Approximation with activation functions and applications

Radhia Bessi

Approximation with activation functions and applications

Radhia Bessi

LAMSIN, ENIT
Université Tunis El Manar
Tunis
Tunisie
radhia.bessi@enit.utm.tn

ABSTRACT. Function approximation arises in many branches of applied mathematics and computer science, in particular in numerical analysis, in finite element theory and more recently in data sciences domain. From most common approximation we cite, polynomial, Chebychev and Fourier series approximations.
In this work we establish some approximations of a continuous function by a series of activation functions. First, we deal with one and two dimensional cases. Then, we generalize the approximation to the multi dimensional case. Examples of applications of these approximations are: interpolation, numerical integration, finite element and neural network. Finally, we will present some numerical results of the examples above.

RÉSUMÉ. La théorie d'approximation des fonctions couvre de nombreuses branches en mathématiques appliquées, en informatique et en sciences de l'ingénieur, en particulier en analyse numérique, en théorie des éléments finis et plus récemment en sciences des données.
Parmi les approximations fortement utilisées nous citons les approximations polynomiale de type Lagrange, Hermite ou au sens de Chebychev. Nous trouvons aussi l'approximation d'une fonction par une séries de Fourier, l'approximation rationnelle...
Dans ce travail, nous établissons quelques résultats d'approximations d'une fonction continue par une série de fonctions de type activation. Nous traitons d'abord les cas d'une fonction à une seule puis à deux variables, puis nous généralisons l'approximation au cas multidimensionnel.
Nous appliquons ces approximations pour l'interpolation et l'intégration numérique, en éléments finis et en réseau neuronal. Nous donnons pour chaque application quelques résultats numériques.

KEYWORDS : Function approximation, interpolation, Runge's phenomenon, Chebychev points, neural network, universal approximation theorem, numerical integration, finite element.

MOTS-CLÉS : Approximation d'une fonction, interpolation, phénomène de Runge, points de Chebychev, réseau neuronal, théorème universel d'approximation, intégation numérique, éléments finis.

1. Introduction

Motivated by the architecture of the human brain, neural networks are composed of multiple hidden layers. Each hidden layer has multiple hidden nodes. Each node is an activation function of an affine transformation of the outputs of the previous layer. Examples of activation functions:

- Heaviside function $H : x \mapsto \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{if} \quad x < 0 \end{cases}$
- Relu function : Relu : $x \to \max(0, x)$.
- Quadratic Relu function: $\text{Relu}_2 : x \to \frac{1}{2}\max(0, x)^2$.
- Sigmoid function: $\sigma : x \to \frac{1}{1+e^{-x}}$.
- Softplus function : $\sigma_p : x \to \log(1 + e^{-x})$.

The starting point of the study of neural networks theory was based on the universal approximation theorem. It was proved in [1] the density of neural networks with one hidden layer in the space of continuous functions.

Later, it was proved in [7] that multilayer feed-forward network with a locally bounded piecewise continuous activation function can approximate any continuous function if and only if the network's activation function is not polynomial.

The authors in [2] proposed an algorithm to find the optimal approximations of convex univariate functions with feed-forward rectified linear unit (Relu) neural networks. They studied the minimal approximation error and the number of approximating linear pieces.

Artificial neural networks with Relu activation function were used in [3] to approximate discontinuous piecewise functions up to L^2 error. Optimal rates for approximating these piecewise functions by Relu neural networks, measuring the complexity of the networks in terms of the number of nonzero weights, were established.

An L^∞ and L^2 error bounds for functions of many variables that are approximated by linear combinations of Relu and squared Relu functions were established in [5]. Approximation of continuous multi-variate functions with Deep Relu neural networks and conventional fully-connected architectures was developed in [6].

Authors in [9] looked into the relationship between deep neural networks with Relu function as the activation function and continuous piecewise linear functions.

Also, they gave a one dimensional example to illustrate that a finite element method using Relu neural network can lead to better approximation result than adaptive finite element method.

Recently, there are increased new research interests for the application of neural network for numerical approximation of nonlinear a PDEs as in [9, 10].

Many forms of interpolation can be constructed by picking a different class of interpolates like, Lagrange or Hermite polynomial interpolation [4], or rational interpolation using Padé approximation [8], or also, trigonometric interpolation, which is interpolation by trigonometric polynomials using Fourier series.

Numerical integration theory is in general based on interpolation functions. It provides a basic and important tool for the numerical resolution of other problems and can be applied, for example, in numerical methods for ordinary or partial differential equations and in finite element method.

In this work, we prove that a continuous function can be explicitly approximated by a neural network with two layers and non linear activation function in the hidden layer. This is a version of the universal approximation theorem in the case of single variate. The novelty here is that we precise weights of the network. But, we do not have linearity of the neural network for multi- dimensional case using Relu or heaviside as activation function. We prove the approximation for one and two dimensional cases and extend the result to the multi-variables continuous functions. We apply these approximations to interpolation, numerical integration and finite element for one and two-variate functions.

2. Approximation of univariate function

Without loss of generality, we consider a function f defined on the interval $[0,1]$. Our goal here is to prove that f is a uniform limit of a series of the form $\sum_{n\in\mathbb{N}} \alpha_n \varphi(w_n x + b_n)$, for different types of activation functions φ. This means that f can be approximated by one hidden layer neural network with φ as activation function.

2.1. Approximation by Relu activation function

Proposition 2.1 Let $f : [0,1] \to \mathbb{R}$ a continuous function. Then, f is the uniform limit on $[0,1]$ of

$$S_n(x) = \sum_{k=0}^{n} \alpha_k \text{ Relu}((k+1) - nx),$$

where the vector $\alpha = \begin{pmatrix} \alpha_0, \alpha_1, ..., \alpha_n \end{pmatrix}^T$ is given by $\alpha = A^{(1)}_{R,n} F_n$, for the matrix $A^{(1)}_{R,n}$ of size $(n+1)$

$$A^{(1)}_{R,n} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -2 & 1 & \ddots & & \vdots \\ \vdots & \ddots & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 & -2 & 1 \\ \vdots & & \ddots & \ddots & \ddots & 1 & -2 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix}, \tag{1}$$

and for the vector $F_n = \begin{pmatrix} f(0), f(\frac{1}{n}), ..., f(\frac{n-1}{n}), f(1) \end{pmatrix}^T$

Proof. Since f is continuous on the closed bounded interval, then it is bounded and uniformly continuous. Let $M = \sup_{\in[0,1]} |f(x)|$ and for a given $\varepsilon > 0$, there exists $\eta > 0$ such that for all $x, x' \in [0,1]$ satisfying $|x - x'| \leq \eta$, we have

$$|f(x) - f(x')| \leq \frac{\varepsilon}{2}.$$

Fixing $n \geq n_0 = E(\frac{1}{\eta}) + 1$, (where here $E(\frac{1}{\eta})$ is the integer part of the real $\frac{1}{\eta}$). Let $x \in [0,1]$. If $x = 1$, then we have clearly $S_n(1) = \alpha_n = f(1)$ and hence $|S_n(1) - f(1)| = 0 \leq \varepsilon$. If $x \in [0,1[$, then, there exists $0 \leq k_x \leq n$ such that $x \in [\frac{k_x}{n}, \frac{k_x+1]}{n}]$.

$$\begin{aligned} S_n(x) &= \sum_{k \geq k_x} \alpha_k \text{ Relu}((k+1) - nx) \\ &= \sum_{n-2 \geq k \geq k_x} \left[f(\frac{k}{n}) - 2f(\frac{k+1}{n}) + f(\frac{k+2}{n})) (k+1-nx) \right] \\ &+ (f(\frac{n-1}{n}) - 2f(1))(n - nx) + f(1)(n+1-nx) \\ &= f(\tfrac{k_x}{n})(k_x + 1 - nx) - f(\tfrac{k_x+1}{n})(k_x - nx). \end{aligned}$$

Then

$$|S_n(x) - f(x)| = |(f(\frac{k_x+1}{n}) - f(\frac{k_x}{n}))(k_x - nx) + f(\frac{k_x}{n}) - f(x)|.$$

Since $x \in [\frac{k_x}{n}, \frac{k_x+1]}{n}]$, then $|k_x - nx| \leq 1$. In addition $|\frac{k_x+1}{n} - \frac{k_x}{n}| = \frac{1}{n} \leq \eta$, and $|\frac{k_x}{n} - x| \leq |\frac{k_x+1}{n} - \frac{k_x}{n}| = \frac{1}{n} \leq \eta$, it follows that

$$|S_n(x) - f(x)| \leq |(f(\frac{k_x+1}{n}) - f(\frac{k_x}{n})| + |f(\frac{k_x}{n}) - f(x)| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

(S_n) converges uniformly to f on [0,1]. ■

Remarks 2.2

1. Coefficients (α_k) are such that : $S_n(\frac{j}{n}) = f(\frac{j}{n})$ for all $j = 0, ..., n$. Since every activation function $\varphi_k : x \mapsto$ Relu$(k+1-nx)$ satisfies

$$\varphi_k(\frac{j}{n}) = \begin{cases} 0 & \text{if} \quad 1 \leq k+1 \leq j \leq n \\ k+1-j & \text{if} \quad 0 \leq j \leq k+1 \leq n+1 \end{cases},$$

then the vector $(\alpha_k)_{0 \leq k \leq n}$ is solution to the triangular linear system $B_{R,n}^{(1)} \alpha = F_n$ with

$$B_{R,n}^{(1)} = \begin{pmatrix} 1 & 2 & 3 & & ... & n & n+1 \\ 0 & 1 & 2 & 3 & \cdots & (n-1) & n \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \ddots & \ddots & 1 & 2 & 3 & 4 \\ \vdots & \cdots & \ddots & \ddots & 1 & 2 & 3 \\ \vdots & \cdots & \cdots & \ddots & \ddots & 1 & 2 \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

It is easy to check that the matrix $B_{R,n}^{(1)}$ is nothing else but the inverse of the triangular matrix $A_{R,n}^{(1)}$. This means that this Relu approximation is also an interpolation of f at $(n+1)$ distinct nodes $x_j = \frac{j}{n}$, $j = 0, ..., n$.

2. A continuous function $f : [a, b] \to \mathbb{R}$, has similarly a Relu approximation of the form

$$f(x) = \lim_{n\to+\infty} \sum_{k=0}^{n} \alpha_k \text{ Relu}(k + 1 - \frac{n}{b-a}(x-a)),$$

where the vector $\alpha = A^{(1)}_{R,n} F_n$, with $A^{(1)}_{R,n}$ is the same matrix (1) and

$$F_n = \begin{pmatrix} f(a) \\ f(a + \frac{b-a}{n}) \\ ... \\ f(a + \frac{(n-1)(b-a)}{n}) \\ f(b) \end{pmatrix}.$$

3. Similarly, a Heaviside-approximation of a continuous function f is a uniform sum of Heaviside functions as follows : $f(x) = \lim_{n\to+\infty} H_n(x)$, with

$$H_n(x) = \sum_{k=0}^{n-1} [f(\frac{k}{n}) - f(\frac{k+1}{n})]H(k + 1 - nx) + f(1)H(n + 1 - nx).$$

Indeed, for $x \in [\frac{k_x}{n}, \frac{k_x+1]}{n}]$, we easily have $|H_n(x) - f(x)| = |f(\frac{k_x}{n}) - f(x)|$. Using uniform continuity of f, we easily deduce uniform convergence of (H_n) to f.

4. The approximation of f by a-one hidden neural network contains a bias term. We modify slightly the last Relu approximation and we prove in a similar way that f is also a uniform limit of

$$T_n(x) = \sum_{k=0}^{n-1} \beta_k \text{ Relu}((k + 1) - nx) + \beta_n,$$

where the vector $\beta = \begin{pmatrix} \beta_0, \beta_1, ..., \beta_n \end{pmatrix}^T$ is given by $\beta = A_n F_n$, where

$$A_n = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -2 & 1 & \ddots & & \vdots \\ \vdots & \ddots & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 & -2 & 1 \\ \vdots & & \ddots & \ddots & \ddots & 1 & -1 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix}. \quad (2)$$

A_n is the inverse of

$$B_n = \begin{pmatrix} 1 & 2 & 3 & & ... & n & 1 \\ 0 & 1 & 2 & 3 & \cdots & (n-1) & 1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \ddots & \ddots & 1 & 2 & 3 & 1 \\ \vdots & \cdots & \ddots & \ddots & 1 & 2 & 1 \\ \vdots & \cdots & \cdots & \ddots & \ddots & 1 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

The bias here is β_n and we retrieve the universal approximation theorem for the one dimensional case.

5. On the contrary of polynomial, cubic spline, regression or trigonometric Fourier approximation, last Relu or Heaviside interpolations are explicit and simple to compute for every integer n. Moreover, they are very easy to integrate and to differentiate.

2.2. Interpolation by Relu function: general case

As it was mentioned before, approximation by Relu function in proposition (2.1) is also an interpolation of f at the $(n+1)$ equidistant points $(x_j)_{j=0,...,n}$ of the interval $[0,1]$. In general, given a data set $(x_i, y_i)_{i=0,...,n}$, with $x_0 < x_1 < ... < x_n$ and $y_i \in \mathbb{R}$, we propose here to find a function g of the form $g_n(x) = \sum_{k=0}^{n-1} \alpha_k \text{ Relu}(w_k x + b_k) + \alpha_n$ satisfying $g_n(x_i) = y_i$.

Proposition 2.3 Let g_n be the function defined by

$$g_n(x) = \sum_{i=0}^{n-1} \alpha_i \text{ Relu}(\frac{x_{i+1} - x}{x_{i+1} - x_i}) + \alpha_n,$$

with the vector $\alpha = \begin{pmatrix} \alpha_0, \alpha_1, ..., \alpha_n \end{pmatrix}^T$ is given by $\alpha = D_n F_n$, where the square matrix D_n of size $n+1$ and the vector F_n are:

$$D_n = \begin{pmatrix} 1 & -\frac{x_2-x_0}{x_2-x_1} & \frac{x_2-x_0}{x_2-x_1} - 1 & 0 & \dots & 0 \\ 0 & 1 & -\frac{x_3-x_1}{x_3-x_2} & \frac{x_3-x_1}{x_3-x_2} - 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -\frac{x_{n-1}-x_{n-3}}{x_{n-1}-x_{n-2}} & \frac{x_{n-1}-x_{n-3}}{x_{n-1}-x_{n-2}} - 1 \\ 0 & \dots & \dots & 0 & 1 & -1 \\ 0 & \dots & \dots & \dots & 0 & 1 \end{pmatrix}$$

and $F_n = \begin{pmatrix} y_0, y_1, ..., y_{n-1}, y_n \end{pmatrix}^T$. Then $g_n(x_i) = y_i$, for all $i = 0, ..., n$.

Proof. To determine g of the form $g_n(x) = \sum_{i=0}^{n-1} \alpha_i \text{ Relu}(\frac{x_{i+1} - x}{x_{i+1} - x_i}) + \alpha_n$ satisfying $g_n(x_i) = y_i$, for all $i = 0, ..., n$, it is sufficient to look for $(\alpha_i)_{i=0,...,n}$ solution to the following system :

$$\begin{cases} \alpha_0 + \frac{x_2-x_0}{x_2-x_1}\alpha_1 + ... + \frac{x_n-x_0}{x_n-x_{n-1}}\alpha_{n-1} + \alpha_n &= y_0 \\ \alpha_1 + \frac{x_3-x_1}{x_3-x_2}\alpha_2 + ... + \frac{x_n-x_1}{x_n-x_{n-1}}\alpha_{n-1} + \alpha_n &= y_1 \\ \vdots & \vdots \\ \alpha_i + \frac{x_{i+2}-x_i}{x_{i+2}-x_{i-1}}\alpha_{i+1} + ... + \frac{x_n-x_i}{x_n-x_{n-1}}\alpha_{n-1} + \alpha_n &= y_i \\ \vdots & \vdots \\ \alpha_{n-2} + \frac{x_n-x_{n-2}}{x_n-x_{n-1}}\alpha_{n-1} + \alpha_n &= y_{n-2} \\ \alpha_{n-1} + \alpha_n &= y_{n-1} \\ \alpha_n &= y_n \end{cases}$$

The matrix of the last linear system is :

$$D'_n = \begin{pmatrix} 1 & \frac{x_2-x_0}{x_2-x_1} & \frac{x_3-x_0}{x_3-x_2} & \dots & \dots & \frac{x_n-x_0}{x_n-x_{n-1}} & 1 \\ 0 & 1 & \frac{x_3-x_1}{x_3-x_2} & \frac{x_4-x_1}{x_4-x_3} & \dots & \frac{x_n-x_1}{x_n-x_{n-1}} & 1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & \frac{x_{n-1}-x_{n-3}}{x_{n-1}-x_{n-2}} & \frac{x_n-x_{n-3}}{x_n-x_{n-1}} & 1 \\ 0 & \dots & \dots & 0 & 1 & \frac{x_n-x_{n-2}}{x_n-x_{n-1}} & 1 \\ 0 & \dots & \dots & \vdots & 0 & 1 & 1 \\ 0 & \dots & \dots & \dots & 0 & 0 & 1 \end{pmatrix}$$

This matrix is clearly non singular and its inverse is the band-upper triangular matrix D_n. ■

Remark 2.4 The proof of the last approximation shows the existence and the uniqueness of the coefficients α_i for $w_i = -\frac{1}{x_{i+1}-x_i}$ and $b_i = \frac{x_{i+1}}{x_{i+1}-x_i}$, but we do not have uniqueness of the weights w_i and bias b_i. If the points are equispaced, we find the approximation of type (2.2) which is similar to the one of proposition (2.1).

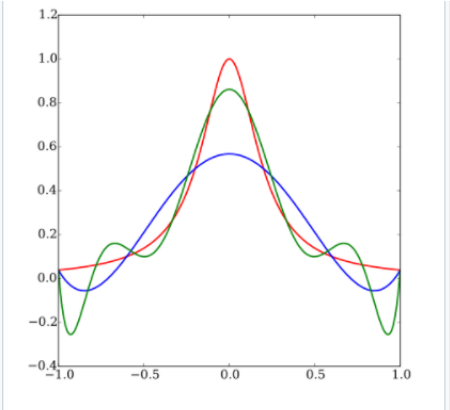
2.3. Applications

There are many applications of the two last Relu-approximation functions. Python programming language is used to code some of these applications presented in the rest of this work.

2.3.1. Interpolation

We can use this approximation in interpolation theory. It is known that for Runge function $f(x) = \frac{1}{1+25x^2}$ for $x \in [-1, 1]$, if we choose equidistant points on $[-1, 1]$, we find Runge's phenomenon which is a problem of oscillation at the edges -1 and 1 of the

interval that occurs when using polynomial interpolation with high degree. This is due to the fact that uniform convergence is not guaranteed, unlike approximations by Relu function.



The Runge function (red, highest central peak); the 5th-order interpolating polynomial with equally spaced interpolating points (blue, lowest central peak); and the 9th-order interpolating polynomial with equally spaced interpolating points (green, medium central peak)

Figure 1 – Runge phenomenen: [Wikipedia].

Our first numerical application for Chebychev and equidistant points will be tested on the interval $[-1, 1]$, since most existing results for Runge phenomenen are implemented in this interval. For the remaining applications, we deal with $f : [0, 1] \to \mathbb{R}$. Figures (2) and (3) display numerical results of Runge function interpolation, first over equidistant points, then over Chebychev points. We present in green color interpolation points $(x_i)_{i=0,\dots,n}$.

For the same function f, we also test Heaviside approximation whith equidistant points for $n = 10$ and $n = 50$. Results are displayed in figure (4).

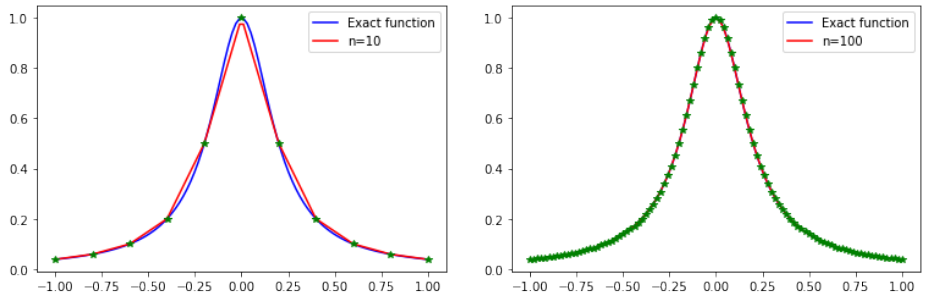


Figure 2 – Relu interpolation with equidistant points.

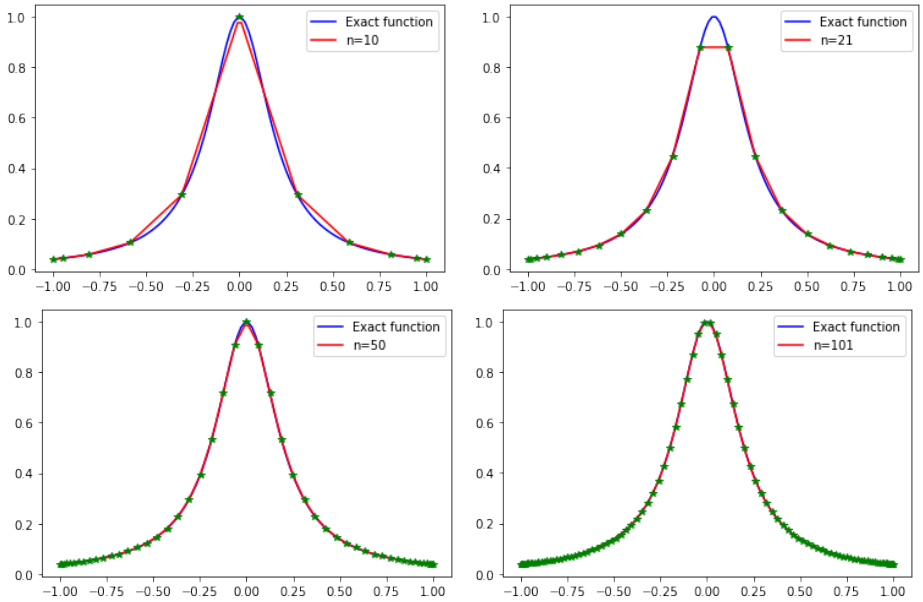


Figure 3 – Relu interpolation with Chebychev points.

We notice that uniform convergence is faster with Relu-approximation than Heaviside approximation as (2) and (4) show:

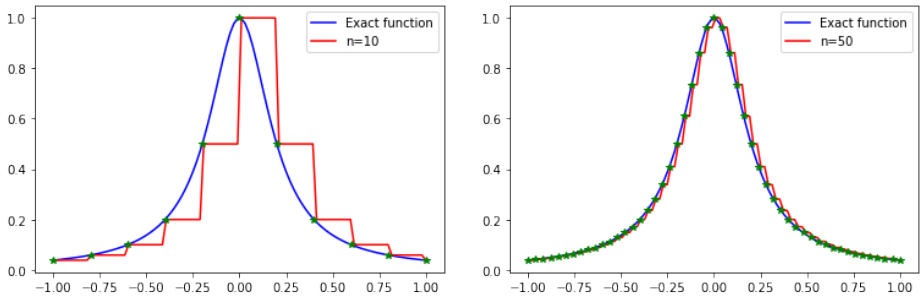


Figure 4 – Heaviside interpolation with equidistant points.

2.3.2. Relu-approximation and quadrature rule

Since Relu functions φ_k are easy to integrate, we can approximate $\int_{[0,1]} f(t)dt$ by replacing f by its Relu-approximation from proposition (2.1) to get the following integration quadrature rule :

$$\int_0^1 f(x)dx \simeq \frac{1}{2n}\sum_{k=0}^{n-1}\alpha_k(k+1)^2 + \alpha_n, \tag{3}$$

We denote by

$$I_n(f) = \int_0^1 S_n(x)dx = \frac{1}{2n}\sum_{k=0}^{n-1}\alpha_k(k+1)^2 + \alpha_n.$$

Replacing α_k by its expression and simplifying, we obtain

$$\int_0^1 f(x)dx \simeq I_n(f) = \frac{1}{2n}\left[f(0) + 2f(\frac{1}{n}) + 2f(\frac{2}{n}) + ... + 2f(\frac{k}{n}) + ... + 2f(\frac{n-1}{n}) + f(1)\right]. \tag{4}$$

We retrieve the composite trapezoidal rule for n equidistant intervals. Since (S_n) converges uniformly to f on $[0,1]$, then $(I_n(f))$ converges to $\int_0^1 f(x)dx$.

2.3.3. Finite element application.

Without loss of generality, we can restrict our study to the domain $]0,1[$ and we will consider the following boundary-value problem: Given $f \in L^2(]0,1[)$ and $c \in L^\infty(]0,1[)$, $c \geq 0$, find the function u solving:

$$\begin{cases} -u''(x) + c(x)u(x) &= f(x),\ \forall\, x \in]0,1[\\ u(0) = u(1) &= 0. \end{cases} \tag{5}$$

It is known that (5) is optimality condition of the quadratic minimization problem:

$$\min_{u \in H_0^1(]0,1[} \frac{1}{2}\int_0^1 u'(x)^2dx + \frac{1}{2}\int_0^1 c(x)u(x)^2dx - \int_0^1 f(x)u(x)dx, \tag{6}$$

Clearly, the piecewise C^1 continuous functions $(\varphi_k : x \mapsto \text{Relu}((k+1) - nx), k = 0, ..., n)$ are a linear independent family of the Sobolev space $H^1(]0,1[)$. For a fixed n, let $V_n = \text{Span}(\varphi_k)_{k=0,...,n}$ and $V_n^0 = \{u \in V_n /\ u(0) = u(1) = 0\}$.

We suppose here that the solution u of (5) is continuous, then $u = \lim_{n\to+\infty}\sum_{k=0}^{n}\alpha_k\ \text{Relu}(k+1-nx)$. The idea of finite element method is to compute $u_n = \sum_{k=0}^{n}\alpha_k\ \text{Relu}(k+1-nx) \in V_n^0$ an approximation of u solution to the approximate minimization problem.

$$\min_{u_n \in V_n^0} \frac{1}{2}\int_0^1 u_n'(x)^2dx + \frac{1}{2}\int_0^1 c(x)u_n(x)^2dx - \int_0^1 f(x)u_n(x)dx, \tag{7}$$

Since we have the Dirichlet conditions $u_n(0) = u_n(1) = 0$, then, α satisfies $\alpha_0 + 2\alpha_1 + 3\alpha_2 + ... + (n+1)\alpha_n = 0$ and $\alpha_n = 0$.

The distributional derivative of u_n is

$$u_n' = \sum_{i=0}^{n} \alpha_i \varphi_i'(x) = -n \sum_{i=0}^{n} \alpha_i H(i+1-nx).$$

Replacing u_n and u_n' in (7), the approximation problem consists in solving the quadratic constrained problem:

$$\begin{array}{l} \min\limits_{\alpha \in \mathbb{R}^{n+1}} \dfrac{1}{2}(E_n\alpha, \alpha) - (b_n, \alpha) \\ \alpha_0 + 2\alpha_1 + ... + n\alpha_{n-1} + (n+1)\alpha_n = 0 \\ \alpha_n = 0 \end{array} , \tag{8}$$

where $(.,.)$ designates the inner product, the matrix E_n and the vector b_n are given by :

$$E_n = \left(\int_{]0,1[} \varphi_i'(x)\varphi_j'(x) + c(x)\varphi_i(x)\varphi_j(x)dx \right)_{0 \le i,j \le n} \text{ and } b_n = \left(\int_{]0,1[} f(x)\varphi_i(x)dx \right)_{0 \le i \le n}.$$

Equality constraints raise from boundary conditions $u(0) = 0$ and $u(1) = 0$.

Example 2.5 Consider the simple homogeneous Laplace problem on one dimensional space:

$$\left\{ \begin{array}{lcl} -u''(x) & = & 1, \ \forall \ x \ \in]0,1[\\ u(0) = u(1) & = & 0 \end{array} \right. \tag{9}$$

Since we have, for $i \le j$,

$$\int_0^1 \varphi_i'(x)\varphi_j'(x)dx = n^2 \int_0^1 H(i+1-nx)H(j+1-nx)dx = n^2 \int_0^{\frac{i+1}{n}} dx = n(i+1),$$

then, the associated symmetric matrix E_n is given by

$$E_n = n \begin{pmatrix} 1 & 1 & 1 & ... & ... & ... & 1 \\ 1 & 2 & 2 & ... & ... & ... & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & ... & \vdots \\ 1 & 2 & 3 & ... & ... & n & n \\ 1 & 2 & 3 & ... & ... & n & n+1 \end{pmatrix}. \tag{10}$$

and the vector $b_n = ((i+1)^2/2n)_{0 \le i \le n}$.

The symmetric matrix E_n is invertible and its inverse is the classical matrix :

$$E_n^{-1} = \frac{1}{n}\begin{pmatrix} 2 & -1 & 0 & \dots & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & -1 & 2 & -1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & \dots & 0 & -1 & 1 \end{pmatrix}.$$

Moreover, clearly E_n^{-1} is positive definite, then E_n is also positive definite. Therefore, the quadratic problem (8) has a unique solution α.

In this work we do not study convergence of this approximation method. However, we compare numerical results for $n = 10$ and $n = 50$ with the exact solution $u(x) = \frac{1}{2}(x - x^2)$. Since we know explicitly the inverse of the matrix E_n, resolution of quadratic problem (8) using Karush Kuhn Tucker condition gives the exact solution to this example. Numerical results for $n = 10$ are good with approximation error $\|u - u_{10}\|_\infty = \max_{0\leq i\leq n} |u(\frac{i}{n}) - u_n(\frac{i}{n})| = 0.009082951015014192$. They are better for $n = 50$ with approximation error $\|u - u_{50}\|_\infty = 0.0003633180406232603$.

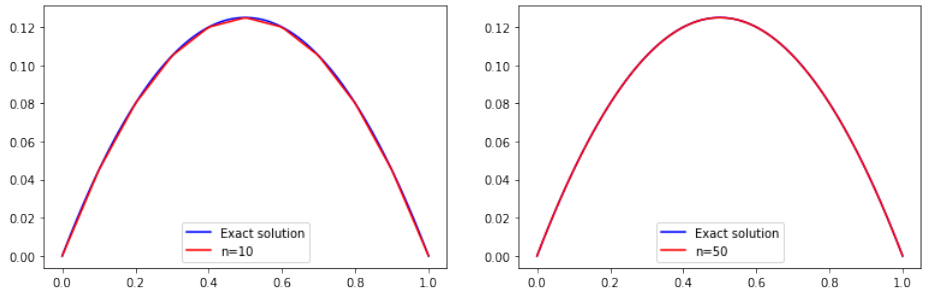


Figure 5 – Relu approximation for finite element method.

2.4. Neural network with one hidden layer

Proposition (2.1) and remark (2.2) are nothing else than a proof of the universal approximation theorem for univariate function. It proves the existence of a neural network of one hidden layer with Relu or Heaviside activation function that approximates a continuous function f. But, in practice, we just know values (y_i) of f on some data (x_i). Interpolation of type detailed in proposition (2.3) can be applied under the condition on training set satisfying $x_i \neq x_j$, for $i \neq j$. Loss error of the neural network is zero, which leads to overfitting if we test it on a data $x_{test} \notin [\min_i x_i, \max_i x_i]$. This makes this type of approximation not beneficial for neural network regression or classification problems.

2.5. Quadratic Relu approximation

We suppose here that f is $C^1([0,1])$. We consider the Relu-approximation of the continuous f' of the form

$$f'(x) = \lim_{n\to+\infty} \sum_{k=0}^{n} \alpha_k \text{ Relu}(k+1-nx)$$

where the vector $\alpha = A_{R,n}^{(1)} G_n^{(1)}$, where $A_{R,n}^{(1)}$ is the matrix (1) and

$$G_n^{(1)} = \left(f'(0), f'(\tfrac{1}{n}), ..., f'(\tfrac{n-1}{n}), f'(1)\right)^T.$$

We obtain the following Relu_2 approximation of f.

Proposition 2.6 The function f is a uniform limit of (S_n), where

$$S_n(x) = -\frac{1}{2n}\sum_{k=0}^{n} \alpha_k \text{ Relu}(k+1-nx)^2 + f(1) + \frac{\alpha_n}{2n} = -\frac{1}{n}\sum_{k=0}^{n} \alpha_k \text{ Relu}_2(k+1-nx) + f(1) + \frac{\alpha_n}{2n}.$$

Proof. It is clear that (S_n) is differentiable on $[0,1]$ and

$$S_n'(x) = \sum_{k=0}^{n} \alpha_k \text{ Relu}(k+1-nx), \ \forall\, x \in [0,1].$$

Then, according to proposition (2.1), (S_n') converges uniformly to f' on $[0,1]$. Moreover, $S_n(1) = -\frac{\alpha_n}{2n} + f(1) + \frac{\alpha_n}{2n} = f(1)$ converges to $f(1)$. The uniform convergence of the sequence of derivatives plus the convergence of the sequence of functions at $x = 1$ imply uniform convergence of (S_n) to f. ■

The Relu_2 approximation interpolates f' at $(\frac{i}{n})_{i=0,...,n}$ and f just at the point 1.
In figure (6), we have used Relu_2 approximation for the classical Runge function $f(x) = \frac{1}{25(2x-1)^2+1}$, for $x \in [0,1]$ and $n = 20$.

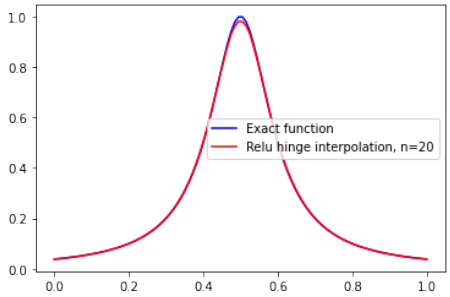


Figure 6 – Relu_2 approximation for n=20

3. The two dimensional case

We study next approximation results for the case of a two-variables function $f : [0,1]^2 \to \mathbb{R}, x = (x_1, x_2) \to f(x)$. We prove that f can be approximated by a sum of separable variables Heaviside or Relu functions.

For this end, we denote by $A \bigotimes B$ the Kronecker product of two matrices A and B.

3.1. Relu-approximation

We study now the following Relu-approximation of f.

Proposition 3.1 For $x = (x_1, x_2) \in [0,1]^2$ and $m = (n+1)^2$, let

$$S_n(x) = \sum_{k,k'=0}^{n} \alpha_{k,k'} \text{ Relu}(k+1-nx_1) \text{ Relu}(k'+1-nx_2),$$

where the vector $\alpha = \left(\alpha_0, \alpha_1, ..., \alpha_m\right)^T = A_{R,n}^{(2)} F_n^{(2)}$. The matrix $A_{R,n}^{(2)}$ of size $m = (n+1)^2$ is given by

$$A_{R,n}^{(2)} = \begin{pmatrix} A_{R,n}^{(1)} & -2A_{R,n}^{(1)} & A_{R,n}^{(1)} & O & \dots & \dots & O \\ O & A_{R,n}^{(1)} & -2A_{R,n}^{(1)} & A_{R,n}^{(1)} & \ddots & & \vdots \\ O & O & A_{R,n}^{(1)} & -2A_{R,n}^{(1)} & A_{R,n}^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & O \\ O & \ddots & \ddots & O & A_{R,n}^{(1)} & -2A_{R,n}^{(1)} & A_{R,n}^{(1)} \\ O & & \ddots & \ddots & O & A_{R,n}^{(1)} & -2A_{R,n}^{(1)} \\ O & \dots & \dots & \dots & \dots & O & A_{R,n}^{(1)} \end{pmatrix} = A_{R,n}^{(1)} \bigotimes A_{R,n}^{(1)},$$

with O is null matrix, $A_{R,n}^{(1)}$ is the $(n+1)$ square matrix of type (1) and the vector $F_n^{(2)} \in \mathbb{R}^m$ defined by:

$$\forall\, 0 \le k, k' \le n, \;\; F_n^{(2)}((n+1)k' + k + 1) = f(\frac{k}{n}, \frac{k'}{n}).$$

Then, the sequence (S_n) converges uniformly to f on $[0,1]^2$.

Proof. The function f is continuous on the compact domain $[0,1]^2$, then it is bounded and uniformly continuous. We denote by $M = \sup_{x \in [0,1]^2} |f(x)|$ and for a given $\varepsilon > 0$, there exists $\eta > 0$ such that for all $x, x' \in [0,1]^2$ satisfying $\|x - x'\|_\infty \le \eta$, we have

$$|f(x) - f(x')| \le \frac{\varepsilon}{4}.$$

Fixing $n \geq n_0 = E(\frac{1}{\eta}) + 1$ and $x = (x_1, x_2) \in [0,1]^2$. If $x_1 = 1$ or $x_2 = 1$, we prove, similarly to the one dimensional case, that $|S_n(x) - f(x)| \leq \varepsilon$. If $x \in [0,1[^2$, then, there exist $0 \leq k_x \leq n$ and $0 \leq k'_x \leq n$ such that : $x \in [\frac{k_x}{n}, \frac{k_x+1]}{n}] \times [\frac{k'_x}{n}, \frac{k'_x+1]}{n}]$ and

$$S_n(x) = \sum_{k'=k'_x}^{n} \sum_{k=k_x}^{n} \alpha_{k,k'}(k+1-nx_1)(k'+1-nx_2).$$

By replacing $\alpha_{k,k'}$ and simplifying, we get

$$\begin{array}{rl} S_n(x) & = f(\frac{k_x}{n}, \frac{k'_x}{n})(k_x+1-nx_1)(k'_x+1-nx_2) - f(\frac{k_x+1}{n}, \frac{k'_x}{n})(k_x-nx_1)(k'_x+1-nx_2) \\ & -f(\frac{k_x}{n}, \frac{k'_x+1}{n})(k_x+1-nx_1)(k'_x-nx_2) + f(\frac{k_x+1}{n}, \frac{k'_x+1}{n})(k_x-nx_1)(k'_x-nx_2) \end{array}.$$

or

$$\begin{array}{rl} S_n(x) & = \left[f(\frac{k_x}{n}, \frac{k'_x}{n}) - f(\frac{k_x+1}{n}, \frac{k'_x}{n})\right](k_x-nx_1)(k'_x+1-nx_2) + f(\frac{k_x}{n}, \frac{k'_x}{n})(1+k'_x-nx_2) \\ & + \left[f(\frac{k_x+1}{n}, \frac{k'_x+1}{n}) - f(\frac{k_x}{n}, \frac{k'_x+1}{n})\right](k_x-nx_1)(k'_x-nx_2) - f(\frac{k_x}{n}, \frac{k'_x+1}{n})(k'_x-nx_2) \end{array}.$$

Then

$$\begin{array}{rl} S_n(x) - f(x) & = \left[f(\frac{k_x}{n}, \frac{k'_x}{n}) - f(\frac{k_x+1}{n}, \frac{k'_x}{n})\right](k_x-nx_1)(k'_x+1-nx_2) \\ & + \left[f(\frac{k_x+1}{n}, \frac{k'_x+1}{n}) - f(\frac{k_x}{n}, \frac{k'_x+1}{n})\right](k_x-nx_1)(k'_x-nx_2) \\ & \left[f(\frac{k_x}{n}, \frac{k'_x}{n}) - f(\frac{k_x}{n}, \frac{k'_x+1}{n})\right](k'_x-nx_2) \\ & + \left[f(\frac{k_x}{n}, \frac{k'_x}{n}) - f(x)\right] \end{array}.$$

Using uniform continuity, and since, $\|x - (\frac{k_x}{n}, \frac{k'_x}{n})\|_\infty \leq \frac{1}{n} \leq \eta$, and $(k'_x - nx_2)$, $(k_x - nx_1), (k'_x + 1 - nx_2)$ and $(k_x + 1 - nx_1)$ are bounded by 1, then

$$|S_n(x) - f(x)| \leq \frac{\varepsilon}{4} + \frac{\varepsilon}{4} + \frac{\varepsilon}{4} + \frac{\varepsilon}{4} = \varepsilon.$$

Uniform convergence then holds. ■

Remark 3.2 We prove later in (4.1) that f is also uniform limit of

$$\sum_{k,k'=0}^{n} \alpha_{k,k'} H(k+1-nx_1)H(k'+1-nx_2),$$

where the vector $\alpha = \big(\alpha_0, \alpha_1, ..., \alpha_m\big)^T = A^{(2)}_{H,n} F^{(2}_n$. The matrix $A^{(2)}_{H,n}$ of size $m = (n+1)^2$ is given by

$$A^{(2)}_{H,n} = \begin{pmatrix} A^{(1)}_{H,n} & -A^{(1)}_{H,n} & O & O & \dots & \dots & O \\ O & A^{(1)}_{H,n} & -A^{(1)}_{H,n} & O & \ddots & & \vdots \\ O & O & A^{(1)}_{H,n} & -A^{(1)}_{H,n} & O & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & O \\ O & \ddots & \ddots & O & A^{(1)}_{H,n} & -A^{(1)}_{R,n} & O \\ O & & \ddots & \ddots & O & A^{(1)}_{H,n} & -A^{(1)}_{H,n} \\ O & \dots & \dots & \dots & \dots & O & A^{(1)}_{H,n} \end{pmatrix} = A^{(1)}_{H,n} \bigotimes A^{(1)}_{H,n}$$

for the $(n+1)$ square matrices, O null matrix and

$$A^{(1)}_{H,n} = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1 & -1 & 0 & \ddots & & \vdots \\ 0 & 0 & 1 & -1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 & 1 & -1 & 0 \\ 0 & & \ddots & \ddots & 0 & 1 & -1 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix}. \tag{11}$$

And the vector $F^{(2)}_n \in \mathbb{R}^m$ is defined by:

$$\forall\ 0 \le k, k' \le n, \quad F^{(2)}_n((n+1)k' + k + 1) = f(\frac{k}{n}, \frac{k'}{n}).$$

3.2. Applications

As for the one dimensional case, we consider some applications presented to interpolation, numerical integration and finite element.

3.2.1. Interpolation

It is easy to check that, for fixed n, the approximated function S_n of a given continuous function f on $[0,1]^2$ cited in proposition (3.1) or in remark (3.2) is such that :

$$S_n(\frac{k}{n}, \frac{k'}{n}) = f(\frac{k}{n}, \frac{k'}{n}),\ \forall\ 0 \le k, k' \le n.$$

In fact, coefficients $\alpha_{k,k'}$ are defined in a way that interpolation conditions are satisfied. This is equivalent to solving the linear system : $B^{(2)}_n \alpha = F^{(2)}_n$, with α and $F^{(2)}_n$ are

the same vectors in proposition (3.1) and remark (3.2). The square matrix of size $m = (n+1)^2$ is

$$B_n^{(2)} = \begin{cases} B_{H,n}^{(2)} & \text{for Heaviside approximation} \\ B_{R,n}^{(2)} & \text{for Relu approximation} \end{cases}$$

with $B_{H,n}^{(2)}, B_{R,n}^{(2)}$ are the following upper triangular matrices :

$$B_{H,n}^{(2)} = \begin{pmatrix} B_{H,n}^{(1)} & B_{H,n}^{(1)} & B_{H,n}^{(1)} & \dots & \dots & \dots & B_{H,n}^{(1)} \\ O & B_{H,n}^{(1)} & B_{H,n}^{(1)} & B_{H,n}^{(1)} & \cdots & & B_{H,n}^{(1)} \\ O & O & B_{H,n}^{(1)} & B_{H,n}^{(1)} & B_{H,n}^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ O & \ddots & \ddots & O & B_{H,n}^{(1)} & B_{H,n}^{(1)} & B_{H,n}^{(1)} \\ O & & \ddots & \ddots & O & B_{H,n}^{(1)} & B_{H,n}^{(1)} \\ O & \dots & \dots & \dots & \dots & O & B_{H,n}^{(1)} \end{pmatrix} = B_{H,n}^{(1)} \bigotimes B_{H,n}^{(1)},$$

where:

$$B_{H,n}^{(1)} = \begin{pmatrix} 1 & 1 & 1 & \dots\dots & \dots & \dots & 1 \\ 0 & 1 & 1 & 1 & \ddots & & 1 \\ 0 & 0 & 1 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 & 1 & 1 & 1 \\ 0 & & \ddots & \ddots & 0 & 1 & 1 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix}.$$

$$B_{R,n}^{(2)} = \begin{pmatrix} B_{R,n}^{(1)} & 2B_{R,n}^{(1)} & 3B_{R,n}^{(1)} & \dots & \dots & \dots & (n+1)B_{R,n}^{(1)} \\ O & B_{R,n}^{(1)} & 2B_{R,n}^{(1)} & 3B_{R,n}^{(1)} & \cdots & & nB_{R,n}^{(1)} \\ O & O & B_{R,n}^{(1)} & 2B_{R,n}^{(1)} & 3B_{R,n}^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ O & \ddots & \ddots & O & B_{R,n}^{(1)} & 2B_{R,n}^{(1)} & 3B_{R,n}^{(1)} \\ O & & \ddots & \ddots & O & B_{R,n}^{(1)} & 2B_{R,n}^{(1)} \\ O & \dots & \dots & \dots & \dots & O & B_{R,n}^{(1)} \end{pmatrix} = B_{R,n}^{(1)} \bigotimes B_{R,n}^{(1)},$$

and:

$$B_{R,n}^{(1)} = \begin{pmatrix} 1 & 2 & 3 & \dots\dots & \dots & \dots & (n+1) \\ 0 & 1 & 2 & 3 & \ddots & & n \\ 0 & 0 & 1 & 2 & 3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 & 1 & 2 & 3 \\ 0 & & \ddots & \ddots & 0 & 1 & 2 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix}$$

It is easy to check, in both cases, that $B_n^{(2)}$ is the inverse of $A_n^{(2)}$ which confirms the interpolation result.
As example, for the parabolic function $f(x_1, x_2) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$, we present in figure (7) the exact function and its Relu interpolation for $n = 20$.

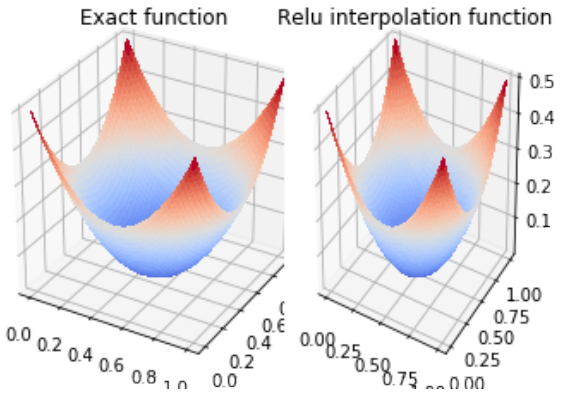


Figure 7 – Relu approximation for n=20

Uniform convergence avoids Runge's phenomenon even in the two-dimensional case. Numerical results in figure (8) are obtained with Runge function f defined on $[0,1]^2$ by $f(x_1, x_2) = \frac{1}{(1+25(2x_1-1)^2)(1+25(2x_2-1)^2)}$.

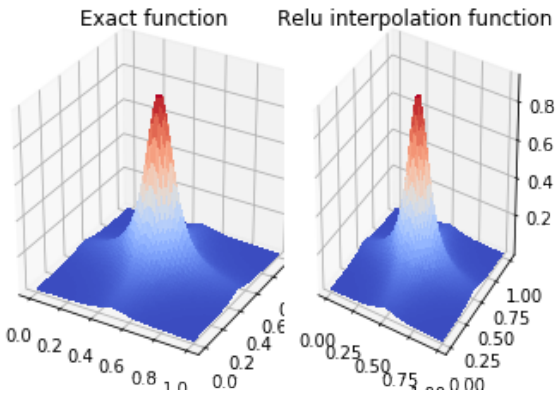


Figure 8 – Relu approximation for n=20

3.3. Finite element application

We denote by $\Omega =]0,1[^2$. Given $f \in L^2(\Omega)$, find the function u solving:

$$\begin{cases} -\Delta u(x) &= f(x), \ \forall\, x \ \in \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{cases} \tag{12}$$

u solution (12) is also solution to the quadratic optimization problem

$$u = \arg\min_{v \in H_0^1(\Omega)} \tfrac{1}{2} \int_\Omega \|\nabla v(x)\|^2 \, dx - \int_\Omega f(x) v(x) dx.$$

Let $m = (n+1)^2$, and $\varphi : x \mapsto \varphi_k(x) =$ Relu$(k+1-nx)$. In order to enumerate the set $(\varphi_k \varphi_{k'})_{k,k'=0,\dots,n}$ of the Sobolev space $H^1(\Omega)$, we consider the family $(\psi)_{i=1,\dots,m} = (\varphi_k \varphi_{k'})_{k,k'=0,\dots,n}$. We denote by $V_n = \text{Span}(\varphi_{k,k'})_{k,k'=0,\dots,n} = \text{Span}(\psi_i)_{i=1,\dots,m}$ and

$$V_n^0 = \{v \in V_n;\ v = 0 \text{ on } \partial\Omega\}.$$

We consider then the approximate quadratic optimization problem

$$u_n = \arg\min_{v_n \in V_n^0} \tfrac{1}{2} \int_\Omega \|\nabla v_n(x)\|^2 \, dx - \int_\Omega f(x) v_n(x) dx.$$

If we write $v_n = \sum_{i=1}^{m} \alpha_i \psi_i$, the last problem becomes equivalent to the quadratic constrained problem:

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}^m} \frac{1}{2}(E_n^{(2)}\alpha, \alpha) - (b_n^{(2)}, \alpha), \\ &u_n = \sum_{i=1}^{m} \alpha_i \psi_i = 0 \text{ on } \partial\Omega, \end{aligned} \tag{13}$$

where the matrix $E_n^{(2)} = (\int_\Omega \psi_i'(x)\psi_j'(x)dx)_{i,j=1,...,m}) = E_n \bigotimes E_n$, for E_n given in (10), and the vector $b_n^{(2)} = (\int_\Omega f(x)\psi_i(x)dx)_{i=1,...,m}$.

For the particular example

$$\begin{cases} -\Delta u(x) &= 1, \ \forall \ x \ \in \Omega, \\ u &= 0 \quad \text{on } \partial\Omega \end{cases} , \tag{14}$$

the exact solution is $u(x_1, x_2) = \frac{1}{4}x_1(1-x_1)x_2(1-x_2)$ and figure (9) shows numerical solution using Relu approximations for $n = 20$ for which $\|u-u_n\|_\infty = 0.003246376497983063$. To solve the minimizing quadratic problem (13), we used the function 'solvers.qp' of the python software package 'cvxopt'.

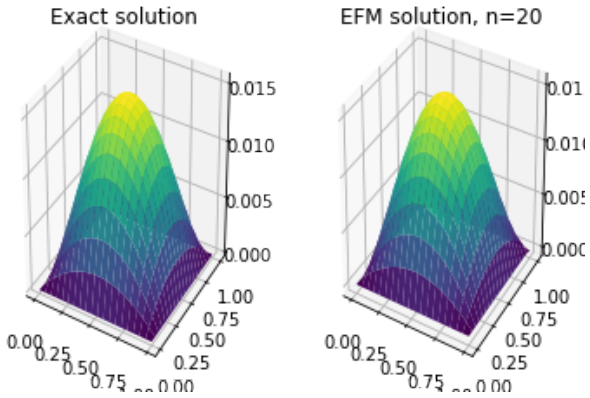


Figure 9 – EFM with Relu approximation for n=20

4. Multi-dimensional case

For $d \in \mathbb{N}^*$, $\Omega = [0,1]^d$ and $f : \Omega \to \mathbb{R}, x = (x_1, ..., x_d) \to \mathbb{R}$ is a continuous function, we denote by the vector $F_n^{(d)}$ of dimension $m = (n+1)^d$, for $n > 1$, representing values of the function f on vertices of decomposition of the multidimensional cubic Ω :

$$F_n^{(d)} = (f(\frac{k_1}{n}, \frac{k_2}{n}, ..., \frac{k_d}{n}))_{0 \le k_1,...,k_d \le n}.$$

4.1. Heaviside -approximation

Proposition 4.1 The function f is a uniform limit of

$$S_n(x) = \sum_{0 \le k_1,k_2,...,k_d \le n} \alpha_{k_1,k_2,...,k_d} H(1+k_1-nx_1)H(1+k_2-nx_2)...H(1+k_d-nx_d),$$

with $\alpha = A_{H,n}^{(d)} F_n^{(d)} \in \mathbb{R}^m$, where $A_{H,n}^{(d)}$ is the $m = (n+1)^d$ square matrix given by recurrence as

$$A_{H,n}^{(d)} = A_{H,n}^{(1)} \bigotimes A_{H,n}^{(d-1)}.$$

The matrix $A_{H,n}^{(d)}$ is the inverse of the interpolation matrix $B_{H,n}^{(d)}$ which is of the form

$$B_n^{(d)} = B_{H,n}^{(1)} \bigotimes B_{H,n}^{(d-1)}.$$

The matrices $A_{H,n}^{(1)}$ and $B_{H,n}^{(1)}$ are the same of those of the bi-dimensional case in section (3).

$$A_{H,n}^{(d)} = \begin{pmatrix} A_{H,n}^{(d-1)} & -A_{H,n}^{(d-1)} & O & O & \dots & \dots & O \\ O & A_{H,n}^{(d-1)} & -A_{H,n}^{(d-1)} & O & \ddots & & \vdots \\ O & O & A_{H,n}^{(d-1)} & -A_{H,n}^{(d-1)} & O & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & O \\ O & \ddots & \ddots & O & A_{H,n}^{(d-1)} & -A_{H,n}^{(d-1)} & O \\ O & & \ddots & \ddots & O & A_{H,n}^{(d-1)} & -A_{H,n}^{(d-1)} \\ O & \dots & \dots & \dots & \dots & O & A_{H,n}^{(d-1)} \end{pmatrix} = A_{H,n}^{(1)} \bigotimes A_{H,n}^{(d-1)},$$

O the null matrix and the $(n+1)$ square matrix $A_{H,n}^{(1)}$ is of the form (11).

In order to prove the last proposition we establish the following general property.

Lemma 4.2 Let $d \in \mathbb{N}^*$, $p = n+1$ and $F = (F_1, F_2, ..., F_{p^d}) \in \mathbb{R}^{p^d}$. If $\alpha = A_{H,n}^{(d)} F = (\alpha_1, ..., \alpha_{p^d})$, then for all $k \geq 1$, we have

$$\sum_{i=k}^{p^d} \alpha_i = F_k. \tag{15}$$

Proof.

Without loss of generality and in order to simplify notations, we prove the lemma for $k = 1$.

The proof is by induction on d. If $d = 1$, we have easily

$$\sum_{i=1}^{p} \alpha_i = \sum_{i=1}^{n} (A_{H,n}^{(1)} F)_i = \sum_{i=k}^{n} (F_i - F_{i+1}) + F_{n+1} = F_1.$$

We suppose that (15) is true for $d \geq 1$, we wish to prove that is true for $d+1$?

Let $\alpha = A_{H,n}^{(d+1)} F \in \mathbb{R}^{p^{d+1}}$, with the block decomposition of α and F: $\alpha = (\alpha^1, \alpha^2, ..., \alpha^p)$ and $F = (F^1, F^2, ..., F^p) \in \mathbb{R}^{p^{d+1}}$, $\alpha^k, F^k \in \mathbb{R}^{p^d}$, for $k = 1, ..., p$. Then

$$\alpha = A_{H,n}^{(1)} \bigotimes A_{H,n}^{(d)} = \begin{pmatrix} A_{H,n}^{(d)} F^1 - A_{H,n}^{(d)} F^2 \\ A_{H,n}^{(d)} F^2 - A_{H,n}^{(d)} F^3 \\ \vdots \\ A_{H,n}^{(d)} F^{p-1} - A_{H,n}^{(d)} F^p \\ A_{H,n}^{(d)} F^p \end{pmatrix}.$$

It follows that :

$$\sum_{i=1}^{p^{d+1}} \alpha_i = \sum_{k=1}^{p} \sum_{k'=1}^{p^d} \alpha_{k'}^k = \sum_{k=1}^{p-1} \left[\sum_{k'=1}^{p^d} (A_{H,n}^{(d)} F^k)_{k'} - \sum_{k'=1}^{p^d} (A_{H,n}^{(d)} F^{k+1})_{k'} \right] + \sum_{k'=1}^{p^d} (A_{H,n}^{(d)} F^p)_{k'}$$

Using induction hypothesis, we have $\sum_{k'=1}^{p^d} (A_{H,n}^{(d)} F^k)'_k = F_1^k$ and then

$$\sum_{i=1}^{p^{d+1}} \alpha_i = \sum_{k=1}^{p-1} (F_1^k - F_1^{k+1}) + F_1^p = F_1^1 = F_1.$$

■

Proof. (**of proposition** (4.1))

The proof is analogous to the one of the case $d = 1$. Let $\varepsilon > 0$ and $\eta > 0$, such that, for $x, x' \in \Omega$ satisfying $\|x - x'\|_\infty \leq \eta$, we have $|f(x) - f(x')|_\infty \leq \varepsilon$. Let $x \in \Omega$, without loss of generality, we suppose that $x \in [0, 1[^d$. Let $k_{1x}, k_{2x}, ..., k_{dx}$ such that $x \in [\frac{k_{1x}}{n}, \frac{k_{1x}+1}{n}] \times [\frac{k_{2x}}{n}, \frac{k_{2x}+1}{n}] \times ... \times [\frac{k_{dx}}{n}, \frac{k_{dx}+1}{n}]$. Replacing $S_n(x)$ and applying lemma (4.2), we deduce

$$S_n(x) - f(x) = \sum_{k_1 \geq k_{1x}, k_2 \geq k_{2x}, ..., k_d \geq k_{dx}} \alpha_{k_1, k_2, ..., k_d} = f(\frac{k_{1x}}{n}, \frac{k_{2x}}{n}, ..., \frac{k_{dx}}{n}) - f(x).$$

Using uniform continuity, and since, $\|x - (\frac{k_{1x}}{n}, \frac{k_{2x}}{n}, ..., \frac{k_{dx}}{n})\|_\infty \leq \frac{1}{n} \leq \eta$, then

$$|S_n(x) - f(x)| \leq \varepsilon.$$

This finishes the proof of the proposition (4.1). ■

4.2. Relu -approximation

We can prove similarly that a continuous function f on the compact domain $[0, 1]^d$ is a uniform limit of

$$S_n(x) = \sum_{0 \leq k_1, k_2, ..., k_d \leq n} \alpha_{k_1, k_2, ..., k_d} \text{ Relu}(1+k_1-nx_1) \text{ Relu}(1+k_2-nx_2) ... \text{ Relu}(1+k_d-nx_d),$$

where $\alpha = A_{R,n}^{(d)} F_n^{(d)} \in \mathbb{R}^{(n+1)^d}$, $A_n^{(d)}$ is the $m = (n+1)^d$ square matrix given by recurrence as $A_{R,n}^{(d)} = A_{R,n}^{(1)} \bigotimes A_{R,n}^{(d-1)}$. The matrix $A_{R,n}^{(d)}$ is the inverse of the interpolation matrix $B_{R,n}^{(d)}$ which is of the form $B_{R,n}^{(d)} = B_{R,n}^{(1)} \bigotimes B_{R,n}^{(d-1)}$, where $A_{R,n}^{(1)}$ and $B_{R,n}^{(1)}$ are the same matrices used before in section (3).

5. Other activation functions

If we replace the Relu function by another activation function, like sigmoid or softplus, determining the coefficients α is not explicit and we need to solve a linear system for every choice of n. Numerical results are similar to Relu approximation case. This is an example for Runge function for $d = 1$ with softplus function then for $d = 2$ with sigmoid activation function:

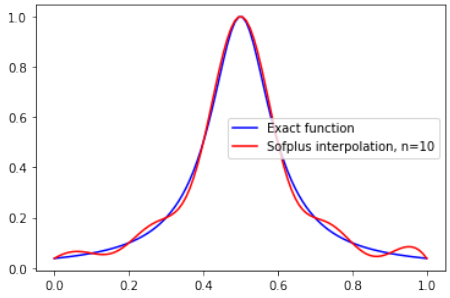


Figure 10 – Softplus approximation for d=1 and n=10

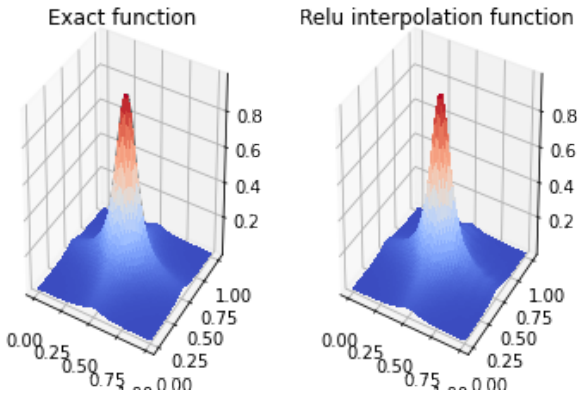


Figure 11 – Sigmoid approximation for d=2 and n=20

6. Conclusion

In this paper, we have considered the following approximation problem : given a continuous function f, we proved that f is a uniform limit of a sum of Relu or heaviside activation functions with separable variables. Numerical applications of these approximations for one and two dimensional cases were implemented.

As future work, we plan to extend these approximations using other activation functions and try to apply them to machine learning models like regression or deep learning models.

7. References

[1] G. CYBENTO, " Approximation by Superpositions of a Sigmoidal Function ", Springer-Varleg, 1989.

[2] B. LIU, Y. LIANG, "Optimal Function Approximation with Relu Neural Networks", Computer Science , 2019

[3] P. PETERSEN, F. VOIGTLAENDER, " Optimal approximation of piecewise smooth functions using deep Relu neural networks ", Neural Networks, vol. 108, num. pages 296-330, 2018

[4] R. KRESS, " Numerical Analysis ", Springer, 1991

[5] J M. KLUSOWSKI, A R. BARRON, " Approximation by Combinations of Relu and squared Relu Ridge Functions With 1 and 0 Controls", IEEE Transactions on Information Theory, vol. 12, December 2018

[6] D. YAROTSKY, "Optimal approximation of continuous functions by very deep Relu networks", Proceedings of Machine Learning Research v: 31st Annual Conference on Learning Theory, vol. 75, pages 1-11, 2018

[7] M. LESHNO, V. LIN, A. PINKUS, S. SCHOKEN , "Multilayer Feedforward Networks With a Nonpolynomial Activation Function Can Approximate Any Function ", Neural Networks, vol. 6, pages 861-867, 1993

[8] H. PADÉ , " Sur la représentation approchée d'une fonction par des fractions rationnelles ", ASENS, 3e série, vol. 9, 1892

[9] J. HE, L. LI, J. XU, CH. ZHENG , " ReLU Deep Neural Networks and Linear Finite Elements ", J. Comput. Math, vol. 38(3) , pages{ 502-527, 2020

[10] W. E, J. HAN AND A. JENTZEN , " Deep learning-based numerical methods for highdimensional parabolic partial differential equations and backward stochastic differential equations ", Communications in Mathematics and Statistics, vol. 5 , pages 349-380, 2017