Advanced Mathematical Models & Applications
Vol.4, No.3, 2019, pp.243-254

RESTORATION OF IMAGES ON GRAYSCALE USING A RECURRENT CUBIC DISCRETE NEURAL NETWORK

Franco Rubio López*, **Orlando Hernández Bracamonte**, **Ronald León Navarro**

Department of Mathematics, University of Trujillo, Trujillo, Perú

Abstract. In this paper it build a recurrent cubic discrete neural network from the fixed points attractors of cubic polynomials, and we use it in the restoration of grayscale images. The goal is to provide a criterion for the assignment of values to the synaptic weights of the neural network; which will guarantee the stability of our neural network.

Keywords: Image restoration, cubic recurrent discrete neural network, attractor fixed point, stability.

AMS Subject Classification: 37D05, 37D25, 37D40, 37D45.

Corresponding author: Franco Rubio López, Department of Mathematics, National University of Trujillo, Avenue Juan Pablo II S/N , Trujillo, Perú, e-mail: *frubio@unitru.edu.pe*

Received: 13 August 2019; Revised: 18 September 2019; Accepted: 21 October 2019;
Published: 21 December 2019.

1 Introduction

The restoration and reconstruction of images is an important area within image processing; that in recent years has a variety of applications in different areas.

Currently, there are different methods to perform the processes of restoration and reconstruction of images; those based on neural networks are among them. Since 1943, when the first mathematical model was developed (McCulloch & Pitts, 1943), until the present date, there are different types of artificial neural networks. An artificial neural network is a mathematical model which serves for the study of nervous systems of living beings. This artificial neural network has an important property that is the ability to acquire and store information.

In the 80s (Hopfield, 1982), it presents a new model of a recurrent discrete neural network, which was an associative memory, and would serve to study different processes: physical, learning, memory, etc., from another point of view , in contrast to the theories that explained the processes of learning and memory (Hebb, 1949). Interest in the scientific community continues until this day; since it allowed to create a new area within neural networks.

Hopfield neural networks were used for image restoration and reconstruction (Zhou et al., 1988), who were the first to use a Hopfield network for restoration; and showed the instability of Hopfield's neuronal network in this process. This situation motivated them to propose an algorithm that allowed to correct this behavior of the network, guaranteeing the stability of the neural network.

A new neural network called Modified Hopfield Neural Network (Paik, 1992) was proposed to restore grayscale images. Other models based on Hopfield neural networks, both discrete and continuous, were used for the restoration and reconstruction of images (Sun et al., 1992; Liu, 1993; Sun et al., 1995; Joudar et al., 2015).

In this paper, it build a new discrete neural network from the fixed points given a priori of cubic polynomial functions (Rubio et al, 2015 - 2017). The goal is to give a rule for the assignment of values to the synaptic weights of the neural network in order to guarantee the stability of the neural network. This neural network is used for the restoration of grayscale images.

This paper use discrete polynomial neural networks (Rubio & Hernándes, 2017a), to construct a discrete neural network based on cubic polynomials (Rubio & Hernándes, 2015), and vector functions (Rubio & Hernándes, 2017b).

2 Cubic Polynomial

The result (Rubio & Hernándes, 2015), in which the points $x_0, x_1, x_2 \in \mathbb{R}$, $x_0 < x_1 < x_2$, are given as fixed points a priori, and a cubic polynomial is determined by

$$f(x) = Ax^3 + Bx^2 + Cx + D \tag{1}$$

with

$$f(x_i) = x_i$$

for all $i = 0, 1, 2$, where

$$A = \frac{-(y_m - x_m)}{(x_0 - x_m)(x_1 - x_m)(x_2 - x_m)} \tag{2}$$

$$B = \frac{-(x_m - y_m)(x_0 + x_1 + x_2)}{(x_0 - x_m)(x_m - x_1)(x_m - x_2)} \tag{3}$$

$$C = -\frac{-x_0x_1\ x_2 + x_0x_1y_m + x_0x_2y_m - x_0x_m^2 + x_1x_2y_m - x_1x_m^2 - x_2x_m^2 + x_m^3}{(x_0 - x_m)(x_m - x_1)(x_m - x_2)} \tag{4}$$

$$D = \frac{-x_0x_1x_2(x_m - y_m)}{(x_0 - x_m)(x_m - x_1)(x_m - x_2)}. \tag{5}$$

The point (x_m, y_m) is given, such that (x_0, x_0), (x_1, x_1), (x_2, x_2) y (x_m, y_m) are not collinear.

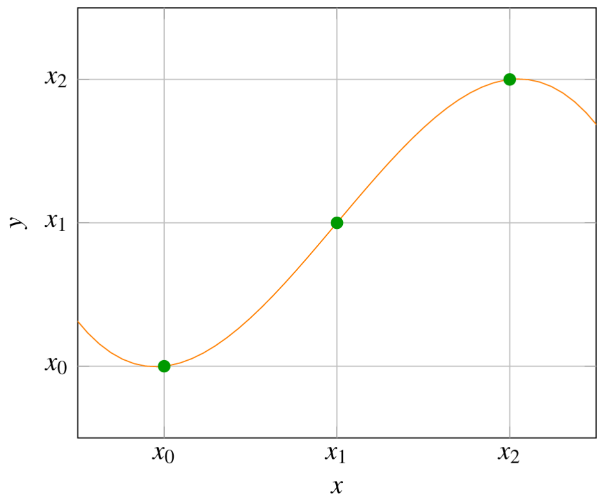


Figure 1: Cubic polynomial with fixed points

Theorem 1. *Let* $\varepsilon \in \mathbb{R}$, $0 < \varepsilon < 1/2$. *Then,*

$$-2\varepsilon + \sqrt{2\varepsilon^2 + 1} < 1 - \varepsilon. \tag{6}$$

Proof. As $0 < \varepsilon < \frac{1}{2}$, then: $0 < \varepsilon^2 < \frac{1}{4}$,

$$0 < 2\varepsilon^2 < \frac{1}{2} \quad ,$$

$$1 < 2\varepsilon^2 + 1 < \frac{3}{2} \quad ,$$

$$1 < \sqrt{2\varepsilon^2 + 1} < \sqrt{\frac{3}{2}} \, . \tag{7}$$

Furthermore, by (7), we obtain:

$$-2\varepsilon + \sqrt{2\varepsilon^2 + 1} - 1 + \varepsilon = \sqrt{2\varepsilon^2 + 1} - 1 - \varepsilon < 0.$$

Therefore:

$$-2\varepsilon + \sqrt{2\varepsilon^2 + 1} < 1 - \varepsilon.$$

□

Theorem 2. *Let* $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$, *then,*

$$-2\varepsilon - \sqrt{2\varepsilon^2 + 1} < -1. \tag{8}$$

Proof. As

$$-2\varepsilon - \sqrt{2\varepsilon^2 + 1} + 1 = -\left(2\varepsilon + \sqrt{2\varepsilon^2 + 1}\right) + 1. \tag{9}$$

and $\varepsilon > 0 : \; 1 < 1 + 2\varepsilon^2$,

$$1 < \sqrt{2\varepsilon^2 + 1} < \sqrt{2\varepsilon^2 + 1} + 2\varepsilon. \tag{10}$$

By (10) in (9):

$$-2\varepsilon - \sqrt{2\varepsilon^2 + 1} < -1.$$

□

Theorem 3. *Let* $x_0 = -1, \quad x_2 = 1, \quad x_m = x_1 + \varepsilon, \quad y_m = x_1, \quad \varepsilon \in \mathbb{R}, \; 0 < \varepsilon < \frac{1}{2}$ *and* $f(x) = Ax^3 + Bx^2 + Cx + D$ *given by* (1). *Then,*

$$\left|f'(x_1)\right| < 1 \iff x_1 \in \left\langle \; -1 \; ; \; -2\varepsilon + \sqrt{2\varepsilon^2 + 1} \; \right\rangle. \tag{11}$$

Proof. As $x_0 = -1$, $x_2 = 1$, $x_m = x_1 + \varepsilon$, $y_m = x$, using (2) - (5):

$$A = \frac{-1}{(1 + x_1 + \varepsilon)(x_1 + \varepsilon - 1)} \tag{12}$$

$$B = \frac{x_1}{(1 + x_1 + \varepsilon)(x_1 + \varepsilon - 1)} \tag{13}$$

$$C = \frac{x_1^2 + 2\varepsilon x_1 + \varepsilon^2}{(1 + x_1 + \varepsilon)(x_1 + \varepsilon - 1)} \tag{14}$$

$$D = \frac{-x_1}{(1 + x_1 + \varepsilon)(x_1 + \varepsilon - 1)}. \tag{15}$$

Moreover,

$$f'(x_1) = 3Ax_1^2 + 2Bx_1 + C. \tag{16}$$

Using (12) - (15) in (16), it is obtained:

$$f'(x_1) = \frac{2\varepsilon x_1 + \varepsilon^2}{(1 + x_1 + \varepsilon)(x_1 + \varepsilon - 1)} \tag{17}$$

Therefore: $|f'(x_1)| = \dfrac{|2\varepsilon x_1 + \varepsilon^2|}{(1 + x_1 + \varepsilon)(1 - x_1 - \varepsilon)}$, for $-1 < x_1 < 1$, $x_1 + \varepsilon < 1$.

1. If $|f'(x_1)| < 1$, then
$$\left|2\varepsilon x_1 + \varepsilon^2\right| < 1 - x_1^2 - 2\varepsilon x_1 - \varepsilon^2, \tag{18}$$
where $1 - x_1^2 - 2\varepsilon x_1 - \varepsilon^2 > 0$, with solution set:
$$U = \langle -1-\varepsilon;\ \ 1-\varepsilon \rangle\,. \tag{19}$$
Solving the inequality (18) with respect to (19):
   (a) $-1 + x_1^2 + 2\varepsilon x_1 + \varepsilon^2 < 2\varepsilon x_1 + \varepsilon^2$. Thus,
$$x_1 \in \langle -1;\ 1 \rangle\,. \tag{20}$$
   (b) $2\varepsilon x_1 + \varepsilon^2 < 1 - x_1^2 - 2\varepsilon x_1 - \varepsilon^2$,
$$x_1^2 + 4\varepsilon x_1 + 2\varepsilon^2 - 1 < 0.$$
Thus,
$$x_1 \in \left\langle -2\varepsilon - \sqrt{2\varepsilon^2+1}\ ;\ \ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle. \tag{21}$$
Using (6), (8), (19), (20), (21), it is obtained the following:
$$x_1 \in \left\langle -1\ ;\ \ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle.$$

2. Conversely, if $x_1 \in \left\langle -1\ ;\ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle$, and as $0 < \varepsilon < \frac{1}{2}$, then:
$$\left\langle -1;\ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle = \left\langle -2\varepsilon - \sqrt{2\varepsilon^2+1}\ ;\ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle \cap \langle -1;\ 1 \rangle \cap \langle -1-\varepsilon, ; 1-\varepsilon \rangle$$
where:
   (a) $\left\langle -2\varepsilon - \sqrt{2\varepsilon^2+1}\ ;\ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle$ is the solution of the inequality:
$$2\varepsilon x_1 + \varepsilon^2 < 1 - x_1^2 - 2\varepsilon x_1 - \varepsilon^2.$$
   (b) $\langle -1;\ \ 1 \rangle$ is the solution of the inequality:
$$-1 + x_1^2 + 2\varepsilon x_1 + \varepsilon^2 < 2\varepsilon x_1 + \varepsilon^2.$$
   (c) $\langle -1-\varepsilon;\ 1-\varepsilon \rangle$ is the solution of the inequality:
$$1 - x_1^2 - 2\varepsilon x_1 - \varepsilon^2 > 0.$$
Such that:
$$\left|2\varepsilon x_1 + \varepsilon^2\right| < 1 - x_1^2 - 2\varepsilon x_1 - \varepsilon^2$$
$$|2\varepsilon x_1 + \varepsilon^2| < (1 + x_1 + \varepsilon)(1 - x_1 - \varepsilon)$$
Therefore, $|f'(x_1)| < 1$. ☐

Theorem 4. *Let $\varepsilon \in \mathbb{R}$, $0 < \varepsilon < \frac{1}{2}$, and $x_0 = -1$, x_1, $x_2 = 1$ fixed points of $f(x) = Ax^3 + Bx^2 + Cx + D$. If $x_1 \in \left\langle -1;\ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle$, then x_1 is a fixed point attractor of f.*

Proof. Using the theorem (3), with $x_1 \in \left\langle -1;\ \ -2\varepsilon + \sqrt{2\varepsilon^2+1}\ \right\rangle$, then $|f'(x_1)| < 1$. Therefore, x_1 is a fixed point attractor. ☐

3 Building a Neural Network

The construction of a discrete neural network using cubic polynomials, it is given by (1). The discrete neural network is given by the mapping:

$$\begin{aligned} F: \ \mathbb{R}^n &\longrightarrow \ \mathbb{R}^n \\ x &\longmapsto \ F(x) = (F_1(x),\ F_2(x),\ \ldots,\ F_n(x)) \end{aligned} \tag{22}$$

where

$$F_i(x) = A_i\left(\sum_{j=1}^{n} w_{ij}x_j\right)^3 + B_i\left(\sum_{j=1}^{n} w_{ij}x_j\right)^2 + C_i\left(\sum_{j=1}^{n} w_{ij}x_j\right) + D_i\ , \tag{23}$$

$A_i,\ B_i,\ C_i,\ D_i$ for all $i = 1, \ldots, n$ are constant.

Now, let $X_p = \left(x_p^1\ ,\ x_p^2\ ,\ \ldots,\ x_p^n\right) \in \mathbb{R}^n$ be such that:

$$f_i\left(x_p^i\right) = x_p^i, \quad \forall i = 1, \ldots, n, \tag{24}$$

where $f_i(y) = A_i y^3 + B_i y^2 + C_i y + D_i\ ,\ \ \forall i = 1, \ldots, n$, is given by (1).

Theorem 5. *Let $X_p = (x_p^1,\ \ldots,\ x_p^n) \in \mathbb{R}^n$ be such that $f_i\left(x_p^i\right) = x_p^i\ ,\ \forall i = 1, \ldots, n$. X_p is a fixed point of $F(x)$ if and only if*

$$\sum_{j=1}^{n} w_{ij}x_p^j = x_p^i, \quad \forall\ i = 1, \ldots, n. \tag{25}$$

Proof. If $F(X_p) = X_p$, then from (23) and (24):

$$A_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right)^3 + B_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right)^2 + C_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right) + D_i = x_p^i, \qquad \forall i = 1, \ldots, n.$$

$$= f_i\left(x_p^i\right) = A_i\left(x_p^i\right)^3 + B_i\left(x_p^i\right)^2 + C_i\left(x_p^i\right) + D_i.$$

Then, $\sum_{j=1}^{n} w_{ij}x_p^j = x_p^i\ ,\quad \forall\ i = 1, \ldots, n$. (Rubio et al., 2015). Conversely, if $\sum_{j=1}^{n} w_{ij}x_p^j = x_p^i\ ,\ \ \forall\ i = 1, \ldots, n$; then

$$\begin{aligned} F_i(X_p) &= A_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right)^3 + B_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right)^2 + C_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right) + D_i \\ &= A_i\left(x_p^i\right)^3 + B_i\left(x_p^i\right)^2 + C_i\left(x_p^i\right) + D_i \\ &= f_i\left(x_p^i\right) = x_p^i\ , \qquad \forall\ i = 1, \ldots, n. \end{aligned}$$

$\therefore \qquad F(X_p) = X_p.$ $\square$

From equation (25), it arise the following system:

$$\left\{ \begin{array}{c} w_{11}x_p^1 + w_{12}x_p^2 + \cdots + w_{1n}x_p^n = x_p^1 \\ w_{21}x_p^1 + w_{22}x_p^2 + \cdots + w_{2n}x_p^n = x_p^2 \\ \vdots \\ w_{n1}x_p^1 + w_{n2}x_p^2 + \cdots + w_{nn}x_p^n = x_p^n \end{array} \right.$$

whose associated matrix is:

$$W = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1n} \\ w_{21} & w_{22} & \ldots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{nn} \end{pmatrix} \tag{26}$$

Furthermore, from (25):

$$w_{ii} = 1 - \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij} \left(\frac{x_p^j}{x_p^i} \right) , \qquad x_p^i \neq 0. \tag{27}$$

4 Restoration of Images

In this section, we use the discrete cubic neural network constructed in the previous section. For grayscale images, in the real case, the pixels take values in the interval $[0;1]$. In this sense, each component of the discrete neural network given by (22), will be built on the interval $[-1,1]$, with $x_0 = -1, x_2 = 1$ and $x_1 \in \langle -1;1 \rangle$, where x_1 is an attractor fixed point, and x_0, x_2 are repellent fixed points.

Following the methodology (Rubio & Hernándes, 2017a), it will give the rule to assign values to the synaptic weights of the discrete cubic neural network, in order to guarantee the stability of the network at the fixed point.

Let $X_p \in \mathbb{R}^n, X_p = (x_p^1, x_p^2, ..., x_p^n)$ be such that

$$x_p^i \in [-1;1], x_p^i \neq 0, \, \forall i = 1,2,...,n \tag{28}$$

$$M = \sum_{i=1}^{n} |x_p^i|, \, h = \frac{1}{M} \tag{29}$$

Then:

1. $If - \dfrac{x_p^j}{x_p^i} > 0$ then $w_{ij} = -h, i \neq j.$ (30)

2. $If - \dfrac{x_p^j}{x_p^i} < 0$ then $w_{ij} = h, i \neq j.$ (31)

Theorem 6. *Let $X_p = (x_p^1, x_p^2, ..., x_p^n) \in \mathbb{R}^n$ be such that $0.5 \leq |x_p^i| < 1, \forall i = 1,2,...,n$. Then*

$$-1 \leq 1 - \frac{1}{|x_p^i|} < 0 \tag{32}$$

Proof. Since $0.5 \leq |x_p^i| < 1$, then

$$\frac{1}{2} \leq |x_p^i| \qquad and \qquad |x_p^i| < 1$$

$$\frac{1}{|x_p^i|} \leq 2 \qquad and \qquad 1 < \frac{1}{|x_p^i|} \tag{33}$$

By (33)

$$1 < \frac{1}{|x_p^i|} \leq 2,$$

$$-1 \leq 1 - \frac{1}{|x_p^i|} < 0.$$

□

Theorem 7. *Let $X_p = (x_p^1, x_p^2, ..., x_p^n) \in \mathbb{R}^n$ be such that $0.5 \leq |x_p^i| < 1, \forall i = 1,2,...,n$. Then*

$$1 - \frac{1}{|x_p^i|} + \frac{1}{M} \leq w_{ii} < 1 \tag{34}$$

Proof. By (29) $M = \sum_{i=1}^{n} |x_p^i|$.
From (30) and (31),

$$-\frac{x_p^j}{x_p^i} w_{ij} < 0.$$

Then

$$-\sum_{j=1, j\neq i}^{n} \frac{x_p^j}{x_p^i} w_{ij} < 0,$$

$$0 < \sum_{j=1, j\neq i}^{n} \frac{x_p^j}{x_p^i} w_{ij} \leq \sum_{j=1, j\neq i}^{n} |w_{ij}| \frac{|x_p^j|}{|x_p^i|}$$

$$= \sum_{j=1, j\neq i}^{n} \frac{|x_p^j|}{M|x_p^i|} = \frac{1}{M|x_p^i|} \sum_{j=1, j\neq i}^{n} |x_p^j|$$

$$= \frac{1}{M|x_p^i|}(M - |x_p^i|) = \frac{1}{|x_p^i|} - \frac{1}{M}$$

Thus

$$0 < \sum_{j=1, j\neq i}^{n} \frac{x_p^j}{x_p^i} w_{ij} \leq \frac{1}{|x_p^i|} - \frac{1}{M} \tag{35}$$

From (35),

$$0 > -\sum_{j=1, j\neq i}^{n} \frac{x_p^j}{x_p^i} w_{ij} \geq \frac{1}{M} - \frac{1}{|x_p^i|}$$

$$1 > 1 - \sum_{j=1, j\neq i}^{n} \frac{x_p^j}{x_p^i} w_{ij} \geq 1 - \frac{1}{|x_p^i|} + \frac{1}{M}$$

$$1 - \frac{1}{|x_p^i|} + \frac{1}{M} \leq 1 - \sum_{j=1, j\neq i}^{n} \frac{x_p^j}{x_p^i} w_{ij} < 1.$$

By using (27),

$$1 - \frac{1}{|x_p^i|} + \frac{1}{M} \leq w_{ii} < 1.$$

□

Theorem 8. *Let $X_p = (x_p^1, x_p^2, ..., x_p^n) \in \mathbb{R}^n$ be such that $0.5 \leq |x_p^i| < 1, \forall i = 1, 2, ..., n$. Then*

$$-1 \leq 1 - \frac{1}{|x_p^i|} + \frac{1}{M} \leq w_{ij} < 1 \tag{36}$$

Proof. From (32),

$$-1 \leq 1 - \frac{1}{|x_p^i|} < 0,$$

Then

$$-1 \leq 1 - \frac{1}{|x_p^i|} < 1 - \frac{1}{|x_p^i|} + \frac{1}{M}$$

By (34)

$$-1 \leq 1 - \frac{1}{|x_p^i|} + \frac{1}{M} \leq w_{ii} < 1.$$

□

Theorem 9. *Let $X_p = (x_p^1, x_p^2, ..., x_p^n) \in \mathbb{R}^n$ be such that $0.5 \leq |x_p^i| < 1, \forall i = 1, 2, ..., n, W = (w_{ij})_{n\times n}$, where $w_{ij}, \forall i, j = 1, ..., n$, are given by (30) or (31). Then,*

$$\|W\|_\infty \leq 1 + \frac{n-1}{M} \tag{37}$$

Proof. We have

$$\begin{aligned} \sum_{j=1}^{n} |w_{ij}| &= |w_{ii}| + \sum_{j=1, j\neq i}^{n} |w_{ij}| = |w_{ii}| + \sum_{j=i, j\neq i}^{n} \frac{1}{M}, \qquad \forall i = 1, ..., n \\ &= |w_{ii}| + \frac{1}{M}(n-1) < 1 + \frac{1}{M}(n-1), \qquad \forall i = 1, ..., n \end{aligned}$$

Thus

$$\sum_{j=1}^{n} |w_{ij}| < 1 + \frac{1}{M}(n-1), \qquad \forall i = 1, ..., n,$$

and since

$$\|W\|_\infty = max\{\sum_{j=1}^{n} |w_{ij}|/i = 1, ..., n\} < 1 + \frac{1}{M}(n-1)$$

Therefore

$$\|W\|_\infty < 1 + \frac{n-1}{M}.$$

$\square$

5 Stability

In this section the proof that establishes the stability of the discrete cubic neural network $F(x)$ is given.

By (23), the components of $F(x)$ are given by:

$$F_i(x) = A_i \left(\sum_{j=1}^{n} w_{ij}x_j\right)^3 + B_i \left(\sum_{j=1}^{n} w_{ij}x_j\right)^2 + C_i \left(\sum_{j=1}^{n} w_{ij}x_j\right) + D_i, \qquad \forall i = 1, ..., n.$$

Thus, the mapping $F(x) = (F_1(x), ..., F_n(x))$ is differentiable of class $C^\infty(\mathbb{R}^n)$.

$$\frac{\partial F_i(x)}{\partial x_k} = 3A_i \left(\sum_{j=1}^{n} w_{ij}x_j\right)^2 w_{ik} + 2B_i \left(\sum_{j=1}^{n} w_{ij}x_j\right) w_{ik} + C_i w_{ik}, \qquad \forall k = 1, ..., n.$$

$$\frac{\partial F_i(x)}{\partial x_k} = \left(3A_i \left(\sum_{j=1}^{n} w_{ij}x_j\right)^2 + 2B_i \left(\sum_{j=1}^{n} w_{ij}x_j\right) + C_i\right) w_{ik} \tag{38}$$

Therefore, from (38) the Jacobian matrix of F in x is

$$JF(x) = \left(\left(3A_i \left(\sum_{j=1}^{n} w_{ij}x_j\right)^2 + 2B_i \left(\sum_{j=1}^{n} w_{ij}x_j\right) + C_i\right) w_{ik}\right)_{n\times n} \tag{39}$$

Now, let $X_p = (x_p^1, ..., x_p^n) \in \mathbb{R}^n$ a fixed point given apriori, with atractor fixed points $x_p^i, \forall i = 1, .., n$, of the functions $f_i(x)$ given by (1). In the following result we show that the norm of the Jacobian matrix at the point X_p is bounded by the norm of the synaptic weigth matrix W; and using the theorem (9), the stability of the discrete cubic neural network is assured.

Theorem 10. *Let* $X_p = (x_p^1, x_p^2, ..., x_p^n) \in \mathbb{R}^n$ *be such that* $0.5 \leq |x_p^i| < 1, f_i\left(x_p^i\right) = x_p^i \ \forall\ i = 1, ..., n, w_{ij}$ *are given by (27), (30) or (31),* $\forall\ i, j = 1, ..., n, \varepsilon \in \mathbb{R},\ 0 < \varepsilon < \frac{1}{2}$. *Then,*

$$\|JF(X_p)\|_\infty < \|W\|_\infty \tag{40}$$

Proof. From (39),

$$\begin{aligned}
\sum_{k=1}^{n} \left|\frac{\partial F_i(X_p)}{\partial x_k}\right| &= \sum_{k=1}^{n} \left|\left(3A_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right)^2 + 2B_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right) + C_i\right) w_{ik}\right| \\
&= \sum_{k=1}^{n} \left|\left(3A_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right)^2 + 2B_i\left(\sum_{j=1}^{n} w_{ij}x_p^j\right) + C_i\right)\right| |w_{ik}| \\
&\sum_{k=1}^{n} \left|3A_i(x_p^i)^2 + 2B_i x_p^i + C_i\right| |w_{ik}| \\
&= \sum_{k=1}^{n} \left|f_i'(x_p^i)\right| |w_{ik}| \\
&< \sum_{k=1}^{n} |w_{ij}|
\end{aligned}$$

Therefore, $\|JF(X_p)\|_\infty \leq \|W\|_\infty$. $\square$

Now, we consider a grayscale image of size $n \times n$ pixels, where each pixel is in the interval of real numbers $[0; 1]$. The image will be represented matrix by $I = (a_{jk})$ of dimension $n \times n$. The following are the steps to follow for the application of network to the restoration of images.

1. Transform I into a vector $X_I \in \mathbb{R}^L, L = n \times n$, through:

$$X_I(m) = I(j, k) \tag{41}$$

   where $m = n(j-1) + k$.

2. Using the function:

$$f(x) = \frac{x}{2} + \frac{1}{2}, \tag{42}$$

   for each component of X_I, we transform X_I into X_p, where:

$$x_p^j \in [0.5; 1], \forall j = 1, ..., L. \tag{43}$$

   it is necessary for the utilization of the theorems (6-10).

3. Now, using X_p as a fixed point given apriori for the construction of the recurrent cubic discrete neural network (23).

4. Using the algorithm of fixed point, with the starting point X_0, it gets an approximation X_{ap} of X_p.

5. Using the inverse function of (42):

$$f^{-1}(x) = 2x - 1, \tag{44}$$

   for each component of X_{ap} and following the inverse process of (41), the restored image is obtained I_{ap}.

6 Computer Simulation

In order to compare the performance of our Recurrent Cubic Discrete Neural Network, in restoring grayscale images, the Wiener restoration method was used. The experiment consisted of restoring a perturbed image, obtained by the use of additive Gaussian noise of zero mean and variance σ , where $0.0001 \leq \sigma \leq 0.2$; applied to a noise free image I. The Euclidean norm was used to estimate the error in the approximation; which is given by:

$$Error = \|X_p - X\|, \qquad X_p, X \in \mathbb{R}^L,\ L = n \times n.$$

where:

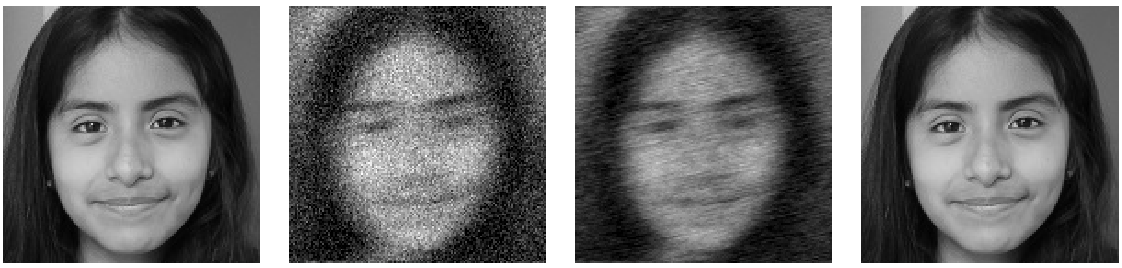
- X_p is the noise-free image.
- X is the perturbed image or restored image.

A noise-free image I of dimension 140×140 pixels in grayscale was chosen, and using the methodology of the previous section, we obtain $X_p \in \mathbb{R}^L, L = 19600$.

The parameters used in the cubic recurrent discrete neural network are: $x_0 = -1, x_2 = 1, tol = 0.1, \varepsilon = 0.1$; where tol is the parameter used in the fixed point algorithm.

For each value of the variance, we observe that the best image restored is that obtained by the Recurrent Cubic Discrete Neural Network, which is closer to the noise-free image X_p; compared to the restored image obtained by the Wiener method X_w.

The next figures show some results.



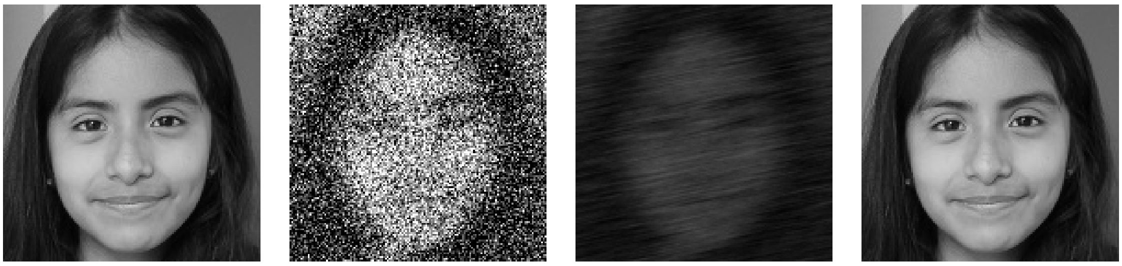
(a) Noise-free image

(b) Noisy image perturbed by Gaussian additive noise of zero mean and variance $\sigma = 0.01$.

(c) Restore image with Wiener method.

(d) Restored image using our Recurrent Cubic Discrete Neural Network.

Figure 2: Results by using Recurrent Cubic Discrete Neural Network



(a) Noise-free image (b) Noisy image perturbed by Gaussian additive noise of zero mean and variance $\sigma = 0.1$. (c) Restore image with Wiener method. (d) Restored image using our Recurrent Cubic Discrete Neural Network.

Figure 3: Results by using Recurrent Cubic Discrete Neural Network

The next table shows other results obtained by applying the Wiener method and the Recurrent Cubic Discrete Neural Network. It is observed that the error obtained using the neural network is close to zero, compared to the error obtained by the Wiener method.

Table 1: Numerical results of the restored images using the Wiener method and our Recurrent Cubic Discrete Neural Network

Variance σ	Error between noise-free image X_p and perturbed image X_d $\|X_p - X_d\|$	Error using Wiener's method $\|X_p - X_w\|$	Error using our Neural Network $\|X_p - X_r\|$
0.0001	5.4553	6.6518	0.0038
0.001	5.8912	5.6233	0.0046
0.01	8.6395	7.1026	0.0141
0.05	14.9775	15.2618	0.0840
0.1	19.1541	19.8580	0.0016
0.2	23.8229	23.6962	0.0033

7 Conclusion

In this paper it is propose a new neural network for restoration of grayscale images, based on cubic polynomial neural networks. The algorithm developed improves the suppression of deformations in the image, conserving the geometric characteristics of the image.

A criterion is given for the assignment of values to the synaptic weights of the neural network; which helps to prove the stability of the neural network from another point of view, without using the Hopfield energy function.

References

Abruck, J. (1990). On the convergence properties of the Hopfield model, *Proc.IEEE*, *78*, 1579-1585.

Demoment, G. (1989). Image reconstruction and restoration: Overview of common estimation structures and problems, *IEEE Trans. Acoust., Speech, Signal Processing*, *37*, 2024–2036.

Hebb, D. O. (1949). *The Organization of Behavior*. New York: Willey.

Hopfield J. (1982). Neural Networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, *79*, 2554-2558.

Hopfield, J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons, *Proc. Natl. Acad. Sci.*, *81*, 3088-3092.

Joudar, N., El Moutouakil, K., Ettaouil, M. (2015). An original Continuous Hopfield Network for optimal images restoration. *WSEAS, Transaction on Computer,* 14, 668-678.

Liu, H-J., Sun, Y. (1993). Blind bilevel image restoration using Hopfield neural networks, *In Proc. IEEE Int. Conf. Neural Networks, San Francisco, CA*, 1656–1661.

McCulloch, W., Pitts, W. (1943). A logical calculus of the ideas immanent in neurons activity, *Bulletin Mathematical Biophysics*, *5*, 115-133.

Paik, J.K., Katsaggelos, A.K. (1992). Image restoration using a modified Hopfield network, *IEEE Trans. Image Processing*, *1*, 49–63.

Rubio, F., Hernández, O. (2016). Construcción de una función polinómica a partir de los puntos fijos dados previamente. *Selecciones Matemáticas*, *2*(01), 54-67.

Rubio, F., Hernández, O. (2017a). Stability of a new Recurrent Quadratic Neural Network, *Advanced Math. Models & Applications*,*2*(02), 97-106.

Rubio, F., Hernández, O. (2017b). Construcción de una función Vectorial a partir de un punto fijo dado previamente. *Selecciones Matemáticas, 4*(01), 124-138.

Rubio, F., Hernández, O. (2017c). Stability of a new Polynomial Discrete Recurrent Neural Network, *Advanced Math. Models & Applications*, *2*(3), 229-239.

Sun, Y., Li, J-G., Yu, S-Y. (1995). Improvement on performance of modified Hopfield neural network for image restoration, *IEEE Trans. Image Processing*, *4*, 688-692.

Sun, Y., Yu, S-Y. (1992). A modified Hopfield neural network used in bilevel image restoration and reconstruction, *In Proc. Int. Symp.Information Theory Application*, *3*, 1412-1414.

Zhou, Y-T., Chellappa, R., Vaid, A., Jenkins, B.K. (1988). Image restoration using a neural network, *IEEE Trans. Acoust., Speech, Signal Processing*, *36*(7), 1141-1151.