Synthesis of Denoising Wavelet Neural Networks

Mohamed Zine El Abidine Skhiri, Mohamed Chtourou

Research Unit on Intelligent Control, Design & Optimization of Complex Systems (ICOS)

Ecole Nationale d'Ingénieurs de Sfax (ENIS), BP.W,3038, Sfax, Tunisia

email:Zine.skhiri@isetso.rnu.tn and Mohamed.chtourou@enis.rnu.tn

Abstract: This paper investigates the use of a wavelet denoising unit based on the wavelet mutiresolution analysis on wavelet networks instead of neural networks on which previously reported works have been performed. This full wavelet compound will certainly provide the possibility of building up two denoising analysis models. The most straight forward model is setup by placing the wavelet denoising unit ahead of the network input layer. In other words, the inputs data embedded in a white Gaussian noise of a wavelet network are firstly denoised then fed to the network. An alternative model is also investigated in which the denoising unit will be placed at the output level of the net since a wavelet network is may be considered as a first step smoothing unit. In this analysis version, the noisy signal is fed to the wavelet network and the corresponding output is then applied to the denoising unit.

Keywords: wavelet neural networks, wavelet denoising unit, soft thresholding, hard thresholding,

Skhiri Mohamed Zine El Abidine: received the B.S and M.S. degrees in Electrical Engineering, from Syracuse University, Syracuse New York, USA. Currently, he is a teaching assistant in the Department of Electrical Engineering at the Institute of Technology in Sousse, Tunisia. He is registered as a PhD student on the National School of Engineers of Sfax, Tunisia, and his current research includes wavelet network systems.

Chtourou Mohamed: received the B.S, M.S., and Ph.D. degrees in electrical engineering, from National School of Engineers of Sfax-Tunisia, Institut National des Sciences Appliquées de Toulouse-France and Institut National Polytechnique de Toulouse-France, respectively. He is currently a professor in the Department of Electrical Engineering of National School of Engineers of Sfax-Tunisia. His current research interests include neural and fuzzy systems, intelligent and adaptive control. He is author and co-author of more than ten papers in international journals.

1. Introduction

Since their appearance wavelets have shown the ability to solve different kind of problems in different domains including numerical analysis, signal processing and mathematical modelling. Looking back to the 80's, Morlet and Grossmann had the idea to use a dilated and a translated function $\psi(t)$ called a wavelet which is well localized in time and frequency [14]. Meyer, Mallat and others had the possibility to seriously contribute to this early work until the paper of Ingrid Daubechies in 1988 that brought the mathematical concept to signal processing, statistics and numerical analysis. However, the most powerful and advanced feature of wavelets is the multiresolution analysis, by which a signal can be decomposed into different time-scale approximations and details. In other words, a signal $f(t)$ is broken down into two different types of signals. In one hand, signals that carry the approximations of the original signal, and on the other, signals that carry the details. It is likely to be considered as the "Mathematical Microscope"[1]. This characteristic will certainly help reducing any presented noise since the signal is totally decomposed to the finest detail. Indeed, wavelet thresholding techniques have proved to be the most powerful denoising tool since the work of Donoho and Jhonstone [5-8]. Therefore, it should be absolutely useful in every domain that deals with noisy data including neural and wavelet networks.

During the last decades neural networks (NN) have not given up showing their ability and success in solving system processing problems, approximating functions etc. On the other hand, the appearance of the wavelet theory has made it even more successful and powerful by creating analogous networks called the wavelet neural networks in which wavelets are used as the activation functions of the hidden neurons in the conventional neural network. Wavelet networks are traced back to the work of Daugman [3] in which Gabor wavelets were used for image compression. They are introduced as a special feedforward neural network, and they became more popular after the work of Pati [15], Zhang [17], and Szu[16]. They have been applied to many different areas and applications such as nonlinear functional approximation and nonparametric estimation [20], system identification and control tasks, modelling and classification. The generated wavelets used in networks are the dilated and the translated versions of a mother wavelet which could be in a continuous or a discrete form.

In this paper, the most wavelet advantages are exploited including the wavelet networks as well as the wavelet multiresolution analysis for denoising. Basically, two possible denoising process models are introduced. In the first model, the inputs data of any static or dynamic wavelet network are firstly denoised based on the wavelet multiresolution analysis. Then, they are fed to a wavelet network having as activation functions some dilated and translated versions of a certain mother wavelet. In the second model, however, the denoising unit is added at the network output level. In other words, the network takes the noisy signal as the input and the corresponding output is then denoised. As far as the adjusted wavelet network parameters are concerned, it should be noticed that their initialization process is somehow delicate, and their number is important. Unlike neural networks in which the weights could be initialized randomly, wavelet networks require a greater care in choosing the initial values of the parameters, especially, the dilations and the translations.

This paper is organized as follows: In section II, the related works are presented followed by the proposed denoising process models in sectionIII. Afterwards, the multiresolution analysis and the wavelet networks are introduced in section IV and V respectively. The simulation results, however, are discussed in section VI. Finally, the last section will conclude the paper.

2. Related Works

Denoising neural networks data has been a serious treated issue and a lot of interesting works have been carried out in various domains using different approaches. For example, in [22] a thresholding neural network TNN has been developed for adaptive noise reduction. It is based on using an infinitely differentiable soft and hard thresholding functions as activation functions. Then, the network is subsequently trained by gradient based techniques. Noisy images and different commonly used signals such as Blocks, Doppler, Bumps etc. were used to illustrate the application of the TTN. Noisy images have also been treated differently in [21]. The treated noisy image is first wavelet transformed into four subbands, then, a trained layered neural network is applied to each subband to generate denoised wavelet coefficients. The denoised image is thereafter obtained by applying the inverse transform. Within the same framework, a set of successive works has been performed in [10-12] to reduce the influence of noise in time series prediction. Throughout this whole work, the denoising process has experienced various phases. The starting phase consists of placing a denoising unit based on the wavelet multiresolution analysis ahead of a neural network input layer [10]. In other words, the data of the network input layer is denoised before getting processed. Successively, this idea has been developed up to integrating two special denoising layers based on the wavelet multiresolution analysis into the layered network [12].

Briefly, it is obvious that the main previous work has dealt with the neural networks platform and with the use of the classic denoising approach which is based on denoising the data before getting processed. In the contrary, this presented work has totally converged the data processing towards the wavelet networks platform instead, and has brought an additional dual denoising approach. In other words, neural networks are replaced by wavelet networks and the wavelet denoising unit is added at the output level of the used network and so the data is processed before getting denoised. In fact, this approach may be considered as a double denoising process since the wavelet network processing phase could be considered as a smoothing unit.

3. Proposed Denoising Process Models

As mentioned earlier, the wavelet network denoising process is ensured by combining wavelet networks and wavelet denoising units based on the wavelet multiresolution analysis. The use of these two basic components provides two possible denoising process models which will be investigated in this paper. In the first model, the denoising unit is placed ahead of the network input layer as shown in Figure 1.a. In other words, the inputs data are firstly denoised then fed to the wavelet network. As far as the denoising unit is concerned, it is also based on the wavelet multiresolution analysis which decomposes the data into different time-scale approximations and details coefficients. However, to discriminate the noise from the useful information, a certain threshold is applied to the generated details coefficients. The denoised data, which will be input to the network, is then reconstructed using an inverse transform. An alternative configuration is also possible to build up a second model in which the denoising unit will be added at the output level instead, as shown in Figure 1.b. In other words, the inputs data are directly fed to the wavelet network and the corresponding output is applied to the denoising unit discussed lately. For the sake of simplicity and clarity, both models are represented using a one dimensional wavelet network.

4. Denoising Based on Multiresolution Analysis

4.1 Multiresolution analysis

Since the denoising process is based on wavelets, the noisy signal could be decomposed in order to discriminate between the significant and the contaminated information. To do so, it is necessary to use the most advanced feature of wavelets which is the multiresolution analysis where a signal can be decomposed into different time-scale approximations and details. This analysis is carried out by means of two types of functions, wavelet functions as well as scaling functions.

4.1.1 Mathematical analysis

Mathematically speaking, the multiresolution analysis is based on a sequence of embedded subspaces V_j Î ¢ of L^2 (space of square integrable functions) [13].

K Ì V_{-1} Ì V_0 Ì V_1 Ì V_2 Ì K Ì L^2 (1)

For a function $f(x)$ Î V_j , there exits a set of scaling functions: $f_k(x) = f(x-\ k)$ k Î ¢ , with $f(x)$ Î L^2 so that $f(x)$ could be expressed as follows [13]:

$$f(x) = \sum_{k \in Z} a_k \phi(x-k) \tag{2}$$

For instance, if $\phi(x)$ is compressed and translated with a small step size more details could be detected. In the contrary, if $\phi(x)$ is stretched and translated with a wider steps coarse information are detected. Hence, it is clear that the extraction of various information is tightly related to the size of the subspace and the spanning set of functions. Basically, these functions have the following forms:

$$\phi_{j,k}(x) = 2^{\frac{j}{2}} \phi(2^j x - k) \tag{3}$$

Where $\phi(x)$ is called the scaling function and j is the scaling index.

Furthermore, a signal can be better described not only by using the scaling functions $\phi_{j,k}(x)$, but also by introducing another set of functions called wavelets $\psi_{j,k}(x)$ orthogonal to $\phi_{j,k}(x)$. Similarly, these wavelets are obtained by scaling and translating the mother wavelet $\psi(x)$:

$$\psi_{j,k}(x) = 2^{\frac{j}{2}} \psi(2^j x - k) \tag{4}$$

They are introduced throughout the subspace W_j which is the orthogonal complement of V_j in V_{j+1}. This relation is stated by the following equation:

$$V_{j+1} = V_j \,\text{Å}\, W_j \tag{5}$$

Where the operator ⊕ stands for the direct sum. In other words, each element of V_{j+1} can be written, in a unique way, as the sum of an element of W_j and an element of V_j. However, it should be mentioned that the scale of the initial space is arbitrary and the value of *j* could have a positive value as well as a negative one. For instance, if the scale of the scaling space starts at 0, then the nested subspaces $W_{j\text{Î ¢}}$ are generated such that each W_j represents the refinement of W_0, and their direct sum is L^2.

$$L^2 = V_0 \,\text{Å}\, W_0 \,\text{Å}\, W_1 \,\text{Å}\, W_2 \,\text{K} \tag{6}$$

Then, any function $f(x)\,\text{Î}\;L^2$ could be expressed as follows:

$$f(x) = \underset{k\text{Î ¢}}{\text{å}}\, a_{j_0,k} f_{j_0,k}(x) + \underset{j^{\text{³}}\, j_0}{\text{å}}\; \underset{k\text{Î ¢}}{\text{å}}\, d_{j,k} y_{j,k}(x) \tag{7}$$

Generally, the value j_0 is fixed so that the coarsest information of the signal is obtained. The coefficients in this expansion are called the discrete wavelet transform.

4.1.2 Signal processing analysis

From a signal analysis point of view, the multiresolution analysis is a hierarchical decomposition of a signal into successive approximations that are themselves decomposed in turn as illustrated in Figure 2. In other words, a signal is broken down into two different types of signals. One type represents the approximations, and the other represents the details. It is clear though that for each level *j*, a *j*-level approximation A_j is generated as well as a *j*-level detail D_j. This decomposition is represented in Figure 2 for an arbitrary signal *f*. On the other hand, *f* could be reconstructed in terms of the generated approximation and details as given by the following expression:

$$f = A_3 + D_3 + D_2 + D_1 \tag{8}$$

To have a closer look at the decomposition process, figure 3 illustrates a three level decomposition of a noisy version of the signal $f_1(x)$ given in (27).

4.2 Wavelet based denoising

Noise reduction has been and certainly will be the most treated issue on signals based applications since the significant information on the used signal could not be avoided from being contaminated by noise. Indeed, different techniques and algorithms have been applied to carry out this task in the time domain as well as in the transform domain. Unlike Fourier based techniques which is a spectrum localization dependent, wavelet based techniques provide both time and frequency localization [2] features presented as coefficients on which the denoising process is applied. As far as the transform domain is concerned, the recent introduction of wavelet transform is considered to be the most denoising powerful tool. Certainly, this is due to the multiresolution analysis that decomposes the signal up to the finest detail. Virtually, it is considered as a microscope that focuses on the noise spread throughout the different levels coefficients. Once the noisy signal is broken down, a thresholding of the generated wavelet coefficients is applied in order to discriminate the signal from the noise. The main idea is based on eliminating the wavelet coefficients of the noisy signal less then a certain threshold.

4.2.1 Soft and hard thresholding

Basically, the performance of the thesholding process is tightly related to the choice of the thesholding function in the first place then to the setting of the threshold value. The commonly used functions are the soft and the hard thresholding functions. Donoho and Johnstone [6], [7] have proposed the soft thresholding function *η(x)* defined by the expression given in (9) and represented in figure4. This function is also called the shrinking function. The name shrinking comes from the fact that all the wavelet coefficients are reduced.

$$\eta(x) = \begin{cases} sign(x).(|x-\tau|) & if\ |x| \succ \tau \\ 0 & if\ |x| \le \tau \end{cases} \tag{9}$$

Also, they proposed the optimal threshold τ called the "universal threshold" such that [7]:

$$t = s\sqrt{2Log(N)} \tag{10}$$

Where s denotes the estimated standard deviation of the noise and N denotes the number of the coefficients in the detail signal $d_{j,k}$. As an estimation of σ they suggested the median value of the wavelet coefficients $d_{j,k}$ [6]:

$$s = \frac{median(d_{j,k})}{0.6745} \tag{11}$$

Level dependent thresholding: Since the noisy signal could be decomposed into different levels, it is possible to assign a threshold parameter to each level. In fact, Jhonstone and Silverman [9] proposed the use of level-dependent thresholds for extracting signals from correlated noise. The corresponding expression for a given level *j* is defined by the following equation:

$$\tau_j = \sigma_j\sqrt{2Log(N_j)} \tag{12}$$

As for the hard (keep or kill) thresholding function, the corresponding expression is defined below in (13) and represented in figure 5. Compared to soft thresholding, hard thresholding conserves the coefficients and generates discontinuities at the locations $x = \pm\tau$. Whereas, soft thresholding avoids this phenomena, shrinks out the coefficients and produces some signal attenuation. However, it should be noticed that the use of one function or the other depends really on the considered application.

$$\eta(x) = \begin{cases} x & if\ |x| \succ \tau \\ 0 & if\ |x| \le \tau \end{cases} \tag{13}$$

4.2.2 Wavelet thresholding

As mentioned earlier, nonlinear thresholding was first proposed by Donoho and Johnstone [5], [8] to denoise signals embedded in white Gaussian noise. Basically, the denoising process can be summarized up to the following three major steps [5]:

For instance, let $y(n)$ be the discrete noisy signal, $x(n)$ the noise-free signal and $\omega(n)$ the additive white Gaussian noise with a zero mean and a variance σ^2, such that:

$$y(n) = x(n) + \omega(n) \tag{14}$$

with $n = 1, 2, \dots, N$.

Step 1: Compute the wavelet coefficient $\tilde{y}(n)$ of the noisy signal $y(n)$ using the Fast Wavelet Transform FWT. The resulting sub-signals will be composed of the approximation coefficients at the coarsest level and the details coefficients.

Step 2: Threshold the details wavelet coefficients on each level using the soft thresholding function defined in (9).

Step 3: Perform the inverse transform of the thresholded signal which include the approximation coefficients and the thresholded details.

5. Wavelet Networks

5.1 Wavelet network structures

Based on the wavelet theory, Zhang and Benveniste have proposed the wavelet network as an alternative to the feedforward neural networks for approximating arbitrary nonlinear functions [17]. These types of networks are similar to the one hidden layer neural networks in which the coefficients can be considered as the connection weights and the wavelets as the activation functions (wavelons). The generated wavelets used in networks are the dilated and translated versions of a mother wavelet $\psi(x)$ which could be in a continuous or a discrete form. In general, the continuous form of $\psi(x)$ is:

$$\psi_{a,b}(x)=\frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right)\quad a\in\mathbb{R}^{*}\quad and\quad b\in\mathbb{R} \tag{15}$$

Where a and b represent the dilation and the translation respectively. A discretized form may also be generated if $a=2^{-j}$ and $b=k2^{-j}$ with $j,k\in\mathbb{Z}$.

So, the corresponding expression will be as follows [2]:

$$\psi_{j,k}(x)=2^{\frac{j}{2}}\psi(2^{j}x-k)\quad j,k\in\mathbb{Z} \tag{16}$$

Compared to neural networks, wavelet networks use an important number of parameters that may be updated. They include weights, translations, dilations and direct terms coefficients. Certainly, this will provide more flexibility to derive different network structures. The most simple one- dimensional input structure is represented by the following expression:

$$y(x)=\sum_{j=1}^{N_w} w_j\psi\left(\frac{x-t_j}{d_j}\right) \tag{17}$$

Wavelet networks are inspired by the one dimensional discrete inverse wavelet transform given by the expression below [19].

$$y(x)=\sum_{j\in\mathbb{Z}^{+},k\in\mathbb{Z}} w(j,k)\alpha^{-\frac{j}{2}}\psi(\alpha^{-j}x-\beta k) \tag{18}$$

Where the $w(j,k)$ is the discrete wavelet transform of $y(x)$, and α and β are the dilation and the translation step sizes respectively.

A more general structure is stated in [18] and represented in Figure 6 that takes into account the multidimensional continuous wavelets as well as the linear property to deal with non linear approximation problems. In this expression direct terms are added:

$$y(x) = \sum_{j=1}^{N_W} w_j \Psi_j(x) \;+\; \sum_{k=1}^{N_i} a_k x_k \quad + \quad a_0 \tag{19}$$

Where: a$_0$ is a scalar quantity added to deal with the nonzero mean functions. N_w is the number of wavelets, N_i is the number of the input variables, $x = (x_1, x_2 \mathrm{K}\; x_{Ni})^T$ is the inputs vector and $\Psi_j(x)$ is the following product:

$$\Psi_j(x) = \prod_{k=1}^{N_i} \psi\left(\frac{x_k - t_{jk}}{d_{jk}}\right)$$

In this case, the multidimensional wavelets are formed by the product of N_i mono-dimensional wavelets. $\psi\left(\frac{x_k - t_{jk}}{d_{jk}}\right)$ is a dilated and translated version of the mother wavelet ψ.

5.2 Parameters initialization and learning algorithms

5.2.1 Learning algorithm

To adjust the parameters of the above networks different types of algorithms could be used. Namely, the gradient-based techniques and the second order methods etc. In this paper, a stochastic gradient algorithm is carried out to approximate a certain function $f_d(x)$. In other words, the parameters are adjusted after each input/output observation. The training is performed by recursively minimizing the criterion (20) using a gradient based technique.

$$J(\theta) = \frac{1}{2}\left[y_n - f_{d_n}\right]^2 \tag{20}$$

Where y_n is the output of the network and f_{d_n} is the desired output for the example *n* (y_n and f_{d_n} are scalar quantities). The parameter vector θ includes: the scalar term a$_0$, the weights w_j, the dilations $d_{j,k}$ and the translations $t_{j,k}$. Therefore, the gradient expression is may be given by (21) as follows:

$$\frac{\P J}{\P q} = \frac{\P y_n}{\P q}(y_n - f_{d_n}) = \frac{\P y_n}{\P q} e_n \tag{21}$$

Where $e_n = (y_n - f_{d_n})$ represents the error between the output of the network and the desired output for the example *n*.

With also: $$\frac{\P y_n}{\P q} = \left.\frac{\P y}{\P q}\right|_{x = x_n}$$

5.2.2 Gradient calculation

At this level, it would be very helpful to have an idea about the calculation procedure of the gradient with respect to each adjusted network parameter. This task is carried out on the wavelet network structure that does not include the direct terms. The corresponding expression of $y(x)$

is given in (22). It should be noticed that this is the same structure that has been used for the simulated results:

$$y(x) = \left[\sum_{j=1}^{N_w} w_j \Psi_j(x)\right] + a_0 \qquad (22)$$

Where : $\Psi_j(x) = \prod_{k=1}^{N_i} \psi\left(\frac{x_k - t_{jk}}{d_{jk}}\right)$ and a$_0$ is a scalar term.

So, for $y(x)$ given in (22) and for a given example n, the corresponding gradient expressions with respect to the scalar term a$_0$, the weights w_j, the dilations d_{jk} and the translations t_{jk} are given below respectively:

$$* \ \frac{\partial J}{\partial a_0} = e_n \qquad (23)$$

$$* \ \frac{\partial J}{\partial w_j} = e_n \Psi_j(x_n) \qquad (24)$$

Where: $\Psi_j(x_n) = \prod_{k=1}^{N_i} \psi\left(\frac{x_{k_n} - t_{jk}}{d_{jk}}\right)$

$$* \ \frac{\partial J}{\partial d_{jk}} = e_n w_j \left[\prod_{L \neq k} \psi\left(\frac{x_{L_n} - t_{jL}}{d_{jL}}\right)\right] \psi'\left(\frac{x_{k_n} - t_{jk}}{d_{jk}}\right)\left(-\frac{\left(x_{k_n} - t_{jk}\right)}{\left(d_{jk}\right)^2}\right) \qquad (25)$$

Where: ψ' is the first derivative of the function ψ with respect to its argument.

$$* \ \frac{\partial J}{\partial t_{jk}} = e_n w_j \left[\prod_{L \neq k} \psi\left(\frac{x_{L_n} - t_{jL}}{d_{jL}}\right)\right] \psi'\left(\frac{x_{k_n} - t_{jk}}{d_{jk}}\right)\left(-\frac{1}{d_{jk}}\right) \qquad (26)$$

5.2.3 Parameters Initialization

In neural networks, the updated parameters may be initialized in a random manner having some values capable to produce neuron outputs that lay in the linear part of the sigmoid. Unlike neural networks, wavelet networks require a great care in choosing the initial parameters especially the dilations and the translations. In fact, these two parameters depend on the input domain in which the examples are distributed since wavelets are characterized by a very fast decay. For example, if the translations are initialized outside the examples domain or the dilations are set to small values, then, the wavelet output will be approximately zero and so its derivative. As a result, the training algorithm will not hold since it is a gradient dependant.

6. Simulation Results

At this level, the denoising unit along with the wavelet network will be tested in the functions approximation and the image processing domains. The approximated functions *$f_1(x)$, $f(x_1,x_2)$* are a mono and a two variable functions [17] given in (27) and (28) respectively. They are presented as a sequence of 200 examples uniformly distributed over the interval [-10,10], and are shown in figure 7 and figure10 respectively. As for the image, it is a TIFF 256x256 image named "cameraman.tif", shown in figure 14, and it is extracted from the Mathlab image processing toolbox. To get processed, this image is fed to the network as a one dimensional vector built up by concatenating the rows of its corresponding matrix. The simulated results will be generated out of two alternative configurations depending on the position of the denoising unit with respect to the network. In the first configuration, the denoising unit will be placed at the network input level. Whereas, in the second one, the denoised unit will be placed at the output level instead. The used wavelet network activation functions, which may be called wavelons, are some dilated and translated versions of a mother wavelet called the Gaussian derivative having the following expressions:

$\psi(x) = -xe^{-\frac{x^2}{2}}$ and $\psi(x_1, x_2) = x_1 x_2 e^{-\frac{(x_1^2 + x_2^2)}{2}}$ in the one and the two dimensional cases respectively.

$$f_1(x) = \begin{cases} -2.186x - 12.864 & \text{if } -10 \le x < -2 \\ 4.246x & \text{if } -2 \le x < 0 \\ 10e^{-0.05x - 0.5} \sin[(0.03x + 0.7)x] & \text{if } \ 0 \le x \le -10 \end{cases} \qquad (27)$$

$$f(x_1, x_2) = (x_1^2 - x_2^2)\sin(0.5x_1) \qquad (28)$$

At this stage, it should be mentioned that the wavelet used by the denoising unit, for all the treated cases, is the symlet 4. While, the number of the decomposition levels denoted by L is set to L= 4 for the functions $f_1(x)$ and $f(x_1, x_2)$, and L= 2 for the image.

● Denoising unit at the input level

In this case, the image and the two functions will be approximated with a denoising process carried out at the input level. The general expression of the noisy functions is as follows: $f_n(x) = f(x) + w(x)$. Where $f(x)$ is the free noise function and $w(x)$ is a zero mean white Gaussian noise with a 10dB SNR (signal to noise ration). The noisy versions of $f_1(x)$ and $f(x_1, x_2)$, are shown in figure 7 and figure 11 respectively. Similarly, the noisy image is expressed as follows: $\Gamma_n = \Gamma + \Gamma_w$, and is represented in figure 15. Where Γ is the free noise image shown in figure 14, and Γ_w is a zero mean white Gaussian with a 10dB SNR. As far as the used wavelet networks are concerned, the corresponding outputs representing $f_1(x)$, $f(x_1, x_2)$ and the processed image are depicted in figure8, figure12 and figure 16 respectively. In addition, the corresponding training parameters are listed in table1.

● Denoising unit at the output level

In this alternative configuration, the previous functions and the image are processed having the denoising unit placed at the output level of the network. In other words, the noisy data is fed to the wavelet network and the corresponding output is then de-noised based on the wavelet multiresolution analysis. In this case, the wavelet network outputs representing $f_1(x), f(x_1, x_2)$ and the processed image are depicted in figure 9, figure 13 and figure 17 respectively. It should be noticed that the same training parameters are used, and they are listed in table1.

Based on the simulated results and their corresponding mean square error values shown in table 1, the best performance is provided by the model with the denoising unit added at the input level. Nevertheless, the model with the denoising unit added at the output level has also generated very satisfactory overall results. Furthermore, it has shown that a wavelet network with a noisy input could be considered as a predenoising unit since it undergoes throughout the training process some kind of signal smoothing at the output.

7. Conclusion

Noise reduction has been and certainly will be the most treated issue on signals based applications including wavelet networks data. However, the most powerful tool to carry out this task turns out to be the denoising based on wavelets. Therefore, it would very convenient to process the noisy inputs or outputs data of any static or dynamic wavelet networks by a wavelet based denoising unit. The position of this type of unit with respect to the wavelet network provides the possibility to generate two different denoising models. One model includes the denoising unit at the network input level, and the second one at the output level. It is clear though that the model with the denoising unit added at the input level has generated more satisfactory results than the second model. However, when the denoising unit is placed at the output level, the whole denoising process is may be considered to be carried out twice. The first is totally related to the wavelet network which undergoes throughout the training process some kind of signal smoothing at the output. In fact, the performance of this process is quite interesting since it preserves at least the main shape of the free noise data. In other words, the wavelet network could be considered as a first step denoising unit. The second process, however, is carried out on the generated network output using the based wavelet denoising unit. Placed whether at the input or at the output of the network, the performance of the denoising unit depends on the type of the used wavelet and the number of the decomposition levels. As far as the wavelet networks are concerned different networks structures could be processed with a special care that should be taken towards the network parameters initialization phase. Unlike neural networks, wavelet networks require a great care in choosing the adjusted parameters especially the dilations and the translations.

REFERENCES

1. BURKE, B., **The mathematical Microscope: waves, wavelets and beyond**, A Positron Named Priscilla, Scientific Discovery at the Frontier, chapter 7, 1994, pp 196-235.

2. DAUBECHIE, I, **The wavelet transform, time-frequency localization and signal analysis**, IEEE Trans. Informat. Theory Vol. 36, 1990.

3. DAUGMAN, J. G., **Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression**, IEEE Trans.Acoust Speech, Signal Process vol. 36, no. 7, 1988, pp. 1169-1179.

4. DONOHO, D.L., **Nonlinear wavelet methods for recovering signals, images, and densities from indirect and noisy data**, Proceedings of symposia in applied Mathematics, vol. 47, 1993, pp. 173-205.

5. DONOHO, D.L., **Denoising by soft-thresholding**, IEEE Trans. Inform. Theory, vol.41, no.3, 1995, pp. 613-627.

6. DONOHO, D.L. and I.M. JOHNSTONE, **Ideal spatial adaptation via wavelet shrinkage**, Biometrika, vol. 81, no. 3, pp. 425-455.

7. DONOHO, D.L. and I.M. JOHNSTONE, **Adapting to unknown smoothness via wavelet shrinkage**, J. Amer: statist. Assoc, vol. 90, no. 432, 1995, pp. 1200-1224.

8. DONOHO, D.L., I.M. JOHNSTONE, KERKYACHARIAN PICARD D., **Wavelet shrinkage: Asymptotia?**, J. Royal statist. Soc., vol. 51, no. 2, 1995, pp. 301-337.

9. JHONSTONE, M., B.W. SILVERMAN, **Wavelet threshold estimators for data with correlated noises**, Journal of the Royal Statistical Society, Vol. 59, 1997, pp. 319 351.

10. LOTRIC, U., DOBNIKAR A., **Wavelet based smoothing in time series prediction with neural networks**. In: V. Kurkova, N. C. Steele, R. Neruda, M. Karny (eds.): Artificial Neural Nets and Genetic Algorithms: Proceedings of the ICANNGA Conference in Prague, Czech Republic, Springer, 2001, pp. 43-46.

11. LOTRIC, U., **Wavelet Based Denoising Integrated into Multilayered Perceptron**, Neurocomputing, vol. 62, 2004, pp. 179-196.

12. LOTRIC, U., A. DOBNIKAR, **Neural Networks with Wavelet Based Denoising Layer for Time Series Prediction**, Neural Computing and Applications, vol. 14, no. 1, 2005, pp.11-17.

13. MALLAT, S., **A theory of multiresolution signal decomposition: the wavelet representation**, IEEE Trans on pattern recognition and Machine Intelligence, vol. 11, 1969, pp. 674-693.

14. MEYER, Y., **Wavelets: algorithms and applications**, Philadelphia, SIAM, 1993.

15. PATI, Y.C. and P.S. KRISHNAPARASAD, **Analysis and synthesis of feedforward neural networks using discrete affine Wavelet transforms**, IEEE Trans. Neural Networks, vol. 4, no. 1, pp. 73-85.

16. SZU, H., B. TELFER, and S. KADAMBE, **Neural network adaptive wavelets for signal representation and classification**, Optical Engineering, 31:1907-1961, 1992.

17. ZHANG, Q.H. and A. BENVENISTE, **Wavelet networks**, IEEE trans. Neural Networks, vol. 3, no. 6, 1992, pp. 889-898.

18. ZHANG, Q.H., **Wavelet networks: the radial structure and an efficient initialization procedure**, Technical Report of Linköping University, LiTH-ISY-I-1423. 1992.

19. ZHANG, Q., **Regressor Selection and Wavelet Networks Construction**, Technical Report of Linköping University, no. 1967. 1993.

20. ZHANG, Q., **Using Wavelet Network in Nonparametric Estimation**, IEEE Trans On Neural Networks, vol. 8, no. 2, 1997.

21. ZHANG, S. and E. SALARI, **Image denoising using a neural network based non-linear filter in wavelet domain**, Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP) IEEE International Conference, Vol. 2, 2005, pp. 989 – 992.

22. ZHANG, XIAO-PING, **Thresholding Neural Network for adaptive noise reduction**, IEEE Trans. Neural Networks, vol. 12, no. 3, 2001.