Optimal approximation of piecewise smooth functions using deep ReLU neural networks

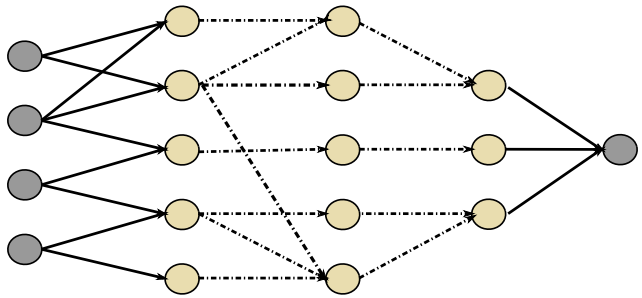
Felix Voigtlaender

This is joint work with Philipp Petersen (also TU Berlin)

Technische Universität Berlin
Institut für Mathematik

Seminar "Mathematics of Computation", Bonn
23. November 2017

Neural Networks and what they can do for you



A neural network is determined by the following:

- Dimension of input layer: $d = N_0 \in \mathbb{N}$,
- Number of layers: $L \in \mathbb{N}$,
- Activation function: $\varrho : \mathbb{R} \to \mathbb{R}$,
- Affine-linear maps $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}, x \mapsto A_\ell x + b_\ell$, $\ell = 1, \dots, L$.

The function $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ computed by the network is

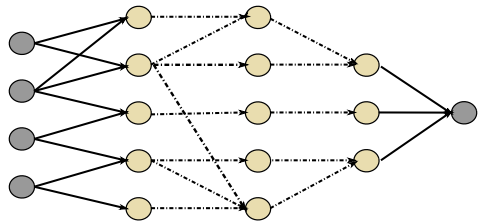
$$\Phi(x) = W_L(\varrho(W_{L-1}(\varrho(\dots \varrho(W_1(x)) \dots)))), \quad x \in \mathbb{R}^d.$$

Note: In the last layer, we do not apply ϱ!

In the following: mainly consider **ReLU** activation function $\varrho(x) = x_+$.

Neural Networks and what they can do for you

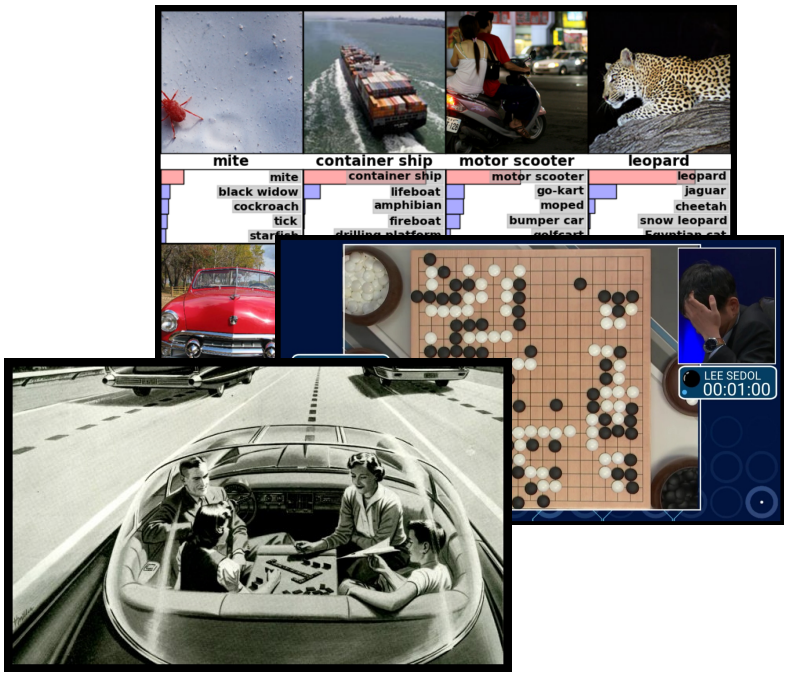
Deep Learning: Efficient training of large multi-layered neural networks:



- Very competitive in classification (and other) tasks
- DNN often yield better **generalization** than other methods
- Little theory explaining this success

Examples:

- Image classification (ImageNet)
- Game intelligence (AlphaGo)
- Self driving cars (Tesla Autopilot)



Why does Deep Learning work?

Usual setting of deep learning:

- Unknown ground truth: Function F (e.g., image classifier).
- Only known: Samples $(X_i, F(X_i))$.
- Using stochastic gradient descent, find network Φ that (more or less) minimizes

$$\sum_{i=1}^{N} |\Phi(X_i) - F(X_i)|^2 .$$

Hope: Φ generalizes, i.e., $\Phi(X) \approx F(X)$ also for $X \neq X_i$.

Why does this work so well? Three main aspects (Poggio *et al.*):

- **Power of the architecture**,
- Efficiency of stochastic gradient descent,
- Surprisingly good generalization.

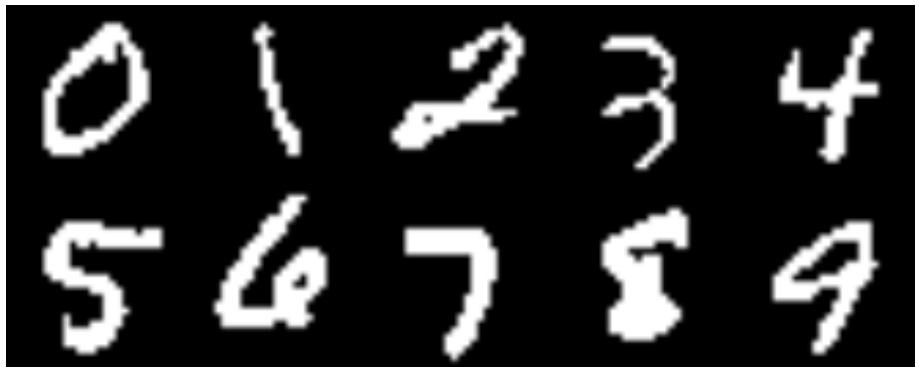
Power of architecture: Given f **from some class of functions**.

1. Is there a network Φ that implements (or approximates) f?
2. How complex does Φ need to be?

$\rightsquigarrow$ **Approximation properties of neural networks.**

Function classes of interest

A crucial task of deep learning is to learn **classifiers**, e.g., classification of hand-written digits:



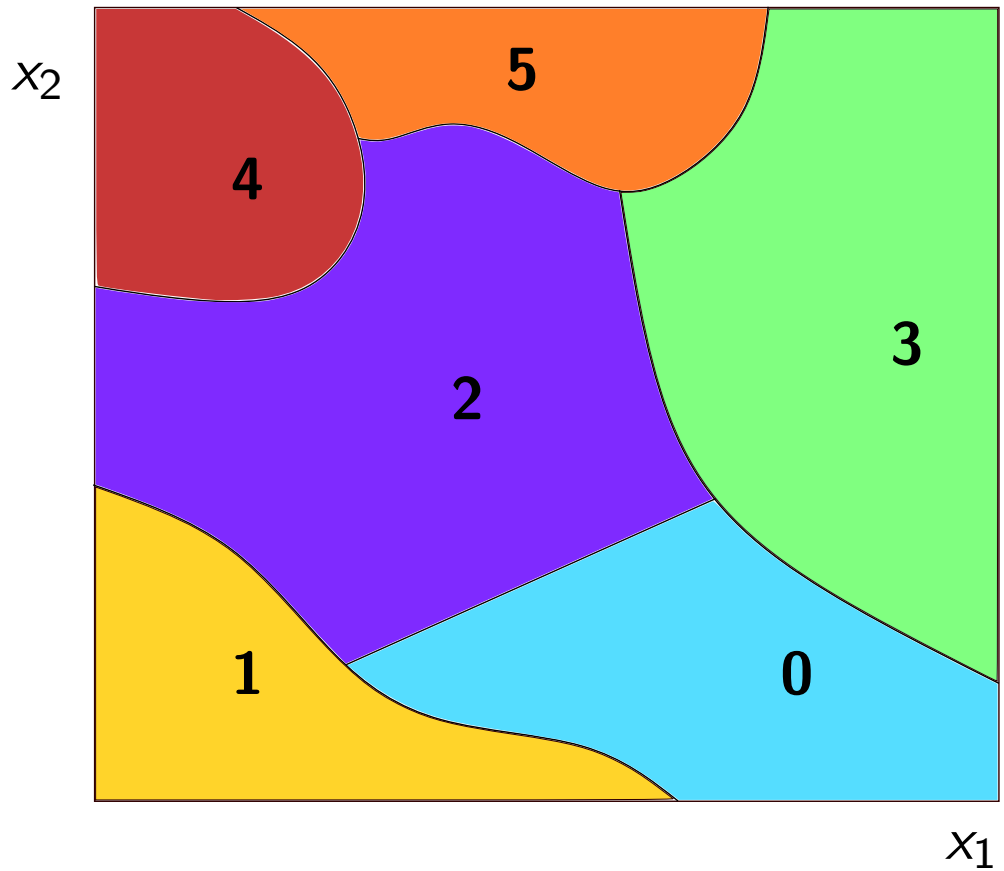
A model for these functions is:

$$f : \mathbb{R}^{N \times N} \to \{0, 1, \ldots, 9\}.$$

These are **piecewise constant functions**.

Function classes of interest

Example: A classifier function according to our model:
$f : [-1/2, 1/2]^2 \to \{0, 1, 2, 3, 4, 5\}$:



Note: In applications, the input dimensions are **much** larger.

A caveat

Our interpretation: A classifier is a function taking only **finitely many values**.

Another interpretation: A classifier function is (an estimate for) the **conditional probability** $\mathbb{P}(i \mid x)$ that input x belongs to class i.

Depending on the probability distribution, this can yield smooth functions as a model for a good classifier.

Depending on the application, both interpretations can be appropriate.

Deep versus shallow networks

A network is **deep** if it has many layers.

Observation: In applications, deep networks perform much better than shallow networks.

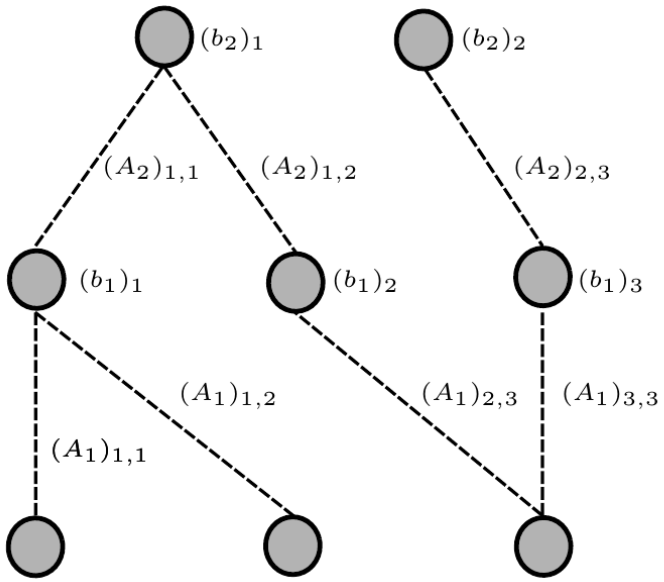
Approximation theory:

- Do deep networks provide provably better approximations than shallow networks?
- What about our specific class of classifier functions?

Complexity and closure properties of the class of ReLU networks

Complexity of neural networks

The affine linear mapping W_ℓ is defined by a matrix $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and a translation $b_\ell \in \mathbb{R}^{N_\ell}$ via $W_\ell(x) = A_\ell\, x + b_\ell$.



$$A_2 = \begin{pmatrix} (A_1)_{1,1} & (A_1)_{1,2} & 0 \\ 0 & 0 & (A_1)_{2,3} \end{pmatrix}$$

$$A_1 = \begin{pmatrix} (A_1)_{1,1} & (A_1)_{1,2} & 0 \\ 0 & 0 & (A_1)_{2,3} \\ 0 & 0 & (A_1)_{3,3} \end{pmatrix}$$

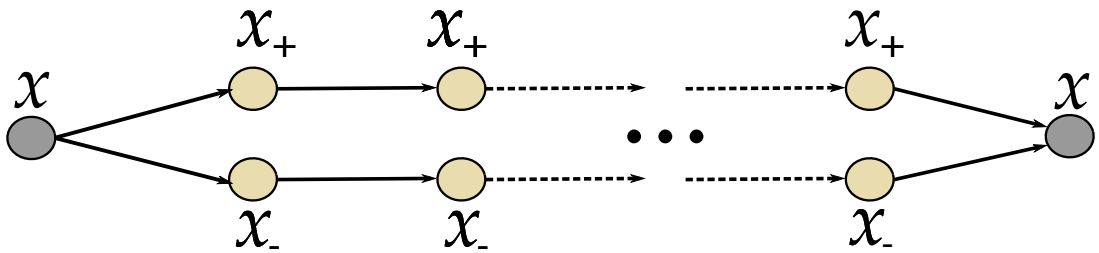
The **number of weights of a network** Φ is

$$M(\Phi) = \sum_{j \le L} (\|A_j\|_{\ell^0} + \|b_j\|_{\ell^0}).$$

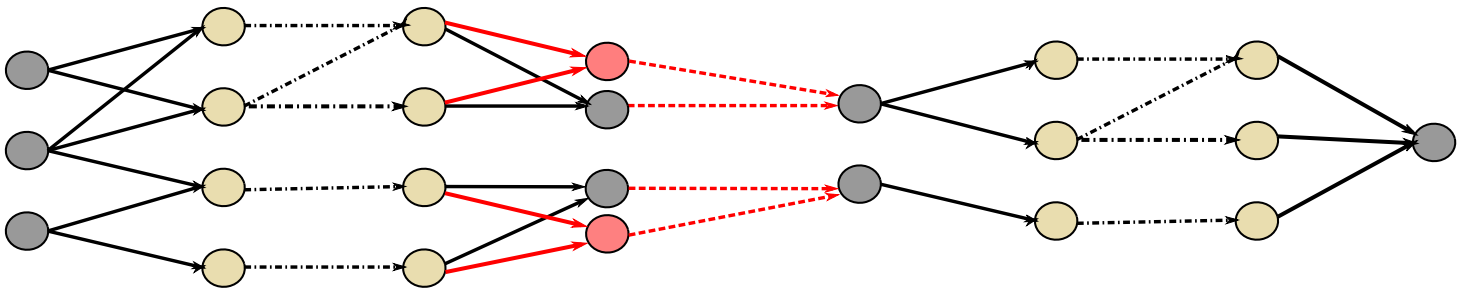
This is our notion of the complexity of a network.

The calculus of ReLU networks, Slide I

Producing identity: Since $x = \varrho(x) - \varrho(-x)$, we can reproduce the identity with a ReLU network $\Phi_{id} : \mathbb{R}^d \to \mathbb{R}^d$ having L layers and $\mathcal{O}(L \cdot d)$ weights.



Composition: Let $\Phi_1 : \mathbb{R}^d \to \mathbb{R}^{d'}, \Phi_2 : \mathbb{R}^{d'} \to \mathbb{R}^{d''}$ be two networks with L_1 and L_2 layers and M_1, M_2 weights. Then $\Phi_1 \circ \Phi_2$ can be realized by a ReLU network with $L_1 + L_2$ layers and $\mathcal{O}(M_1 + M_2 + d')$ weights.

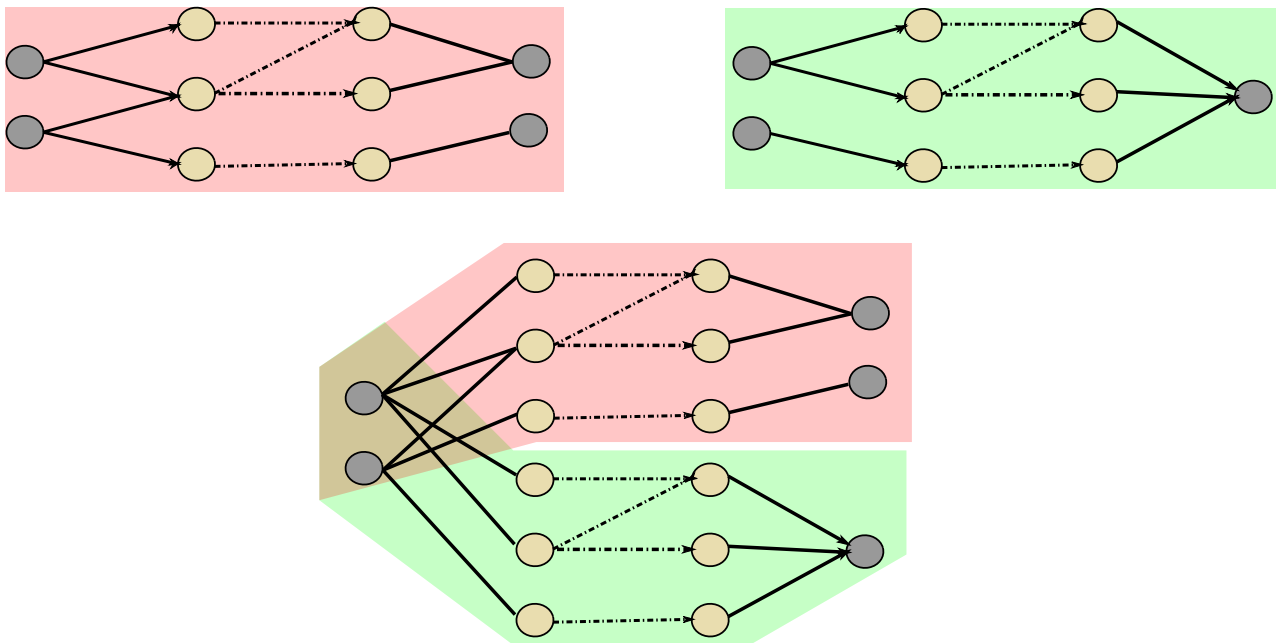


Increasing the depth: If $\Phi_1 : \mathbb{R}^d \to \mathbb{R}^{d'}$ has L_1 layers and M_1 weights, then, for every $L_2 > L_1$ there is a network Φ_2 with L_2 layers and $M_1 + \mathcal{O}((L_2 - L_1) \cdot d')$ weights with $\Phi_1(x) = \Phi_2(x)$ for all $x \in \mathbb{R}^d$.

The calculus of ReLU networks, Slide II

Cartesian product of two networks: If $\Phi_1 : \mathbb{R}^d \to \mathbb{R}^{d'}$ has L_1 layers and M_1 weights, and $\Phi_2 : \mathbb{R}^d \to \mathbb{R}^{d''}$ has L_2 layers and M_2 weights, then there exists $\Phi_3 : \mathbb{R}^d \to \mathbb{R}^{d'+d''}$ with $\max\{L_1, L_2\}$ layers and $M_1 + M_2 + \mathcal{O}\left((d' + d'') \cdot |L_1 - L_2|\right)$ weights such that

$$\Phi_3(x) = (\Phi_1(x), \Phi_2(x)) \qquad \forall\, x \in \mathbb{R}^d.$$



$\rightsquigarrow$ The class of ReLU networks is **closed under linear combinations**, and the complexity of the linear combination can be controlled.

Approximation with neural networks

Approximation with neural networks is easy

We have the **universal approximation theorem**:

Theorem (Cybenko; 1989, Hornik; 1991, Pinkus; 1999)

Let $K \subset \mathbb{R}^d$ compact, $f : K \to \mathbb{R}$ continuous, $\varrho : \mathbb{R} \to \mathbb{R}$ continuous and not a polynomial.

For each $\varepsilon > 0$, there exist $N \in \mathbb{N}$, $a_k, b_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$:

$$\left\| f - \sum_{k=1}^{N} a_k \cdot \varrho(\langle w_k, \bullet \rangle - b_k) \right\|_{L^\infty} \leq \varepsilon.$$

$\rightsquigarrow$ Neural networks with **two layers** can approximate **any continuous function** on a compact set **arbitrarily well**.

Approximation with neural networks is easy — Or is it?!

Universal approximation theorem:
Every continuous function can be approximated up to an error of $\varepsilon > 0$ using a two-layer neural network with N neurons.

$\rightsquigarrow$ **No information on the connection between ε and N!**

Goal for today:

Derive upper and lower bounds on the complexity of ReLU networks for approximating classifier functions.

Classifier functions

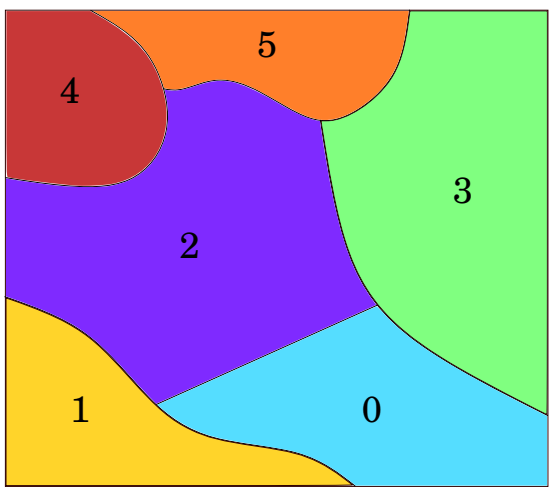
Model: A classifier function of regularity $\beta > 0$ is of the form

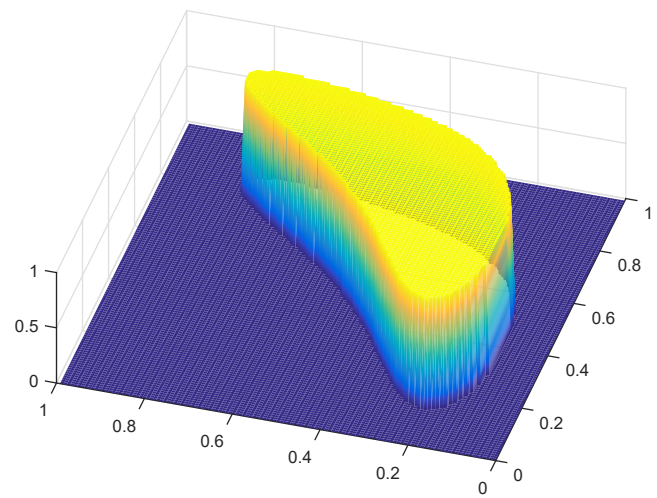
$$f = \sum_{i=1}^{n} a_i \cdot \mathbb{1}_{B_i},$$

where $a_i \in \mathbb{R}$ and $B_i \subset \mathbb{R}^d$ with $\partial B_i \in C^\beta$ for $i \in \{1, \ldots, n\}$.

Note: Class of ReLU networks of a certain complexity (up to constant factors) closed under linear combinations.

$\Longrightarrow$ Need only study approximation of $f = \mathbb{1}_B$.

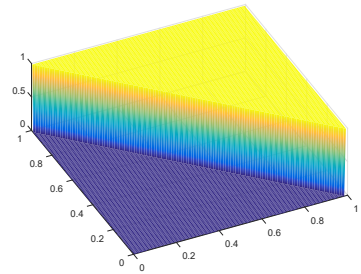




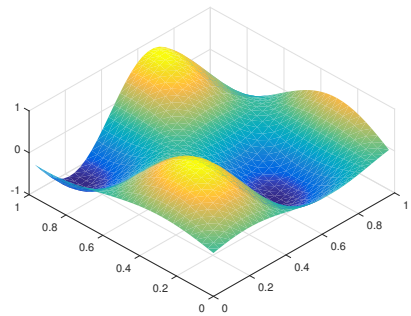
Roadmap for approximation

To approximate classifier functions, we proceed in four steps:

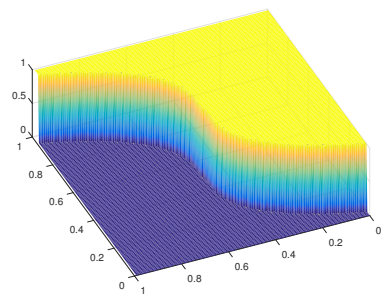
1. Jumps along straight lines:



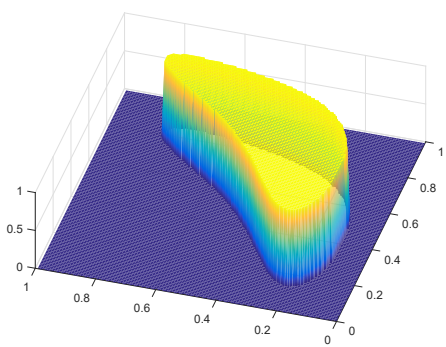
2. Smooth functions:



3. Horizon functions:



4. Classifier functions:



Jumps along straight lines

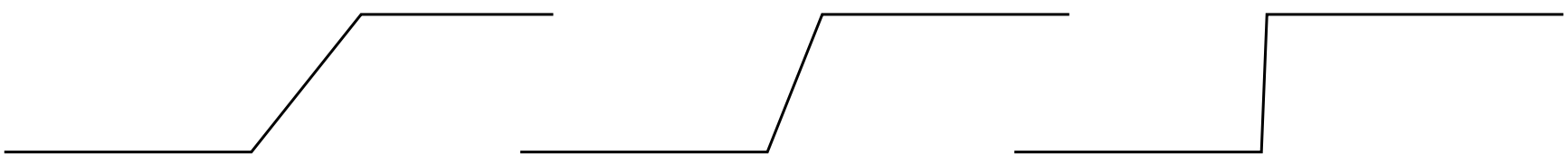
Lemma (Petersen, V.; 2017)

Let $H := \mathbb{1}_{[0,\infty)\times\mathbb{R}^{d-1}}$. For every $\varepsilon > 0$ there exists a ReLU network Φ_ε^H, with two layers, and five weights, such that

$$\|H - \Phi_\varepsilon^H\|_{L^2([-\frac{1}{2},\frac{1}{2}]^d)} \leq \varepsilon.$$

Furthermore, $|H(x) - \Phi_\varepsilon^H(x)| \leq \mathbb{1}_{0\leq x_1\leq \varepsilon^2}$.

Construction: $\Phi_\varepsilon^H(x) = \varrho(x_1/\varepsilon^2) - \varrho(x_1/\varepsilon^2 - 1)$.



Approximation of smooth functions — The function class

For $\beta, B > 0$, let

$$\mathcal{F}_{\beta,d,B} := \left\{ f \in C^n([-1/2,1/2]^d) : \|f\|_{C^\beta} \leq B < \infty \right\},$$

where for $\beta = n + \sigma$, $n \in \mathbb{N}_0$, $\sigma \in (0,1]$

$$\|f\|_{C^\beta} := \max \left\{ \max_{|\alpha| \leq n} \|\partial^\alpha f\|_{\sup}, \max_{|\alpha|=n} \sup_{x \neq y} \frac{|\partial^\alpha f(x) - \partial^\alpha f(y)|}{|x-y|^\sigma} \right\}.$$

Note: $\mathcal{F}_{1,d,B} \not\subseteq C^1$!

Approximation of smooth functions — Existing results

Theorem (Yarotsky; 2016)

For any $f \in \mathcal{F}_{n,d,1}$ and $\varepsilon \in (0,1)$, there is a ReLU network Φ_ε^f that

- *satisfies $\|f - \Phi_\varepsilon^f\|_{L^\infty} \leq \varepsilon$,*
- *has at most $c \cdot \varepsilon^{-d/n} \cdot (\log_2(1/\varepsilon) + 1)$ nonzero weights and neurons,*
- *has depth at most $c \cdot (\log_2(1/\varepsilon) + 1)$,*

with $c = c(d, n)$.

Comments:

- The error is measured with respect to the L^∞ norm.
- The depth of Φ_ε^f depends on ε.

Proof:

1. Implement (approximate) **multiplication**.
2. Implement (approximate) **Taylor polynomials** and **part. of unity**.

The approximate multiplication needs $\log_2(1/\varepsilon)$ layers.

Approximation of smooth functions — L^2-approximation

Theorem (Petersen, V.; 2017)

There are $c' > 0$ and $c = c(d, \beta, B) > 0$, such that for any function $f \in \mathcal{F}_{\beta,d,B}$ and any $\varepsilon \in (0, 1/2)$, there is a ReLU network Φ_ε^f that

- *satisfies $\|\Phi_\varepsilon^f - f\|_{L^2([-1/2,1/2]^d)} < \varepsilon$,*
- *has at most $c \cdot \varepsilon^{-d/\beta}$ nonzero weights,*
- *has at most $c' \cdot (1 + d^{-1}\beta) \cdot \log_2(2 + \beta)$ layers.*

Comments:

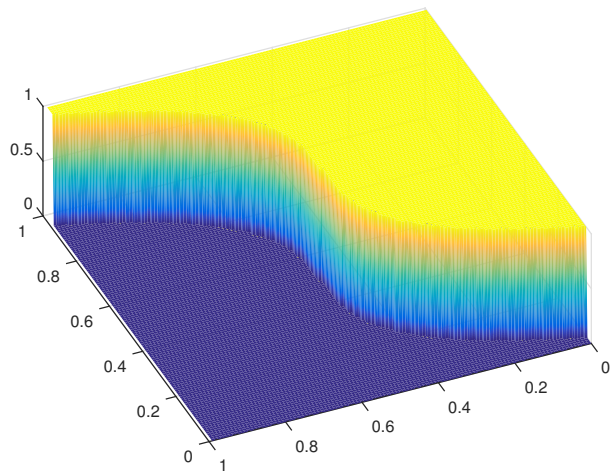
- Depth of Φ_ε^f independent of ε; it only depends on β, d.
- **Proof**:
  - Again: Taylor polynomials and partition of unity,
  - **But**: Use multiplication operator with depth independent of ε.

Approximation of Horizon functions — The function class

Let $d \in \mathbb{N}_{\geq 2}$, and $\beta, B > 0$. The **class of horizon functions** with parameters β, d, B is

$$\begin{aligned}\mathcal{HF}_{\beta,d,B} &:= \left\{\mathbb{1}_{x_i \geq \gamma(x_1,\ldots,\hat{x_i},\ldots,x_d)} \,:\, \gamma \in \mathcal{F}_{\beta,d-1,B} \text{ and } i \in \{1,\ldots,d\}\right\} \\ &= \Big\{f \circ T \,:\, f(x) = H(x_1 - \gamma(x_2,\ldots,x_d), x_2, \ldots, x_d), \\ &\qquad\qquad \gamma \in \mathcal{F}_{\beta,d-1,B},\, T \in \mathrm{Perm}(\mathbb{R}^d)\Big\} \\ &\subset L^\infty\big([-1/2,1/2]^d\big),\end{aligned}$$

where $H := \mathbb{1}_{[0,\infty)\times\mathbb{R}^{d-1}}$ is the Heaviside function.



Approximation of horizon functions — The theorem

Lemma (Petersen, V.; 2017)

There are $c' > 0$ and $c = c(d, \beta, B) > 0$, such that for any $f \in \mathcal{HF}_{\beta,d,B}$ and any $\varepsilon \in (0, 1/2)$, there is a ReLU network Φ_ε^f that

- *satisfies $\|\Phi_\varepsilon^f - f\|_{L^2([-1/2,1/2]^d)} < \varepsilon$,*
- *has at most $c \cdot \varepsilon^{-2(d-1)/\beta}$ nonzero weights,*
- *has at most $c' \cdot (1 + d^{-1}\beta) \cdot \log_2(\beta)$ layers.*

Comments:

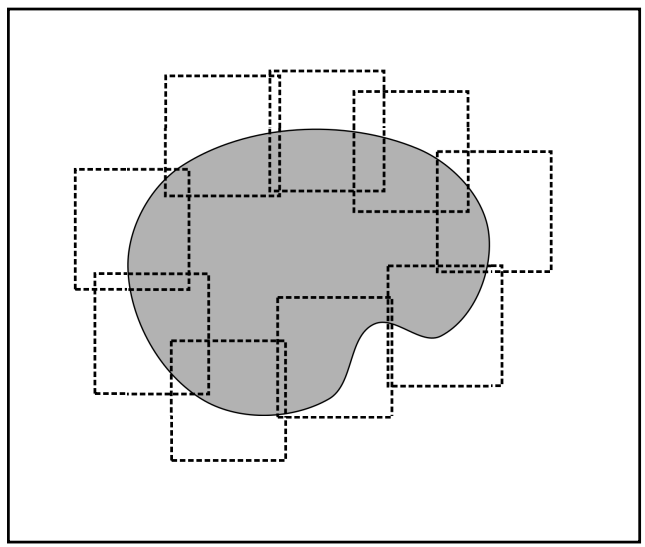
- The necessary depth depends on d, β, but not on ε.
- The size of Φ_ε^f approximating $f \in \mathcal{HF}_{\beta,d,B}$ is $\varepsilon^{-2(d-1)/\beta}$, compared to $\varepsilon^{-d/\beta}$ for $f \in \mathcal{F}_{\beta,d,B}$.

Sets with smooth boundary

Sets with smooth boundaries are those that are locally described by horizon functions.

Precisely, let $r \in \mathbb{N}$, $d \in \mathbb{N}_{\geq 2}$, and $\beta, B > 0$. Then we define

$$\mathcal{K}_{r,\beta,d,B} := \Big\{ \mathbb{1}_K \,:\, K \subset [-1/2,1/2]^d \text{ such that } \forall\, x \in [-1/2,1/2]^d : \exists\, f_x \in \mathcal{HF}_{\beta,d,B} : \mathbb{1}_K = f_x \text{ on } [-1/2,1/2]^d \cap \overline{B_{2^{-r}}}^{\|\cdot\|_{\ell^\infty}}(x) \Big\}.$$



Approximation of sets with smooth boundary

Theorem (Petersen, V.; 2017)

There are $c' > 0$ and $c = c(d, \beta, r, B) > 0$, such that for arbitrary $\mathbb{1}_K \in \mathcal{K}_{r,\beta,d,B}$ and $\varepsilon \in (0,1)$, there exists a ReLU network Φ_ε^K that

- *satisfies $\|\Phi_\varepsilon^K - \mathbb{1}_K\|_{L^2([-1/2,1/2]^d)} \leq \varepsilon$,*
- *has at most $c \cdot \varepsilon^{-2(d-1)/\beta}$ nonzero weights,*
- *has at most $c' \cdot \log_2(2+\beta) \cdot (1+\beta/d)$ layers.*

Comments

- Very smooth boundary $\Longrightarrow$ Good approximation with smaller networks.
- These networks are **smaller but deeper**.
- **Proof**: Approximation of horizon functions + Partition of unity.

By taking linear combinations: **Can approximate classifier functions** (piecewise constant functions).

Optimality

Optimality — Does it hold?

Previous part of the talk: Rates of approximation for certain function classes using neural networks.

Can these rates be improved?

No optimality in full generality:

Theorem (Maiorov, Pinkus; 1999)

There exists an activation function $\varrho_{\mathrm{weird}} : \mathbb{R} \to \mathbb{R}$ *that*

- *is analytic and strictly increasing,*
- *satisfies* $\lim_{x\to-\infty} \varrho_{\mathrm{weird}}(x) = 0$ *and* $\lim_{x\to\infty} \varrho_{\mathrm{weird}}(x) = 1$,

such that for any $d \in \mathbb{N}$, *any* $f \in C([0,1]^d)$ *and any* $\varepsilon > 0$, *there is a neural network* Φ_ε^f *with activation function* ϱ_{weird} *and* **two hidden layers of dimensions** $3d$ **and** $6d+3$ *such that* $\|f - \Phi_\varepsilon^f\|_{L^\infty} \leq \varepsilon$.

With this activation function, networks of **fixed size** *can approximate* **every** *continuous function* **arbitrarily well**.

Optimality

In order to obtain meaningful optimality results, we have to exclude pathological examples like the activation function ϱ_{weird}.

$\rightsquigarrow$ **Introduce additional assumptions!**

Options:

(A) Place restrictions on activation function (e.g. only consider ReLU), thereby excluding pathological examples like ϱ_{weird}.
($\rightsquigarrow$ VC dimension bounds)

(B) Place restrictions on the weights.
($\rightsquigarrow$ Information theoretical bounds, entropy arguments)

(C) Use still other concepts like continuous N-widths.

Lower bounds using VC dimension arguments

A **network architecture** $\mathcal{A}$ determines

- The input dimension $d = N_0$, and the number of layers L,
- the number of neurons N_ℓ on layer ℓ,
- the possible weights (entries of A_ℓ, b_ℓ) which are allowed to be nonzero.

For an architecture $\mathcal{A}$,

- $\mathcal{NN}(\mathcal{A})$ is the set of ReLU networks conform with $\mathcal{A}$.
- $W(\mathcal{A})$ is the maximal number of weights of networks in $\mathcal{NN}(\mathcal{A})$.

Theorem (Yarotsky; 2016)

Let $d, n \in \mathbb{N}$. For a network architecture $\mathcal{A}$ with input dimension d, let

$$\varepsilon_{\mathcal{A}} := \sup_{f \in \mathcal{F}_{n,d,1}} \inf_{\Phi \in \mathcal{NN}(\mathcal{A})} \|f - \Phi\|_{L^\infty L^\infty([-1/2,1/2]^d)}.$$

Let $p \geq 0$ and $c_1 > 0$. There is $c' = c'(d, n, p, c_1) > 0$, such that if $\varepsilon_{\mathcal{A}} < 1/2$, and if $\mathcal{A}$ has at most $c_1 \cdot \ln^p(1/\varepsilon_{\mathcal{A}})$ layers, then

$$W(\mathcal{A}) \geq c' \cdot \varepsilon_{\mathcal{A}}^{-d/n} / \ln^{1+2p}(1/\varepsilon_{\mathcal{A}}).$$

The approximation is always with respect to the L^∞ norm!

Another approach to lower bounds

- Approximation with L^∞-norm not appropriate in our setting (piecewise constant functions are discontinuous).
- It seems that the VC dimension arguments of Yarotsky do not generalize to the L^2 setting.

$$\rightsquigarrow \textit{ Use a different notion of optimality.}$$

Idea [Bölcskei, Grohs, Kutyniok, Petersen; 2017]: Assume that the weights of the approximating networks can be encoded using at most $K = K(\varepsilon)$ bits.

For $\varrho : \mathbb{R} \to \mathbb{R}$, denote by $\mathcal{NN}^{\varrho}_{M,K,d}$ the set of neural networks that

- have d-dimensional input and 1-dimensional output,
- use the activation function ϱ,
- have at most M nonzero weights,
- where **each weight can be encoded with K bits**.

Idea for getting lower bounds:

Assume: Good approx. of class $\mathcal{C} \subset L^2(\Omega)$ with $\mathcal{NN}^{\varrho}_{M,K,d}$

$\rightsquigarrow$ (Not too) lossy encoder/decoder for $\mathcal{C}$ with small code length

But: Such codecs must have a long code length!

Approximating networks yield codecs

Let $\mathcal{C} \subset L^2(\Omega)$.

- A **codec of length** ℓ for $\mathcal{C}$ is a pair (E_ℓ, D_ℓ) with
$$E_\ell : \mathcal{C} \to \{0,1\}^\ell, \qquad D_\ell : \{0,1\}^\ell \to L^2(\Omega).$$
- A codec (E_ℓ, D_ℓ) is ε**-accurate** if
$$\|f - D_\ell(E_\ell f)\|_{L^2} \leq \varepsilon \qquad \forall\, f \in \mathcal{C}.$$

Lemma (Bölcskei, Grohs, Kutyniok, Petersen; 2017)

Let ϱ be an activation function with $\varrho(0) = 0$. There is a constant $C = C(d) \in \mathbb{N}$, and an injective **encoding map**

$$E_{M,K} : \mathcal{NN}^{\varrho}_{M,K,d} \to \{0,1\}^{\flat(K,M)}, \quad \flat(K,M) := C \cdot M \cdot (K + \lceil \log_2 M \rceil).$$

Therefore: If

$$\forall\, \mathbb{1}_\Lambda \in \mathcal{K}_{r,\beta,d,B} \quad \exists\, \Phi_\Lambda \in \mathcal{NN}^{\varrho}_{M,K,d} \,:\, \|\mathbb{1}_\Lambda - \Phi_\Lambda\|_{L^2} \leq \varepsilon,$$

then $\mathcal{K}_{r,\beta,d,B}$ admits an ε-accurate codec of length $\flat(K,M)$.

Accurate codecs must be long

Using known results about the entropy numbers of

$$\{f \in C^{\beta}([0,1]^d) : \|f\|_{C^{\beta}} \leq B\}, \quad \text{equipped with } \|\cdot\|_{L^1},$$

and transferring these to the class $\mathcal{K}_{r,\beta,d,B}$, one can show the following:

Lemma (Petersen, V.; 2017. Chandrasekara et al.; 2009)

Each ε-accurate codec for $\mathcal{K}_{r,\beta,d,B}$ has length $\ell \gtrsim \varepsilon^{-2(d-1)/\beta}$, if $0 < \varepsilon \ll 1$.

Lower bounds for networks with encodable weights

Fix $\varrho : \mathbb{R} \to \mathbb{R}$ with $\varrho(0) = 0$ and $C_0 > 0$. For $\varepsilon > 0$, let

$$K_\varepsilon := \lceil C_0 \cdot \log_2(1/\varepsilon) \rceil.$$

Finally, for $f \in L^2([-1/2, 1/2]^d)$ define

$$M_\varepsilon(f) := \min \left\{ M \in \mathbb{N} : \exists\, \Phi \in \mathcal{NN}^{\varrho}_{M, K_\varepsilon, d} \text{ s.t. } \|f - \Phi\|_{L^2} \leq \varepsilon \right\}.$$

Theorem (Petersen, V.; 2017)

For $0 < \varepsilon \ll 1$, we have

$$\sup_{f \in \mathcal{K}_{\beta,d,B}} \sup_{f \in \mathcal{K}_{\beta,d,B}} M_\varepsilon(f) \gtrsim \varepsilon^{-2(d-1)/\beta} \Big/ \log_2(1/\varepsilon).$$

Theorem (Petersen, V.; 2017)

There is a function $f_0 \in \mathcal{K}_{r,\beta,d,B}$ and a null-sequence $(\varepsilon_k)_k$ with

$$M_{\varepsilon_k}(f_0) \gtrsim \varepsilon_k^{-2(d-1)/\beta} \Big/ (\log_2(1/\varepsilon_k))^2.$$

Optimality of depth

Theorem (Petersen, V.; 2017)

Let $\Omega \subset \mathbb{R}^d$ be nonempty, open, bounded, and connected. Let $f \in C^3(\Omega)$ be **nonlinear**. *Then there is a constant $C_f > 0$ satisfying*

$$\|f - \Phi\|_{L^p} \geq C_f \cdot (M(\Phi) + d)^{-2 \cdot L(\Phi)}$$

for all $1 \leq p < \infty$ and each ReLU neural network Φ.

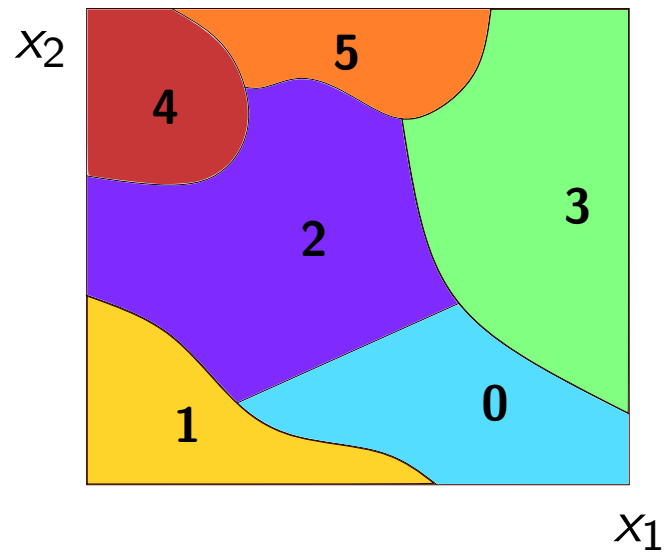
Comments:

- If $(\Phi_\varepsilon)_{\varepsilon > 0}$ satisfies $\|f - \Phi_\varepsilon\|_{L^p} \leq \varepsilon$ and $M(\Phi_\varepsilon) \lesssim \varepsilon^{-\theta}$, then for $0 < \varepsilon \ll 1$, we have $L(\Phi_\varepsilon) \geq c/\theta$ for an absolute constant $c > 0$.
- [Yarotsky; 2017] derived a similar result for L^∞ approximation.
- For $p = 2$, [Safran and Shamir; 2016] obtained a similar result.

Conclusion

Motivated by the problem of approximating classifier functions, we determined the optimal approximation rates of ReLU networks for

- Horizon functions,
- smooth functions,
- piecewise constant functions.



Also possible: Piecewise smooth functions.

A word on depth: Smoother functions allow better approximation rates, but **achieving these rates requires deeper networks**!

Thank you!

References:


P. Petersen, F. Voigtlaender
Optimal approximation of piecewise smooth functions using deep ReLU neural networks.
arXiv:1709.05289


H. Bölcskei, P. Grohs, G. Kutyniok, P. Petersen
Optimal Approximation with Sparsely Connected Deep Neural Networks.
arXiv:1705.01714


D. Yarotsky
Error bounds for approximation with deep ReLU networks.
Neural Netw., 2017


M. Anthony, P.L. Bartlett
Neural Network Learning: Theoretical Foundations.
Cambridge University Press, 2009


G.F. Clements,
Entropies of several sets of real valued functions.
Pacific J. Math., 1963


M. Telgarsky,
Representation benefits of deep feedforward networks.
arXiv:1509.08101


I. Safran, O. Shamir
Depth-width tradeoffs in approximating natural functions with neural networks.
arXiv:1610.09887

Proof that accurate codecs must be long

Observation: If $f_1, \dots, f_{2^N} \in \mathcal{C}$ satisfy $\|f_i - f_j\|_{L^2} > 2\varepsilon$, every ε-accurate codec (E, D) for $\mathcal{C}$ has length $\ell \geq N$.

Proof.

If $\ell < N$, then $|\{0,1\}^\ell| < 2^N$, so that $E(f_i) = E(f_j)$ for certain $i \neq j$. Hence,

$$2\varepsilon < \|f_i - f_j\|_{L^2} \leq \|f_i - D(E(f_i))\|_{L^2} + \|D(E(f_j)) - f_j\|_{L^2} \leq \varepsilon + \varepsilon. \qquad \square$$

[Clements; 1963] estimated the entropy numbers of $\mathcal{F}_{\beta,d,1}$ equipped with the L^1 metric. This yields the following:

Lemma

For $0 < \varepsilon \ll 1$, there is $N \geq 2^{c \cdot \varepsilon^{-(d-1)/\beta}}$ and $f_1, \dots, f_N \in \mathcal{F}_{\beta,d-1,B}$ with $\|f_i - f_j\|_{L^1} \geq 5\varepsilon$ for $i \neq j$.

Finally, for $0 < B \ll 1$, we have $\|\mathrm{HF}_\gamma - \mathrm{HF}_\theta\|_{L^2} = \|\gamma - \theta\|_{L^1}^{1/2}$ with

$$\mathrm{HF}_\gamma = \mathbb{1}_{x_1 \leq \gamma(x_2,\dots,x_d)} \quad \text{for} \quad \gamma \in \mathcal{F}_{\beta,d-1,B}.$$

For $0 < \varepsilon \ll 1$, every ε-accurate codec for $\mathcal{K}_{\beta,d,B}$ has length $\ell \gtrsim \varepsilon^{-2(d-1)/\beta}$.

Proof idea for the optimality of depth

Theorem (Petersen, V.; 2017)

Let $\Omega \subset \mathbb{R}^d$ be nonempty, open, bounded, and connected. Let $f \in C^3(\Omega)$ be nonlinear. Then there is a constant $C_f > 0$ satisfying

$$\|f - \Phi\|_{L^p} \geq C_f \cdot (M(\Phi) + d)^{-2 \cdot L(\Phi)}$$

for all $1 \leq p < \infty$ and each ReLU neural network Φ.

Proof idea:

- [Telgarsky; 2015] showed for a ReLU network Φ with L layers that $t \mapsto \Phi(tv + v_0)$ is affine-linear with $\mathcal{O}(M(\Phi)^L)$ many "pieces".
- We have $\|\alpha \cdot x^2 - (\beta x + \gamma)\|_{L^p([a,b])} \gtrsim |\alpha| \cdot (b-a)^{2+p^{-1}}$.
- A C^3 function with bounded f''' is approximated to order $(b-a)^3$ by its Taylor polynomial on $[a, b]$.
- Thus, $\|f(x) - (\beta x + \gamma)\|_{L^p([a,b])} \gtrsim |b-a|^{2+p^{-1}}$ for $0 < b - a \ll 1$.
- Using a Fubini argument, reduce to the one-dimensional case.