
Felix Voigtländer


Gitta Kutyniok


Morten Nielsen

Approximation with deep networks

Rémi Gribonval - Inria Rennes - Bretagne Atlantique

remi.gribonval@inria.fr

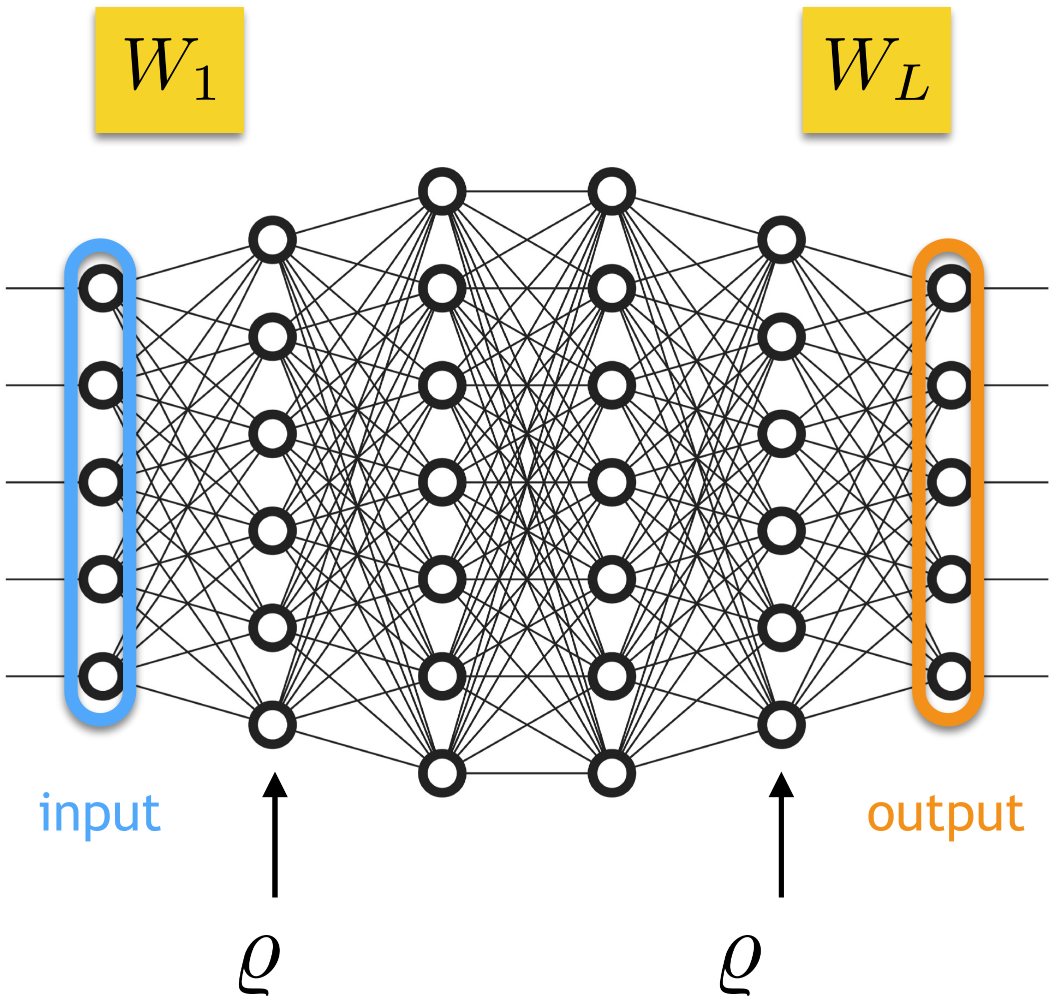
preprint: https://arxiv.org/abs/**1905.01208**

Agenda

- **Generalities on feedforward neural networks**
- Why sparsely connected networks ?
- Approximation spaces
- Benefits of depth

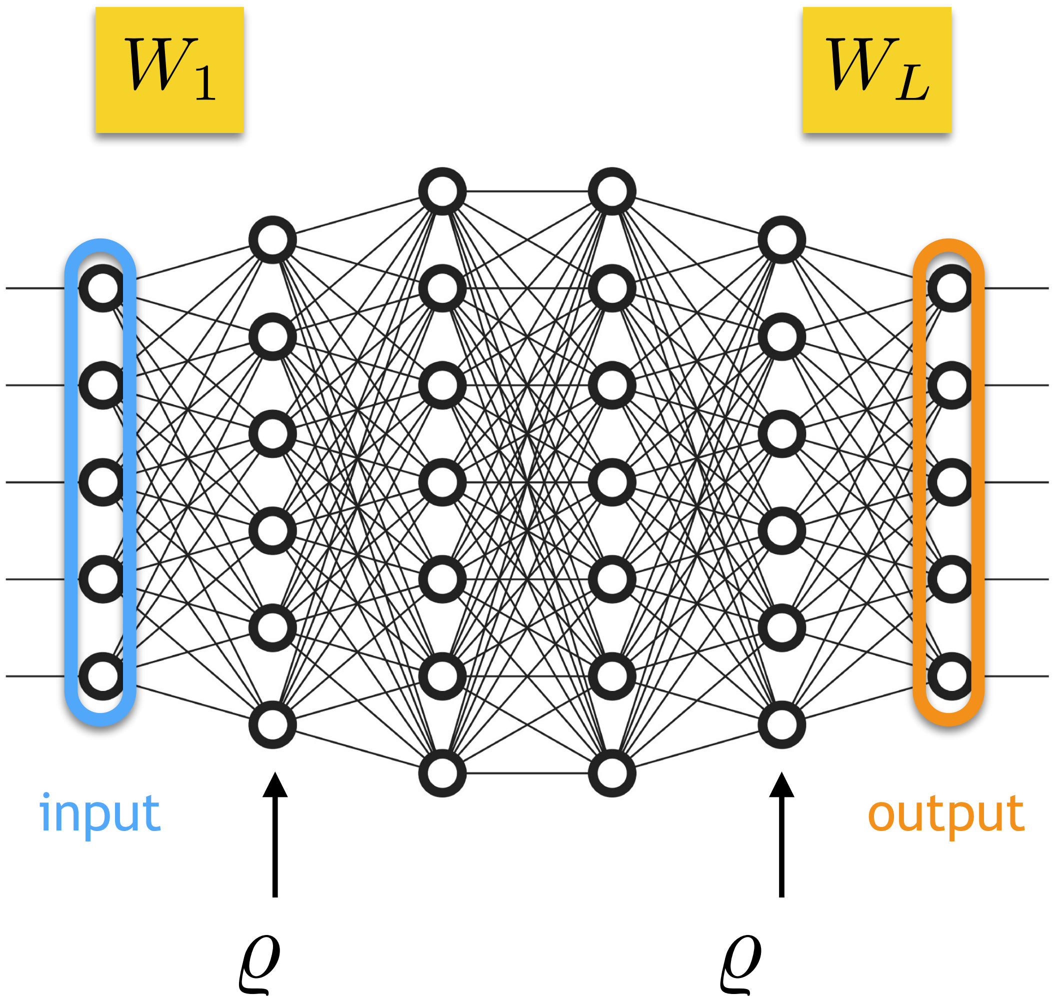
Feedforward neural networks

- **Feedforward network**
  - vector input $x \in \mathbb{R}^d$
  - parameters
    - L affine ("linear") layers W_ℓ
    - L-1 (hidden) nonlinear layers
  - vector output $y \in \mathbb{R}^k$



Feedforward neural networks

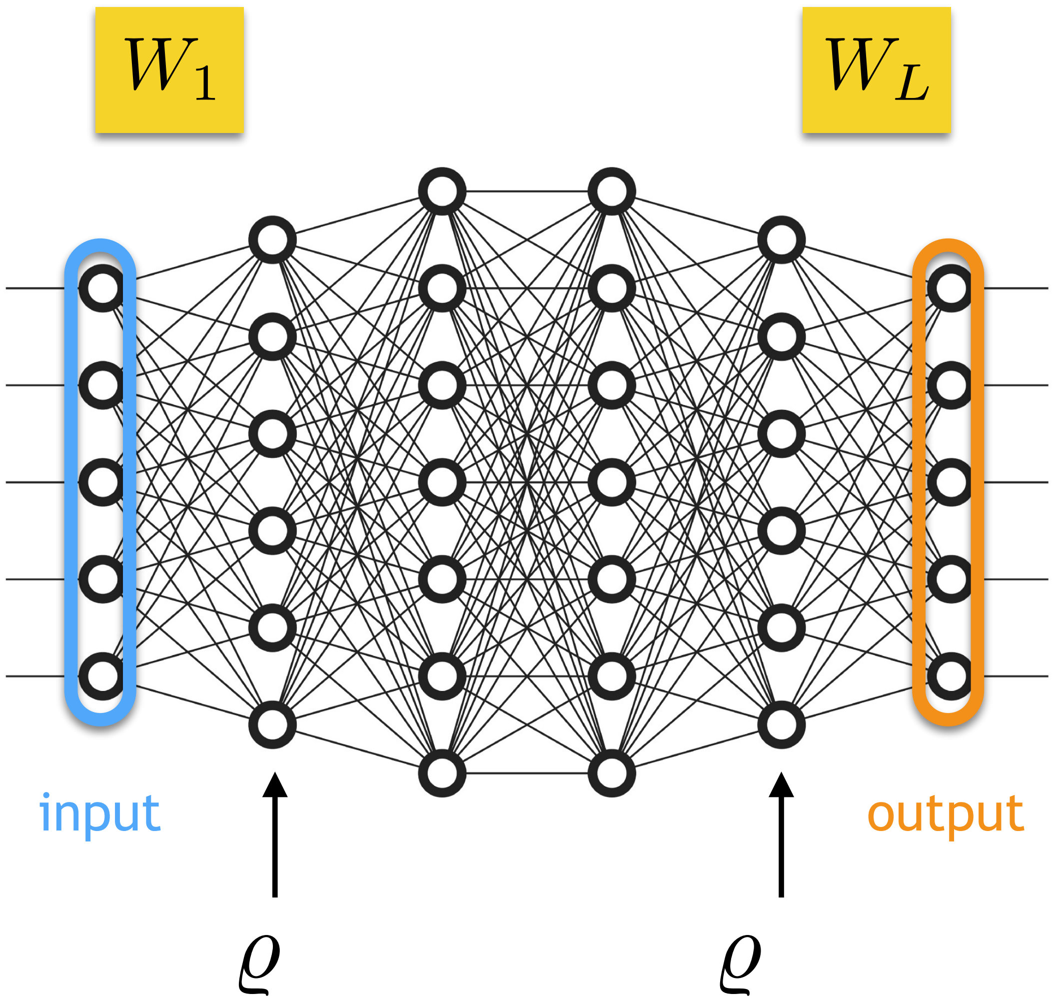
- **Feedforward network**
  - vector input $x \in \mathbb{R}^d$
  - parameters
    - L affine ("linear") layers W_ℓ
    - L-1 (hidden) nonlinear layers
  - vector output $y \in \mathbb{R}^k$
  - description $\theta = (W_\ell)_{\ell=1}^L$
  - ***realization*** $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$



$$f_\theta = W_L \circ \varrho \circ W_{L-1} \circ \cdots \circ \varrho \circ W_1$$

Feedforward neural networks

- **Feedforward network**
  - vector input $x \in \mathbb{R}^d$
  - parameters
    - L affine ("linear") layers W_ℓ
    - L-1 (hidden) nonlinear layers
  - vector output $y \in \mathbb{R}^k$
  - description $\theta = (W_\ell)_{\ell=1}^{L}$
  - ***realization*** $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$

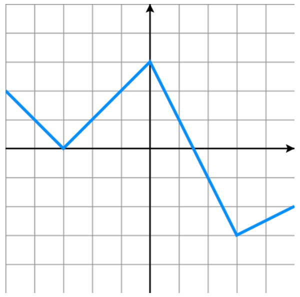


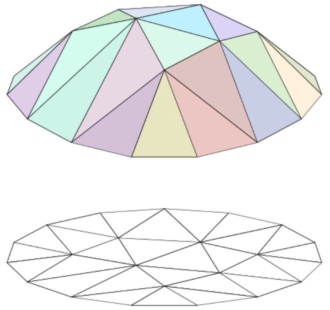
$$f_\theta = W_L \circ \varrho \circ W_{L-1} \circ \cdots \circ \varrho \circ W_1$$

- other ingredients: max-pooling, skip connections, conv … NOT IN THIS TALK

Example: ReLU networks

- **Definition** $\varrho(t) = \mathrm{ReLU}(t) = \max(t, 0) = t_+$
  - popular in practice for computational reasons
- **Properties:**
  - any realization of a ReLU-network is continuous and piecewise (affine) linear





  - **d=1:** *any* piecewise linear function is a realization of a ReLU-network with L=2 (one hidden layer)
  - **d>1:** no longer true (with L=2 layer the realization is not compactly supported)

Studying the expressivity of DNNs

- **DNN training = function fitting**
  - e.g. regression

$$f_{\hat{\theta}}(x) \approx \mathbb{E}(Z|X = x)$$

  - typically stochastic gradient descent: NOT THIS TALK

- **Best achievable approximation ?**
- **Role of “architecture” ?**
  - activation function(s)
  - depth
  - number of neurons, of connections …

Universal approximation property

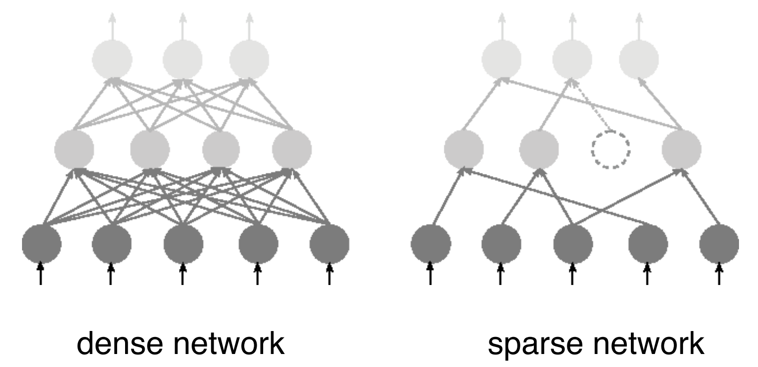
- **A celebrated result**
  - L=2 (*one hidden layer*) is enough to approximate any continuous function arbitrarily well on any compact subset of $\mathbb{R}^d$, with any "sigmoid-like" activation
    - Hornik, Stinchcombe, White 1989; Cybenko 1989
- **Tradeoffs ?**
  - One hidden layer is enough … with large enough #neurons
  - *Approximation rates* wrt #neurons for "smooth" function
    - Barron, DeVore, Mhaskar, and many more since the 1990s

Agenda

- Generalities on feedforward neural networks
- **Why sparsely connected networks ?**
- Approximation spaces
- Benefits of depth

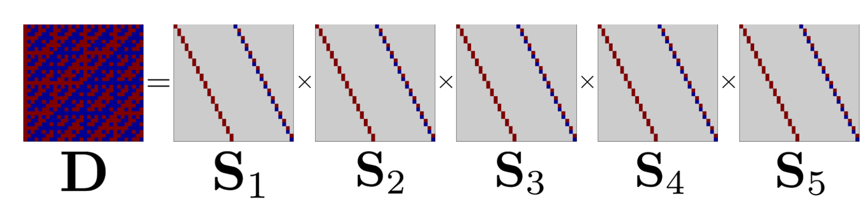
Why sparsely connected networks ?

- **Definition**: sparsity of network with parameters θ
  - $\|\theta\|_0$ = # connections <= n



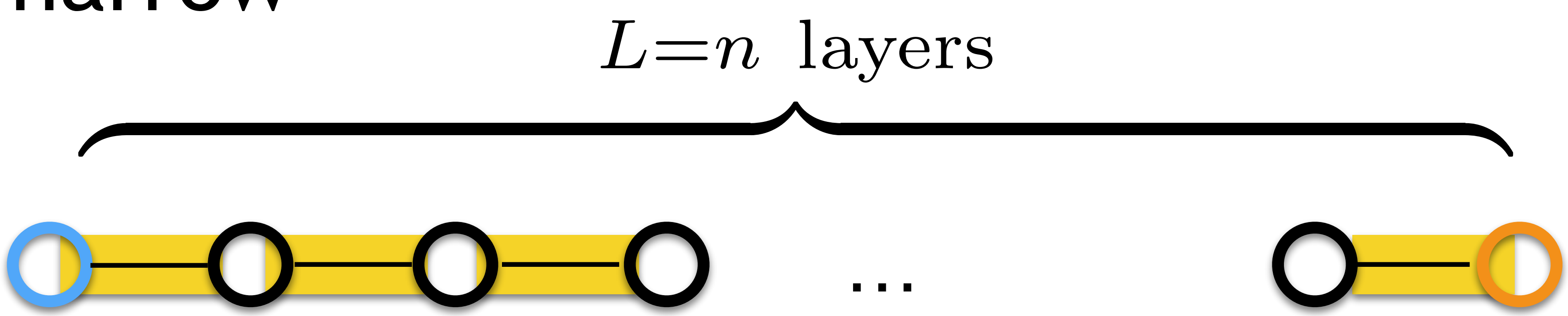
- **Reasonable** proxy to estimate
  - Flops
  - Bits & bytes
  - Sample complexity, e.g. VC dimension
    - see e.g. Bartlett et al 2017

- **Example**: fast linear transforms

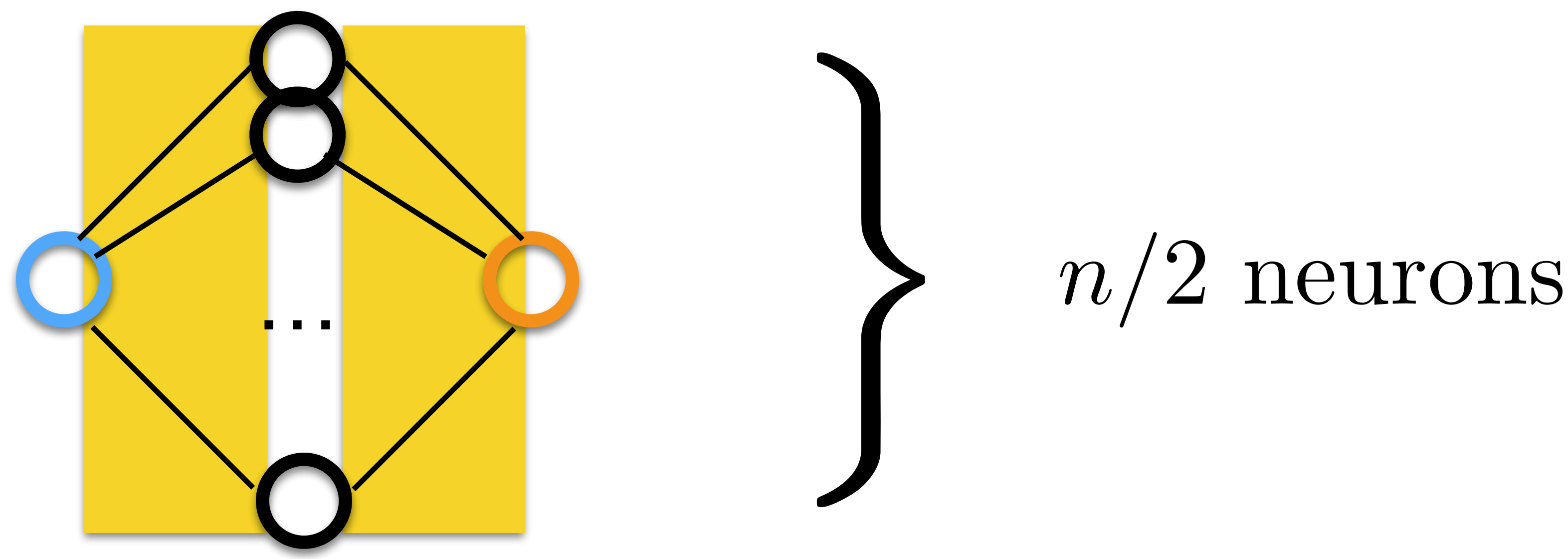


Same sparsity - various network shapes

- Deep & narrow

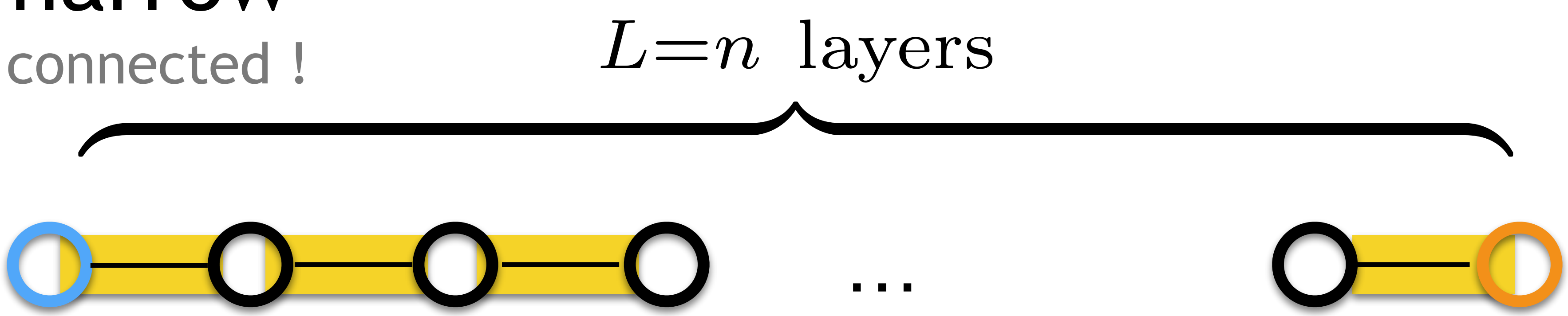


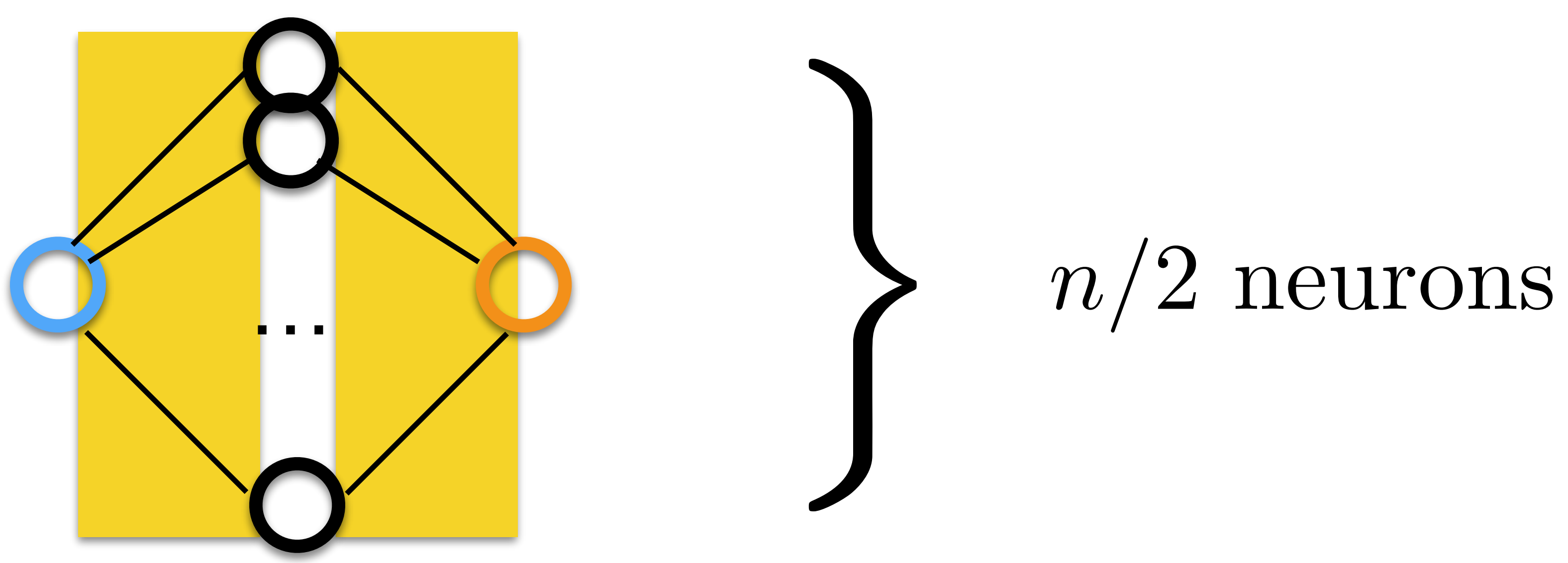
- Shallow & wide



Same sparsity - various network shapes

- Deep & narrow
  - ... fully connected !

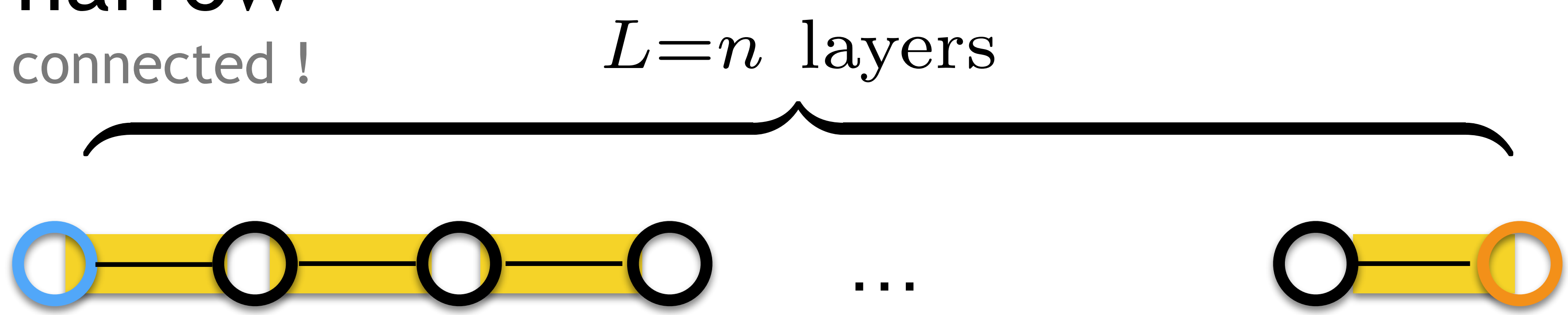




- Shallow & wide
  - ... fully connected !

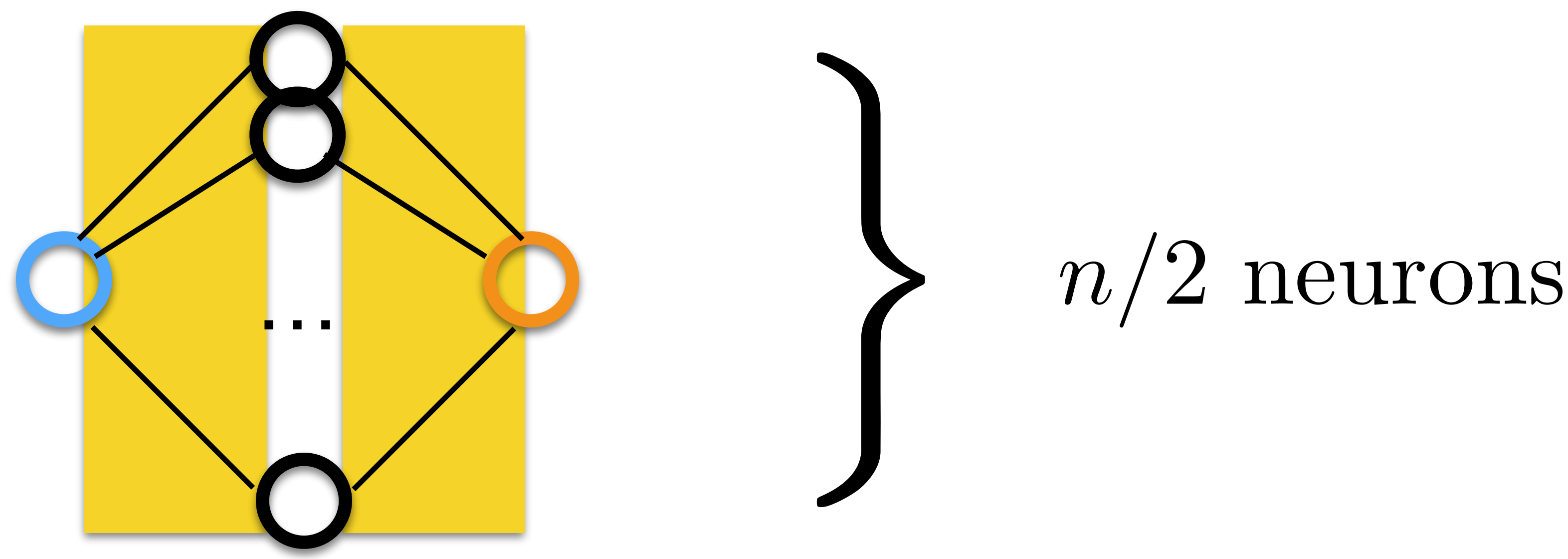
Same sparsity - various network shapes

- Deep & narrow
  - … fully connected !



- … and many more *sparsely* connected possibilities

- Shallow & wide
  - … fully connected !



Approximation with sparse networks

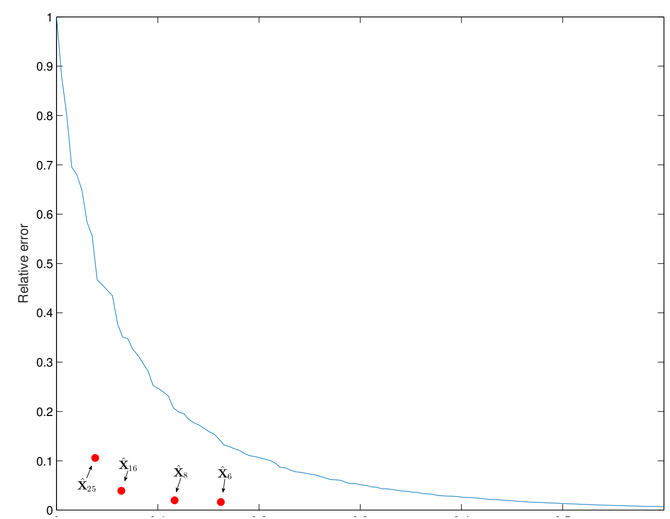
- **Approximation error:** given $\Omega \subset \mathbb{R}^d$ and $f \in L^p(\Omega)$

$$E_n(f) = \inf_{\theta} \|f - f_\theta\|_p$$

  - subject to **sparse connection** constraint $\|\theta\|_0 \leq n$
  - + other constraints (**depth** $L(n)$, choice of ϱ, ...)

- **Tradeoffs error / #connections**

$E_n(f)$



n

example: FAuST (learned fast transforms) vs SVD

Direct *vs* inverse estimate

f is "smooth" (belongs to Sobolev / Besov / modulation space, is "cartoon-like", …)

Direct estimates →

$$E_n(f) \lesssim n^{-\alpha}$$

Direct *vs* inverse estimate

f is "smooth" (belongs to Sobolev / Besov / modulation space, is "cartoon-like", ...)

→ Direct estimates →

$$E_n(f) \lesssim n^{-\alpha}$$

- Optimal rate for these function classes:
  - known (nonlinear width)
  - achieved by deep networks :-)
  - same as wavelets, curvelets
  - cf e.g. work of Philip Grohs and co-workers

Direct *vs* inverse estimate

f is "smooth" (belongs to Sobolev / Besov / modulation space, is "cartoon-like", ...)

Direct estimates →

← Inverse estimates ?

$$E_n(f) \lesssim n^{-\alpha}$$

- Optimal rate for these function classes:
  - known (nonlinear width)
  - achieved by deep networks :-)
  - same as wavelets, curvelets
  - cf e.g. work of Philip Grohs and co-workers

- What can we say about *f* ?
- *Role of activation* ϱ *?*
- *Role of depth ?*

Agenda

- Generalities on feedforward neural networks
- Why sparsely connected networks ?
- **Approximation spaces**
  - Role of skip connections
  - Role of activation function
- Benefits of depth

Notion of approximation space

- **Definition: approximation *class***

$$A^\alpha := \{f \in L^p(\Omega) : E_n(f) = O(n^{-\alpha})\}$$

- *+variants with finer measures of decay*
- *class depends on network "architecture"*
  - *presence of skip-connections*
  - *choice of activation function(s)* ϱ …
  - *fixed or varying depth*

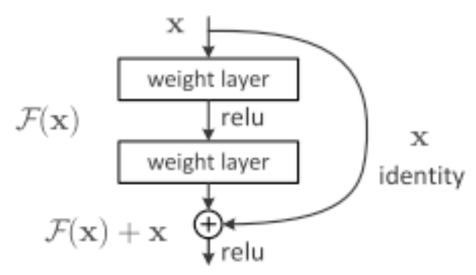
- ***larger class = more expressive architecture***

Role of skip-connections

- **Strict networks**
  - *same* activation at all neurons

$$\varrho$$

  - limitation: cannot implement skip-connections, ResNets, U-nets ?



- **Generalized networks**
  - *two* possible activations at each neuron

$$\varrho \text{ or } \mathtt{id}$$

Role of skip-connections

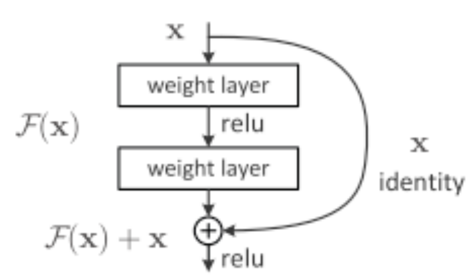
- **Strict networks**
  - *same* activation at all neurons

$$\varrho$$

  - limitation: cannot implement skip-connections, ResNets, U-nets ?

- **Generalized networks**
  - *two* possible activations at each neuron

$$\varrho \text{ or } \mathtt{id}$$



- **Theorem 1:** under some assumptions the class A^α equipped with $\|f\|_{A^\alpha} := \|f\|_p + \sup_n n^\alpha E_n(f)$ is
  - *a complete normed vector space;*
  - *identical for strict & generalized networks*

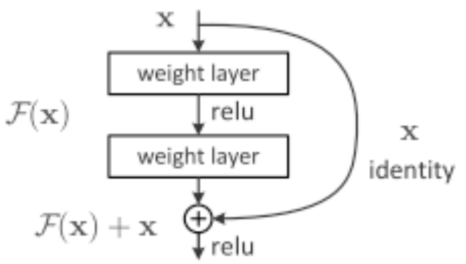
- *assumptions are satisfied by the ReLU and its powers,* $\mathtt{ReLU}^r, r \geq 1$

Role of skip-connections

- **Strict networks**
  - *same* activation at all neurons

$$\varrho$$

  - limitation: cannot implement skip-connections, ResNets, U-nets ?



- **Generalized networks**
  - *two* possible activations at each neuron

ϱ or $\mathtt{id}$

- **Theorem 1: under some assumptions the class A^α equipped with $\|f\|_{A^\alpha} := \|f\|_p + \sup_n n^\alpha E_n(f)$ is**
  - *a complete normed vector space;*
  - *identical for strict & generalized networks* → **Denoted** $A^\alpha(\varrho)$

- *assumptions are satisfied by the ReLU and its powers,* $\mathtt{ReLU}^r, r \geq 1$

Role of skip-connections

- **Strict networks**
  - *same* activation at all neurons

    ϱ

  - lin
    sk
    U-

- **Generalized networks**
  - *two* possible activations at each neuron

    ρ or $\mathtt{id}$

Suggests (TBC) unchanged expressiveness with / without skip-connections (WIP)

- **Theorem 1:** under some assumptions the class A^α equipped with $\|f\|_{A^\alpha} := \|f\|_p + \sup_n n^\alpha E_n(f)$ is
  - *a complete normed vector space;*
  - *identical for strict & generalized networks* → **Denoted** $A^\alpha(\varrho)$

- *assumptions are satisfied by the ReLU and its powers,* $\mathtt{ReLU}^r, r \geq 1$

Role of activation function ϱ

- **(Very) degenerate cases exist**
  - Case of *affine* activation function :
    - A^α = space of all affine transforms
  - Case of *polynomial* activation, with *bounded depth*:
    - A^α = (sub)space of polynomials

Role of activation function ϱ

- **(Very) degenerate cases exist**
  - Case of *affine* activation function :
    - A^α = space of all affine transforms
  - Case of *polynomial* activation, with *bounded depth*:
    - A^α = (sub)space of polynomials
  - There is a (pathological) *analytic* activation such that with L=3 (two hidden layers) and $n = 3d^2(6d+3)$ connections, for any $f \in L^p([0,1]^d), 0 < p < \infty$

$$E_n(f) = 0$$

    - Maiorov & Pinkus 99

Role of activation function ϱ

- **(Very) degenerate cases exist**
  - Case of *affine* activation function :
    - A^α = space of all affine transforms
  - Case of *polynomial* activation, with *bounded depth*:
    - A^α = (sub)space of polynomials
  - There is a (pathological) *analytic* activation such that with L=3 (two hidden layers) and $n = 3d^2(6d+3)$ connections, for any $f \in L^p([0,1]^d), 0 < p < \infty$

$$E_n(f) = 0$$

    - Maiorov & Pinkus 99
    - in other words, approximation space is trivial

$$A^\alpha = L^p([0,1]^d)$$

Piecewise polynomial activation

Theorem 2

- Under mild assumptions on domain and depth growth L(n)
  - If ϱ is continuous and *piecewise polynomial* of degree at most *r*, then $A^\alpha(\varrho) \subset A^\alpha(\mathtt{ReLU}^r)$
  - Moreover, *the expressivity of ReLU powers saturates at r=2*

$$A^\alpha(\mathtt{ReLU}) \subsetneq A^\alpha(\mathtt{ReLU}^2) = A^\alpha(\mathtt{ReLU}^r) \subsetneq L^p, \quad \forall r \geq 2$$

Piecewise polynomial activation

Theorem 2

- Under mild assumptions on domain and depth growth L(n)
  - If ϱ is continuous and *piecewise polynomial* of degree at most *r*, then $A^\alpha(\varrho) \subset A^\alpha(\mathtt{ReLU}^r)$
  - Moreover, *the expressivity of ReLU powers saturates at r=2*

$$A^\alpha(\mathtt{ReLU}) \subsetneq A^\alpha(\mathtt{ReLU}^2) = A^\alpha(\mathtt{ReLU}^r) \subsetneq L^p, \quad \forall r \geq 2$$

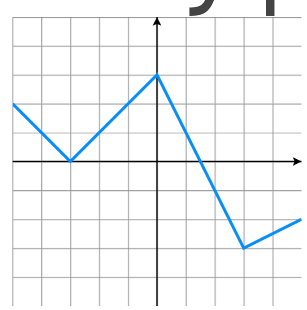
Suggests to explore training squared-ReLU networks ?
Maybe harder to train (vanishing / exploding gradients)

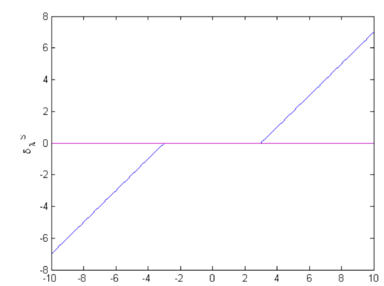
Agenda

- Generalities on feedforward neural networks
- Why sparsely connected networks ?
- Approximation spaces
- **Benefits of depth**

Benefits of depth ?

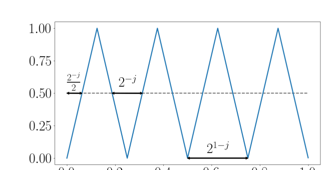
- **ReLU-networks in dimension d=1**
  - Can implement *any* piecewise affine function





    - For L=2 (one hidden layer), #breakpoints = #neurons
    - For large L #breakpoints can be exponential in #neurons

- **Recent work on the benefits of depth**
  - Given #neurons, some functions *implemented* by deep networks are ***badly approximated*** by shallow ones
    - `see e.g. Mhaskar & Poggio 2016, Telgarsky 2016`
    - typical example: “triangular waves” / sawtooth function



“Shallow” ReLU-nets have limited expressivity

- **Theorem 3:**
  - Compactly supported smooth functions approximated at best at rate 2L

    if $\alpha > 2L$ then $C_c^3(\mathbb{R}^d) \cap A^\alpha(\texttt{ReLU}, L) = \{0\}$

  - Cf Theorem 4.5 in: Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. arXiv preprint arXiv:1709.05289, 2017.

- **Corollary:**
  - Consider a function space B such that $C_c^3(\mathbb{R}^d) \cap B \neq \{0\}$

    examples: Sobolev or Besov space, of *arbitrary* positive smoothness

    if $B \subset A^\alpha(\texttt{ReLU}, L)$ then $L > \alpha/2$

“Shallow” ReLU-nets have limited expressivity

- **Theorem 3:**
  - Compactly supported smooth functions approximated at best at rate 2L

    if $\alpha > 2L$ then $C_c^3(\mathbb{R}^d) \cap A^\alpha(\mathtt{ReLU}, L) = \{0\}$
  - Cf Theorem 4.5 in: Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. arXiv preprint arXiv:1709.05289, 2017.

- **Corollary:**
  - Consider a function space B such that $C_c^3(\mathbb{R}^d) \cap B \neq \{0\}$

    examples: Sobolev or Besov space, of *arbitrary* positive smoothness

    if $B \subset A^\alpha(\mathtt{ReLU}, L)$ then $L > \alpha/2$

With ReLU: “If architecture is expressive then it is deep”

Role of depth

Theorem 4

- Direct estimate for Besov spaces

$$B^{\alpha d} \subset A^{\alpha}(\mathtt{ReLU}^r, L)$$

  - for a certain range of rates α

- Inverse estimate for Besov spaces (d=1)

$$A^{\alpha}(\mathtt{ReLU}^r, L) \subset B^{\alpha/\lfloor L/2 \rfloor}$$

  - cannot be improved, for any *d*

Role of depth

Theorem 4

- Direct estimate for Besov spaces

$$B^{\alpha d} \subset A^{\alpha}(\mathtt{ReLU}^r, L)$$

  - for a certain range of rates α

- Inverse estimate for Besov spaces (d=1)

$$A^{\alpha}(\mathtt{ReLU}^r, L) \subset B^{\alpha/\lfloor L/2 \rfloor}$$

  - cannot be improved, for any *d*

Proof sketch

- Direct result
  - Characterize Besov with wavelets
  - Implement n-term wavelet expansion with *O(n)*-sparsely connected network of depth L=3

Role of depth

Theorem 4

- Direct estimate for Besov spaces

$$B^{\alpha d} \subset A^{\alpha}(\texttt{ReLU}^r, L)$$

  - for a certain range of rates α

- Inverse estimate for Besov spaces (d=1)

$$A^{\alpha}(\texttt{ReLU}^r, L) \subset B^{\alpha/\lfloor L/2 \rfloor}$$

  - cannot be improved, for any *d*

Proof sketch

- Direct result
  - Characterize Besov with wavelets
  - Implement n-term wavelet expansion with *O(n)*-sparsely connected network of depth L=3
- Inverse result
  - **Lemma:** if $\|\theta\|_0 \leq n$ then f_θ is piecewise poly with $O(n^{\lfloor L/2 \rfloor})$ pieces
  - Apply Petrushev's inverse estimate for free-knot splines

Role of depth

Theorem 4

- Direct estimate for Besov spaces

$$B^{\alpha d} \subset A^{\alpha}(\mathtt{ReLU}^{r}, L)$$

  - for a certain range of rates α

- Inverse estimate for Besov spaces (d=1)

$$A^{\alpha}(\mathtt{ReLU}^{r}, L) \subset B^{\alpha/\lfloor L/2 \rfloor}$$

  - cannot be improved, for any *d*

Proof sketch

- Direct result
  - Characterize Besov with wavelets
  - Implement n-term wavelet expansion with *O(n)*-sparsely connected network of depth L=3

- Inverse result
  - **Lemma:** if $\|\theta\|_0 \leq n$ then f_θ is piecewise poly with $O(n^{\lfloor L/2 \rfloor})$ pieces
  - Apply Petrushev's inverse estimate for free-knot splines

deeper DNN → expresses *rougher* functions

Summary & perspectives

Summary: Approximation with DNNs

Role of architecture

- **Strict vs generalized networks**: same expressiveness
- **Challenge:** expressiveness of **plain vs skip connections / ResNets?**

➡ *main / only difference =* ***ease of training*** *with stochastic gradient ?*

Summary: Approximation with DNNs

Role of architecture

- **Strict vs generalized networks:** same expressiveness
- **Challenge:** expressiveness of **plain vs skip connections / ResNets?**

➡ *main / only difference = **ease of training** with stochastic gradient ?*

Role of nonlinearity

- $\mathtt{ReLU}(t) = \max(t, 0) = t_+$ as expressive as any piecewise affine activation
- $\mathtt{ReLU}^2$ as expressive as any continuous piecewise polynomial activation
- Expressiveness of $\mathtt{ReLU}^r$ "saturates" at r=2

➡ **Challenge:** training of $\mathtt{ReLU}^2$-networks ? vanishing gradients ?

Summary: Approximation with DNNs

Role of architecture

- **Strict vs generalized networks**: same expressiveness
- **Challenge:** expressiveness of **plain vs skip connections / ResNets?**

➡ *main / only difference = **ease of training** with stochastic gradient ?*

Role of nonlinearity

- $\mathtt{ReLU}(t) = \max(t, 0) = t_+$ as expressive as any piecewise affine activation
- $\mathtt{ReLU}^2$ as expressive as any continuous piecewise polynomial activation
- Expressiveness of $\mathtt{ReLU}^r$ "saturates" at r=2

➡ **Challenge**: training of $\mathtt{ReLU}^2$-networks ? vanishing gradients ?

Role of depth

- Deep enough, any dimension: DNN **strictly more expressive than wavelets**

Summary: Approximation with DNNs

Role of architecture

- **Strict vs generalized networks**: same expressiveness
- **Challenge:** expressiveness of **plain vs skip connections / ResNets?**

➡ *main / only difference = **ease of training** with stochastic gradient ?*

Role of nonlinearity

- $\mathtt{ReLU}(t) = \max(t, 0) = t_+$ as expressive as any piecewise affine activation
- $\mathtt{ReLU}^2$ as expressive as any continuous piecewise polynomial activation
- Expressiveness of $\mathtt{ReLU}^r$ "saturates" at r=2

➡ **Challenge**: training of $\mathtt{ReLU}^2$-networks ? vanishing gradients ?

Role of depth

- Deep enough, any dimension: DNN **strictly more expressive than wavelets**

Last: counting neurons *vs* counting weights:

- can similarly define family of approximation spaces with same properties

$$A^{\alpha}_{\mathtt{weights}}(\varrho) \subset A^{\alpha}_{\mathtt{neurons}}(\varrho) \subset A^{\alpha/2}_{\mathtt{weights}}(\varrho)$$

Overall summary & perspectives

- **First step: expressivity of different architectures**
  - *… spaces yet to be better* ***characterized***
  - ***convolutional architectures****,* ***ResNets****,* ***U-nets****,* ***max-pooling*** *?*

preprint: https://arxiv.org/abs/1905.01208
see also
[Daubechies, DeVore, Foucart, Hanin, Petrova 2019]

- **Next steps ?**
  - *… constructive approximation/training* ***algorithms*** *?*
  - *…* ***guidelines*** *for choosing a DNN architecture ?*
  - *…* ***statistical guarantees*** *?*