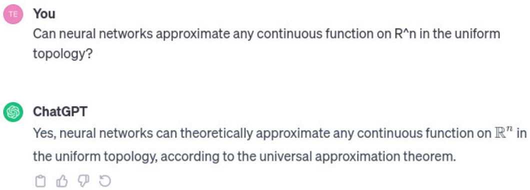
Noncompact uniform universal approximation



Teun van Nuland[*,†]

[*]Email: teunvn@gmail.com
[†]Delft Institute of Applied Mathematics
Delft University of Technology, The Netherlands

23d November 2023
at: McMaster University

Summary of the talk

Theorem [Cybenko 1989, Hornik et al 1989]

Every continuous function can be uniformly approximated by neural networks on a compact subset.

How about on the whole input set?

The answer will give new connections to functional analysis, algebra, and quantum theory. It also gives new insight in neural networks.

Summary of the talk

Theorem [Cybenko 1989, Hornik et al 1989]

Every continuous function can be uniformly approximated by neural networks **on a compact subset**.

How about on the whole input set?

The answer will give new connections to functional analysis, algebra, and quantum theory. It also gives new insight in neural networks.

Summary of the talk

Theorem [Cybenko 1989, Hornik et al 1989]

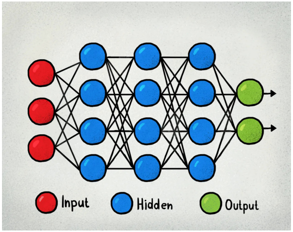
Every continuous function can be uniformly approximated by neural networks **on a compact subset**.

How about on the whole input set?

The answer will give new connections to functional analysis, algebra, and quantum theory. It also gives new insight in neural networks.

What is a (feedforward) neural network?

We fix an activation function $\varphi : \mathbb{R} \to \mathbb{R}$, and an architecture $(n, k_1, \ldots, k_l, k)$ like so:

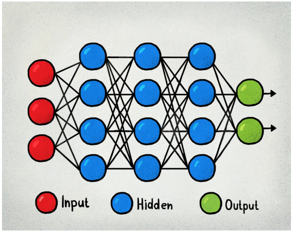


Given affine maps $A^l : \mathbb{R}^{k_l} \to \mathbb{R}^{k_{l+1}}$ each consisting of a $k_{l+1} \times k_l$-matrix a^l of weights and a vector of biases $b^l \in \mathbb{R}^{k_{l+1}}$, the corresponding neural network $f : \mathbb{R}^n \to \mathbb{R}^k$ is

$$f = A^l \circ \varphi^{\otimes k_l} \circ \cdots \circ A^1 \circ \varphi^{\otimes k_1} \circ A^0 .$$

What is a (feedforward) neural network?

We fix an activation function $\varphi : \mathbb{R} \to \mathbb{R}$, and an architecture $(n, k_1, \ldots, k_l, k)$ like so:

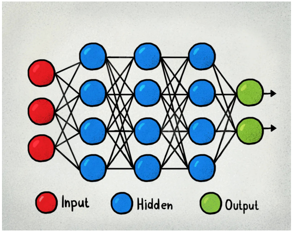


Given affine maps $A^l : \mathbb{R}^{k_l} \to \mathbb{R}^{k_{l+1}}$ each consisting of a k_{l+1}xk_l-matrix a^l of weights and a vector of biases $b^l \in \mathbb{R}^{k_{l+1}}$, the corresponding neural network $f : \mathbb{R}^n \to \mathbb{R}^k$ is

$$f = A^l \circ \varphi^{\otimes k_l} \circ \cdots \circ A^1 \circ \varphi^{\otimes k_1} \circ A^0 .$$

What is a (feedforward) neural network?

We fix an activation function $\varphi : \mathbb{R} \to \mathbb{R}$, and an architecture $(n, k_1, \ldots, k_l, k)$ like so:



Given affine maps $A^l : \mathbb{R}^{k_l} \to \mathbb{R}^{k_{l+1}}$ each consisting of a k_{l+1}xk_l-matrix a^l of weights and a vector of biases $b^l \in \mathbb{R}^{k_{l+1}}$, the corresponding neural network $f : \mathbb{R}^n \to \mathbb{R}^k$ is

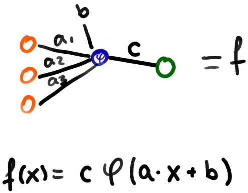
$$f = A^l \circ \varphi^{\otimes k_l} \circ \cdots \circ A^1 \circ \varphi^{\otimes k_1} \circ A^0 \,.$$

What is a (feedforward) neural network?

A 1-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$f(x) = \sum_{j=1}^{k_1} c_j \varphi(a_j \cdot x + b_j)$$

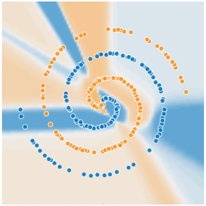
for $a_j \in \mathbb{R}^n$, $b_j, c_j \in \mathbb{R}$.



A 2-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$f(x) = \sum_{j_2} c^2_{j_2} \varphi\Big(\sum_{j_1} c^1_{j_1 j_2} \varphi(a_{j_1} \cdot x + b^1_{j_1}) + b^2_{j_2}\Big)$$

et cetera.



TU Delft

What is a (feedforward) neural network?

A 1-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

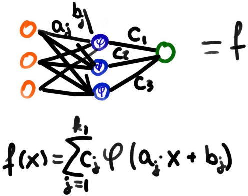
$$f(x) = \sum_{j=1}^{k_1} c_j \varphi(a_j \cdot x + b_j)$$

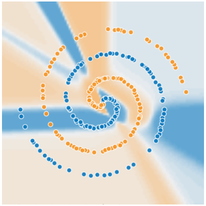
for $a_j \in \mathbb{R}^n$, $b_j, c_j \in \mathbb{R}$.

A 2-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$f(x) = \sum_{j_2} c^2_{j_2} \varphi\Big(\sum_{j_1} c^1_{j_1 j_2} \varphi(a_{j_1} \cdot x + b^1_{j_1}) + b^2_{j_2}\Big)$$

et cetera.





What is a (feedforward) neural network?

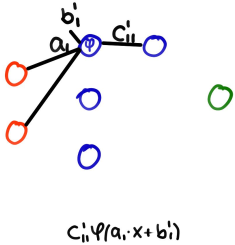
A 1-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

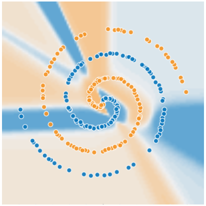
$$f(x) = \sum_{j=1}^{k_1} c_j \varphi(a_j \cdot x + b_j)$$

for $a_j \in \mathbb{R}^n$, $b_j, c_j \in \mathbb{R}$.
A 2-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$f(x) = \sum_{j_2} c^2_{j_2} \varphi(\sum_{j_1} c^1_{j_1 j_2} \varphi(a_{j_1} \cdot x + b^1_{j_1}) + b^2_{j_2}))$$

et cetera.





What is a (feedforward) neural network?

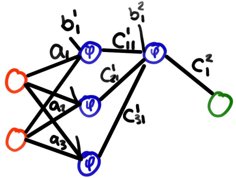
A 1-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$f(x) = \sum_{j=1}^{k_1} c_j \varphi(a_j \cdot x + b_j)$$

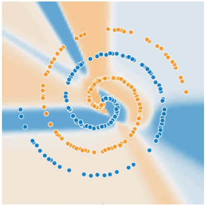
for $a_j \in \mathbb{R}^n$, $b_j, c_j \in \mathbb{R}$.
A 2-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$f(x) = \sum_{j_2} c^2_{j_2} \varphi(\sum_{j_1} c^1_{j_1 j_2} \varphi(a_{j_1} \cdot x + b^1_{j_1}) + b^2_{j_2}))$$

et cetera.



$$\varphi(\sum_{j=1}^{k_1} c^1_{j1} \varphi(a_j \cdot x + b^1_j) + b^2_1)$$



What is a (feedforward) neural network?

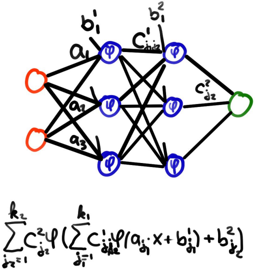
A 1-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

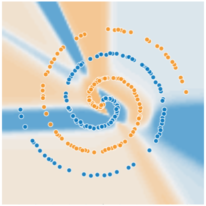
$$f(x) = \sum_{j=1}^{k_1} c_j \varphi(a_j \cdot x + b_j)$$

for $a_j \in \mathbb{R}^n$, $b_j, c_j \in \mathbb{R}$.
A 2-layer neural network $f : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$f(x) = \sum_{j_2} c^2_{j_2} \varphi(\sum_{j_1} c^1_{j_1 j_2} \varphi(a_{j_1} \cdot x + b^1_{j_1}) + b^2_{j_2}))$$

et cetera.





Vector spaces of neural networks

Definition

Let $n \in \mathbb{N}$ and $\varphi : \mathbb{R} \to \mathbb{R}$. The space of 1-layer neural networks with n inputs, 1 output, and activation function φ is

$$\mathcal{N}_\varphi^1(\mathbb{R}^n) := \operatorname{span}\left\{x \mapsto \varphi(a \cdot x + b) \;\middle|\; a \in \mathbb{R}^n,\; b \in \mathbb{R}\right\}. \tag{1}$$

The corresponding space of l-layer neural networks is

$$\mathcal{N}_\varphi^l(\mathbb{R}^n) := \operatorname{span}\left\{x \mapsto \varphi(f(x) + b) \;\middle|\; f \in \mathcal{N}_\varphi^{l-1}(\mathbb{R}^n),\; b \in \mathbb{R}\right\}.$$

A neural network is then any element $f \in \mathcal{N}_\varphi^l(\mathbb{R}^n)^{\oplus k}$.

Universal Approximation

Theorem [Cybenko 1989, Hornik et al 1989, Pinkus 1999, etc]

Let $n, l \in \mathbb{N}$ and $\varphi : \mathbb{R} \to \mathbb{R}$ be continuous and nonpolynomial. Then $\overline{\mathcal{N}^l_\varphi(\mathbb{R}^n)}^{\text{c.c.}} = C(\mathbb{R}^n)$, where closure is taken with respect to the compact convergence topology. In other words,

$$\overline{\mathcal{N}^l_\varphi([0,1]^n)} = C([0,1]^n).$$

Proof is an excellent application of Functional Analysis.
Does not say how functions are approximated in practice, but was and is still highly influential.

The noncompact case: why?

1. It is interesting mathematically. The uniform topology, defined by
$$\|f\|_\infty := \sup_{x\in\mathbb{R}^n} |f(x)|$$
$$f_n \to f \text{ iff } \|f_n - f\|_\infty \to 0$$
is in many ways more natural than the compact convergence topology.
2. After training of the network, one might want consistent results regardless of the size of the input
3. Inputs are often not bounded (salary, speed, costs)
4. Even if they are, they might be big, and $\mathbb{R}^n$ is a good approximation of a big set

You

Can neural networks approximate any continuous function on R^n in the uniform topology?

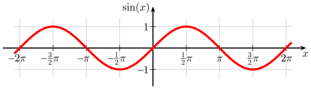
ChatGPT

Yes, neural networks can theoretically approximate any continuous function on $\mathbb{R}^n$ in the uniform topology, according to the universal approximation theorem.

Let's first debunk this...

What can you *not* approximate?

Let $\varphi = \tanh$. $\varphi(\pm\infty) \in \mathbb{R}$. Take $n = 1$.
You will never uniformly approximate sin with neural networks.



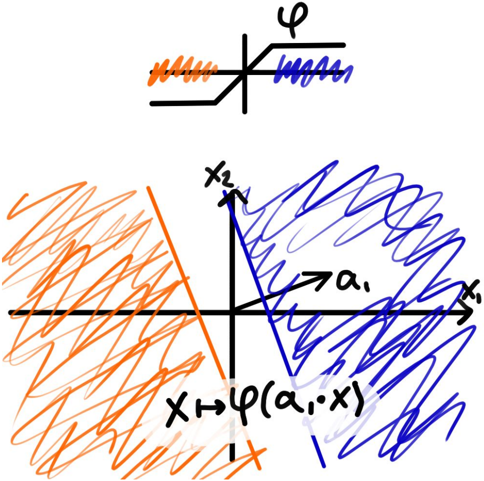
Proof in the case $l = 1$, $n = 1$.

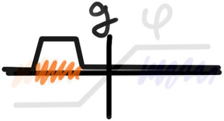
Let $f \in \mathcal{N}_\varphi^1(\mathbb{R})$, and write $f(x) = \sum_{j=1}^k c_j \varphi(a_j x + b_j)$. Then

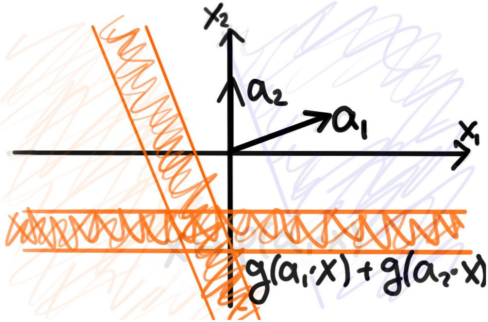
$$\lim_{x\to\infty} f(x) = \sum_{j=1}^k c_j \lim_{x\to\infty} \varphi(a_j x + b_j)$$
$$= \sum_{j=1}^k c_j \varphi(\pm\infty) \in \mathbb{R}$$

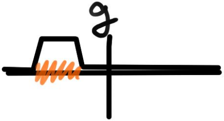
Therefore $\|f - \sin\|_\infty \geqslant \frac{1}{2}$. So $\sin \notin \overline{\mathcal{N}_\varphi^1(\mathbb{R})}$. □

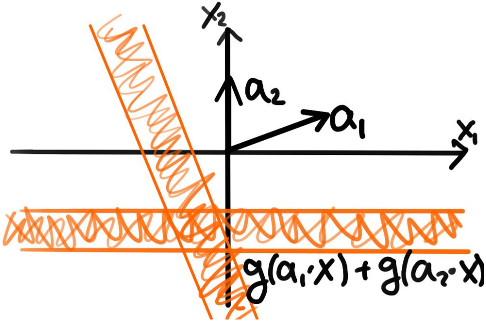
TUDelft

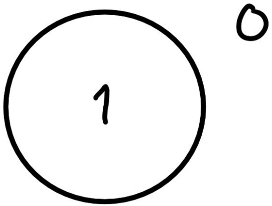

φ
x₂
a₁
x₁
x ↦ φ(a₁·x)

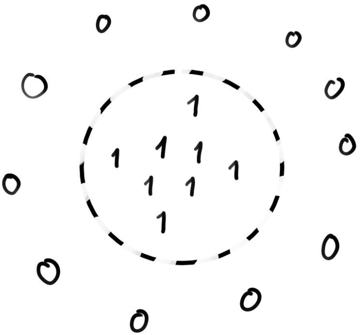

g
φ

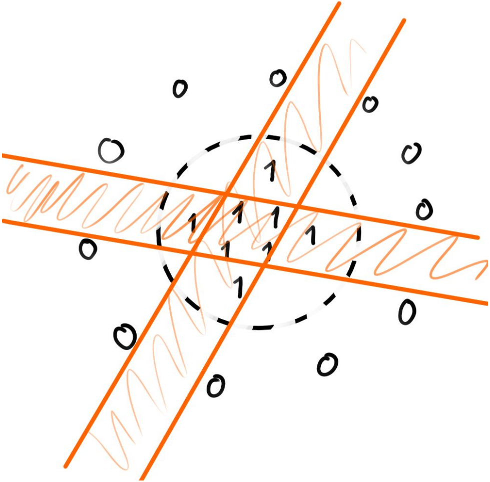

x_2
a_2
a_1
x_1
$g(a_1 \cdot x) + g(a_2 \cdot x)$


g


x2
a2
a1
x1
g(a1·x)+g(a2·x)


1
0


1
1
1
1
1
1
1
1
1
0
0
0
0
0
0
0
0
0
0
0
0
0


0
0
0
0
0
0
0
1
1
1
1
1
1
1
1
0
0
0
0

It is a fundamental question whether all functions in $C_0(\mathbb{R}^n)$ can be approximated by 1-layer neural networks.

Typical universal approximation theorems separate compact regions. They do not guarantee that these regions can themselves be separated from infinity.

In fact **no** 1-layer neural networks are in $C_0(\mathbb{R}^n)$, except 0.

It is a fundamental question whether all functions in $C_0(\mathbb{R}^n)$ can be approximated by 1-layer neural networks.

Typical universal approximation theorems separate compact regions. They do not guarantee that these regions can themselves be separated from infinity.

In fact **no** 1-layer neural networks are in $C_0(\mathbb{R}^n)$, except 0.

We are saved by the following fact:

Theorem [vN,2023]

Let $\varphi \in \Phi$ and let $n \in \mathbb{N}$. Any function in $C_0(\mathbb{R}^n)$ can be uniformly approximated by functions of the form

$$x \mapsto \sum_{j=1}^{k} c_j \varphi(a_j \cdot x + b_j)$$

for some $a_1, \ldots, a_k \in \mathbb{R}^n, b_1, \ldots, b_k, c_1, \ldots, c_k \in \mathbb{R}$. In other words,

$$C_0(\mathbb{R}^n) \subseteq \overline{\mathcal{N}_\varphi^1(\mathbb{R}^n)}.$$

Here Φ includes all nonpolynomial and asymptotically polynomial $\varphi : \mathbb{R} \to \mathbb{R}$ (e.g. ReLU, LReLU, smooth versions of those), step functions, and more.

TUDelft

We are saved by the following fact:

Theorem [vN,2023]

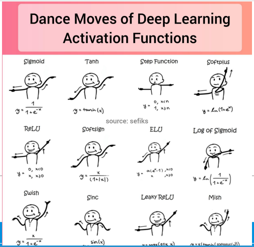
Let $\varphi \in \Phi$ and let $n \in \mathbb{N}$. Any function in $C_0(\mathbb{R}^n)$ can be uniformly approximated by functions of the form

$$x \mapsto \sum_{j=1}^{k} c_j \varphi(a_j \cdot x + b_j)$$

for some $a_1, \ldots, a_k \in \mathbb{R}^n$, $b_1, \ldots, b_k, c_1, \ldots, c_k \in \mathbb{R}$. In other words,

$$C_0(\mathbb{R}^n) \subseteq \overline{\mathcal{N}_\varphi^1(\mathbb{R}^n)}.$$

Here Φ includes all nonpolynomial and asymptotically polynomial $\varphi : \mathbb{R} \to \mathbb{R}$ (e.g. ReLU, LReLU, smooth versions of those), step functions, and more.



We are saved by the following fact:

Theorem [vN,2023]

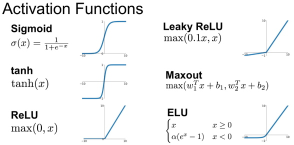
Let $\varphi \in \Phi$ and let $n \in \mathbb{N}$. Any function in $C_0(\mathbb{R}^n)$ can be uniformly approximated by functions of the form

$$x \mapsto \sum_{j=1}^{k} c_j \varphi(a_j \cdot x + b_j)$$

for some $a_1, \ldots, a_k \in \mathbb{R}^n, b_1, \ldots, b_k, c_1, \ldots, c_k \in \mathbb{R}$. In other words,

$$C_0(\mathbb{R}^n) \subseteq \overline{\mathcal{N}^1_\varphi(\mathbb{R}^n)}.$$

Here Φ includes all nonpolynomial and asymptotically polynomial $\varphi : \mathbb{R} \to \mathbb{R}$ (e.g. ReLU, LReLU, smooth versions of those), step functions, and more.



Proof sketch

Although $\mathcal{N}^1_\varphi(\mathbb{R}^n) \cap C_0(\mathbb{R}^n) = \emptyset$, we do have $\overline{\mathcal{N}^1_\varphi(\mathbb{R}^n)} \cap C_0(\mathbb{R}^n) \neq \emptyset$. Proof sketch:

Figure: $f_2(x,y) = \frac{1}{2}\varphi(x) + \frac{1}{2}\varphi(y)$

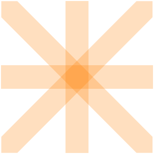


Figure: f_4

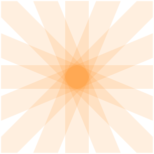


Figure: f_8

If $\varphi = 1_{[-1,1]}$ and $a_j^{(n)} = (\cos\frac{\pi j}{n}, \sin\frac{\pi j}{n})$, then

$$f_n := \sum_{j=1}^{n} \frac{1}{n}\varphi(a_j^{(n)} \cdot x) \to f \in C_0(\mathbb{R}^2).\ \text{[not completely trivial]}$$

As any continuous function on a compact set $K \Subset \mathbb{R}^n$ can be extended to a function in $C_0(\mathbb{R}^n)$, the statement $C_0(\mathbb{R}^n) \subseteq \overline{\mathcal{N}_\varphi^1(\mathbb{R}^n)}$ recovers the usual universal approximation theorem.

If $\varphi \in \Phi$ is continuous,

$$\overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)}^{\text{c.c.}} = C(\mathbb{R}^n)$$

$$C_0(\mathbb{R}^n) \subset \overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)} \subset C(\mathbb{R}^n)$$

As any continuous function on a compact set $K \Subset \mathbb{R}^n$ can be extended to a function in $C_0(\mathbb{R}^n)$, the statement $C_0(\mathbb{R}^n) \subseteq \overline{\mathcal{N}_\varphi^1(\mathbb{R}^n)}$ recovers the usual universal approximation theorem.

If $\varphi \in \Phi$ is continuous,

$$\overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)}^{\text{c.c.}} = C(\mathbb{R}^n)$$

$$C_0(\mathbb{R}^n) \subset \overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)} \subset C(\mathbb{R}^n)$$

The bounded case

If $\varphi \in \Phi$ is continuous and bounded,

$$C_0(\mathbb{R}^n) \subset \overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)} \subset C_b(\mathbb{R}^n)$$

Two cases: $\varphi(-\infty) = \varphi(\infty)$ and $\varphi(-\infty) \neq \varphi(\infty)$ The space $\overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)}$ can be two things, but is otherwise independent from φ and $l \geqslant 2$.

The case $\varphi(-\infty) = \varphi(\infty)$.

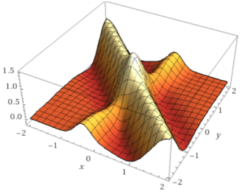
Let us assume $\varphi \in C_0(\mathbb{R})$.

Theorem

Let $\varphi \in C_0(\mathbb{R})$. For all $n, l \in \mathbb{N}$ we have

$$\overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)} = \overline{\text{span}} \left\{ x \mapsto g(P(x)) \;\middle|\; \begin{array}{l} P : \mathbb{R}^n \to \mathbb{R}^k \text{ linear} \\ g \in C_0(\mathbb{R}^k),\ 0 \leqslant k \leqslant n \end{array} \right\}.$$

The right-hand side is known as the commutative resolvent algebra $C_{\mathcal{R}}(\mathbb{R}^n)$, which appears in quantum physics problems. [vN 2019]



Some intuition behind

$$\overline{\mathcal{N}^l_\varphi(\mathbb{R}^n)} =$$

$$C_{\mathcal{R}}(\mathbb{R}^n) := \overline{\text{span}} \left\{ x \mapsto g(P(x)) \;\middle|\; \begin{array}{l} P : \mathbb{R}^n \to \mathbb{R}^k \text{ linear} \\ g \in C_0(\mathbb{R}^k),\ 0 \leqslant k \leqslant n \end{array} \right\} :$$

Note $C_0(\mathbb{R}^n) \subseteq C_{\mathcal{R}}(\mathbb{R}^n)$ and $[x \mapsto \varphi(a \cdot x)] \in C_{\mathcal{R}}(\mathbb{R}^n)$ for all $a \in \mathbb{R}^n$ and $\varphi \in C_0(\mathbb{R})$. Also, multiplying two such functions is again in $C_{\mathcal{R}}(\mathbb{R}^n)$.

This allows us to prove $g \circ (g_1 \circ P_1 + g_2 \circ P_2) \in C_{\mathcal{R}}(\mathbb{R}^n)$ etc, hence, adding layers preserves $C_{\mathcal{R}}(\mathbb{R}^n)$. (Details: approximate g by a polynomial $p_k(x) = a_k x^k + \cdots a_0$ on the range of f and note that $g \circ f = a_k f^k + \cdots + a_1 f + a_0 \in C_{\mathcal{R}}(\mathbb{R}^n)$ for $f \in C_{\mathcal{R}}(\mathbb{R}^n)$.)

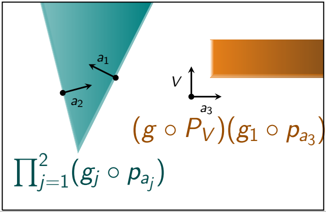
The case $\varphi(-\infty) \neq \varphi(\infty)$

Theorem

Let $\varphi \in C(\mathbb{R})$ be such that the limits $\varphi(-\infty), \varphi(\infty)$ are finite and satisfy $\varphi(-\infty) \neq \varphi(\infty)$. Then for all $n \in \mathbb{N}, l \in \mathbb{N}_{\geqslant 2}$ the space of approximable functions equals

$$\overline{\mathcal{N}_{\varphi}^{l}(\mathbb{R}^n)} = \overline{\text{span}} \left\{ x \mapsto \prod_{j=1}^{m} \tanh(a_j \cdot x) \;\middle|\; m \in \mathbb{Z}_{\geqslant 0}, a_j \in \mathbb{R}^n \right\}.$$

"tanh" can be replaced with any strictly monotonous bounded continuous function.

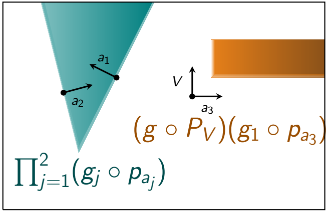


TU Delft

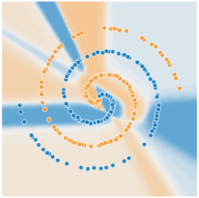
$$\overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)} = \overline{\text{span}}\left\{ x \mapsto \prod_{j=1}^m \tanh(a_j \cdot x) \;\middle|\; m \in \mathbb{Z}_{\geqslant 0}, a_j \in \mathbb{R}^n \right\}$$

can be explained as: neural nets are indistinguishable from sums of 'wedge functions'.

These structures have to appear at large enough scale!



In fact, the scale doesn't have to be too large.



`https://www.matlabsolutions.com/visualize-neural-network/neural-network.html`

New research questions

- In both bounded cases, $\overline{\mathcal{N}_\varphi^l(\mathbb{R}^n)}$ is an algebra. Actually, a commutative unital C*-algebra. C*-algebras were recently used to generalize neural networks [Hashimoto et al. 2022].
- Relation to tropical geometry
- Applications to quantum algebra [Buchholz, vN, 2023]
- What if amount of nodes are restricted? Cf. [Kidger, Lyons, 2020]
- How about convolutional neural networks? Recurrent?

Lots of fun mathematics left to explore here!