Approximation with activation functions and applications

Bessi Radhia

Approximation with activation functions and applications

Avril 2020

Bessi Radhia

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

ABSTRACT. Function approximations 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. We deal with first one and two dimensional cases, then we generalize the approximation to the multi dimensional case. Application of these approximations to: interpolation, numerical integration, finite element and neural network. Some examples will be presented.

RÉSUMÉ. La théorie d'approximations 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 applications 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'appoximation, 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 from the previous layer. Examples of activation functions:

- Heaviside function $H : x \mapsto \begin{cases} 1 & \text{if } \ > 0 \\ 0 & \text{if } \ > 0 \end{cases}$
- Relu function : **Relu** : $x \to \max(0, x)$.
- Relu-Hinge function : $x \to \frac{1}{2}\max(0, x)^2$.
- 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 universal approximation theorem. It was proved in [1] the density of 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 given 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, was established.

An L^∞ and L^2 error bounds for functions of many variables that are approximated by linear combinations of ReLU and squared ReLU ridge 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].

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 function which provides a basic and important tool for the numerical solution of other problems and it can be for example applied in numerical methods for ordinary or partial differential equations and in finite element.

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 universal approximation theorem in the case of single variate. The novelty here is that we precise weights of the network. But, we don't have linearity of the neural network for multi- dimensional case using ReLU 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 a one hidden layer neural network with φ 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

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

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

$$A_n = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -2 & 1 & \ddots & & \vdots \\ \vdots & \ddots & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 & -2 & 1 \\ \vdots & & \ddots & \ddots & \ddots & 1 & -1 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix},$$

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 $\frac{1}{\eta}$), for all $x \in [0,1]$, then there exists $0 \leq k_x \leq n$ such that $x \in [\frac{k_x}{n}, \frac{k_x+1]}{n}]$.

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

Then

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

Since $x \in [\frac{k_x}{n}, \frac{k_x+1]}{n}]$, then $|k_x + 1 - nx| \leq 1$. 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 $\textbf{Relu}_k : x \mapsto \textbf{Relu}((k+1-nx)$ satisfies

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

then the vector $(\alpha_k)_{0 \leq k \leq n}$ is solution to the triangular linear system $A'_n \alpha = F_n$ with

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

It is easy to check that the matrix A'_n is nothing else but the inverse of the triangular matrix A_n.

2) We can prove similarly that f is the uniform limit of $(T_n(x))$, with

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

with $\alpha = A_n^{(1)} F_n^{(1)}$, $F_n^{(1)}$ is the same vector of proposition (2.1) and the matrix

$$A_n^{(1)} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -2 & 1 & \ddots & & \vdots \\ \vdots & \ddots & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 & -2 & 1 \\ \vdots & & \ddots & \ddots & \ddots & 1 & -2 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \tag{1}$$

is the inverse of the following matrix

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

3) On the contrary of polynomial, cubic spline, regression or trigonometric Fourier approximation, ReLU interpolation in the form of (2.3) is explicit and simple to compute for every integer n. Moreover, it is very easy to integrate and to derive.

4) 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

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

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

2.2. Interpolation by ReLU function

Given a data set $(x_i, y_i)_{i=0,\dots,n}$, with $x_i \in \mathbb{R}$ and $y_i \in \mathbb{R}$, we propose here to find a function g of the form $g_n(x) = \sum_{k=0}^{n} \alpha_k \textbf{Relu}(w_k x + b_k)$ satisfying $g_n(x_i) = y_i$.

Proposition 2.3 Let g_n be the function defined by

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

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

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

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

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

The matrix of the last linear system is :

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

This matrix is clearly invertible and its inverse is the quadri-triangular matrix D_n which achieves the proof of this result. ■

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 don't have the uniqueness of the weights w_i and bias b_i. If the points are equispaced, we find the 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 equispaced points on $[-1, 1]$, we meet 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 ReLU approximations. Figures (1) and (2) display numerical results of Runge function interpolation, first over equispaced points, then over Chebychev points. For the same function f, we also test Heaviside approximation whith equispaced points for $n = 10$ and $n = 20$. Results are displayed in figure (3).

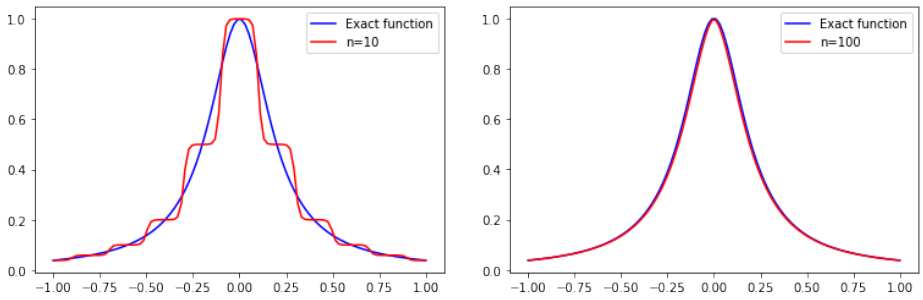

Figure 1. *ReLU interpolation with equispaced points.*

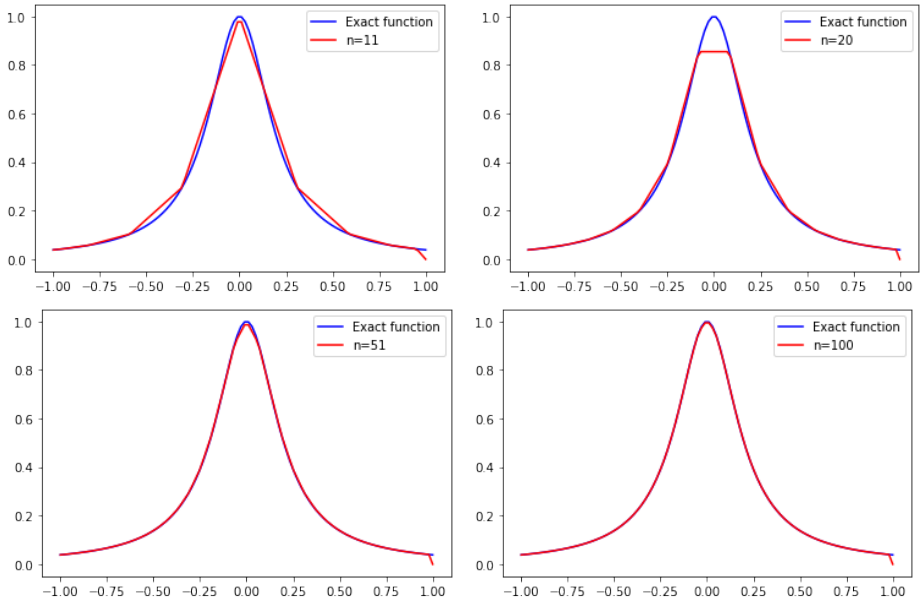

Figure 2. *ReLU interpolation with Chebychev points.*

We notice that over small Chebychev points number, interpolation is not good at the middle of the interval, this is because points of interpolation are located near 1 and -1. We also notice that uniform convergence is faster with ReLU-approximation than Heaviside approximation as (1) and (3) show:

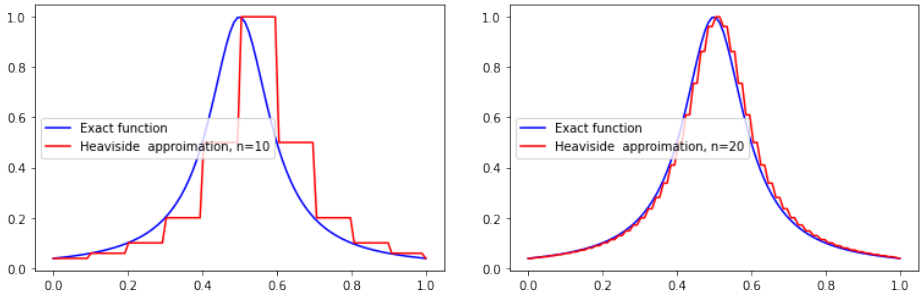

Figure 3. *Heaviside intepolation with equispaced points.*

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 of proposition (2.1) to get the following quadrature rule integration :

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

We denote by

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

Replacing α_k by its expression and simplifying, we obtain

$$\int_0^1 f(x)dx \simeq I_n(f) = \frac{1}{n}\left[f(0) + 2f(\frac{1}{n}) + 2f(\frac{2}{n}) + ... + 2f(\frac{k}{n}) + ... + 2f(\frac{n-1}{n}) + (1-n)f(1)\right]. \tag{3}$$

Clearly, since (S_n) converges uniformly to f on $[0,1]$, then $(I_n(f))$ converges to $\int_0^1 f(x)dx$.
This quadrature rule is exact only on polynomial functions of degree one. Numerical results are very close to those obtained with composite trapezoidal rule for the same n. Table (2.3.2) gives our experimental results for functions defined on $[0,1]$, $x \mapsto e^{-x^2}$ and $x \mapsto \frac{1}{25x^2+1}$ with $n = 10, 20$ and 50.

f	$x \mapsto e^{-x^2}$	$x \mapsto \frac{1}{25x^2+1}$
$I = \int_0^1 f(x)dx$	0.7468241328124271	0.2746801533890032
Composite trapezoidal rule (n=50)	0.7467996071893512	0.2746776880807905
$I_{10}(f)$	0.7462107961317493	0.27462081624602197
$I_{20}(f)$	0.7466708369398734	0.27466475095467746
$I_{50}(f)$	0.7467996071893512	0.2746776880807905

Table 1. *Application of formula* (3)

2.3.3. Finite element application.

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

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

It is easy to prove that the piecewise C^1 continuous functions $(\varphi_k : x \mapsto \mathbf{Relu}((k+1) - nx), k = 0, ..., n)$ is a basis of the Sobolev space $H^1(]0,1[)$.

The variational formulation of problem (8) consists in finding $u \in H^1(]0,1[)$ such that:

$$\int_{]0,1[} u'(x)v'(x)dx + \int_{]0,1[} c(x)v(x)dx = \int_{]0,1[} f(x)v(x)dx, \ \forall\ v \in H_0^1(]0,1[).$$

The last variational formulation is equivalent to find $u_n \in V_n = \text{Span}(\varphi_k)_{k=0,...,n}$, such that:

$$\int_{]0,1[} u_n'(x)v_n'(x)dx + \int_{]0,1[} c(x)u_n(x)v_n(x)dx = \int_{]0,1[} f(x)v_n(x)dx, \ \forall\ v_n \in V_n.$$

This is equivalent to finding $u_n \in V_n$ satisfying:

$$\int_{]0,1[} u_n'(x)\varphi_k'(x)dx + \int_{]0,1[} c(x)\varphi_k(x)dx = \int_{]0,1[} f(x)\varphi_k(x)dx, \ \forall\ k = 0, ..., n.$$

The variational formulation consists in solving the quadratic constrained problem:

$$\begin{array}{l} \min\limits_{\alpha\in\mathbb{R}^n} \dfrac{1}{2}(A_n\alpha,\alpha) - (b_n,\alpha) \\ \alpha_0 + 2\alpha_1 + ... + (n+1)\alpha_n = 0 \\ \alpha_n = 0. \end{array}, \tag{5}$$

where 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 $u(0) = u(1) = 0$.

Consider the simple homogeneous Laplace problem on dimension 1 of space:

$$\left\{\begin{array}{lcl} -u''(x) & = & 1, \ \forall\ x \in]0,1[\\ u(0) = u(1) & = & 0 \end{array}\right. \tag{6}$$

The exact solution u to this problem is: $u(x) = \frac{1}{2}(x - x^2)$. The associated symmetric matrix E_n is given by $E_n = n \begin{pmatrix} 1 & 1 & 1 & ... & ... & ... & 1 \\ 1 & 2 & 2 & ... & ... & ... & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & ... & \vdots \\ 1 & 2 & 3 & ... & ... & n-1 & n-1 \\ 1 & 2 & 3 & ... & ... & n-1 & n \end{pmatrix}$ and the vector $b_n = ((i+1)^2/n)_{0\le i\le n-1}$.

The symmetric matrix E_n is invertible and its inverse is:

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

Moreover, clearly E_n^{-1} is positive definite, then E_n is also positive definite. Therefore, the quadratic problem (5) has a unique solution α.

Resolution of quadratic problem (5) is explicit in this case, numerical results for $n = 10$ are good with approximation error $\|u - u_{10}\|_\infty = 0.009082951015014192$. They are better for $n = 50$ with approximation error $\|u - u_{50}\|_\infty = 0.0003633180406232603$.

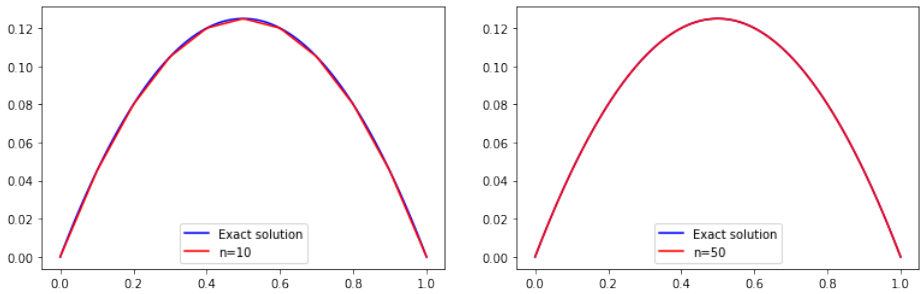

Figure 4. *ReLU approximation for finite element method.*

2.4. Neural network with one hidden layer

Proposition (2.1) and remark (3) are nothing else than a proof of 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 proposition (3.1) 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 over fitting if we test it on a data $x_{test} \notin [\min_i x_i, \max_i x_i]$. This make this type of approximation not beneficial for neural network regression or classification problems.

2.5. ReLU-Hinge approximation

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

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

where here the vector $\alpha = A_n^{(1)} G_n^{(1)}$, for $A_n^{(1)}$ 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 have the following ReLU-Hinge approximation of f.

Proposition 2.5 The function f is a uniform limit of (S_n), where

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

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

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

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.

■

We have used ReLU Hinge approximation for Runge function $f(x) = \frac{1}{25(2x-1)^2+1}$ to obtain (5)

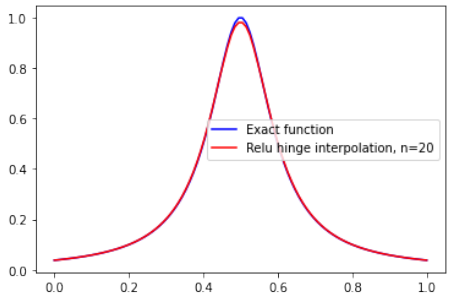

Figure 5. *ReLU-Hinge approximation for n=20*

We notice that, from last numerical results, and for the same integer n, ReLU Hinge approximation is better then ReLU approximation, which is better than Heaviside approximation.

3. The two dimensional case

We have also 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 ReLU functions.

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

3.1. ReLU-approximation

We suppose that $f : [0,1]^2 \to \mathbb{R}$ is a continuous function . As in the one dimensional case, we consider 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

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

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

$$A_n^{(2)} = \begin{pmatrix} A_n^{(1)} & -2A_n^{(1)} & A_n^{(1)} & O_n & \dots & \dots & O_n \\ O_n & A_n^{(1)} & -2A_n^{(1)} & A_n^{(1)} & \ddots & & \vdots \\ O_n & O_n & A_n^{(1)} & -2A_n^{(1)} & A_n^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & O_n \\ O_n & \ddots & \ddots & O_n & A_n^{(1)} & -2A_n^{(1)} & A_n^{(1)} \\ O_n & & \ddots & \ddots & O_n & A_n^{(1)} & -2A_n^{(1)} \\ O_n & \dots & \dots & \dots & \dots & O_n & A_n^{(1)} \end{pmatrix} = A_n^{(1)} \bigotimes A_n^{(1)}$$

for the $(n+1)$ squared matrices, O_n null matrix and

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

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

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

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 $\|x - x'\|_\infty \leq \eta$, we have

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

Fixing $n \geq n_0 = E(\frac{1}{\eta}) + 1$ and $x = (x_1, x_2) \in [0,1]^2$.

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

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

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.

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), is such that :

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

In fact, coefficients $\alpha_{k,k'}$ are such that last interpolation conditions are satisfied. This is equivalent to solving the linear system : $B_n^{(2)}\alpha = F_n^{(2)}$, with α and $F_n^{(2)}$ are the same vectors in proposition (3.1) and the $m = (n+1)^2$ squared matrix $B_n^{(2)}$ is given by :

$$B_n^{(2)} = \begin{pmatrix} B_n^{(1)} & 2B_n^{(1)} & 3B_n^{(1)} & \dots & \dots & \dots & (n+1)B_n^{(1)} \\ O_n & B_n^{(1)} & 2B_n^{(1)} & 3B_n^{(1)} & \dots & & nB_n^{(1)} \\ O_n & O_n & B_n^{(1)} & 2B_n^{(1)} & 3B_n^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ O_n & \ddots & \ddots & O_n & B_n^{(1)} & 2B_n^{(1)} & 3B_n^{(1)} \\ O_n & & \ddots & \ddots & O_n & B_n^{(1)} & 2B_n^{(1)} \\ O_n & \dots & \dots & \dots & \dots & O_n & B_n^{(1)} \end{pmatrix} = B_n^{(1)} \bigotimes B_n^{(1)}.$$

with

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

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

(6) the following results.

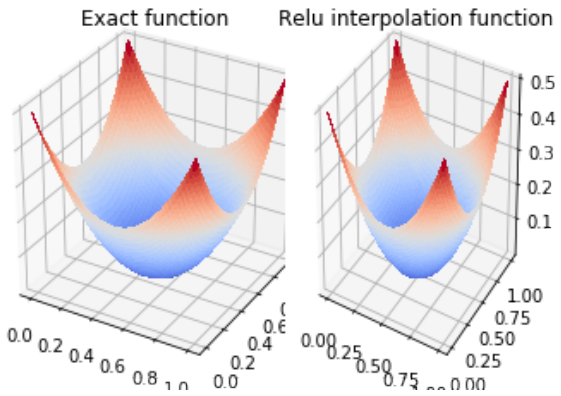

Figure 6. *ReLU approximation for n=20*

Uniform convergence avoids Runge's phenomenon even in the two-dimensional case. Numerical results in figure (7) are obtained with Runge function f defined on $[0,1]^2$ by $f(x_1,x_2) = \frac{1}{(1+25(2x_1-1)^2)(1+25(2x_2-1)^2)}$.

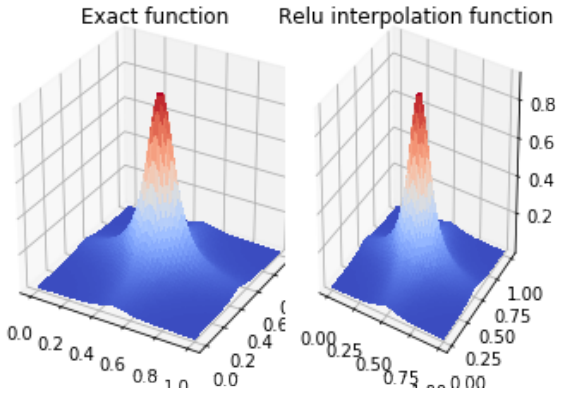

Figure 7. *ReLU approximation for n=20*

3.3. Numerical integration

We can approximate $\int_{[0,1]^2} f(x)dx$ by $\int_{[0,1]^2} S_n(x)dx$ and obtaining the explicit simple quadrature rule :

$$\begin{aligned}\int_{[0,1]^2} f(x)dx &= \sum_{k'=0}^{n}\sum_{k=0}^{n}\alpha_{k,k'}\int_{[0,1]^2}\mathbf{Relu}(k+1-nx_1)\mathbf{Relu}(k'+1-nx_2)dx \\ &= \tfrac{1}{4n^2}\sum_{k'=0}^{n}\sum_{k=0}^{n}\alpha_{k,k'}(k+1)^2(k'+1)^2\end{aligned} \quad . \tag{7}$$

As example, for $f(x_1,x_2) = x_1x_2$, the numerical integration formula (7) gives $I = 0.25 = \int_{[0,1]^2} f(x)dx \simeq 0.26009999999560096$ for $n = 50$. This result for $n = 50$ is not efficient comparing it with simple trapezoidal integration formula which is exact in this case.

3.4. Finite element application

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

$$\left\{\begin{array}{lll} -\Delta u(x) &= & f(x),\ \forall\, x\ \in \Omega \\ u & = 0 & \text{on } \partial\Omega \end{array}\right. \tag{8}$$

Consider $(\varphi_{k,k'} : x \mapsto \mathbf{Relu}((k+1)-nx)\mathbf{Relu}((k'+1)-nx), k,k' = 0,...,n)$ the basis of the Sobolev space $H^1(\Omega)$.

The variational formulation of problem (8) consists in finding $u \in H^1(\Omega)$ such that:

$$\int_\Omega u'(x)v'(x)dx = \int_\Omega f(x)v(x)dx,\ \forall\, v \in H_0^1(\Omega).$$

and the corresponding approximation variational formulation consists in finding $u_n \in V_n = \text{Span}(\varphi_{k,k'})_{k,k'=0,...,n}$, such that:

$$\int_\Omega u_n'(x)v_n'(x)dx = \int_\Omega f(x)v_n(x)dx,\ \forall\, v_n \in V_n.$$

This is equivalent to find $u_n \in V_n$ satisfying:

$$\int_\Omega u_n'(x)\varphi_k'(x)dx = \int_\Omega f(x)\varphi_{k,k'}(x)dx,\ \forall k,k' = 0,...,n-1.$$

The variational formulation consists in solving the quadratic constrained problem:

$$\begin{array}{l}\min\limits_{\alpha\in\mathbb{R}^m}\ \dfrac{1}{2}(E_n^{(2)}\alpha,\alpha) - (b_n,\alpha), \\ u_n = 0 \text{ on } \partial\Omega\end{array} \tag{9}$$

where the matrix A_n and the vector b_n are given by :

$$E_n^{(2)} = (\int_\Omega(\varphi_k'(x)\varphi_{k'}'(x))_{k,k'}, \text{and } b_n = (\int_\Omega f(x)\varphi_k(x)dx)_k.$$

For the particular example

$$\begin{cases} -\Delta u(x) & = & 1, \ \forall \ x \ \in \Omega \\ u & = 0 & \text{on } \partial\Omega \end{cases}, \tag{10}$$

exact solution is $u(x_1, x_2) = \frac{1}{4}x_1(1-x_1)x_2(1-x_2)$ and figure(8) shows numerical solution using ReLU approximations for $n = 20$ for which $\|u-u_n\|_\infty = 0.003246376497983063$. To slove the minimizing quadratic problem (9) we used the function 'solvers.qp' of the python software package 'cvxopt'.

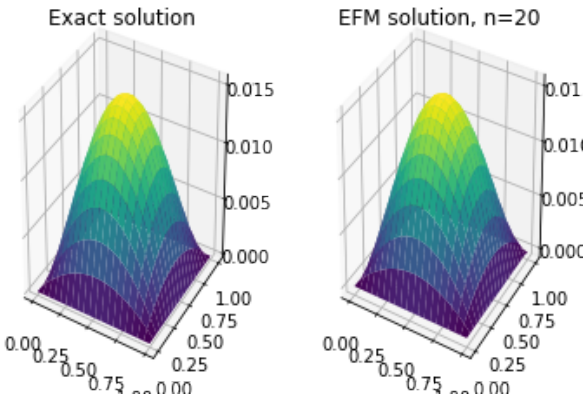

Figure 8. *EFM with ReLU approximation for n=20*

4. Multi-dimensional case

For $d \geq 1$ and $\Omega = [0,1]^d$ and for $f : \Omega \to \mathbb{R}, x = (x_1, ..., x_d) \to \mathbb{R}$ a continuous function, we can prove that f is a uniform limit of

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

with $\alpha \in \mathbb{R}^{(n+1)^d}$ solution to the system $A_n^{(d)}\alpha = F_n^{(d)}$, with $A_n^{(d)}$ is the $(n+1)^d$ squared matrix given by recurrence as

$$A_n^{(d)} = A_n^{(1)} \bigotimes A_n^{(d-1)}.$$

The matrix $A_n^{(d)}$ is the inverse of the interpolation matrix $B_n^{(d)}$ which is of the form

$$B_n^{(d)} = B_n^{(1)} \bigotimes B_n^{(d-1)}.$$

The vector $F_n^{(d)}$ is of dimension $(n+1)^d$ and represents values of the function f on vertices of decomposition of the multidimensional cubic Ω :

$$F(n^{d-1}k_d + n^{d-2}k_{d-1} + ... + nk_2 + k_1) = f(\frac{k_1}{n}, \frac{k_2}{n}, ..., \frac{k_d}{n}), \forall \ 0 \leq k_1, ..., k_d \leq n.$$

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 softtplus function then for $d = 2$ with sigmoid activation function:

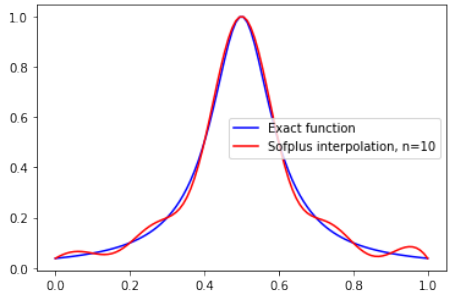

Figure 9. *Softplus approximation for d=1 and n=10*

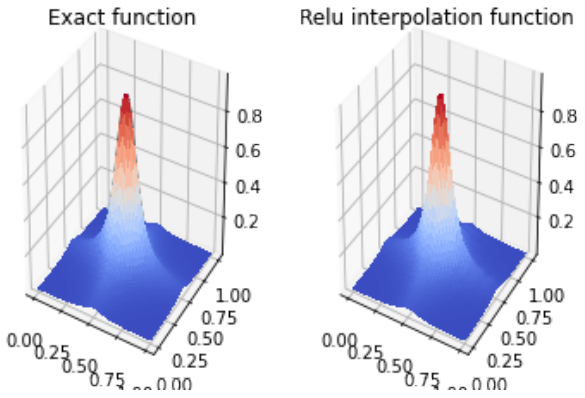

Figure 10. *Sigmoid approximation for d=2 and n=20*

6. Conclusion

In this paper, we have considered the following approximation problem : given a continuous function f and the number of approximating n, we proved that f is a uniform limit of a sum of separable variables activation functions. Numerical applications of this 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 neural network.

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] JASON M. KLUSOWSKI, ANDREW 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, num. 1-11, 2018

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

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