

A New Algorithm for Initialization and Training of Beta Multi-Library Wavelets Neural Network

Wajdi Bellil, Mohamed Othmani, Chokri Ben Amar
and Mohamed Adel Alimi
University of Sfax (National school of engineers of Sfax)
Tunisia

1. Introduction

The resolutions of neurons networks training problems by gradient are characterized by their noticed inability to escape of local optima [Mich93], [Fabr94] and in a least measure by their slowness [Wess92], [Zhan92]. The evolutionist algorithms bring in some domains a big number of solutions: practice of networks to variable architecture [With90], automatic generation of Booleans neurons networks for the resolution of a class of optimization problems [Grua93]. However the effort of research was especially carried on the generation and the discreet network training.

In this chapter, we propose a new algorithm of wavelets networks training, based on gradient that requires:

- A set of training examples: the wavelets networks are parametrables functions, used to achieve statistical models from examples (in the case of classification) or of measures (in the case of modeling); their parameters are calculated from these examples or couples {input, output}.
- The definition of a cost function that measures the gap between the input of the wavelets network and the desired output (in the case of classification) or the measured values (in case of modeling) present on the set of training.
- A minimization algorithm of the cost function.
- An algorithm of selection of basic function to initialize the network parameters.

We try then to show the importance of initialization of the network parameters. Since the output is non linear in relation to these parameters, the cost function can present local minima, and the training algorithms don't give any guarantee to find the global minimum.

We note that if we have a good initialization, the local minimum problem can be avoided, it is sufficient to select the best regressions (the best based on the training data) from a finished set of regressors. If the number of regressors is insufficient, not only some local minima appear, but also, the global minimum of the cost function doesn't necessarily correspond to the values of the searched parameters, it is useless then in this case to put an expensive algorithm to look for the global minimum.

With a good initialization of the network parameters the efficiency of training increases. A very important factor that it is necessary to underline is: whatever the chosen algorithm, the quality of training wavelets networks is as much better than we have an optimal initialization.

Source: Advances in Robotics, Automation and Control, Book edited by: Jesús Arámburo and Antonio Ramírez Treviño, ISBN 78-953-7619-16-9, pp. 472, October 2008, I-Tech, Vienna, Austria

2. New wavelets networks architecture

2.1 Presentation

From a given wavelets network architecture, it is possible to generate a family of parametrables functions by the values of the network coefficients (weight, translations, dilations).

The objective of the wavelets networks training phase is to find, among all these functions, the one that approaches the most possible regression (Beta function for example). This one is unknown (otherwise it would not be necessary to use an approximation by wavelets networks); we only know the observed values (values of the regression to which are added noise) for several values valued by the input (points of the training set).

We consider wavelets networks as follows [Belli07]:

$$\hat{y} = \sum_{i=1}^{N_i} \omega_i^1 \psi_i^1(x) + \sum_{i=1}^{N_i} \omega_i^2 \psi_i^2(x) + \dots + \sum_{i=1}^{N_M} \omega_i^M \psi_i^M(x) + \sum_{k=0}^{N_i} a_k x_k \quad (1)$$

$$= \sum_{j=1}^M \sum_{i=1}^{N_{Mw}} \omega_i^j \psi_i^j(x) + \sum_{k=0}^{N_i} a_k x_k \quad (2)$$

$$= \sum_{l(i,j)=1}^{N_{Mw}} \omega_l \psi_l(x) + \sum_{k=0}^{N_i} a_k x_k \quad (3)$$

$$\text{with } N_{Mw} = \sum_{l=1}^M N_l, \quad i = [1, \dots, N], j = [1, \dots, M], x_0 = 1 \quad (4)$$

Where \hat{y} is the network output and $x = \{x_1, x_2, \dots, x_{N_i}\}$ the input vector; it is often useful to consider, in addition to the wavelets decomposition, that the output can have a linear component in relation to the variables: the coefficients a_k ($k = 0, 1, \dots, N_i$).

N_i is the number of selected wavelets for the mother wavelet family Ψ_i .

The index l depends on the wavelet family and the choice of the mother wavelet.

The network can be considered as constituted of three layers:

- A first layer with N_i input.
- A hidden layer constituted by N_{Mw} wavelets of M mothers wavelets each to a wavelet family of size N_i .
- A linear output neuron receiving the pondered wavelets outputs and the linear part.

This network is illustrated by the figure 1.

2.2 Description of the procedure of library construction

The first stage of the training procedure consists in the construction of the Beta library. We intend to construct a several mother wavelets families library for the network construction. Every wavelet has different dilations following different inputs.

This choice presents the advantage to enrich the library, and to get a better performance for a given wavelets number. The inconvenience introduces by this choice concerns the size of the library. A wavelet library having several wavelets families is more voluminous than the

one that possesses the same wavelet mother. It implies a more elevated calculation cost during the stage of selection.

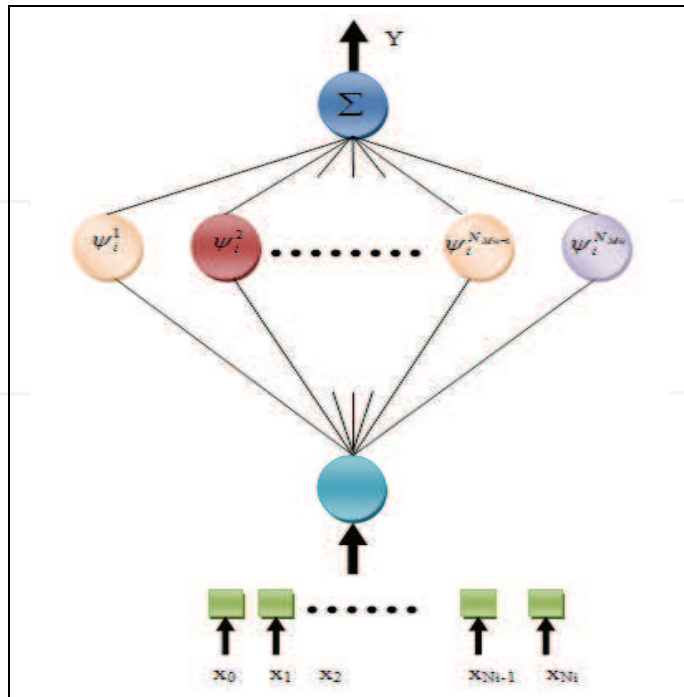


Fig. 1. Graphic representation of the new wavelets network architecture

Nevertheless, using classic algorithms optimization, the selection of wavelets is often shorter than the training of the dilations and translations; the supplementary cost introduced by different dilations can be therefore acceptable.

We have a sequence of training formed of N examples distributed in the interval [a, b]. Let Ψ^j a mother wavelet family, x the variable, t_i the translation parameter and d_i the dilation parameter. The wavelet Ψ_i^j of the family Ψ^j having for parameters t_i and d_i is defined as:

$$\Psi_i^j(x) = \Psi^j(d_i(x - t_i)) \tag{5}$$

The wavelet library W, generated from the mother wavelet family, is defined as:

$$W = \{ \Psi^j(d_i(x - t_i)), d_i \in R_+^*, t_i \in R, j = [1, \dots, M] \} \tag{6}$$

$$= \{ W^1, \dots, W^N \} \tag{7}$$

$$= \{ \psi^1(d_i(x - t_i)), \dots, \psi^M(d_i(x - t_i)) \} \tag{8}$$

3. New algorithm: MLWNN (Multi-Library Wavelet Neural Network)

In this paragraph, we propose a new selection wavelets algorithm [Bellil07-a], based on the architecture given on the figure 1 that permits to make:

- The initialization of weights, translations and dilations of wavelets networks,
- The optimization of the network parameters,
- The construction of the optimal library,
- The construction of the wavelets networks based on discrete transform.

The new architecture of wavelets networks founded on several mother wavelets families having been defined. Consequently, we can ask the question of construction of a model, constituted of wavelets network for a given process.

The parameters to determine for the construction of the network are:

- The values to give to the different parameters of the network: structural parameters of wavelets, and direct terms.
- The necessary number of wavelets to reach a wanted performance.

The essential difficulty resides in the determination of the parameters of the network. Because the parameters take discrete values we can make profit to conceive methods of wavelets selection in a set (library) of discrete wavelets. The conceived performance depends on the initial choice of the wavelets library, as well as a discriminating selection in this library.

3.1 Principle of the algorithm

The idea is to initialize the network parameters (translations, dilations and weights) with values near to the optimal values. Such a task can be achieved by the algorithm "Orthogonal Forward Regression (OFR)" based on the algorithm of orthogonalization of Gram - Schmidt [Chen89], [Ouss98], [Chen06], [Ho01].

Contrary to the OFR algorithm in which the best regressors are first selected [Lin03], [Rao04], [Xiao04], [Angr01], then adjusted to the network, the algorithm presented here integrates in every stage the selection and the adjustment. Before every orthogonalization with a selected regressor, we apply a summary optimization of the parameters of this one in order to bring it closer to the signal.

Once optimized, this new regressor replaces the old in the library and the orthogonalization will be done using the new regressor.

We describe this principle below in detail.

3.2 Description of the algorithm

The proposed algorithm depends on three stages:

3.2.1 Initialization

Let's note by Y the input signal; we have a library that contains N_{Mw} wavelets. To every wavelet Ψ_i' we associate a vector whose components are the values of this wavelet according to the examples of the training sequence. We constitute a matrix thus constituted V_w of the blocks of the vectors representing the wavelets of every mother wavelet where the expression is:

$$V_w = \begin{pmatrix} V_1^1(x_1) & \dots & V_N^1(x_1) & \dots & V_1^J(x_1) & \dots & V_N^J(x_1) & \dots & V_1^M(x_1) & \dots & V_N^M(x_1) \\ V_1^1(x_2) & \dots & V_N^1(x_2) & \dots & V_1^J(x_2) & \dots & V_N^J(x_2) & \dots & V_1^M(x_2) & \dots & V_N^M(x_2) \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ V_1^1(x_N) & \dots & V_N^1(x_N) & \dots & V_1^J(x_N) & \dots & V_N^J(x_N) & \dots & V_1^M(x_N) & \dots & V_N^M(x_N) \end{pmatrix} \quad (9)$$

$$V_w(t, d) = \{V_i^j\} i=[1..N], j=[1..M] \quad (10)$$

We note by:

- $g(x)$ the constructed network,
- $N_w=1$ the number of wavelet,
- $T=\{t_i\} i=[1..N]$ the translations,
- $D=\{d_i\} i=[1..N]$ the dilations.

3.2.2 Selection

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

To the first iteration, the signal is $Y = Y_1$, and the regressors vectors are the $V_w(t,d)$ defined by (10). The selected regressor is the one for which the absolute value of the cosine with the signal Y_1 is maximal. The most pertinent vector from the family V_1 carries the index i_{pert1} that can be written as the following manner:

$$i_{pert1}(i,j) = \arg \max_{i,j} \frac{\langle Y | V_i^j \rangle}{\|Y\| \cdot \|V_i^j\|} \text{ with } i=[1..N], j=[1..M] \quad (11)$$

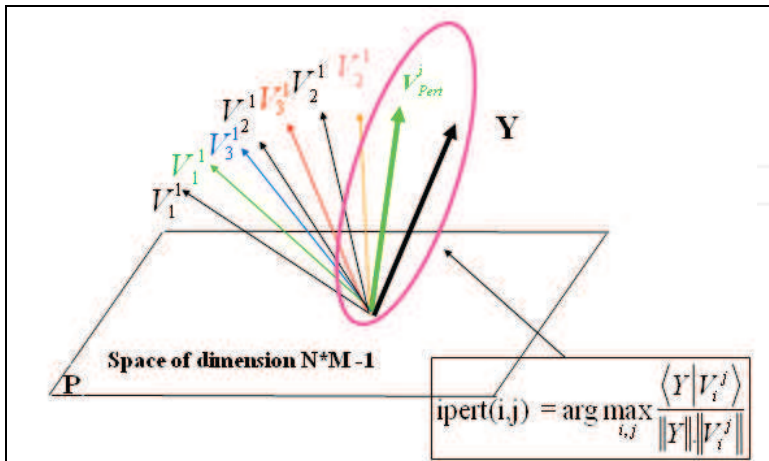


Fig. 2. Selection of the pertinent vector

Once the regressor $V_{i_{pert}} 1$ is selected, it can be considered like a parametrable temporal function used for modeling Y . We calculate the weight W_i defined by:

$$w_1 = \frac{Y}{V_{i_{pert}} 1} \quad (12)$$

We define the normalized mean square error of training (NMSET) as:

$$NMSET(\omega_{i_{pert}}, t_{i_{pert}}, d_{i_{pert}}) = \frac{1}{N} \sum_{k=1}^N (Y(k) - \omega_{i_{pert}} * V_{i_{pert}})^2 \quad (13)$$

With $Y(k)$ is the desired output corresponding to the example k , and $\omega_{i_{pert}} * V_{i_{pert}}$ is the wavelets network output corresponding to the example k .

3.2.3 Optimization of the regressor

The optimization of the regressor is made by using the gradient method. Let's note by:

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

$$\frac{\partial NMSET}{\partial w_{i_{pert}} 1} = \sum_{i=1}^{N_w} e(x) \Psi \left(\frac{x - t_{i_{pert}} 1}{d_{i_{pert}} 1} \right) \quad (14)$$

$$\frac{\partial NMSET}{\partial d_{i_{pert}} 1} = \sum_{i=1}^{N_w} e(x) w_{i_{pert}} 1 \frac{\partial \Psi \left(\frac{x - t_{i_{pert}} 1}{d_{i_{pert}} 1} \right)}{\partial d_{i_{pert}} 1} \quad (15)$$

$$\frac{\partial NMSET}{\partial t_{i_{pert}} 1} = \sum_{i=1}^{N_w} e(x) w_{i_{pert}} 1 \frac{\partial \Psi \left(\frac{x - t_{i_{pert}} 1}{d_{i_{pert}} 1} \right)}{\partial t_{i_{pert}} 1} \quad (16)$$

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

After optimization, the parameters: $t_{i_{pert}}^{opt} 1$, $d_{i_{pert}}^{opt} 1$, $w_{i_{pert}}^{opt} 1$ of the regressor $V_{i_{pert}} 1$ are adjusted, and are solutions of the optimization problem defined by:

$$V_{i_{pert}}^{opt} = V_{i_{pert}}(\omega_{i_{pert}}^{opt}, t_{i_{pert}}^{opt}, d_{i_{pert}}^{opt}) \quad (17)$$

Considering the optimal regressor, we reset the network with this regressor that is going to replace the old in the library and the orthogonalization will be done using the new regressor.

$$V_w = \begin{pmatrix} V_1^1(x_1) & \dots & V_N^1(x_1) & \dots & V_{i_{pert}}^{opt}(x_1) & \dots & V_1^M(x_1) & \dots & V_N^M(x_1) \\ V_1^1(x_2) & \dots & V_N^1(x_2) & \dots & V_{i_{pert}}^{opt}(x_2) & \dots & V_1^M(x_2) & \dots & V_N^M(x_2) \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ V_1^1(x_N) & & V_N^1(x_N) & & V_{i_{pert}}^{opt}(x_N) & & V_1^M(x_N) & & V_N^M(x_N) \end{pmatrix} \quad (18)$$

After one iteration we will have:

$$V_{w1}(t,d) = \{V_i^j\}_{i=[1..N],j=[1..M]} \cup \{V_{i_{pert}}^{opt}\} \quad (19)$$

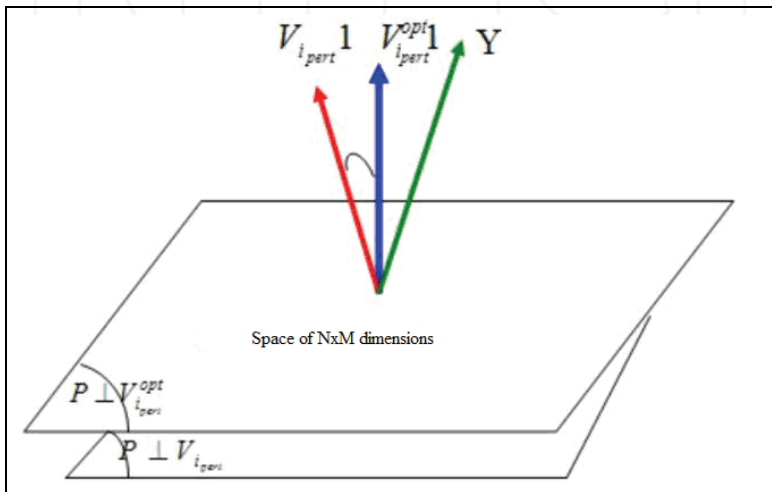


Fig. 3. Regressor optimization

3.2.4 Orthogonalization

The vectors V_i^j are always linearly independent and non orthogonal (because $N \gg M_w$). The vectors V_i^j generate a sub-vector-space of M^*N dimensions. We orthogonalize the $M^*N - 1$ remaining regressor, and the vector $Y1$ according to the adjusted regressor $V_{i_{pert}}^{opt} 1$:

$$V_i^{j\perp} = V_i^j - \langle V_i^j | V_{i_{pert}}^{opt} 1 \rangle V_{i_{pert}}^{opt} 1 \quad (20)$$

$$Y^\perp 1 = Y - \langle Y | V_{i_{pert}}^{opt} 1 \rangle V_{i_{pert}}^{opt} 1 \quad (21)$$

Therefore, we make the library updating:

$$V_w = \begin{pmatrix} V_1^{1\perp}(x_1) & \dots & V_N^{1\perp}(x_1) & \dots & V_1^{j\perp}(x_1) & \dots & V_N^{j\perp}(x_1) & \dots & V_1^{M\perp}(x_1) & \dots & V_N^{M\perp}(x_1) \\ V_1^{1\perp}(x_2) & \dots & V_N^{1\perp}(x_2) & \dots & V_1^{j\perp}(x_2) & \dots & V_N^{j\perp}(x_2) & \dots & V_1^{M\perp}(x_2) & \dots & V_N^{M\perp}(x_2) \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ V_1^{1\perp}(x_N) & \dots & V_N^{1\perp}(x_N) & \dots & V_1^{j\perp}(x_N) & \dots & V_N^{j\perp}(x_N) & \dots & V_1^{M\perp}(x_N) & \dots & V_N^{M\perp}(x_N) \end{pmatrix} \quad (22)$$

We will have:

$$V_w(t,d) = \{V_i^{j\perp}\}_{(i=[1..N], j=[1..M]) \setminus \{i_{pert}\}} \quad (23)$$

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

The model being at this stage, $g(X) = \omega_i 1 * V_{i_{pert}}^{opt} 1$, can be represented by the figure 4.

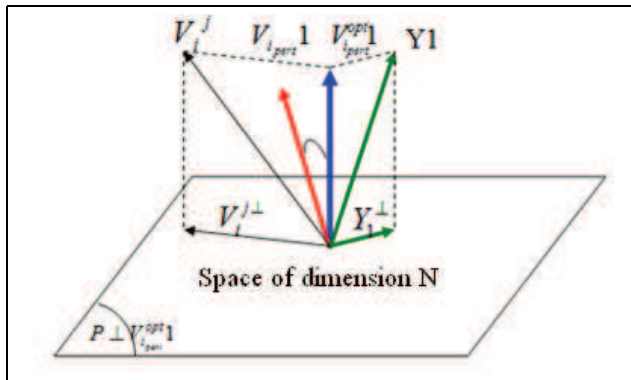


Fig. 4. Orthogonal projection on the optimal regressor

To the following iteration we increment the number of $N_w=N_w+1$ wavelet. We apply the same stages described above. Let's suppose achieved $i-1$ iterations: We did $i-1$ selections, optimizations, and orthogonalizations in order to get the $i-1$ adjusted regressors ($V_{i_{pert}}^{opt} 1, \dots, V_{i_{pert}}^{opt} i-1$) we reset $i-1$ parameters of the network.

The network $g(x)$ can be written at the end of the iteration $i-1$ as:

$$g(x) = \sum_{i=1}^{i-1} \omega_i^{opt} * V_{i_{pert}}^{opt} \quad (24)$$

We have N_w-i+1 regressors to represent the signal Y_i in a space of N^*M-i+1 dimensions orthogonal to ($V_{i_{pert}}^{opt} 1, \dots, V_{i_{pert}}^{opt} i-1$)

We apply the same principle of selection as previously. The index i_{pert} of the selected regressor can be written as:

$$i_{pert} i(i,j) = \arg \max_{i,j} \frac{\langle Y | V_i^j \rangle}{\|Y\| \|V_i^j\|} \tag{25}$$

with $(i=[1..N], j=[1..M]) \setminus \{i_{pert}^1, \dots, i_{pert}^{i-1}\}$

Since the regressor is orthogonal to $(V_{i_{pert}}^{opt} 1, \dots, V_{i_{pert}}^{opt} i - 1)$, we make the updating of the library, then we optimize the regressor and finally an orthogonalization.

Finally, after N iteration, we construct a wavelets network of N wavelets in the hidden layer that approximates the signal Y.

As a consequence, the parameters of the network are:

$$T^{opt} = \left\{ t_{i_{pert}}^{opt} \right\}_{i_{pert}=[1..Nopt]} \tag{26}$$

$$d^{opt} = \left\{ d_{i_{pert}}^{opt} \right\}_{i_{pert}=[1..Nopt]} \tag{27}$$

$$\omega^{opt} = \left\{ \omega_{i_{pert}}^{opt} \right\}_{i_{pert}=[1..Nopt]} \tag{28}$$

The obtained model $g(x)$ can be written under the shape:

$$g(x) = \sum_{i=1}^{Nopt} \omega_i^{opt} * V_i^{opt} \tag{29}$$

4. Interpolation of 1D data

4.1 Mathematical formulation

The mathematical formulation of the interpolation of 1D data can be presented in the following way:

Let the set of the points $E = \{(x_k, y_k) / k = 0, 1, \dots, k\}$. We want to recover N samples of $f(x)$ as $f(x) = y_k$ for $k = 0, 1, \dots, k$.

The set E represents the constraints of the problem. With this formulation the function $f(x)$ pass inevitably by the set points of E. In practice, the constraints can contain noise. In this case, the signal that we want to rebuild doesn't necessarily pass by the points of the set E, the interpolation becomes then a problem of approximation:

Once we know the function $f(x)$ on the set of the domain $x \in [0, \dots, N]$, the problem is to recover $f(x)$ for $x > N$. This formulation will be called extrapolation of the signal $f(x)$.

Interpolation is a problem of signal reconstruction from samples is a badly posed problem by the fact that infinity of solutions passing by a set of points (Figure 5). For this reason supplementary constraints that we will see in the presentation of the different methods of interpolation, must be taken in consideration to get a unique solution.

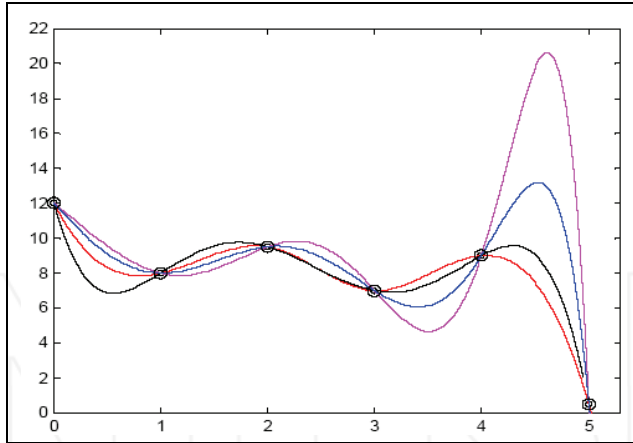


Fig. 5. Infinity of curves passing by a set of points

4.2 Interpolation of 1D data using wavelets networks

4.2.1 Regularization

The variational method is often used for the regularization [Buli80], [Boor63], [Terz86]. The principle of this method is to minimize the function $\xi(f)$ defined as the following manner:

$$\xi(f) = \alpha S(f) + \gamma C(f) \text{ with } \alpha = 1 - \gamma \tag{30}$$

The function $\xi(f)$ is the sum of a stabilizing function $S(f)$ and of a cost function $C(f)$. The parameter $\gamma \in [0, 1]$ is a constant of adjustment between these two functions. When γ goes toward zero, the problem of interpolation turns into a problem of approximation. The stabilizing function $S(f)$ fixes the constraint of curve smoothing and it is defined as the following way:

$$S(f) = \int_D \left(\frac{\partial^2 f}{\partial x^2} \right)^2 dx \tag{31}$$

Where, D represents the domain of interest. The cost function $C(f)$ characterizes the anomalies between the rebuilt curve and the initial constraints, this function can be written by:

$$C(f) = \sum_{x_k \in E_p} [f(x_k) - y_k]^2 \tag{32}$$

Where $E_p = \{(x_k, y_k) \mid k = 1, 2, \dots, K\}$ represents the set of the known points or the signal constraints.

4.2.2 Discretization

With the regularization, it is difficult to get an analytic solution. The discretization is useful and several methods can be used. Grimson [Grim83] uses the finished differences to approximate the differential operators; while Terzopoulos [Terz86] uses finished elements to get and solve an equation system (Grimson and Terzopoulos used the quoted methods

while treating the case 2D). In the approach that follows, the function f is written like a linear combination of basic functions:

$$\hat{f}(x) = \sum_{i=0}^{N-1} W_i \Psi_i(x) \tag{33}$$

Where N is the domain dimension and W_i the coefficients. The basic functions $\Psi_i(x)$ are localized to $x = i\Delta$, $\Psi_i(x) = \Psi(x - i\Delta)$

While substituting (33) in (32) and (31), the function $\xi(f)$ can be rewritten as the following way:

$$\xi = \alpha \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} W_i t_{ij} W_j + \gamma \sum_{k=1}^K \left[\sum_{i=0}^{N-1} W_i \Psi_i(x_k) - y_k \right]^2 \tag{34}$$

Where t_{ij} is a function of basic functions:

$$t_{ij} = \int_D \left(\frac{\partial^2 \Psi_i}{\partial x^2} \right) \left(\frac{\partial^2 \Psi_j}{\partial x^2} \right) dx \tag{35}$$

Several wavelets functions can be used like activation function. The Figure 6 gives the curve of a new wavelets based on Beta function given by the following definition:

4.2.3 Definition

The 1D Beta function as presented in [Alim03] and in [Aoui02] is a parametrable function defined by $\beta(x) = \beta_{x_0, x_1, p, q}(x)$ with x_0, x_1, p and q as real parameters verifying: $x_0 < x_1$, and:

$$x_c = \frac{px_1 + qx_0}{p + q} \tag{36}$$

Only the case for $p > 0$ and $q > 0$ will be considered. In this case the Beta function is defined as:

$$\beta(x) = \begin{cases} \left(\frac{x - x_0}{x_c - x_0} \right)^p \left(\frac{x_1 - x}{x_1 - x_c} \right)^q & \text{if } x \in [x_0, x_1] \\ 0 & \text{else} \end{cases} \tag{37}$$

4.2.4 Axiom [Ben Amar 2006]

$\forall n \in \mathbb{N}, (p, q) \in \mathfrak{R}_+^2, p = q$, and $n < p$; the n^{th} derivatives of 1D Beta function are wavelets [Amar06]. Let's note by Beta^n the n^{th} derivative of Beta function.

$$\Psi_n(x) = \text{Beta}^n = \frac{d^n \beta(x)}{dx^n} \tag{38}$$

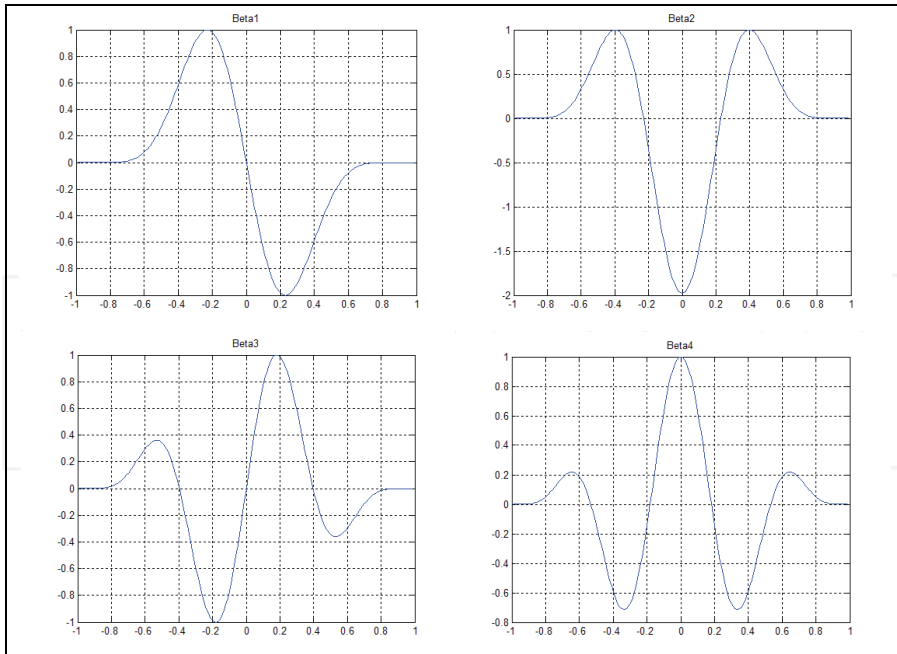


Fig. 6. Different shapes of Beta wavelets

4.2.5 Example: Interpolation of 1D data using classical wavelets network CWNN

In this example, we want to rebuild three signals $F_1(x)$, $F_2(x)$ and $F_3(x)$ defined by equations (39), (40) and (41). We have a uniform distribution with a step of 0.1 known samples. For the reconstruction, we used a CWNN composed of 12 wavelets in hidden layer and 300 trainings iterations. We note that for Beta wavelets we fix the parameter $p=q=30$.

$$F_1(x) = \begin{cases} -2.186x - 12.864 & \text{for } x \in [-10, -2[\\ 4.246x & \text{for } x \in [-2, -0[\\ 10 \exp(-0.05x - 0.5) \sin(x(0.03x + 0.7)) & \text{for } x \in [0, 10[\end{cases} \quad (39)$$

$$F_2(x) = 0.5x \sin(x) + \cos^2(x) \text{ for } x \in [-2.5, 2.5] \quad (40)$$

$$F_3(x) = \sin c(1.5x) \text{ for } x \in [-2.5, 2.5] \quad (41)$$

Table 1. gives the final normalized root mean square error (NRMSE) of test given by equation (42) after 300 trainings iterations for the F_1 , F_2 and F_3 signals. We define the NRMSE as:

$$NRMSE = \frac{1}{N} \sqrt{\frac{\sum_{i=1}^N (\bar{y}_i - y_i)^2}{\sum_{i=1}^N (y_i)^2}} \quad (42)$$

Where N is the number of sample and \bar{y}_i the real output.

	M-hat	Polywog ¹	Slog ¹	Beta ¹	Beta ²	Beta ³
F ₁	7.2928e-2	4.118e-2	3.1613e-2	1.7441e-2	1.2119e-2	1.4645e-1
F ₂	7.6748e-8	1.0343e-6	2.3954e-6	3.7043e-8	5.3452e-9	1.2576e-9
F ₃	3.1165e-7	2.3252e-6	1.2524e-6	5.5152e-7	1.9468e-7	1.9674e-6

Table 1. Normalized root mean square error of test for different activation wavelets function

4.2.5 Example: interpolation of 1D data using MLWNN

We intend to approximate the F₁, F₂, F₃ using MLWNN, composed of a library of 6 mother wavelets (from Beta¹ to Beta³, Mexican hat, polywog¹ and Slog¹), in the same condition as the example of approximation using CWNN.

	M-hat	Polywog ¹	Slog ¹	Beta ¹	Beta ²	Beta ³	NRMSE
F ₁	4	0	3	3	0	2	3.79841e-3
F ₂	1	3	3	1	1	3	4.66143e-11
F ₃	1	3	1	1	1	5	3.84606e-8

Table 2. Normalized root mean square error of test and selected mother wavelets

To reconstruct the F₁ signal with a NRMSE of 3.79841e-3 using 12 wavelets in hidden layer the best regressors for MLWNN are: 4 wavelets from the Mexican hat mother wavelet, 0 wavelet from the polywog¹, 3 wavelets from the Slog¹, 3 wavelets from Beta¹, 0 wavelet from Beta² and 2 wavelets from the Beta³ mother wavelets. When using a CWNN the best NRMSE of reconstruction is obtained with Beta² mother wavelet and it is equal to 1.2119e-2. For F₂ signal the NRMSE is equal to 4.66143e-11 using MLWNN whereas it is of 1.2576e-9 using CWNN with Beta³ mother wavelet. Finally for F₃ signal we have a NRMSE of 3.84606e-8 for a MLWNN over 1.9468e-7 as the best value for a CWNN.

5. 2 Dimensional data interpolation

Previously, for every described method in 1D, the case in two dimensions is analogous, while adding one variable in the equations.

5.1 Mathematical formulation

The mathematical formulation of 2D data interpolation can be presented in an analogous way to the one described in the 1D case [Yaou94] (we will suppose that we want to rebuild an equally-sided surface):

Let E the set of the points $E = \{(x_k, y_k, z_k) / k = 0, 1, \dots, k\}$ we wants to recover N×N samples of f(x, y) as f(x_k, y_k) = z_k for k = 1, ..., K.

The set E represents the constraints of the problem. With this formulation the function f(x, y) passes inevitably by the points of the set E. In practice, the constraints can be noisy. In this

case, the signal that we want to rebuild doesn't necessarily pass by the points of the set E. The interpolation becomes then a problem of approximation.

5.2 Method using wavelets networks

The formulations for the 2D case are given by:

$$S(f) = \iint_{\Omega} \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right] d\Omega \quad (43)$$

$$C(f) = \sum_{x_k \in E} \sum_{y_k \in E} [f(x_k, y_k) - Z_k]^2 \quad (44)$$

$$f(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} w_{ij} \Psi_{ij}(x, y) \quad (45)$$

Since the interpolated basic functions (wavelets) are separable, this will always be the case in this survey:

$$\Psi_{ij}(x, y) = \Psi_i(x) \Psi_j(y) \quad (46)$$

$$t_{ijrs} = \iint_{\Omega} \left[\left(\frac{\partial^2 \Psi_{ij}}{\partial x^2} \right) \left(\frac{\partial^2 \Psi_{rs}}{\partial x^2} \right) + 2 \left(\frac{\partial^2 \Psi_{ij}}{\partial x \partial y} \right) \left(\frac{\partial^2 \Psi_{rs}}{\partial x \partial y} \right) + \left(\frac{\partial^2 \Psi_{ij}}{\partial y^2} \right) \left(\frac{\partial^2 \Psi_{rs}}{\partial y^2} \right) \right] d\Omega \quad (47)$$

5.2.1 Example: approximation of 2D data using CWNN

To compare the performances of the method using wavelets networks and classic wavelets networks, four surfaces have been chosen. These surfaces are used by Franke [Fran79] and are also described in [Renk88] for x and $y \in [0, 1]$:

$$S_1(x, y) = \frac{e^{-\frac{81}{16}[(x-0.5)^2 + (y-0.5)^2]}}{3} \quad (48)$$

$$S_2 = \left(\frac{x+2}{2} \right)^5 \left(\frac{2-x}{2} \right)^5 \left(\frac{y+2}{2} \right)^5 \left(\frac{2-y}{2} \right)^5 \quad (49)$$

$$S_3(x, y) = \frac{3.2(1.25 + \cos(5.4y))}{(6 + 6(3x-1)^2)} \quad (50)$$

$$S_4(x, y) = (x^2 - y^2) \sin(5x) \quad (51)$$

The Figure 7 represents these four surfaces of 21X21 samples.

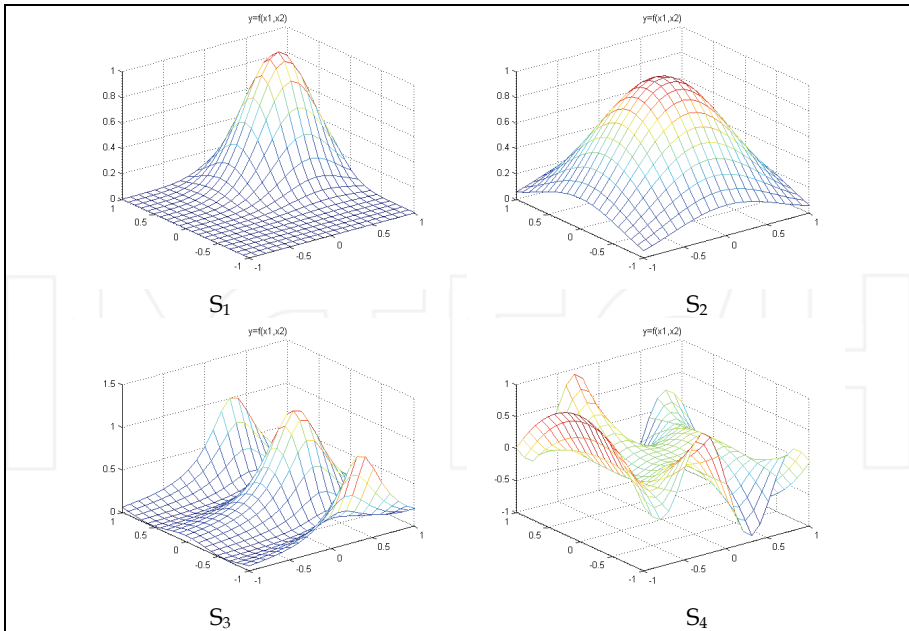


Fig. 7. Surfaces $S_1(x, y)$, $S_2(x, y)$, $S_3(x, y)$ and $S_4(x, y)$ represented by their 21X21 samples
 The four following samplings are considered in order to rebuild these surfaces:

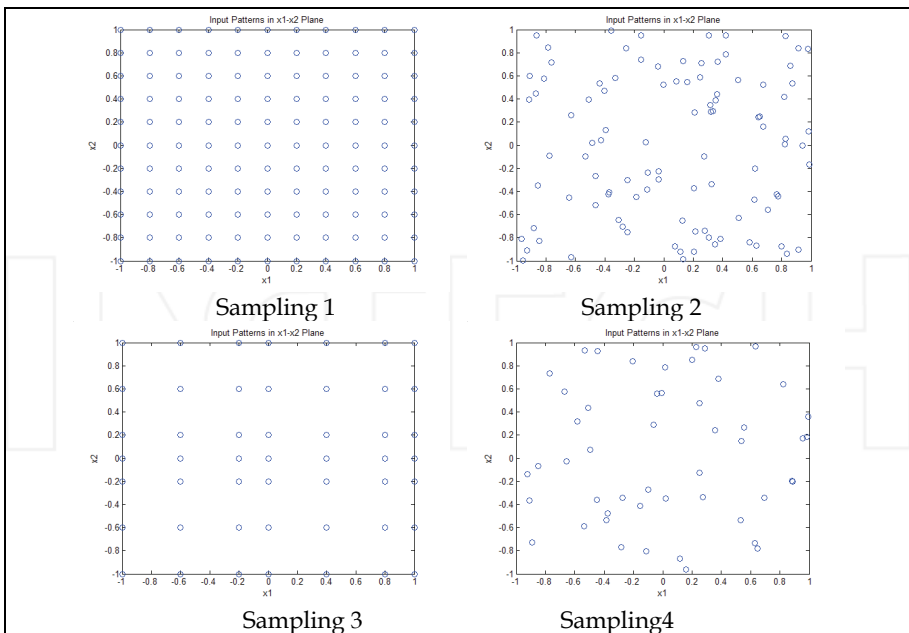


Fig. 8. Samplings considered for reconstruction of the surfaces $S_1(x, y)$, $S_2(x, y)$, $S_3(x, y)$ and $S_4(x, y)$

Surfaces	Sampling	Beta ¹	Beta ²	Beta ³	M-hat	Slog ¹	Polywog ¹
S ₁	1	4.5472e-6	1.5881e-5	1.2783e-5	3.6561e-5	6.2610e-6	8.0144e-7
S ₁	2	3.6612e-6	1.9917e-6	2.6849e-6	4.2004e-5	1.4369e-5	7.6421e-6
S ₁	3	1.7730e-6	2.3737e-6	1.5610e-6	3.5810e-6	9.3080e-6	5.2441e-7
S ₁	4	2.4055e-7	4.1873e-6	8.6403e-7	7.6317e-6	5.2528e-6	6.7456e-6
Surfaces	Sampling	Beta ¹	Beta ²	Beta ³	M-hat	Slog ¹	Polywog ¹
S ₂	1	5.2167e-6	7.9647e-6	3.0107e-6	2.2445e-5	8.4106e-5	5.2539e-6
S ₂	2	6.2544e-7	1.2262e-6	7.9519e-7	4.3444e-5	4.5899e-5	9.1619e-7
S ₂	3	7.2351e-7	2.8619e-6	1.9735e-6	8.7916e-5	3.0003e-5	4.7101e-7
S ₂	4	2.1238e-7	8.2108e-7	5.5915e-7	1.7681e-5	1.0731e-5	1.0287e-6
Surfaces	Sampling	Beta ¹	Beta ²	Beta ³	M-hat	Slog ¹	Polywog ¹
S ₃	1	9.3210e-4	7.4748e-4	6.0998e-4	4.4958e-3	2.7186e-3	9.4392e-4
S ₃	2	4.0176e-4	2.6073e-4	1.5930e-4	1.2234e-3	1.2249e-3	7.5690e-4
S ₃	3	2.3361e-4	2.7506e-4	2.7690e-4	1.1170e-3	8.5758e-4	1.3850e-4
S ₃	4	1.1620e-4	2.6754e-4	1.9940e-4	2.1405e-4	1.1916e-4	2.2120e-4
Surfaces	Sampling	Beta ¹	Beta ²	Beta ³	M-hat	Slog ¹	Polywog ¹
S ₄	1	3.8187e-3	4.6391e-3	2.2186e-3	3.3708e-2	3.0934e-3	4.8993e-3
S ₄	2	1.5365e-3	2.4675e-3	1.3099e-3	9.9976e-3	4.8904e-3	2.5167e-3
S ₄	3	3.7501e-3	1.7048e-3	5.4345e-4	1.2712e-2	3.0359e-3	2.2196e-3
S ₄	4	3.2641e-4	1.4836e-3	6.2990e-4	1.5668e-3	2.0464e-4	2.2768e-4

Table 3. Normalized Root Mean square error of test for the surfaces $S_1(x, y)$, $S_2(x, y)$, $S_3(x, y)$ and $S_4(x, y)$ using CWNN

Table 3 represents the NRMSE of reconstruction of the four considered surfaces, using classical wavelets network constructed with 12 wavelets in hidden layer and based on Beta, Mexican Hat, Slog¹ and Polywog¹ wavelets [Bellii05]. This table informs that the number of samples to consider as well as their disposition for the reconstruction is important.

For a same number of samples, it is preferable to use a uniform sampling than a non uniform one, the more the number of samples is important and the better is the quality of reconstruction.

5.2.1 Example: approximation of 2D data using MLWNN [Bellii07-b]

The same surfaces are used in the same conditions but using MLWNN with a library composed of two mother wavelets (Beta¹ and Beta³). Experimental results are given in the following table.

Table 3 and table 4 inform that the number of samples to consider as well as their disposition for the reconstruction is important:

For a same number of samples, it is preferable to use a uniform sampling than a non uniform one, the more the number of samples is important and the better is the quality of reconstruction.

When comparing table 3 and table 4 we can say that the performances obtained in term of NRMSE using the MLWNN algorithm are often very better that the one obtained with the CWNN. This shows that the proposed procedure brings effectively a better capacity of approximation using the parametrable Beta wavelets [Belli07-b].

Surfaces	Sampling	Beta ¹ + Beta ³		
		NMSE	P ₁	P ₂
S ₁	1	6.0509e-7	5	7
S ₁	2	3.4558e-7	5	7
S ₁	3	5.7987e-7	4	8
S ₁	4	4.0519e-7	5	7
Surfaces	Sampling	Beta ¹ + Beta ³		
		NMSE	P ₁	P ₂
S ₂	1	6.1536e-6	3	9
S ₂	2	1.1164e-6	2	10
S ₂	3	7.0519e-7	1	11
S ₂	4	1.1629e-6	4	8
Surfaces	Sampling	Beta ¹ + Beta ³		
		MSET	P ₁	P ₂
S ₃	1	5.1778e-4	6	6
S ₃	2	8.8298e-5	7	5
S ₃	3	7.8227e-5	6	6
S ₃	4	1.1098e-4	6	6
Surfaces	Sampling	Beta ¹ + Beta ³		
		NMSE	P ₁	P ₂
S ₄	1	2.1e-3	8	4
S ₄	2	1.4e-3	6	6
S ₄	3	1.1e-3	10	2
S ₄	4	1.1e-3	6	6

Table 4. Normalized Root Mean square error of test for the surfaces S₁(x, y), S₂(x, y), S₃(x, y) and S₄(x, y) using MLWNN

6. 3 Dimensional data interpolation

The case of 3 dimensions is analogous to 1D or 2D case. The reconstruction of sampled data using wavelets networks is deduced from the 1D or 2D case.

6.1 Mathematical formulation

The mathematical formulation of 3D data interpolation can be presented in an analogous way to the one described for the 1D case (we will suppose that we want to rebuild an equally-sided volume):

Let E the set of the points $E = \{(x_k, y_k, z_k, q_k) / k = 0, 1, \dots, k\}$ we wants to recover N×N×N samples of f(x, y, z) as $f(x_k, y_k, z_k) = q_k$ for $k = 1, \dots, K$.

The set E represents the constraints of the problem. With this formulation the function f(x, y, z) passes inevitably by the points of the set E. In practice, the constraints can be noisy.

In this case, the signal that we want to rebuild doesn't necessarily pass by the points of the set E. The interpolation becomes then a problem of approximation:

The problem of extrapolation is to recover the values of the function $f(x, y, z)$ for x, y and z not belonging to the domain of interpolation.

The problem of reconstruction of a volume from samples is a badly definite problem because an infinity volume passing by a set of points exists. For this reason, some supplementary constraints must be taken into consideration to get a unique solution.

6.2 Method using wavelets networks

Formulations for the 3D case are given by:

$$S(f) = \iiint_V \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 + \left(\frac{\partial^2 f}{\partial z^2} \right)^2 + 2 \left(\frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \right)^2 + 2 \left(\frac{\partial f}{\partial x} \frac{\partial f}{\partial z} \right)^2 + 2 \left(\frac{\partial f}{\partial y} \frac{\partial f}{\partial z} \right)^2 \right] dV \quad (52)$$

$$C(f) = \sum_{x_k \in E} \sum_{y_k \in E} \sum_{z_k \in E} [f(x_k, y_k, z_k) - q_k]^2 \quad (53)$$

$$f(x, y, z) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} w_{ijl} \Psi_{ijl}(x, y, z) \quad (54)$$

Since the interpolated basic functions (wavelets) are separable, this will always be the case in this survey:

$$\Psi_{ijl}(x, y, z) = \Psi_i(x) \Psi_j(y) \Psi_l(z) \quad (55)$$

$$t_{ijrstu} = \iiint_V \left[\left(\frac{\partial^2 \Psi_{ijr}}{\partial x^2} \right) \left(\frac{\partial^2 \Psi_{stu}}{\partial x^2} \right) + \left(\frac{\partial^2 \Psi_{ijr}}{\partial y^2} \right) \left(\frac{\partial^2 \Psi_{stu}}{\partial y^2} \right) + \left(\frac{\partial^2 \Psi_{ijr}}{\partial z^2} \right) \left(\frac{\partial^2 \Psi_{stu}}{\partial z^2} \right) + \right. \\ \left. 2 \left(\frac{\partial \Psi_{ijr}}{\partial x} \right) \left(\frac{\partial \Psi_{ijr}}{\partial y} \right) \left(\frac{\partial \Psi_{stu}}{\partial x} \right) \left(\frac{\partial \Psi_{stu}}{\partial x} \right) + \right. \\ \left. 2 \left(\frac{\partial \Psi_{ijr}}{\partial x} \right) \left(\frac{\partial \Psi_{ijr}}{\partial z} \right) \left(\frac{\partial \Psi_{stu}}{\partial z} \right) \left(\frac{\partial \Psi_{stu}}{\partial z} \right) + \right. \\ \left. 2 \left(\frac{\partial \Psi_{ijr}}{\partial y} \right) \left(\frac{\partial \Psi_{ijr}}{\partial z} \right) \left(\frac{\partial \Psi_{stu}}{\partial y} \right) \left(\frac{\partial \Psi_{stu}}{\partial z} \right) \right] dV \quad (56)$$

6.3 Example: approximation of 3D data using MLWNN

We used the GavabDB 3D face database for automatic facial recognition experiments and other possible facial applications like pose correction or register of 3D facial models. The database GavabDB contains 427 images of 3D meshes of the facial surface. These meshes correspond to 61 different individuals (45 male and 16 female), and 9 three dimensional images are provided for each person. The total of the database individuals are Caucasian and their age is between 18 and 40 years old.

Each image is a mesh of connected 3D points of the facial surface without the texture information for the points. The database provides systematic variations in the pose and the facial expressions of the individuals. In particular, there are 2 frontal views and 4 images with small rotations and without facial expressions and 3 frontal images that present different facial expressions.

The following experiment is performed on the GavabDB 3D face database and its purpose is to evaluate the MLWNN that we employ against the CWNN in term of 3D face reconstruction.

For faces reconstruction quality measurement we adopt the common use of NMSE given by:

$$NRMSE = \frac{1}{N * M} \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M (\overline{Z}_{i,j} - Z_{i,j})^2}{\sum_{i=1}^N \sum_{j=1}^M (Z_{i,j})^2}} \quad (57)$$

Where $\overline{Z}_{i,j}$ is the depth value of pixels in the reconstructed face; and $Z_{i,j}$ is the depth value of pixels in the original face. To check the utility of the MLWNN, experimental studies are carried out on the GavabDB 3D face database. We used 61 frontal views with two different dimensions: 20*20 and 50*50. The results of comparison are presented in Table 5.

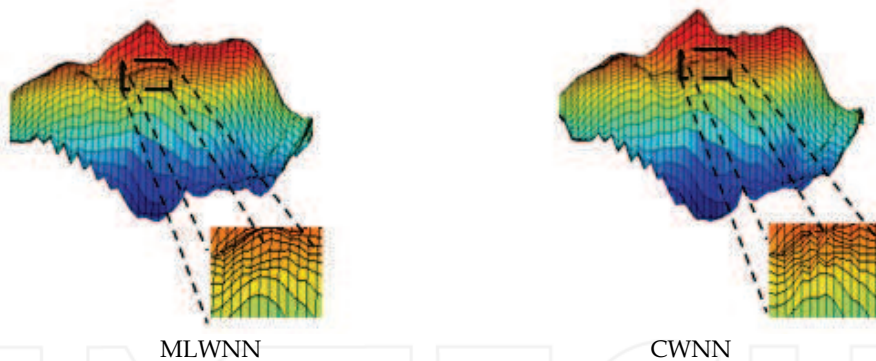


Fig. 9. 3D face reconstruction using MLWNN and CWNN

	Face 20*20		Face 50*50	
	CWNN	MLWNN	CWNN	MLWNN
5 faces	3.52000*10-5	6.3500*10-6	0.90560*10-5	5.6960*10-6
20 faces	4.55375*10-5	9.0250*10-6	1.13640*10-5	6.8320*10-6
40 faces	4.94625*10-5	9.1562*10-6	1.25040*10-5	7.4600*10-6
61 faces	5.42500*10-5	9.1762*10-6	1.31970*10-5	7.9121*10-6

Table 5. Evaluation in term of NMSE of 3D face reconstruction using MLWNN and CWNN

7. Conclusion

In this chapter, we described a new training algorithm for multi library wavelets network. We needed a selection procedure, a cost function and an algorithm of minimization for the evaluation. To succeed a good training, we showed that it was necessary to unite good ingredients. Indeed, a good algorithm of minimization finds a minimum quickly; but this one is not necessarily satisfactory.

The use of a selection algorithm is fundamental. Indeed, the good choice of regressors guarantees a more regular shape of the cost function; the global minima correspond well to the "true" values of the parameters, and avoid the local minimum multiplication. So the cost function present less local minima and the algorithms of evaluation find the global minimum more easily.

For the validation of this algorithm we have presented a comparison between the CWNN and MLWNN algorithm in the domain of 1D, 2D and 3D function approximation.

Many examples permitted to compare the capacity of approximation of MLWNN and CWNN. We deduce from these examples that:

- The choice of the reconstruction method essentially depends on the type of data that we treat,
- The quality of reconstruction depends a lot on the number of samples used and on their localizations.

Also we have define a new Beta wavelets family that some one can see that they are more superior then the classic one in term of approximation and we demonstrate in [BELLIL07] that they have the capacity of universal approximation.

As future work we propose a hybrid algorithm, based on MLWNN and genetic algorithm and the GCV (Generalised Cross validation) procedure to fix the optimum number of wavelets in hidden layer of the network, in order to model and synthesis PID controller for non linear dynamic systems.

8. References

- A. Alimi, (2003). The Bêta-Based Neuro-Fuzzy System: Genesis and Main Properties, *submitted to TASK Quarterly Journal, Special Issue on "Neural Networks"* edited by W. Duch and D. Rutkowska, vol. 7, no. 1, pp. 23-41, 2003.
- C. Ben Amar, W. Bellil, M.A. Alimi, (2006). Beta Function and its Derivatives: A New Wavelet Family, *Transactions on Systems, Signals & Devices Volume 1, Number 3*, pp. 275-293, 2005-2006.
- C. Aouiti, A.M. Alimi and A. Maalej, (2002). Genetic Designed Bêta 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, pp. 975-1005, 2002.
- C. De Boor, (1978). A practical guide to splines, *Springer-Verlag, New-York*, 1978.
- C. J. Lin, C. C. Chin, C. L. Lee, (2003). A wavelet based neuro-fuzzy system and its applications, *Proceedings of the Int. Joint Conference on Neural Networks*, vol. 3, pp. 1921-1926, 2003.
- D. W. C. Ho, P. A. Zhang, J. Xu, (2001). Fuzzy wavelet networks for function learning, *IEEE Trans. on Fuzzy Systems*, vol. 9, issue 1, pp. 200-211, 2001.

- D. Withley et al, (1990). Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity, *Parallel Computing*.
- F. Gruau and D. Whitley, "Adding learning to the cellular developmental process: a comparative study", *Evolutionary Computation*, 1(3), 1993.
- Fabrice Rossi and Cédric Gégout, J. (1994). Geometrical Initialization, Parametrization and Control of Multilayer Perceptrons: Application to Function Approximation, *In Proceedings of WCCI ICNN, volume 1*, pp. 546-550, June 1994, Orlando (Florida).
- L. Angrisani, P. Daponte, M. D'Apuzzo, (2001) Wavelet network-based detection and classification of transients, *IEEE Trans. Instrum. Meas.* 50 (5), pp. 1425-1435, 2001.
- L. Wessels and E. Barnard. (1992). Avoiding False Local Minima by Proper Initialisation of Connections, *IEEE Trans. on Neural Networks*, 3(6), 1992.
- Q. Zhang, A. Benveniste, (1992). Wavelet Networks, *IEEE Trans. on Neural Networks* 3 (6) pp. 889-898, 1992.
- R. Bulirsch and J. Stoer, (1980). Introduction to numerical analysis, *Springer-Verlag, New-York*, 1980.
- R. M. Rao, A. S. Bopardikar, (2004). Wavelet Transform: Introduction to theory and applications, *Pearson education*, 2004.
- R.H. Franke, (1979). A critical comparison of some methods for interpolation of scattered data, *Tech. Rep. NPS-53-79-003, Dept. of Mathematics*, Naval Postgraduate School, Monterey, Calif., 1979.
- R.J. Renka, (1988). Multivariate interpolation of large sets of scattered data, *ACM Trans. on Math. Software*, vol. 14, n° 2, pp. 139-148, 1988.
- S. Xiaoyun et al. (2004). Neural network with adaptive genetic algorithm for eddy current nondestructive testing, *IEEE Proceedings of the fifth world congress on intelligent control and automation*, pp. 2034-2037, 2004.
- S.Chen, C. Cowan, and P. Grant, (1989). Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Trans. On Neuronal Networks*, vol. 2, pp.302-309, March 1989.
- W. Bellil, C. Ben Amar C. and M.A Alimi, (2007-a). Multi Library Wavelet Neural Network for lossless image compression, *International Review on Computers and Software (IRECOS) Vol. 2 N. 5*, ISSN 1828-6003, pp 520-526, September 2007.
- W. Bellil, C. Ben Amar C. And M.A Alimi, (2005). Beta wavelets networks for function approximation, *International Conference on Adaptive and Natural Computing Algorithms, ICANNGA05*, Coimbra Portugal, SpringerWien NewYork, pp 18-21, 2005.
- W. Bellil, M. Othmani, C. Ben Amar, (2007-b). Initialization by Selection for multi library wavelet neural network training, *Proceeding of the 3rd international workshop on artificial neural networks and intelligent information processing ANNIIP07, in conjunction with 4th international conference on informatics in control, automation and robotics, ICINCO*, pp30-37, 2007.
- W.E.L. Grimson, (1983). An implementation of a computational theory of visual surface interpolation, *Comp. Vis. Grap. Ima. Proc.*, vol. 22, pp. 39-69, 1983.
- Y. Chen, B. Yang, J. Dong, (2006). A Local Linear Adaptive Wavelet Neural Network, *Transaction of the Institute of Electric Engineers of Japan*, v. 69, no. 4-6, pp. 449-465, 2006.

- Y. Oussar et al, '(1998). Training wavelet networks for nonlinear dynamic input-output modeling, *Neurocomputing*, 20(1-3):pp. 173-188, 1998.
- Z. Michalewicz, (1993). A Hierarchy of Evolution Programs: An Experimental Study, *Evolutionary Computation*.

INTECH

INTECH



Advances in Robotics, Automation and Control

Edited by Jesus Aramburo and Antonio Ramirez Trevino

ISBN 978-953-7619-16-9

Hard cover, 472 pages

Publisher InTech

Published online 01, October, 2008

Published in print edition October, 2008

The book presents an excellent overview of the recent developments in the different areas of Robotics, Automation and Control. Through its 24 chapters, this book presents topics related to control and robot design; it also introduces new mathematical tools and techniques devoted to improve the system modeling and control. An important point is the use of rational agents and heuristic techniques to cope with the computational complexity required for controlling complex systems. Through this book, we also find navigation and vision algorithms, automatic handwritten comprehension and speech recognition systems that will be included in the next generation of productive systems developed by man.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wajdi Bellil, Mohamed Othmani, Chokri Ben Amar and Mohamed Adel Alimi (2008). A New Algorithm for Initialization and Training of Beta Multi-Library Wavelets Neural Network, *Advances in Robotics, Automation and Control*, Jesus Aramburo and Antonio Ramirez Trevino (Ed.), ISBN: 978-953-7619-16-9, InTech, Available from:

http://www.intechopen.com/books/advances_in_robotics_automation_and_control/a_new_algorithm_for_initialization_and_training_of_beta_multi-library_wavelets_neural_network

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821