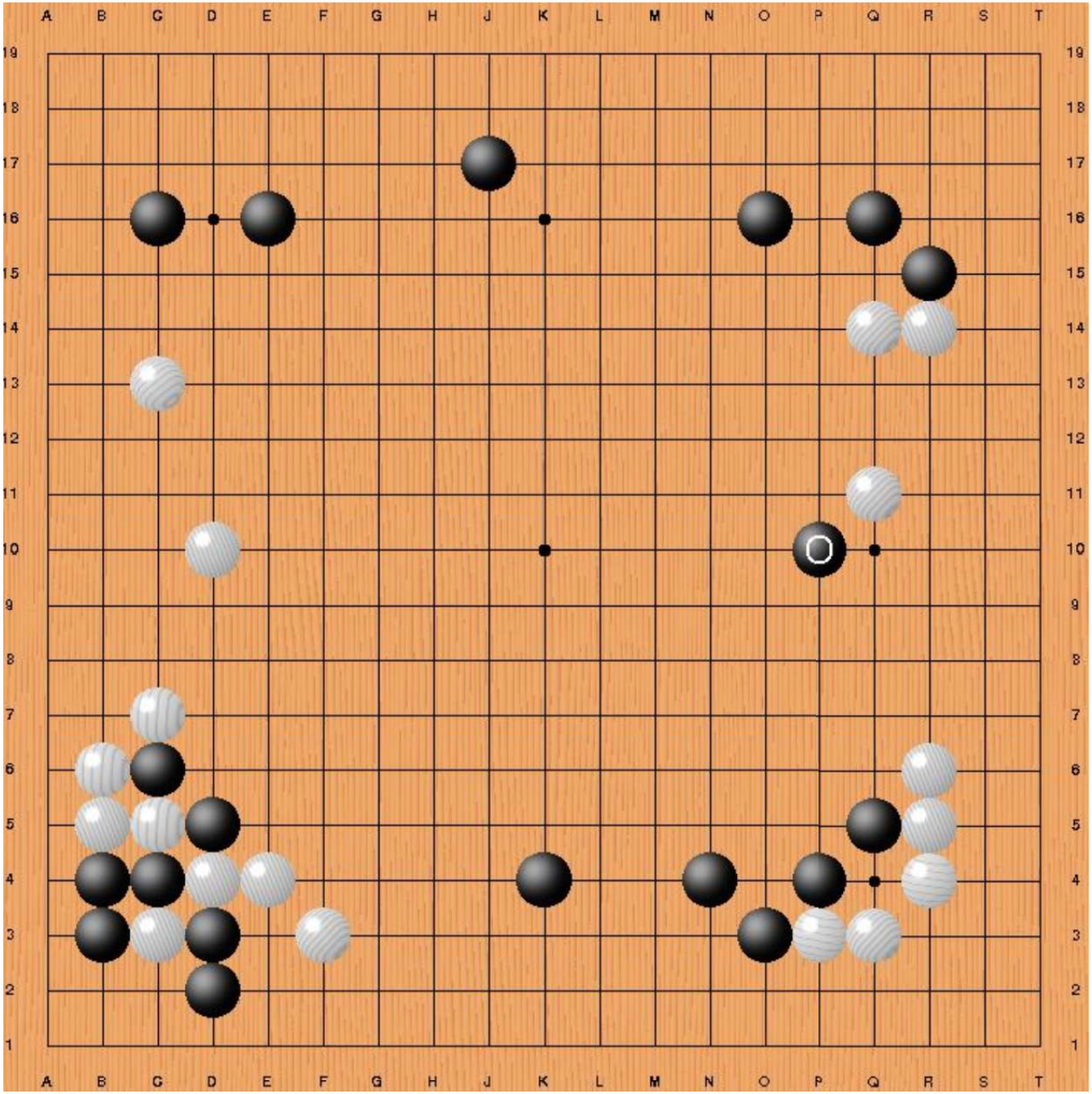
Approximation Theory of Deep Neural Networks: Part 1

Research school: HiDaDeeL

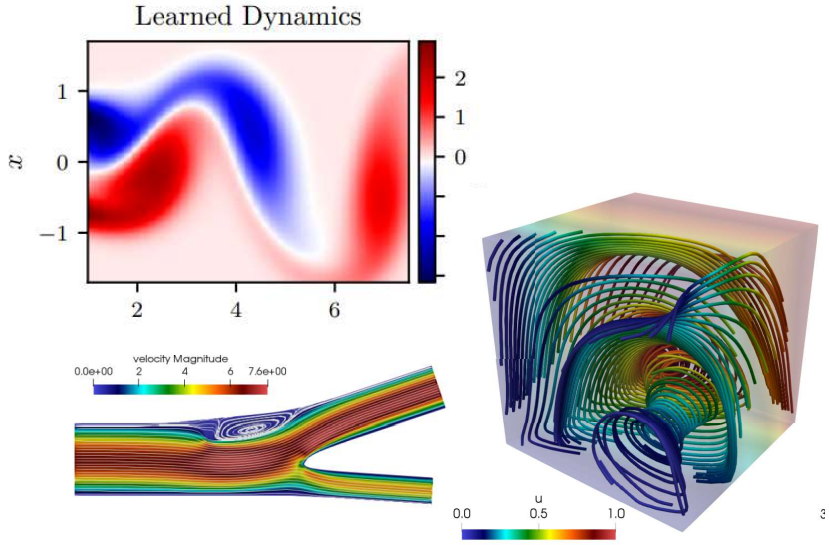
16.05 - 20.05. 2022

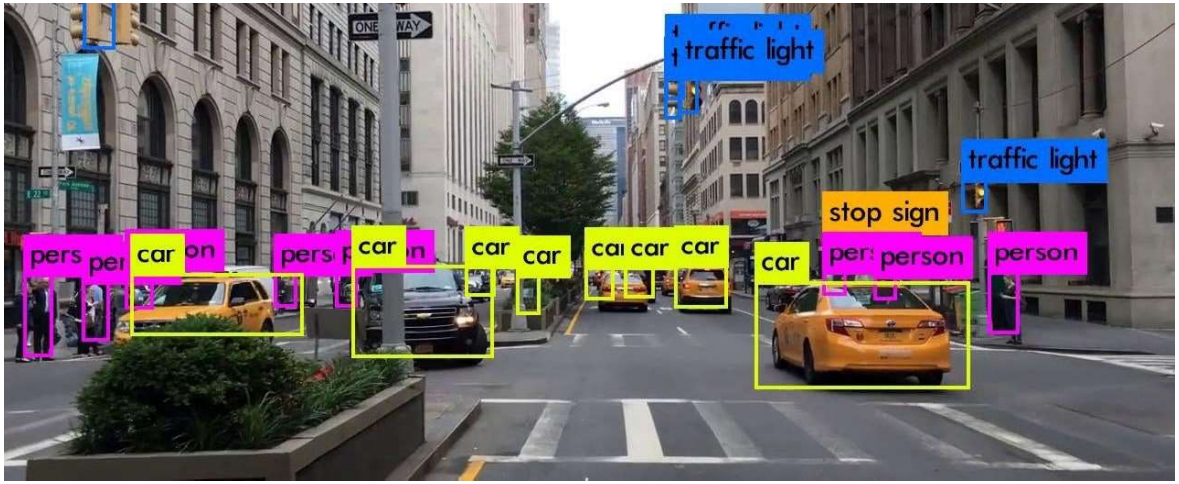
Nantes

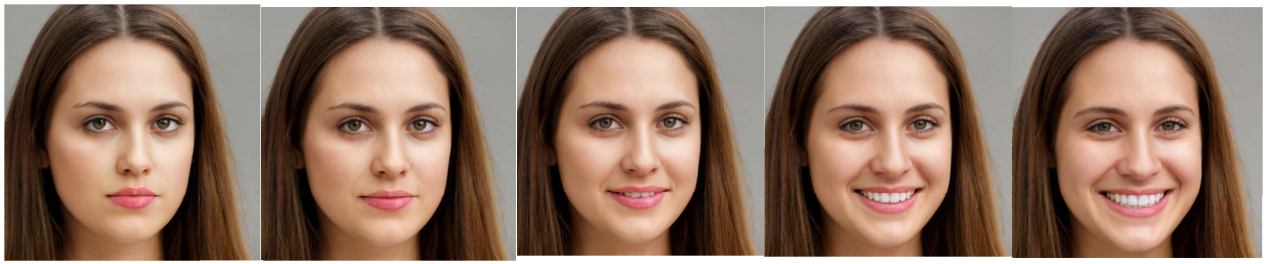
Philipp Petersen


A B C D E F G H J K L M N O P Q R S T
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Deep Learning







Support The Guardian
Available for everyone, funded by readers
Contribute → Subscribe →
Search jobs | Sign in | Search | International
The Guardian
News | Opinion | Sport | Culture | Lifestyle | More
The Guardian view | Columnists | Cartoons | Opinion videos | Letters

Opinion
Artificial intelligence (AI)

This article is more than 2 months old

A robot wrote this entire article. Are you scared yet, human?

GPT-3

Tue 8 Sep 2020 09.45 BST

69,889 1,188

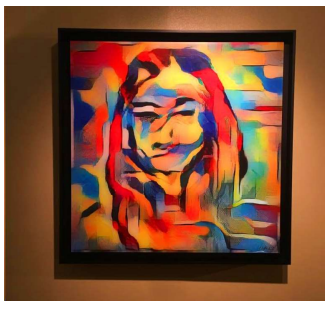
We asked GPT-3, OpenAI's powerful new language generator, to write an essay for us from scratch. The assignment? To convince us robots come in peace

For more about GPT-3 and how this essay was written and edited, please read our editor's note below

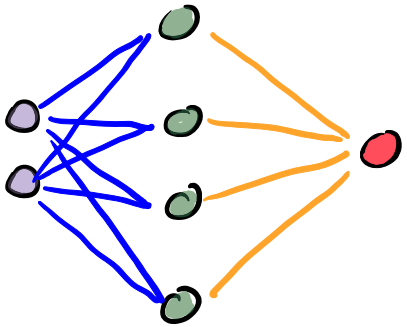
TEXT PROMPT

an armchair in the shape of an avocado [...]

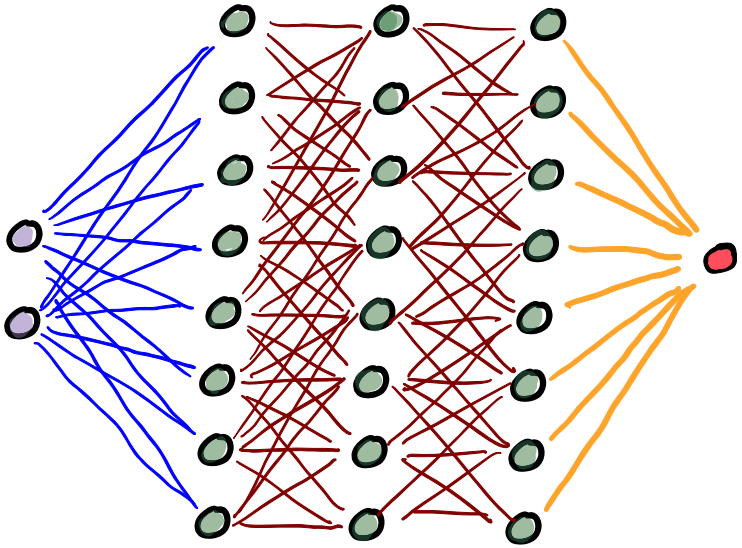
AI-GENERATED IMAGES



Key player: Neural network

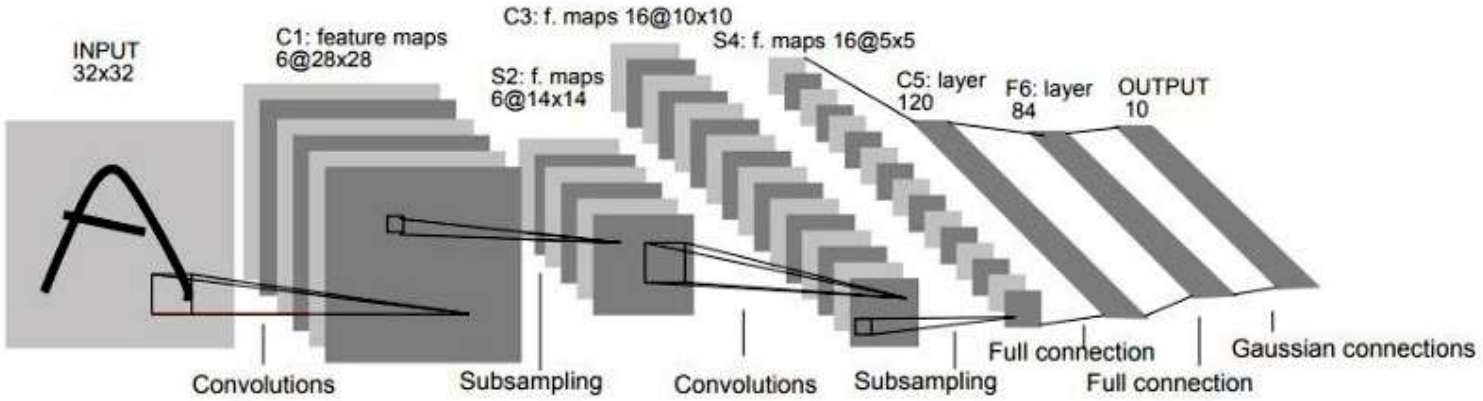


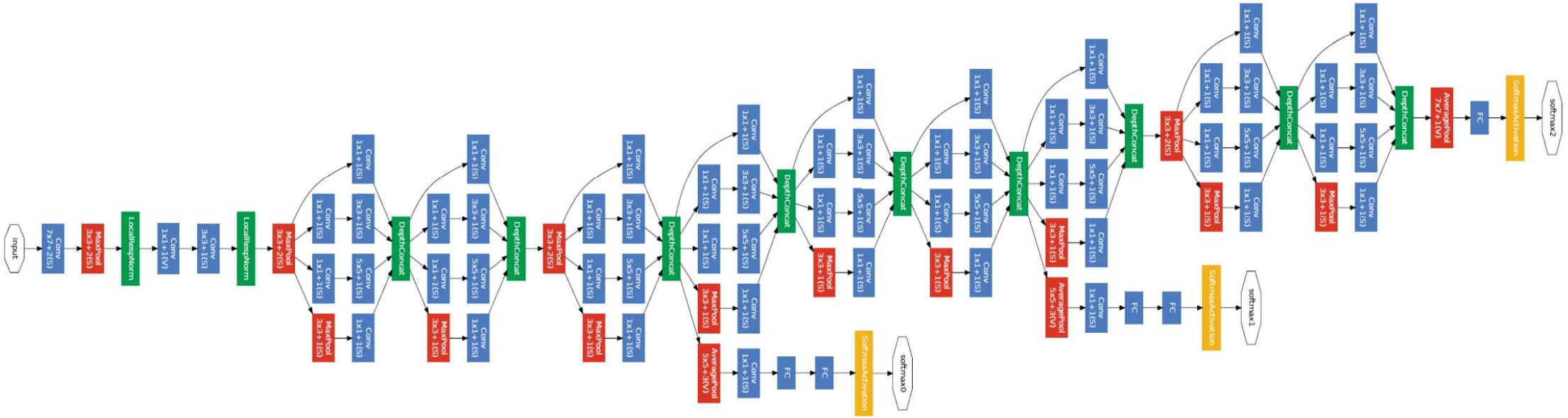
$$\Phi(x) = \sum_{i=1}^{4} c_i \rho(\langle a_i, x \rangle + b)$$



$$\Phi(x) = T_4 \rho(T_3 \rho(T_2 \rho(T_1(x))))$$

In practice:


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections


input
softmax0
softmax1
softmax2

Supervised deep learning

Data: $(x_i, y_i)_{i=1}^M$, images/labels

Objective: $\min_{h \in \text{Neural network}} \sum_{i=1}^{M} |h(x_i) - y_i|^2 \to \min$

1. Why is it possible to get the minimum small?
2. Why do deep NNs perform better?
3. How can this work, if X has a massive dimension (like an image)

Neural networks (More precise)

Def: A neural network (NN) with input dimension $d \in \mathbb{N}$, L layers, and architecture $(d, N_1, \dots, N_L) \in \mathbb{N}^{L+1}$ is a series of matrix-vector tuples

$$\Phi = ((A_1, b_1), \dots, (A_L, b_L)),$$

where $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$, for $\ell = 1, \dots L$, $N_0 = d$.

We call $N(\Phi) = d + \sum_{\ell=1}^{L} N_\ell$ the number of neurons,

$$M(\Phi) = \sum_{\ell=1}^{L} M_\ell(\Phi) := \sum_{\ell=1}^{L} \|A_\ell\|_0 + \|b_\ell\|_0$$ the number of weights.

Realisations of neural networks

Def:

For a NN Φ and an activation function $\rho : \mathbb{R} \to \mathbb{R}$, we def. the associated realisation of the NN as

$$R(\Phi) : \mathbb{R}^d \longrightarrow \mathbb{R}^{N_L},$$

$$R(\Phi)(x) = x_L,$$

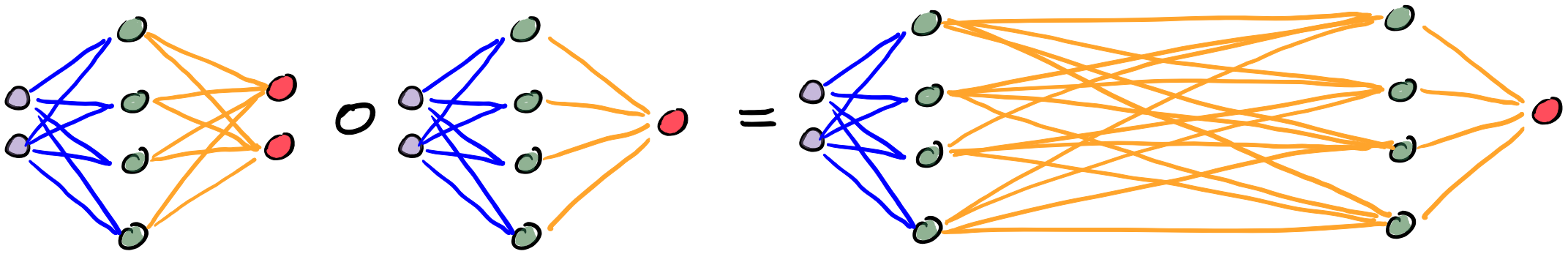
where x_L results from

$$x_0 = x,$$

$$x_\ell = \rho(A_\ell x_{\ell-1} + b_\ell), \quad \text{for } \ell = 1, \dots, L-1,$$

$$x_L = A_L x_{L-1} + b_L$$

Basic operations (concatenation)



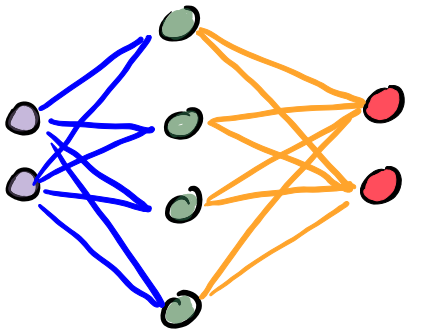
Def: Let $L_1, L_2 \in \mathbb{N}$ and let $\Phi_1 = ((A_1^1, b_1^1), \dots, (A_{L_1}^1, b_{L_1}^1))$, $\Phi_2 = ((A_1^2, b_1^2), \dots, (A_{L_2}^2, b_{L_2}^2))$, be such that the output dim of Φ_2 is equal to the input dimension of Φ_2. Then

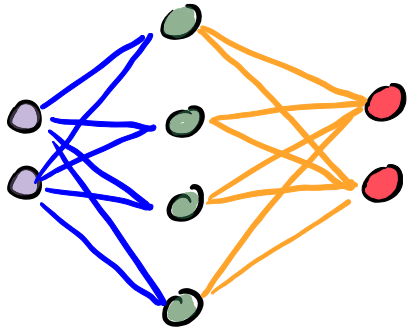
$$\Phi_1 \bullet \Phi_2 = \left((A_1^2, b_1^2), \dots, (A_{L_2-1}^2, b_{L_2-1}^2), (A_1^1 A_{L_2}^2, A_1^1 b_{L_2}^2 + b_1^1), (A_2^1, b_2^1), \dots, (A_{L_1}^1, b_{L_1}^1)\right),$$

is called the concatenation of Φ_1 and Φ_2

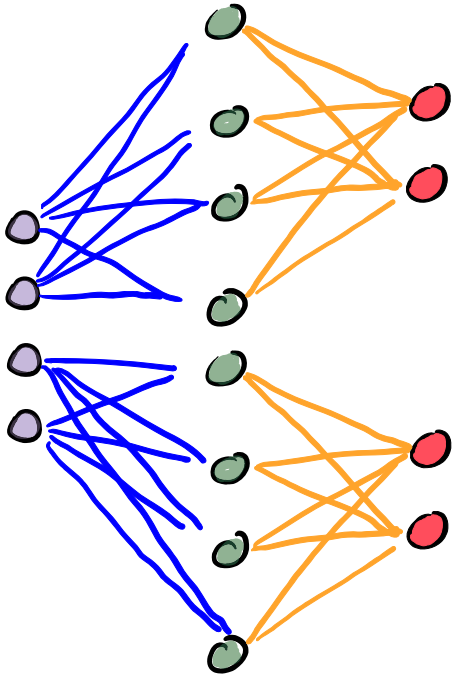
Ex.: $R(\Phi_1 \bullet \Phi_2) = R(\Phi_1) \circ R(\Phi_2)$

Basic operations (Parallelisation)

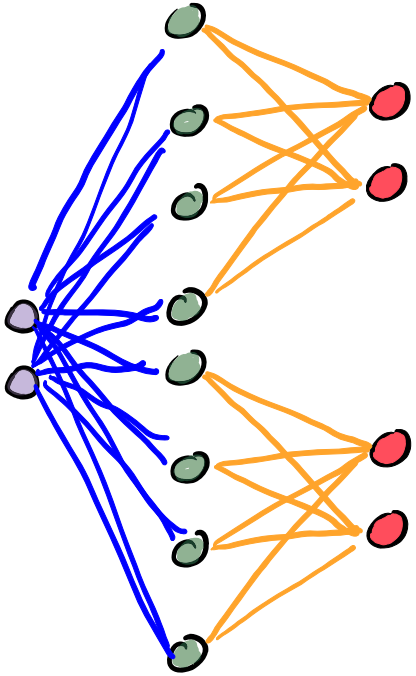




full parallelisation



Parallelisation with shared inputs



Basic operations (Parallelisation)

Def: Let $L, d_1, d_2 \in \mathbb{N}$ and let $\Phi_1 = ((A_1^1, b_1^1), \dots (A_L^1, b_L^1))$,

$\Phi_2 = ((A_1^2, b_1^2), \dots (A_L^2, b_L^2))$. We define

1. $P(\Phi_1, \Phi_2) = ((\hat{A}_1, \hat{b}_1), (\tilde{A}_2, \tilde{b}_2), \dots, (\tilde{A}_L, \tilde{b}_L))$ (only if $d_1 = d_2$)

2. $FP(\Phi_1, \Phi_2) = ((\tilde{A}_1, \tilde{b}_1), (\tilde{A}_2, \tilde{b}_2), \dots, (\tilde{A}_L, \tilde{b}_L))$,

where

$$\hat{A}_1 = \begin{pmatrix} A_1^1 \\ A_1^2 \end{pmatrix}, \quad \hat{b}_1 = \begin{pmatrix} b_1^1 \\ b_1^2 \end{pmatrix}, \quad \tilde{A}_\ell = \begin{pmatrix} A_\ell^1 & 0 \\ 0 & A_\ell^2 \end{pmatrix}, \quad \tilde{b}_\ell = \begin{pmatrix} b_\ell^1 \\ b_\ell^2 \end{pmatrix},$$

for $\ell = 1, \dots L$.

$\Rightarrow$ summing.

Operations

- $M(P(\Phi_1, \Phi_2)) = M(FP(\Phi_1, \Phi_2))$

  $= M(\Phi_1) + M(\Phi_2)$,

- $R(P(\phi_1, \phi_2))(x) = \big(R(\phi_1)(x), R(\phi_2)(x)\big)$,
- $R(FP(\phi_1, \phi_2))(x_1, x_2) = \big(R(\phi_1)(x_1), R(\phi_2)(x_2)\big)$.

Operations (Extension)

Can we make NNs deeper without changing realisation?

A.) Yes, if there ex. NN $\Phi : d$ with $R(\Phi) = id$.

$\Rightarrow$ $\Phi_{id} \bullet \Phi$ is deeper than Φ with same real.

B.) If no, then approximately.

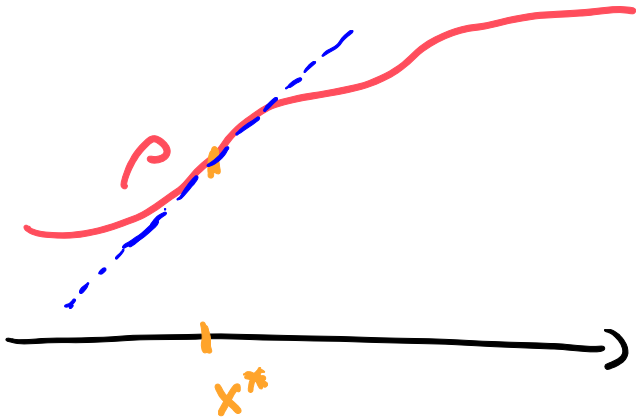
Prop: Let $d \in \mathbb{N}$, $K \subset \mathbb{R}^d$ compact and $\rho : \mathbb{R} \to \mathbb{R}$ differentiable and not constant on an open set.

Then, for every $\varepsilon > 0$, $\exists$ NN $\Phi = ((A_1, b_1), (A_2, b_2))$ s.t. $A_1, A_2 \in \mathbb{R}^{d \times d}$, $b_1, b_2 \in \mathbb{R}^d$, $M(\Phi) \leq 4d$ and

$$|R(\Phi)(x) - x| < \varepsilon \quad \text{, for all } x \in K.$$

Proof:

Proof for $d=1$ is enough. We can apply full parallelisation to get to arbitrary $d \in \mathbb{N}$.



$$\Rightarrow \quad \lambda \frac{\rho\left(\frac{x}{\lambda} + x^*\right) - \rho(x^*)}{\rho'(x^*)} \to x$$

for $\lambda \to 0$.

Choose

$$A_1 = \frac{1}{\lambda}, \quad b_1 = x^*, \quad A_2 = \frac{\lambda}{\rho'(x^*)}, \quad b_2 = -\frac{\lambda \rho(x^*)}{\rho'(x^*)}.$$

Approximation theory

Function class $\mathcal{C} \subset L^2(K) / C(K)$ for $K \subset \mathbb{R}^d$ compact.

Neural nets with M parameters NN_M.

$$\sup_{f \in \mathcal{C}} \inf_{\Phi \in NN_M} \| f - R(\Phi) \| = h(M).$$

What is $h(M)$?

Universality

If ρ is a locally bounded, p.w. continuous function and not a polynomial, then

$$h(M) \longrightarrow \mathcal{O}.$$

This result is called universal approximation theorem.

[Hornik; 1989], [Cybenko; 1989], [Leshno, Lin, Pinkus, Schocken; 1993].

Proof $\longrightarrow$ Lecture by Sophie Langer.

Theorem: [Mhaskar, 1993]

Let $k, d \in \mathbb{N}$, $K > 0$ and $\rho : \mathbb{R} \to \mathbb{R}$ be sigmoidal of order $q \geq 2$. Then, for every $f \in C^s([0,1]^d)$ with $\|f\|_{C^s} \leq 1$ and every $\varepsilon > 0$, there ex. a NN Φ:

$$\|f - R(\Phi)\|_\infty \leq \varepsilon$$

where $M(\Phi) \lesssim \varepsilon^{-\frac{d}{s}}$,

$$L(\Phi) = \lceil \log_2(d) \rceil + \max\{\log_q(\lceil s \rceil - 1), 0\} \rceil + 1.$$

$\rightsquigarrow h(M) \lesssim M^{-\frac{s}{d}}$.

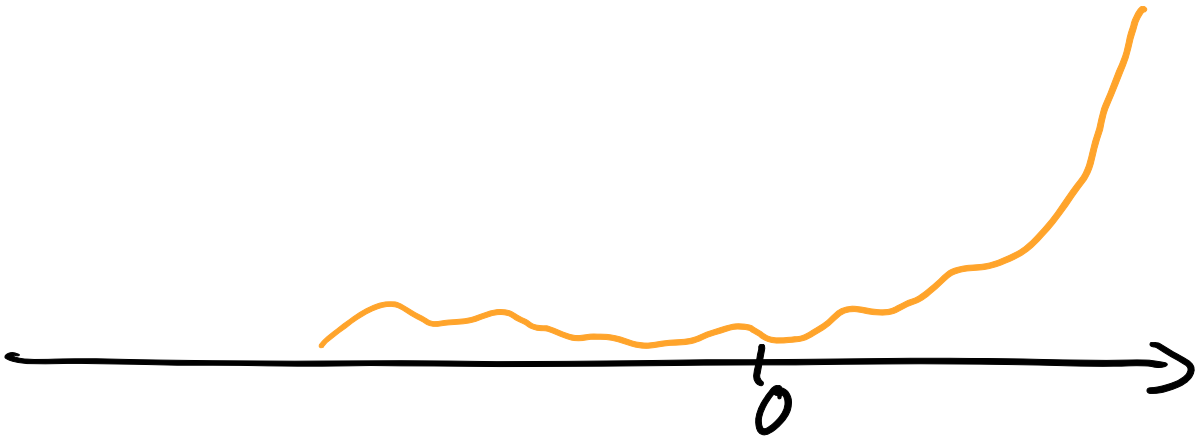
Higher-order sigmoidal functions

Def: A function $\rho : \mathbb{R} \to \mathbb{R}$ is called sigmoidal of order $q \in \mathbb{N}$, if $\rho \in C^{q-1}(\mathbb{R})$ and

$$\frac{\rho(x)}{x^q} \to 0 \quad \text{for } x \to -\infty,$$

$$\frac{\rho(x)}{x^q} \to 1 \quad \text{for } x \to \infty, \text{ and}$$

$$|\rho(x)| \leq (1+|x|)^q \quad \text{for all } x \in \mathbb{R}.$$



Spline approximation

Cardinal B-spline: $N_k(x) = \frac{1}{(k-1)!} \sum_{\ell=0}^{k} (-1)^\ell \binom{k}{\ell} (x-\ell)_+^{k-1}$.

Shifts and scaling: $N_{\ell,t,k} = N_k(2^\ell(\cdot - t))$.

Multidimensional: $N_{\ell,t,k}^d = \prod_{i=1}^{d} N_k(2^\ell(\cdot - t))$.

Set: $B^k = \{ N_{\ell,t,k}^d : \ell \in \mathbb{N}, t_\ell \in 2^{-\ell}\mathbb{Z}^d \}$.

Thm:

For $f \in C^s([0,1]^d)$, $N \in \mathbb{N}$, $s \leq k$ $\exists (B_i)_{i=1}^N \subset B^k$,

$(c_i)_{i=1}^N \subset \mathbb{R}$:

$$\| f - \sum_{i=1}^{N} c_i B_i \|_{L^p} \leq N^{-\frac{s}{d}} \|f\|_{C^s}.$$

Reapprox. of splines

Can we build this with a NN?

$$N_k(x) = \frac{1}{(k-1)!} \sum_{\ell=0}^{k} (-1)^\ell \binom{k}{\ell} (x-\ell)_+^{k-1}$$

A.) It is enough to build this.
The rest follows from parallelisation!

B.) Note that

$$\frac{1}{\lambda^q} \rho(\lambda x) \longrightarrow x_+^q .$$

c.) $x_t^q \rightsquigarrow x_t^{q-1}$ by

derivative trick.

$$x \longmapsto \frac{\rho(\lambda(x+\varepsilon)) - \rho(\lambda x)}{\lambda \varepsilon}$$

is realisation of a small NN.

$\Rightarrow$ $x \mapsto (x-\ell)_t^{q-j}$ can be approx. arbitrarily well by small NN for all $j < k$.

D.) If $k > q$, then we note that

$$\lambda^{-q^n} \underbrace{\rho \circ \rho \circ \dots \circ \rho}_{n\text{-times}} (\lambda^{-n} x) \to x_+^{q^n}.$$

Same argument as before

$\Rightarrow$ $x \longmapsto (x-\ell)^{k-1}$ can be approxed by the realisation of a small NN with fewer than $\max\{\log_q(s), 1\} + 1$ layers.

E.) $N_k \to N_{k,\ell,t}$ is only affine trafo.

(only finish layer needs to be adapted.)

F.) $N_{k,\ell,t} \longrightarrow N_{k,\ell,t}^{d}$ requires multiplication.

Note:

$$xy = \frac{1}{4}\left[(x+y)^2 - (x-y)^2\right] = \frac{1}{4}\Big[(x+y)_+^2 + (-x-y)_+^2 - (x-y)_+^2 - (y-x)_+^2\Big].$$

We can approx these with a NN with two layers.

Theorem: [Mhaskar, 1993]

Let $k, d \in \mathbb{N}$, $K > 0$ and $\rho : \mathbb{R} \to \mathbb{R}$ be sigmoidal of order $q \geq 2$. Then, for every $f \in C^s([0,1]^d)$ with $\|f\|_{C^s} \leq 1$ and every $\varepsilon > 0$, there ex. a NN Φ:

$$\|f - R(\Phi)\|_\infty \leq \varepsilon$$

where $M(\Phi) \lesssim \varepsilon^{-\frac{d}{s}}$,

$$L(\Phi) = \lceil \log_2(d) \rceil + \max\{\log_q(\lceil s \rceil - 1), 0\} \rceil + 1.$$

$\rightsquigarrow h(M) \lesssim M^{-\frac{s}{d}}$.

spline result

multiplication of d elements

approx of $(x-\ell)^{k-1}$

Remarks:

- The idea to lift approximation results from dictionaries to NNs has been applied very often. (Wavelets, sinusoids, finite elements, e.t.c.)
  [general principle]

- More efficient approx. by high order polynomials requires deeper networks.

- Multiplication and x_+^k approx. were possible due to special property of ρ.

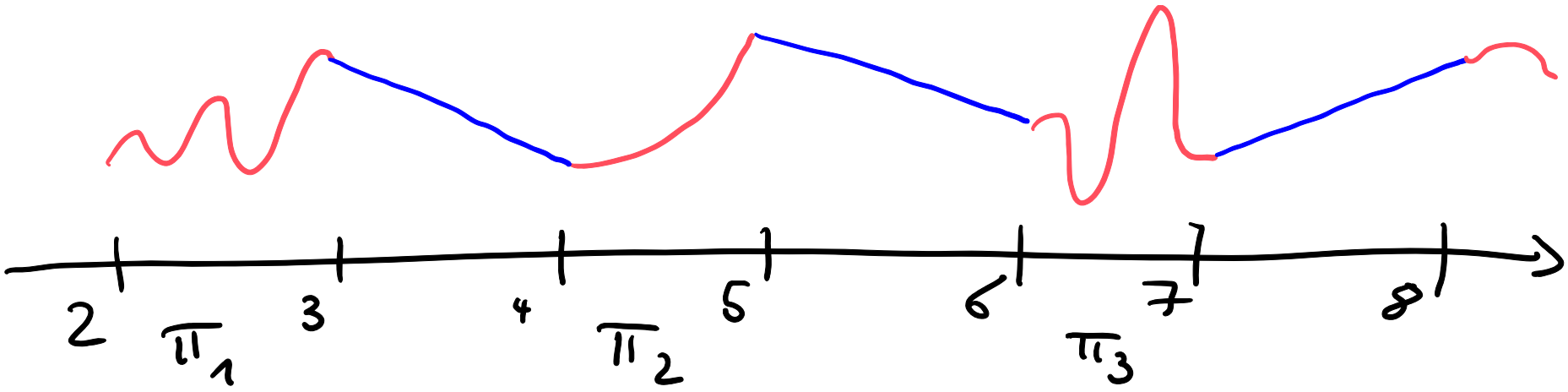
Even better activation functions.

Prop: There exists a continuous p.w. polynomial activation function $\rho : \mathbb{R} \to \mathbb{R}$ such that for every $f \in C([0,1])$ and every $\varepsilon > 0$ there is a NN $\Phi_{f,\varepsilon}$ s.th $M(\Phi_{f,\varepsilon}) \leq 3$, $L(\Phi_{f,\varepsilon}) = 2$ and

$$\| f - R(\Phi_{f,\varepsilon}) \|_\infty \leq \varepsilon.$$

Proof:

$(\pi_i)_{i=1}^{\infty}$ = { dense set of polynomials in $C([0,1])$. }



For $f \in C([0,1])$, $\varepsilon > 0 \Rightarrow \exists\ \pi_j : \|f - \pi_j\| < \varepsilon$.

$\Rightarrow \| f - p(\cdot - 2j) \|_\infty < \varepsilon$.

Same result for $d > 1$ possible?

Thm. (Kolmogorov–Arnold superposition theorem)

For every $d \in \mathbb{N}$, there are $2d^2 + d$ univariate, continuous, and increasing functions $\phi_{p,q}$, $p = 1 \dots d$, $q = 1, \dots, 2d+1$ s.t. for every $f \in C([0,1]^d)$, we have that, for all $x \in [0,1]^d$:

$$f(x) = \sum_{q=1}^{2d+1} g_q \left(\sum_{p=1}^{d} \phi_{p,q}(x_p) \right),$$

where g_q, $q = 1, \dots, 2d+1$, are univariate cont. functions depending on f.

Neural networks

$$f(x) = \sum_{q=1}^{2d+1} g_q \left(\sum_{p=1}^{d} \phi_{p,q}(x_p) \right),$$

1d-result

parallelisation

1d-result

full parallelisation

$\rightsquigarrow$ Φ with 3 layers, and $C(d)$ weights suffices to represent <u>every</u> cont. function arbitrarily well.

Remark:

- No fundamental limit on approx. power from NN architecture.

- A dense subset of $C([0,1]^d)$ is parametrised by a small number of real numbers.

- ρ from results above is in practice useless.

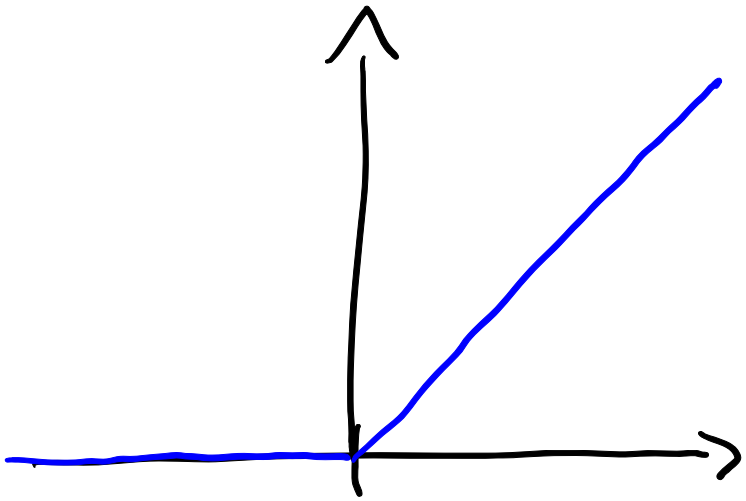
  (recall continuous nonlinear n-width)

The ReLU

Most frequently used activation function:

$$\rho : \mathbb{R} \to \mathbb{R}^+$$

$$\rho(x) = \max\{0, x\}.$$



Properties:

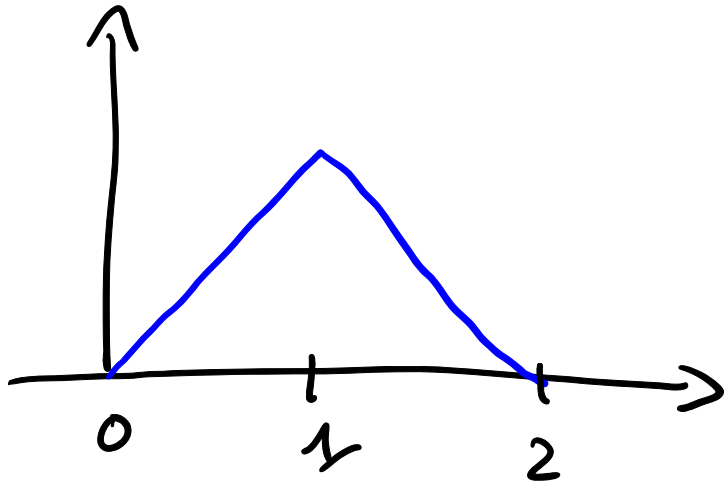
piecewise linear, homogeneous, easy to evaluate, derivative is simple,

$\rho' \in \{0, 1\}$, $x = \rho(x) - \rho(-x)$, $|x| = \rho(x) + \rho(-x)$,

$$\max\{a, b\} = \frac{a+b}{2} - \frac{|a-b|}{2}.$$

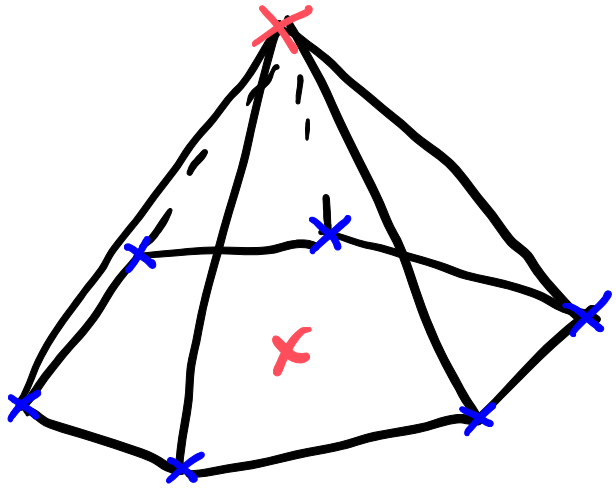
Continuous piecewise affine functions

$$\text{hat}(x) = \rho(x) - 2\rho(x-1) + \rho(x-2) =$$



↝ ReLU NNs in 1d can rep. every continuous p.w. affine function.

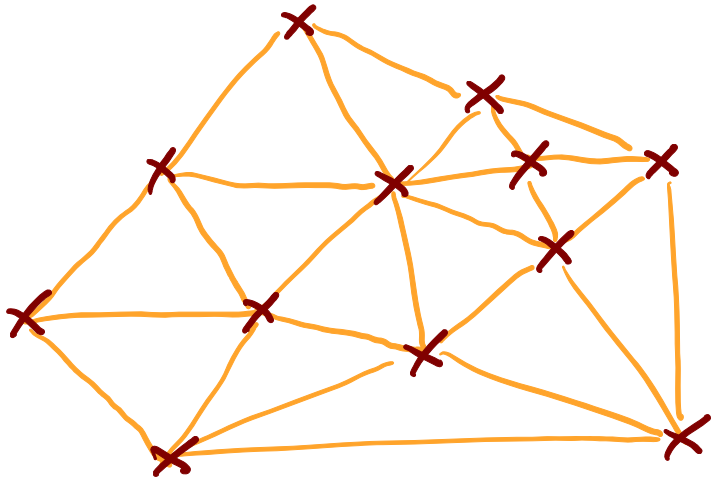
Can we do something similar in high dimensions?



Convex simplicial meshes

Def: Let $d \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$. A set $\mathcal{T} \subset \mathcal{P}(\Omega)$ s.t $\bigcup \mathcal{T} = \Omega$, $\mathcal{T} = (\tau_i)_{i=1}^{M_T}$, where each τ_i is a *d-simplex* and s.t $\tau_i \cap \tau_j \subset \partial\tau_i \cap \partial\tau_j$ is an n-simplex with $n \leq d$ is called a *simplicial mesh*.

We call the τ_i the *cells of the mesh* and the extrem points of the τ_i are called *nodes of the mesh*. We denote the nodes by $(\eta_i)_{i=1}^{M_N}$. We assume that $\eta_i \in \tau_k \Rightarrow \eta_i$ is extreme point of τ_k.



Finite element spaces

- A mesh $\mathcal{T}$ is locally convex, if for all η_i:
  $\bigcup \{\tau_j : \eta_i \in \tau\}$ is convex.
- We call $V_{\mathcal{T}} := \{f \in C(\Omega) : f_{|\tau_i}$ affine for all $i = 1, \dots, M_{\mathcal{T}}\}$,
  linear finite element space.

- $f \in V_{\mathcal{T}}$ is uniquely defined by values on $(\eta_i)_{i=1}^{M_N}$.
- For $i = 1 \dots M_N$, we call $\phi_{i,\mathcal{T}}$ the Courant element if $\phi_{i,\mathcal{T}}(\eta_j) = \delta_{ij}$.

Finite element spaces

Prop: Let $d \in \mathbb{N}$ and $\mathcal{T}$ be a simplicial mesh, then we have that

$$f(x) = \sum_{i=1}^{M_N} f(\eta_i)\, \phi_{i,\mathcal{T}}(x)$$

for all $f \in V_{\mathcal{T}}$

Prop: Let $d \in \mathbb{N}$ and $\mathcal{T}$ be loc. convex. Then

$$\phi_{i,\mathcal{T}} = \max\left\{0, \min_{j \in F_i} g_j\right\},$$

where F_i are all $j : \tau_j \ni \eta_i$ and g_j the affine function vanishing on all nodes in τ_j except η_i.

$$\phi_{i,\mathcal{T}} = \max\Big\{0, \min_{j\in F_i} g_j\Big\} = \rho\Big(\min_{j\in F_i} g_j\Big)$$

affine.

simple with ReLU.

Prop: [He, Li, Xu, Zheng; 2018]

Let $d \in \mathbb{N}$, $\mathcal{T}$ a mesh and $\kappa_{\mathcal{T}}$ be the maximal number of neighbouring cells. Then, for every $i = 1, \dots, M_N$, there ex. a NN ϕ_i with

$$L(\phi_i) = \lceil \log_2(\kappa_{\mathcal{T}}) \rceil + 2 \quad \text{and} \quad M(\phi_i) \le C \cdot (\kappa_{\mathcal{T}} + d)\, \kappa_{\mathcal{T}} \log_2(\kappa_{\mathcal{T}})$$

for a universal $C > 0$, and

$$R(\phi_i) = \phi_{i,\mathcal{T}}.$$

Thm: [He, Li, Xu, Zhen; 2018]

Let $\mathcal{T}$ be a locally convex partition, $d \in \mathbb{N}$. Let $\mathcal{T}$ have M_N nodes. Then, for every $f \in V_T$, there ex. a NN ϕ^f s.t.

$$L(\phi^f) \leq \lceil \log_2(k_T) \rceil + 2$$

$$M(\phi^f) \leq C M_N \cdot (k_T + d) k_T \log_2(k_T)$$

$$R(\phi^f) = f,$$

for a constant $C > 0$.

Remark:

- Every cont. p.w. affine function on an appropriate mesh with N degrees of freedom can be represented by a NN with $C \cdot N$ degrees of freedom.
- NNs can approx. as well as linear FEM.
- E.g.

$$\inf_{g \in V_{T_n}} \|f - g\|_{L^2(\Omega)} \leq n^{-\frac{2}{d}} \|f\|_{W^{2,2d/d+2}}$$

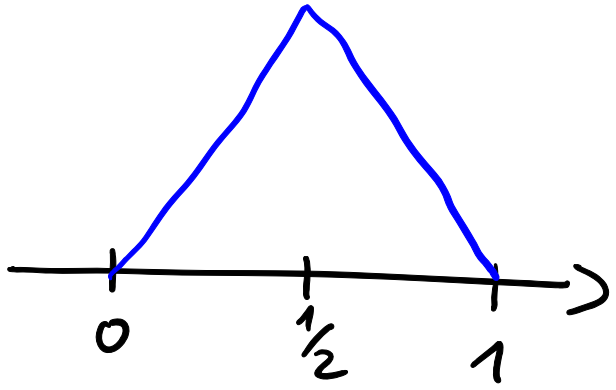
holds for appropriate meshes with $\# T_n \leq n$.

High-order approximation

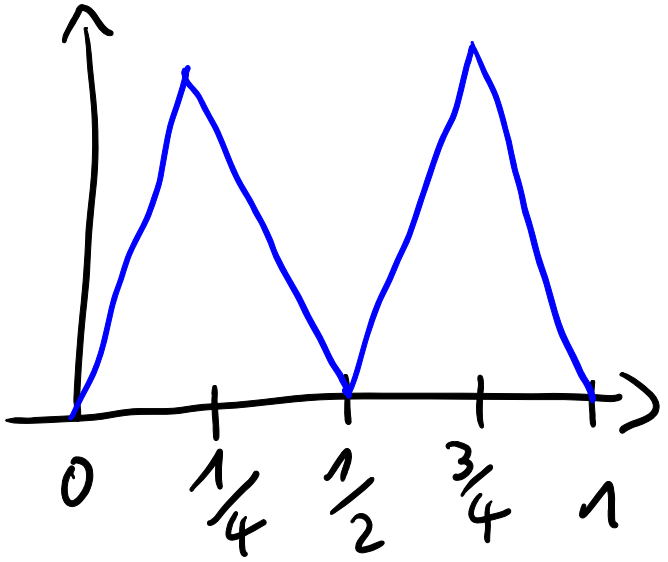
- Recall the construction of a 1d hat function:

$$\text{hat}(x) = \rho(2x) - 2\rho(2x-1) + \rho(2x-2)$$

$$= R(\phi^{\wedge})(x)$$



- $R(\hat{\phi} \bullet \hat{\phi}) = \text{hat} \circ \text{hat}$.



High-order approximation

Prop.: For $n \in \mathbb{N}$:

$$F_n = R\left(\underbrace{\phi^{\wedge} \bullet \dots \bullet \phi^{\wedge}}_{n\text{-times}}\right) = R\left(\phi_n^{\wedge}\right)$$

satisfies for $x \in (0,1)$

$$F_n(x) := \begin{cases} 2^n (x - i2^{-n}) & \text{for } x \in [i2^{-n}, (i+1)2^{-n}] \text{ i even,} \\ 2^n ((i+1)2^{-n} - x) & \text{for } x \in [i2^{-n}, (i+1)2^{-n}] \text{ i odd.} \end{cases}$$

Moreover $F_n = 0$ for $x \notin (0,1)$. Additionally,

$$L(\phi_n^{\wedge}) = n+1, \quad M(\phi_n) \leq 12n - 2.$$

Remark:

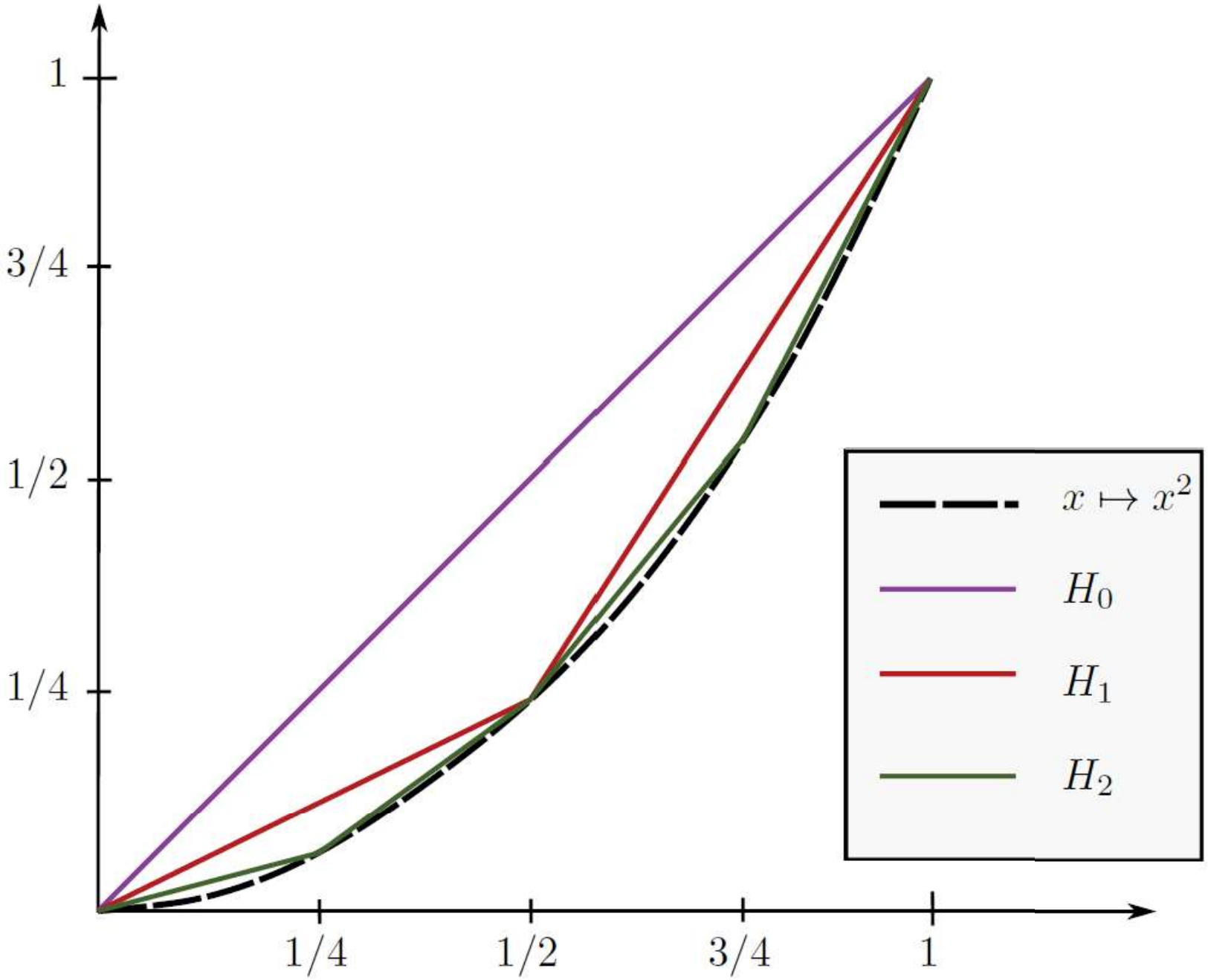
- F_n has 2^n affine pieces on $[0,1]$, while Φ_n only has $O(n)$ weights.
- Note that the realisation of a NN with two layers, $d=1$, is of the form $x \mapsto \sum_{j=1}^{N} c_j \rho(a_j x + b_j) + d$, and cannot have more than N pieces.
- This was first observed in [Telgarsky, 2015].

High dimensional approximation

Prop. [Yarotsky; 2017]

For every $x \in [0,1]$ and $N \in \mathbb{N}$, we have that

$$\left| x^2 - x + \sum_{n=1}^{N} \frac{F_n(x)}{2^{2n}} \right| \leq 2^{-2N-2}.$$


1
3/4
1/2
1/4
1/4
1/2
3/4
1
$x \mapsto x^2$
H_0
H_1
H_2

How to build $x - \sum_{n=1}^{N} \frac{F_n(x)}{2^{2n}}$?

We know how to build, $x, (F_n)_{n=1}^{N}$, but all NNs have different depths.

Def: $\Phi^{id} := ((A_1, b_1), (A_2, b_2))$, where

$$A_1 = \begin{pmatrix} I_d \\ -I_d \end{pmatrix}, \; b_1 = 0, \quad A_2 = (I_d, -I_d), \; b_2 = 0.$$

It holds that $R(\Phi^{id}) = id$.

Similarly:

$$\Phi_L^{id} = \left(\left(\begin{pmatrix} I_d \\ -I_d \end{pmatrix}, 0 \right), (I_d, 0), \dots (I_d, 0), \left((I_d, -I_d), 0 \right) \right).$$

Using deep identities, we can deepen networks without changing their output.

The Φ^{id} is also important in the following definition:

Def: Let $L_1, L_2 \in \mathbb{N}$ and Φ_1, Φ_2 be two NNs s.th. the input dim of Φ_1 equals the output dim of Φ_2. We define the sparse concatenation

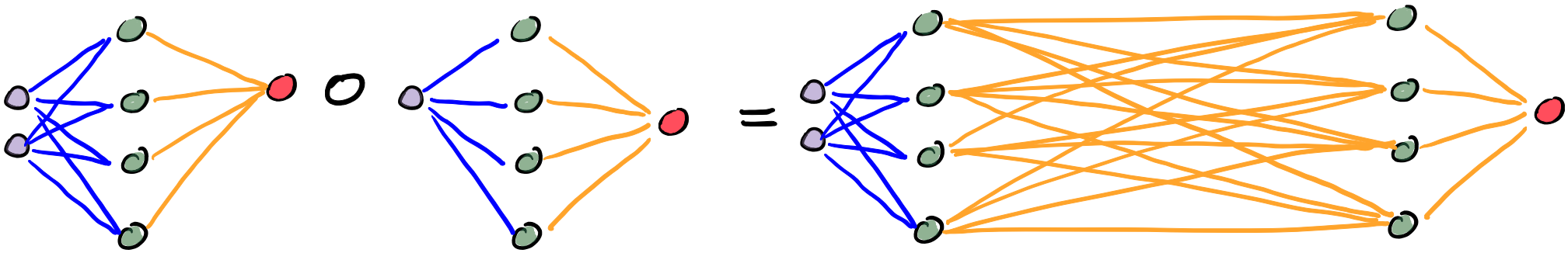
$$\Phi_1 \odot \Phi_2 = \Phi_1 \bullet \Phi^{id} \bullet \Phi_2.$$

It holds that $L(\Phi_1 \odot \Phi_2) = L(\Phi_1) + L(\Phi_2)$,

$R(\Phi_1 \odot \Phi_2) = R(\Phi_1) \circ R(\Phi_2)$ and

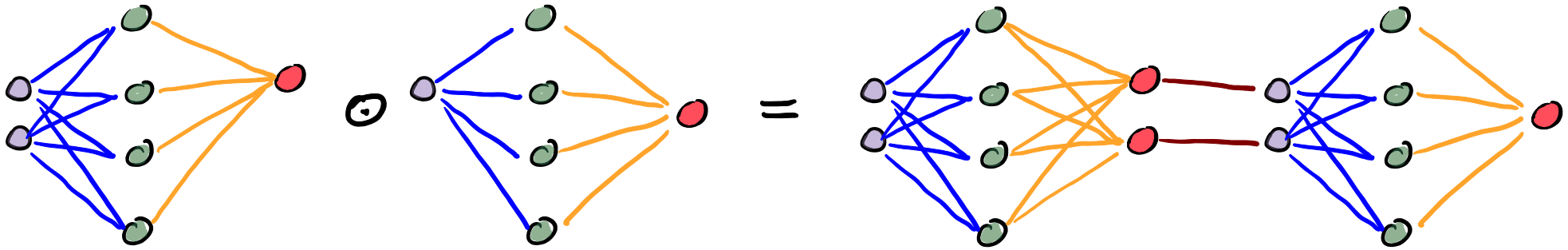
$M(\Phi_1 \odot \Phi_2) \leq 2M(\Phi_1) + 2 \cdot M(\Phi_2)$.

Normal concatenation:



Worst case $M(\Phi_1 \bullet \Phi_2) \sim M(\Phi_1) \cdot M(\Phi_2)$

Sparse concatenation



$$x - \sum_{n=1}^{N} \frac{F_n(x)}{2^{2n}} \longrightarrow \phi_{N-n}^{Id} \odot \Phi_n .$$

$$\Phi_{N+1}^{Id}$$

$$\Rightarrow \quad \phi^{sq,N} := \left(\begin{pmatrix} 1 \\ -2^{-2} \\ -2^{-4} \\ \vdots \end{pmatrix}, 0 \right) \bullet P\left(\Phi_{N+1}^{Id}, \phi_{N-1}^{Id} \odot \Phi_1, \dots \right)$$

satisfies

$$|x^2 - R(\phi^{sq,n})| \leq 2^{-2N-2},$$

$$M(\phi^{sq,N}) \sim N^2$$

$$L(\phi^{sq,N}) \sim N.$$

Alternative multiplication using bit extraction

$$\mathbb{1}_{x \geq \frac{1}{2}} \circ F_n = \begin{cases} 1 & \text{if } x \in [2^{-n-1} j,\ 2^{-n-1}(j+1)] \text{ for } j \text{ odd} \\ 0 & \text{else} \end{cases}$$

$\Rightarrow$ $\mathbb{1}_{x \geq \frac{1}{2}} \circ F_n \left(\cdot - 2^{-n-1} \right)$ extracts n'th bit of binary rep of x.

Also $$xy = \sum_{i=1}^{\infty} x_i\, y\, 2^{-i} = \sum_{i=1}^{\log_2(1/\varepsilon)} x_i\, y_i\, 2^{-i} + \mathcal{O}(\varepsilon)$$

Note $x_i y = \rho(y - x_i)$ if $x_i \in [0,1]$, $y \in [0,1]$.

$\mathbb{1}_{x \geq \frac{1}{2}}$ can be approx. by ReLU.

Thm: [Yarotsky, 2017]

Let $1/2 > \varepsilon > 0$. There ex. a NN $\phi^{sq,\varepsilon}$ s.t.

$$L(\phi^{sq,\varepsilon}) = \mathcal{O}(\log_2(1/\varepsilon)),$$
$$M(\phi^{sq,\varepsilon}) = \mathcal{O}(\log_2^2(1/\varepsilon)),$$
$$|R(\phi^{sq,\varepsilon})(x) - x^2| \leq \varepsilon, \quad \text{for all } x \in [0,1].$$

From x^2 to products

Prop 3.16: Let $p \in \mathbb{N}$, $K \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$. There ex. a NN $\Phi^{\text{mult}, p, \varepsilon}$ s. th.

$$L(\Phi^{\text{mult}, p, \varepsilon}) = \mathcal{O}\left(\log_2(K) + \log_2(1/\varepsilon)\right),$$

$$M(\Phi^{\text{mult}, p, \varepsilon}) = \mathcal{O}\left(\log_2^2(K) + \log_2^2(1/\varepsilon)\right),$$

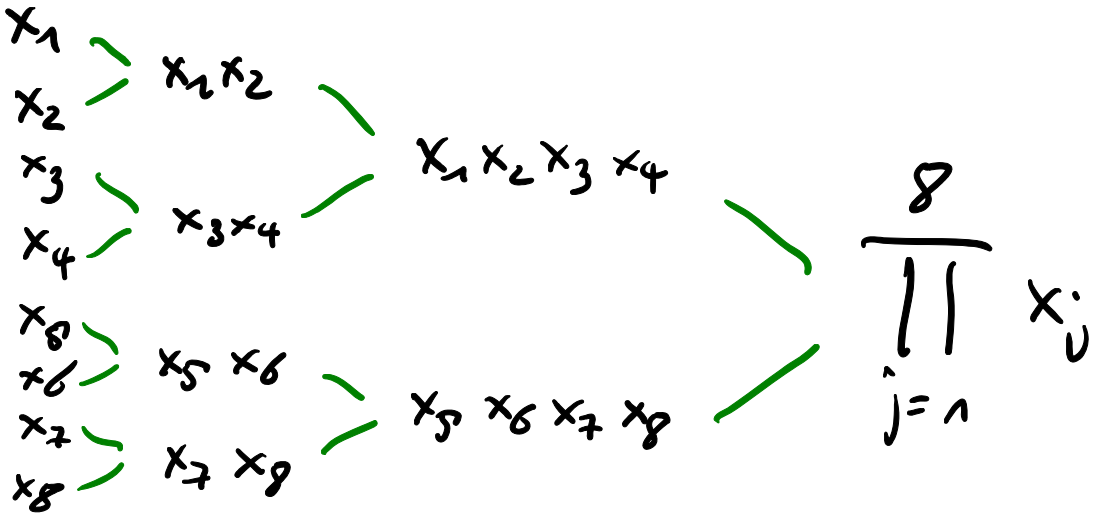
$$\left| R(\Phi^{\text{mult}, p, \varepsilon})(x) - \prod_{\ell=1}^{p} x_\ell \right| \leq \varepsilon,$$

for $x = (x_1, \ldots, x_p) \in [-K, K]^p$.

Proof:

$$xy = K^2\left(\left(\frac{x+y}{2K}\right)^2 - \left(\frac{x-y}{2K}\right)^2\right)$$

$$= K^2\left(\left(\frac{\rho(x+y)}{2K} + \frac{\rho(-x-y)}{2K}\right)^2 - \left(\frac{\rho(x-y)}{2K} - \frac{\rho(y-x)}{2K}\right)^2\right).$$



Recall the B-spline

$$N_k(x) = \frac{1}{(k-1)!} \sum_{\ell=0}^{k} (-1)^\ell \binom{k}{\ell} (x-\ell)_+^{k-1}.$$

$$(\rho(x-\ell))^{k-1}.$$

A NN with $\sim \log_2(1/\varepsilon)$ layers and $\log_2^2(1/\varepsilon)$ weights approximates this to an error of ε.

Smooth functions

Theorem 3.19 [Yarotsky, 2017]

Let $d \in \mathbb{N}$, $s > \delta > 0$ and $p \in (0, \infty]$. Then there exists a constant $C > 0$ such that, for every $f \in C^s([0,1]^d)$ with $\|f\|_{C^s} \leq 1$ and every $\frac{1}{2} > \varepsilon > 0$, there ex. a NN Φ s.t.

$$L(\Phi) \leq C \log_2(1/\varepsilon),$$
$$M(\Phi) \leq C \varepsilon^{-d/s} \log_2^2(1/\varepsilon)$$

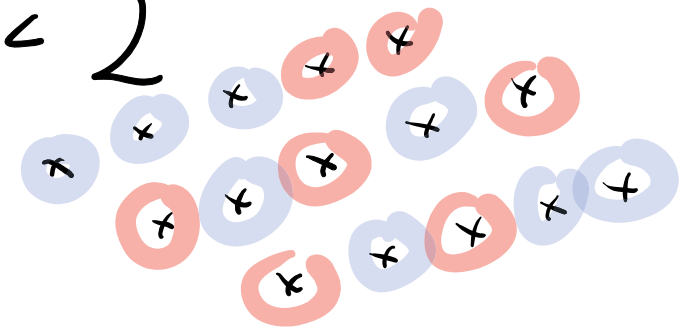
$$\|f - R(\Phi)\|_{L^p} \leq \varepsilon.$$

Lower bounds

Bartlett:

For NNs with L layers, : For all $(x_i)_{i=1}^K$, with $K > M\left(L\log(M)+L^2\right)$

$$\#\left\{\left(\operatorname{sign} R(\phi)\right)\Big|_{(x_i)_{i=1}^K} : M(\phi)\left(\log(M/\phi)L(\phi)+L(\phi)^2\right) < K\right\} < 2^K$$



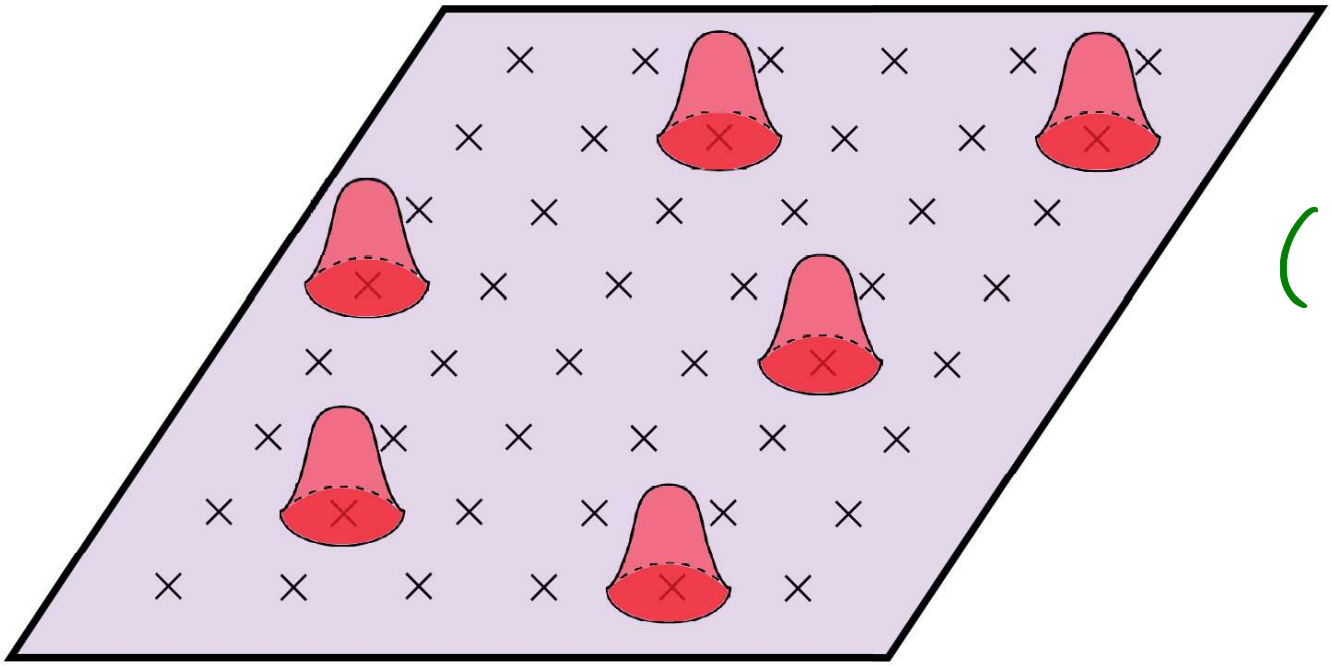
If approx. with M weights to error $M^{-\frac{k+\delta}{d}}$ where possible,

Then, we can find $K > M L \log(M)$ points with distance $\sim K^{-\frac{1}{d}}$. We can associate to each point x_i a bump function ϕ_i :

$\|\phi_i\|_{C^k} \leq 1$, $\quad \phi_i(x_i) = K^{-\frac{k}{d}} > M^{-\frac{k+\delta}{d}}$.

$\operatorname{supp}\phi_i \cap \operatorname{supp}\phi_j = \emptyset$ for $i \neq j$ is possible.

Approximade $\sum_{i=1}^{K} c_i \phi_i$ $\quad c \in \{\pm 1\}^K$



(by constr. $\|\sum c_i \phi_i\|_{C^k} \leq 1$)

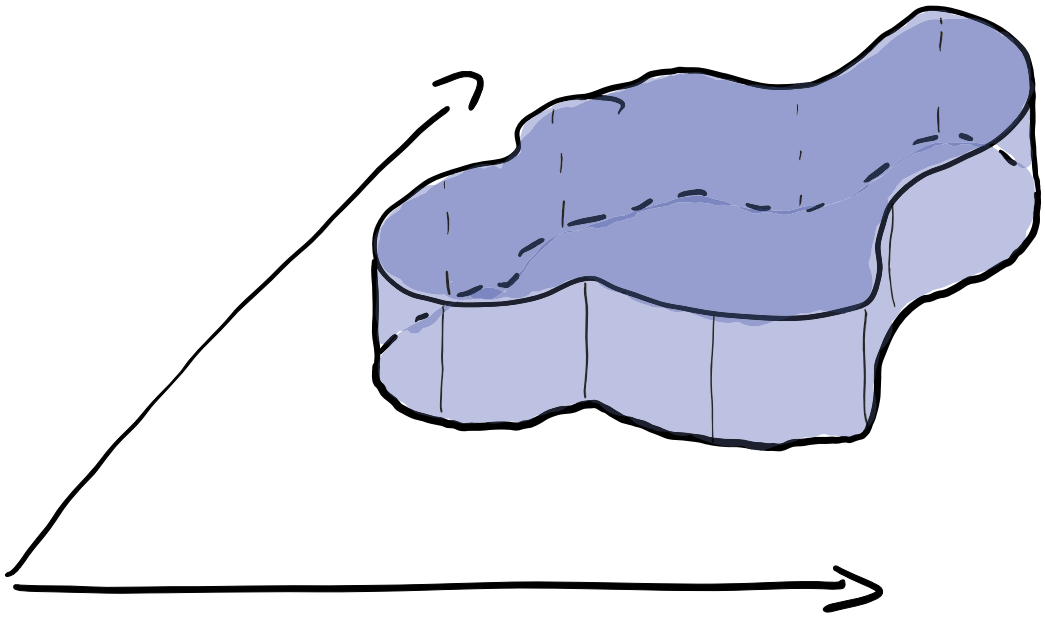
by NN $\Rightarrow$

$$\#\left\{(\operatorname{sign} R(\phi))_{\|(x_i)_{i=1}^K} : M(\phi)\left(L(\phi)\log(M(\phi)) + L(\phi)^2\right) \leq K\right\} \geq 2^K$$

Some of my work, based on these results

Piecewise smooth functions:

$$f = f_1 + \chi_B f_2$$



$f_i : \mathbb{R}^d \supset \Omega \longrightarrow \mathbb{R}$

have prescribed regularity, constant in our case.

$$\chi_B := \begin{cases} 1 & \text{on } B \\ 0 & \text{else} \end{cases}$$

∂B has prescribed regularity

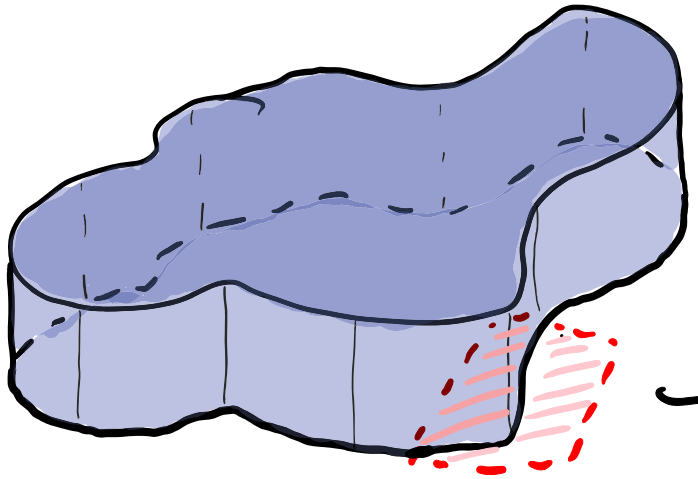
Theorem: (P., Voigtlaender; 2017)

Let $B \subset [0,1]^d$, $f = f_1 + \chi_B f_2$, $f_i \in C^k$, $\partial B \in C^k$. Then, there exists a series of NNs $\Phi_n : \mathbb{R}^d \to \mathbb{R}$, with M_n weights* and $L \in \mathbb{N}$,

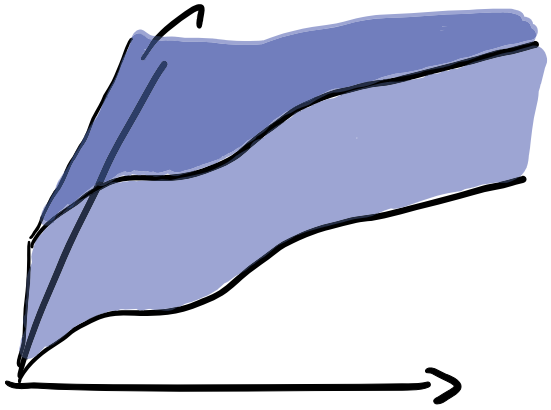
$$\|f - R(\Phi_n)\|_{L^2} \lesssim M_n^{-\frac{k}{2(d-1)}}, \quad L(\Phi_n) \lesssim \frac{k}{d},$$

where $M_n \to \infty$ for $n \to \infty$.

Proof:



a)

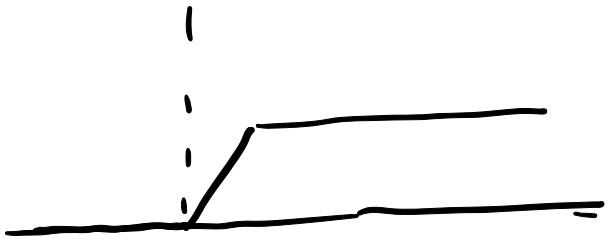


Horizon function

$$\chi_{\{x_1 \le \gamma(x_2,\dots,x_d)\}}$$

$$= H(\gamma(x_2 \dots x_d) - x_1)$$

b)



$$n \cdot \left(\rho(x) - \rho\left(x - \tfrac{1}{n}\right)\right)$$

c)

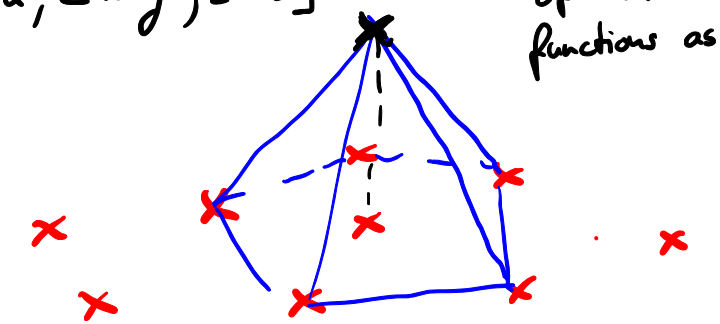
C^k function $\rightsquigarrow$ classical NN approx.*

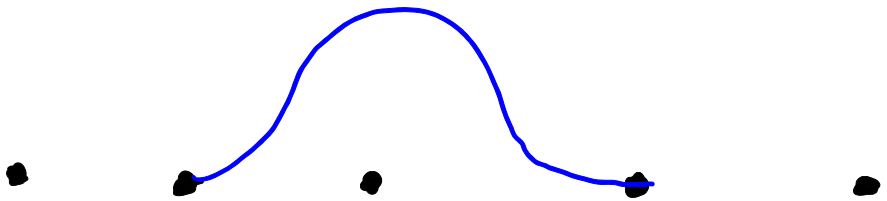
* not really, but almost.

High-order FEM

[He, Li, Xu, Zheng; 2018]

Exact representation of 1st order shape functions as ReLU NNs.



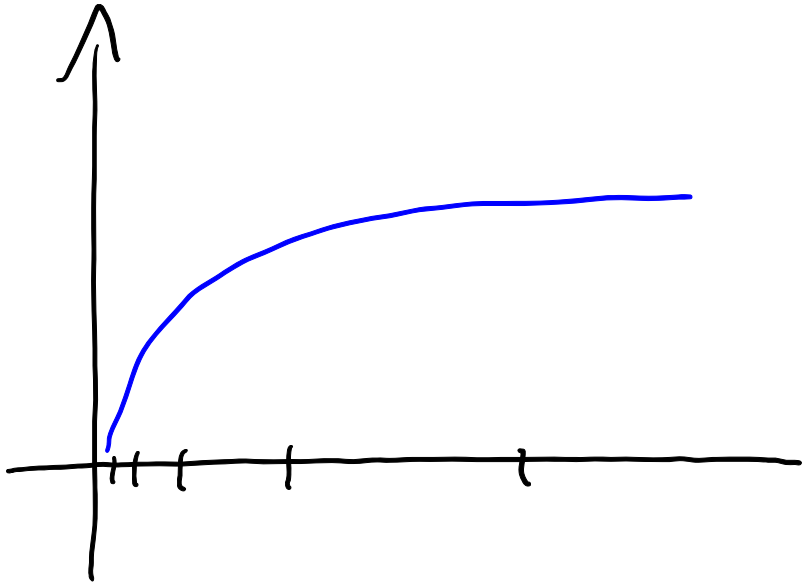


High order elements that are

<u>exact</u> outside of cells.

Exponential accuracy in $H^1([0,1])$ norm.

[Opschoor, P., Schwab; 2020]



Exponential approx. rates for weighted analytic functions.

How did we get H^1?

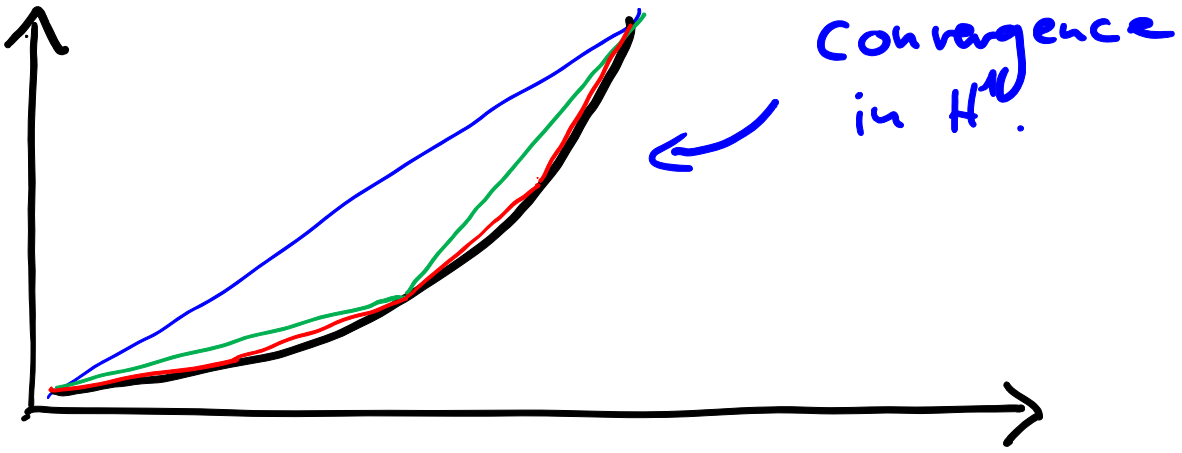
ψ = polynomial vanishing at $\{-1, 1\}$

$\Rightarrow \psi = \phi \cdot (1 - x^2)$

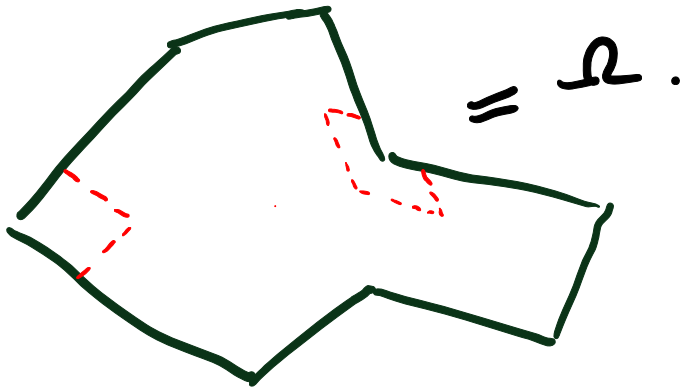
$\tilde{\psi} = \phi \cdot \rho(1 - x^2)$ is p.w. poly. with support in $[-1, 1]$.

can be emulated by ReLU NNs in $W^{1,\infty}$.

[Yarotsky; 2017, Gühring, Kutyniok, P.; 2019]



Example



Let f be an analytic function on $\overline{\Omega}$ ← elliptic, analytic coeffs.

$$\mathcal{L}(u) = f$$

$$\mathcal{B}(u) = 0 \quad \leftarrow \text{ either Dir. or Neumann on faces.}$$

Theorem: (Marcati, Opschoor, P., Schwab; 2020)

For $\varepsilon \in (0,1)$, there exists a NN $\Phi_{\varepsilon,u}$ such that

$$\| \Phi_{\varepsilon,u} - u \|_{H^1(\Omega)} < \varepsilon$$

$$M(\Phi_{\varepsilon,u}) \lesssim |\log_2(\varepsilon)|^5 + 1.$$

Conclusion

- Deep neural networks can reproduce a lot of classical approx. results
- High degree spline approx, if activation function is h.o. sigmoidal.
- linear FEM with ReLUs.
- high degree spline approx, if deep ReLU NNs are used.
- hp-FEM if deep nets are used.

J. Berner, P. Grohs, G. Kutyniok, and P. Petersen. The modern mathematics of deep learning. *arXiv preprint arXiv:2105.04026*, 2021.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White, *Multilayer feedforward networks are universal approximators*, Neural Networks **2** (1989), no. 5, 359–366.

George Cybenko, *Approximation by superpositions of a sigmoidal function*, Mathematics of Control, Signals and Systems **2** (1989), no. 4, 303–314.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken, *Multilayer feedforward networks with a nonpolynomial activation function can approximate any function*, Neural Networks **6** (1993), no. 6, 861–867.

Hrushikesh N Mhaskar, *Neural networks for optimal approximation of smooth and analytic functions*, Neural Computation **8** (1996), no. 1, 164–177.

Vitaly Maiorov and Allan Pinkus, *Lower bounds for approximation by MLP neural networks*, Neurocomputing **25** (1999), no. 1-3, 81–91.

Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng, *ReLU deep neural networks and linear finite elements*, Journal of Computational Mathematics **38** (2020), no. 3, 502–527.

Matus Telgarsky, *Representation benefits of deep feedforward networks*, 2015, arXiv preprint arXiv:1509.08101.

Dmitry Yarotsky, *Error bounds for approximations with deep ReLU networks*, Neural Networks **94** (2017), 103–114.

Philipp Petersen and Felix Voigtlaender, *Optimal approximation of piecewise smooth functions using deep ReLU neural networks*, Neural Networks **108** (2018), 296–330.

Joost Opschoor, Philipp Petersen, and Christoph Schwab, *Deep ReLU networks and high-order finite element methods*, Analysis and Applications (2020), no. 0, 1–56.

Carlo Marcati, Joost Opschoor, Philipp Petersen, and Christoph Schwab, *Exponential ReLU neural network approximation rates for point and edge singularities*, 2020, ETH Zurich SAM Research Report.

Thank you for the attention!