Łukasz SOROKOSZ, Włodzimierz ZIENIUTYCZ

Gdańsk University of Technology

Artificial Neural Networks in Microwave Components and Circuits Modeling

***Abstract**. This paper reviews the idea of implementing the artificial neural networks (ANN) in microwave components and circuits design and optimization. The principles and basic assumptions of conventional ANN modeling are presented. Capabilities of this method are drawn on the base of exemplary project of the microstrip patch antenna. In addition, various strategies extending the potential of ANN modeling by exploiting the microwave knowledge are presented. Additionally, commonly used training techniques are reviewed with regards to the microwave filter modeling.*

Streszczenie.** Artykuł dotyczy wykorzystania sztucznych sieci neuronowych (SNN) w projektowaniu i optymalizacji układów mikrofalowych. Zaprezentowano podstawowe zasady i założenia modelowania z użyciem SNN. Możliwości opisywanej metody opisano wykorzystując przykładowy projekt anteny łatowej. Przedstawiono różne strategie modelowania układów, które wykorzystują możliwości opisywanej metody w połączeniu z wiedzą mikrofalową. Porównano również dokładność powszechnie stosowanych metod uczenia SNN w oparciu o projekty filtrów mikrofalowych.* ***(Sztuczne sieci neuronowe w modelowaniu obowdów mikrofalowych)

Keywords: artificial neural networks, microwave devices modeling, optimization methods.
Słowa kluczowe: sztuczne sieci neuronowe, modelowanie układów mikrofalowych, metody optymalizacjne.

Introduction

Artificial neural networks (ANNs) are in fact information (signal) processing systems, whose design and principle of the operation are based on the studies of the human brain. Generally, ANNs can be learned/trained to model multidimensional nonlinear input-output relationship from given data [1],[2]. This interesting feature results in number of applications for neural networks such as image or speech recognition and matching, control systems, data grouping/filtering etc.

History of ANNs starts in 1940s, yet in the 1980s, with growth of computational power, they began to be applied in various engineering problems (e.g. biomedical, electrical). In the last two decades many neural network applications in microwave engineering have been reported. One of the reasons for which ANNs were applied in this field was the ability to model poorly known physical processes in some specific structures (neural networks are trained to model the behavior of the circuit). The other, more general reason was to accelerate the optimization and modeling of microwave circuits. Since full-wave modeling with computer-aided-design (CAD) tools requires a lot of time- and computational power consuming simulations, there was a need for some approximated methods of modeling.

ANNs have been successfully used in microwave devices (circuits) modeling, such as passive microwave structures (e.g. filters [3],[4]), transistors [5], amplifiers [6], etc. Except developing microwave devices, this technique is also being used in optimization processes, where combined with simulation tools saves time needed for many model reevaluations.

In this paper, we present a review of concepts of ANN oriented on microwave components and circuits design. In section II we describe ANN fundamentals and the most commonly used ANN structure - multilayer perceptrons (MLP). Whole conventional neural modeling process along with the results obtained for an exemplary structure developed with this technique are presented in section III. Next section contains the description of chosen, innovative microwave knowledge based modeling strategies introduced to the conventional ANN approach. In this section, additionally, we also show the comparison of different ANN training techniques. Finally some concluding remarks are drawn.

Neural networks - principles

General concept

The basic neural networks structure consists of two kind of components: neurons - the processing elements and interconnections (synapses, links). Each link in the network is described by the weight parameter. Neurons can be classified into 3 groups: input, output and hidden neurons. Input neurons receive and process the signal from outside the networks, output neurons produce the outcoming information (result) and neurons whose inputs and outputs are connected to other neurons are called hidden neurons.

Knowing the components of the network, we try to build a neural model of the structure/device which we focus on. Let us call the input vector $\boldsymbol{x}$ (of $\boldsymbol{N}$ elements), desired output vector $\boldsymbol{y}$ (of $\boldsymbol{M}$ elements) and links weight vector $\boldsymbol{w}$ (trained network will model the $\boldsymbol{x}$ - $\boldsymbol{y}$ relationship in terms of $\boldsymbol{w}$ vector). Next, a set of exemplary x - y pairs (training data) need to be gathered. In terms of microwave engineering they can be obtained either from the known formulas calculations, electromagnetic simulations or measurement. For example, in CAD modeling problem, the input vector might consist of circuits physical dimensions and corresponding output vector - its scattering parameters.

Next step is network training. During this process weights of the neural network are being adjusted in order to produce outputs ($\boldsymbol{d}$) as close as possible to training data outputs ($\boldsymbol{y}$). The difference between the $\boldsymbol{d}$ and $\boldsymbol{y}$, called training error, is calculated. After each cycle the network weights are being updated in order to minimize the training error.

The aim of the training process is to teach the network to produce valid response for inputs from outside the training data - this ability is called generalization. Trained neural network has fixed weight vector and can be validated with a set of completely new x - y pairs. Error between obtained and desired output, called generalization error, is then calculated.

Neural network - structure

The most popular structure of neural networks is multilayer perceptron (MLP), where neurons are grouped into 3 kinds of layers. First (input) layer consist of input neurons, likewise the output layer contains the output neurons. Remaining layers are called hidden layers. The size of MLP depends on the number of hidden layers. The scheme of MLP3 (one hidden layer) neural network is presented in the Fig. 1. Theoretically, three layer MLP can approximate any nonlinear continuous multidimensional function [2].

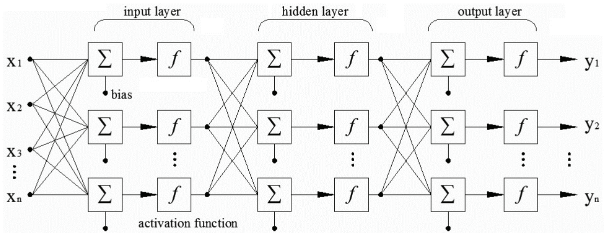

Fig.1. Functional MLP3 neural network structure

Let the total number of layers be L and the number of neurons in l^{th} layer - N_l $(l=1,2, ..., L)$. Let assume that output of l-1^{th} layer is connected with the input of l^{th} layer. Consequently, the weight of the link between j^{th} neuron of l-1^{th} layer with i^{th} neuron of l^{th} layer is represented by w^l_{ij}. Weight vector also contains the bias of each neuron w^l_{i0}. The weight vector parameters are initialized before training and during that process they are updated.

Neurons process incoming information in two steps. Firstly, inputs are multiplied by the adequate weights, including the bias, next to be added. Secondly, the produced sum is processed by the activation function. Mostly used activation functions is sigmoid functions. Among them the commonly used are arc-tangent and hyperbolic-tangent (tansig) functions (Fig. 2 (a)). Linear activation function are applicable in microwave design, where modeled parameters are continuous (e.g. purelin, see Fig. 2 (b)).

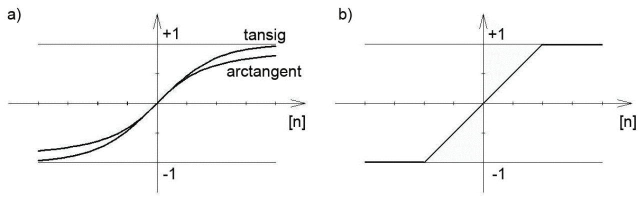

Fig.2. Commonly used activation functions (a) arctangent and tansig, b) purelin)

Network size is crucial in terms of learning accuracy. The number of neurons in hidden layer depends on the nonlinearity of the x - y relationship. High nonlinearity constrains usage of grater number of hidden neurons. Still, there is no specified way of defining the appropriate network size. Number of hidden neurons depends on the engineer and his experience or on the results of the 'cut-and-try' process.

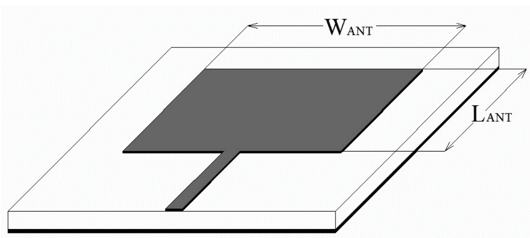

Fig.3. Layout of the microstrip line feed patch antenna

Neural network modeling - the process

Let us describe the neural modeling process on the example of a rectangular patch antenna [7]. In this case, the input vector contains all the parameters necessary to design the radiator: dielectric constant ε_r, substrate thickness h and antenna operation band frequencies f_l and f_h. The output vector consists of patch dimensions: the length L_{ANT} and the width W_{ANT}. Layout of the microstrip line feed patch antenna is presented in the Fig. 3.

Data generation

First step in developing the neural model is to formulate the problem and define input and desired outputs of neural network. For different type of modeled devices different type of outputs can be selected. Two important factor must be therefore taken into consideration: i) complexity of data generation (this step is repeated many times), ii) ease of implementing the chosen outputs in the neural network structure.

Since the outputs are established, we have to define the range of input data (training and testing data). The range of generated training data should be slightly wider than the range of data which the model would be designed to work with. For the considered rectangular patch antenna the developed neural model was designed to produce patch dimensions W_{ANT} and L_{ANT}, having ε_r, h and f_l and f_h as an inputs. The range of inputs for training data and for the modeling problem are gathered in the table 1.

Table 1. Range of data for rectangular patch antenna modeling [7]

parameter	training data range	model working range
ε_r	1.5 - 3.5	2.0 - 3.0
h	1.0 - 2.5	1.0 - 1.9
f_l	obtained from IE3D	6.1 - 7.9
f_h		6.5 - 9.1
L_{ANT}	11.0 - 13.5	modeled
W_{ANT}	13.0 - 17.0	

Operation band frequencies f_l and f_h for training data were calculated with IE3D simulating software, whereas patch dimensions are the subject of neural modeling. We can now, define desired output vector as $\boldsymbol{d} = [L_{ANT}, W_{ANT}]$ corresponding to the input vector $\boldsymbol{x} = [\varepsilon_r, h, f_l, f_h]$.

Together, 245 $(\boldsymbol{x}, \boldsymbol{d})$ pairs were prepared, out of which 230 were used for network training and the rest for testing the trained neural network. The authors of the paper used typical MLP3 structure with 4 input, 2 output and 40 hidden neurons. Used neural network architecture is feedforward network, which means that output of input layer is used to feed the second layer. For this problem, the authors proposed tansig and purelin activation functions (Fig. 2).

ANN Training

Having all the needed data prepared, the training process can be initialized. First, the weight vector w are set to some arbitrary values. Commonly, the weights are initialized with small random values. Other strategies can be found in the literature, e.g. range of values inversely proportional to the square root of number of links for each neuron or different ranges along with different random values distributions.

Next, we define the training error E_k for one input vector $\boldsymbol{x_k}$ as:

$$E_k(\boldsymbol{w}) = \frac{1}{2}\sum_{j=1}^{M}\left(d_{jk} - y_j(\boldsymbol{x}_k, \boldsymbol{w})\right)^2 \qquad (1)$$

where d_{jk} is j^{th} element of d_k vector and y_j $(\boldsymbol{x_k}, \boldsymbol{w})$ is j^{th} output of the neural network. Total training error is given by

$$E_{train}(\boldsymbol{w}) = \sum_{k=1}^{K} E_k(\boldsymbol{w}) \qquad (2)$$

MOST WIEDZY Pobrano z mostwiedzy.pl

where K is the length of the training data set. Minimizing the training error function $E_{train}(\boldsymbol{w})$ is the goal of the neural network training process. Because this function is nonlinear, the weight vector w is often adjusted using iterative algorithms. Most popular gradient techniques involves updating the w basing on the value of $E_{train}(\boldsymbol{w})$ function and its derivative $\partial E_{train}/\partial \boldsymbol{w}$. Then the $p+1$ step weight vector w_{p+1} is calculated as: $w_{p+1} = w_p + \eta \boldsymbol{h}$, where $\boldsymbol{h}$ is called direction vector and η is a positive step size (learning rate).

The most popular training algorithm, called backpropagation method (BP) [8], is based on updating the w along negative direction of the calculated gradient, $\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \eta(\partial Etrain/\partial \boldsymbol{w})$. Other gradient-based techniques will be described in the following section. In the literature, the methods of finding the global optimal ANN weight parameters, e.g. genetic algorithm, are also reported [9]. Still, this approach is more time consuming in terms of neural network training.

Gradient-based techniques require calculating the error function derivative. Standard approach is called error backpropapagation (EBP) [5]. Let us go back to (1) and define error between ith neural network output and corresponding training data output:

$$\delta_i = d_{ik} - y_i(\boldsymbol{x}_{k,}\boldsymbol{w}). \tag{3}$$

This error can be backpropagated from output layer into the previous (hidden) layer. We calculate the local error for i^{th} neuron in the l^{th} layer as:

$$\delta_i^l = \left(\sum_{j=1}^{N_{l+1}} \delta_j^{l+1} w_{ji}^{l+1} \right) \cdot z_i^l \cdot \left(1 - z_i^l\right), \quad l = L-1,\ L-2, \ldots 3,2 \tag{4}$$

where z_i^l is the output of the i^{th} neuron in the l^{th} layer and N_{l+1} is the number of neurons in $l+1^{th}$ layer. For the last, we can calculate the desired derivatives, with respect to (1):

$$\frac{\partial E_{train}}{\partial w_{ji}^l} = \sum_{k=1}^{K} \frac{\partial E_k}{\partial w_{ji}^l} = \sum_{k=1}^{K} \left(\delta_i^l \cdot z_j^{l-1} \right), \quad l = L-1,\ L-2, \ldots 3,2 \tag{5}$$

Last step in neural network training is to validate the quality of the trained network. For that matter, mentioned testing data inputs (independent from the training data) are introduced into the neural network. Obtained error between desired output and produced result is now called test error and it is one of the networks quality measures. If the test error fulfils the design assumption, the process is terminated and the weight vector is fixed.

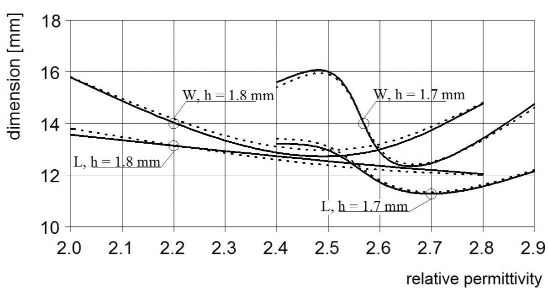

Fig.4. Comparison of results of IE3D (dotted lines) and ANN (solid lines); on the base of results from [7]

Now the designed network can be stimulated with inputs from outside of the training or testing set. The networks ability to produce the appropriate answer for those kind of input signals is called generalization. The principle of the ANN training is to determine the networks parameters such that it will provide effectively small generalization error.

To summarize this section, we present the result of testing the neural network designed to model the rectangular patch antenna resonating around $7\ GHz$ that was mentioned before. Patch dimensions were calculated using ANN and IE3D simulation software. Results are presented for substrate thicknesses $h = 1.7\ mm$ and $1.8\ mm$ in the Fig. 4.

Fig. 4 indicates satisfactory level of agreement of the results for different permittivity of the substrate. That effect proves that neural network model can effectively predict dimensions of the patch antenna, while the inputs of the network are in the range of training data.

Microwave knowledge based strategies and training techniques

Neural modeling strategies

Conventional neural modeling, presented in the previous sections, despite its many advantages has also a few drawbacks. First is the time and effort required to gather training and testing data, especially if we want to generate data from measurement. Secondly, neural networks find the solution lying only in the training region. Finally, number of training samples grows exponentially with the desired smoothness of the approximated function. Those shortcomings are important from the engineer point of view. Still, they can be improved by combining the ANN and the microwave knowledge. Popular methods, frequently described in the literature:

- Neural Space Mapping (NSM) [10],
- Knowledge Based Neural Networks (KBNN) [11],
- Difference Method [12][13],

are described below in more details.

Neural Space Mapping (NSM)

Neural SM optimization is a combination of classic SM method and elements of ANN modeling. Firstly, the coarse model solution x^*_c that produces the desired response R^* is found by optimizing the coarse model (term from SM theory, circuit model). We treat this optimized model as a fine model ($x^*_c = x_f$). The fine model response R_f (SM theory, full-wave electromagnetic model) is calculated. If it is sufficiently close to R^* (from the designers points of view) the procedure is terminated.

If not, the $Bp = 2n$ (n is the number of design parameters) points around x_f are chosen and they form the ANN training set. Neural network is trained to minimize the error defined as the difference between its response and the response R_f. Neural model is then used as a starting point in new optimization process which effect is a new fine model (x_f). The termination condition is evaluated ($R_f \approx R^*$). If its response R_f is still not close enough to R^*, the Bp is incremented by one and the procedure is repeated.

This method allows to optimize only some design parameters which implies different possibilities of defining the neural network model. In the literature we can find an exemplary application of NSM method in microwave filter modeling [10].

Knowledge based Neural Networks (KBNN)

This approach is based on implementing the microwave empirical or analytical knowledge (e.g. empirical formulas) into the neural network. KBNN do not have the conventional MLP structure, empirical functions are used as neurons activation functions. Those formulas can be adjusted in the learning process, where weight vector $\boldsymbol{w}$, additionally contains their adjustable parameters. This method

MOST WIEDZY Pobrano z mostwiedzy.pl

employed to model a simple FET consisting of three coupled transmission lines is reported in [11].

Difference method (DM)

This method, also called hybrid EM-ANN method, assumes training the neural network with the difference between accurate data (e.g. measurement or full-wave simulation) and rough data (e.g. equivalent circuit). Neural network trained in that matter can be combined with the rough model for standard optimization. Depending on the rough model accuracy, this kind of approach might help to reduce the number of accurate data needed for the proper ANN training. Still, as shown in [14], for high-temperature superconducting (HTS) filter modeling the difference between rough and accurate data might take as much time and effort as direct accurate data modeling.

Exploring the literature we see that other methods, also oriented on microwave engineering, are developed

(e.g. Neural Inverse Space Mapping [14], Prior Knowledge Input Method [13] and knowledge based automatic model generation (KAMG) [15]).

Training techniques

Despite the modeling strategies, presented above, user can also chose from variety of different ANN training techniques. Training algorithms, other than presented BP, that are commonly used in microwave neural modeling, are: scaled conjugate gradient method [16], Levenberg-Marquardt method [17], Bayesian Regularization [18], Newton and quasi-Newton method [17], etc. We will describe the principles of most commonly used strategies and present the comparison of their efficiencies basing on microwave filter design [19]. See the cited literature for detailed explanations of the algorithms.

Scaled Conjugate Gradient method (SCG)

This method is based on general BP methods, yet the quadratic minimization is introduced for more careful calculations of the search direction and the step size. The error function $E_q(s)$, calculated in the surrounding s around the w_p (point in the weight space $\boldsymbol{w}$), is defined as

$$E_q(s) = E(w_p) + E'(w_p) \cdot s + \frac{1}{2} s \cdot E''(w_p) \quad (6)$$

In order to find the minimum of the function $E_q(s)$ - the critical points for $E_q(s)$ need to be determined. They can be found as a solution of the equation:

$$E_q'(s) = E''(w_p) \cdot s + E'(w_p) = 0. \quad (7)$$

Critical points are not necessarily a minimum, it can a saddle or a maximum. Generally, conjugate gradient methods show linear convergence on the most of optimization problems. Scaled CG is faster since the step size scaling omits line sample per sample search during the learning process [20].

Levenberg-Marquardt method (LM)

This algorithm, like the quasi-Newton, is based on approximating the Hessian matrix, instead of calculating the exact $\boldsymbol{H}$ matrix. This approximation is defined as:

$$\boldsymbol{H}^* = \boldsymbol{J}^T \boldsymbol{J} \quad (8)$$

and the gradient is represented by:

$$\boldsymbol{g} = \boldsymbol{J}^T \boldsymbol{E}_k(\boldsymbol{w}) \quad (9)$$

where $\boldsymbol{J}$ is a matrix containing the error first derivatives, called Jacobian matrix. Calculation of the Jacobian, especially for large problems, is less time and effort consuming than calculating the Hessian matrix. Once having the Jacobian matrix computed, we can state the learning update as:

$$\boldsymbol{w}_{p+1} = \boldsymbol{w}_p - \left(\boldsymbol{J}^T \boldsymbol{J} - \mu \boldsymbol{I}\right)^{-1} \boldsymbol{J}^T \boldsymbol{E}_k(\boldsymbol{w}_p) \quad (10)$$

where μ is a scalar that is designed to adjust the algorithm speed and direction.

Since the Newton's method is faster and more accurate around the minimums, the goal is to shift torwards this method as quickly as possible (μ equal to zero). For large μ the gradient step is being reduced after every methods step (unless the error increases).

Bayesian Regularization (BR)

In this method, in contrast to conventional approach, the goal of the training is to minimize the objective function F defined as follows:

$$F = \alpha \cdot E_W + \beta \cdot E_D \quad (11)$$

where E_W is the sum of squares of network weights, E_D is the sum of squared errors and α, β are the F function parameters. The α to β ratio determines the goal of the training (e.g. for $\alpha >> \beta$ the weights are reduced, the networks error occurs, yet the response is smoother).

The regularization problem is to find the optimal values of α and β parameters. The optimization process is iterative and employs the Bayes rules and one step of Levenberg-Marquardt algorithm [17]. Main advantage of this approach is the significant reduction of the generalization error.

Training techniques - comparison

Described ANN training techniques were compared in terms of microwave filters modeling [19]. In the Fig. 5(a) and Fig. 5(b) we present the values of errors for coefficients of Chebyshev (n=11, 0.01 dB ripple) and Butterworth (n=19) filters, respectively. Presented errors were calculated with respect to the filter coefficients obtained with conventional filter design method.

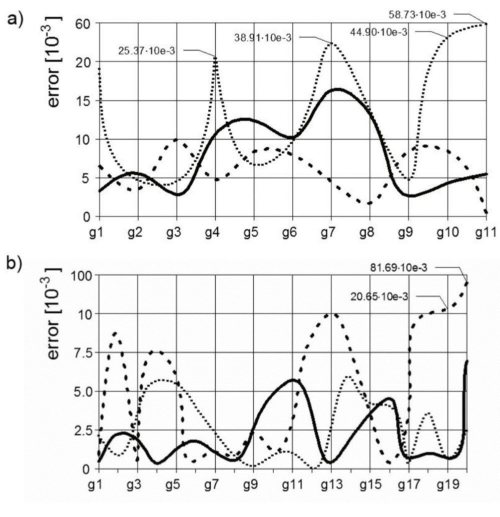

Fig.5. Comparison of error in ANN modeled filter coefficients (a) Chebyshev Filter (n=19, 0.01 dB ripple), (b) Butterworth Filter (n=11); solid line - LM, dashed line - SCG, dotted line - BR; on the base of results from [19]

MOST WIEDZY Pobrano z mostwiedzy.pl

Analyzing the Fig. 5, we can observe that different training methods imply different error, depending on the problem. For Chebyshev filter the highest error is obtained with Scaled CG method, whereas for Butterworth filter it is BR. This effect can be caused by different neural networks for both filters. ANN for Chebyshev filter has two inputs: order and ripple, in contrast for Butterworth filter the only input is the filter order. It is worth mentioning that in both cases LM method allows to achieve acceptable level of coefficient values error.

Conclusion

This paper raises the issue of implementing the artificial neural networks in microwave circuits design and optimization. The aim of this work was to present the idea and the principles of this method, from the engineers point of view. Conventional neural networks modeling is presented basing on the exemplary project of a microstrip patch antenna. Some drawback of the ANN based modeling are stated. Various strategies of improving the ANN approach, employing the microwave knowledge (NSM, KBNN, DM) are reviewed. For last the most popular neural network training techniques are described and compared in terms of microwave filters design.

Presented strategy of employing the neural network algorithms in microwave engineering is used nowadays. In the future there could be drawn new applications for the ANNs. Since it has been successfully implemented in solving either transient or frequency domain problems, there might a need for ANN-based integrated transient-frequency domain problems. Innovative strategies of exploiting of the microwave knowledge merged with ANNs are still developed, for example KAMG.

This work was supported by the Polish Ministry of Science and Higher Education from sources for science partially under COST Action IC0803 (decision No 618/N-COST/09/2010/0) and partially under funding for Statutory Activities for ETI Gdansk University of Technology.

REFERENCES

[1] Żurada J., Barski M., Jędruch W., *Sztuczne Sieci Neuronowe*, PWN, Warsaw (Poland) 1996.

[2] Zhang Q.J., Gupta K.C., *Neural Networks for RF and Microwave Design*, Artech House, USA 2000

[3] Xiaofei Q., Xianliang W., A Design of Dual-Stop Band DGS Filter Based on ANN, *4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008. WiCOM '08.

[4] Michalski J.J., Artificial Neural Networks Approach in Microwave Filter Tuning, *Progress In Electromagnetics Research M*, vol. 13, 2010, 173-188

[5] Zhang Q.J., Gupta K.C., Devabhaktuni V.K., Artificial Neural Networks for RF and Microwave Design - From Theory to Practice, *IEEE Trans. Microwave Theory Tech.*, vol. 51, no. 4, April 2003, 1339-1350

[6] Xu J. and the others, Neural-based Dynamic Modeling of Nonlinear Microwave Circuits, *IEEE Trans. Microwave Theory Tech.*, vol. 5, December 2002, 2769-2780

[7] Thakare V.V., Singhal P., Naural Network Based CAD Model for the Design of Rectangular Patch Antenna, J*ournal of Engineering and Technology Research*, vol. 2(7), July 2010, 126-129

[8] Wang F. and the others, Neural Network Structure and Training Algorithms For Microwave Applications, *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 9, 1999, 216-240

[9] Sexton R.s., Dorsey R.E., Johnson J.D., Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing, *European Journal of Operational Research*, vol. 114(3), May 1999, 589-601

[10] Bakr M.H. and the others, Neural Space-Mapping Optimization for EM-Based Design, *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 12, December 2000, 2307-2315

[11] Gaoua S. and the others, CAD Tools for Efficient RF/Microwave Transistor Modeling and Circuit Design, *Analog Integrated Circuits And Signal Processing*, vol. 63, no. 1, 2010, 59-70

[12] Watson P.M., Gupta K.C., EM-ANN Models For Microstrip Vias And Interconnects In Dataset Circuits, *IEEE Trans. Microwave Theory Tech.*, vol. 44, no. 12, December 1996, 2495-2503

[13] Watson P.M., Gupta K.C., Mahajan R.L., Development of Knowledge Based Artificial Neural Network Models for Microwave Components, *1998 IEEE MTT-S International Microwave Symposium Digest*, vol. 1, 1998, 9-12

[14] Bandler J.W. and the others, Neural Inverse Space Mapping EM-Optimization, *2001 IEEE MTT-S International Microwave Symposium Digest*, vol. 2, 2001, 1007-1010

[15] Devabhaktuni and the others, Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks, and space mapping, *IEEE Trans. Microwave Theory Tech.*, vol. 51(7), July 2003, 1822-1833

[16] Orozco J., Garcia C.A.R., Detecting Pathologies from Infant Cry Applying Scaled Conjugate Gradient Neural Networks, *Proceedings of European Symposium on Artificial Neural Networks*, 23-25 April 2003, Burges (Belgium), 349-354

[17] The MathWorks Inc., MATLAB - product documentation

[18] Foresee F.D., Hagan M.T., Gauss-Newton Approximation to Bayesian Learning, *International Conference on Neural Networks* [1997], vol. 3, 1930-1935

[19] Raghavan S., Ilango, C., Artificial neural network modeling of microwave filters, *Proceedings of the 9th International Conference on ElectroMagnetic Interference and Compatibility* 2006, 561-564

[20] Moller, A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning, *Neural Networks J.*, vol. 6(4), 1993, 525-533

***Authors**: M.Sc. Łukasz Sorokosz, Ph.D. M.Eng. with habilitation Włodzimierz Zieniutycz, Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Dept. of Microwave and Antenna Engineering, Narutowicza Street 11/12, 80-233 Gdańsk, e-mail: lukasz.sorokosz@eti.pg.gda.pl*

MOST WIEDZY Pobrano z mostwiedzy.pl