A SURVEY ON WAVELET NETWORK, MULTI LIBRARY WAVELET NETWORK TRAINING, 1D-2D FUNCTION APPROXIMATION AND A NEW IMAGE COMPRESSION METHOD

Wajdi Bellil [1), Chokri Ben Amar [2), Adel M.Alimi [3)

[1) Faculty of sciences, University of Gafsa, City Zarroug, Gafsa, Tunisia wajdi.bellil@ieee.org
[2) Department of Electrical Engineering, University of Sfax, Tunisia Chokri.benamar@ieee.org
[3) Department of Electrical Engineering, University of Sfax, Tunisia Adel.Alimi@ieee.org

Abstract: *This paper presents an original architecture of Wavelet Neural Network (WNN) based on multi Wavelets activation function and uses a selection method to determine a set of best wavelets whose centers and dilation parameters are used as initial values for subsequent training library WNN for color image compression and coding which consists to transform an RGB image into Luminance-Chrominance space and then segment the luminance in a set of m blocks n by n pixels. These blocks should be transferred row by row (1D input vector) to the input of our wavelet network. Every input vector will be considered as unknown functional mapping and then it will be approximated by the network.*

Keywords: *Wavelet Neural Network, Multi Library Wavelet Neural Network, Image compression and coding, Beta wavelets.*

1. INTRODUCTION

Wavelet Neural Networks (WNN) were introduced by Zhang and Benveniste [1-3] in 1992 as a combination of artificial neural networks and wavelet decomposition. WNN have recently attracted great interest, because of their advantages over radial basis function networks (RBFN) as they are universal approximators but achieve faster convergence and are capable of dealing with the so-called "curse of dimensionality." In addition, WNN are generalized RBFN. However, the generalization performance of WNN trained by least-squares approach deteriorates when outliers are present.

Feed forward neural networks such as multilayer perceptrons (MLP) and radial basis function networks (RBFN) have been widely used as an alternative approach to functions approximation since they provide a generic black-box functional representation and have been shown to be capable of approximating any continuous function defined on a compact set in Rn with arbitrary accuracy [4]. Following the concept of locally supported basis functions such as RBFN, a class of wavelet neural networks (WNN) which originate from wavelet decomposition in signal processing has become more popular lately [5, 6, 7, 8, 9]. In addition to the salient feature of approximating any non-linear function, WNN outperforms MLP and RBFN due to

its capability in dealing with the so-called "curse of dimensionality" and non-stationary signals and in faster convergence speed [10]. It has also been shown that RBFN is a special case of WNN.

This paper comprises four sections. Section 2 discusses the architecture of Multi Library Wavelet Neural Networks (MLWNN) and a new training algorithm based on selection. Section 3 contributes to Beta MLWNN and its performance function approximation. Section 4 presents a direct solution method based on wavelet networks for lossless color image compression. Finally, Section 5 gives conclusions and summary for present research work and other possibilities of future research directions.

2. THEORETICAL BACKGROUND

2.1 CLASSICAL WAVELET NEURAL NETWORK ARCHITECTURE

Wavelets occur in family of functions and each is defined by dilation a_i which controls the scaling parameter and translation t_i which controls the position of a single function, named the mother wavelet $\psi(x)$. Mapping functions to a time-frequency phase space, WNN can reflect the time-frequency properties of function. Given an n-element training set, the overall response of a WNN is:

$$\hat{y}(w) = w_0 + \sum_{i=1}^{N_p} w_i \, \Psi_i \, , where \; \Psi_i = \Psi\left(\frac{x - t_i}{a_i}\right) \qquad (1)$$

where Np is the number of wavelet nodes in the hidden layer and wi is the synaptic weight of WNN.

This can also be considered as the decomposition of a function in a weighted sum of wavelets, where each weight w_j is proportional to the wavelet coefficient scaled and shifted by ai and ti. This establishes the idea for wavelet networks [11, 12].

This network can be considered composed of three layers: a layer with Ni inputs, a hidden layer with N_p wavelets and an output linear neuron receiving the weighted outputs of wavelets. Both input and output layers are fully connected to the hidden layer.

2.2 MULTI LIBRARY WAVELET NEURAL NETWORK ARCHITECTURE

A MLWNN can be regarded as a function approximator which estimates an unknown functional mapping:

$$y = f(x) + \varepsilon \qquad (2)$$

where *f* is the regression function and the error term ε is a zero-mean random variable of disturbance. Constructing a *MLWNN* involves two stages: First, we should construct a wavelet library $W=\{W_1, W_2,...,W_n\}$ of discretely dilated and translated versions of some mothers wavelets function $\Psi_1, \Psi_2,..., \Psi_n$:

$$W_j = \left\{ \begin{array}{l} \Psi_i^{\,j} : \Psi_i^{\,j}(x) = \alpha_i \Psi^{\,j}\left(a_i(x - t_i)\right), \\[2mm] \alpha_i = \left(\sum_{k=1}^{N} \left[\Psi^{\,j}\left(a_i\left(x_k - t_i\right)\right)\right]^2 \right)^{-\frac{1}{2}} \\[2mm] i = 1,..., L \; and \; j = 1,..., n \end{array} \right\} \qquad (3)$$

where x_k is the sampled input and L is the number of wavelets in each sub library W_j. Then select the best *M* wavelets based on the training data from multi wavelet library *W*, in order to build the regression.

$$\hat{y}(x) = \sum_{i \in I} w_i \, \Psi_i^1(x) + \sum_{i \in I} w_i \, \Psi_i^2(x) + ... + \sum_{i \in I} w_i \, \Psi_i^n(x) \qquad (4)$$

2.3 AN INITIALIZATION PROCEDURE USINGA SELECTION METHOD

It is very inadvisable to initialize the dilations and translations randomly, as is usually the case for the weights of a standard neural network with sigmoid activation function. In the case of wavelet neural network and due to the fact that wavelets are rapidly vanishing functions, a wavelet may be too local if its dilation parameter is too small (it may sit out of the domain of interest), if the translation parameter is not chosen appropriately.

We propose to make use of multi library wavelet using a selection method to initialize the translation and dilation parameters of wavelet networks trained using gradient-based techniques. The procedure comprises four steps:

2.3.1 INITIALIZATION

Let Y the signal to be approximated, we have the same library that previously. This library contains N_{Mw} wavelet. We associate to every wavelet a vector whose components are the values of this wavelet according to the examples of the training sequence. We constitute a matrix V_w thus constitutes blocks of the vectors representing the wavelet of every mother wavelet:

$$V_w = \begin{vmatrix} V_1^1(x_1) & ... & V_N^1(x_1) & ... & V_1^j(x_1) & ... & V_N^j(x_1) & ... & V_1^M(x_1) & ... & V_N^M(x_1) \\ V_1^1(x_2) & \cdots & V_N^1(x_2) & ... & V_1^j(x_2) & \cdots & V_N^j(x_2) & ... & V_1^M(x_2) & ... & V_N^M(x_2) \\ \vdots & ... & \vdots & ... & \vdots & ... & \vdots & ... & \vdots & ... & \vdots \\ \vdots & ... & \vdots & ... & \vdots & ... & \vdots & ... & \vdots & ... & \vdots \\ \vdots & ... & \vdots & ... & \vdots & ... & \vdots & ... & \vdots & ... & \vdots \\ V_1^1(x_N) & ... & V_N^1(x_{N_i}) & \cdots & V_1^j(x_N) & ... & V_N^j(x_{N_i}) & ... & V_1^M(x_{N_i}) & \cdots & V_N^M(x_{N_i}) \end{vmatrix} \qquad (5)$$

$$V_w(t,d) = \{V_i^j\} \, i=[1..N], \, j=[1..M] \qquad (6)$$

We note by :
g(X) the constructed network,
$N_w=1$, the wavelet number
$T=\{t_i\} \, i=[1..N]$, the translation vector
$D=\{d_i\} \, i=[1..N]$, the dilation vector

2.3.2 SELECTION

The library being constructed, a method of selection is applied in order to determine the most meaningful wavelet for approximation the considered signal. In general the wavelets in W are not all meaningful to estimate the signal. Let's suppose that we want to construct a network *g(x)* of wavelets with m wavelets, the problem is to select m wavelets of W.

To the first iteration, the signal is $y = Y_1$, and the vector regressor is the $V_w(t,ds)$ definite by (6), the selected regressor is the one for which the absolute value of the cosine with the *Y1* signal is maximal. We define ipert1 as:

$$ipert1(i,j) = \arg \max_{i,j} \frac{\langle Y | V_i^{\,j} \rangle}{\|Y\|.\|V_i^{\,j}\|}, \qquad (7)$$
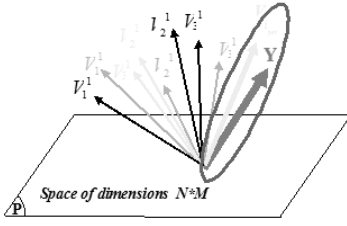
with i=[1..N], j=[1..M]

Fig. 1 – Selection of the pertinent vector.

The regressor $V_{i_{pert}}1$ can be considered like an adjustable and temporal function used to approximate Y (see figure 1). We calculate the weight w_1 defined by:

$$w_1 = \frac{Y}{V_{i_{pert}}1} \tag{8}$$

We calculate thereafter, the mean square error of training (MSET) definite by:

$$\text{MSET}(\omega_{i\,pert}, t_{i\,pert}, d_{i\,pert}) = \frac{1}{N}\sum_{k=1}^{N}(Y(k) - \omega_{i\,pert} * V_{i_{pert}})^2 \tag{9}$$

Y(k) is the desired output correspondent to the example *k*, and $\omega_{i\,pert} * V_{i_{pert}}$ is the wavelet network output to the example *k*.

2.3.3 REGRESSOR OPTIMIZATION

To optimize the regressor we used the pressure gradient method:

We notes by:

$e(x) = Y_d(x) - Y(x)$, with Yd: desired output and *Y*: the real network output.

$$\frac{\partial MSET}{\partial w_{ipert}1} = \sum_{i=1}^{Nw} e(x)\Psi\left(\frac{x - t_{ipert}1}{d_{ipert}1}\right) \tag{10}$$

$$\frac{\partial MSET}{\partial d_{iper}1} = \sum_{i=1}^{Nw} e(x)w_{ipert}1\frac{\partial \Psi\left(\frac{x - t_{ipert}1}{d_{ipert}1}\right)}{\partial d_{ipert}1} \tag{11}$$

$$\frac{\partial MSET}{\partial t_{iper}1} = \sum_{i=1}^{Nw} e(x)w_{ipert}1\frac{\partial \Psi\left(\frac{x - t_{ipert}1}{d_{ipert}1}\right)}{\partial t_{ipert}1} \tag{12}$$

This optimization has the advantage to be fast because we only optimize the three structural parameters of the network.

To the exit of this optimization, the parameters: $t_{i_{pert}}^{opt}1, d_{i_{pert}}^{opt}1,\ \omega_{i_{pert}}^{opt}1$ of the regressor $V_{i_{pert}}1$ are adjusted, and are solutions of the optimization problem defined by:

$$V_{i_{pert}}^{opt} = V_{i_{pert}}(\omega_{i_{pert}}^{opt}, t_{i_{pert}}^{opt}, d_{i_{pert}}^{opt}) \tag{13}$$

Considering the optimal regressor, we reset the network with this regressor that is going to replace the old in the library. The orthogonalization will be done in relation to the optimal regressor as shown in figure 2.

$$V_W = \begin{vmatrix} V_1^1(x_1) & \dots & V_N^1(x_1) & \dots V_{i_{pert}}^{opt}(x_1) & \cdots & V_1^M(x_1) & \dots & V_N^M(x_1) \\ V_1^1(x_2) & \cdots & V_N^1(x_2) & \cdots V_{i_{pert}}^{opt}(x_2) & \cdots & V_1^M(x_2) & \cdots & V_N^M(x_2) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ V_1^1(x_N) & & V_N^1(x_{N_i}) & \cdots V_{i_{pert}}^{opt}(x_{N_i}) & \dots & V_1^M(x_{N_i}) & \cdots & V_N^M(x_{N_i}) \end{vmatrix} \tag{14}$$

After an iteration we will have:

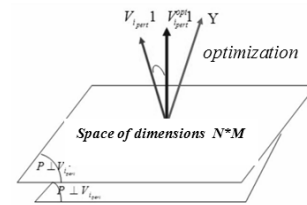$$V_{w1}(t,d) = \{V_i^j\}_{i=[1..N], j=[1..M]} \cup \{V_{i_{pert}}^{opt}\} \tag{15}$$

Fig. 2 – Regressor Optimization.

2.3.4 ORTHOGONALIZATION

The vectors V_i^j are always linearly independent, (because $N >> M_W$) and non orthogonal. The vectors V_i^j generate one sub-vector-space of *M*N* dimension. We orthogonalize the $N*M_S-1$ remaining regressors, and the vector *Y₁* in relation to the adjusted regressor $V_{i_{pert}}^{opt}1$:

$$V_i^{j\perp} = V_i^j - \left\langle V_i^j \middle| V_{i_{pert}}^{opt}1\right\rangle V_{i_{pert}}^{opt}1 \tag{16}$$

$$Y^\perp 1 = Y - \left\langle Y \middle| V_{i_{pert}}^{opt}1\right\rangle V_{i_{pert}}^{opt}1 \tag{17}$$

We make then, the updating of the library:

$$V_W = \begin{vmatrix} V_1^{1\perp}(x_1) & \dots & V_N^{1\perp}(x_1) & \dots V_1^{J\perp}(x_1) & \dots & V_N^{J\perp}(x_1) & \dots & V_1^{M\perp}(x_1) & \dots & V_N^{M\perp}(x_1) \\ V_1^{1\perp}(x_2) & \cdots & V_N^{1\perp}(x_2) & \cdots V_1^{J\perp}(x_2) & \cdots & V_N^{J\perp}(x_2) & \cdots & V_1^{M\perp}(x_2) & \cdots & V_N^{M\perp}(x_2) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ V_1^{1\perp}(x_N) & & V_N^{1\perp}(x_N) & \cdots V_1^{J\perp}(x_N) & & V_N^{J\perp}(x_{N_i}) & \cdots & V_1^{M\perp}(x_N) & \cdots & V_N^{M\perp}(x_{N_i}) \end{vmatrix} \tag{22}$$

We obtain:

$$V_w(t,d) = \{V_i^{j\perp}\}_{(i=[1..N], j=[1..M]) \setminus \{i_{pert}\}} \tag{18}$$

$Y^\perp 1$ and $\{V_i^{j\perp}\}$ are respectively what remains the signal and regressors in the space orthogonal to $V_{i_{pert}}^{opt} 1$.

The model being to this stage, $g(X) = \omega_i 1 * V_{i_{pert}}^{opt} 1$, is shown in figure 3.
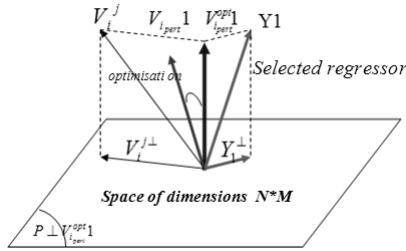

Fig. 3 – Orthogonal projection on the optimal regressor.

To the following iteration we increment the number of wavelet $N_w = N_w + 1$. We applied the same stages decry above. Let's suppose achieved *i-1* iterations: we did *i - 1* selections, optimizations, and orthogonalizations. To get the *i-1* adjusted regressors we reset i-1 parameter of the network.

At the end of the iteration *i -1*, the expression of the network *g(X)*, is given by:

$$g(X) = \sum_{i=1}^{i-1} \omega_i^{opt} * V_{i_{pert}}^{opt} \tag{19}$$

We have Nw-i+1 régressors to represent the signal Yi in a space of N*M - i +1 dimension orthogonal to ($V_{i_{pert}}^{opt} 1$,..., $V_{i_{pert}}^{opt} i - 1$)

We apply the same procedure of selection as previously, the indication iperti of the selected regressor is the one for which the absolute value of the cosine with the signal Yi is maximal; iperti is given by (7)

Finally and after N iteration, we construct a wavelet network with N wavelet in the hidden layer that approximates the input signal Y.

So the parameters of the network are:

$$T^{opt} = \left\{t_{i_{pert}}^{opt}\right\}_{i_{pert}=[1..Nopt]} \tag{20}$$

$$d^{opt} = \left\{d_{i_{pert}}^{opt}\right\}_{i_{pert}=[1..Nopt]} \tag{21}$$

$$\omega^{opt} = \left\{\omega_{i_{pert}}^{opt}\right\}_{i_{pert}=[1..Nopt]} \tag{22}$$

The obtained model *g(X)* can be written under the shape:

$$g(X) = \sum_{i=1}^{N_{opt}} \omega_i^{opt} * V_i^{opt} \tag{23}$$

3. BETA MULTI LIBRARY WAVELET NETWORK

3.1 BETA WAVELET FAMILY

The Beta function [14] is defined as:
if p>0, q>0, (p, q) $\in$ IN

$$\beta(x) = \begin{cases} (\frac{x-x_0}{x_c-x_0})^p (\frac{x_1-x}{x_1-x_c})^q \; if \; x \in [x_0, x_1] \\ 0 \quad else \end{cases} \tag{24}$$

$$\text{Where, } x_c = \frac{px_1 + qx_0}{p+q}$$

We prove in [15] that all the derivatives of Beta function $\in L^2(\mathfrak{R})$, are of class C^∞ and satisfy the admissibility wavelet condition for *p=q*.

3.2 EXAMPLE 1:1-D FUNCTION APPROXIMATION

The first example is the approximation of a function of a single variable function, without noise, given by:

$$f(x) = \begin{cases} -2.186x - 12.864 \; for \; x \in [-10, -2[\\ 4.246x \; for \; x \in [-2, -0[\\ 10\exp(-0.05x - 0.5)\sin(x(0.03x + 0.7)) \; for \; x \in [0,10[\end{cases} \tag{25}$$

First, simulations on the 1-D function approximation are conducted to validate and compare the proposed MLWNN with the classical WNN. The input x is constructed by the uniform distribution on [-10 10]. The training sequence is composed of 101 points. The performance of the model is estimated using a test set of 101 equally spaced examples different from the training set.

We define the *NMSE* (Normalized Mean Square Error) as evaluation criteria.

$$NMSE = \frac{1}{N} \sum_{k=1}^{N} \left(\hat{f}(x_k) - y_k\right)^2 \tag{26}$$

In the following, we present the results obtained with a network of 12 Beta wavelets, chosen as mother wavelets (second and third derivative of Beta function), for training network. Figure 4 shows the initial error histogram (a) obtained when the 101 input patterns are initialized with the classical architecture and the final error histogram (b) obtained when the 101 input patterns are training after 1000 iterations. Figure 4 (c) shows the initial error histogram obtained when the 101 trainings are

initialized with the initialization by selection procedure using MLWNN and the final error histogram (d) obtained when the 101 input patterns are training after 1000 iterations. We can see clearly that the initialization by selection using MLWNN leads to:

- The best result in term of NMSE,

- Less scattered results both on the training set and on the test set.

- Using multi wavelet mothers as activation function gives best approximation.
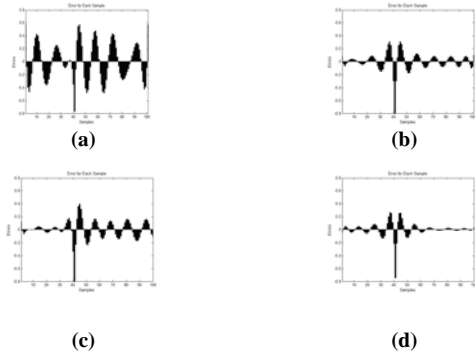

(a)   **(b)**

(c)   **(d)**

Fig. 4 – Evolution of the initial and final error for each sample after initialization using classical WNN architecture and MLWNN architecture.

Figure 5 shows the evolution of the NMSE according to the iteration; (a) shows the initial error for each sample after initialization using classical WNN architecture, (b) gives the final error for each sample after initialization using classical WNN architecture, when figure (c) and (d) show respectively the initial and final error for each sample after initialization using MLWNN architecture. We can see the superiority of the proposed initialization selection algorithm based on multi wavelet library over the classical WNN based on one mother wavelet.
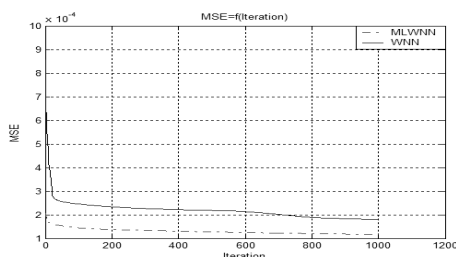

Fig. 5 – Evolution of the NMSE according to the iteration.

3.3 EXAMPLE 2: 2-D FUNCTION APPROXIMATION

The process to be modeled is simulated by a function of two variables without noise. The expression of this function is given by:

$$f(x_1, x_2) = \left(\frac{x_1 + 2}{2}\right)^5 \left(\frac{x_2 + 2}{2}\right)^5 \left(\frac{2 - x_1}{2}\right)^5 \left(\frac{2 - x_2}{2}\right)^5 \tag{27}$$

In the following, we present the results obtained with a network of 9 Beta wavelets, chosen as mother wavelets (second and third derivative of Beta function), for training network. The training set contains 11x11 uniform spaced points. The test set V is constructed by 21x21 stochastic points on [-1,1]x[-1,1]. Figure 6 shows the final error histogram (a) obtained when the 121 trainings are initialized with the classical architecture initialization and the final error histogram (b) obtained when the 121 trainings are initialized with a selection procedure using MLWNN.

(a)   **(b)**

Fig. 6 – Final error for each sample after initialization using classical WNN architecture (a) and MLWNN architecture (b).

These results show that the effect of the classical WNN initialization is much smaller than when the wavelet centers and dilations are initialized by selection using a multi library WNN, used together with Beta wavelets, it makes wavelet neural network training very efficient because of the adjustable parameters of Beta function.

4. BETA WAVELET NEURAL NETWORK FOR LOSSLESS COLOR IMAGE COMPRESSION

The idea consists to transform an RGB image into luminance-chrominance space. Compression is achieved by determining the value of N_p. The input (Luminance) is split up into blocks or vectors of 8x8, 16x16 or 32x32 pixels. When the input vector is referred to as 1-dimensional which is equal to the number of pixels included in each block, each wavelet at the hidden layer can be represented by three parameters: weight, translation and dilation, which can also be described by three matrix of order 3x N_p. Image compression is achieved by training the network in such a way the network output scale the input and produces the optimum output value which makes the quadratic error between input and output minimum.

In accordance with the neural network structure shown in Fig.1 the encoding phase can be described as follows:

$$e(x) = y(x) - \hat{y}(x) \tag{28}$$

The energy function is define as

$$E = \frac{1}{2}\sum_{x=1}^{X} e^2(x) \tag{29}$$

The mean square error is optimised according to these equations

$$\frac{\partial E}{\partial w_k} = -\sum_{x=1}^{X} e(x)\Psi(\tau) \tag{30}$$

$$\frac{\partial E}{\partial x_k} = -\sum_{x=1}^{X} e(x)w_k \frac{\partial\Psi(\tau)}{\partial x_k} \tag{31}$$

$$\frac{\partial E}{\partial d_k} = -\sum_{x=1}^{X} e(x)w_k\tau \frac{\partial\Psi(\tau)}{\partial x_k} = \tau\frac{\partial E}{\partial x_k} \tag{32}$$

$$with\,\tau = \frac{x - x_k}{d_k} \tag{33}$$

$$\Delta w = -\frac{\partial E}{\partial w}, \Delta t = -\frac{\partial E}{\partial t}, \Delta d = -\frac{\partial E}{\partial d} \tag{34}$$

$$\underline{w}(n+1) = \underline{w}(n) + \mu_w\Delta\underline{w} \tag{35}$$

$$\underline{t}(n+1) = \underline{t}(n) + \mu_t\Delta\underline{t} \tag{36}$$

$$\underline{d}(n+1) = \underline{d}(n) + \mu_d\Delta\underline{d} \tag{37}$$

μ is the training constant.

where $x \in [0, 1]$ denotes the normalized pixel values for RGB images with intensity levels $[0, 255]$. The reason for using normalized pixel values is due to the fact that wavelet networks can operate more efficiently when both their inputs and outputs are limited to a range of $[0, 1]$ [5].

In the first phase, a set of image samples is designed to train the network via the descent gradient learning rule which uses each input vector as the desired output. This is equivalent to compressing the input into the narrow channel represented by the hidden layer and then reconstructing the input from the hidden to the output layer.

The second phase simply involves the entropy coding of the state vector Ψ at the hidden layer. Since the hidden wavelet output is real valued, quantization is required for fixed length entropy coding which is normally designed as 32 level uniform quantization corresponding to 5 bits entropy coding [9,14].

4.1 PERFORMANCE ASSESSMENTS

Around the Wavelet Neural Network, we have described the scheme to achieve image data compression can normally be assessed by considering two measurements. One is the compression ratio or bit rate which is used to measure the compression performance, and the other

is mainly used to measure the quality of reconstructed images with regards to a specific compression ratio or bit rate. The definition of this measurement, however, is a little ambiguous at present [16]. In practice, there exists two acceptable measurements for the quality of reconstructed images which are *PSNR* (peak-signal-to noise ratio) and *NMSE*. Their definitions can be given, respectively, as follows:

$$PSNR = 10\log\frac{255^2}{\frac{1}{nN}\sum_{i=1}^{n}\sum_{j=1}^{N}\left(\overline{P_{ij}} - P_{ij}\right)^2}(dB) \tag{38}$$

$$NMSE = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N}\left(\overline{P_{ij}} - P_{ij}\right)^2}{\sum_{i=1}^{n}\sum_{j=1}^{N} P_{ij}^2} \tag{39}$$

Where $\overline{P_{ij}}$ is the intensity value of pixels in the reconstructed images; and P_{ij} the intensity value of pixels in the original images which are split up into n input vectors: $x_i = \{P_{i1}, P_{i2},..., P_{iN}\}$.

4.2 PERFORMANCES ACCORDING TO THE PSNR, AND NMSE FOR COLOR IMAGE COMPRESSION USING MLWNN

We use 3 RGB test images: Nature1, Nature2 and Lena. For each image we compute The PSNR, SNR and NMSE using MLWNN composed of 4 and 10 neurons in hidden layer.
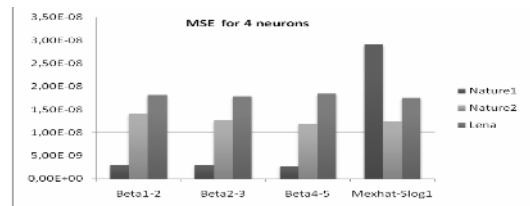

Fig. 7 – Variation of NMSE in term of wavelets library for Nature 1, Nature 2 and Lena, using 4 neurons and MLWNN architecture.
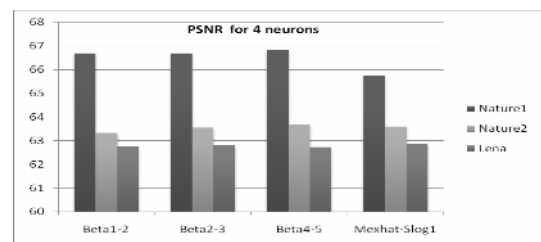

Fig. 8 – Variation of PSNR in term of wavelets library for Nature 1, Nature 2 and Lena, using 4 neurons and MLWNN architecture.
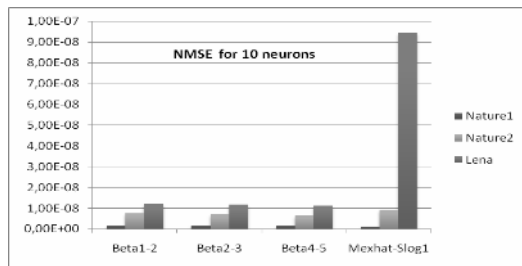
Fig. 9 – Variation of NMSE in term of wavelets library for Nature 1, Nature 2 and Lena, using 10 neurons and MLWNN architecture.
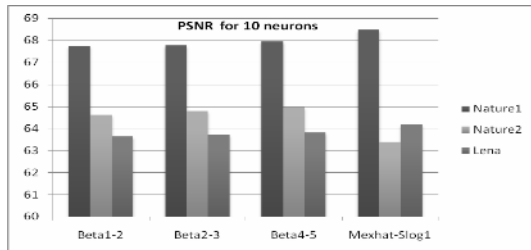

Fig. 10 – Variation of PSNR in term of wavelets library for Nature 1, Nature 2 and Lena, using 10 neurons and MLWNN architecture.

From these results (figure 7, 8, 9 and 10) we see that the NMSE depends on the wavelet library and the number of wavelets in the hidden layer. For example for nature1 we have an NMSE equal to 2.9929e-9 for a library constructed by Beta1 and Beta2 and for a net of 4 neurons in hidden layer where it is equal to 1.8244e-9 when we used a net of 10 neurons. Of course the time of compression is an essential factor: increasing the number of wavelets in hidden layer increases the time processing. For a multi library wavelet network based on selection procedure for initialization we can see that the difference of *NMSE* between 4 neurons and 10 neurons is small. To minimize time processing without biggest variation in *NMSE* we can reduce the number of neurons in hidden layer.

4.3 PERFORMANCES ACCORDING TO THE PSNR, AND NMSE FOR COLOR IMAGE COMPRESSION USING CLASSICAL WNN
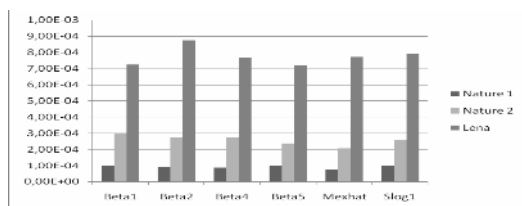

Fig. 11 – Variation of NMSE in term of wavelets library for Nature 1, Nature 2 and Lena, using 10 neurons and classical architecture.
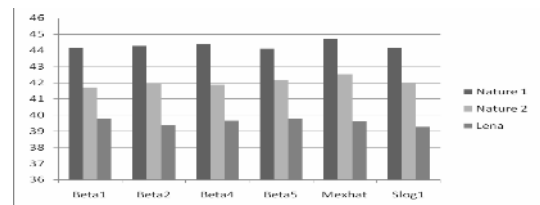

Fig. 12 – Variation of PSNR in term of wavelets library for Nature 1, Nature 2 and Lena, using 10 neurons and MLWNN architecture.

Comparing results given for the classical and proposed architecture (figure 11 and figure 12), we can conclude that the proposed architecture wavelet network, for the same value of compression ratio, gives results more interesting in term of mean square error. We can confirm that using more than one regressor (more than one wavelet in the library) in the wavelet library fit better the real output to the desired output.

5. CONCLUSION

Wavelet networks are a class of neural networks consisting of wavelets. In this paper, we have proposed a new Initialization by Selection algorithm for Multi library Wavelet Neural Network Training for the purpose of lossless color image compressing that provides improved efficiency compared to the classical wavelet neural networks.

From these results we can see the superiority of Beta wavelets family over the Mexican hat in the term of lossless color image compression.

We have shown that, when used a multi library wavelet networks and a selection procedure leads to results that are much more interesting than the classical architecture initialization.

Finally, although wavelet neural network approximation can be slow, we have shown that the loss in speed can be largely corrected by decreasing the number of wavelets in hidden layer without decreasing considerably the compression ratio and the PSNR.

As future research directions, we propose to use MLWNN in the case of adaptive self tuning PID controllers. The MLWNN is needed to learn the characteristics of the plant dynamic systems and make use of it to determine the future inputs that will minimize error performance index so as to compensate the PID controller parameters.

6. REFERENCES

[1] S. Mallat, A wavelet tour of signal processing. academic press 1998.

[2] Q. Zang, Wavelet Network in Nonparametric Estimation. IEEE Trans. Neural Networks, 1997. 8(2):227-236.

[3] Q. Zang et al., Wavelet networks. IEEE Trans.

Neural Networks, vol. 3, 1992. p. 889-898.

[4] H. Bourlard, Y. Kamp, Autoassociation by multilayer perceptrons and singular values decomposition, Biol. Cybernet. 1988. 291-294.

[5] A. Averbuch, D. Lazar, Image compression using wavelet transform and multiresolution decomposition, IEEE Trans. Image Process. 1996. p. 4-15.

[6] Hamdy S. Soliman, Mohammed Omari, A neural networks approach to image data compression, Applied Soft Computing, 2006. p. 258–271.

[7] G. Candotti, S. Carrato et al., Pyramidal multiresolution source coding for progressive sequences, IEEE Trans. Consumer Electronics, 1994. p. 789-795.

[8] S. Carrato, Neural networks for image compression, Neural Networks: Adv. and Appl. 2 ed., Gelenbe Pub, North-Holland, Amsterdam, 1992. p. 177-198.

[9] O.T.C. Chen et al., Image compression using self-organisation networks, IEEE Trans. Circuits Systems For Video Technol. 1994. p. 480-489.

[10] Slaven Marusic, Guang Deng, Adaptive prediction for lossless image compression, Signal Processing: Image Communication. 2002. p. 363–372.

[11] N.A. Laskaris, S. Fotopoulos, A novel training scheme for neural-network based vector quantizers and its application in image compression, Neurocomputing 2004. p.421-427.

[12] C. Foucher and G. Vaucher, Compression d'images et réseaux de neurones, revue Valgo n°01-02, Ardèche, 2001.

[13] Q. Zhang, Using Wavelet Network in Nonparametric Estimation, IEEE Trans. Neural Network, Vol. 8, 1997. p. 227-236.

[14] C. Aouiti, M.A Alimi, and A. Maalej, Genetic Designed Beta Basis Function Neural Networks for Multivariable Functions Approximation, Systems Analysis, Modeling, and Simulation, Special Issue on Advances in Control and Computer Engineering, vol. 42, no. 7, 2002. p. 975-1005.

[15] C. Ben Amar, W. Bellil and A. Alimi. Beta Function and its Derivatives: A New Wavelet Family. Transactions on Systems, Signals & Devices Volume 1, Number 3, 2005-2006. p. 275-293.

Wajdi BELLIL *was born in Gafsa, Tunisia, in 1977. He received his Master in Automatic and industrial Informatics from the National school of engineering, University of Sfax, Tunisia, in 2003. He is currently pursuing his doctoral degree in the Department of Electrical Engineering. His research interests are in the areas of wavelet neural network architecture sand learning algorithms, and applications of neural networks and machine learning techniques in image processing and pattern recognition.*

Chokri Ben Amar *was born in Sfax-Tunisia in 1964. he has been the studies director and the vice director of the National Engineering School of Sfax (ENIS) since June 2005. He has been a lecturer in Industrial Informatics at the department of Electrical Engineering in the National Engineering School of Sfax (ENIS) since 1999. He obtained the Master degree and the Doctorat at the Institut National des Sciences Appliquées de Lyon-France (INSA) in Industrial Automatics in 1990 and 1994. Since 1995, he has been an associate professor at the Ecole Nationale d'Ing´enieurs de Sfax. His research interests include multi-resolution image and video analysis, wavelets and neural Networks,with applications to function approximation and face recognition.*

Adel M. Alimi *was born in Sfax-Tunisia in 1966. He graduated in Electrical Engineering 1990, obtained a PhD and then an HDR both in Electrical Engineering in 1995 and 2000 respectively. Now, He is professor in Electrical & Computer Engineering at the University of Sfax. His research interest includes applications of intelligent methods (neural networks, fuzzy logic, genetic algorithms) to pattern recognition, robotic systems, vision systems, and industrial processes. He focuses his research on intelligent pattern recognition, learning, analysis and intelligent control of large scale complex systems. He is associate editor of the international journal: Pattern Recognition Letters. He was guest editor of several special issues of international journals (e.g. Fuzzy Sets & Systems, Soft Computing, Journal of Decision Systems, Integrated Computer Aided Engineering, Systems Analysis Modelling & Simulations). He was the general chairman of the International Conference on Machine Intelligence ACIDCA-ICMI'2005 & 2000. He is an IEEE senior member and member of IAPR, INNS and PRS.*