Regularity theory of deep ReLU networks in the context of partial differential equations

Julius Berner

Faculty of Mathematics,
University of Vienna, Austria

School on 'Mathematical and Computational Aspects of Machine Learning' at Scuola Normale Superiore

October 9, 2019

Motivating Question

$$f_d(x_1, \ldots, x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with sufficiently regular } f_{ijk} \colon \mathbb{R} \to \mathbb{R}$$

Is there a ReLU network approximating f_d?

Motivating Question

$$f_d(x_1, \dots, x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with sufficiently regular } f_{ijk} \colon \mathbb{R} \to \mathbb{R}$$

Is there a ReLU network approximating f_d and its derivative approximating Df_d?

Motivating Question

$$f_d(x_1, \dots, x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with sufficiently regular } f_{ijk} \colon \mathbb{R} \to \mathbb{R}$$

Is there a ReLU network approximating f_d and its derivative approximating Df_d efficiently/without curse of dimensionality?

Motivating Question

$$f_d(x_1, \ldots, x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with sufficiently regular } f_{ijk} \colon \mathbb{R} \to \mathbb{R}$$

Is there a ReLU network approximating f_d and its derivative approximating Df_d globally efficiently/without curse of dimensionality?

Motivating Question

$$f_d(x_1, \dots, x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with sufficiently regular } f_{ijk} \colon \mathbb{R} \to \mathbb{R}$$

Is there a ReLU network approximating f_d and its derivative approximating Df_d globally efficiently/without curse of dimensionality?

Applications - Partial Differential Equations

- emulation of classical approximation methods based on sparse expansions
- efficient approximation of PDE solutions with general low-rank structures

Example - Kolmogorov equation

Definition (Kolmogorov equation)

$$\begin{cases} \partial_t u(t,x) = \frac{1}{2}\mathrm{Trace}\big(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\big) + \mu(x)\cdot\nabla_x u(t,x) \\ u(0,x) = \varphi(x) \end{cases}$$

Example - Kolmogorov equation

Definition (Kolmogorov equation)

$$\begin{cases} \partial_t u(t,x) = \frac{1}{2}\mathrm{Trace}\big(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\big) + \mu(x)\cdot\nabla_x u(t,x) \\ u(0,x) = \varphi(x) \end{cases}$$

low-rank structure

[Elbrächter et al. '18]

efficient approximation of PDE solution u

Example - Kolmogorov equation

Definition (Kolmogorov equation)

$$\begin{cases} \partial_t u(t,x) = \frac{1}{2}\mathrm{Trace}\big(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\big) + \mu(x) \cdot \nabla_x u(t,x) \\ u(0,x) = \varphi(x) \end{cases}$$

efficient approximation of φ, σ, μ

Feynman-Kac formula,
simulation of MC sampling

[Grohs et al. '18; Jentzen et al. '18]

low-rank structure

[Elbrächter et al. '18]

efficient approximation of PDE solution u

Example - Kolmogorov equation

Definition (Kolmogorov equation)

$$\begin{cases} \partial_t u(t,x) = \frac{1}{2}\mathrm{Trace}\big(\sigma(x)\sigma^T(x)\mathrm{Hess}_x u(t,x)\big) + \mu(x)\cdot\nabla_x u(t,x) \\ u(0,x) = \varphi(x) \end{cases}$$

efficient approximation of φ, σ, μ

low-rank structure

Feynman-Kac formula,
simulation of MC sampling
[Grohs et al. '18; Jentzen et al. '18]

[Elbrächter et al. '18]

efficient approximation of PDE solution u

Feynman-Kac formula,
statistical learning theory

[B., Grohs, Jentzen '18]

efficient learning of PDE solution u via ERM

Feed-Forward Neural Network

- parametrization of neural network with architecture $(N_0, \dots, N_L)$:

$$\Phi = ((A_\ell, b_\ell))_{\ell=1}^L$$

where $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$

Feed-Forward Neural Network

- parametrization of neural network with architecture $(N_0, \dots, N_L)$:

$$\Phi = ((A_\ell, b_\ell))_{\ell=1}^{L}$$

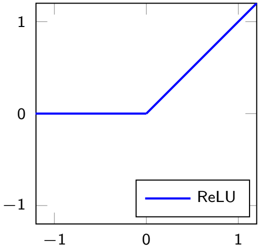
  where $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$
- $\#neurons(\Phi) := \sum_{\ell=0}^{L} N_\ell$

Feed-Forward Neural Network

- parametrization of neural network with architecture $(N_0, \ldots, N_L)$:

$$\Phi = ((A_\ell, b_\ell))_{\ell=1}^{L}$$

  where $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$
- $\#neurons(\Phi) := \sum_{\ell=0}^{L} N_\ell$
- activation function ϱ: piecewise linear with at least one breakpoint
  - $\text{ReLU}(x) = max\{x, 0\}$

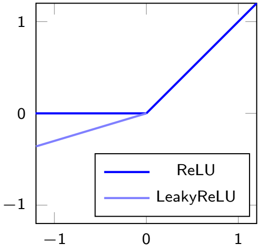


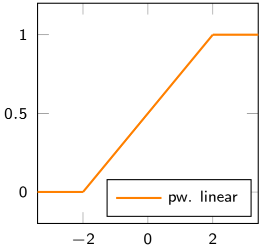
Feed-Forward Neural Network

- parametrization of neural network with architecture $(N_0, \dots, N_L)$:

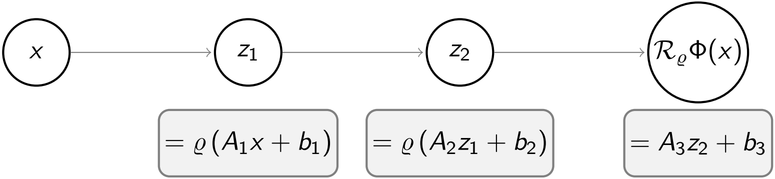
$$\Phi = ((A_\ell, b_\ell))_{\ell=1}^L$$

  where $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$
- $\#neurons(\Phi) := \sum_{\ell=0}^{L} N_\ell$
- activation function ϱ: piecewise linear with at least one breakpoint
  - $\mathrm{ReLU}(x) = max\{x, 0\}$
  - $\mathrm{LeakyReLU}(x) = \max\{\alpha x, x\}, \quad \alpha \in (0,1)$

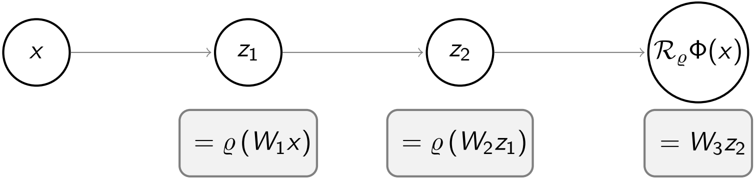




Artifical Feed-Forward Neural Network



Artifical Feed-Forward Neural Network



Definition (realization of a parametrization)

Realization $\mathcal{R}\Phi$ of parametrization $\Phi = ((A_\ell, b_\ell))_{\ell=1}^{L}$:

$$\mathcal{R}\Phi := W_L \circ \varrho \circ W_{L-1} \circ \ldots \circ \varrho \circ W_1$$

where $W_k(x) := A_k x + b_k$ and ϱ is applied component-wise.

Local $\mathcal{L}^\infty$ Approximation [Yarotsky '16]

- $K \subset \mathbb{R}^d$ compact

Local $\mathcal{L}^\infty$ Approximation [Yarotsky '16]

- $K \subset \mathbb{R}^d$ compact

✓ $\exists\ \Phi$ with $\|f - \mathcal{R}\Phi\|_{\mathcal{L}^\infty(K)} \leq \varepsilon$ (Universal Approximation Theorem)

Local $\mathcal{L}^\infty$ Approximation [Yarotsky '16]

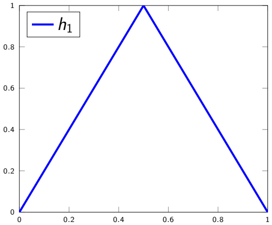
- $K \subset \mathbb{R}^d$ compact

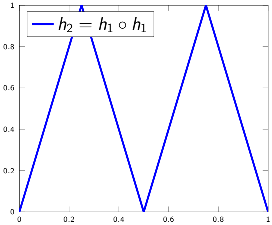
✓ $\exists\ \Phi$ with $\|f - \mathcal{R}\Phi\|_{\mathcal{L}^\infty(K)} \leq \varepsilon$ (Universal Approximation Theorem)

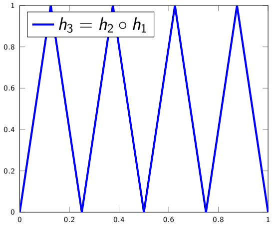
? $\#neurons \lesssim \dots$

Local $\mathcal{L}^\infty$ Approximation [Yarotsky '16]

- sawtooth function $\#neurons \lesssim \log(\#teeth)$



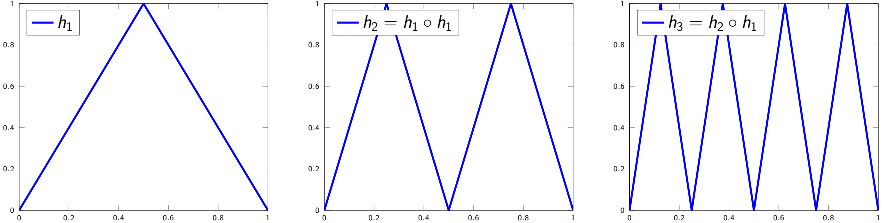




Local $\mathcal{L}^\infty$ Approximation [Yarotsky '16]

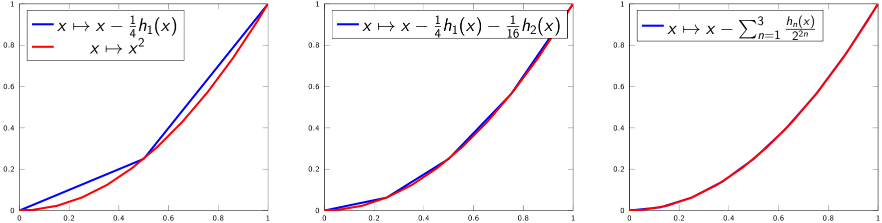
- sawtooth function

$\#neurons \lesssim \log(\#teeth)$



$\Rightarrow$ squaring function†

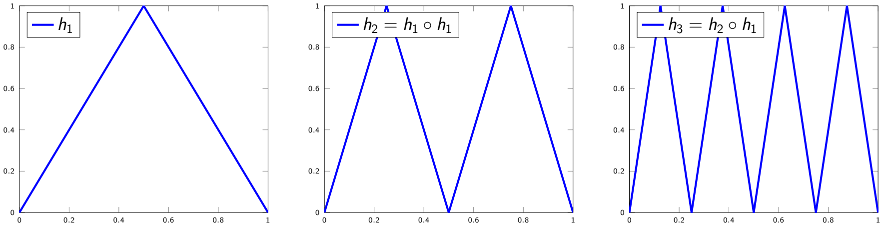
$^\dagger \#neurons \lesssim \log(\varepsilon^{-1})$



Local $\mathcal{L}^\infty$ Approximation [Yarotsky '16]

- sawtooth function

$$\#\textit{neurons} \lesssim \log(\#\textit{teeth})$$



$\Rightarrow$ squaring function†

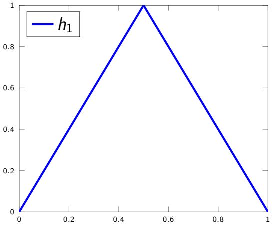
$\Rightarrow$ multiplication†

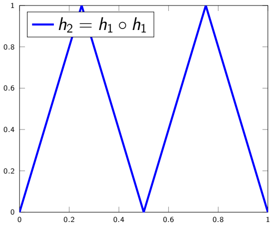
$$^\dagger\#\textit{neurons} \lesssim \log(\varepsilon^{-1})$$

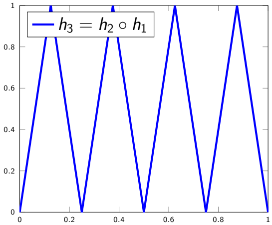
$$xy = \left|\frac{x+y}{2}\right|^2 - \left|\frac{x-y}{2}\right|^2$$

- sawtooth function

$\#neurons \lesssim \log(\#teeth)$







$\Rightarrow$ squaring function†

$\Rightarrow$ multiplication†

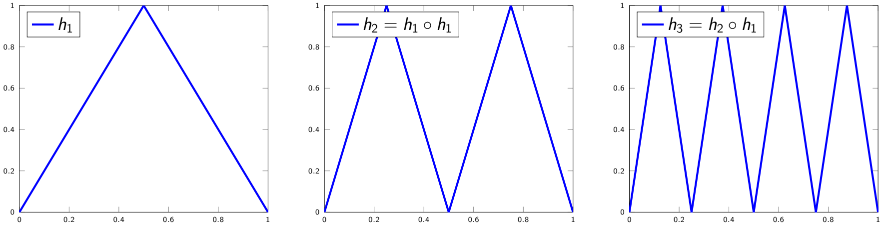
$\Rightarrow$ polynomials†

† $\#neurons \lesssim \log(\varepsilon^{-1})$

Local $\mathcal{L}^\infty$ Approximation [Yarotsky '16]

- sawtooth function

$\#neurons \lesssim \log(\#teeth)$



$\Rightarrow$ squaring function†

$\Rightarrow$ multiplication†

$\Rightarrow$ polynomials†

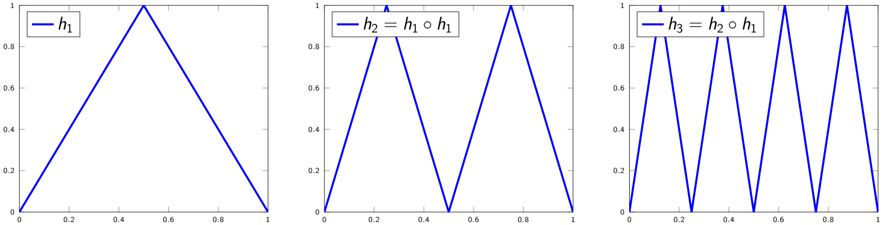
†$\#neurons \lesssim \log(\varepsilon^{-1})$

$\Rightarrow$ $\mathcal{C}^n$-functions

$\#neurons \lesssim \varepsilon^{-\frac{dimension}{n}} \log^2(\varepsilon^{-1})$

- sawtooth function

$\#neurons \lesssim \log(\#teeth)$



$\Rightarrow$ squaring function†

$^\dagger\#neurons \lesssim \log(\varepsilon^{-1})$

$\Rightarrow$ multiplication†

$\Rightarrow$ polynomials†

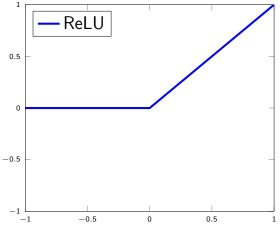
$\Rightarrow$ $\mathcal{C}^n$-functions

$\#neurons \lesssim \varepsilon^{-\frac{dimension}{n}} \log^2(\varepsilon^{-1})$

Goal: simultaneous approximation of f and Df ($\sout{\|\cdot\|_{\mathcal{L}^\infty}} \to \|\cdot\|_{\mathcal{W}^{1,\infty}}$)

Neural Network Derivative $\mathcal{D}$

- Problem: chain rule fails!

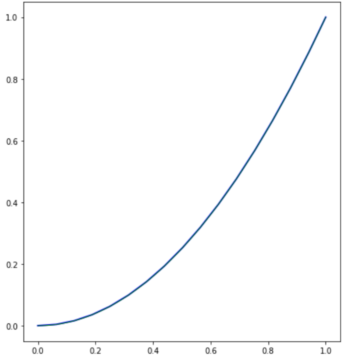


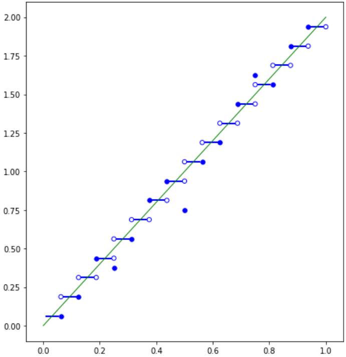
Neural Network Derivative $\mathcal{D}$

- Problem: chain rule fails! $\Rightarrow$ define $\mathcal{D}\,\mathrm{ReLU}(x) := \begin{cases} 1, & x > 0 \\ c, & x = 0 \\ 0, & x < 0 \end{cases}$
- deep learning libraries (TensorFlow, PyTorch): $c = 0 \quad \Rightarrow$ sparsity

Neural Network Derivative $\mathcal{D}$

- Problem: chain rule fails! $\Rightarrow$ define $\mathcal{D}\,\mathrm{ReLU}(x) := \begin{cases} 1, & x > 0 \\ c, & x = 0 \\ 0, & x < 0 \end{cases}$
- deep learning libraries (TensorFlow, PyTorch): $c = 0 \quad \Rightarrow$ sparsity
- $\mathcal{D}\Phi$ not necessarily in the subdifferential (regardless of choice of c)





Neural Network Derivative $\mathcal{D}$

- Problem: chain rule fails! $\Rightarrow$ define $\mathcal{D}\,\mathrm{ReLU}(x) := \begin{cases} 1, & x > 0 \\ c, & x = 0 \\ 0, & x < 0 \end{cases}$
- deep learning libraries (TensorFlow, PyTorch): $c = 0$ $\Rightarrow$ sparsity
- $\mathcal{D}\Phi$ not necessarily in the subdifferential (regardless of choice of c)

Lemma (properties of $\mathcal{D}$ [B., Elbrächter, Grohs, Jentzen '19])

- well-defined: $\mathcal{D}\Phi = D[\mathcal{R}\Phi]$ a.e.
- chain-rule: $\mathcal{D}(\Psi \circ \Phi) = \mathcal{D}\Psi(\mathcal{R}\Phi) \cdot \mathcal{D}\Phi$

Local $\mathcal{W}^{1,\infty}$ Approximation

$\Rightarrow$ Use Yarotsky's construction and chain rule

Local $\mathcal{W}^{1,\infty}$ Approximation

$\Rightarrow$ Use Yarotsky's construction and chain rule

Theorem (upper bounds [Gühring, Kutyniok, Petersen '19])

For every

$$f \in \mathcal{W}^{n,\infty}(K) \quad \text{with} \quad \|f\|_{\mathcal{W}^{n,\infty}} \leq C$$

there exists Φ with $\|f - \mathcal{R}\Phi\|_{\mathcal{W}^{1,\infty}(K)} \leq \varepsilon$ and

$$\#\textit{neurons} \lesssim \varepsilon^{-\frac{\textit{dimension}}{n-1}} \log^2(\varepsilon^{-1})$$

Local $\mathcal{W}^{1,\infty}$ Approximation

$\Rightarrow$ Use Yarotsky's construction and chain rule

Theorem (upper bounds [Gühring, Kutyniok, Petersen '19])

For every

$$f \in \mathcal{W}^{n,\infty}(K) \quad \text{with} \quad \|f\|_{\mathcal{W}^{n,\infty}} \leq C$$

there exists Φ with $\|f - \mathcal{R}\Phi\|_{\mathcal{W}^{s,\infty}(K)} \leq \varepsilon$ and

$$\#neurons \lesssim \varepsilon^{-\frac{dimension}{n-s}} \log^2(\varepsilon^{-1}) \qquad s \in [0,1]$$

Local $\mathcal{W}^{1,\infty}$ Approximation

$\Rightarrow$ Use Yarotsky's construction and chain rule

Theorem (upper bounds [Gühring, Kutyniok, Petersen '19])

For every

$$f \in \mathcal{W}^{n,\infty}(K) \quad \text{with} \quad \|f\|_{\mathcal{W}^{n,\infty}} \leq C$$

there exists Φ with $\|f - \mathcal{R}\Phi\|_{\mathcal{W}^{s,\infty}(K)} \leq \varepsilon$ and

$$\#neurons \lesssim \varepsilon^{-\frac{dimension}{n-s}} \log^2(\varepsilon^{-1}) \qquad s \in [0, 1]$$

Goal: approximation without curse of dimensionality

(~~general Sobolev-regular function~~ $\rightarrow$ low-rank structure)

Local Curseless $\mathcal{W}^{1,\infty}$ Approximation

Definition (approximation without curse of dimensionality)

$f_d \colon \mathbb{R}^d \to \mathbb{R}$, $d \in \mathbb{N}$, can be approximated by networks $\{\Phi_{\varepsilon,d}\}$ without curse of dimensionality if

$$\|f_d - \mathcal{R}\Phi_{\varepsilon,d}\|_{\mathcal{W}^{1,\infty}(K)} \leq \varepsilon \quad \text{and} \quad \#\textit{neurons} \leq \mathrm{poly}(\varepsilon^{-1}, d)$$

Informal Theorem (sufficient conditions)

Under mild conditions functions f_d given by

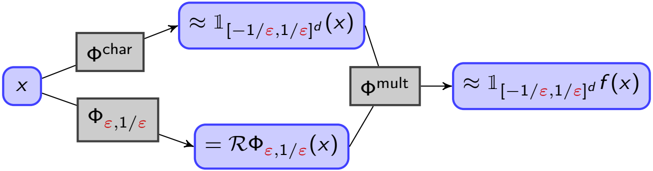
- linear combinations, multivariate products, multivariate maxima/minima

of

- Sobolev-regular functions depending only on k variables

can be approximated without curse.

- given local approximations $\|f - \mathcal{R}\Phi_{\varepsilon,B}\|_{\mathcal{W}^{1,\infty}((-B,B)^d)} \leq \varepsilon$ for f with at most polynomially (with degree κ) growing derivative



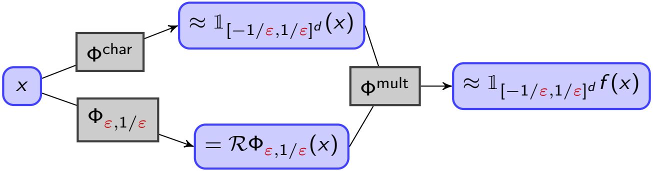
Global Approximation

- given local approximations $\|f - \mathcal{R}\Phi_{\varepsilon,B}\|_{\mathcal{W}^{1,\infty}((-B,B)^d)} \leq \varepsilon$ for f with at most polynomially (with degree κ) growing derivative

Theorem (global estimates [B., Elbrächter, Grohs, Jentzen '19])

There exists Ψ with

- $|f(x) - \mathcal{R}\Psi(x)| \leq \varepsilon(1 + \|x\|^{\kappa+2}) \quad \forall x \in \mathbb{R}^d$
- $\|Df(x) - \mathcal{D}\Psi(x)\| \leq \varepsilon(1 + \|x\|^{\kappa+2}) \quad \text{a.e. } x \in \mathbb{R}^d$



Global Approximation

- given local approximations $\|f - \mathcal{R}\Phi_{\varepsilon,B}\|_{\mathcal{W}^{1,\infty}((-B,B)^d)} \leq \varepsilon$ for f with at most polynomially (with degree κ) growing derivative

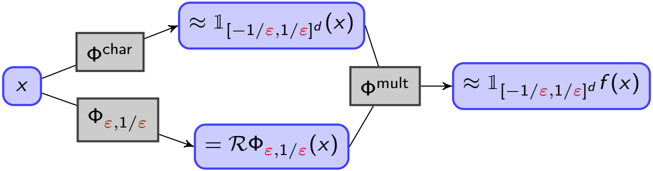
Theorem (global estimates [B., Elbrächter, Grohs, Jentzen '19])

There exists Ψ with

- $\#neurons(\Psi) \lesssim \#neurons(\Phi_{\varepsilon,1/\varepsilon}) + \log(d + \varepsilon^{-1})$

and

- $|f(x) - \mathcal{R}\Psi(x)| \leq \varepsilon(1 + \|x\|^{\kappa+2}) \quad \forall x \in \mathbb{R}^d$
- $\|Df(x) - \mathcal{D}\Psi(x)\| \leq \varepsilon(1 + \|x\|^{\kappa+2}) \quad$ a.e. $x \in \mathbb{R}^d$



Motivating Question

$$f_d(x_1, \dots, x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with } \|f_{ijk}\|_{\mathcal{W}^{n,\infty}} \leq C$$

Motivating Question

$$f_d(x_1, \dots, x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with } \|f_{ijk}\|_{\mathcal{W}^{n,\infty}} \leq C$$

$\Rightarrow$ there exists a ReLU network Ψ with

- $\#neurons \lesssim \mathrm{poly}(\varepsilon^{-1}, d)$
- $|f_d(x) - \mathcal{R}\Psi(x)| \leq \varepsilon(1 + \|x\|^{\kappa+2}) \quad \forall x \in \mathbb{R}^d$
- $\|Df_d(x) - \mathcal{D}\Psi(x)\| \leq \varepsilon(1 + \|x\|^{\kappa+2}) \quad$ a.e. $x \in \mathbb{R}^d$

Motivating Question

$$f_d(x_1,\dots,x_d) = \max_{i=1}^{d} \sum_{j=1}^{d} \prod_{k=1}^{d} f_{ijk}(x_k) \quad \text{with } \|f_{ijk}\|_{\mathcal{W}^{n,\infty}} \le C$$

$\Rightarrow$ there exists a ReLU network Ψ with

- $\#neurons \lesssim \mathrm{poly}(\varepsilon^{-1}, d)$
- $|f_d(x) - \mathcal{R}\Psi(x)| \le \varepsilon(1 + \|x\|^{\kappa+2}) \quad \forall x \in \mathbb{R}^d$
- $\|Df_d(x) - \mathcal{D}\Psi(x)\| \le \varepsilon(1 + \|x\|^{\kappa+2}) \quad$ a.e. $x \in \mathbb{R}^d$

✓ constructive proof, quantitative rates

Thank you for your Attention!

Julius Berner, Philipp Grohs, and Arnulf Jentzen. "Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations". In: *arXiv:1809.03062* (2018).

Julius Berner, Dennis Elbrächter, Philipp Grohs, and Arnulf Jentzen. "Towards a regularity theory for ReLU networks–chain rule and global error estimates". In: *arXiv:1905.04992* (2019). Accepted for presentation at SampTA 2019.