A Comparison Between Neural Network and Sampling Kantorovich Operators in Terms of Image denoising

A. M. Acu[1], M. Ilina[1], F. Sofonea[1], A. Travaglini[1,2(✉)], and G. Vinti[1,3]

[1] Department of Mathematics and Informatics, Lucian Blaga University of Sibiu, 5-7, Str. Dr. I. Ratiu, 550012 Sibiu, Romania
{anamaria.acu,mihai.ilina,florin.sofonea}@ulbsibiu.ro,
arianna.travaglini@unibas.it, gianluca.vinti@unipg.it

[2] Department of Engineering, University of Basilicata, 10, V.le dell'Ateneo Lucano, 85100 Potenza, Italy

[3] Department of Mathematics and Computer Science, University of Perugia, 1, Via Vanvitelli, 06123 Perugia, Italy

Abstract. Neural Network (NN) operators have gained significant attention in recent years due to their strong connections with Approximation Theory and their wide range of applications. In this paper, we investigate the potential of their multidimensional formulation, implementing an algorithm for digital image reconstruction. Specifically, we compare the performances of NN operators with those ones of the well-known sampling Kantorovich (SK) operators, whose implementation is a fairly recently used algorithm in image processing that serves as both a smoothing filter and a resolution enhancement tool. The comparison is conducted through a quantitative evaluation using three similarity indices: the Structural Similarity Index (SSIM), the Peak Signal-to-Noise Ratio (PSNR), and a newly introduced metric called ꟻLIP. A dataset of three reference images is used to assess the reconstruction quality of both approaches.

Keywords: Neural Network operators · sampling Kantorovich operators · digital image processing · sigmoidal activation functions

1 Introduction

The theory of artificial neural networks (ANNs) was developed in the 20th century as an attempt to create a simplified model of the human brain. Over time, ANNs have proven to be highly effective in a wide range of fields (see, e.g., [11,26,32]). Today, their relevance continues to grow, particularly due to their strong connections with artificial intelligence (AI). One of the key mathematical tools related to ANNs is the theory of neural network (NN) operators. These operators play a crucial role in the approximation processes underlying neural

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15892, pp. 379–399, 2026.
https://doi.org/10.1007/978-3-031-97638-4_24

networks. These operators, inspired by the activation functions of artificial neural networks, were first introduced by Cybenko [22] and further developed by researchers like Cardaliaguet and Euvrard (see [9]), who considered bell-shaped activation functions with compact support. However, since most commonly used activation functions, such as the logistic and hyperbolic tangent sigmoidal functions, have unbounded support, the theory of NN operators has been extended in recent years, yielding to the study of sigmoidal activation functions (see, e.g. [7,8,10,17,24]).

Let $\mathcal{R} := [a_1, b_1] \times \cdots \times [a_s, b_s]$, $s \in \mathbb{N}$, then the multivariate Neural Network operators, activated by sigmoidal function σ considered in this paper, are of the form:

$$F_n(f, \underline{x}) = \frac{\sum_{k_1=\lceil na_1 \rceil}^{\lfloor nb_1 \rfloor} \cdots \sum_{k_s=\lceil na_s \rceil}^{\lfloor nb_s \rfloor} f\left(\frac{\underline{k}}{n}\right) \Psi_\sigma \left(n\underline{x} - \underline{k}\right)}{\sum_{k_1=\lceil na_1 \rceil}^{\lfloor nb_1 \rfloor} \cdots \sum_{k_s=\lceil na_s \rceil}^{\lfloor nb_s \rfloor} \Psi_\sigma \left(n\underline{x} - \underline{k}\right)},$$

for every $\underline{x} \in \mathcal{R}$, where $f : \mathcal{R} \to \mathbb{R}$ is a bounded function, $\underline{k} := (k_1, \ldots, k_s) \in \mathbb{Z}^s$ and Ψ_σ represents its multidimensional kernel, given by the product of density (kernel) functions, as specified later, which has to satisfy suitable assumptions.

This definition and the related approximation properties are the fundamental theoretical background for the implementation of NN operators to reconstruct digital images, allowing a comparison with the well known sampling Kantorovich (SK) algorithm.

Indeed, among sampling-type operators, the sampling Kantorovich operator represents a valuable tool for handling not necessarily continuous functions. Its multidimensional extension was introduced in [18] as follows:

$$(SK_w^\chi f)(\underline{x}) = \sum_{\underline{k} \in \mathbb{Z}^n} \chi(w\underline{x} - \underline{k}) \cdot \left[\frac{w^s}{A_{\underline{k}}} \cdot \int_{R_{\underline{k}}^w} f(\underline{u}) d\underline{u} \right] \quad (\underline{x} \in \mathbb{R}^s, w > 0), \tag{1}$$

where χ is the kernel of the operator and $f : \mathbb{R}^s \to \mathbb{R}$ is a locally integrable function such that the above series converges for each $\underline{x} \in \mathbb{R}^s$. The family of operators Eq. (1) has been implemented, enabling not only image reconstruction but also resolution enhancement. This is due to its dual role as both a smoothing (low-pass) filter and a resizing tool, effectively allowing the increasing of the information content of images (see [13]). For these reasons, the sampling Kantorovich algorithm has been successfully applied in both biomedical and engineering fields, yielding satisfactory results (see, e.g., [12–14,27–29]).

Taking advantage of the popularity of NN operators, which have seen significant growth in recent years not only due to their practical applications but also because of their relevance in Approximation Theory, in this paper, we exploit their multidimensional version to implement it for digital image reconstruction, allowing a comparison with the sampling Kantorovich algorithm.

The performances of both algorithms are evaluated and compared, considering a dataset of three reference images which have been corrupted by gaussian

noise, by means of three similarity indices widely used in literature: the Structural Similarity Index (SSIM), the Peak Signal-to-Noise Ratio (PSNR) and a newly developed one, called ƎLIP.

The paper is organized as follows: in Sect. 2 the approximation properties of the Neural Network operators are recalled, starting from the concept of sigmoidal activation functions. In particular, we propose here a method for generating them by an average of a given sigmoidal function (Prop. 1), together with an estimate between them (Prop. 2); moreover several examples of sigmoidal activation functions are discussed; in Sect. 3 the definition and the approximation results regarding the sampling Kantorovich operators, together with some examples of kernels satisfying the required assumptions, are mentioned; Sect. 4 deals with the implementations of both NN and SK operators: the pseudo-codes of the two algorithm are shown; in Sect. 5 the dataset and the similarity indices employed for the comparisons are presented, while in Sect. 6 the performances of the two algorithms related to the image reconstruction tasks are discussed; in Sect. 7 we summarize some remarks which conclude the paper.

2 Neural Network (NN) Operators

This section deals with the main approximation properties of the Neural Network operators, starting by sigmoidal activation functions: a method for generating them by an average of a given sigmoidal function is introduced and some examples are recalled. The multivariate version of the NN operators, together with the related approximation results, which serves as background for the applications to digital images, is also presented.

2.1 Sigmoidal Activation Functions

This section presents key definitions and fundamental results that will serve as the basis for constructing NN operators. A function $\sigma : \mathbb{R} \to \mathbb{R}$ is called a *sigmoidal function* if the following limits hold

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

Let σ be a non-decreasing sigmoidal function σ such that the following assumptions are satisfied

($\Sigma 1$) $\sigma(x) - 1/2$ is an odd function;
($\Sigma 2$) $\sigma \in C^2(\mathbb{R})$ is concave for $x \geq 0$;
($\Sigma 3$) $\sigma(x) = \mathcal{O}(|x|^{-\alpha-1})$ as $x \to -\infty$, for some $\alpha > 0$.

The following result provides a method for generating sigmoidal functions starting from a given sigmoidal function.

Proposition 1. *If $\sigma : \mathbb{R} \to \mathbb{R}$ is a sigmoidal function that verifies the conditions (Σ_1)-(Σ_3), then the function*

$$\rho(x) := \int_{x-1/2}^{x+1/2} \sigma(t)dt,\ x \in \mathbb{R} \tag{2}$$

is also a sigmoidal function that satisfies the conditions (Σ_1)-(Σ_3).

Proof. Since $\rho'(x) = \sigma\left(x + 1/2\right) - \sigma\left(x - 1/2\right) \geq 0$, it follows that ρ is a non-decreasing function. The condition (Σ_1) is equivalent to $\sigma(-x)+\sigma(x) = 1$, $x \in \mathbb{R}$. Therefore,

$$\int_{x-1/2}^{x+1/2} \left[\sigma(t) + \sigma(-t)\right] dt = 1.$$

This can be expressed as

$$\int_{-x-1/2}^{-x+1/2} \sigma(t)dt + \int_{x-1/2}^{x+1/2} \sigma(t)dt = 1.$$

Consequently, we have $\rho(-x)+\rho(x) = 1$, which implies that ρ satisfies condition (Σ_1).

Next, we compute the second derivative

$$\rho''(x) = \sigma'\left(x + 1/2\right) - \sigma'\left(x - 1/2\right).$$

Since σ is concave on $[0, \infty)$, it follows σ' is a non-increasing function on $[0, \infty)$.

Case I: $x - 1/2 \geq 0$. Then, $[x - 1/2, x + 1/2] \subset [0, \infty)$. Thus,

$$\sigma'\left(x - 1/2\right) \geq \sigma'\left(x + 1/2\right),$$

and $\rho''(x) \leq 0$.

Case II: $x - 1/2 \leq 0$ and $x \geq 0$. Then, $x + 1/2 \geq 0$, and

$$\sigma'\left(x + 1/2\right) = \sigma'\left(-x - 1/2\right) \leq \sigma'(x - 1/2),$$

since on $(-\infty, 0]$ the function σ' is increasing.

Thus, we find $\rho''(x) \leq 0$ on $[0, \infty]$. Therefore, ρ satisfies (Σ_2).

Finally, using condition (Σ_3) and definition of the function ρ, it is straightforward to verify that $\rho(x) = \mathcal{O}(|x|^{-1-\alpha})$.

Proposition 2. *Let σ be a sigmoidal function satisfying conditions (Σ_1)-(Σ_3). Then*

$$\rho(x) \geq \sigma(x),\ \textit{for } x \leq -\frac{1}{2},\ \textit{and}$$
$$\rho(x) \leq \sigma(x),\ \textit{for } x \geq \frac{1}{2},$$

where the function ρ is defined in Eq. (2).

Proof. Let $x \leq -\frac{1}{2}$. Then $[x-1/2, x+1/2] \subset (-\infty, 0]$, hence σ is convex on $[x-1/2, x+1/2]$. Using Hermite-Hadamard inequality, we get

$$\sigma(x) \leq \int_{x-1/2}^{x+1/2} \sigma(t)dt, \text{ i.e., } \sigma(x) \leq \rho(x).$$

In a similar way one obtains the second inequality.

The result presented above is illustrated in Fig. 1 using the sigmoidal function $\sigma(x) = 1/2[\tanh(x)+1]$ (see Sect. 2.2). By applying the integral transform Eq. (2), we derive the following sigmoidal function

$$\rho(x) = -\frac{\ln(\mathrm{e}^{2x-1}+1)}{2} + \frac{\ln(\mathrm{e}^{2x+1}+1)}{2}.$$

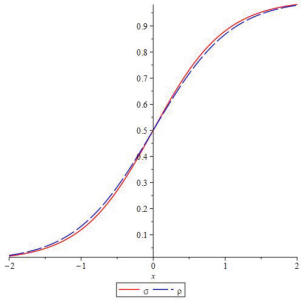


Fig. 1. Plot of the sigmoidal functions σ and ρ.

Now, we recall the definition of the density (kernel) function ϕ_σ generated by the sigmoidal function σ:

$$\phi_\sigma(x) := \frac{1}{2}[\sigma(x+1) - \sigma(x-1)],\ x \in \mathbb{R}.$$

It is easy to verify that ϕ_σ is non-negative and even, it is non-decreasing on $(-\infty, 0]$ and non-increasing on $[0, +\infty)$.

Moreover, let $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceiling and the floor of a given number, respectively. The function ϕ_σ satisfies the following properties

i) $\sum\limits_{k\in\mathbb{Z}} \phi_\sigma(x-k) = 1,\ x \in \mathbb{R}$;

ii) $0 < \phi_\sigma(1) \leq \sum\limits_{k=\lceil na \rceil}^{\lfloor nb \rfloor} \phi_\sigma(nx-k) \leq 1$, for every $a, b \in \mathbb{R}$ with $a < b$, $x \in [a, b]$ and $n \in \mathbb{N}^+$, such that $\lceil na \rceil \leq \lfloor nb \rfloor$;

iii) $\phi_\sigma(x) = \mathcal{O}\left(|x|^{-\alpha-1}\right)$, as $x \to \pm\infty$,

see, e.g. [15].

2.2 Examples of Sigmoidal Activation Functions

In the following, we illustrate a few examples of sigmoidal activation functions that fulfill the necessary conditions (Σ_1)-(Σ_3).

A widely recognized smooth sigmoidal function is the logistic function, expressed as $\sigma_l(x) = (1+e^{-x})^{-1}$,where $x \in \mathbb{R}$. Another example is based on the hyperbolic tangent sigmoidal function, defined by $\sigma_h(x) = 1/2[\tanh(x)+1]$ with $x \in \mathbb{R}$.

Note that both these functions satisfy the condition $(\Sigma 3)$ for every $\alpha > 0$, due to their exponential decay to zero as $x \to -\infty$.

However, it is also possible to consider a non-smooth sigmoidal function, the ramp function, denoted as

$$\sigma_R(x) := \begin{cases} 0, \ x < -1/2, \\ x + 1/2, \ -1/2 \le x \le 1/2, \\ 1, \ x > 1/2. \end{cases}$$

Although this function does not satisfy the $(\Sigma 2)$, is useful for the approximation results shown later, since condition $(\Sigma 2)$ is used only to prove that ϕ_σ is non-decreasing for $x \le 0$ and non-increasing for $x \ge 0$; indeed, having ϕ_{σ_R} this property, this means that also non-smooth sigmoidal functions are suitable for the following theory.

Let $(x)_+ := \max\{x, 0\}$. Using the B-spline function of order $n \in \mathbb{N}^+$

$$M_n(x) := \frac{1}{(n-1)!} \sum_{k=0}^{n} (-1)^k \binom{n}{k} \left(\frac{n}{2} + x - k\right)_+^{n-1}, \ x \in \mathbb{R},$$

a sigmoidal function can be constructed as follows

$$\sigma_{M_n}(x) := \int_{-\infty}^{x} M_n(t)dt, \ x \in \mathbb{R}.$$

It is important to observe that for $n = 1$, $\sigma(x)$ is the ramp function.

Finally, we can consider the well-known rectified linear unit (ReLU) activation function (see, e.g., [2,23]):

$$\sigma_{ReLU}(x) := (x)_+, \quad x \in \mathbb{R},$$

and we obtain the following relation:

$$\sigma_{M_1}(x) = \sigma_R(x) = \sigma_{ReLU}(x + 1/2) - \sigma_{ReLU}(x - 1/2). \tag{3}$$

This allows us to express the corresponding density function in terms of the ReLU activation function, as:

$$\begin{aligned} \phi_{\sigma_R}(x) = \phi_{\sigma_{M_1}}(x) = & \frac{1}{2} \left[\sigma_{ReLU}(x + 3/2) - \sigma_{ReLU}(x + 1/2)\right. \\ & \left. -\sigma_{ReLU}(x - 1/2) + \sigma_{ReLU}(x - 3/2)\right]. \end{aligned}$$

As a direct consequence of this relationship, the neural network (NN) operators F_n, when activated by σ_{M_1} (or equivalently σ_R), can be interpreted as a NN operators activated by the above linear combination of ReLU functions.

For more details on such sigmoidal functions and, more in general, on NN operators, the reader may find useful to consult the works [7,8,10,17,19,20,24].

2.3 Approximation Properties of Neural Network Operators

Denote by $\mathcal{R} := [a_1, b_1] \times \cdots \times [a_s, b_s]$, $s \in \mathbb{N}$. Let $C^0(\mathcal{R})$ and $C^0(\mathbb{R}^s)$ be the space of all continuous functions defined on $\mathcal{R}$ and $\mathbb{R}^s$ respectively, equipped with the supremum norm $\|\cdot\|_\infty$.

In this section, we focus on the multivariate neural network operators. To this aim, we consider the multidimensional kernel

$$\Psi_\sigma(\underline{x}) := \phi_\sigma(x_1)\phi_\sigma(x_2)\dots\phi_\sigma(x_s),\ \underline{x} := (x_1, \dots, x_s) \in \mathbb{R}^s.$$

It is possible to see that the function Ψ_σ satisfies the following properties

i) $\Psi_\sigma(\underline{x}) = \mathcal{O}(\|\underline{x}\|_2^{-1-\alpha})$, as $\|x\|_2 \to +\infty$, where $\|\cdot\|_2$ is the usual Euclidean norm of $\mathbb{R}^s$;

ii) $\Psi_\sigma \in L^1(\mathbb{R}^s)$ and for every $\underline{x} \in \mathbb{R}^s$, $\sum_{\underline{k} \in \mathbb{Z}^s} \Psi_\sigma(\underline{x} - \underline{k}) = 1$;

iii) for every $\gamma > 0$, $\lim_{n \to +\infty} \sum_{\|\underline{x} - \underline{k}\|_2 > \gamma n} \Psi_\sigma(\underline{x} - \underline{k}) = 0$, uniformly with respect to $\underline{x} \in \mathbb{R}^s$,

(see e.g., [16]).

The linear positive multivariate neural network operator, activated by sigmoidal function σ, is defined as

$$F_n(f, \underline{x}) = \frac{\sum_{k_1 = \lceil na_1 \rceil}^{\lfloor nb_1 \rfloor} \cdots \sum_{k_s = \lceil na_s \rceil}^{\lfloor nb_s \rfloor} f\left(\frac{\underline{k}}{n}\right) \Psi_\sigma\left(n\underline{x} - \underline{k}\right)}{\sum_{k_1 = \lceil na_1 \rceil}^{\lfloor nb_1 \rfloor} \cdots \sum_{k_s = \lceil na_s \rceil}^{\lfloor nb_s \rfloor} \Psi_\sigma\left(n\underline{x} - \underline{k}\right)}, \tag{4}$$

for every $\underline{x} \in \mathcal{R}$, where $f : \mathcal{R} \to \mathbb{R}$ is a bounded function, and $\underline{k} := (k_1, \dots, k_s) \in \mathbb{Z}^s$.

Obviously, $\prod_{i=1}^{s} \sum_{k_i = \lceil na_i \rceil}^{\lfloor nb_i \rfloor} \Psi_\sigma\left(n\underline{x} - \underline{k}\right) > 0$ for every $\underline{x} \in \mathcal{R}$ and therefore $F_n(f, \underline{x})$ turns out to be well-defined.

In [16], the following results regarding pointwise and uniform convergence for the classical NN operators were proven.

Theorem 1. *[16] Let $f : \mathcal{R} \to \mathbb{R}$ be bounded. Then,*

$$\lim_{n\to+\infty} F_n(f, \underline{x}) = f(\underline{x})$$

at each point $\underline{x} \in \mathcal{R}$ of continuity of f. Moreover, if $f \in C^0(\mathcal{R})$, then

$$\lim_{n\to+\infty} \sup_{\underline{x}\in\mathcal{R}} |F_n(f, \underline{x}) - f(\underline{x})| = \lim_{n\to+\infty} \|F_n(f, \cdot) - f(\cdot)\|_\infty = 0.$$

The above results will be a fundamental theoretical tool for their implementation to Image Processing faced in Sect. 4.

3 Sampling Kantorovich (SK) Operators

Since the '80 s, applications of sampling type operators to signal processing have been extensively studied. In particular, the sampling Kantorovich operators were introduced in [4] to reconstruct one-dimensional signals that are not necessarily continuous, thus proving to have a crucial role in sampling and signal theory.

3.1 Definition of SK Operators and Approximation Properties

A multivariate extension of the results in [4] was developed in [18], allowing to apply this theory to image processing: the family of **sampling Kantorovich operators** $(SK_w)_{w>0}$ are defined as:

$$(SK_w f)(\underline{x}) := \sum_{\underline{k}\in\mathbb{Z}^s} \chi(w\underline{x} - \underline{k}) \left[w^s \int_{R^w_{\underline{k}}} f(\underline{u}) d\underline{u} \right], \quad \underline{x} \in \mathbb{R}^s, \quad w > 0 \qquad (5)$$

where $f : \mathbb{R}^s \to \mathbb{R}$ is a locally integrable function (representing the signal or the image function) such that the above series converges for every $\underline{x} \in \mathbb{R}^s$. The sets

$$R^w_{\underline{k}} := \left[\frac{k_1}{w}, \frac{k_1+1}{w}\right] \times \left[\frac{k_2}{w}, \frac{k_2+1}{w}\right] \times \ldots \times \left[\frac{k_n}{w}, \frac{k_n+1}{w}\right]$$

represent the regions where the mean values of the signal f are computed.

The function $\chi : \mathbb{R}^s \to \mathbb{R}$, known as *kernel*, is chosen to satisfy the following conditions:

(χ_1) χ belongs to $L^1(\mathbb{R}^s)$ and is bounded within a ball centered at the origin in $\mathbb{R}^s$.

(χ_2) For every $\underline{x} \in \mathbb{R}^s$, the following identity (partition of unity) holds:

$$\sum_{\underline{k}\in\mathbb{Z}^s} \chi(\underline{x} - \underline{k}) = 1.$$

(χ_3) There exists a constant $\beta > 0$ such that the discrete absolute moment of order β of χ is finite, i.e.,

$$m_\beta(\chi) := \sup_{\underline{u}\in\mathbb{R}^s} \sum_{\underline{k}\in\mathbb{Z}^s} |\chi(\underline{u} - \underline{k})| \cdot \|\underline{u} - \underline{k}\|_2^\beta < +\infty.$$

It is well known that the operators SK_w, defined for $w > 0$ in Eq. (5) serve as approximation operators with the property of reconstructing continuous and bounded signals both pointwise and uniformly as $w \to +\infty$. Furthermore, these operators have proven effective in reconstructing signals that are not necessarily continuous, e.g. belonging to L^p-space (see, e.g., [4,18]).

In the following, we present some approximation results established in [18], which represent the fundamental theoretical background for applying the SK operators to digital image processing. As will be detailed in Sect. 4, the implementation of these results enables both the reconstruction and the enhancement of images.

Specifically, when dealing with continuous signals, the following convergence result holds.

Theorem 2. *Let $f \in C^0(\mathbb{R}^s)$. Then, for every $\underline{x} \in \mathbb{R}^s$,*

$$\lim_{w\to+\infty} (SK_w^\chi f)(\underline{x}) = f(\underline{x}).$$

In particular, if $f \in C(\mathbb{R}^s)$, then

$$\lim_{w\to+\infty} ||SK_w^\chi f - f||_\infty = 0,$$

where $C(\mathbb{R}^s)$ represents the space of uniformly continuous and bounded functions on $\mathbb{R}^s$.

Another significant result proves that the SK operators allow the reconstruction of signals that are not necessarily continuous. In fact, the following L^p-convergence theorem holds.

Theorem 3. *For every $f \in L^p(\mathbb{R}^s)$, where $1 \le p < +\infty$, we have*

$$\lim_{w\to+\infty} ||SK_w^\chi f - f||_p = 0.$$

Furthermore, the following inequality holds:

$$||SK_w^\chi||_p \le \delta^{-n/p}(m_0(\chi))^{(p-1)/p}||\chi||_1^{1/p}||f||_p \quad (f \in L^p(\mathbb{R}^s)).$$

In recent years, the study of sampling Kantorovich operators has been further expanded in several works, addressing various aspects, including their convergence in different functional spaces, the study of their approximation order, as well as saturation, inverse results, and regularization properties (see [21]), while their extension in the nonlinear setting was studied in [30]. Furthermore, for what concerns applications, the sampling Kantorovich operator has been implemented and successfully applied to various practical problems, including image reconstruction, enhancement, and segmentation (see, e.g., [12–14,27–29]). For modifications of Kantorovich operators, see e.g. [1].

3.2 Examples of Kernels

It is important to note that the requirements (χ_1), (χ_2), and (χ_3) on the kernel functions are not restrictive, since are typically met by discrete approximate identities, (see, e.g., [6]). Moreover, we highlight that the choice of the kernel function plays a crucial role in signal reconstruction using sampling Kantorovich operators and in their implementation. In this context, we recall the definitions of some kernels that satisfy the assumptions (χ_i), $i = 1, 2, 3$ (see, e.g., [6]).

For instance, *central B-splines* of order $r \in \mathbb{N}$ serve as useful examples of one-dimensional symmetric kernels with compact support. They are defined as follows:

$$M_r(x) := \frac{1}{(r-1)!} \sum_{i=0}^{r} (-1)^r \binom{r}{i} \left(\frac{r}{2} + x - i\right)_+^{r-1},$$

where the function $(x)_+ = \max\{x, 0\}$ represents the positive part of $x \in \mathbb{R}$. The multidimensional extension of these kernels can be obtained through the following tensor-product of one-dimensional kernels:

$$\mathcal{M}_r^s(\underline{x}) := \prod_{i=1}^{s} M_r(x_i), \quad \underline{x} = (x_1, \ldots, x_s) \in \mathbb{R}^s.$$

Using this tensor product representation, it is straightforward to verify that all the required assumptions on the kernel $\mathcal{M}_r^s$ are satisfied (see Fig. 2 a)).

Another significant class of kernels, widely employed in image enhancement algorithms, is the so-called Jackson-type kernels (see Fig. 2 b)):

$$J_k^s(\underline{x}) := \prod_{i=1}^{s} J_k(x_i), \quad \underline{x} \in \mathbb{R}^s,$$

where the one-dimensional Jackson-type kernels are given by:

$$J_k(x) = c_k \operatorname{sinc}^{2k}\left(\frac{x}{2k\pi\alpha}\right), \text{ being } sinc(x) := \begin{cases} \frac{sin(\pi x)}{\pi x} & x \in \mathbb{R} \setminus \{0\} \\ 1 & x = 0, \end{cases}$$

and c_k is a non-zero normalization coefficient given by

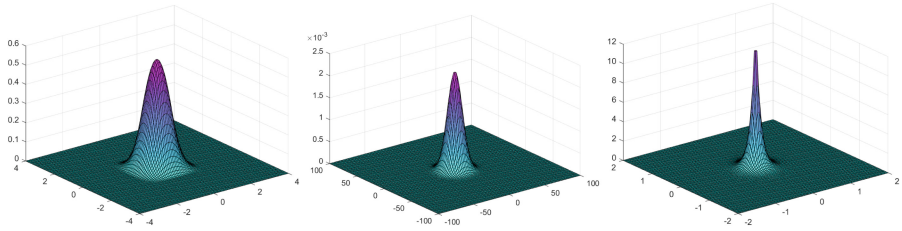
$$c_k := \left[\int_{\mathbb{R}} sinc^{2k}\left(\frac{u}{2k\pi\alpha}\right) du \right]^{-1}.$$

Jackson-type kernels are symmetric functions with unbounded support exhibiting a rapid decay.

Finally, there are also kernels that are not of the product type. One of such examples is given by the Picard kernel (see, e.g., [6,25]), which is a radial kernel defined as:

$$P_w(\underline{x}) = \frac{w^s \Gamma(s/2)}{2\pi^{s/2}\Gamma(n)} e^{-w|\underline{x}|}, \quad \underline{x} \in \mathbb{R}^s, w > 0,$$

where Γ is the Gamma Euler function (see Fig. 2 c)).



a) b) c)

Fig. 2. Graphs of the above mentioned kernels: a) B-spline kernel M_3^2, b) Jackson-type kernel J_{12}^2, c) Picard's kernel P_{10}.

4 Implementation of NN and SK Algorithms

In this section the implementations of NN and SK operators for digital image reconstruction are discussed.

4.1 Implementation of NN Operators

The following pseudo code (Algorithm 1) represents an implementation in the C# language using .NET framework, that illustrates the action of the Neural Network Operator. This implementation is designed to efficiently compute function approximations using parallelized numerical methods. The core procedure evaluates a set of points and applies transformation functions, including sigmoid-based activation and structured matrix operations, to derive meaningful results from the input data.

Algorithm 1: Reconstruction of the original image M by the Neural Network (NN) operator based upon the bi-dimensional density function Ψ_σ.

1. **Inputs**:
   - The gray 8-bit scale matrix M of the image, of size $H \times W$.
   - The sampling rate n.
   - Two vectors representing discrete sampling points:
     (a) A vector spanning the interval $[a1, a2]$ (the width of the image).
     (b) A vector spanning the interval $[b1, b2]$ (the height of the image).
2. **Steps**:
   - Cartesian pairs are formed from these vectors, creating a structured grid of sampling points.
   - Choice of the sigmoidal function σ.
3. **Iteration**:
   - For each vector pair (x, y).
   - The indices k_1 and k_2 are iterated over a fixed range.
   - The normalized indices are computed as $k_1^{'} = \frac{k_1}{n}$ and $k_2^{'} = \frac{k_2}{n}$.
   - The one-dimensional density functions are computed as: $\phi_\sigma(nx_1 - k_1)$ and $\phi_\sigma(nx_2 - k_2)$.
   - The bi-dimensional density function is evaluated as the product of one-dimensional density functions: $\Psi_\sigma(nx_1 - k_1, nx_2 - k_2) = \phi_\sigma(nx_1 - k_1) \cdot \phi_\sigma(nx_2 - k_2)$.
   - The operator value is obtained by computing:

$$F_n(f; x_1, x_2) = \frac{\sum_{k_1=\lceil na_1 \rceil}^{\lfloor nb_1 \rfloor} \sum_{k_2=\lceil na_2 \rceil}^{\lfloor nb_2 \rfloor} M\left(k_1^{'}, k_2^{'}\right) \Psi_\sigma(nx_1 - k_1, nx_2 - k_2)}{\sum_{k_1=\lceil na_1 \rceil}^{\lfloor nb_1 \rfloor} \sum_{k_2=\lceil na_2 \rceil}^{\lfloor nb_2 \rfloor} \Psi_\sigma(nx_1 - k_1, nx_2 - k_2)}.$$

4. **Output**: The reconstructed gray scale image of resolution $H \times W$ through the Neural Network operator.

The convolutive nature of the NN operators reflects in a low-pass filtering behavior when reconstructing digital images.

4.2 Implementation of SK Operators

First of all, it is necessary to recall that a grayscale image is mathematically represented as a matrix $A = (a_{ij})_{i,j=1,\ldots,m}$, which can be described using the so-called image function,

$$I(x, y) := \sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} \cdot \mathbf{1}_{ij}(x, y), \quad (x, y) \in \mathbb{R}^2$$

where

$$\mathbf{1}_{ij}(x,y) = \begin{cases} 1, & \text{if } (x,y) \in (i-1,i] \times (j-1,j], \\ 0, & \text{otherwise.} \end{cases}$$

Thus, it is possible to reconstruct I (which plays the role of f) using NN and SK operators in their two-dimensional form, selecting a suitable sigmoidal and kernel function, respectively for NN and SK operators, as previously described. The following pseudo code, described by ALgorithm 2, represents an implementation in MATLAB© of the sampling Kantorovich operators for the reconstruction and rescaling of digital images.

Algorithm 2: Reconstruction of the original image M by the sampling Kantorovich operators upon bi-dimensional kernel χ.

1. **Inputs**:
   - The gray 8-bit scale matrix M of the image, of size $H \times W$.
   - The sampling rate w.
   - The scaling factor R.
2. **Steps**:
   - Calculation of the image function I.
   - Choice and definition of the kernel function χ.
   - Size of the reconstructed image $(H \cdot R) \times (W \cdot R)$.
   - Computation of the matrices of the mean values (samples) of I by means of the Kronecker matrix product.
   - Definition of the vectors containing the arguments of χ.
3. **Iteration**: Summation over $\underline{k}$ of all non-zero terms of the form $\chi(w\underline{x} - \underline{k}) \cdot \left[w^2 \int_{R^w_{\underline{k}}} I(\underline{u}) d\underline{u} \right]$, for a suitable fixed grid of points $\underline{x}$.
4. **Output**: The reconstructed gray scale image of resolution $(H \cdot R) \times (W \cdot R)$ through the sampling Kantorovich operator.

When applied to digital images, the SK algorithm acts both as a smoothing filter, due to its convolutive nature, with the effect of reduction of noise, and as a rescaling algorithm, allowed by the scaling factor R, resulting in an increasing of the information content of the image. Another fundamental role in SK-reconstruction algorithm is played by the parameter w, which represents the sampling rate: in accordance with the theoretical results represented by Theorems 2 and 3 of Sect. 3, a higher sampling rate leads to a more accurate image reconstruction.

5 Experimental Data and Methodology

In order to display the action of the NN algorithm, compared to the one of the SK algorithm, we selected the following reference images, shown in Fig. 3 a), in order: "autumn.png" (345×206), "cameraman.png" (256×256) and "Lena.png"

(200×200). We choose images with different characteristics to enable a comparative analysis across variations in size, brightness, and texture. Gaussian noise of mean 0 and variance 0.001 was applied to the reference images (see Fig. 3 b)). Then, we have applied the NN algorithm, activated by the sigmoidal function generated by a linear combination of ReLU (see (3) of Sect. 2) and the hyperbolic tangent functions, to these noised images, for n ranging from 1 to 7. For what concerns the SK algorithm, we have chosen the bi-dimensional Jackson type kernel J_{12}^2, scaling factor $R = 1$ (for the correct comparisons), for values of $w = 10, 15, 20, 25, 30, 35, 40$.



Fig. 3. a) Reference images and b) corresponding noised images used for testing NN and SK algorithms.

For our assessment, we took into account three similarity metrics commonly referenced in the literature: Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR) and ƎLIP:

- The SSIM, (see, e.g., [31]), evaluates the similarity between two images focusing on three key components: luminance, contrast, and structural information. The SSIM index between two images X and Y of the same size $N \times M$ is defined as:

$$SSIM(X,Y) = l(X,Y) \cdot c(X,Y) \cdot s(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}.$$

  Here, μ_X and μ_Y denote the mean intensity values of images X and Y, respectively. The terms σ_X^2 and σ_Y^2 represent the variance of each image, while σ_{XY} corresponds to their covariance. The constants C_1 and C_2 are introduced to avoid numerical instability. When computed, SSIM produces a score within the range $[-1, 1]$, where 1 indicates perfect correspondence, 0 suggests no correlation, and -1 corresponds to perfect anti-correlation between the two images.

- The PSNR quantifies the qualitative similarity between two images X and Y: the higher its value, the greater their similarity.
  It is expressed as:

$$PSNR(X,Y) = 10 \cdot \log \frac{f_{max}^2}{\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{|X(i,j) - Y(i,j)|^2}{NM}}$$

  where f_{max} represents the highest possible pixel intensity in the image, i.e. 255 for 8-bit images, while the denominator is the Mean Square Error (MSE).
- ꟻLIP is a metric, recently introduced in [3], which evaluates differences in color and edges based on, or inspired by, models of the human visual system. ꟻLIP generates an error map that accurately reflects the magnitude of the perceived difference between the two images at each pixel, as shown in Fig. 4. ꟻLIP ranges from 0, when no differences are perceived between the two images, to 1 when the two images are totally different. Being ꟻLIP designed to handle aspects related to the human perception, each user has to enter the display width in meters and in pixels, and a script then computes the appropriate viewing distance at which the user should position himself in order to observe 67 pixels per degree of visual angle, which represents the value used when computing all error maps, according to the default setup used in [3].



a) b) c)

Fig. 4. Example of the application of ꟻLIP metric: a) Reference image, b) Test image, c) ꟻLIP error map with an output value of 0.055. The heatmap is shown to the right, where bright yellow corresponds to maximum error and dark blue to minimum error.

6 Comparisons Between the Two Algorithms and Numerical Results

In this section we analyze and compare the results obtained applying the NN and SK algorithms to the noised dataset for different values of n and w respectively. We remark that the chosen values for n and w are those for which the best indices are achieved.

Table 1. Numerical results in terms of SSIM, PSNR and ꟻLIP indexes, obtained by reconstructing the reference dataset a) by the NN algorithm for different values of n, with the hyperbolic tangent as activation function and b) by the SK algorithm with the bi-dimensional Jackson type kernel J^2_{12}, scaling factor $R = 1$, for different values of w.

a) Reconstructions using NN algorithm

Original Image	*n=1*	*n=2*	*n=3*	*n=4*	*n=5*	*n=6*	*n=7*
	SSIM:0.8256 PSNR:27.3652 ꟻLIP:0.0749	SSIM:0.8543 PSNR:29.6230 ꟻLIP:0.0699	SSIM:0.8376 PSNR:29.4580 ꟻLIP:0.0725	SSIM:0.8341 PSNR:29.4030 ꟻLIP:0.0730	SSIM:0.8337 PSNR:29.3960 ꟻLIP:0.0730	SSIM:0.8337 PSNR:29.3945 ꟻLIP:0.0731	SSIM:0.8337 PSNR:29.3942 ꟻLIP:0.0731
	SSIM:0.8218 PSNR:25.6632 ꟻLIP:0.0767	SSIM:0.8349 PSNR:28.0481 ꟻLIP:0.0697	SSIM:0.8152 PSNR:27.9996 ꟻLIP:0.0715	SSIM:0.8114 PSNR:27.9631 ꟻLIP:0.0719	SSIM:0.8109 PSNR:27.9587 ꟻLIP:0.0719	SSIM:0.8108 PSNR:27.9574 ꟻLIP:0.0719	SSIM:0.8108 PSNR:27.9572 ꟻLIP:0.0719
	SSIM:0.8371 PSNR:27.3253 ꟻLIP:0.0792	SSIM: 0.8609 PSNR:29.0849 ꟻLIP:0.0762	SSIM:0.8465 PSNR:28.8127 ꟻLIP:0.0798	SSIM:0.8435 PSNR:28.7442 ꟻLIP:0.0804	SSIM:0.8431 PSNR:28.7346 ꟻLIP:0.0805	SSIM:0.8430 PSNR:28.7336 ꟻLIP:0.0805	SSIM:0.8430 PSNR:28.7334 ꟻLIP:0.0805

b) Reconstructions using SK algorithm

Original Image	*w=10*	*w=15*	*w=20*	*w=25*	*w=30*	*w=35*	*w=40*
	SSIM:0.8211 PSNR:27.1496 ꟻLIP:0.0795	SSIM:0.8539 PSNR:29.8929 ꟻLIP:0.0668	SSIM:0.8416 PSNR:31.5011 ꟻLIP:0.0594	SSIM:0.8128 PSNR:31.6169 ꟻLIP:0.0547	SSIM:0.7857 PSNR:31.0632 ꟻLIP:0.0525	SSIM:0.7670 PSNR:30.5516 ꟻLIP:0.0521	SSIM:0.7534 PSNR:30.1790 ꟻLIP:0.0523
	SSIM:0.8102 PSNR:25.9844 ꟻLIP:0.0808	SSIM:0.8334 PSNR:28.7327 ꟻLIP:0.0685	SSIM:0.8168 PSNR:30.7104 ꟻLIP:0.0607	SSIM:0.7833 PSNR:31.3482 ꟻLIP:0.0548	SSIM:0.7522 PSNR:31.0655 ꟻLIP:0.0512	SSIM:0.7312 PSNR:30.6424 ꟻLIP:0.0500	SSIM:0.7155 PSNR:30.2735 ꟻLIP:0.0500
	SSIM:0.8252 PSNR:27.4109 ꟻLIP:0.0807	SSIM:0.8596 PSNR:29.8454 ꟻLIP:0.0674	SSIM:0.8573 PSNR:31.2926 ꟻLIP:0.0602	SSIM:0.8359 PSNR:31.4715 ꟻLIP:0.0554	SSIM:0.8125 PSNR:30.9903 ꟻLIP:0.0530	SSIM:0.7956 PSNR:30.5064 ꟻLIP:0.0524	SSIM:0.7835 PSNR:30.1499 ꟻLIP:0.0526

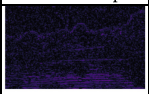
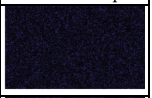
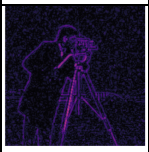
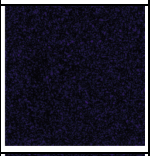
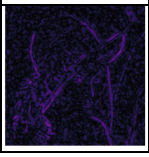
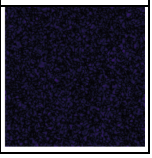
In Table 1 a), we can see the performances of the NN algorithm in terms of SSIM, PSNR and ꟻLIP, for $n = 1, 2, 3, 4, 5, 6, 7$, with the hyperbolic tangent as activation function. As we can see from the results obtained, SSIM index is relatively high and close to 1 for each value of n. For what concerns the PSNR index, the values obtained when comparing the original images with the

Table 2. Comparison of PSNR, SSIM, and ꟻLIP: a) Results for different values of n using the linear combination of ReLU and the hyperbolic tangent (Tanh) as activation functions. b) Best NN (with the linear combination of ReLUs) vs SK results together with the ꟻLIP maps.

a)

Original Image	n	PSNR		SSIM		ꟻLIP	
		ReLU	Tanh	ReLU	Tanh	ReLU	Tanh
	1	29.3439	27.3652	0.8620	0.8256	0.0633	0.0749
	2	30.2934	29.6230	0.8338	0.8543	0.0677	0.0699
	3	30.2934	29.4580	0.8338	0.8376	0.0677	0.0725
	4	30.2934	29.4030	0.8338	0.8341	0.0677	0.0730
	5	30.2934	29.3960	0.8338	0.8337	0.0677	0.0730
	6	30.2934	29.3945	0.8338	0.8337	0.0677	0.0731
	7	30.2934	29.3942	0.8338	0.8337	0.0677	0.0731
	1	27.6775	25.6632	0.8471	0.8218	0.0656	0.0767
	2	29.0441	28.0481	0.8081	0.8349	0.0673	0.0697
	3	29.0441	27.9996	0.8081	0.8152	0.0673	0.0715
	4	29.0441	27.9631	0.8081	0.8114	0.0673	0.0719
	5	29.0441	27.9587	0.8081	0.8109	0.0673	0.0719
	6	29.0441	27.9574	0.8081	0.8108	0.0673	0.0719
	7	29.0441	27.9572	0.8081	0.8108	0.0673	0.0719
	1	29.2646	27.3253	0.8703	0.8371	0.0653	0.0792
	2	29.6574	29.0849	0.8466	0.8609	0.0739	0.0762
	3	29.6574	28.8127	0.8466	0.8465	0.0739	0.0798
	4	29.6574	28.7442	0.8466	0.8435	0.0739	0.0804
	5	29.6574	28.7346	0.8466	0.8431	0.0739	0.0805
	6	29.6574	28.7336	0.8466	0.8430	0.0739	0.0805
	7	29.6574	28.7334	0.8466	0.8430	0.0739	0.0805

b)

Original Image	*NN algorithm (ReLU)*				*SK algorithm*			
	SSIM	PSNR	ꟻLIP	ꟻLIP map	SSIM	PSNR	ꟻLIP	ꟻLIP map
	0.8620	30.2934	0.0633		0.8539	31.6169	0.0521	
	0.8471	29.0441	0.0656		0.8334	31.3482	0.0500	
	0.8703	29.6574	0.0653		0.8596	31.4715	0.0524	

reconstructed ones are quite low, suggesting that the pixel by pixel analysis of this index do not yield to significant outcomes in terms of resemblance, for the study under analysis. Finally, the ꟻLIP metric produces good results, being its values around 0.07, i.e. close to 0.

Table 1 b) displays the results of the application of the SK algorithm to the noised dataset in terms of SSIM, PSNR and ꟻLIP, for $w = 10, 15, 20, 25, 30, 35, 40$. The SSIM index yields good results, which are a little bit lower with respect to those ones produced by NN algorithm (the best results of SSIM for both the two algorithms are obtained on all the dataset for $n = 2$ and $w = 15$, respectively). The PSNR results indicate a notable correspondence between the reconstructed images and the reference ones: the best values for this index are achieved for $w = 25$ and it is bigger than the best values of the ones

Table 3. SSIM, PSNR and ꟻLIP metrics between the reference and noised images.

Image (Reference vs Noised)	SSIM	PSNR	ꟻLIP
Autumn	0.7470	29.9598	0.0527
Cameraman	0.7090	30.0741	0.0502
Lena	0.7773	29.9299	0.0529

produced by the NN algorithm (for $n = 2$). Moreover, it is worth noting that the PSNR index, taking into account its pixel-wise comparison, it is a reliable index for evaluating the considered problem, i.e. denoising. Finally, the SK algorithm outperforms the NN one, also in terms of the ꟻLIP index, which is around 0.05.

In Table 2 a), a comparison related to the NN algorithm with two different activation functions, the hyperbolic tangent and a linear combination of the ReLU functions, is shown for $n = 1, 2, 3, 4, 5, 6, 7$. As we can see from the results, both activation functions yield to good performance in image reconstruction, with a a global advantage for the NN algorithm implemented with the linear combination of ReLUs, which achieves a sort of saturation for $n \geq 2$.

Furthermore, in Table 2 b), we compare the best performances of the NN algorithm, using the linear combination of ReLUs as activation function, to the best performances of the SK algorithm, according to SSIM, PSNR and ꟻLIP metrics. We enclose also the ꟻLIP error map, to better understand the discrepancies between the originals and reconstructed images, in terms of the ꟻLIP index. The results gained, suggest that the SK algorithm outperforms the NN one for all the three images under analysis, for the PSNR and ꟻLIP indices, while the SSIM indices are slightly lower than the ones of the NN algorithm. These considerations are also supported by the ꟻLIP error maps, which are almost black for the images reconstructed with the SK algorithm, indicating very good similarity, while for the images reconstructed with NN, there are errors, marked in dark blue, located in the edges, thus suggesting that the NN algorithm does not perform as well on the edges as the SK algorithm.

Finally, to evaluate the performance of the two algorithms in terms of denoising, it is useful to compare the SSIM, PSNR, and ꟻLIP metrics obtained by comparing the original images with the noisy ones and the original images with the reconstructed ones. Table 3 shows the results achieved comparing the reference dataset (original images) with the noised images, in terms of the three metrics: it can be deduced that the images reconstructed with the SK algorithm are closer to the original ones across all three metrics, while the NN algorithm performs better only in terms of SSIM. The above consideration, together with the previous results, suggests that the NN algorithm, although its low-pass filter behavior, is less suitable for noise removal compared to the SK algorithm.

7 Conclusions

In this study, we investigate the potential of Neural Network (NN) operators for digital image reconstruction and denoising, comparing their performance with the well-known sampling Kantorovich (SK) operators. The evaluation is conducted using three similarity metrics: the Structural Similarity Index (SSIM), the Peak Signal-to-Noise Ratio (PSNR), and the recently introduced ꟻLIP metric.

Our analysis demonstrates that the NN algorithm, particularly when implemented with a linear combination of ReLUs as the activation function, provides good performance in image reconstruction. The SSIM values are consistently high, indicating a strong structural similarity with the original images. However, the PSNR values are relatively low, suggesting that a pixel-by-pixel comparison may not fully capture the overall quality of the reconstructed images. The ꟻLIP metric, which aligns better with human perception, shows favorable results, with values close to 0. The best performance for the NN algorithm is observed for $n = 1, 2$ with the ReLU-based activation function. On the other hand, the SK algorithm exhibits overall superior performance. The SSIM values reach good similarity. The PSNR values are also very good, indicating a higher fidelity in pixel-wise reconstruction and denoising. The ꟻLIP metric confirmed these findings, with error maps displaying near-zero discrepancies.

A direct comparison between the best configurations of both methods further emphasize the superiority of the SK algorithm, which achieves better results for both PSNR and ꟻLIP indices, while the SSIM indices are slightly lower than the ones produced by the NN algorithm. The ꟻLIP error maps provide additional insights, revealing that the SK algorithm preserves image details more effectively, while the NN algorithm exhibits errors along the edges.

In conclusion, while NN-based reconstruction techniques show promising results and can achieve reasonable performance with appropriate activation functions, the SK algorithm remains the more effective approach in terms of accuracy and perceptual similarity, especially as denoising algorithm, when dealing with noised images.

Acknowledgments. All the authors have been supported within the project "Mathematical Methods and Models for Biomedical Applications" financed by National Recovery and Resilience Plan PNRR-III-C9-2022-I8.

Disclosure of Interests. The authors declare that they have no conflict of interest and competing interest.

References

1. Acu, A.M., Buscu, I.C., Rasa, I.: Generalized Kantorovich modifications of positive linear operators. Math. Found. Comput. **6**(1), 54–62 (2023)
2. Akhilesh, A.W., Brijesh, K.S.: Performance analysis of sigmoid and ReLU activation functions in deep neural network. In: Intelligent Systems, Proceedings of SCIS (2021)

3. Andersson, P., Nilsson, J., Akenine-Möller, T., Oskarsson, M., Åström, K., Fairchild, M.D.: FLIP: a difference evaluator for alternating images. Proc. ACM Comput. Graph. Interact. Tech. **3**(2), 1–23 (2020)
4. Bardaro, C., Vinti, G., Butzer, P.L., Stens, R.L.: Kantorovich-type generalized sampling series in the setting of Orlicz Spaces. Sampling Theory Sig. Image Process. **6**(1), 29–52 (2007). https://doi.org/10.1007/BF03549462
5. Butzer, P.L., Fischer, A., Stens, R.L.: Generalized sampling approximation of multivariate signals; theory and some applications. Note di Matematica **10**(1), 173–191 (1990)
6. Butzer, P.L., Nessel, R.J.: Fourier Analysis and Approximation I. Academic Press, New York-London (1971)
7. Cao, F., Chen, Z.: The approximation operators with sigmoidal functions. Comput. Math. Appl. **58**(4), 758–765 (2009)
8. Cao, F., Chen, Z.: The construction and approximation of a class of neural networks operators with ramp functions. J. Comput. Anal. Appl. **14**(1), 101–112 (2012)
9. Cardaliaguet, P., Euvrard, G.: Approximation of a function and its derivative with a neural network. Neural Netw. **5**(2), 207–220 (1992)
10. Cheang, G.: Approximation with neural networks activated by ramp sigmoids. J. Approx. Theory **162**, 1450–1465 (2010)
11. Cohen, Y., et al.: Recent advances at the interface of neuroscience and artificial neural networks. J. Neurosci. **42**(45), 8514–8523 (2022)
12. Costarelli, D., Seracini, M., Travaglini, A., Vinti, G.: Alzheimer biomarkers esteem by sampling Kantorovich algorithm. Math. Methods Appl. Sci. **46**(12), 13506–13520 (2023)
13. Costarelli, D., Seracini, M., Vinti, G.: A comparison between the sampling Kantorovich algorithm for digital image processing with some interpolation and quasi-interpolation methods. Appl. Math. Comput. **374**, 125046 (2020)
14. Costarelli, D., Seracini, M., Vinti, G.: A segmentation procedure of the pervious area of the aorta artery from CT images without contrast medium. Math. Methods Appl. Sci. **43**(1), 114–133 (2020)
15. Costarelli, D., Spigler, R.: Approximation results for neural network operators activated by sigmoidal functions. Neural Netw. **44**, 101–106 (2013)
16. Costarelli, D., Spigler, R.: Multivariate neural network operators with sigmoidal activation functions. Neural Netw. **48**, 72–77 (2013)
17. Costarelli, D., Spigler, R.: Approximation by series of sigmoidal functions with applications to neural networks. Annali di Matematica **194**(1), 289–306 (2015)
18. Costarelli, D., Vinti, G.: Approximation by multivariate generalized sampling Kantorovich operators in the setting of Orlicz spaces. Bollettino dell'Unione Matematica Italiana **4**(3), 445–468 (2011)
19. Costarelli, D., Vinti, G.: Convergence for a family of neural network operators in Orlicz spaces. Math. Nachr. **290**(2–3), 226–235 (2017)
20. Costarelli, D., Vinti, G.: Saturation classes for max-product neural network operators activated by sigmoidal functions. RM **72**(3), 1555–1569 (2017). https://doi.org/10.1007/s00025-017-0692-6
21. Costarelli, D., Vinti, G.: Approximation properties of the sampling Kantorovich operators: regularization, saturation, inverse results and Favard classes in L^p-spaces. J. Fourier Anal. Appl. **28**(3), 2–42 (2022)
22. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Sig. Syst. **2**, 303–314 (1989)
23. Goebbels, S.: On sharpness of error bounds for univariate approximation by single hidden layer feedforward neural networks. Results Math. **75**, 109 (2020)

24. Kyurkchiev, N., Markov, S.: Sigmoid functions: some approximation and modelling aspects, some moduli in programming environment mathematica. LAP Lambert Acad. Publ. **4**, 34 (2015)
25. Nessel, R.J.: Contributions to the theory of saturation for singular integrals in several variables. III Radial kernels. Indagationes Math. Proc. **70**, 65–73 (1967)
26. Paul, A.K., Prasad, A., Kumar, A.: Review on artificial neural network and its application in the field of engineering. J. Mech. Eng. Prakash **1**(1), 53–61 (2022)
27. Rinelli, L., Travaglini, A., Vescera, N., Vinti, G.: An approximation-based approach versus an AI one for the study of CT images of abdominal aorta aneurysms (2024)
28. Travaglini, A., Vinti, G., Scalera, G.B., Scialpi, M.: A large scale analysis for testing a mathematical model for the study of vascular pathologies. Mathematics **11**(8), 1831 (2023)
29. Travaglini, A., Vinti, G.: A mathematical model for the analysis of eye fundus images in healthy and diabetic patients. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications – ICCSA 2023 Workshops. ICCSA 2023, LNCS, vol. 14108, pp. 558–576. Springer, Cham (2023)
30. Vinti, G., Zampogni, L.: Approximation by means of nonlinear Kantorovich sampling type operators in Orlicz spaces. J. Approx. Theory **161**, 511–528 (2009)
31. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
32. Zarate, G., Pedrycz, W., Moreno-Garcia, J., Sánchez-Solís, J.: Innovative Applications of Artificial Neural Networks to Data Analytics and Signal Processing. Studies in Computational Intelligence, Springer, Cham (2024)