MULTIDIMENSIONAL POLYNOMIAL POWERS OF SIGMOID (PPS) WAVELET NEURAL NETWORKS

João Fernando Marar
Department of Computing, Adaptive Systems and Computational Intelligence Laboratory
Faculdade de Ciências, São Paulo State University, Bauru, São Paulo, Brazil

Helder Coelho
Department of Informatics, Laboratory of Agent Modelling
Faculdade de Ciências, Lisbon University, Lisbon, Portugal

Keywords: Artificial Neural Network, Function Approximation, Polynomial Powers of Sigmoid (PPS), Wavelets Functions, PPS-Wavelet Neural Networks, Activation Functions, Feedforward Networks.

Abstract: Wavelet functions have been used as the activation function in feedforward neural networks. An abundance of R&D has been produced on wavelet neural network area. Some successful algorithms and applications in wavelet neural network have been developed and reported in the literature. However, most of the aforementioned reports impose many restrictions in the classical backpropagation algorithm, such as low dimensionality, tensor product of wavelets, parameters initialization, and, in general, the output is one dimensional, etc. In order to remove some of these restrictions, a family of polynomial wavelets generated from powers of sigmoid functions is presented. We described how a multidimensional wavelet neural networks based on these functions can be constructed, trained and applied in pattern recognition tasks. As an example of application for the method proposed, it is studied the exclusive-or (XOR) problem.

1 INTRODUCTION

Wavelet functions have been successfully used in many problems as the activation function of feedforward neural networks. There are claims that many biological fundamental properties can emerge from wavelet transformation (Marar, 1997). An abundance of R&D has been produced on wavelet neural network area. Some successful algorithms and applications in wavelet neural network have been developed and reported in the literature (Zhang and Benveniste, 1992; Marar, 1997; Oussar and Dreyfus, 2000; Chen and Hewit, 2000; Zhang and San, 2004; Fan and Wang, 2005; Zhang and Pu, 2006; Chen et al., 2006; Avci, 2007; Jiang et al., 2007; Misra et al., 2007).

However, most of the aforementioned reports impose many restrictions in the classical backpropagation algorithm, such as low dimensionality, tensor product of wavelets, parameters initialization, and, in general, the output is one dimensional, etc.

In order to remove some of these restrictions, we develop a robust Three Layer PPS-Wavelet multidimensional strongly similar to classical Multilayer Perceptron. The great advantage of this new approach is that PPS-Wavelets offers the possibility choice of the function that will be used in the hidden layer, without need to develop a new learning algorithm. This is a very interesting property for the design of new wavelet neural networks architectures. This paper is organized as follows. Section 2 covers basic theoretical aspects in function approximation. Section 3 introduces the wavelet sigmoidal function. Section 4 presents the framework used in this research. Section 5 deals with application of exclusive-or (XOR) problem. Section 6 concludes this paper.

2 FUNCTION APPROXIMATION

Multilayer perceptron networks (MLP) have been intensely studied as efficient tools for arbitrary function approximation. Amongst the developments achieved in the theory of function approximation using MLP, the work carried out by Hecht-Nielsen resulted in an improved version for the superposition theorem defined by Sprecher (Hecht-Nilsen, 1987). Galant and White in 1988 showed that a feedforward network with one hidden layer of processing units that use flat

Fernando Marar J. and Coelho H. (2008).
MULTIDIMENSIONAL POLYNOMIAL POWERS OF SIGMOID (PPS) WAVELET NEURAL NETWORKS.
In *Proceedings of the First International Conference on Bio-inspired Systems and Signal Processing*, pages 261-268
DOI: 10.5220/0001067302610268

cosines as the activation function correspond to a special case of Fourier networks that can approximate a Fourier series for a given function. Cybenko developed a rigorous demonstration that MLPs with only one hidden layer of processing elements is sufficient to approximate any continuous function with support in a hypercube (Cybenko, 1989).

The theorem is directly applied to MLP. The sigmoid, radial basis and wavelets functions are a common choice for the network construction since it satisfies the conditions imposed in the theorem. The theorem of function approximation provides a mathematical basis that gives support to the approximation of any continuous arbitrary function. Furthermore, it defines for the case of MLP that a network composed of only one hidden layer neurons is sufficient to compute, in a given problem, a mapping from the input space to the output space, based on a set of training examples. However, with respect to training speed and ease of implementation, the theorem does not provide any insight about the solutions developed. The choice of activation functions and the learning algorithm defines which particular network is used. In any situation, the neurons operate as a set of functions that generate an arbitrary basis for function approximation which is defined based on the information extracted from the input-output pairs. For training a feedforward network, the backpropagation algorithm is one of the most frequently employed in practical applications and can be seen as an optimization.

3 WAVELET FUNCTIONS

Two categories of wavelet functions, namely, orthogonal wavelets and wavelet frames (or non-orthogonal), were developed separately by different interests. An orthogonal basis is a family of wavelets that are linearly independent and mutually orthogonal, this eliminates the redundancy in the representation. However, orthogonal wavelets bases are difficult to construct because the wavelet family must satisfy stringent criteria (Daubechies, 1992; Chui, 1992). This way, for these difficulties, orthogonal wavelets is a serious drawback for their application to function approximation and process modeling (Oussar and Dreyfus, 2000). 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.

Let $\varphi_j(x)$ a wavelet, the relation:

$$\varphi_j(x) = \varphi(d_j.(x - t_j))$$

where t_j is the translation factors and d_j is the dilation factors $\in \mathbf{R}$. The family of functions generated by $\mho$ can be defined as:

$$\mho = \left\{\varphi(d_j.(x - t_j)), t_j \textit{ and } d_j \in \mathbf{R}\right\}$$

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

$$c\|f\|^2 \leq \sum_j |<\varphi_j, f>|^2 \leq C\|f\|^2$$

where $\varphi_j \in \mho$, $\|f\|$ denotes the norm of function f and $< \varphi_j, f >$ the inner product of functions. Families of wavelet frames of $L^2(\mathbf{R})$ are universal approximators (Zhang and Benveniste, 1992; Pati and Krishnaprasad, 1993). In this work, we will show that wavelet frames allow practical implementation of multidimensional wavelets. This is important when considering problems of large input and output dimension. For the modeling of multi-variable processes, such as, the artificial neural networks biologically plausible, multidimensional wavelets must be defined. In the present work, we use multidimensional wavelets constructed as linear combination of sigmoid, denominated Polynomial Powers of Sigmoid Wavelet (PPS-wavelet).

3.1 Sigmoidal Wavelet Functions

In (Funahashi, 1989) is showed that:

Let $s(x)$ a function different of the constant function, limited and monotonically increase. For any $0 < \alpha < \infty$ the function created by the combination of sigmoid is described in Equation 1:

$$g(x) = s(x+\alpha) - s(x-\alpha) \quad (1)$$

where $g(x) \in L^1$ $(\mathbf{R})$, i.e,

$$\int_{-\infty}^{\infty} g(x) < \infty$$

in particular, the sigmoid function satisfies this property.

Using the property came from the Equation 1, in (Pati and Krishnaprasad, 1993) boundary suggest the construction of wavelets based on addition and subtraction of translated sigmoidal, which denominates wavelets of sigmoid. In the same article show a process of construction of sigmoid wavelet by the substitution of the function $s(x)$ by $\Upsilon(qx)$ in the Equation 1. So, the Equation 2 is the wavelet function created in (Pati and Krishnaprasad, 1993).

$$\psi(x) = g(x+r) - g(x-r) \quad (2)$$

where $r > 0$. By terms of sigmoid function, the Equation 2, $\psi(x)$ is given by:

$$\psi(x) = \Upsilon(qx+a+r) - \Upsilon(qx-a+r) - \Upsilon(qx+a-r) + \Upsilon(qx-a-r) \quad (3)$$

where $q > 0$ is a constant that control the curve of the sigmoid function and α and $r \in \mathbf{R} > 0$.

Pati and Krishnaprasad demonstrated that the function $\psi(x)$ satisfies the admissibility condition for wavelets (Daubechies, 1992; Chui, 1992). The Fourier Transform of the function $\psi(x)$ is given by the Equation 4:

$$\int_{-\infty}^{\infty} \psi(x) e^{-iwx} dx = -i \frac{4\pi}{q} \frac{sin(w\alpha) sin(wr)}{sinh(\frac{\pi w}{q})} \quad (4)$$

In particular, we accepted for analysis and practical applications the family of sigmoid wavelet generated by the parameters $q = 2$ and $\alpha = r$, as example. So, the Equation 3 can be rewritten the following form:

$$\psi(x) = \Upsilon(2x+m) - 2\Upsilon(2x) - \Upsilon(2x-m) \quad (5)$$

where $m = \alpha + r$.

Following, partially, this research line, we present in the next section a technique for construction of wavelets based on linear combination of sigmoid powers.

4 POLYNOMIAL POWERS OF SIGMOID

The Polynomial Powers of Sigmoid (PPS) is a class of functions that have been used in recent years to solve a wide range of problems related to image and signal processing (Marar, 1997). Let $\Upsilon : \mathbf{R} \rightarrow \mathbf{[0,1]}$ be a sigmoid function defined by $\Upsilon(x) = \frac{1}{1+e^{-x}}$. The $n^{th}-$power of the sigmoid function is a function $\Upsilon^n : \mathbf{R} \rightarrow \mathbf{[0,1]}$ defined by $\Upsilon^n(x) = \left(\frac{1}{1+e^{-x}}\right)^n$.

Let Θ be set of all power functions defined by (6):

$$\Theta = \{\Upsilon^0(x), \Upsilon^1(x), \Upsilon^2(x), \ldots, \Upsilon^n(x), \ldots\} \quad (6)$$

An important aspect is that the power these functions, still keeps the form of the letter *S*. Looking the form created by the power functions of sigmoid, suppose that the n^{th} power of the sigmoid function to be represented by the following form:

$$\Upsilon^n(x) = \frac{1}{a_0 + a_1 e^{-x} + a_2 e^{-2x} + \cdots + a_n e^{-nx}} \quad (7)$$

where $a_0, a_1, a_2, \ldots, a_n$ are some integer values. The extension of the sigmoid power can be viewed like lines of a Pascal′s triangle. The set of function written by linear combination of polynomial powers of sigmoid is defined as PPS function. The degree of the PPS is given by the biggest power of the sigmoid terms.

4.1 Polynomial Wavelet Family on PPS

The derivative of a function $f(x)$ on $x = x_0$ is defined by:

$$f'(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

since the limits there is. So, if we do the computation of the Equation 8 :

$$\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \quad (8)$$

for a small value of Δx , showed have a good approximation for $f'(x_0)$. Naturally, Δx can be positive or negative. So, if is we use negative value for Δx, the expression will be:

$$\frac{f(x_0 - \Delta x) - f(x_0)}{-\Delta x} \quad (9)$$

This way, we can say that the arithmetic measure of the Equations 8 and 9 will be a good approximation for $f'(x_0)$ too. Then, we can write the following Equation 10:

$$f'(x_0) \simeq \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} \quad (10)$$

By convenience, we consider $p = 2\Delta x$ and its substitution in the Equation 10. So, we have the Equation 11:

$$f'(x_0) \simeq \frac{f(x_0 + \frac{p}{2}) - f(x_0 - \frac{p}{2})}{p} \quad (11)$$

this point we computed an approximated value for the second derivative of $f(x)$ in $x = x_0$. From the Equation 11, changing $f(x)$ by $f'(x)$, we obtain the Equation 12 :

$$f''(x_0) \simeq \frac{f'(x_0 + \frac{p}{2}) - f'(x_0 - \frac{p}{2})}{p} \quad (12)$$

reusing the Equation 11, we can write:

$$f'(x_0 + \frac{p}{2}) \simeq \frac{f(x_0 + p) - f(x_0)}{p}$$

and

$$f'(x_0 - \frac{p}{2}) \simeq \frac{f(x_0) - f(x_0 - p)}{p}$$

using these results in the Equation12, we have an approximation of the second derivative of $f(x)$ in $x = x_0$ that is given by:

$$f''(x_0) \simeq \frac{f(x_0+p) - 2f(x_0) + f(x_0-p)}{p^2} \quad (13)$$

The approximation given by the Equation 13 is extremely adequate for the that $f(x)$ is a sigmoid function. Suppose that $f(x)$ is a sigmoid, for example, $\Upsilon(x)$. So, the second derivative of $\Upsilon(x)$ is approximated by the Equation 14:

$$\Upsilon''(x_0) \simeq \frac{\Upsilon(x_0+p) - 2\Upsilon(x_0) + \Upsilon(x_0-p)}{p^2} \quad (14)$$

Due the fact of the sigmoid function to be continuous and differentiable for any $x \in \mathbf{R}$, we can say that the Equation 14 is true for any x_0, then we can write the Equation 15, defined for all $x \in \mathbf{R}$.

$$\Upsilon''(x) \simeq \frac{\Upsilon(x_0+p) - 2\Upsilon(x) + \Upsilon(x-p)}{p^2} \quad (15)$$

Comparison the Equations 15 and 5, we do there analysis for the approximation of the second derivative of sigmoid function. The first for values of $p \geq 1$ and the second for values of $p < 1$.

Case $p \geq 1$:

It is clear that the function given by the sigmoid second derivative approximation, Equation 15, also will have the same form of the Pati and Krishnaprasad functions, except of a p^2 constant that divides their amplitude. So, the following result is true: when $p > 1$ always there is a sigmoid wavelet which integral of the admissibility condition (Daubechies, 1992; Chui, 1992) limited the same integral of the Equation 15. Therefore, the approximation of the second derivative of the sigmoid function is a wavelet too.

Case $p < 1$:

In this case, we will analyze when p is going to zero, i.e.,

$$\lim_{p \to 0} \frac{\Upsilon'(x_0+p) - 2\Upsilon(x) + \Upsilon'(x-p)}{p^2} \quad (16)$$

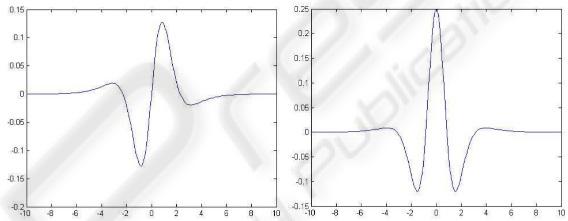
this limit tends to the second derivative of the function is given on PPS terms by:

$$\varphi_2(x) = 2\Upsilon(x)^3 - 3\Upsilon(x)^2 + \Upsilon(x) \quad (17)$$

where we denominated $\varphi_2(x)$ the first wavelet the sigmoid function. The others derivatives, begin on the second, we considered true by derivative property by Fourier Transform (Marar, 1997). The successive derivation process of sigmoid functions, allowed to join a family of wavelets polynomial functions. Among many applications for this family of PPS-wavelets, special one is that those functions can be used like activation functions in artificial neurons. The following results correspond to the the analytical functions for the elements $\varphi_3(x)$ and $\varphi_4(x)$ that are represented by:

$$\varphi_3(x) = -6\Upsilon^4(x) + 12\Upsilon^3(x) - 7\Upsilon^2(x) + \Upsilon(x)$$

$$\varphi_4(x) = 24\Upsilon^5(x) - 60\Upsilon^4(x) + 50\Upsilon^3(x) - 15\Upsilon^2(x) + \Upsilon(x)$$



$\varphi_4(x)$ $\varphi_5(x)$

Figure 1: PPS-wavelets examples.

4.2 Estimating the Coefficients of PPS-wavelets

Considering j the number of wavelets that are to be defined, the algorithm below calculates a matrix of integer values that estimates the coefficients of the PPS-wavelets.

Step 1: Initialization

$$C_{1,1} \leftarrow 1;$$
$$C_{1,2} \leftarrow 1;$$

The initial values are considered only auxiliary variables. The matrix of value associated with the process of wavelet construction is obtained from the second row.

Step 2: Calculate the coefficient of the PPS of the highest degree

$$n \leftarrow 3;$$
$$n \leftarrow n+1; \quad (n \leq j)$$
$$C_{n-1,n} \leftarrow C_{n-2,n-1} * (n-1) * (-1)^{n+1};$$

Step 3: Calculate the coefficients of the remaining terms of the polynomial

$$\begin{aligned} k &\leftarrow n; \\ k &\leftarrow k-1; \quad (k>2) \\ C_{n-1,k-1} &\leftarrow C_{n-2,k-1} * (k-1) + \\ &\quad\; C_{n-2,k-2} * (k-2) * (-1)^k; \end{aligned}$$

Step 4: Calculate the coefficients of the first power variable

$$C_{n-1,1} \leftarrow 1$$

It is important to notice that steps 2 and 3 are cascaded by an inherent dependence on variable n. By proceeding in above way, a family of polynomial wavelets are generated.

4.3 PPS Wavelet Neural Network

Let us consider the canonical structure of the multidimensional PPS-wavelet neural network (PPS-WNN), as shown in Figure 2.

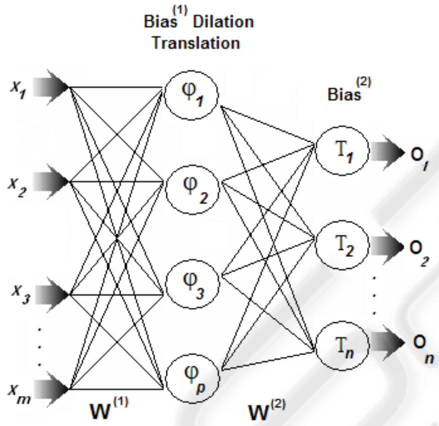


Figure 2: PPS-wavelet neural network Architectures.

For the PPS-WNN in Figure 2, when a input pattern $X = (x_1, x_2, \ldots, x_m)^T$ is applied at the input of the network, the output of the i^{th} neuron of output layer is represented as a function approximation problem, ie, $f : \mathbf{R}^m \rightarrow \mathbf{[0,1]}^n$, given by:

$$O_i(x) \simeq \Upsilon_i\left(\sum_{j=1}^{p} w_{ij}^{(2)} \varphi_j\left(d_j.\left(\sum_{k=1}^{m} w_{jk}^{(1)} x_k - b_j^{(1)}\right) - t_j\right) - b_i^{(2)}\right) \quad (18)$$

where p is number of hidden neurons, $\Upsilon(.)$ is sigmoid function, $\varphi(.)$ is the PPS-wavelet, $w^{(2)}$ are weight between the hidden layer to the output layer, $w^{(1)}$ are weights between the input to the hidden layer, d are dilation factors and t are translation factors of the PPS-wavelet, $b^{(1)}$ and $b^{(2)}$ are bias factors of the hidden layer and output layer, respectively.

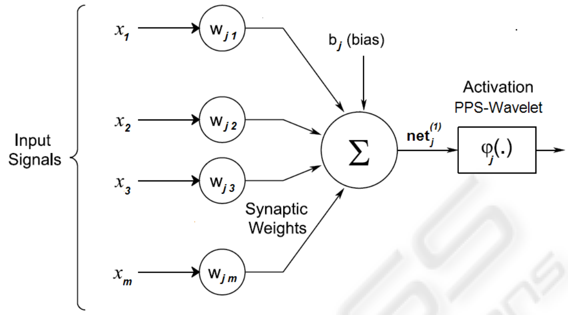


Figure 3: The Hidden Neuron of PPS-Wavelet Neural Network.

The PPS-WNN contains PPS-wavelets as the activation function in the hidden layer (Figure 3) and sigmoid function as the activation function in the output layer (Figure 4).

The output of the j^{th} PPS-wavelet hidden neuron (Figure 3) is given by :

$$\circledast_j = \varphi_j(d_j.(net_j^{(1)} - t_j))$$

where

$$net_j^{(1)} = \sum_{k=1}^{m} w_{jk}^{(1)} x_k - b_j^{(1)}$$

The output of the i^{th} output layer neuron (Figure 4)

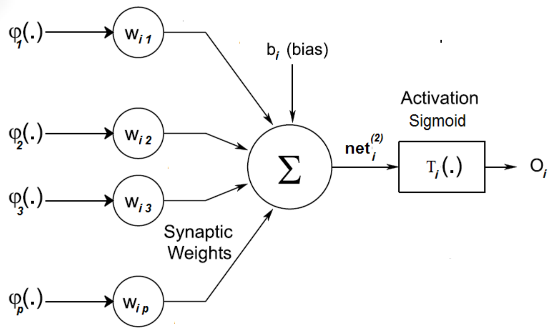


Figure 4: The Output Neuron of PPS-Wavelet Neural Network.

is given by:

$$\circledcirc_i = \frac{1}{1 + exp(-net_i^{(2)})}$$

where

$$net_i^{(2)} = \sum_{j=1}^{p} w_{ij}^{(2)} \varphi_j(d_j.(net_j^{(1)} - t_j)) - b_i^{(2)}$$

The adaptive parameters of the PPS-WNN consist of all weights, bias, translations and dilation terms. The sole purpose of the training phase is to determine the "optimum" setting of the weights, bias, translations and dilation terms so as to minimize the difference between the network output and the target output. This difference is referred to as training error of the network. In the conventional backpropagation algorithm, the error function is defined as:

$$E = \frac{1}{2}\sum_{q=1}^{s}\sum_{i=1}^{n}(y_{qi} - o_{qi})^2 \quad (19)$$

where n is the dimension of output space, s is the number of training input patterns

The most popular and successful learning method for training the multilayer perceptrons is the backpropagation algorithm. The algorithm employs an iterative gradient descendent method of minimization which minimizes the mean squared error (L^2 norm) between the desired output (y_i) and network output (o_i). From Equations (18) and (19), we could deduce the partial derivatives of the error to each PPS-wavelet neural network parameter′s, which is given by:

Partial Equations of the Output Layer

$$\frac{\partial E}{\partial w_{ij}^{(2)}} = -\sum_{q=1}^{s}(y_{qi} - o_{qi}).o_{qi}.(1 - o_{qi}).\varphi_j(d_j.(net_{qj}^{(1)} - t_j)) \quad (20)$$

$$\frac{\partial E}{\partial b_i^{(2)}} = \sum_{q=1}^{s}(y_{qi} - o_{qi}).o_{qi}.(1 - o_{qi}) \quad (21)$$

Partial Equations of the Hidden Layer

$$\frac{\partial E}{\partial w_{jk}^{(1)}} = -d_j.\sum_{q=1}^{s}[\varphi'_j(d_j.(net_{qj}^{(1)} - t_j)).x_{qk}.\sum_{i=1}^{n}(y_{qi} - o_{qi}).o_{qi}.(1 - o_{qi}).w_{ij}^{(2)}] \quad (22)$$

$$\frac{\partial E}{\partial b_j^{(1)}} = \sum_{q=1}^{s}[\varphi'_j(d_j.(net_{qj}^{(1)} - t_j)).d_j.\sum_{i=1}^{n}(y_{qi} - o_{qi}).o_{qi}.(1 - o_{qi}).w_{ij}^{(2)}] \quad (23)$$

Partial Equations of the PPS-Wavelet Parameters

$$\frac{\partial E}{\partial d_j} = \sum_{q=1}^{s}\{[\varphi'_j(d_j.(net_{qj}^{(1)} - t_j)).(net_{qj}^{(1)} - t_j)].\sum_{i=1}^{n}(y_{qi} - o_{qi}).o_{qi}.(1 - o_{qi}).w_{ij}^{(2)}\} \quad (24)$$

$$\frac{\partial E}{\partial t_j} = d_j\sum_{q=1}^{s}[\varphi'_j(d_j.(net_{qj}^{(1)} - t_j)).\sum_{i=1}^{n}(y_{qi} - o_{qi}).o_{qi}.(1 - o_{qi}).w_{ij}^{(2)}] \quad (25)$$

After computing all partial derivatives the network parameters are updated in the negative gradient direction. A learning constant γ defines the step length of the correction, r is the iteration and momentum factor is β. The corrections are given by:

$$w_{ij}^{(2)}(r+1) = w_{ij}^{(2)}(r) - \gamma.\frac{\partial E}{\partial w_{ij}^{(2)}} + \beta.(w_{ij}^{(2)}(r) - w_{ij}^{(2)}(r-1))$$

$$b_i^{(2)}(r+1) = b_i^{(2)}(r) - \gamma.\frac{\partial E}{\partial b_i^{(2)}} + \beta.(b_i^{(2)}(r) - b_i^{(2)}(r-1))$$

$$w_{jk}^{(1)}(r+1) = w_{jk}^{(1)}(r) - \gamma.\frac{\partial E}{\partial w_{jk}^{(1)}} + \beta.(w_{jk}^{(1)}(r) - w_{jk}^{(1)}(r-1))$$

$$b_j^{(1)}(r+1) = b_j^{(1)}(r) - \gamma.\frac{\partial E}{\partial b_j^{(1)}} + \beta.(b_j^{(1)}(r) - b_j^{(1)}(r-1))$$

$$d_j(r+1) = d_j(r) - \gamma.\frac{\partial E}{\partial d_j} + \beta.(d_j(r) - d_j(r-1))$$

$$t_j(r+1) = t_j(r) - \gamma.\frac{\partial E}{\partial t_j} + \beta.(t_j(r) - t_j(r-1))$$

4.4 Algorithm to PPS Wavelet Neural Network

In this section, the learning algorithm to the PPS-wavelet neural network is proposed by using the back-propagation method.

Begin

initialize-choice-PPS-function();

initialize-architecture();

initialize-weights();

initialize-PPSwavelet-neurons-dilatations();

initialize-PPSwavelet-neurons-translations();
initialize-neurons-bias();
Do-While ($epoch \leq epoch_{max}$)
or ($\frac{1}{2} total_{error} > acceptable_{error}$)
Begin$_{Do-While}$
$total_{error} \leftarrow 0$;
randomize-input-patter-order();
For pattern counter $q = 1..s$
Begin$_{for}$
read input pattern $x_{(q,j)} : j = 1..m$
read input target vector $y_{(q,i)} : i = 1..n$
acc-param-h-layer(); by Eqs. (22)- (25)
compute $O_{(q,i)}$ by Eq. (18)
acc-param-o-layer(); by Eqs. (20)- (21)
$total_{error} \leftarrow total_{error} + (y_{(p,k)} - O_{(p,k)})^2$
End$_{for}$
IF ($total_{error} > acceptable_{error}$) Then
Begin$_{Then}$
update-param-o-layer();
update-param-h-layer()
End$_{then}$
$epoch \leftarrow$ epoch+1
End$_{Do-While}$

End

where the initialization procedures, attribute random values on **[0,1]** to the parameters. However, improvements in the initialization process have been proposed by the selection of basic functions PPS-wavelet (de Queiroz and Marar, 2007).

5 PATTERN RECOGNITION AND THE XOR PROBLEM

The pattern recognition problem consists of designing algorithms that automatically classify feature vectors associated with specific patterns as belonging to one of a finite number of classes. A benchmark problem in the design of pattern recognition systems is the exclusive OR (XOR) problem. However, to solve this problem, effectively ended research interest in the area of Artificial Neural Networks for over 21 years, which highlights the importance of the XOR problem in the design of pattern recognition systems. The standard XOR problem is depicted in Figure 5:

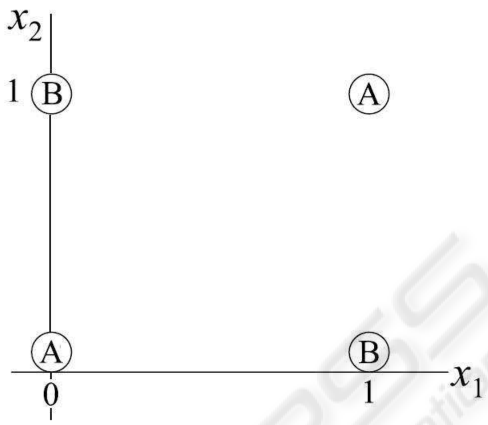


Figure 5: The exclusive or (XOR) problem: points (0,0) and (1,1) are members of class A; points (0,1) and (1,0) are members of class B.

Here the diagonally opposite corner-pairs of the unit square form two classes, A and B. From the Figure 5, it is clear that it is not possible to draw a single straight line which will separate the two classes. This observation is crucial in explaining the the complexity to solve this problem. This problem can be solved using multi-layer perceptrons (MLPs), or by using more elaborate single-layer artificial neural network such as the PPS Wavelet neural network, can be trained to solve this problem in a straightforward manner. In order to demonstrate the adaptive capacity of the PPS neural networks, we accomplished a study with the functions $\varphi_2(x)$ and $\varphi_5(x)$. The results are illustrated in Figures 6 and 7 respectively:

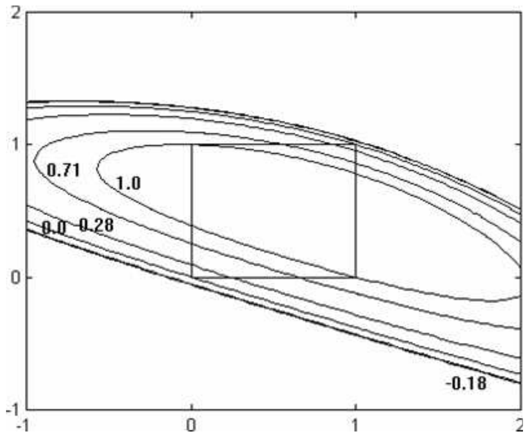


Figure 6: XOR problem based on $\varphi_2(x)$.

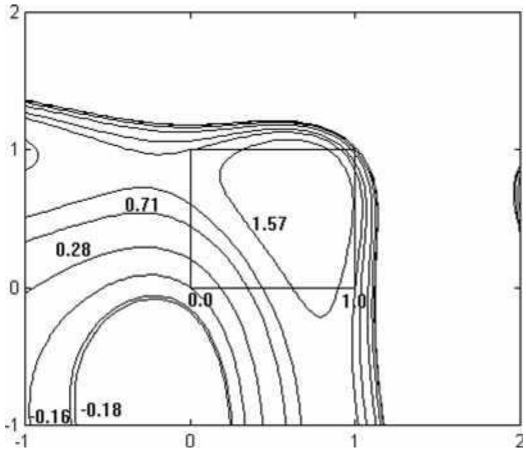


Figure 7: XOR problem based on $\varphi_5(x)$.

6 CONCLUSIONS

Neural networks and wavelet transform have been recently seen as attractive tools for developing efficient solutions for many real world problems in function approximation. The combination of neural networks and wavelet transform gives rise to an interesting and powerful technique for function approximation referred to as wavenets. Function approximation is a very important task in environments where computation has to be based on extracting information from data samples in the real world processes. So, mathematical model is a very important tool to guarantee the development of the neural network area.

ACKNOWLEDGEMENTS

We would like to thank the CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) process number $3634/06-0$ and the Lisbon University that supported this investigation.

REFERENCES

Avci, E. (2007). An expert system based on wavelet neural network-adaptive norm entropy for scale invariant texture classification. *Expert Systems with Applications*, 32:919–926.

Chen, H. and Hewit, J. (2000). Application of wavelet transform and neural networks to recognition and classivication of blemishes. *Mechatronics*, 10:699–711.

Chen, Y., Yang, B., and Dong, J. (2006). Time series prediction using a local linear wavelet neural network. *Neurocomputing*, (69):449–465.

Chui, C. (1992). *An Introduction to Wavelets*. Academic Press.

Cybenko, G. (1989). Approximation by superposition of a sigmoidal function. *Mathematics of Control, signals and Systems*, 3:303–314.

Daubechies, I. (1992). *Ten lecture on wavelets*. Society for Industrial and Applied Mathematics (SIAM).

de Queiroz, R. A. B. and Marar, J. F. (2007). Algorítmos heurísticos para a seleção de neurônios em redes neurais polinomios potências de sigmoide (pps)-wavelet. *TEMA Tend. Mat. Apl. Comput.*, 8(1):129–138.

Fan, J. and Wang, X. F. (2005). A wavelet view of small-world networks. *IEEE Transactions on Circuits and Systems*, pages 1–4.

Funahashi, K. (1989). On the approximate realization of continuos mappings by neural networks. *Neural Networks*, (2):183–192.

Hecht-Nilsen, R. (1987). Kolmogorov's mapping neural networks existence theorem. In *1st IEEE Inter. Conf. on Neural Networks*, volume 3, pages 11–14.

Jiang, X., Mahadevan1, S., and Adeli, H. (2007). Bayesian wavelet packet denoising for structural system identification. *Struct. Control Health Monit.*, 14:333–356.

Marar, J. F. (1997). *Polinomios Potências de Sigmoide (PPS): Uma nova Técnica para Aproximação de Funções, Construção de Wavenets e suas aplicações em Processamento de Imagens e Sinais*. PhD thesis, Universidade Federal de Pernambuco - Departamento de Informática.

Misra, B. B., Dash, P. K., and Panda, G. (2007). Pattern classification using local linear wavelet neural network. *International Conference on Information and Communication Technology*, pages 92–95.

Oussar, Y. and Dreyfus, G. (2000). Initialization by selection for wavelet neural traing. *Neurocomputing*, 34:131–143.

Pati, Y. and Krishnaprasad, P. (1993). Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. *IEEE Transactions on Neural Networks*, 4(1):73–85.

Zhang, H. and Pu, J. (2006). A novel self-adaptive control framework via wavelet neural netwok. In *6th World congress on intelligent control and automation*, pages 2254–2258.

Zhang, Q. and Benveniste, A. (1992). Wavelet networks. *IEEE Transactions on Neural Networks*, 3(6):889–898.

Zhang, Z. and San, Y. (2004). Adaptive wavelet neural network for prediction of hourly nox and no2 concentrations. In *Winter Simulation Conference*, pages 1770–1778.