Published in *Neurocomputing vol. 34, pp. 131-143 (2000).*

Initialization by Selection for Wavelet Network Training

Yacine OUSSAR, Gérard DREYFUS

Laboratoire d'Électronique
École Supérieure de Physique et de Chimie Industrielles
10, rue Vauquelin
F - 75231 PARIS Cedex 05, FRANCE.
Phone: 33 1 40 79 45 41 Fax: 33 1 40 79 44 25
E-Mail: Gerard.Dreyfus@espci.fr, Yacine.Oussar@espci.fr

Abstract

We present an original initialization procedure for the parameters of feedforward wavelet networks, prior to training by gradient-based techniques. It takes advantage of wavelet frames stemming from the discrete wavelet transform, 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. Results obtained for the modeling of two simulated processes are compared to those obtained with a heuristic initialization procedure, and demonstrate the effectiveness of the proposed method.

Keywords: Wavelet networks, Training, Initializing parameters, Nonlinear static modeling.

I. Introduction.

Among the applications of wavelet functions, nonlinear regression by wavelet networks is attracting a growing interest. Wavelet networks have been used both for static [10, 12] modeling and for dynamic input-output modeling [9].

It was proved in [4] that families of wavelet functions - particularly wavelet frames - are universal approximators, which gives a theoretical basis to their use in the framework of function approximation and process modeling. For wavelet functions, this property can be expressed as follows: any function of $L^2(R)$ can be approximated to any prescribed accuracy with a finite sum of wavelets. Therefore, wavelet networks can be considered as an alternative to neural and radial basis function networks.

In the present article, we present a network initialization procedure that takes advantage of the properties of discrete wavelet frames in order to improve the training efficiency of continuous wavelet frames. We focus on wavelet frames rather than on orthogonal wavelet bases, because the latter must comply with conditions that are seldom feasible.

We first present the wavelet networks that we use, their architectures and the principle of their training. The difference between continuous and discrete wavelet frames will be emphasized, since both approaches will be used at different stages of wavelet network training. After outlining the problem of parameter initialization of a wavelet network, we first introduce a heuristic procedure, and subsequently describe the proposed method. Finally, two examples of static modeling are presented, and the efficiency of the procedure is shown, by comparing its results to those obtained with the heuristic initialization.

II. Wavelet frames and wavelet networks.

Two categories of wavelet functions, namely, orthogonal wavelets and wavelet frames, were developed separately by different groups. Orthogonal wavelet decomposition is usually associated to the theory of multiresolution analysis [8]. The fact that orthogonal wavelets cannot be expressed in closed form is a serious drawback for their application to function approximation and process modeling. Conversely, wavelet frames are constructed by simple operations of translation and dilation of a single fixed function called the *mother wavelet*, which must satisfy conditions that are less stringent than orthogonality conditions.

A wavelet $\phi_j(x)$ is derived from its mother wavelet $\phi(z)$ by the relation:

$$\phi_j(x) = \phi\left(\frac{x - m_j}{d_j}\right) = \phi(z_j) \tag{1}$$

where the translation factor m_j and the dilation factor d_j are real numbers in $\boldsymbol{R}$ and $\boldsymbol{R}_+^*$ respectively. The family of functions generated by ϕ can be defined as:

$$\Omega_c = \left\{ \frac{1}{\sqrt{d_j}}\,\phi\left(\frac{x - m_j}{d_j}\right),\ m_j \in R \text{ and } d_j \in R_+^* \right\} \tag{2}$$

A family Ω_c is said to be a frame of $L^2(R)$ if there exist two constants $c>0$ and $C<+\infty$ such that for any square integrable function f the following inequalities hold:

$$c\,\| f \|^2 \leq \sum_{\substack{j \\ \phi_j \in \Omega_c}} \left| \langle \phi_j , f \rangle \right|^2 \leq C\,\| f \|^2 \tag{3}$$

where $\| f \|$ denotes the norm of function f and $\langle f , g \rangle$ the inner product of functions f and g. Families of wavelet frames of $L^2(R)$ are universal approximators.

For the modeling of multi-variable processes, multidimensional wavelets must be defined. In the present work, we use multidimensional wavelets constructed as the product of N_i scalar wavelets (N_i being the number of variables):

$$\Phi_j(\boldsymbol{x}) = \prod_{k=1}^{N_i} \phi(z_{jk}) \quad \text{with} \quad z_{jk} = \frac{x - m_{jk}}{d_{jk}} \tag{4}$$

where m_j and d_j are the translation and dilation vectors respectively. Families of multidimensional wavelets generated according to this scheme have been shown to be frames of $L^2(R^{N_i})$ [7].

Wavelet networks were first presented in the framework of static modeling in [10, 12]. In the present work we use a similar architecture, where the network output y is computed as:

$$y = y(\boldsymbol{x}) = \sum_{j=1}^{N_w} c_j\, \Phi_j(\boldsymbol{x}) + \sum_{k=0}^{N_i} a_k\, x_k \tag{5}$$

It can be viewed as a network with an input vector of N_i components, a layer of N_w weighted multidimensional wavelets and a linear output neuron. The coefficients of the linear part of the networks will be called direct connections. Such a network is shown on figure 1.

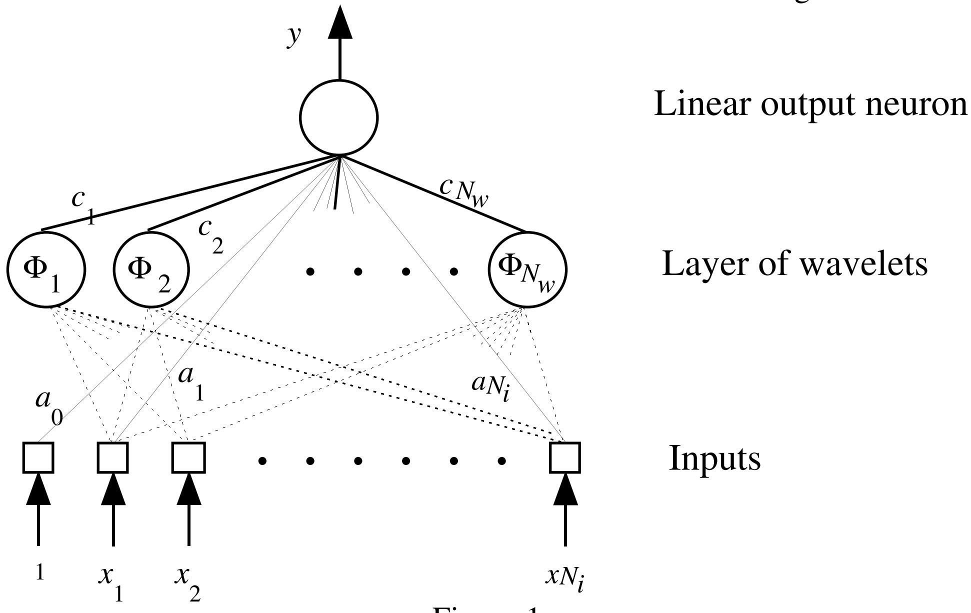


Figure 1
A feedforward wavelet network.

Wavelet network training consists in minimizing the usual least squares cost function:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{n=1}^{N}\left(y_p^n - y^n\right)^2 \tag{6}$$

where vector $\boldsymbol{\theta}$ includes all network parameters to be estimated: translations, dilations, weights of the connections between wavelets and output, and weights of the direct connections; N is the number of elements of the training set, y_p^n is the output of the process for example n, and y^n is the corresponding network output.

Depending on the nature (continuous or discrete) of the wavelet transform it stems from, wavelet network training must be performed differently. This is discussed in the following.

II.1. The continuous approach.

Wavelets stemming from the continuous wavelet transform have the form of relation (2). Since their parameters can take on any continuous real value (note that the dilations must be positive, non-zero), they can be considered as the coefficients of a conventional neural network, to be trained using gradient-based techniques such as stochastic gradient [1, 12] or second order methods.

II.2. The discrete approach.

In the framework of the discrete wavelet transform, a family of wavelets can be defined as:

$$\Omega_d = \left\{ \alpha^{m/2}\, \phi\left(\alpha^m\, x - n\, \beta\right),\ \left(m, n\right) \in \boldsymbol{Z}^2 \right\} \tag{7}$$

where α and β are constants that fully determine, together with the mother wavelet ϕ, the family Ω_d. Actually, relation (7) can be considered as a special case of relation (2), where:

$$\begin{cases} m_j = n\, \alpha^{-m}\, \beta \\ d_j = \alpha^{-m} \end{cases} \tag{8}$$

These relations show that, unlike the continuous approach, wavelet parameters cannot be varied continuously; therefore, gradient-based techniques cannot be used to adjust them. Generally, training wavelet networks stemming from the discrete transform [6, 13] is performed using the Gram-Schmidt selection method [3], which will be briefly described below. This approach usually generates large networks, which are less parsimonious than those trained by gradient-based techniques. This may be a drawback for many applications. In section III below, we show that, although such networks are not very suitable for applications, their properties can be taken advantage of for initializing the translations and dilations of wavelets with continuous parameters.

III. Initializing wavelet networks.

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 (as defined by the examples of the training sequence), if the translation parameter is not chosen appropriately.

Therefore, 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 next sections, we present two initialization procedures that take into account the specific properties of wavelets. The first one is heuristic, whereas the second is based on the selection of discrete wavelets.

III.1. A heuristic initialization procedure.

The present procedure takes into account the domain of input space where the wavelets are not zero. We use the following mother wavelet:

$$\phi(x) = -x\, e^{-\frac{1}{2}x^2} \tag{9}$$

We denote by $[a_k, b_k]$ the domain containing the values of the k-th component of the input vectors of the examples.

- The center of wavelet j is initialized at the center of the parallelepiped defined by the N_i intervals $\{[a_k, b_k]\}$. For the k-th input we have:

$$m_{jk} = \frac{1}{2}\left(a_k + b_k\right) \tag{10}$$

- The dilations parameters of wavelet j are initialized to:

$$d_{jk} = 0.2\left(b_k - a_k\right) \tag{11}$$

These initializations guarantee that the wavelets extend initially over the whole input domain. The choice of the weights is less critical. They are initialized to small random values. This procedure is very simple and requires a small number of operations. Examples of results are shown in section IV.

III.2. An initialization procedure using a selection method.

We propose to make use of wavelet frames stemming from the discrete transform (relation 7) to initialize the translation and dilation parameters of wavelet networks trained using gradient-based techniques. The procedure comprises three steps:
(i) generate a library of wavelets, using a family of wavelets described by relation (7),
(ii) rank all wavelets in order of decreasing relevance,
(iii) use the translations and dilations of the most relevant wavelets as initial values and use a gradient method to train the network thus initialized.
These steps are described in detail in the following subsections.

III.2.1. Generating the library.

Generating the library amounts to finding, among all the members of family Ω_d (relation 7), those wavelets

- -whose dilations belong to a set of discrete values.
- -whose translations are within the parallelepiped defined by the N_i intervals $\{[a_k, b_k]\}$.

Typical values of α and β are $\alpha = 2$ and $\beta = 1$.

In the following, we describe a method for choosing the sets of dilations and translations in the case of scalar wavelets (single input model). Due to the fact that we use multidimensional wavelets as the product of scalar wavelets, the method can be easily extended to the multidimensional case by applying it separately to each input.

The dilation set.

As shown by relation (8), and taking into account our choice for the values of α and β, the dilations are given by 2^{-m}, where m is an integer.

We consider three successive dilations, where the largest gives a wavelet extending over the whole domain of the corresponding variable. Generally, more dilations result in too large a library. For the mother wavelet that we use, this condition can be expressed as:

$$2^{-m} \leq 0.2\left(b_k - a_k\right) \tag{12}$$

Hence:

$$m \geq -\frac{\ln\left(0.2\left(b_k - a_k\right)\right)}{\ln 2} \tag{13}$$

Since m must be an integer, the three values that we consider are:

$$\left\{ \left[-\frac{\ln\left(0.2\left(b_k - a_k\right)\right)}{\ln 2}\right] + 1,\ \left[-\frac{\ln\left(0.2\left(b_k - a_k\right)\right)}{\ln 2}\right] + 2,\ \left[-\frac{\ln\left(0.2\left(b_k - a_k\right)\right)}{\ln 2}\right] + 3 \right\} \tag{14}$$

where [ ] is the integer part operator.

The translation set.

For each dilation from the set (14), we keep in the library all the wavelets from family Ω_d whose translations are in $[a_k, b_k]$. This results in the following relation:

$$a_k \le 2^{-m} n \le b_k \tag{15}$$

We are interested in all the values of n obeying the previous condition. They are solutions of:

$$2^m a_k \le n \le 2^m b_k \tag{16}$$

Since n must be an integer, the values that we consider are:

$$\left\{ \left[2^m a_k\right]+1, \left[2^m a_k\right]+2\,, \ldots, \left[2^m b_k\right] \right\} \tag{17}$$

Unlike the dilations, the designer need not choose the number of translations: it depends on the value of m. This number increases exponentially with m.

Note that we use multidimensional wavelets with different dilations for each input variable. Other authors [13] used radial wavelets. Our choice leads to larger libraries with better modeling capabilities. The price to pay is a longer computation time for wavelet selection; however, this duration is but a small fraction of the time required to train the network, so that the overhead introduced by the larger library size is unimportant.

III.2.2. Ranking the wavelets.

After generating the library as described above, the wavelets must be ranked in order of decreasing relevance. This is performed in three steps:

(i) estimate the weights of the direct connections of the network by standard least squares,

(ii) derive a training sequence by subtracting the output of the linear model derived in (i) from the initial training sequence,

(iii) rank the wavelets by the Gram-Schmidt method.

III.2.2.1. Training the direct connections (linear model).

Since we want to rank the wavelets, we are mainly interested here in the nonlinear part of the model; therefore, we first build a linear model. Having taken care of whatever can be explained by a linear model, we can proceed to generate the nonlinear part of the model, for which wavelet ranking is important. Therefore, the wavelets (without the direct connections) are ranked and selected using the residuals of the linear model.

III.2.2.2. The Gram-Schmidt method for ranking wavelets.

The ranking method adopted here is the Gram-Schmidt procedure. For a detailed presentation of this method see for instance [3]. In the following we present its principle. Figure 2 shows graphically a two dimensional example (a two-input model with a training sequence of two examples).

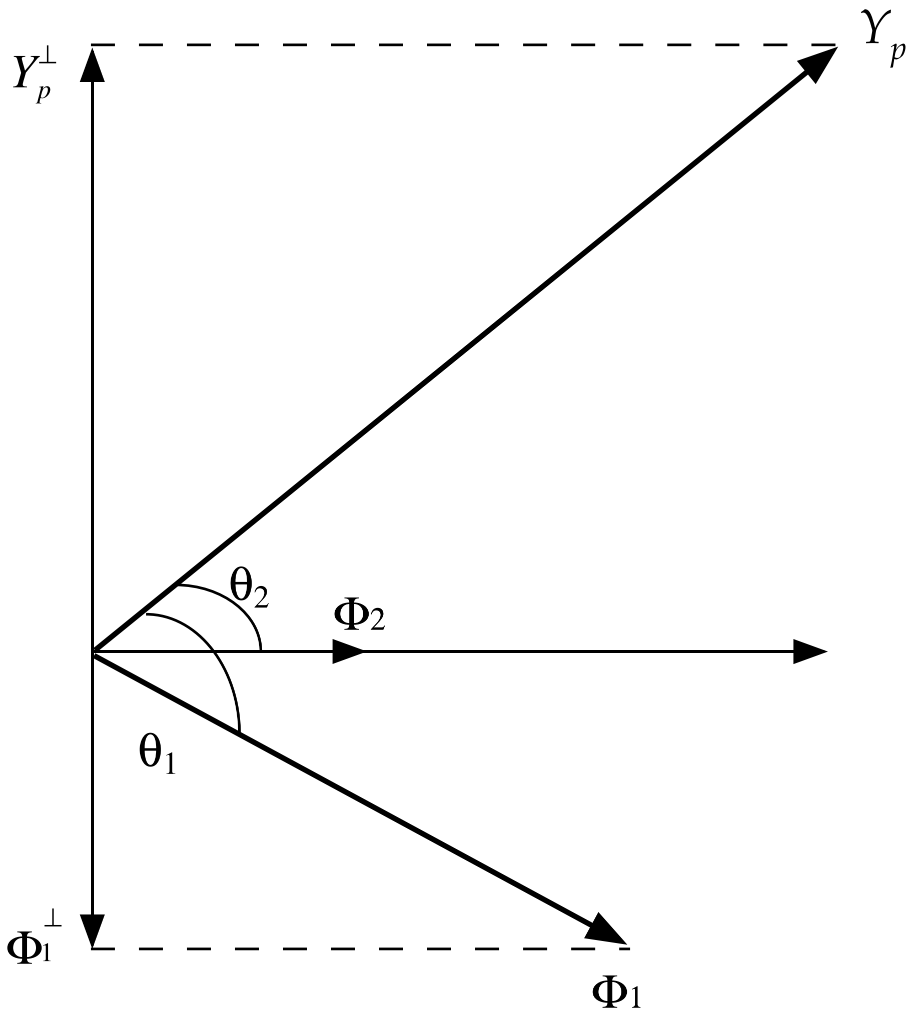


Figure 2
A geometric illustration of the selection by orthogonalization method.

Consider a model, linear with respect to its parameters, with N_i inputs. For a training set of N examples, we define the N-vector $\boldsymbol{Y}$ of the measured outputs, and the N_i input vectors $\boldsymbol{X}^j$, j = 1 to N_i, which are also N-vectors. The inputs are ranked as follows: first, the input vector that has the smallest angle with the vector output is selected. Then all other input vectors, and the output vector, are projected onto the subspace orthogonal to the selected input vector. In this subspace of dimension N-1, the procedure is iterated, and it is terminated when all inputs are ranked.

Since the output of the wavelet network is linear with respect to the weights, the above procedure can easily be used, where each input is actually the output of a multidimensional wavelet. Therefore, at the end of the procedure, all the wavelets of the library are ranked.
As shown on Figure 2, angle θ_2 is smaller than θ_1: therefore, wavelet Φ_2 is selected as the most significant to model the process. The part of the process output not yet modeled and the wavelets not yet selected are subsequently projected onto the subspace orthogonal to the selected regressor. The procedure is repeated until all wavelets are ranked.
In all simulations presented below, we have used the "Modified Gram-Schmidt method", which has been shown to have better numerical stability than the classical one [2].

IV. Numerical experiments.

In the present section, we illustrate the wavelet initialization-by-selection procedure on several examples, and compare its effectiveness to that of the heuristic procedure described in section III.1.
In all the simulations presented below, the following procedure was used for each network:

- the wavelet parameters (dilations and translations) were initialized either by the heuristic procedure (section III.1) or by the selection procedure (section III.2);
- the weights of the connections were initialized to random values, uniformly distributed between -10^{-2} and $+10^{-2}$;
- wavelet parameters and connection weights (including those of the direct connections) were trained simultaneously by the BFGS algorithm [11]; training is terminated when the norm of the gradient of the cost function is too small, or when a maximum number of epochs is reached.

It should be noted that in all the experiments reported here, the connection weights are the only parameters which are initialized randomly; the translations and dilations are initialized in a deterministic fashion, on the basis of the available training data. For each simulation reported below, 100 different trainings were performed, with 100 different weight initializations.

IV.1. Example 1.

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

$$f(x) = \begin{cases} -2.186\,x - 12.864 \quad \text{for} \quad x \in [-10, -2[\\ 4.246\,x \quad \text{for} \quad x \in [-2, 0[\\ 10 \exp\left(-0.05\,x - 0.5\right) \sin\left(x\left(0.03\,x + 0.7\right)\right) \quad \text{for} \quad x \in [0, 10] \end{cases} \qquad (18)$$

This example was first proposed in [13], which is one of the seminal papers on wavelet networks. The graph of this function is shown on Figure 3.

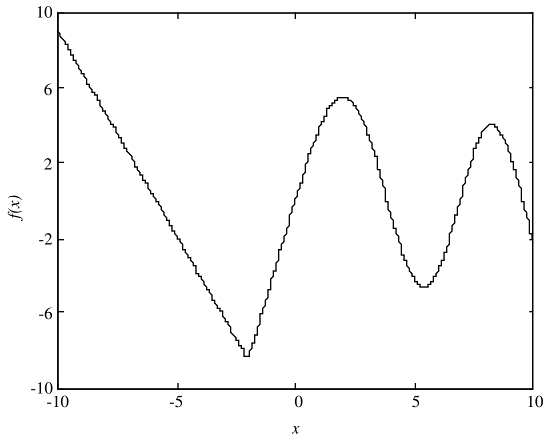


Figure 3
The process output in the domain of interest.

A wavelet network approximation in the domain [–10, 10] is to be found from a training sequence of 300 examples, uniformly distributed in the interval of interest. We define the TMSE (Training Mean Square Error) as $\frac{2J}{N_T}$, where J is the cost function given by relation (6), computed on the training set. N_T is the number of examples in the training set.

The performance of the model is estimated using a test set of 1,000 equally spaced examples.

We define the PMSE (Performance Mean Square Error) as $\frac{2J}{N_P}$ where J is the quadratic cost function computed on the test set. N_P is the number of examples in the test set.

Several architectures were tested. In the following, we present the results obtained with a network of 10 wavelets. Figure 4 shows the TMSE histogram (a) and the PMSE histogram (b) obtained when the 100 trainings are initialized with the heuristic procedure. Figure 5 shows the TMSE histogram (a) and the PMSE histogram (b) obtained when the 100 trainings are initialized with the initialization-by-selection procedure. Comparing figures 4 and 5 shows clearly that the initialization by selection leads to

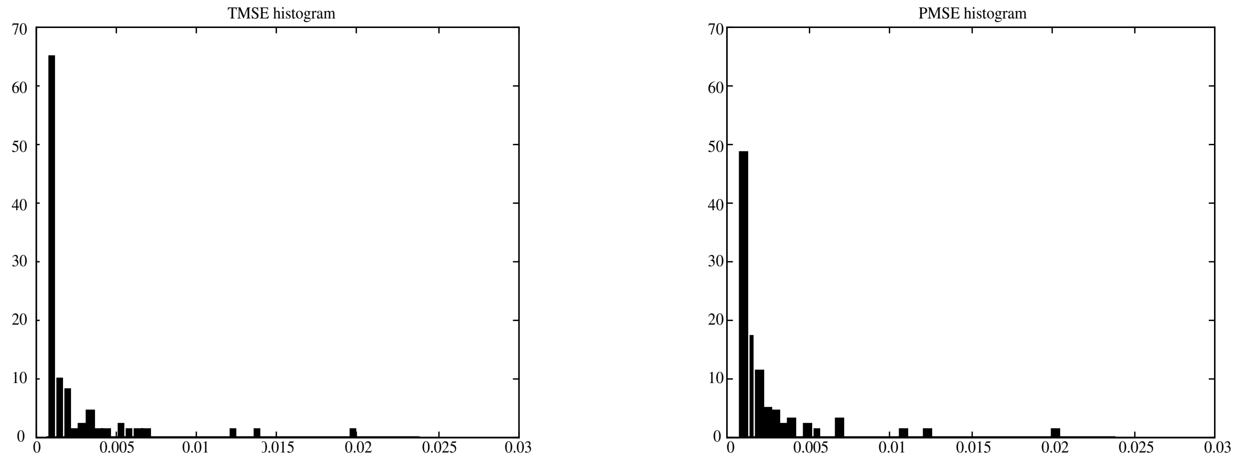
- a more frequent occurrence of the best result,
- less scattered results both on the training set and on the test set.

IV.2. Example 2.

The process to be modeled is simulated by a function of two variables with additive output noise. The expression of this function is given by :

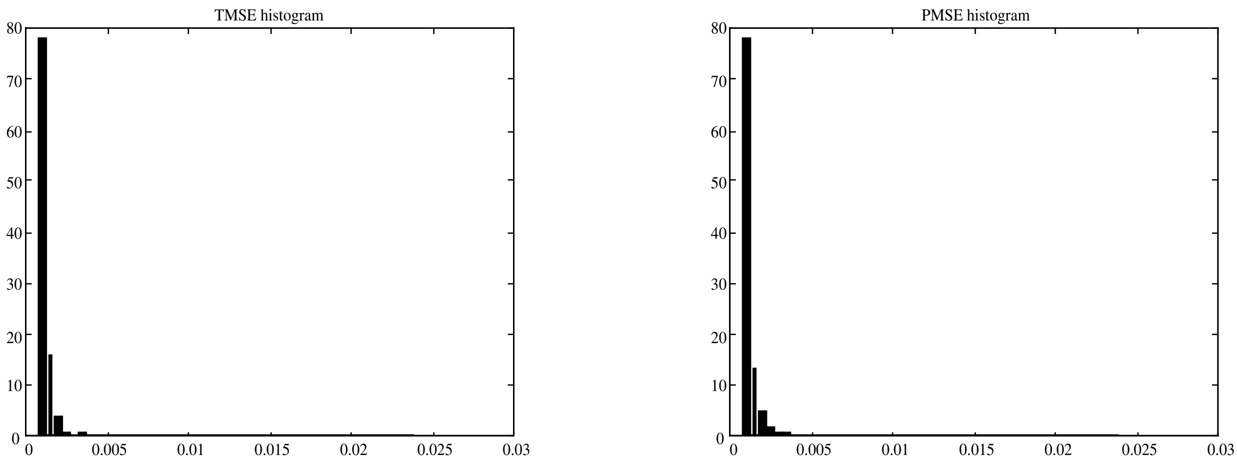
$$f(x_1, x_2)=1.335\left[1.5\left(1-x_1\right)\right]+\exp\left(2\,x_1-1\right)\,\sin\left(3\,\pi\left(x_1-0.6\right)^2\right)+\exp\left(3\left(x_2-0.5\right)\right)\,\sin\left(4\,\pi\left(x_2-0.9\right)^2\right) + w \tag{19}$$

where w is a pseudo-random variable, uniformly distributed with variance 10^{-2}.



(a) (b)

Figure 4
TMSE (a) and PMSE (b) histograms for 100 trainings
performed with the heuristic initialization procedure.



(a) (b)

Figure 5
TMSE (a) and PMSE (b) histograms for 100 trainings
performed with the selection-based initialization procedure.

The domain of interest is defined by the interval [0, 1] x [0, 1]. Figure 6 is a plot of the surface defined by relation (19) without noise ($w = 0$); this example was first proposed in [5]. The training sequence is a set of 1,000 examples, uniformly distributed. The performance sequence is a set of 1,600 examples, equally spaced on a grid. As in the first example, several architectures are tested; for each network, 100 trainings are performed with different random weight initializations. We present the results obtained with a network of 10 wavelets, an architecture which allow us to reach a performance close to the noise variance, without overtraining. Figure 7 shows the TMSE histogram (a) and the PMSE histogram (b) obtained when the 100 trainings are initialized with the heuristic procedure. Figure 8 shows the TMSE histogram (a) and the PMSE histogram (b) obtained when the 100 trainings are initialized with the selection-based initialization procedure. As observed with the first example, initialization by selection leads to the best performance (where the mean-square prediction error is equal to the variance of the noise) more frequently than the heuristic procedure; such a minimum of the cost function is obtained 97 times out of 100.

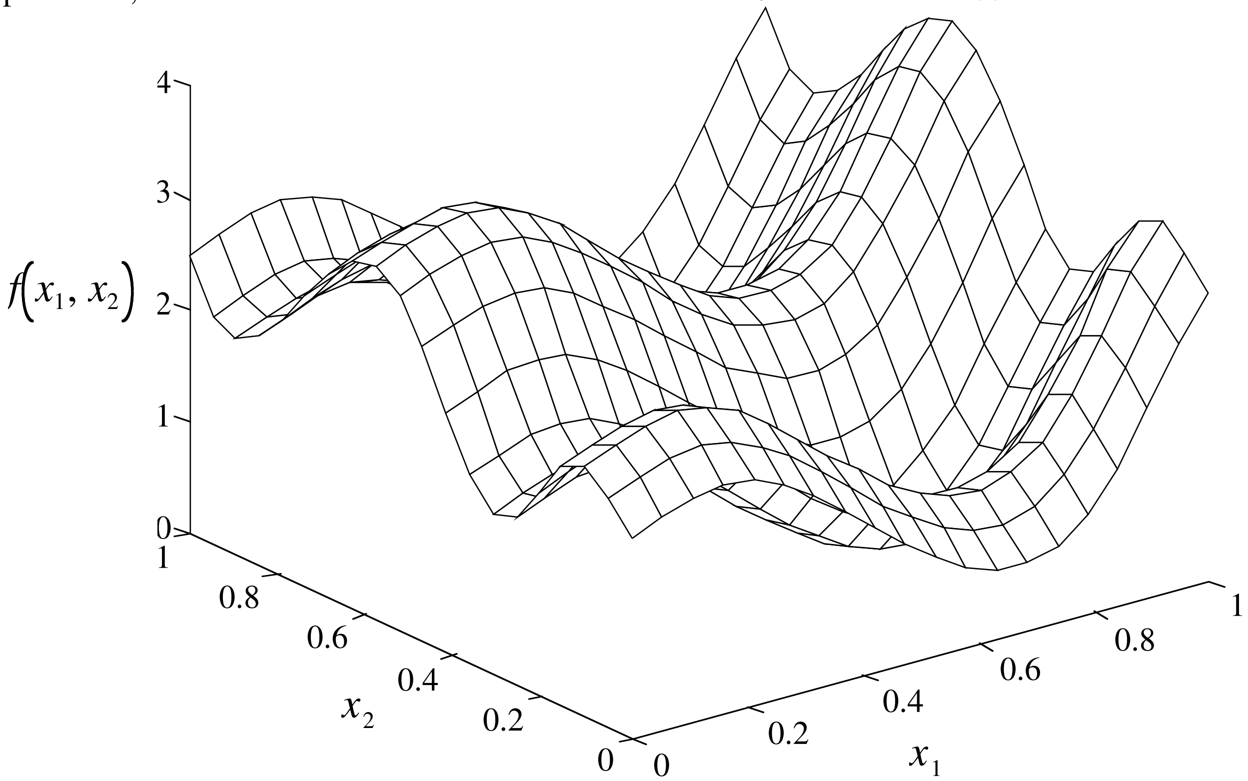
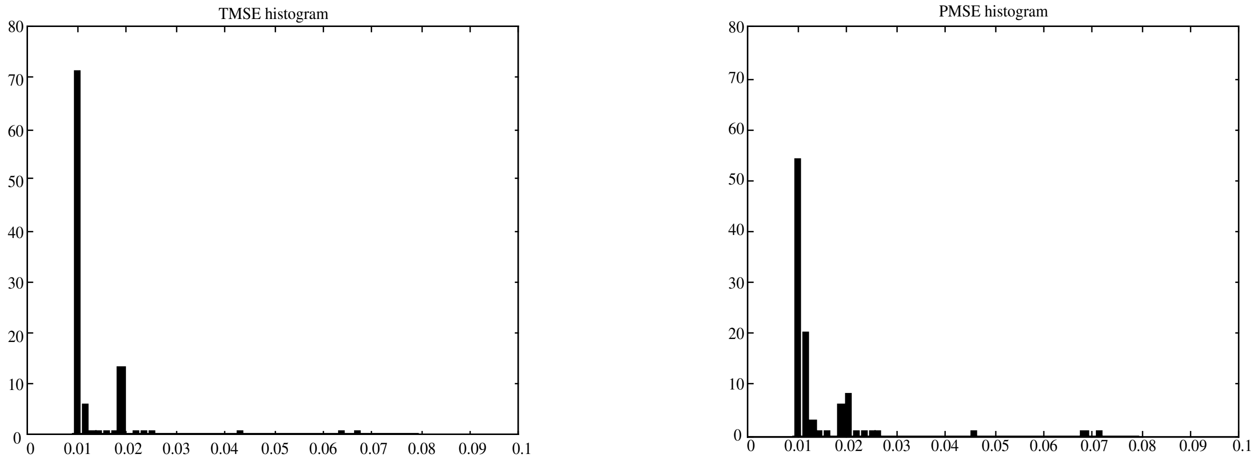


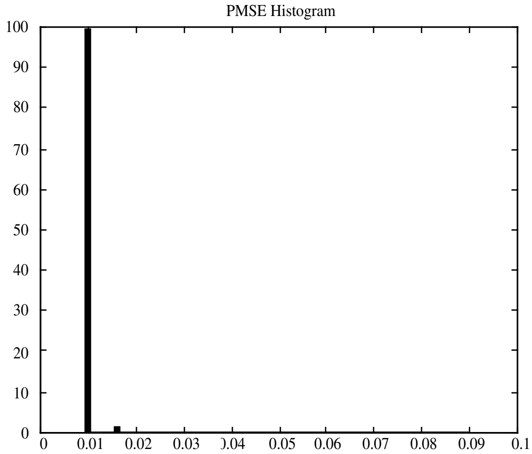
Figure 6



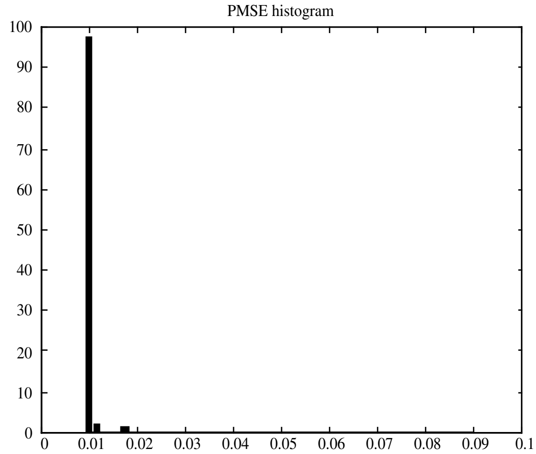
(a) (b)

Figure 7
TMSE (a) and PMSE (b) histograms for 100 trainings
performed with the heuristic initialization procedure.

These results show that the effect of the random initialization of the weights is much smaller when the wavelet centers and dilations are initialized by selection than when they are initialized heuristically; used together with second-order gradient methods, it makes wavelet network training very efficient.



(a)



(b)

Figure 8
TMSE (a) and PMSE (b) histograms for 100 trainings
performed with the selection-based initialization procedure.

V. Conclusion.

Wavelet networks are an alternative to sigmoid neural networks for black-box modeling of processes having a small number of inputs, if both the weights of the connections and the parameters of the wavelets are adjusted from training data. However, due to the local character of the wavelets, the initialization of their translations and dilations requires more care than the initialization of the weights of a conventional neural net. We have proposed an initialization procedure for the centers and dilations, which is more principled than the usual heuristics, based on the properties of discrete wavelets,. We have shown that, when used together with efficient training algorithms, this initialization leads to results that are much more independent from the random initialization of the weights than the results obtained by a heuristic procedure.

LITERATURE REFERENCES

[1] R. Baron, *Contribution à l'Étude des Réseaux d'Ondelettes,* Thèse de Doctorat de l'École Normale Supérieure de Lyon, (1997)

[2] A. Björck, *Solving Linear Least Squares Problems By Gram-Schmidt Orthogonalization,* Nordisk Tidshrift for Informationsbehadlung, 7 (1967) 1-21.

[3] S. Chen, S.A. Billings and W. Luo, *Orthogonal Least Squares Methods and Their Application to Non–linear System Identification,* Int. Journal of Control 50 (5) (1989) 1873-1896.

[4] I. Daubechies, *Ten Lectures on Wavelets,* CBMS-NSF Regional Series in Applied Mathematics, SIAM, Philadelphia, (1992).

[5] J-N. Hwang, S-R. Lay, M. Maechler, R. douglas Martin & J. Schimert, *Regression Modeling in Back-Propgation and Projection Pursuit Learning,* IEEE Transactions on Neural Networks, 5 (3) (1994) 342-353.

[6] A. Juditsky, Q. Zhang, B. Delyon, P. Y. Glorennec and A. Benveniste, *Wavelets in Identification: wavelets, splines, neurons, fuzzies: how good for identification?,* Rapport INRIA No. 2315 (1994).

[7] T. Kugarajah and Q. Zhang, *Mutidimensional Wavelet Frames,* IEEE Trans. on Neural Networks 6 (6) (1995) 1552-1556.

[8] S. Mallat, *A Theory for Multiresolution Signal Decomposition: The Wavelet Transform*, IEEE Trans. Pattern Anal. Machine Intell. 11 (7) (1989) 674-693.

[9] Y. Oussar, I. Rivals, L. Personnaz and G. Dreyfus, *Training Wavelet Networks for Nonlinear Dynamic Input-Output Modeling,* Neurocomputing 20 (1-3) (1998) 173-188.

[10] Y. C. Pati and P. S. Krishnaparasad, *Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transformations,* IEEE Trans. on Neural Networks 4 (1) (1993) 73-85.

[11] W. H. Press, S. A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge (1992).

[12] Q. Zhang and A. Benveniste, *Wavelet Networks*, IEEE Trans. on Neural Networks 3 (6) (1992) 889-898.

[13] Q. Zhang, *Using Wavelet Network in Nonparametric Estimation,* IEEE Trans. on Neural Networks 8 (2) (1997) 227-236.