{g \circ \cdots \circ g}_{m \text{ times}}$ is a sawtooth function with 2^m peaks.

ReLU Calculus: The Square Function

Lemma

For all $x \in [0, 1]$ it holds that

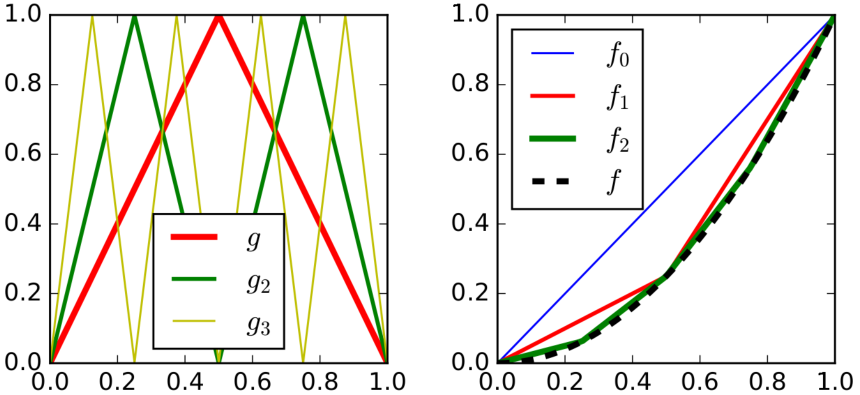
$$x^2 = x - \sum_{m=1}^{\infty} 2^{-2m} g_m(x).$$

ReLU Calculus: The Square Function

Lemma

For all $x \in [0,1]$ it holds that

$$x^2 = x - \sum_{m=1}^{\infty} 2^{-2m} g_m(x).$$



ReLU Calculus: The Square Function

Theorem [Yarotzky (2016)]

For all $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and nodes such that

$$\sup_{x \in [0,1]} |x^2 - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

ReLU Calculus: The Square Function

Theorem [Yarotzky (2016)]

For all $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and nodes such that

$$\sup_{x \in [0,1]} |x^2 - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

Proof.

Use the fact that

$$x^2 = x - \sum_{m=1}^{\infty} 2^{-2m} g_m(x).$$

together with the fact that g_m can be represented by a NN with $\mathcal{O}(m)$ layers and nodes. □

ReLU Calculus: Multiplication

Theorem [Yarotzky (2016)]

For all $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and nodes such that

$$\sup_{(x_1,x_2)\in[0,1]^2} |x_1 x_2 - R_{ReLU}(\Phi_\epsilon)(x_1, x_2)| \leq \epsilon.$$

ReLU Calculus: Multiplication

Theorem [Yarotzky (2016)]

For all $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and nodes such that

$$\sup_{(x_1,x_2)\in[0,1]^2} |x_1 x_2 - R_{ReLU}(\Phi_\epsilon)(x_1, x_2)| \leq \epsilon.$$

Proof.

Use the fact that $x_1 x_2 = \frac{(x_1+x_2)^2 - x_1^2 - x_2^2}{2}$ together with the previous result on the approximation of squares. □

ReLU Calculus: Polynomials

Theorem [Yarotzky (2016)]

Let $p(x) = \sum_{i=0}^{l} a_i x^i$. For all $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\mathrm{polylog}(1/\epsilon))$ layers and nodes such that

$$\sup_{x \in [0,1]} |p(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

ReLU Calculus: Polynomials

Theorem [Yarotzky (2016)]

Let $p(x) = \sum_{i=0}^{l} a_i x^i$. For all $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\mathrm{polylog}(1/\epsilon))$ layers and nodes such that

$$\sup_{x \in [0,1]} |p(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

Proof.

Use the fact that monomials can be approximated by iteratively approximating the square and multiplication with the identity, as well as the fact that ReLU networks are closed w.r.t. linear combinations. $\square$

ReLU Calculus: Smooth Functions I

Theorem

Suppose that f is *Gevrey*, i.e., there is $C, R, \sigma \in (0, \infty)$ with

$$\sup_{x \in [0,1]} |f^{(k)}(x)| \leq CR^k (k!)^\sigma \quad \text{for all } k \in \mathbb{N}.$$

Then for every $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\mathrm{polylog}(1/\epsilon))$ layers and nodes such that

$$\sup_{x \in [0,1]} |f(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

ReLU Calculus: Smooth Functions I

Theorem

Suppose that f is *Gevrey*, i.e., there is $C, R, \sigma \in (0, \infty)$ with

$$\sup_{x\in[0,1]} |f^{(k)}(x)| \leq CR^k(k!)^\sigma \quad \text{for all } k \in \mathbb{N}.$$

Then for every $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\mathrm{polylog}(1/\epsilon))$ layers and nodes such that

$$\sup_{x\in[0,1]} |f(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

Proof.

Use the fact that f can be well approximated by (local) polynomials and the NN approximation of polynomials. $\square$

ReLU Calculus: Smooth Functions II

Theorem

Suppose that f is satisfies $\|f\|_{C^n[0,1]} \leq 1$. Then for every $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and $\mathrm{size}_0(\Phi_\epsilon) = \mathcal{O}(\epsilon^{-1/n})$ such that

$$\sup_{x \in [0,1]} |f(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

ReLU Calculus: Smooth Functions II

Theorem

Suppose that f is satisfies $\|f\|_{C^n[0,1]} \leq 1$. Then for every $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and $\mathrm{size}_0(\Phi_\epsilon) = \mathcal{O}(\epsilon^{-1/n})$ such that

$$\sup_{x \in [0,1]} |f(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

Proof.

Use the fact that f can be well approximated by (local) polynomials and the NN approximation of polynomials. □

ReLU Calculus: Smooth Functions II

Theorem

Suppose that f is satisfies $\|f\|_{C^n[0,1]} \leq 1$. Then for every $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and $\mathrm{size}_0(\Phi_\epsilon) = \mathcal{O}(\epsilon^{-1/n})$ such that

$$\sup_{x \in [0,1]} |f(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

Proof.

Use the fact that f can be well approximated by (local) polynomials and the NN approximation of polynomials. $\square$

This is optimal!

ReLU Calculus: Smooth Functions II

Theorem

Suppose that f is satisfies $\|f\|_{C^n[0,1]} \leq 1$. Then for every $\epsilon \in (0, \infty)$ there is a NN Φ_ϵ with $\mathcal{O}(\ln(1/\epsilon))$ layers and $\mathrm{size}_0(\Phi_\epsilon) = \mathcal{O}(\epsilon^{-1/n})$ such that

$$\sup_{x \in [0,1]} |f(x) - R_{ReLU}(\Phi_\epsilon)(x)| \leq \epsilon.$$

Proof.

Use the fact that f can be well approximated by (local) polynomials and the NN approximation of polynomials. $\square$

This is optimal! $\rightsquigarrow$ VC Dimension argument...

ReLU Calculus: Wavelet Approximation

Consider Wavelet ONB $(\psi_{j,k})_{j,k}$ of $L^2[0,1]$, where $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$ and ψ (analytic) wavelet.

ReLU Calculus: Wavelet Approximation

Consider Wavelet ONB $(\psi_{j,k})_{j,k}$ of $L^2[0,1]$, where $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$ and ψ (analytic) wavelet.

Theorem

Suppose that f has N-term approximation rate s, meaning that

$$\inf_{\#I \leq N;\ g = \sum_{(j,k)\in I} c_{j,k}\psi_{j,k}} \|f - g\|_{L^2} = \mathcal{O}(N^{-s}).$$

ReLU Calculus: Wavelet Approximation

Consider Wavelet ONB $(\psi_{j,k})_{j,k}$ of $L^2[0,1]$, where $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$ and ψ (analytic) wavelet.

Theorem

Suppose that f has N-term approximation rate s, meaning that

$$\inf_{\#I \leq N;\ g=\sum_{(j,k)\in I} c_{j,k}\psi_{j,k}} \|f - g\|_{L^2} = \mathcal{O}(N^{-s}).$$

Then for all M there is a NN Φ_M with $\mathrm{size}_0(\Phi_M) \leq M$ and

$$\|f - R_{ReLU}(\Phi_M)\|_{L^2} = \mathcal{O}(M^{-s}\mathrm{polylog}(M)).$$

ReLU Calculus: Wavelet Approximation

Consider Wavelet ONB $(\psi_{j,k})_{j,k}$ of $L^2[0,1]$, where $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$ and ψ (analytic) wavelet.

Theorem

Suppose that f has N-term approximation rate s, meaning that

$$\inf_{\#I \leq N;\ g=\sum_{(j,k)\in I} c_{j,k}\psi_{j,k}} \|f - g\|_{L^2} = \mathcal{O}(N^{-s}).$$

Then for all M there is a NN Φ_M with $\mathrm{size}_0(\Phi_M) \leq M$ and

$$\|f - R_{ReLU}(\Phi_M)\|_{L^2} = \mathcal{O}(M^{-s}\mathrm{polylog}(M)).$$

Proof.

Observe that analytic ψ can be well approximated and dilation is affine. $\square$

ReLU Calculus: Wavelet Approximation

Consider Wavelet ONB $(\psi_{j,k})_{j,k}$ of $L^2[0,1]$, where $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$ and ψ (analytic) wavelet.

Theorem

Suppose that f has N-term approximation rate s, meaning that

$$\inf_{\#I \leq N;\ g = \sum_{(j,k)\in I} c_{j,k}\psi_{j,k}} \|f - g\|_{L^2} = \mathcal{O}(N^{-s}).$$

Then for all M there is a NN Φ_M with $\mathrm{size}_0(\Phi_M) \leq M$ and

$$\|f - R_{ReLU}(\Phi_M)\|_{L^2} = \mathcal{O}(M^{-s}\mathrm{polylog}(M)).$$

Proof.

Observe that analytic ψ can be well approximated and dilation is affine. $\square$

This is optimal!

ReLU Calculus: Wavelet Approximation

Consider Wavelet ONB $(\psi_{j,k})_{j,k}$ of $L^2[0,1]$, where $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$ and ψ (analytic) wavelet.

Theorem

Suppose that f has N-term approximation rate s, meaning that

$$\inf_{\#I \leq N;\ g = \sum_{(j,k)\in I} c_{j,k}\psi_{j,k}} \|f - g\|_{L^2} = \mathcal{O}(N^{-s}).$$

Then for all M there is a NN Φ_M with $\mathrm{size}_0(\Phi_M) \leq M$ and

$$\|f - R_{ReLU}(\Phi_M)\|_{L^2} = \mathcal{O}(M^{-s}\mathrm{polylog}(M)).$$

Proof.

Observe that analytic ψ can be well approximated and dilation is affine. $\square$

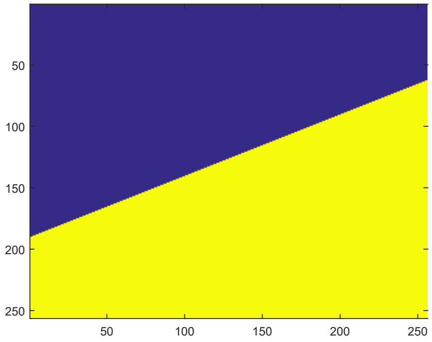
This is optimal! $\rightsquigarrow$ Phase Transition Argument...

A Numerical Experiment

We computed a NN approximation of a function with a line singularity via backpropagation.

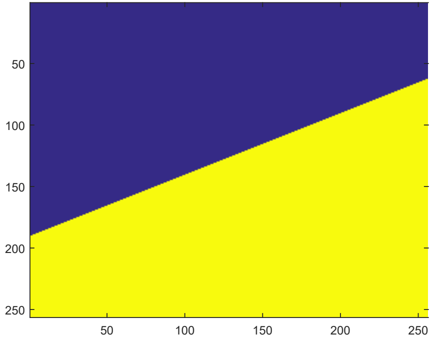
A Numerical Experiment

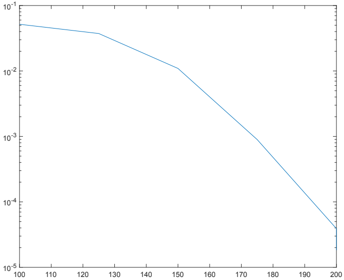
We computed a NN approximation of a function with a line singularity via backpropagation.



A Numerical Experiment

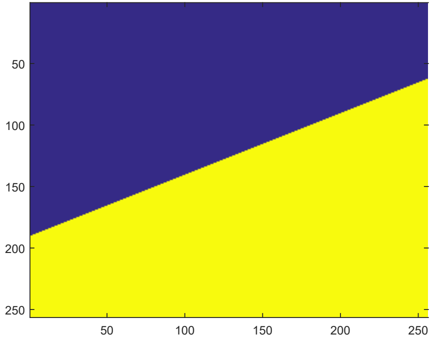
We computed a NN approximation of a function with a line singularity via backpropagation.

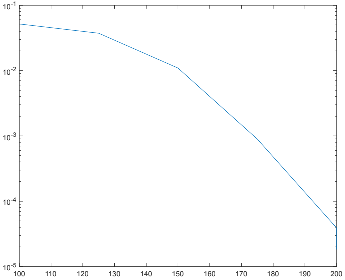




A Numerical Experiment

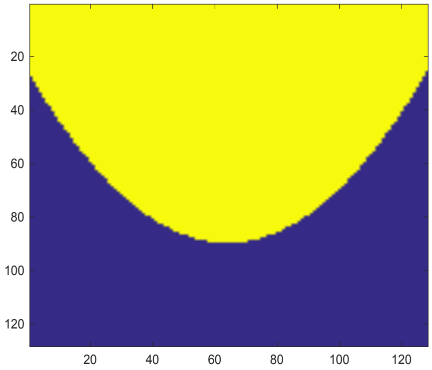
We computed a NN approximation of a function with a line singularity via backpropagation.



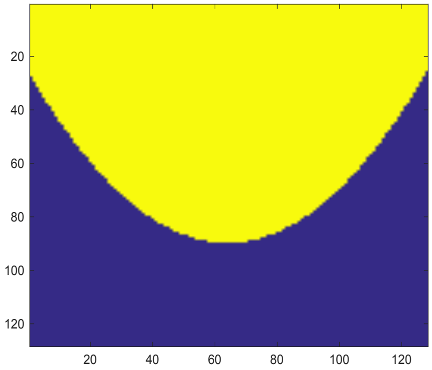


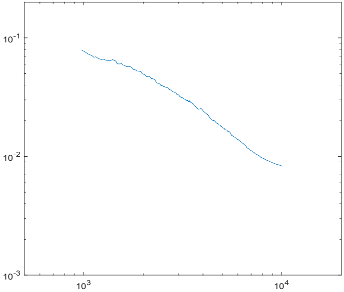
Convergence as expected.

A Numerical Experiment II

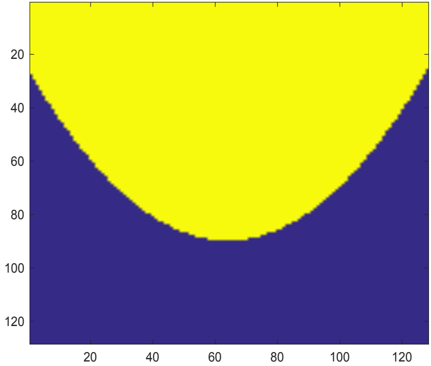


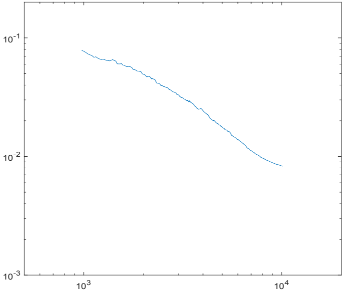
A Numerical Experiment II





A Numerical Experiment II

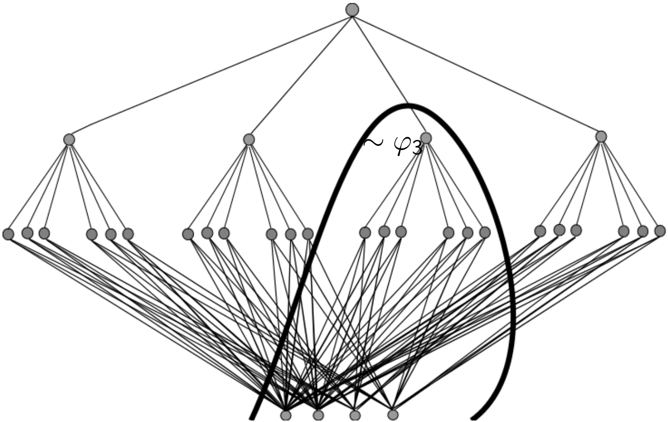




Convergence again as expected.

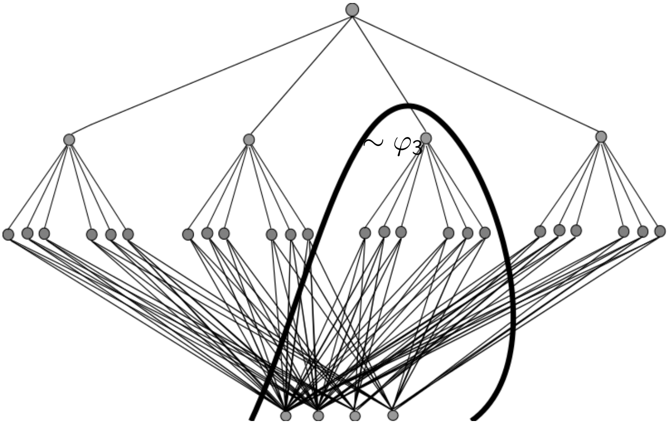
Recall

Suppose that $\sum_{i=1}^{4} c_i \varphi_i$ is a sparse approximation of $F \in \mathcal{D}$.



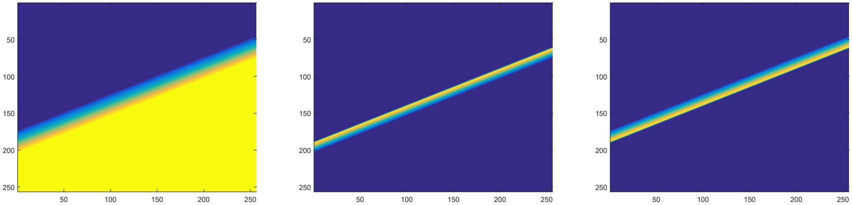
Recall

Suppose that $\sum_{i=1}^{4} c_i \varphi_i$ is a sparse approximation of $F \in \mathcal{D}$.



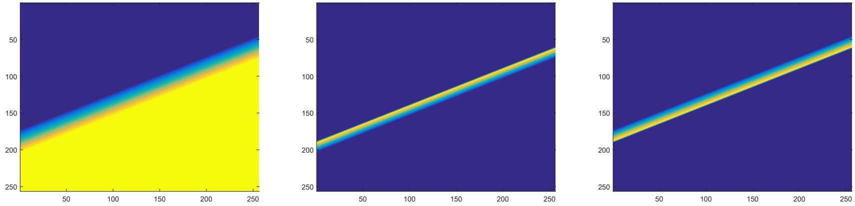
How do these subnetworks look? Do they look like elements in the optimal dictionary (ridgelets)?

A Surprise



Plot of the three most prominent subnetworks in the approximation of line singularity

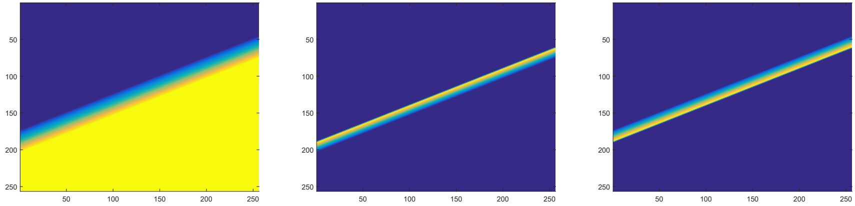
A Surprise



Plot of the three most prominent subnetworks in the approximation of line singularity

Backpropagation automatically finds ridgelet-like subnetworks! We observed the same behaviour for functions with curved singularities and shearlet-like subnetworks.

A Surprise



Plot of the three most prominent subnetworks in the approximation of line singularity

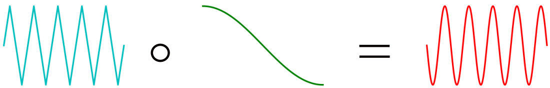
Backpropagation automatically finds ridgelet-like subnetworks! We observed the same behaviour for functions with curved singularities and shearlet-like subnetworks.

Question

Can this be explained??

ReLU Calculus: Modulation [G-Pekrekreshtenko-Elbrächter-Bölcskei 2018]

- $\cos(\Lambda x)$ can be well approximated by neural networks of size $\sim \log(\Lambda)$:



- Let $\Lambda \in \mathbb{R}^d$. Suppoose that $f(x)$ can be well approximated by neural networks. Then the modulation

$$M_\Lambda f(x) := \exp(2\pi i x \cdot \Lambda) \cdot f(x)$$

can be well approximated by neural networks (combine previous observation with multiplication).

- Gabor systems and all decomposition spaces can be well approximated by neural networks.

ReLU Calculus: Tensor Products [Elbrächter-G-Jentzen-Schwab 2018]

Suppose that $f_i : \mathbb{R} \to \mathbb{R}$ can be well approximated by neural networks. Then the tensor product

$$(x_1, \ldots, x_d) \mapsto \prod_{i=1}^{d} f_i(x_i)$$

can be well approximated by neural networks without curse of dimensionality (combine bivariate product with the observation that $x_1 \cdot x_2 \cdot x_3 \cdot x_4 = (x_1 \cdot x_2) \cdot (x_3 \cdot x_4)$).

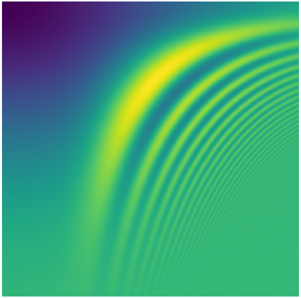
ReLU Calculus: Maxima

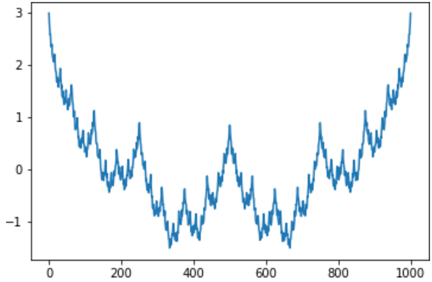
Suppose that $f_i : \mathbb{R} \to \mathbb{R}$ can be well approximated by neural networks. Then the function

$$(x_1, \ldots, x_d) \mapsto \max_{i=1}^{d} f_i(x_i)$$

can be well approximated by neural networks without curse of dimensionality (combine $\max\{x, y\} = ReLU(x - y) + y$ with the observation that $\max\{x_1, x_2, x_3, x_4\} = \max\{\max\{x_1, x_2\}, \max\{x_3, x_4\}\}$).

Non-standard Function Classes





Left: Function of the form $\cos(Ag(x)) \cdot h(x)$ with A large and g, h smooth can be approximated to within accuracy ϵ with a NN Φ_ϵ of size $\lesssim \log(\epsilon^{-1}) + \log(A)$.

Right: Weierstrass function $W(x) = \sum_{k=0}^{\infty} 2^{-k/2} \cos(2^k \pi x)$ can be approximated to within accuracy ϵ with a NN Φ_ϵ of size $\lesssim \log(\epsilon^{-1})$.