3

Artificial Neural Networks and Efficient Optimization Techniques for Applications in Engineering

Rossana M. S. Cruz[1], Helton M. Peixoto[2] and Rafael M. Magalhães[3]
[1]*Federal Institute of Education, Science and Technology of Paraiba, João Pessoa, PB,*
[2]*Federal University of Rio Grande do Norte, Natal, RN,*
[3]*Federal University of Paraiba, Rio Tinto, PB,*
Brazil

1. Introduction

This chapter proposal describes some artificial neural network (ANN) neuromodeling techniques used in association with powerful optimization tools, such as natural optimization algorithms and wavelet transforms, which can be used in a variety of applications in Engineering, for example, Electromagnetism (Cruz, 2009), Signal Processing (Peixoto et al., 2009b) and Pattern Recognition and Classification (Magalhães et al., 2008).

The application of ANN models associated with RF/microwave devices (Cruz et al., 2009a, 2009b; Silva et al., 2010a) and/or pattern recognition (Lopes et al., 2009) becomes usual. In this chapter, we present neuromodeling techniques based on one or two hidden layer feedforward neural network configurations and modular neural networks – trained with efficient algorithms, such as Resilient Backpropagation (RPROP) (Riedmiller & Braun, 1993), Levenberg-Marquardt (Hagan & Menhaj, 1999) and other hybrid learning algorithms (Magalhães et al., 2008), in order to find the best training algorithm for such investigation, in terms of convergence and computational cost. The mathematical formulation and implementation details of neural network models, wavelet transforms and natural optimization algorithms are also presented.

Natural optimization algorithms, which are stochastic population-based global search methods inspired in nature, such as genetic algorithm (GA) and particle swarm optimization (PSO) are effective for optimization problems with a large number of design variables and inexpensive cost function evaluation (Kennedy & Eberhart, 1995; R. Haupt & S. Haupt, 2004). However, the main computational drawback for optimization of nonlinear devices relies on the repetitive evaluation of numerically expensive cost functions (Haupt & Werner, 2007; Rahmat-Samii, 2003). Finding a way to shorten the optimization cycle is highly desirable. In case of GA, for example, several schemes are available in order to improve its performance, such as: the use of fast full-wave methods, micro-genetic algorithm, which aims to reduce the population size, and parallel GA using parallel computation (R. Haupt & S. Haupt, 2004; Haupt & Werner, 2007). Therefore, this chapter

also describes some hybrid EM-optimization methods, using continuous-GA and PSO algorithms, blended with multilayer perceptrons (MLP) artificial neural network models. These methods are applied to design spatial mesh filters, such as frequency selective surfaces (FSSs). Moreover, the MLP model is used for fast and accurate evaluation of cost function into continuous GA and PSO simulations, in order to overcome the computational requirements associated with full wave numerical simulations.

Wavelets and artificial neural networks (ANN) have generated enormous interest in recent years , both in science and practical applications (Peixoto et al., 2009c). The big advantage of using wavelets is the fact of these functions make a local behavior, not only in the frequency domain but also in the field space and time. ANNs are capable of learning from a training set, which makes it useful in many applications, especially in pattern recognition. The purpose of using wavelet transforms is to find an easier way to compress and extract the most important features present in images, thereby creating a vector of descriptors that should be used to optimize the pattern recognition by a neural network. The array of descriptors contains elements whose values accurately describe the image content, and should take up less space than a simple pixel by pixel representation. The greatest difficulty found in this process is the generation of this vector, where the image content of interest should be very well described, in order to show really relevant features. Thus, this chapter proposal also shows a way to use a multiresolution technique, such as the wavelet transforms, to perform filtering, compression and extraction of image descriptors for a later classification by an ANN.

This chapter is organized in six sections. Section 2 presents the most important fundamentals of artificial neural networks and the methodology used for the investigated applications. In section 3, artificial neural networks are optimized using wavelet transforms for applications in image processing (extraction and compression). Section 4 presents an EM-optimization using artificial neural networks and natural optimization algorithms for the optimal synthesis of stop-band filters, such as frequency selective surfaces. Section 5 shows a modular artificial neural network implementation used for pattern recognition and classification. Finally, section 6 presents important considerations about the artificial neural network models used in association with efficient optimization tools for applications in Engineering.

2. Artificial Neural Networks

2.1 Fundamentals

Rosenblatt (1958) perceptron is the most used artificial neuron in neural network configurations and is based on the nonlinear model proposed by McCulloch and Pitts (1943). In this model, neurons are signal processing units composed by a set of input connections (weights), an adder (for summing the input signals, weighted by the respective synapses of a neuron, constituting a linear combiner) and an activation function, that can be linear or nonlinear, as shown in Fig. 1(a). The input signals are defined as x_i, $i = 0, 1,\ldots, N_i$, whose result corresponds to the level of internal activity of a neuron net_j, as defined in (1), where $x_0 = +1$ is the polarization potential (or bias) of the neurons. The output signal y_j is the activation function response $\varphi(\cdot)$ to the activation potential net_j, as shown in (2) (Silva et al., 2010b).

$$net_j = \sum_{i=0}^{Ni} w_{ji} \cdot x_i \quad (1)$$

$$y_j = \varphi(net_j) \quad (2)$$

For a feedforward neural network (FNN), the artificial neurons are set into layers. Each neuron of a layer is connected to those of the previous layer, as illustrated in Fig. 1(b). Signal propagation occurs from input to output layers, passing through the hidden layers of the FNN. Hidden neurons represent the input characteristics, while output neurons generate the neural network responses (Haykin, 1999).
Modular artificial neural network is based on a principle commonly used: divided to conquer. This concept aims to divide a large and complex task in a set of sub-tasks that are easier to be solved. The modular artificial neural network could be defined, in summary, as a set of learning machines, also called experts, whose decisions are combined to achieve a better answer than the answers achieved individually, that is, a machine with a better performance.
In the past few years, one of the main areas of learning machine is the characterization of methods capable to design this kind of machines. There are two types of these machines: static and dynamic structures. Modular neural networks, as seen in Fig. 1(c), is a dynamic type. The input signal is used by the gating network to design the global response. An advantage of modular artificial neural networks, when compared with other neural networks, is the learning speed. The machine learning process is accelerated in case of problems where it is observed a natural decomposition of data at simple functions. To develop the modular machine architecture and to implement the experts, it is usual to apply multilayer perceptrons (MLP) neural networks.

2.2 Methodology

Generally, the design of a neural network is composed by three main steps: configuration – how layers are organized and connected; learning – how information is stored; generalization – how neural network produces reasonable outputs for inputs not found in the training (Haykin, 1999). In this work, we use feedforward and modular neural networks associated with supervised learning to develop neural network models.
In the computational simulation of supervised learning process, a training algorithm is used for the adaptation of neural network synaptic weights. The instantaneous error $e(n)$, as defined in (3), represents the difference between the desired response, $d(n)$, and the neural network output, $z(n)$, at the iteration n, corresponding to the presentation of the nth training pattern $[x(n);(d(n)]$ (Silva et al., 2010b).

$$e(n) = z(n) - d(n) \quad (3)$$

Supervised learning can be illustrated through the block diagram of Fig. 2(a) and has as objective the minimization of the mean square error $E(t)$, given in (4), where the index t represents the number of training epochs (one complete presentation of all training examples, n = 1, 2,…, N, where N is the total number of examples, called an epoch) (Silva et al., 2010b).

$$E(t) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}e(n)^2 \tag{4}$$

Currently, there are several algorithms for the training of neural networks that use different optimization techniques (Peixoto et al., 2009a). The most popular training algorithms are those derived from backpropagation algorithm (Rumelhart et al., 1986). Among the family of backpropagation algorithms, the RPROP shows to be very efficient in the solution of complex electromagnetic problems. In this work, the stop criteria are defined in terms of the maximum number of epochs and/or the minimum error and the activation function used was the sigmoid tangent (Haykin, 1999).

After training, the neural network is submitted to a test, in order to verify its capability of generalizing to new values that do not belong to the training dataset, for example, parts of the region of interest where there is not enough knowledge about the modeled device/circuit. Therefore, the neural network operates like a "black box" model that is illustrated in Fig. 2(b) (Silva et al., 2010b).

Resilient backpropagation algorithm is a first-order local adaptive learning scheme. The basic principle of RPROP is to eliminate the harmful influence of the partial derivative size in the weight update. Only the sign of the derivative is considered to indicate the direction of the weight update, Δw_{hp}, as given in (5).

$$\Delta w_{hp}(t) = \begin{cases} -\Delta_{hp}(t), \; if \; \dfrac{\partial E}{\partial w_{hp}}(t) > 0 \\ +\Delta_{hp}(t), \; if \; \dfrac{\partial E}{\partial w_{hp}}(t) < 0 \\ 0, \; elsewhere \end{cases} \tag{5}$$

The second step of RPROP algorithm is to determine the new update-values $\Delta_{hp}(t)$. This is based on a sign-dependent adaptation process, similar to the learning-rate adaptation shown by Jacobs (1988). The changes in the weight size are exclusively determined by a weight 'update-value', Δ_{hp}, as given in (6).

$$\Delta_{hp}(t) = \begin{cases} \eta^{+} \cdot \Delta_{hp}(t-1), \; \dfrac{\partial E}{\partial w_{hp}}(t-1) \cdot \dfrac{\partial E}{\partial w_{hp}}(t) > 0 \\ \eta^{-} \cdot \Delta_{hp}(t-1), \; \dfrac{\partial E}{\partial w_{hp}}(t-1) \cdot \dfrac{\partial E}{\partial w_{hp}}(t) < 0 \\ \Delta_{hp}(t-1), \; elsewhere \end{cases} \tag{6}$$

Here, the following RPROP parameters were employed: $\eta^{+} = 1.2$ and $\eta^{-} = 0.5$. The update-values were restricted to the range $10^{-6} \leq \Delta_{hp} \leq 50$.

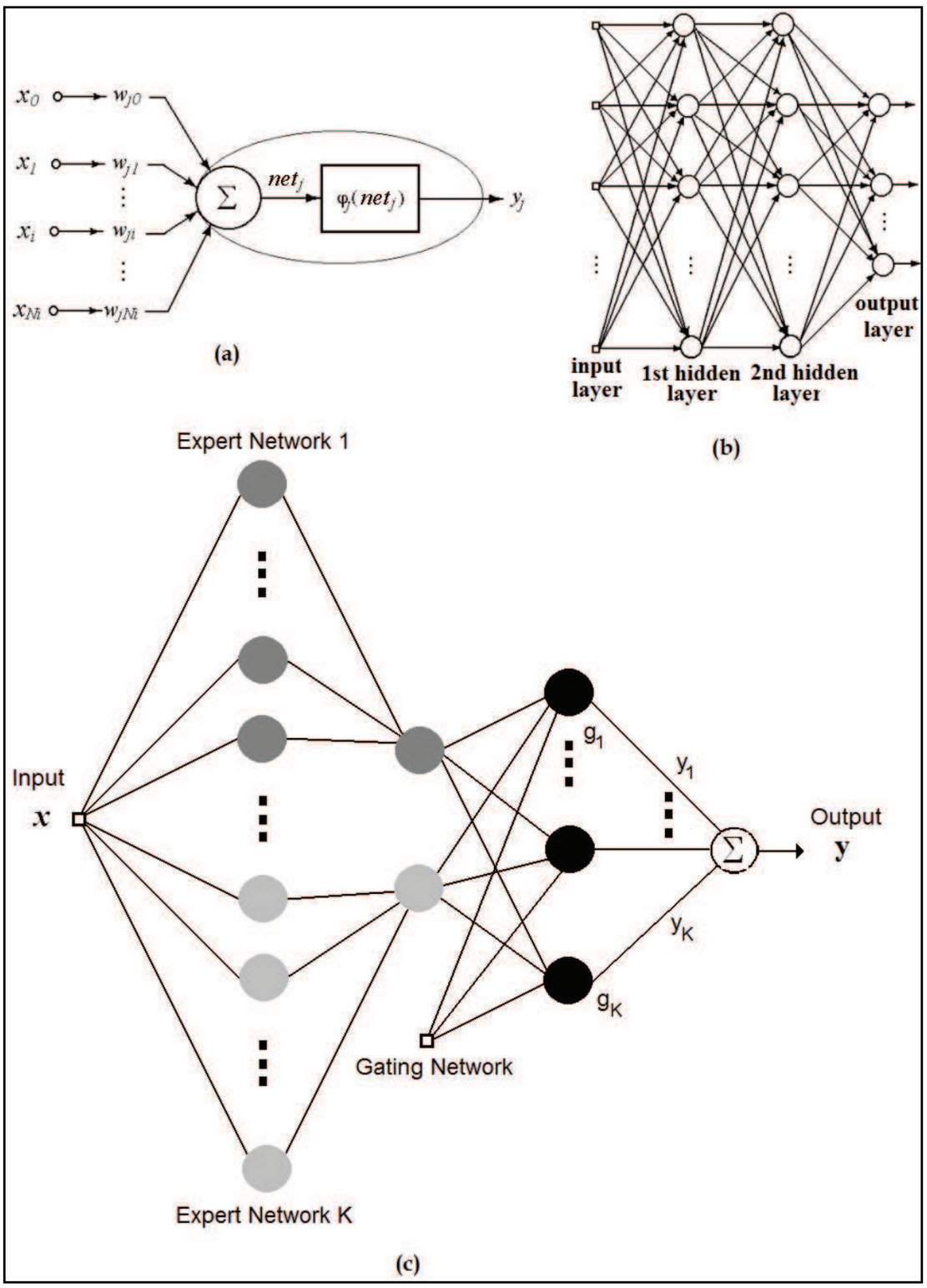


Fig. 1. (a) Nonlinear model of an artificial neuron; (b) FNN configuration with two hidden layers; (c) Extended modular neural network configuration with *K* experts

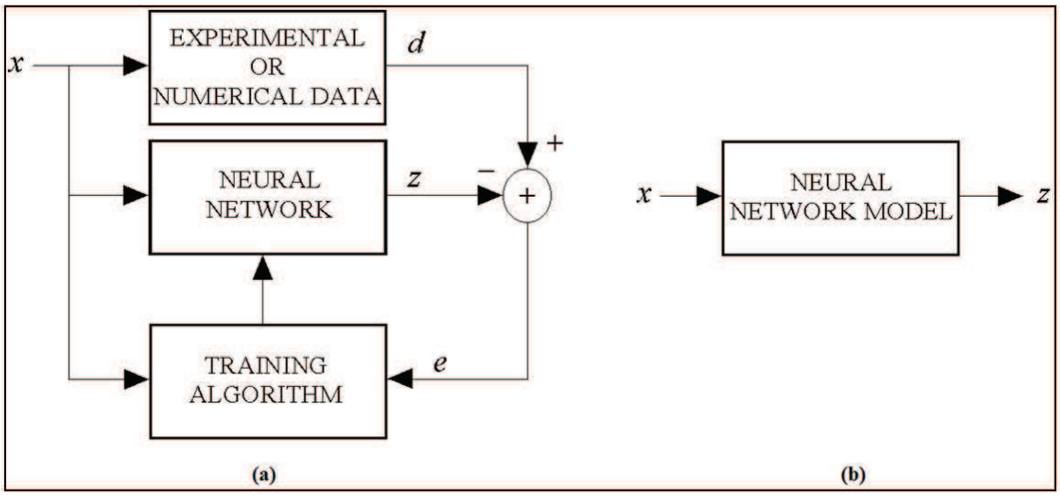


Fig. 2. (a) Block diagram of supervised learning; (b) neural network "black box" model

3. Artificial Neural Networks optimization using wavelet transforms

This section introduces the main concepts about wavelet transforms and some of their most important features used to optimize artificial neural networks: the extraction of characteristic information descriptors in order to improve training and pattern classification mechanisms. Moreover, a model that can be used to solve various problems related to pattern recognition is presented and a neural classifier is implemented to validate the importance of such optimization. Transforms are widely used mathematical tools to understand and analyze different signal behaviors. The objective of this analysis is to extract important information (or features) that can essentially represent the signal from some decomposition or transformation performed on it.

3.1 Wavelet transforms

Wavelet transforms have generated enormous interest from scientists, resulting in the development of applications in various areas, such as computer vision (Wang et al., 2009), seismology (Parolai et al., 2008), radar (Masnadi-Shirazi et al., 2009), astronomy (Ottensamer et al., 2008), image compression (Bhatia et al., 2009), signal filtering (Vimal et al., 2009), system optimization (Pinto, 2009) and many others. In general, the major advantage of using wavelet transforms is the possibility of applying it to non-stationary signals, which allows the study of function local behaviors, in both frequency and time-scale domains.

3.1.1 Advantages of use

Traditional methods of signal analysis based on Fourier transform can determine all the frequencies present in the signal, however its relationship to the time domain does not exist. To overcome this problem, it was created the Gabor transform (or STFT - Short Time Fourier Transform); the main idea of this transform is to introduce a new measure of local frequency as the local transformation observed the signal through a narrow window within which the signal remains nearly stationary (Oliveira, 2007). The problems in time and frequency domain resolution are result of a physical phenomenon known as Heisenberg's uncertainty principle (it is impossible to know the exact frequency and time that a signal occurs). This

phenomenon is independent of the transformation used (Oliveira, 2007). Therefore, the wavelet transform was developed as an alternative to the Gabor transform to solve the resolution problem.

Wavelets are mathematical transformations that separate signals in different components and extract each one of them with a resolution apropriated to its corresponding scale. According to the transformation characteristics, there is the Continuous Fourier Transform (CFT), that can be expressed as:

$$F(w) = \int_{-\infty}^{+\infty} f(t) e^{(-j2\pi ft)} dt \tag{7}$$

Knowing the spectrum *F(w)* of a signal, it is possible to obtain it in time domain, using the inverse transform concept, according to (8):

$$f(t) = \int_{-\infty}^{+\infty} \frac{1}{2\pi} F(w) e^{(-j2\pi ft)} dw \tag{8}$$

On the other hand, the Continuous Wavelet Transform (CWT) is given by:

$$CWT(\tau, a) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-\tau}{a} \right) dt \tag{9}$$

and its corresponding inverse can be expressed according to (10):

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} CWT(a,b) \left\{ \frac{1}{\sqrt{a}} \psi \left(\frac{t-\tau}{a} \right) \right\} \frac{dadb}{a^2} \tag{10}$$

where $\Psi(t)$ is the mother wavelet, τ and a are the translation and scale parameters, respectively.

3.1.2 Discrete wavelets

The continuous wavelet transform is calculated by performing continuous translations and scalings of a function over a signal. In practice, this transformation is not feasible because it is necessary to make endless translations and scalings, requiring much time, effort and computational redundancy. Discrete wavelets were introduced to overcome this problem and therefore they are used in this work. They are not translated or scaled continuously but in discrete steps, which is achieved from a modification of the continuous wavelet:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{|s|}} \psi \left(\frac{t-\tau}{s} \right) \tag{11}$$

$$\psi_{j,k}(t) = \frac{1}{\sqrt{|s_0^j|}} \psi \left(\frac{t - k\tau_0 s_0^j}{s_0^j} \right) \tag{12}$$

where j and k are integers; $s_0 > 1$ is a fixed dilation parameter; τ_0 is the translation factor, which depends on the dilation factor.

Generally, it is chosen $s_0 = 2$ in order to have a sampling frequency called dyadic sampling $\tau_0 = 1$ is chosen for the temporal sampling, also dyadic This can be shown in (13) (Oliveira, 2007):

$$\psi_{j,k}(t) = \sqrt{2^j}\psi\left(2^j t - k\right) \tag{13}$$

When discrete wavelets are used to analyze a signal, the result is a series of wavelet coefficients, also called the series of wavelet decomposition (Oliveira, 2007). As a wavelet can be viewed as a bandpass filter, the scaled wavelet series can be seen as a set of bandpass filters, with a Q factor (set of filters fidelity factor). In practice, there is a discretized wavelet, with upper and lower limits for translations and scales. The wavelet discretization, associated with the idea of passing the signal through a filter set, results in the well known subband coding (Oliveira, 2007).

3.1.3 Multiresolution analysis

Mathematical transformations are used in a dataset to obtain additional information not available in the primitive data model. For example, it may be necessary to use a transformation that detects changes in color tones of a pixel neighborhood and its corresponding spatial location, and even that efficiently transposes these changes in a multiresolution space (Castleman, 1996). The multiresolution analysis using wavelet transforms has become increasingly popular with the release of JPEG-2000 standard (Weeks, 2007) and consists of a signal processing strategy where it is used a set of specialized filters to extract signal information, such as the range of frequencies present in it and their location as a function of the signal duration at different resolutions (Castleman, 1996). A brief description of the multiresolution analysis enables to display two functions responsible for generating the entire wavelet system: the scale function and the primary wavelet (or mother wavelet). The term mother comes from the fact that functions with different sizes are used in the process of transformation and all of them are originated from a specific or mother wavelet.

Scale function $\phi_{j,k}$ and the primary wavelets $\psi_{j,k}$ are considered orthogonal to follow the condition shown in (14):

$$\int_{-\infty}^{+\infty} \phi_{j,k}(x)\psi_{j,k}(x)dx = 0 \tag{14}$$

where $j \in Z$ corresponds to the scale function parameter and $k \in Z$ corresponds to the translation of $\frac{k}{2^j}$ in relation to the scale function and the primary wavelet, given by $j = 0$ and $k = 0$, respectively. Both scale and wavelet functions are defined in the set of real (**R**) numbers, by scalings and translations of the mentioned functions. The translation parameter corresponds to time information in the transform domain and the scaling parameter is the process of compression and expansion of the signal (Mallat, 2009). In Fig. 3 is shown an example of a scale function and a primary Haar wavelet.

Therefore, one can say that multiresolution analysis using discrete time wavelets corresponds to successive band-pass filtering, through which signals are decomposed at

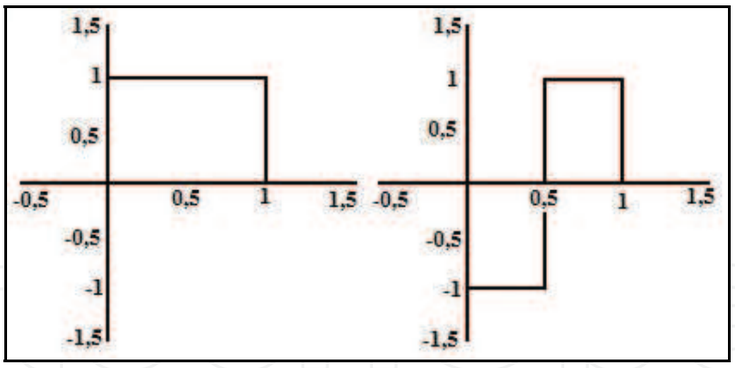


Fig. 3. Scale function and the primary Haar wavelet

each step, in terms of approximation and details. Fig. 4 illustrates this procedure being applied to an input image, where filters are used in successive rows and columns, creating the new scales. The reverse process, which performs the sum of subspaces, can reconstruct the original image. The dilation equations expressed by $h(k)$ represent the low-pass filters that generate the approximations of the original image. However, the translation equations $g(k)$ represent the high-pass filters and are responsible for obtaining the details of the original image. The decomposition of detail functions consists of details on vertical (high-pass filter at the rows and low-pass filters at the columns), details on horizontal (low-pass filter at the rows and high-pass filters at the columns), details on diagonal (high-pass filter at the rows and columns).

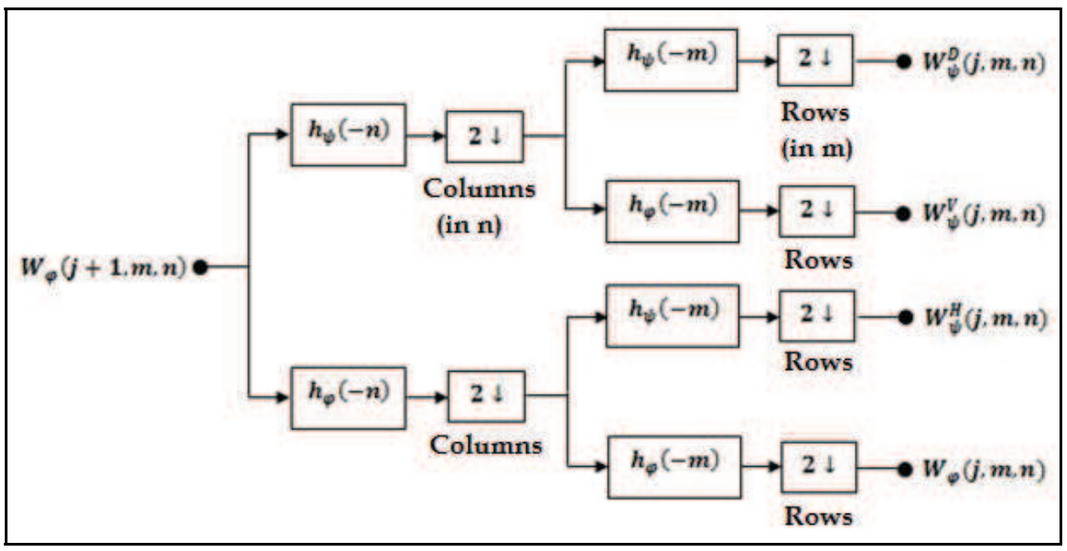


Fig. 4. Wavelet decomposition

Then, using an input image with scale of $(j+1)$, with m rows and n columns, it is shown in Fig. 5 a two level image decomposition example.

The original image and its approximations are the lighter areas of the picture. The other three remaining subpictures correspond to the three detail functions of the original image. In the second step, the lightest part of the picture is decomposed again, generating a new image approximation and three new detail subpictures. Thus, a twice smaller scale image

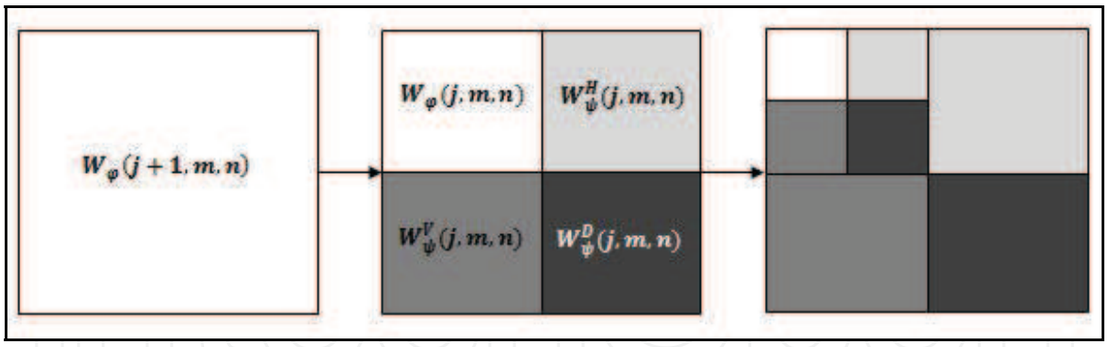


Fig. 5. Multiresolution representation of an image

was generated. Fig. 6 shows an initial image and the degree of refinement obtained with the wavelet transform. This form of decompose and reconstruct images can be implemented quickly and effectively, because of the use of wavelet transforms.

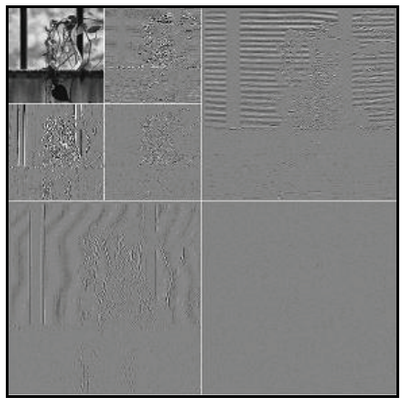


Fig. 6. Multiresolution analysis of an image

3.2 Wavelet pre-processing

Pre-processing step aims to improve the performance of the next steps. Then, the refinement of these data that will be used for training and classification are of fundamental importance. Designing a neural classifier consists of choosing architecture, training algorithm, quality of training dataset, among other aspects that must be optimized, in order to reduce the time used in network training at the same time that the accuracy of the results is increased. This optimization can be performed using wavelet transforms. The block diagram described in Fig. 7 shows that the data input used by the neural network may be submitted to a wavelet pre-processing or not. Here, both the situations are analyzed in parallel.

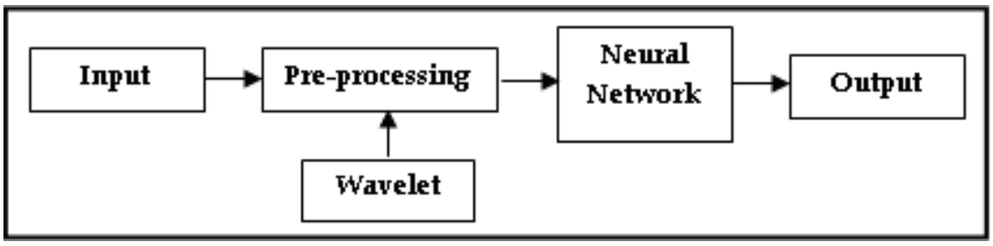


Fig. 7. Block diagram of a neural network system

3.3 Neural network implementation

A simple but effective neural classifier has been implemented at this step, using the classic aspects of this type of project. It is important to know that even robust classifiers as SVM (Support Vector Machine) can not function properly without a pre-processing. Fig. 8 shows a sample training set used in the experiment. This consists of 50 images containing the vowels A, E, I, O, U, with slight 2D dislocation and no noise.

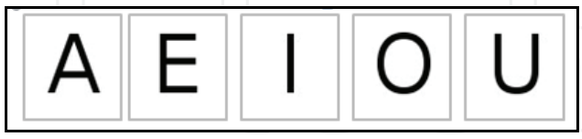


Fig. 8. Training dataset model

Fig. 9 shows a sample of the test dataset (composed by 25 images), where there is the inclusion of uncorrelated noise to images up to four levels, aiming to difficult the neural network generalization.

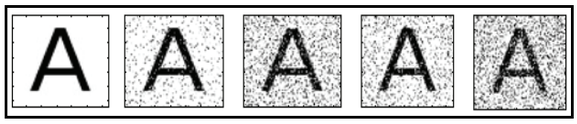


Fig. 9. Test dataset model

Here, the neural network configuration used the following parameters: a multilayernperceptron neural network; the Levenberg-Marquardt training algorithm; number of epochs (150) or Error (10e-5) as stop criteria; two hidden layers and the sigmoid tangent as activation function.
The results are shown in Table 01, according to two categories of pre-processing: without wavelet - larger training set (designed pixel by pixel) and higher time to neural network learning; with wavelet (Daubechies, level 2) - smaller and more efficient training dataset due to the choice of better wavelet descriptors (higher intensity), which provides a faster network learning. It is important to observe that the stop criterion was the number of epochs while the error was kept around 10e-4 and the generalization result was about 92%, which enabled to recognize images of the vowels a, e, i, o, u with small 2D dislocation levels (<5%) and noise uncorrelated (<50%).

PP	CT	TT (min)	Error (e-04)	GE
Without wavelet	4096x50	4.27	1.8120	94%
With wavelet	1000x50	1.91	7.4728	96%

Table 1. Results: PP - Pre-processing; TD – Training Dataset; TT –Time of Training; GE - Generalization

The MLP Neural Network was used efficiently in the presented pattern classification, generalizing in both cases above 94%. The extraction process of features was simplified and improved by the use of wavelet transform at the pre-processing step in order to optimize all

the classification step, since the time of training until the generalization of results. It is worth to say that not only the method, but also its ability of reducing the matrix of data, preserving the meaning of the information, was of extreme importance to reduce the computational cost more than twice.

4. Hybrid EM-optimization method for optimal design of FSS

4.1 Frequency selective surfaces

Frequency selective surface (FSS) (Wu, 1995; Munk, 2000; Campos, 2009; Cruz, 2009) is a two-dimensional periodic array of elements (patches or apertures) in a conducting screen, which could be either freestanding or supported by dielectric substrates, as shown in Fig. 10(a). FSSs are used for a variety of applications, such as: hybrid radomes (Cruz et al., 2010), polarizers and dichroic reflectors (Munk, 2000), waveguide filters (Monorchio, 2005), artificial magnetic conductors (Genovesi et al., 2006) and microwave absorbers (Liu et al., 2004). To provide high-performance filtering properties at microwave bands, electromagnetic engineers have investigated various types of FSS devices: reconfigurable FSS screens, fractal self-similar geometries (Cruz et al., 2009a), pixelized FSS screens, multilayered FSS structures, as well as FSSs on anisotropic substrates (Silva et al., 2007; Campos et al., 2001).

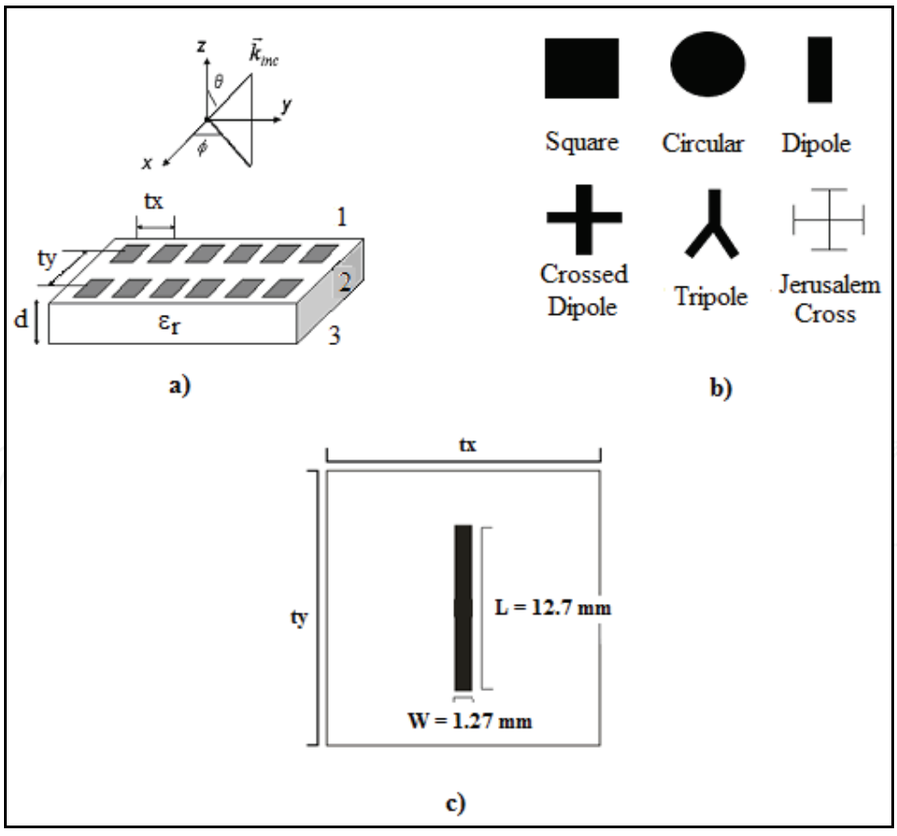


Fig. 10. (a) The conventional geometry of a FSS; (b) some patch element shapes; (c) thin dipole unit cell configuration

There is no closed form solution directly from a given desired frequency response to the corresponding FSS (Hwang et al, 1990). The analysis of scattering characteristics from FSS devices requires the application of rigorous full-wave techniques. Besides that, due to the computational complexity of using a full-wave simulator to evaluate the FSS scattering variables, many electromagnetic engineers still use a trial-and-error process until to achieve a given design criterion. Obviously, this procedure is very laborious and human dependent. Therefore, optimization techniques are required to design practical FSSs with desired filter specifications. Some authors have employed neural networks, PSO and GA for FSS design and optimization (Cruz et al., 2009b; Silva et al., 2010a; 2010b).

4.2 Natural optimization algorithms

Natural optimization algorithms are stochastic population-based global search methods that start with an initial population of candidate individuals for an optimal solution. Assuming an optimization problem with N_{var} input variables and N_{pop} individuals, the population at the kth iteration is a matrix $\boldsymbol{P}(k)_{Npop \times Nvar}$ of floating-point elements, denoted by $p_{m,n}^{k}$, with each row corresponding to an individual (Cruz, 2009; Silva et al., 2010b). Under GA and PSO jargons, for example, individuals are named *chromosomes* and *particles* (or agents), respectively. The implementation details of these algorithms are described.

4.2.1 Continuous Genetic Algorithm

Continuous genetic algorithm is very similar to the binary-GA but works with floating-point variables. Continuous-GA chromosomes are defined by (15), where each one corresponds to a vector with N_{var} floating-point optimization variables. Each chromosome is evaluated by means of its associated cost, which is computed through the cost function E given in (4) (Cruz, 2009; Silva et al., 2010b).

$$chromosome(k,m) = \left[p_{m,1}^{k}, \quad p_{m,2}^{k}, \quad \ldots, \quad p_{m,N\,\mathrm{var}}^{k} \right], \quad m = 1,2,\ldots,Npop \tag{15}$$

$$\cos t(k,m) = E\left(chromosome(k,m)\right) \tag{16}$$

Based on the cost associated to each chromosome, the population evolves through generations with the application of genetic operators, such as: selection, crossover and mutation. Fig. 11(a) gives a "big picture" overview of continuous-GA. The mating step includes roulette wheel selection presented in (Haupt & Werner, 2007; R. Haupt & S. Haupt, 2004). Population selection is performed after the N_{pop} chromosomes are ranked from lowest to highest costs. Then, the N_{keep} most-fit chromosomes are selected to form the mating pool and the rest are discarded to make room for the new offspring. Mothers and fathers pair in a random fashion through the blending crossover method (R. Haupt & S. Haupt, 2004). Each pair produces two offspring that contain traits from each parent. In addition, the parents survive to be part of the next generation. After mating, a fraction of chromosomes in the population will suffer mutation. Then, the chromosome variable selected for real-value mutation is added to a normally distributed random number (Cruz, 2009; Silva et al., 2010b).

4.2.2 Particle Swarm Optimization

Particle swarm optimization algorithm was first formulated by Kennedy and Eberhart (1995). The thought process behind the algorithm was inspired by the social behavior of animals, such as bird flocking or fish schooling. PSO algorithm is similar to continuous-GA since it begins

with a random population matrix. Unlike GA, PSO has no evolution operators such as crossover and mutation. Each particle moves along the cost surface with an individual velocity. A flow chart of the PSO algorithm is shown in Fig. 11(b). The implemented PSO algorithm updates the velocities and positions of the particles based on the best local and global solutions according to (17) and (18), respectively (Cruz, 2009; Silva et al., 2010b).

$$v_{m,n}^{k+1} = C\left[r_0 v_{m,n}^{k} + \Gamma_1 \cdot r_1 \cdot \left(p_{m,n}^{local\,bes(k)} - p_{m,n}^{k} \right) + \Gamma_2 \cdot r_2 \cdot \left(p_{m,n}^{global\,best(k)} - p_{m,n}^{k} \right) \right] \quad (17)$$

$$p_{m,n}^{k+1} = p_{m,n}^{k} + v_{m,n}^{k+1} \quad (18)$$

Here, $v_{m,n}$ is the particle velocity, pm,n is the particle variables, r_0, r_1 and r_2 are independent uniform random numbers; Γ_1 and Γ_2 are is the cognitive and social parameters, respectively, $p_{m,n}^{localbest(k)}$ and $p_{m,n}^{globalbest(k)}$ are the best local and global solutions, respectively; C is the constriction parameter (Kennedy & Eberhart, 1995). If the best local solution has a cost less than the best cost of the current global solution, then the best local solution replaces the best global solution. PSO is a very simple natural optimization algorithm, easy to implement and with few parameters to adjust.

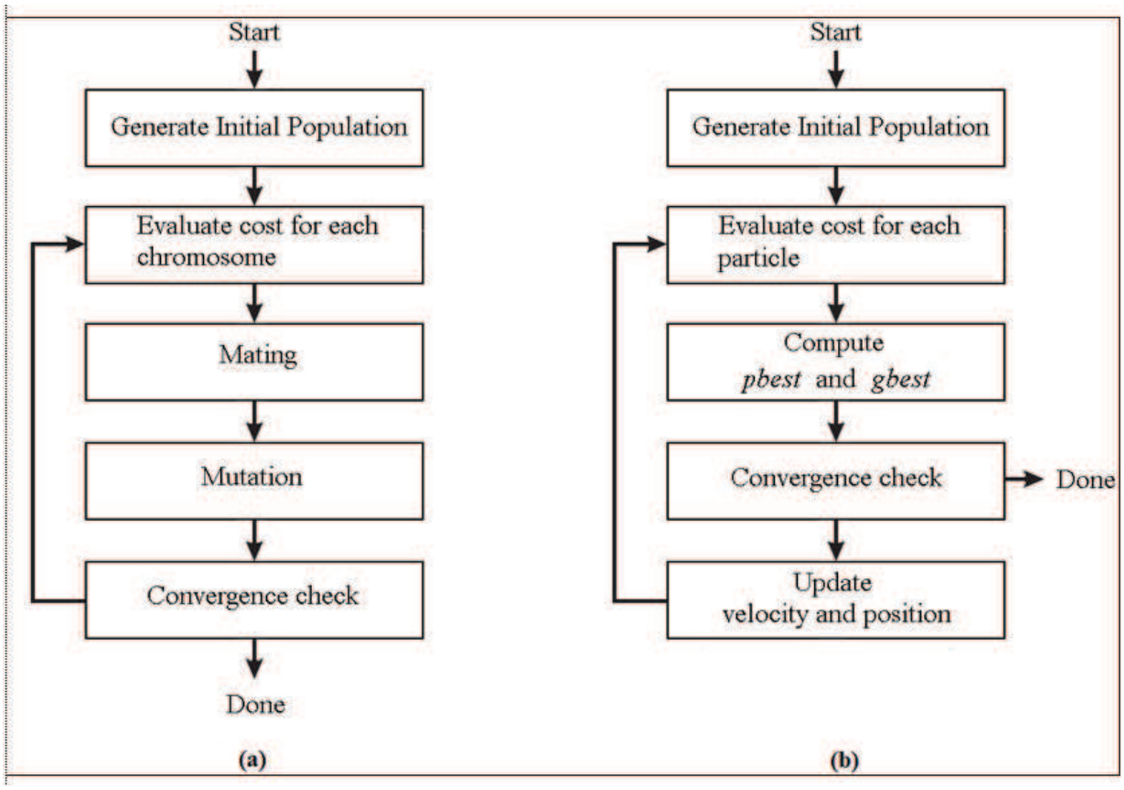


Fig. 11. Flow charts of the natural optimization algorithms: (a) GA; (b) PSO

4.3 EM-optimization method

The FSS design methodology, which blends MLP models, GA and PSO algorithms, is applied for a thin dipole FSS synthesis, considering electromagnetic parameters such as

resonant frequency (f_r) and –3dB bandwidth (BW) as a function of the substrate thickness (d) and the periodicity of elements ($t = t_x = t_y$) corresponding to a square unit cell. The dimensions of the thin dipole elements used in this work are shown in Fig. 10(c). The width (W) and length (L) of the patch remain the same. The region of interest (or search space) defined by desired input variables is a rectangle: $15.24 \le t \le 19.05$ mm and $0.1 \le d \le 2.0$ mm.
The FSS optimization problem consists of obtaining an optimal solution (d^*; t^*) in the search space that results in a minimal thickness FSS with the desired specifications for resonant frequency and bandwidth. After the definition of the FSS filter-type, design input variables and search space, a parametric analysis is performed in order to observe the FSS EM-behavior. At this step, the FSS transmission coefficients at 6.0–14.0 GHz band were obtained by means of Method of Moments (MoM) simulations considering substrate anisotropy (Campos et al., 2001). Representative EM-datasets must be obtained from the parametric analysis for supervised learning of FSS synthesis MLP models.
The second step consists of modeling f_r and BW responses by the synthesis MLP network using the conventional neuromodeling technique (Zhang & Gupta, 2000). The third step just consists of implementing the natural optimization algorithms for FSS optimal synthesis. In particular, the continuous-GA and PSO algorithms were used.
Through the FSS neuromodeling, we avoid the very intensive application of the CPU for MoM analysis in GA and PSO simulations. The ANN model was developed for FSS synthesis using one hidden layer MLP configuration. The MLP learning processes were carried out using the RPROP training algorithm, with standard parameters (Riedmiller & Braun, 1993), in order to minimize the mean square error as defined in (4). In this case, the RPROP training is performed until the mean square error reaches a minimum pre-established value. The synthesis MLP network configuration (composed by three inputs: -1, d, t; twenty hidden neurons and two outputs: f_r, BW) simultaneously approaches both the mappings $f_r(d,t)$ and $BW(d,t)$. The MLP configuration is similar to that presented in Fig. 1(b). To generate the synthesis training dataset were assumed the vectors given in (19) for the design input variables d and t, adding up 72 training examples.

$$\begin{cases} d = [0.1 \quad 0.3 \quad 0.5 \quad 0.7 \quad 0.9 \quad 1.0 \quad 1.2 \quad 1.4 \quad 1.5 \quad 1.6 \quad 1.8 \quad 2.0] \text{ mm} \\ tx = ty = t = [15.24 \quad 15.87 \quad 16.51 \quad 17.14 \quad 18.41 \quad 19.05] \text{ mm} \end{cases} \tag{19}$$

GA and PSO algorithms were implemented in Matlab® for optimal FSS synthesis. The assumed design input variables (substrate thickness and periodicity) were symbolized by d_m^k and t_m^k to take into account the m-th individual at the k-th iteration. Given a desired FSS filter specification, ($fr_{desired}$, $BW_{desired}$), the aim is the minimization of the quadratic cost function as defined in (20) in terms of absolute percent errors.

$$\cos t(k,m) = \left(\frac{\left| fr_{desired} - fr\left(d_m^k, t_m^k\right) \right|}{f_{desired}} + \frac{\left| BW_{desired} - BW\left(d_m^k, t_m^k\right) \right|}{BW_{desired}} \right)^2, \tag{20}$$

$$m=1,2,\ldots Npop$$

To evaluate the cost function shown in (20), the synthesis MLP approach the EM relations $f_r\left(d_m^k, t_m^k\right)$ and $BW\left(d_m^k, t_m^k\right)$. When the population evolves, each individual is constrained to the region of interest using (21), where the dummy variable ξ can be replaced by d or t.

$$\xi_m^k = \min\left(\max\left(\xi_m^k, \xi_{\min}\right), \xi_{\max}\right) \tag{21}$$

The parameters used for continuous-GA and PSO simulations are shown in Table 2.

Continuous-GA	PSO
initial population = 50	initial population = 25
maximum iteration number (generations) = 500	maximum iteration number = 200
crossover probability = 0.9	constriction parameter C = 0.8
	cognitive parameter, $\Gamma_1 = 2$
mutation rate = 0.01	cognitive parameter, $\Gamma_2 = 2$

Table 2. GA and PSO parameters used in the simulations

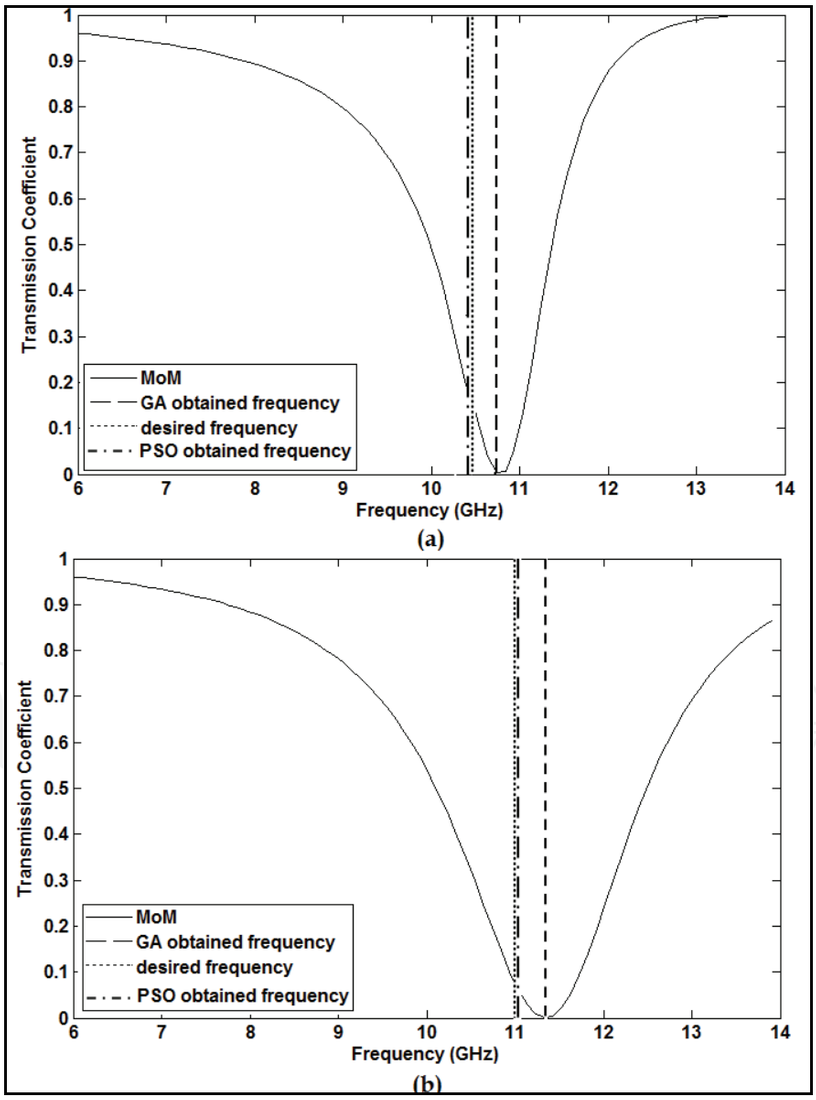


Fig. 12. Tranmission coefficients of the thin dipole FSS: (a) first and (b) second examples

In this section, examples that use the hybrid EM-optimization technique for the optimal synthesis of FSSs are presented. Random values for the resonant frequency and bandwidth were chosen within the search space used for the MoM simulations of the synthesis MLP network. The first example corresponds to output data (f_r; BW) = (10.5; 1.5) GHz and the second example corresponds to output data (f_r; BW) = (11.0; 2.5) GHz.
Figs. 12(a) and 12(b) show the transmission coefficients obtained from MoM, GA and PSO simulations for the first and second examples, respectively.
It is observed that although GA algorithm response is closer to MoM simulations, PSO algorithm presents the best solutions of resonant frequency and bandwidth, compared to the desired values. Table 3 shows the input/output values obtained from both algorithms.
The cost surface contours, the initial, intermediate, and final populations, as well as the best path for both algorithms can be found in (Silva et al., 2010a; 2010b). One can also find a new version of GA algorithm called improved GA in (Cruz, 2009).

First example investigated - (f_r; BW) = (10.5; 1.5) GHz		
Parameters	**Results**	
	GA	**PSO**
Thickness of the substrate d (mm)	0.7685	0.5401
Periodicity of elements t (mm)	18.193	15.283
obtained f_r (GHz)	10.7407	10.4917
obtained BW (GHz)	1.3827	1.8632
Second example investigated - (f_r; BW) = (11.0; 2.5) GHz		
Parameters	**Results**	
	GA	**PSO**
Thickness of the substrate d (mm)	0.195	1.9457
Periodicity of elements t (mm)	15.826	19.006
obtained f_r (GHz)	11.4444	11.1121
obtained BW (GHz)	2.2716	2.2924

Table 3. Input/output values obtained from GA and PSO algorithms

5. Hybrid and high processing pattern classification

In recent years, literature (Dieterrich, 1998) indicated some development directions in machine learning which have a highlight in this research scene, including the committee machines and also methods for scaling up supervised learning algorithms. Ensembles studies, which deal with aggregation among many learning machines on the resolution of others complex problems, had an increased effort with empiric data, formalization and new methods (Valentini & Masulli, 2002; Kuncheva, 2004; Dimitrakakis & Bengio, 2005). New training methods, such as hierarchy and parallel processing, have been developed and evaluated in several aplications.
This section describes a pattern classification problem that deal with electric power line systems. The quality of the energy provided by an electric system is one of the greatest points of interest for concessionaires and electric energy consumers. Literature presents distinct approches in the acquisition, characterization and classification of disturbs that may occur in high potential transmission nets. Some pre-processing methods, like wavelet transform, are used for characterization of voltage or current signals.

Considering high performance and hybrid architectures neural networks as a develoment pespective, this work treats a learning algorithm for extended modular neural networks. The motivation is the unsatisfactory results in pattern classifications and function approximations commonly used (Jacobs et al, 1991). In such cases, modular neural experts are composed by a single linear neuron layer and training process is performed through associative gaussian mixture models.
The architecture proposed for solution of this problem uses multilayer preceptrons experts, as shown in Fig. 1(c). The modular neural network has K experts, L layers of q neurons with linear or nonlinear activation function and bias, as well as a gating network with similar architecture.

5.1 The hybrid algorithm

The modified modular artificial neural network is trained with a developed algorithm adapted according to (Jacobs et al., 1991), for the model of Gaussian mixing associative and also the error backpropagation algorithm, by including the calculation of the descend gradient. The main steps of this algorithm is in this section; more details can be found in (Magalhães et al., 2008).
In this description, some conventions are utilized: the stucture has MLP experts (indexed by $i = 1, ..., k$) with L^{Esp_i} layers (indexed by $l = 1, ..., L^{Esp_i}$), with $q^{(l)Esp_i}$ neurons in each layer (indexed by $j = 1, ..., q^{(l)Esp_i}$). Also, there is a MLP gating network with L^{Pas} layers (indexed by $l = 1, ..., L^{Pas}$), with $q^{(l)Pas}$ neurons in each layer (indexed by $j = 1, ..., q^{(l)Pas}$). The neurons activation functions in all networks can be linear or non-linear. The processing algorithm can be divided in three steps.

5.1.1 First step

The first step of the algorithm is the calculation of *a priori* probability associated with the i-th output layer of the gating network, as expressed by (22):

$$g_i(n) = \frac{\exp\left(u_i^{(l)}(n)\right)}{\sum_{j=1}^{K} \exp\left(u_i^{(l)}(n)\right)} \tag{22}$$

where $u_i^{(l)}(n)$ is the i-th output neuron of the l-th layer of the gating network and is giving by:

$$u_i^l(n) = \varphi_{pi}^{(l)}\left(v_{pi}^{(l)}(n)\right) \tag{23}$$

where $\varphi_{pi}^{(i)}$ is the derivative function from the activation potential $\upsilon_{pi}^{(i)}$, from the i-th neuron of the l-th layer of the gating network, as shown in (24):

$$v_{pi}^{(l)}(n) = \sum_{j=0}^{q^{(l)Pas}} a_{ij}^{(l)} u_i^{l-1}(n) \tag{24}$$

where a_{ij} is the weight associated to the j-th input neuron i of the l-th layer of the gating network.

5.1.2 Second step

The second step of the algorithm is responsible for evaluating the values of *posteriori* probabilities $h_i(n)$, associated to the i-th output neuron of the output gating network. The *posteriori* probabilities can be expressed as:

$$h_i(n) = \frac{g_i(n)\exp(\frac{-1}{2}||d(n) - y_{e_i^{(K)}}(n)||^2)}{\sum_{j=1}^{K} g_j(n)\exp(\frac{-1}{2}||d(n) - y_{e_i^{(K)}}(n)||^2)} \tag{25}$$

where $d(n)$ is the desired response and $y_{ei}{}^{(k)}(n)$ is the actual response of the i-th neuron of the l-th layer of the K-th expert, as shown in (26):

$$y_{e_i^{(K)}}^{(l)}(n) = \varphi_{e_i^{(K)}}^{(l)}(v_{e_i^{(K)}}^{(l)}(n)) \tag{26}$$

where $\varphi(l)$ is the activation potential derivative $v(l)$ from the i-th neuron, l-th layer and k-th expert.

5.1.3 Third step

The principal point in this algorithm is the incrementation in synaptic weights of the modular network, that is done with the multiple layers. The synaptic weights from the networks experts are updated according to:

$$w_{e_i^{(K)}j}^{(l)}(n+1) = w_{e_i^{(K)}j}^{(l)}(n) + \eta\delta_{e_i^{(K)}}^{(l)}(n)y_j^{(l-1)}(n) \tag{27}$$

where η is the learning rate, and the gradient $\delta_{ei^{(k)}}^{(i)}$ for the output layer neurons is obtained by:

$$\delta_{e_i^{(K)}}^{(l)}(n) = h_i(n)e_i(n)^{(l)}\varphi_{e_i^{(K)}}^{'(l)}(v_{e_i^{(K)}}^{(l)}(n)) \tag{28}$$

where e_i is the difference between d_i and $y_e(K)$. The gradient for the neurons of hidden layers is computed as:

$$\delta_{e_i^{(K)}}^{(l)}(n) = \varphi_{e_i^{(K)}}^{'(l)}(v_{e_i^{(K)}}^{(l)}(n))\sum_{m=1}^{q^{(l+1)Esp_i}}\delta_m^{(l+1)}(n)w_{me_i^{(K)}}^{(l+1)}(n) \tag{29}$$

The synaptic weight update of the gating network is accomplished according to (30):

$$a_{p_i j}^{(l)}(n+1) = a_{p_i j}^{(l)}(n) + \eta\delta_{p_i}^{(l)}(n)y_j^{(l-1)}(n) \tag{30}$$

where the output layer gradient is obtained by:

$$\delta_{pi}^{(l)}(n) = [h_i(n) - g_i(n)]\varphi_{pi}^{'(l)}(v_{pi}^{(l)}(n)) \tag{31}$$

The error is the difference between h_i and g_i. The gradient of hidden layers is calculated by:

$$\delta_{pi}^{(l)}(n) = \varphi_{pi}^{'(l)}(v^{(l)} \sum_{m=1}^{q^{(l+1)p_{as}}} \delta_{m}^{(l+1)}(n) a_{mpi}^{(l+1)}(n)) \quad (32)$$

Thus, the error backpropagates from the gating network to its hidden layers.

5.2 The application and results

As mentioned before, the application implemented was a pattern classification for power line disturbances. The implementation methodology, data files and information were the same presented in (Medeiros et al., 2007). The application evaluated the performance of an intelligent system classifier, in this case, a modular artificial neural network, in electric disturbances classification. The approach is performed into four main steps: getting the signal, pre-processing, definition and classification of descriptors.

First, the electrical signals were obtained using the oscillograph network of São Francisco Hydro Electric Company (CHESF) and also from the simulation via Transient Alternative Program (ATP). The electric network consists of 370 oscillographs operation with a sampling rate ranging between 20 and 256 samples/cycle. Here, the signals were collected in voltage line levels of 69, 230 and 500 kV, with a rate of 128 samples/cycle during 14 cycles. The pre-processing stage aimed to suggest descriptors that characterize the signal variations when diverted from a certain standard. The third step, which deals with the descriptor definitions, is performed by the decomposition of signals from the previous step. Once obtained the descriptors, four disturbances classes were defined as: Voltage Sag, Voltage Swell, Harmonics and Transitories. The last step (classification) is performed by the application of classifiers based on modular artificial neural networks. Several different architectures were tested, as shown in Table 4.

Net	MOD-0	MOD-1	MOD-2	MOD-3
Number of Experts	3	3	3	3
Expert Architecture	10:3:4	10:5:4	10:10:4	10:15:4
Gating Architecture	10:5:4	10:5:4	10:10:4	10:15:4
Classification (%)	98,46	99,48	100	100

Table 4. Modular Neural Network Architecture

The classification step used two datasets, one for the training at computer, and the other for validation at multiprocessor on chip, utilizing a FPGA system architecture. The training set consisted of 800 patterns formed by 4 output classes. To validate the modular neural network, 344 input patterns were used with their respective expected responses, consisting only of data obtained from the oscillographs.

Table 4 shows that the modular neural network with the proposed algorithm reaches a high amount of accuracy, approximately 100%. Besides, the utilization of two processors on a FPGA system increased the optimization speed-up by 47%, while using four processors, the speed-up was increased by 87%.

6. Conclusion

This chapter described some of the neuromodeling methodology used for applications in various areas of Engineering, in particular, EM-optimization, Signal Processing and Pattern Classification and Recognition. Some original contributions were shown, such as the hybrid EM-optimization for optimal design of FSS, the artificial neural network optimization using wavelet transforms and the modular neural network used to pattern classification and recognition.

The choice of the activation function of a FNN or a modular neural network strongly influences the neural model performance. However, there is not an universal activation function that can be used to solve any kind of nonlinear modeling problem. Using additional information from the hybrid neural models, as well as sharing the training process with modular neural networks, increased the efficiency and therefore the generalization ability and model reliability of the resulting neural models.

The fast and accurate obtained results for all the applications demonstrated the improvements with the utilization of these models, and proved that the MLP network global approximations are able to generalize. In addition, the idea of blending artificial neural networks and natural optimization algorithms, as well as mathematical transforms (for EM-optimization and pattern recognition, respectively) shows to be very interesting. PSO algorithm needs less individuals and reaches the best solution at a few iterations, compared to GA algorithm. The PSO implementation is simpler than GA, since it is not required the presence of genetic operatiors such as crossover and mutation. Therefore, PSO algorithm has proved to be an interesting EM-optimization tool, with few parameters to adjust and low computational cost.

The characteristics of ANN models (precision, CPU efficiency and flexibility) can be perfectly used in association with these optimization techniques in order to develop powerful soft computing tools. A good point in combining these complementary tools is the possibilty of multiprocessing application, using parallel processors and computers, not only to increase performance and execution speed, but also to enable an improved type of competition and collaboration. Each individual tool allows solving the problem or part of the problem and, at the same time, they can collaborate one each other to improve the solution in a higher level. This is the intelligent computing that this chapter wants to apply in efficient engineering.

7. References

Beale, E. M. L. (1972). A derivation of conjugate gradients, In: *Numerical Methods for Nonlinear Optimization,* F. A. Lootsma, (Ed.), Academic Press, London.

Bhatia, P.; Bharti, S. & Gupta, R. (2009). Comparative analysis of image compression techniques: a case study on medical images, In: *International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 820-822.

Campos, A. L. P. de S. (2009). *Superfícies seletivas em frequência: análise e projeto,* IFRN, 198 p., Natal, RN, Brazil.

Campos, A. L. P. S.; D'Assunção, A. G. & Melo, M. A. B. (2001). Investigation of scattering parameters for frequency selective surfaces using thin dipoles on anisotropic layers. *Proceedings of the SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference,* Vol. 1, pp. 413-416.

Castleman, K. R. (1996). *Digital Imaging Processing.* Prentice Hall, 1st Ed, Englewood Cliffs, New Jersey.

Cruz, R. M. S. (2009). Análise e otimização de superfícies seletivas de frequência utilizando redes neurais artificiais e algoritmos de otimização natural, Ph.D. Dissertation, Federal University of Rio Grande do Norte, Natal, RN, Brazil.

Cruz, R. M. S.; Santos, A. F. & D'Assunção, A. G. (2010). Spectral analysis of electromagnetic waves in uniaxial anisotropic dielectric materials, Proceedings of 14º SBMO – Simpósio Brasileiro de Microondas e Optoeletrônica and 9º CBMag – Congresso Brasileiro de Eletromagnetismo, MOMAG2010, pp. 440-444, Vila Velha, ES, Brazil.

Cruz, R. M. S.; Silva, P. H. da F. & D'Assunção, A. G. (2009a). Neuromodeling stop-band properties of Koch island patch elements for FSS filter design. In: *Microwave and Optical Technology Letters,* Vol. 51, No. 12, pp. 3014-3019.

_____. (2009b). Synthesis of crossed dipole frequency selective surfaces using genetic algorithms and artificial neural networks, *Proceedings of the 2009 International Joint Conference on Neural Networks*, pp. 627-633, ISBN 978-1-4244-3553-1, Atlanta, GA, USA.

Dietterich, T. G. (1998). Machine-learning research: four current directions, *The AI Magazine*, Vol. 4, pp. 97–136.

Dimitrakakis, C. & Bengio, S. (2005). Online adaptive policies for ensemble classifiers. *Neurocomputing*, Vol. 64, pp. 211-221.

Genovesi, S. *et al.* (2006). Particle swarm optimization of frequency selective surfaces for the design of artificial magnetic conductors, In: *IEEE Antennas and Propagation Society International Symposium*, Vol. 9, No. 14, pp. 3519–3522.

Hagan, M. T. & Menhaj, M. (1999). Training feed-foward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*. Vol. 5, No. 6, pp. 989-993.

Haupt, R. L. & Haupt, S. (2004). *Practical genetic algorithms,* Wiley & Sons, Inc., 2nd Ed, New Jersey.

Haupt, R. L. & Werner, D. H. (2007). *Genetic algorithms in electromagnetic,* John Willey & Sons, Inc., 1st Ed, New York.

Hwang, J. N.; Chan, C. H. & Marks, R. J. (1990). Frequency selective surface design based on iterative inversion of neural networks. *Proceedings of the International Joint Conference on Neural Networks*, Vol. 1, pp. 39-44.

Haykin, S. (1999). *Neural networks: a comprehensive foundation,* Prentice-Hall, 2nd Ed, Upper Saddle River, New Jersey.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J. & Hinton, G. E. (1991). Adaptive mixture of local experts. *Neural Computation,* MIT Press, Vol. 3, pp. 79-87.

Kennedy, L. & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IV IEEE Conference On Neural Networks*, Piscataway, New Jersey.

Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*, John Wiley & Sons, Inc., New Jersey.

Liu, J. C. *et al.* (2004). Implementation of broadband microwave absorber using FSS screens coated with Ba(MnTi)$Fe_{10}O_{19}$ ferrite, In: *Microwave and Optical Technology Letters*, Vol. 41, No. 4, pp. 323–326.

Lopes, D. C. et al. (2009). Implementation of a Modular Neural Network in a Multiple Processor System on FPGA to Classify Electric Disturbance. In: *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, Porto, Portugal.

Magalhães, R. M. et al. (2008). Application of a Hybrid Algorithm in the Modular Neural Nets Trainning with Multilayers Specialists in Electric Disturbance Classication. In: *INES 2008 12th International Conference on Intelligent Engineering Systems,* Miami, Florida.

Mallat, S. (2009). *A Wavelet Tour of Signal Processing*, Elsevier Inc, 3rd ed.

Masnadi-Shirazi, R.; Mohseni, A. & Sheikhi, M. A. (2009). Efficient compression of wavelet packet OFDM radar signals.

McCculloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133.

Medeiros, M. F. et al. (2007). Influence of signal processing in the Efficiency of algorithms Based on Neural Networks for Disturbanc Classification, *Computacional Intelligence in Image and Signal Processing*, pp. 95 -100.

Monorchio, A. *et al.* (2005). Design of waveguide filters by using genetically optimized frequency selective surfaces, In: *IEEE Microwave Wireless Component Letters*, Vol. 15, No. 6, pp. 407–409.

Munk, B. A. (2000). *Frequency selective surfaces - theory and design*, John Wiley & Sons, Inc., ISBN 0-471-37047-9, New York, USA.

Oliveira, H. M. (2007). Análise de Fourier e Wavelets: sinais estacionários e não estacionários. Ed. Universítária UFPE, 1ª Ed, Recife.

Ottensamer, J.; Bobin, J. & Starck, R. (2008). Compressed sensing in astronomy, In: *IEEE Journal of Selected Topics in Signal Processing*, Vol. 2, No. 5, pp. 718-726.

Parolai, J. J.; Galiana-Merino, J. L. & Rosa-Herranz, S. (2008). Seismic P phase picking using a kurtosis-based criterion in the stationary wavelet domain, In: *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 46, No. 11, pp. 3815-3826.

Peixoto, H. M.; Cruz, R. M. S.; Guerreiro, A. M. G.; Dória Neto, A. D. & D'Assunção, A. G. (2009a). A comparison of multilayer perceptrons training algorithms for the optimization of frequency selective surfaces, *Proceedings of 8th International Information and Telecommunication Technologies Symposium*, Florianópolis, SC, Brazil.

Peixoto, H. M.; Guerreiro, A. M. G. & Dória Neto, A. D. (2009b). Image processing for eye detection and classification of the gaze direction, *Proceedings of the 2009 International Joint Conference on Neural Networks*, pp. 2475-2480, ISBN 978-1-4244-3553-1, Atlanta, GA, USA.

Peixoto, H. M.; Rego, J. B. A.; Guerreiro, A. M. G. & Dória Neto, A. D. (2009c). Otimizando redes neurais artificiais com transformadas *wavelets*, *Anais do IX Congresso Brasileiro de Redes Neurais – Inteligência Computacional*, Ouro Preto, MG, Brazil.

Pinto, L. F. Q.; Silveira, L. G. Q. S.; Junior, F. M. & Assis, E. L. (2009). Analysis and optimization of wavelet-coded communication systems, Vol. 8, No. 2, pp. 563-657.

Rahmat-Samii, Y. (2003). Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in Engineering Electromagnetics, *Proceedings of 17TH International Conference on Applied Electromagnetics and Communications*, Dubrovnik, Croatia.

Riedmiller, M. & Braun, H. (1993). A direct adaptative method for faster backpropagation learning: the RPROP algorithm. *Proceedings of IEEE International Conference on Neural Networks*, pp. 586–591, San Francisco, USA.

Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, Vol. 65, No. 6, pp. 386-408. Retrieved September 19, 2010, from

http://www.ling.upenn.edu/courses/Fall_2007/cogs501/Rosenblatt1958.pdf.

Rumelhart, D. E.; Hinton, G. E. & Williams, R. J. (1986). Learning internal representations by error propagation. In: *Parallel Distributed Processing: Explorations in the microstructure of cognition,* D. E. Rumelhart & J. L. McClelland (Ed.), MIT Press, Vol. 1, pp. 318-362, Cambridge, MA.

Silva, P. H. da F. & Campos, A. L. P. S. (2008). Fast and accurate modeling of frequency selective surfaces using a new modular neural network configuration of multilayer perceptrons, *Microwaves, Antennas e Propagation,* IET, Vol. 2, No. 5, pp. 503–511.

Silva, P. H. da F.; Cruz, R. M. S & D'Assunção, A. G. (2010a). Blending PSO and ANN for optimal design of FSS filters with Koch island patch elements, *IEEE Transactions on Magnetics*, Vol. 46, No. 8.

_____. (2010b). Neuromodeling and natural optimization of nonlinear devices and circuits, In: *System and Circuit Design for Biologically-Inspired Intelligent Learning*, Turgay Temel (Ed.), IGI Global, ISBN 9781609600181, Hershey PA. No prelo.

Silva, P. H. da F.; Lacouth, P.; Fontgalland, G.; Campos, A. L. P. S. & D'Assunção, A. G. (2007). Design of frequency selective surfaces using a novel MoM-ANN-GA technique. *Proceedings of the SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference,* Vol. 1, pp. 275-279.

Valentini, G. & Masulli, F. (2002). Ensemble of learning machines, in Neural Nets WIRN Vietri-02, *Series Lectures Notes in Computer Sciences*, Springer-Verlag.

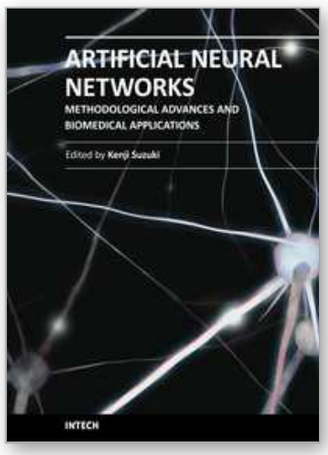
Vimal, P. S.; Kumar, R. & Arumugabathan, C. (2009). Wavelet based ocular artifact removal from EEG signals using arma method and adaptive filtering, pp. 667-671.

Wang, X. Q. G.; Liu, D. G. Z.; Li, Z. & Wang, H. (2009). Object categorization using hierarchical wavelet packet texture descriptors, In: *11th IEEE International Symposium on Multimedia*, pp. 44-51.

Weeks, M. (2007). *Digital Signal Processing Using MATLAB and Wavelets*, Infinity Science Press LLC Hingham, Georgia State University, Massachusetts.

Wu, T. K. (1995). *Frequency selective surface and grid array*, John Wiley & Sons, Inc., ISBN 0-471-31189-8, New York, USA.

Zhang, Q. J. & Gupta, C. (2000). *Neural networks for RF and microwaves design,* Artech House, Norwood, MA.



Artificial Neural Networks - Methodological Advances and Biomedical Applications

Edited by Prof. Kenji Suzuki

ISBN 978-953-307-243-2
Hard cover, 362 pages
Publisher InTech
Published online 11, April, 2011
Published in print edition April, 2011

Artificial neural networks may probably be the single most successful technology in the last two decades which has been widely used in a large variety of applications in various areas. The purpose of this book is to provide recent advances of artificial neural networks in biomedical applications. The book begins with fundamentals of artificial neural networks, which cover an introduction, design, and optimization. Advanced architectures for biomedical applications, which offer improved performance and desirable properties, follow. Parts continue with biological applications such as gene, plant biology, and stem cell, medical applications such as skin diseases, sclerosis, anesthesia, and physiotherapy, and clinical and other applications such as clinical outcome, telecare, and pre-med student failure prediction. Thus, this book will be a fundamental source of recent advances and applications of artificial neural networks in biomedical areas. The target audience includes professors and students in engineering and medical schools, researchers and engineers in biomedical industries, medical doctors, and healthcare professionals.

How to reference

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

Rossana M. S. Cruz, Helton M. Peixoto and Rafael M. Magalhães (2011). Artificial Neural Networks and Efficient Optimization Techniques for Applications in Engineering, Artificial Neural Networks - Methodological Advances and Biomedical Applications, Prof. Kenji Suzuki (Ed.), ISBN: 978-953-307-243-2, InTech, Available from: http://www.intechopen.com/books/artificial-neural-networks-methodological-advances-and-biomedical-applications/artificial-neural-networks-and-efficient-optimization-techniques-for-applications-in-engineering

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

IntechOpen