Scientific Machine Learning - Theoretical aspects

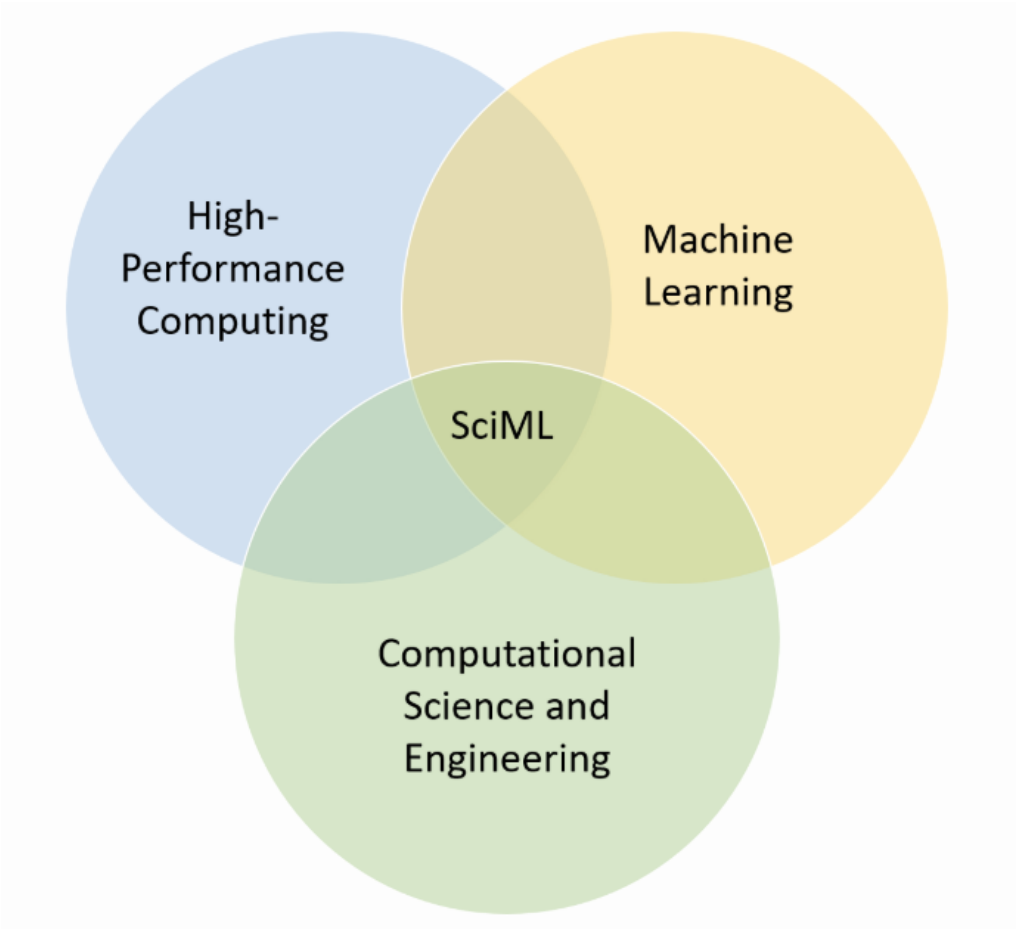
Stéphane Descombes, Stéphane Lanteri

2025-2026

Outline

Outline

Introduction to SciML



In recent years, the combination of numerical methods and machine learning has gained an ever increasing interest as a research field within Numerical Mathematics and Scientific Computing

Introduction to SciML

What is Scientific Machine Learning ?

- SciML is an emerging discipline within the data science community
- SciML seeks to address domain-specific data challenges and extract insights from scientific data sets through innovative methodological solutions
- SciML draws on tools from both Machine Learning (ML) and scientific computing to develop new methods for robust learning and data analysis
- SciML will be critical in driving the next wave of data-driven scientific discovery in the physical and engineering sciences
- SciML is multidisciplinary and leverages expertise from applied and computational mathematics, computer science, and the physical sciences

Introduction to SciML

Why Scientific Machine Learning ?

- New innovations in ML and Big Data are beginning to drive advances in scientific disciplines
- But the full potential of these techniques for data-driven discovery has yet to be fully realized
- One barrier to data-driven discovery is that existing methods often do not meet the needs of scientific users
- Application-agnostic algorithms, or those designed for more traditional ML applications such as image or natural language processing, can not typically be directly applied to scientific data sets and require non-trivial, task-specific modifications

Introduction to SciML

Why Scientific Machine Learning ?

- In many applications only limited or low-quality labels are available, while massive unlabeled (often class imbalanced) data sets are common
- Scientific data are often high-dimensional, noisy, heterogeneous, low-signal-to-noise, and multiscale
- Models should respect or incorporate physical laws, constraints, and other scientific domain knowledge
- Robust methods and an ability to quantify uncertainty are required for scientific rigor

Outline

Poisson equation

- Dirichlet boundary conditions

$$\begin{cases} -u''(x) + c(x)u(x) = f(x),\ x \in]-1,1[, \\ u(-1) = 0,\ u(1) = 0. \end{cases}$$

Exact solution for $c = 0$ and $f(x) = \pi^2 \sin(\pi x)$, $-1 \leq x \leq 1$,

$$u(x) = \sin(\pi x),\ -1 \leq x \leq 1.$$

- Dirichlet/Neumann boundary conditions

$$\begin{cases} -u''(x) + c(x)u(x) = f(x),\ x \in]-1,1[, \\ u(-1) = 0,\ u'(1) = 4. \end{cases}$$

Exact solution for $c = 0$ and $f(x) = -2$, $-1 \leq x \leq 1$,

$$u(x) = (x+1)^2,\ -1 \leq x \leq 1.$$

- Dirichlet/Robin boundary conditions

$$\begin{cases} -u''(x) = f(x),\ x \in]0,1[, \\ u(-1) = 0,\ u'(1) = u(1). \end{cases}$$

Same exact solution for $c = 0$ and $f(x) = -2$, $-1 \leq x \leq 1$.

Heat equation

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) - \dfrac{\partial^2 u}{\partial x^2}(t,x) = 0, x \in]-1,1[,\ t > 0, \\ u(t,-1) = u(t,1) = 0, t > 0, \\ u(0,x) = u_0(x),\ x \in]-1,1[. \end{cases}$$

u_0 is the initial condition

Advection equation

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) + a\dfrac{\partial u}{\partial x}(t,x) = 0,\ (t,x) \in \mathbb{R}^+ \times \mathbb{R} \\ u(0,x) = u_0(x),\ x \in \mathbb{R}, \end{cases}$$

u_0 is the initial condition, a is a constant, $a > 0$.

- If u_0 belongs to $C^1(\mathbb{R})$, existence and uniqueness of a classical solution u belonging to $C^1(\mathbb{R}^+ \times \mathbb{R})$

$$u(t,x) = u_0(x - at).$$

Inviscid Burgers equation / Viscid Burgers equation

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) + u(t,x)\dfrac{\partial u}{\partial x}(t,x) = 0,\ (t,x) \in \mathbb{R}^+ \times \mathbb{R} \\ u(0,x) = u_0(x),\ x \in \mathbb{R}, \end{cases}$$

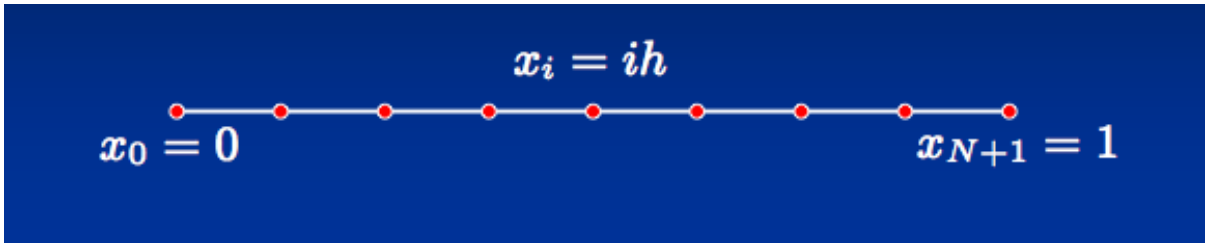
u_0 is the initial condition.

- If u_0 belongs to $C^1(\mathbb{R})$, bounded over $\mathbb{R}$ with a derivative bounded over $\mathbb{R}$, if $u_0' \geq 0$: existence and uniqueness of a classical solution u belonging to $C^1(\mathbb{R}^+ \times \mathbb{R})$.

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) + u(t,x)\dfrac{\partial u}{\partial x}(t,x) = \nu \dfrac{\partial^2 u}{\partial x^2}(t,x),\ (t,x) \in \mathbb{R}^+ \times \mathbb{R} \\ u(0,x) = u_0(x),\ x \in \mathbb{R}, \end{cases}$$

u_0 is the initial condition, ν is a constant, $\nu > 0$.
Hopf-Cole transformation, transform the viscid Burgers equation into a linear heat equation.

Forward Euler scheme

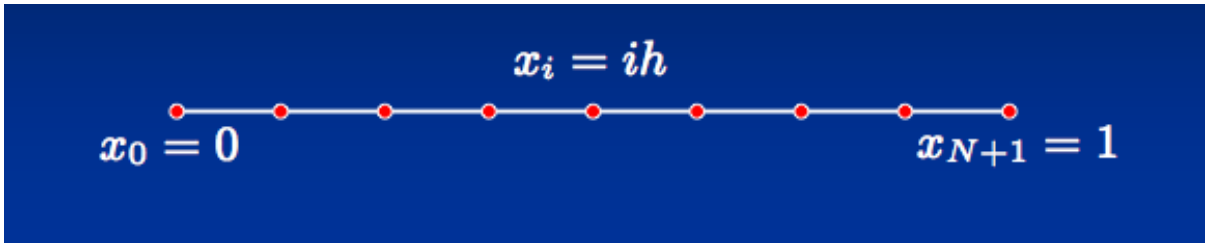


Forward Euler scheme

1

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} = 0,\ i = 1, \cdots, N,\ n = 0, \cdots, M-1,$$

Forward Euler scheme



Forward Euler scheme

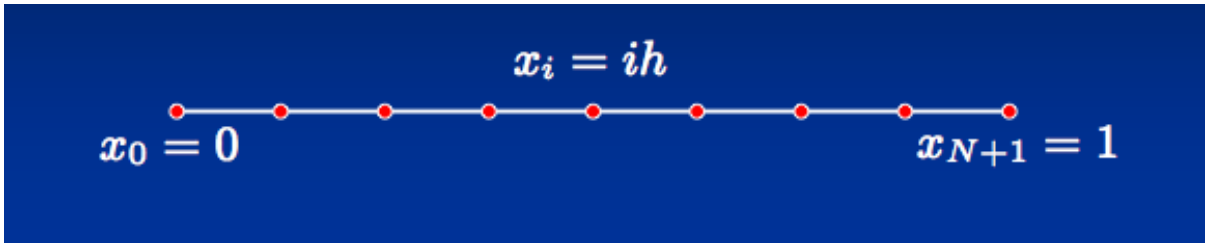
1

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} = 0,\ i = 1, \cdots, N,\ n = 0, \cdots, M-1,$$

2

$$u_i^0 = u_0(x_i), i = 1, \cdots, N,$$

Forward Euler scheme



Forward Euler scheme

1

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} = 0,\ i = 1, \cdots, N,\ n = 0, \cdots, M-1,$$

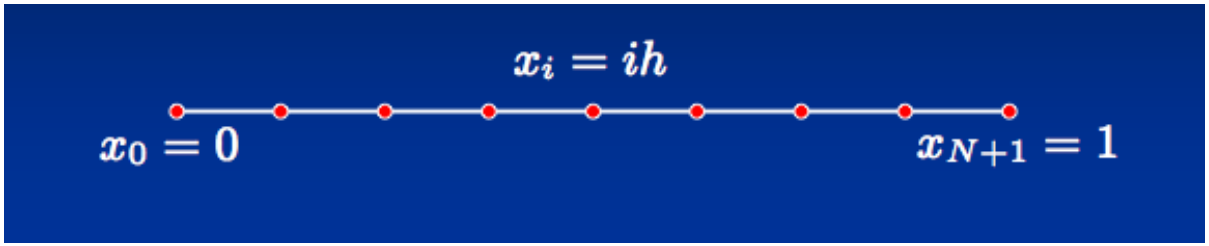
2

$$u_i^0 = u_0(x_i), i = 1, \cdots, N,$$

3

$$u_0^n = u_{N+1}^n = 0, n = 0, \cdots, M.$$

Forward Euler scheme



Forward Euler scheme

1

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} = 0,\ i = 1, \cdots, N,\ n = 0, \cdots, M-1,$$

2

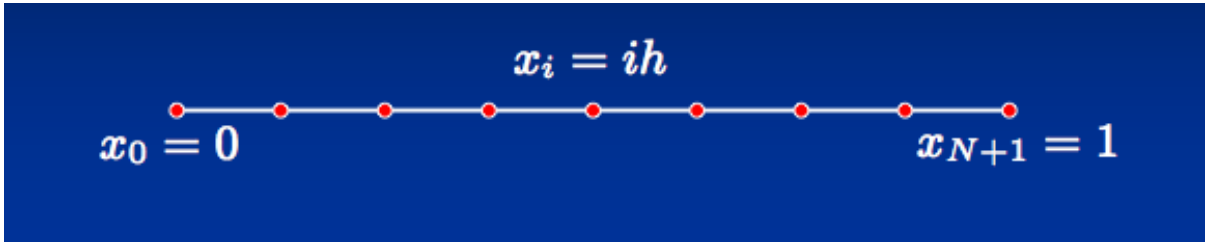
$$u_i^0 = u_0(x_i), i = 1, \cdots, N,$$

3

$$u_0^n = u_{N+1}^n = 0, n = 0, \cdots, M.$$

$$\lambda = \frac{\Delta t}{h^2}$$

Backward Euler scheme

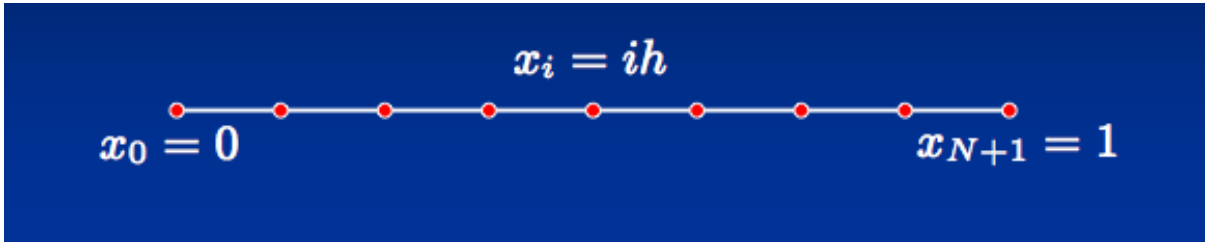


Backward Euler scheme

1

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2} = 0,\ i = 1, \cdots, N,\ n = 1, \cdots, M-1,$$

Backward Euler scheme



Backward Euler scheme

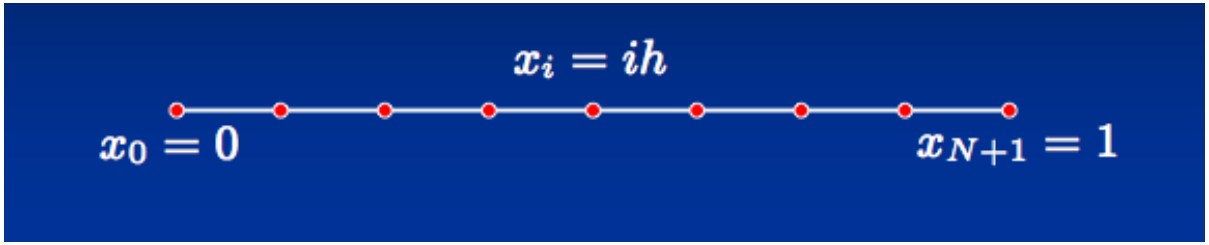
1

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2} = 0,\ i = 1, \cdots, N,\ n = 1, \cdots, M-1,$$

2

$$u_i^0 = u_0(x_i), i = 1, \cdots, N,$$

Backward Euler scheme



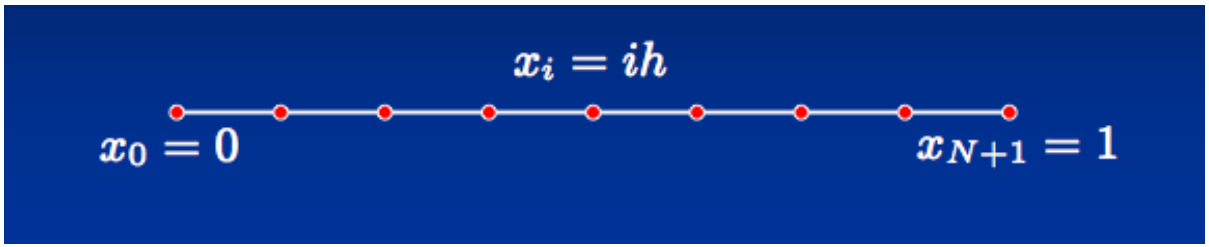
Backward Euler scheme

1. $$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2} = 0,\ i = 1, \cdots, N,\ n = 1, \cdots, M-1,$$

2. $$u_i^0 = u_0(x_i), i = 1, \cdots, N,$$

3. $$u_0^n = u_{N+1}^n = 0, n = 0, \cdots, M.$$

Upwind scheme / Lax-Wendroff scheme



1. $$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{h} = 0,$$

2. $$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} - \frac{a^2\Delta t}{2}\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = 0.$$

L^2 stability analysis

1. Let u be a 1-periodic function, the Fourier coefficients $\hat{u}(k)$, $k \in \mathbb{Z}$ are complex numbers defined by the integrals

$$\hat{u}(k) = \int_0^1 u(x) e^{-2\pi i k x} dx.$$

2. Parseval formula

$$\|u\|_2^2 = \sum_{k \in \mathbb{Z}} |\hat{u}(k)|^2.$$

3. Von Neumann analysis, there exists a constant $K > 0$ independent of Δt and h such that for all n, $1 \leq n \leq M$,

$$\|u^n\|_2 = \left(\sum_{j=1}^{N} h |u_j^n|^2 \right)^{1/2} \leq K \|u^0\|_2 = \left(\sum_{j=1}^{N} h |u_j^0|^2 \right)^{1/2}.$$

4. Stability, consistency, convergence - Lax-Richtmyer theorem.

Outline

Back to the future

Let n in $\mathbb{N}^*$, σ a function from $\mathbb{R}$ to $\mathbb{R}$, $M([0,1]^n)$ the space of finite, signed regular Borel measures and $C([0,1]^n)$ the space of continuous functions on $[0,1]^n$.

Definition.

We say that σ is **sigmoidal** if

$$\lim_{t \to -\infty} \sigma(t) = 0, \lim_{t \to +\infty} \sigma(t) = 1,$$

we say that σ is **discriminatory** if for a measure μ in $M([0,1]^n)$

$$\int_{[0,1]^n} \sigma\left(y^t x + \theta\right) d\mu(x) = 0$$

for all y in $\mathbb{R}^n$ and θ in $\mathbb{R}$ implies that $\mu = 0$. □

A sigmoidal function can not be a polynomial function.

Back to the future

Let n in $\mathbb{N}^*$, σ a function from $\mathbb{R}$ to $\mathbb{R}$, $M([0,1]^n)$ the space of finite, signed regular Borel measures and $C([0,1]^n)$ the space of continuous functions on $[0,1]^n$.

Theorem 1, Cybenko (1989).

Let σ be any continuous discriminatory function. The finite sums of the form

$$G(x) = \sum_{j=1}^{N} \alpha_j \sigma\left(y_j^t x + \theta_j\right)$$

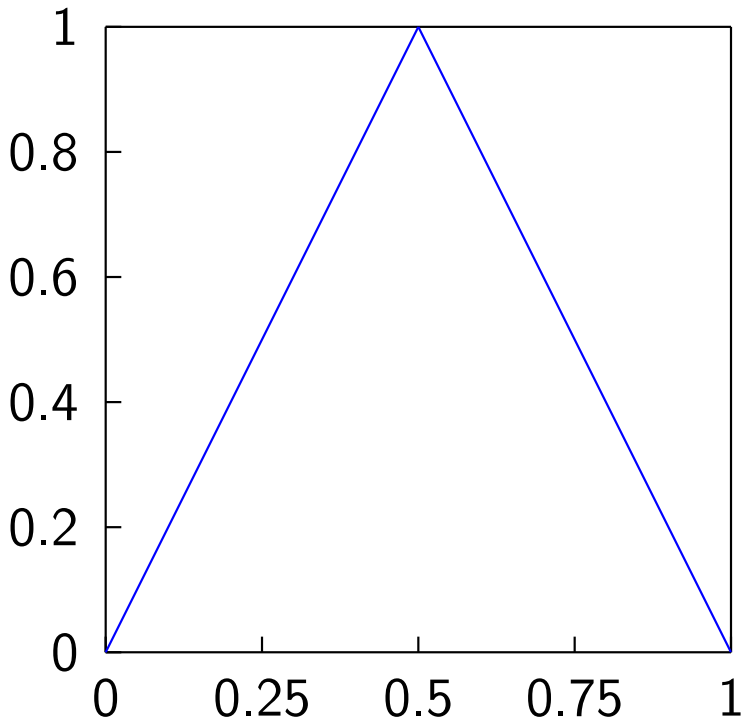
are dense in $C([0,1]^n)$. □

Lemma 1, Cybenko (1989).

Any bounded, measurable sigmoidal function, σ, is discriminatory. In particular, any continuous sigmoidal function is discriminatory. □

Approximation by ReLU Neural Networks

$$g(x) = \begin{cases} 2x, & 0 \le x \le 1/2, \\ 2(1-x), & 1/2 < x < 1. \end{cases}$$



Approximation of a one dimensional function by a Neural Networks (NN)

Let us suppose we want to approximate a one dimensional function $f \in C^n([0,1])$ with to simplify $\|f^{(k)}\|_\infty \leq 1$, $0 \leq k \leq n$, by Neural Networks. The main idea with the ReLU is to replace the piecewise linear functions by piecewise polynomials and we need the following ingredients :

- A NN f_1^* approximating linear functions

$$\sup_{x\in[0,1]} |f_1^*(x) - x| \leq \varepsilon$$

- A NN f_2^* approximating quadratic functions

$$\sup_{x\in[0,1]} |f_2^*(x) - x^2| \leq \varepsilon$$

- Approximate a partition of unity :

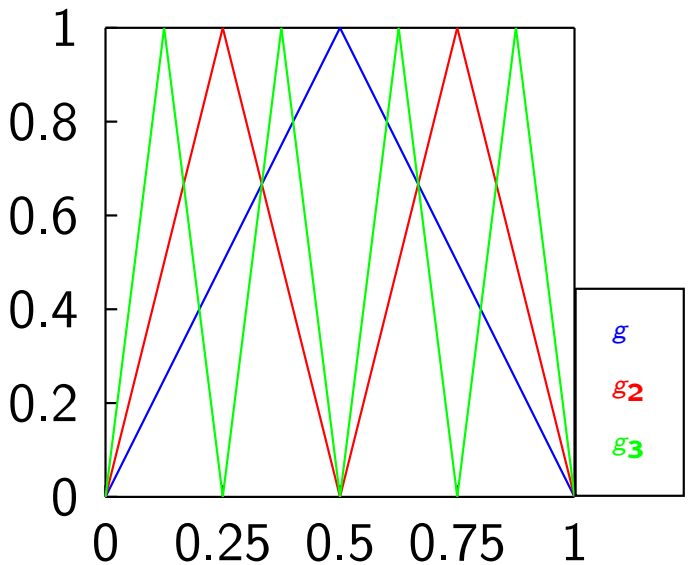
$$\psi_j \geq 0, \sum_{j=1}^{N} \psi_j = 1, \operatorname{supp} \psi_j \subset \left[\frac{j-1}{N}, \frac{j+1}{N}\right] \cap [0,1], j = 0, \cdots, N.$$

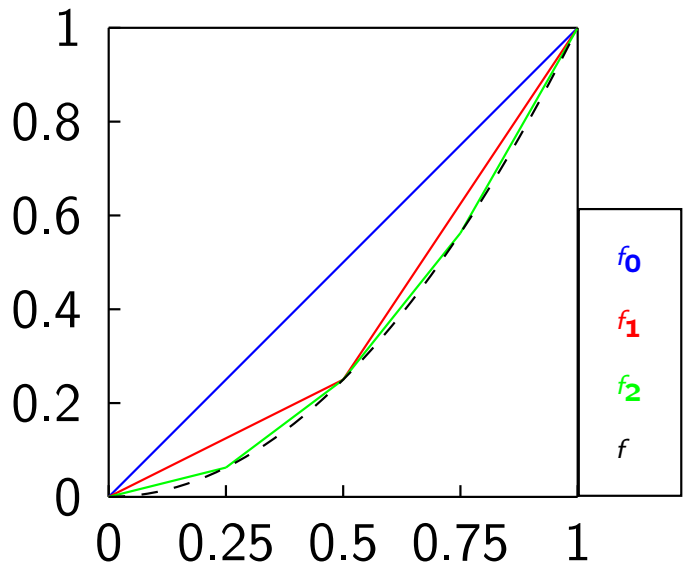
Approximation by ReLU Neural Networks

Step 1 : $x = \max(x,0) - \max(-x,0) = ReLU(x) - ReLU(-x) = \sigma(x) - \sigma(-x)$.
Remark : $\max(x,0) + \max(-x,0) = |x|$.
Step 2 :

Proposition 1, Yarotsky (2017).

The function $x \mapsto f(x) = x^2$ on the segment $[0,1]$ can be approximated with any error $\varepsilon > 0$ by a ReLU network. □





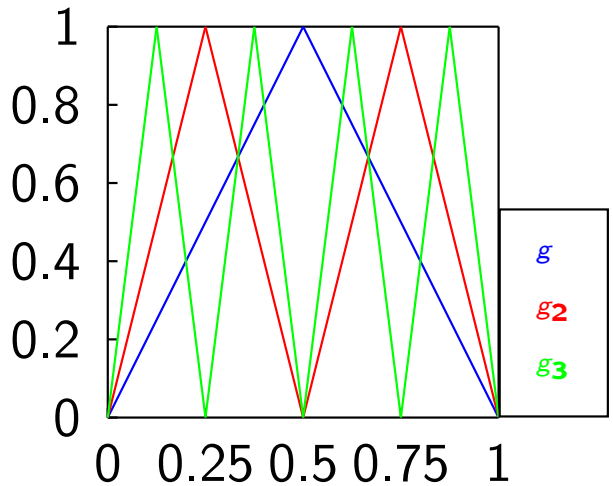
Approximation by ReLU Neural Networks

Lemma 2, Telgarsky (2015).

Let real $x \in [0,1]$ and positive integer m be given, and choose the unique nonnegative integer $i_m \in \{0, \cdots, 2^{m-1}\}$ and real $x_m \in [0,1[$ so that $x = (i_m + x_m)2^{1-m}$. Then

$$g_m(x) = \begin{cases} 2x_m, & 0 \le x_m \le 1/2, \\ 2(1-x_m), & 1/2 < x_m < 1. \end{cases}$$

□

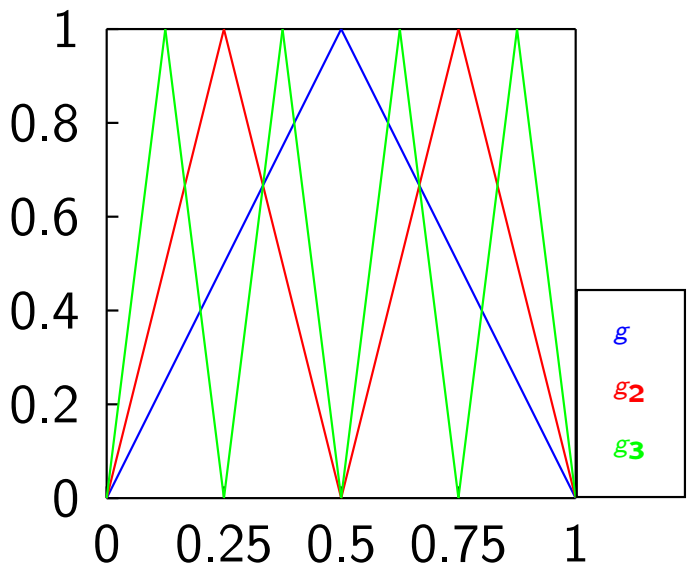


Approximation by ReLU Neural Networks

Lemma 3, Telgarsky (2015).

$$g_m(x) = \begin{cases} 2^m \left(x - \frac{2k}{2^m}\right), & x \in \left[\frac{2k}{2^m}, \frac{2k+1}{2^m}\right], k = 0, 1, \cdots, 2^{m-1} - 1 \\ 2^m \left(\frac{2k}{2^m} - x\right), & x \in \left[\frac{2k-1}{2^m}, \frac{2k}{2^m}\right], k = 1, 2, \cdots, 2^{m-1}. \end{cases}$$

□



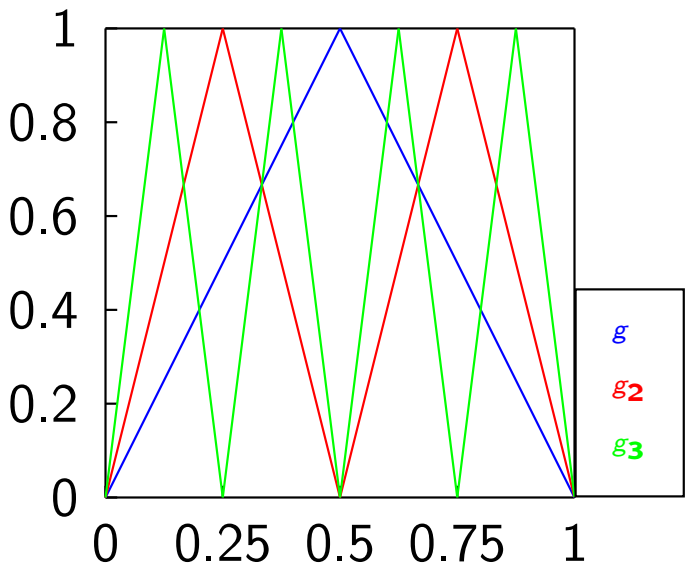
Approximation by ReLU Neural Networks

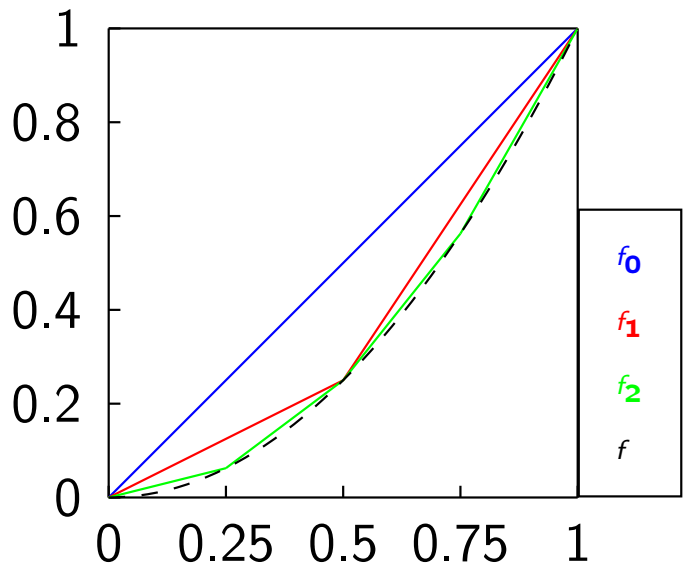
Let f_m be the piecewise linear interpolation of f with $2^m + 1$ uniformly distributed breakpoints $k/2^m$, $k = 0, \cdots, 2^m$,

$$f_m\left(\frac{k}{2^m}\right) = \left(\frac{k}{2^m}\right)^2, \ k = 0, \cdots, 2^m.$$

We have for all $x \in [0, 1]$

$$f_{m-1}(x) - f_m(x) = \frac{g_m(x)}{2^{2m}}, \ f_m(x) = x - \sum_{\ell=1}^{m} \frac{g_\ell(x)}{2^{2\ell}}$$

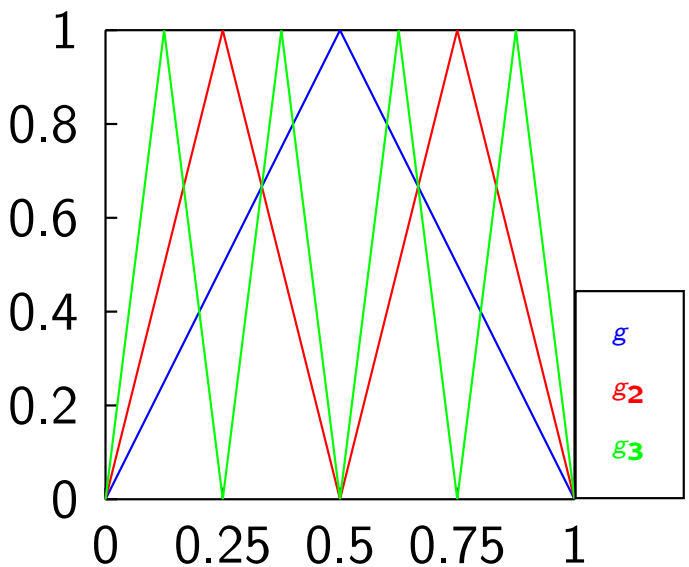


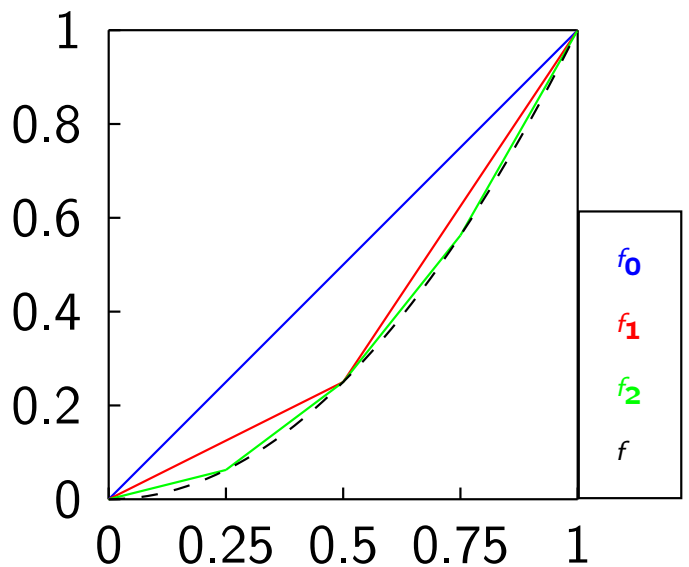


Approximation by ReLU Neural Networks

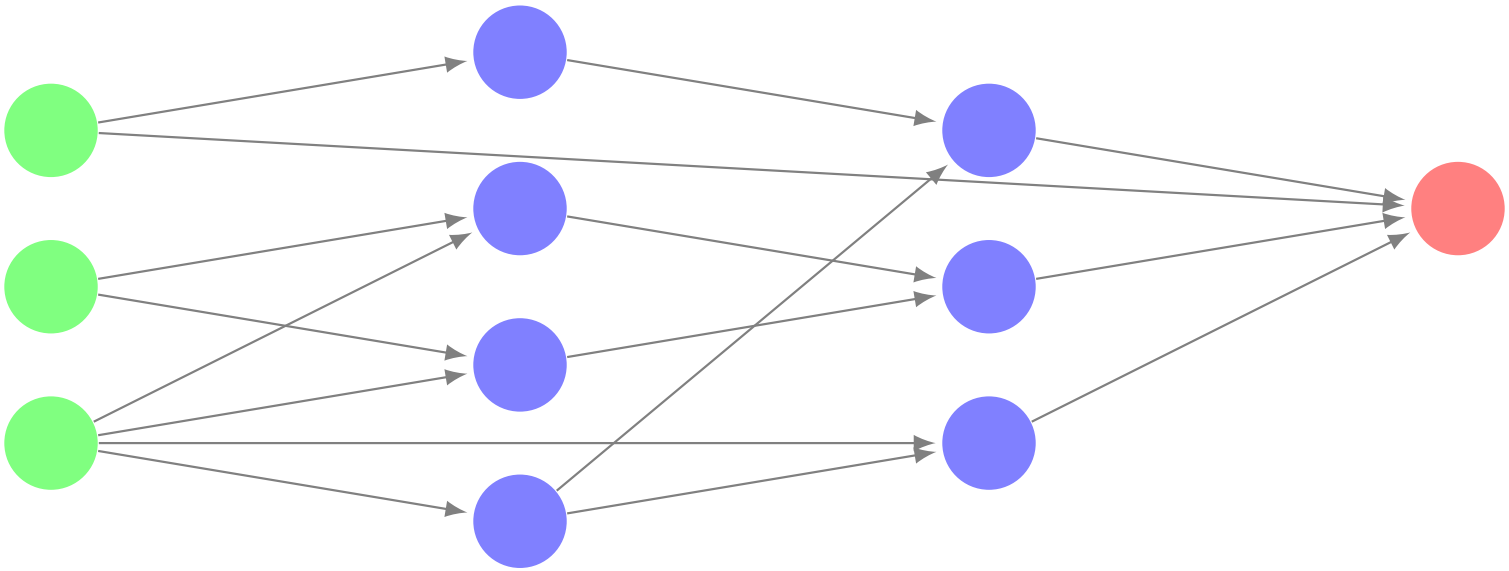
We have (for all $x \in [0,1]$) - (Exercices)

- $$\|f_m - f\|_{[0,1]} \leq 2^{-2m},$$
- $$g(x) = 2\sigma(x) - 4\sigma\left(x - \frac{1}{2}\right) + 2\sigma(x-1).$$





Approximation by ReLU Neural Networks



A feedforward neural network having 3 input units, 1 output unit, and 7 computation units with nonlinear activation. The network has 4 layers and $16 + 8 = 24$ weights.

Proposition 2, Yarotsky (2017).

Given $M > 0$ and $\varepsilon \in]0, 1[$, there exists a ReLU network η with two input units that implements a function $\omega : \mathbb{R}^2 \to \mathbb{R}$ such that

1. for any inputs x, y if $|x| \leq M$ and $|y| \leq M$, then $|\omega(x, y) - xy| \leq \varepsilon$,
2. if $x = 0$ or $y = 0$ then $\omega(x, y) = 0$.

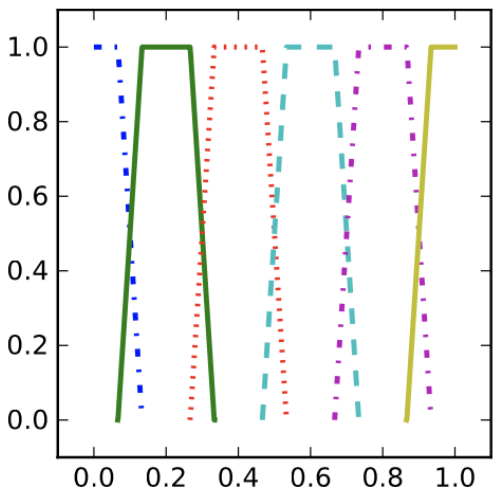
Approximation by ReLU Neural Networks

Step 3 : Partition of unity

$$\psi(x) = \begin{cases} 1 & |x| < 1, \\ 0 & 2 < |x|, \\ 2 - |x| & 1 \leq |x| \leq 2. \end{cases}$$

Let $N \in \mathbb{N}^*$, $x \in [0,1]$, $\psi_j(x) = \psi(3N(x - j/N))$, $j = 0, \cdots, N$,

$$\sum_{j=0}^{N} \psi_j(x) = 1.$$



Approximation by ReLU Neural Networks

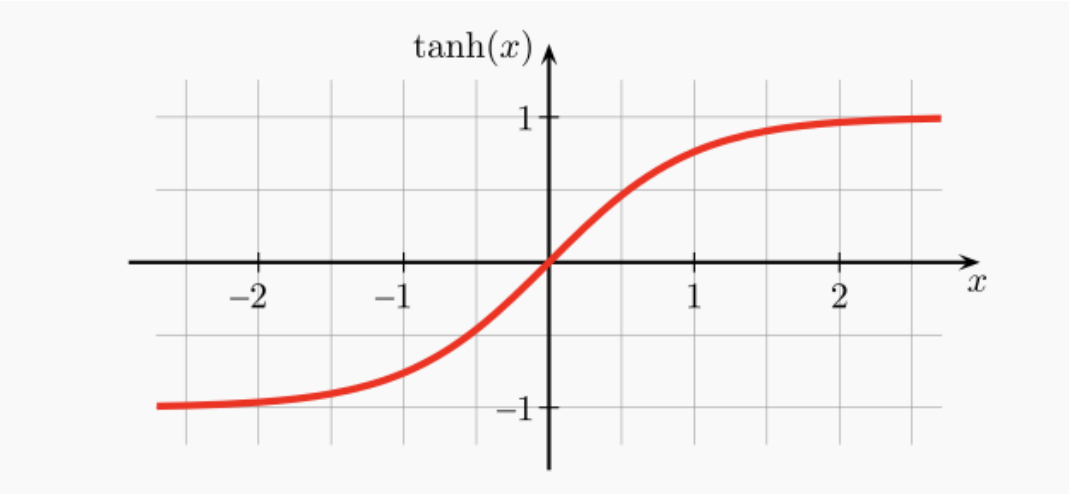
Theorem 2, Yarotsky (2017) - Upper bounds.

For any $n \in \mathbb{N}^*$ and $\varepsilon \in [0,1]$, a function f belonging to $C^n([0,1])$ can be approximated by a ReLU network with error ε. Moreover there exists a constant $c(n)$ such that this ReLU network has the depth at most $c(n)(\ln(1/\varepsilon)+1)$ and at most $c(n)\varepsilon^{-1/n}(\ln(1/\varepsilon)+1)$ weights and computation units. □

Towards PINNs : Approximation by tanh Neural Networks

New activation function defined for all x in $\mathbb{R}$ by

$$\sigma(x) = \tanh(x).$$



σ is a C^∞ function satisfying for x in $\mathbb{R}$,

$$\sigma'(x) = 1 - \sigma^2(x) = \frac{4}{(e^x + e^{-x})^2}$$

Towards PINNs : Approximation by tanh Neural Networks

Theorem 3, De Ryck, Lanthaler, Mishra (2021).

For any $n \in \mathbb{N}^*$ and $\varepsilon \in [0,1]$, a function f belonging to $C^n([0,1])$ can be approximated by a tanh network with error ε. Moreover the tanh network has two hidden layers and there exists a constant $c(n,f)$ such that one is of width at most

$$3 \min \left(p \in \mathbb{N} | p \geq \frac{n}{2} \right) + \frac{c(n,f)}{\sqrt[n]{\varepsilon}} - 1$$

and the other of width at most

$$\frac{6c(n,f)}{\sqrt[n]{\varepsilon}}.$$

□

Towards PINNs : Approximation by tanh Neural Networks

Proposition 3, De Ryck, Lanthaler, Mishra (2021).

Let $p \in \mathbb{N}^*$, the function defined for all x in $[0,1]$ by x^p can be approximated with any error $\varepsilon > 0$ by a tanh network. □

- $p = 1$, exercice
- p odd, use pth order central finite difference approximation with y in $[0,1]$, $h > 0$,

$$\delta^p_{y,h}[\sigma] = \sum_{i=0}^{p} (-1)^i C_p^i \sigma\left(\left(\frac{p}{2} - i\right) yh\right).$$

- p even, $\sigma^{(p)}(0) = 0$, use for y in $\mathbb{R}$ and $\alpha > 0$,

$$y^p = \frac{1}{2\alpha(p+1)} \left((y+\alpha)^{p+1} - (y-\alpha)^{p+1} - 2 \sum_{k=0}^{p/2-1} C_{p+1}^{2k} \alpha^{p-2k+1} y^{2k} \right)$$

Towards PINNs : Approximation by tanh Neural Networks

Proposition 3, De Ryck, Lanthaler, Mishra (2021).

Let $p \in \mathbb{N}^*$, the function defined for all x in $[0,1]$ by x^p can be approximated with any error $\varepsilon > 0$ by a tanh network. □

Consequence.

Proposition 3 + Stone-Weirstrass theorem : A continuous function on $[0,1]$ by x^p can be approximated with any error $\varepsilon > 0$ by a tanh network. □

Consequence 2.

Approximation with error ε of the product of two reals x and y by a tanh network. □

Towards PINNs : Approximation by tanh Neural Networks

Step 3 : from "Partition of unity" to "Approximation of partition of unity"
Let $\varepsilon > 0$ and $N \in \mathbb{N}^*$,

$$[0,1] = \bigcup_{j=1}^{N} \left[\frac{j-1}{N}, \frac{j}{N} \right].$$

We fix $\alpha > 0$ large enough such that

$$1 - \sigma\left(\frac{\alpha}{N}\right) \leq \varepsilon.$$

For j, $2 \leq j \leq N-1$ and y in $\mathbb{R}$, we define

$$\rho_j^N(y) = \frac{1}{2}\sigma\left(\alpha\left(y - \frac{j-1}{N}\right)\right) - \frac{1}{2}\sigma\left(\alpha\left(y - \frac{j}{N}\right)\right)$$

and

$$\rho_1^N(y) = \frac{1}{2} - \frac{1}{2}\sigma\left(\alpha\left(y - \frac{1}{N}\right)\right),$$

$$\rho_N^N(y) = \frac{1}{2}\sigma\left(\alpha\left(y - \frac{N-1}{N}\right)\right) + \frac{1}{2}.$$

Towards PINNs : Approximation by tanh Neural Networks

Step 3 : from "Partition of unity" to "Approximation of partition of unity"
Let $\varepsilon >$ and $N \in \mathbb{N}^*$,

$$[0,1] = \bigcup_{j=1}^{N} \left[\frac{j-1}{N}, \frac{j}{N} \right].$$

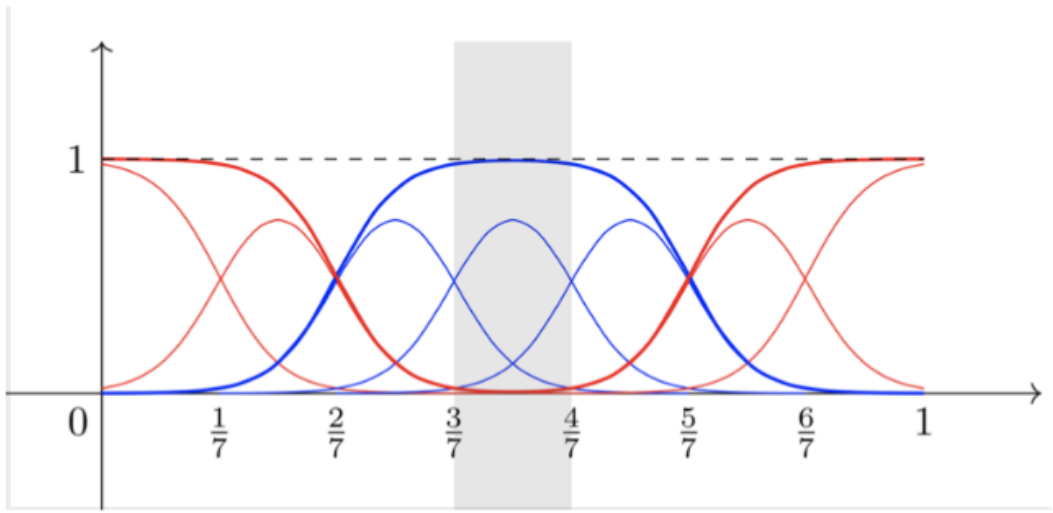


Figure – An example with $N = 7$

Towards PINNs : Approximation by tanh Neural Networks

Lemma 3, De Ryck, Lanthaler, Mishra (2021).

For a fixed j and $\varepsilon > 0$,

$$\left\| \sum_{\ell=j-1}^{j+1} \rho_\ell^N - 1 \right\|_{[\frac{j-1}{N}, \frac{j}{N}]} \leq \varepsilon$$

and for $\ell \neq j-1, j, j+1$,

$$\left\| \rho_\ell^N \right\|_{[\frac{j-1}{N}, \frac{j}{N}]} \leq \varepsilon.$$

□

Almost the same properties as the family of functions ψ_i, $0 \leq i \leq N$.

PINNs : Two disappointing results

Back to the model of dynamics with frictions.

- $mu'' + \mu u' = 0$ on $]0, T[$, $T > 0$.
- Challenge : the physical model is incomplete because the initial conditions are unknown but we have n noisy observations y_i at the points x_i, $1 \leq i \leq n$.
- Other data : a sample of independent and identically distributed random variables $x_j^{(r)}$, $1 \leq j \leq n_r$, uniformly distributed on $]0, T[$.
- Loss function :

$$R_{n,n_r}(u_{NN}(.,\theta)) = \frac{1}{n}\sum_{i=1}^{n} |u_{NN}(x_i,\theta) - y_i|^2 + \frac{1}{n_r}\sum_{j=1}^{n_r} \left((mu''_{NN} + \mu u'_{NN}) \left(x_j^{(r)}, \theta \right) \right)^2$$

PINNs : Two disappointing results

Theorem 3 (overfitting), Doumèche, Biau, Boyer (2023).

Consider the dynamics with friction model and assume that there are two observations such that $y_i \neq y_j$, $i \neq j$. Then, whenever one hidden size L of the neural network satisfies $L \geq n - 1$, for any integer n_r, for all $x_i^{(r)}$, $1 \leq i \leq n_r$, there exists a minimizing sequence $(u_{NN}(., \theta_p))_{p \in \mathbb{N}}$ such that

$$\lim_{p \to +\infty} R_{n,n_r}(u_{NN}(., \theta_p)) = 0$$

but

$$\lim_{p \to +\infty} \int_0^T \left(\left(m u''_{NN} + \mu u'_{NN} \right)(x, \theta_p) \right)^2 dx = +\infty$$

So, this PINN estimator is not consistent. □

PINNs : Two disappointing results

Key ingredient : For θ in $\mathbb{R}$ and x in $\mathbb{R}$

$$\tanh_\theta(x) = \tanh(\theta x).$$

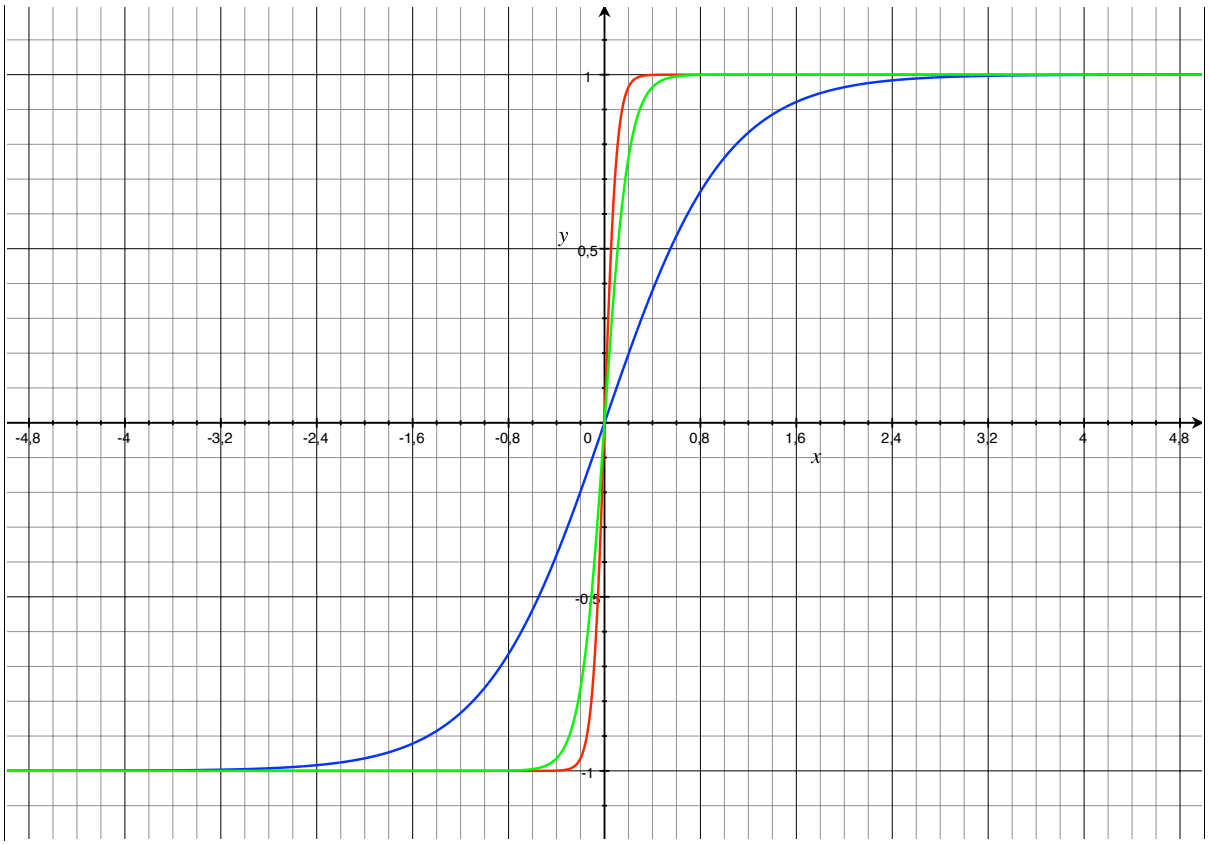


Figure – Graphs of $\tanh_\theta$ for $\theta = 1, 5, 10$