

Research Article

Adaptive Forming of the Beam Pattern of Microstrip Antenna with the Use of an Artificial Neural Network

Janusz Dudczyk and Adam Kawalec

Institute of Radioelectronics, Faculty of Electronics, Military University of Technology, 00-908 Warsaw, Poland

Correspondence should be addressed to Janusz Dudczyk, jdudczyk@wp.pl

Received 16 January 2012; Accepted 2 September 2012

Academic Editor: Dalia N. Elshiekh

Copyright © 2012 J. Dudczyk and A. Kawalec. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Microstrip antenna has been recently one of the most innovative fields of antenna techniques. The main advantage of such an antenna is the simplicity of its production, little weight, a narrow profile, and easiness of integration of the radiating elements with the net of generators power systems. As a result of using arrays consisting of microstrip antennas; it is possible to decrease the size and weight and also to reduce the costs of components production as well as whole application systems. This paper presents possibilities of using artificial neural networks (ANNs) in the process of forming a beam from radiating complex microstrip antenna. Algorithms which base on artificial neural networks use high parallelism of actions which results in considerable acceleration of the process of forming the antenna pattern. The appropriate selection of learning constants makes it possible to get theoretically a solution which will be close to the real time. This paper presents the training neural network algorithm with the selection of optimal network structure. The analysis above was made in case of following the emission source, setting to zero the pattern of direction of expecting interference, and following emission source compared with two constant interferences. Computer simulation was made in MATLAB environment on the basis of Flex Tool, a programme which creates artificial neural networks.

1. Introduction

The problem of forming the beam pattern of radiating antenna arrays may be come down to the problem of precise updating the amplitude and phases of signals coming from every element of an antenna array before adding them up [1, 2]. There are some well-known ways of calculating weights which are based on maximization of the quotient of the desired signal to the interfering signal (Applebaum array, Shor array) or the conception of minimal mean (LMS array) [3, 4]. LMS and Applebaum arrays are able to follow in time several desired signals. Those can maximize the initial signal/interference quotient and control the shape of the beam pattern in order to set to zero the pattern on the direction of interfering signals [5]. The values of optimal weights are defined in such a way to provide minimal energy function, the definition of which is characteristic for particular algorithms. Artificial neural network both one-way, and recurrent ones in their work do the minimization of the energy function accompanied by a particular network.

The energy function which is defined on the stage of designing the structure of a network has a support role, but it is not the subject in this paper. The main aim of this paper is defining the vector of optimal weights. It can be also said that algorithms basing on ANN use high parallelism of actions which accelerate the realization of the process described above. With the use of appropriate learning constants theoretically it is possible to gain in time a solution close to the real time.

2. Microstrip Phased Arrays

Antenna phased arrays are constructed as conventional antenna arrays, the only difference between them is their architecture in which phase shifting modules are used. After entering, induced signals on particular radiators are added up in order to form the initial signal. The direction on which maximal pattern is expected is controlled by phase shift regulation between particular elements of an array [6, 7]. The beam pattern of an antenna linear array according to

the characteristic multiply factor is defined as relation (1), where $F_e(\theta, \phi)$ is the beam pattern of the radiation of a singular element of the array, and $F(\theta)$ is the pattern of the isotropic elements radiation system:

$$F(\theta, \phi) = F_e(\theta, \phi) \cdot F(\theta). \quad (1)$$

From the relation above, it can be said that mainly the beam pattern $F(\theta)$, called the system multiplier, determines the shape of a radiation system. The phasal shift of input signals on particular antennas is set in a way that maximum pattern covers with the planned direction of the desired emission source. A simplified block schema of the adaptation antenna was presented on Figure 1. The process of adding up the signals from particular elements is called the process of forming an antenna beam. The direction, on which the pattern of radiation array reaches its maximum, is adjusted to the desired signal. Steering the antenna beam can be made by appropriate delay of signals before connecting into an output signal as a result of the use of phase shifters. Elasticity, with which weights can be adjusted, implicates a crucial feature which can be used to eliminate the direction of the signal with a frequency identical to this of the expected signal. Removing one redundant signal by controlling the zero of the antenna beam pattern causes the consumption of one freedom degree. To limit setting to zero, only to redundant signals, the desired signal has to be paired with a steering vector, and then the weights calculation has to be done by using particular algorithms. The information about the steering vector is necessary as it protects the signal from being removed. Setting to zeros the antenna system beam pattern on the directions, on which there are emission sources, is done through weights calculation. These weights are represented by complex numbers which are assigned to particular elements of antenna arrays. The processes described above can be realized with the use of an ANN. A prototype of an artificial neural network is the biological nerve system with the structure consisting of nerve cells (neurons) with particular connections. Originally, artificial neural networks were an attempt of modelling mechanisms due to which nerve cells function in further perspective; they are quite useful while constructing artificial intelligence as it is similar in its structure to the human brain [8, 9]. Works on ANN can be divided into a few periods [10]. Early works concerning the first theoretical model of McCulloch and Pitts neuron [11] and later Hebb's work [12]. Further special interest in this subject is connected generally with Rosenblatt works [13] and Widrow's and Hoff's works [14]. Later low interest in this subject was caused by some limits of the single-layer perceptron which were presented in Minsky and Papert's work [15]. Only Hopfield's work [16] from 1982 and later works of different researchers concerning the backpropagation algorithm caused some increase in interest in artificial neural networks, which is still seen today.

Backpropagation is a famous training method used in the multilayer networks, but it often suffers from a local minima problem. To avoid this problem, a new backpropagation training based on chaos was proposed in [17]. The algorithm shown in [17] is comparable with the Levenberg-Marquardt

algorithm, but simpler and easier to implement comparing to Levenberg-Marquardt algorithm. Also neural networks and support vector machines (SVMs), idea of learning, fuzzy logic (FL), and data mining (DM) with computational intelligence are shown in [18, 19].

A single neuron consists of a kernel, dendrites which are entries, and an axon which is a single entry to the nerve cell. At the end of dendrites and an axon there are synapses which conduct information. The axon of a particular neuron is connected by synapses with dendrites of other neurons. Input signals x_i , $i = 1, 2, \dots, N$ have binary values depending on the appearance of an input impulse in the moment k . Defining the output signal by y , the rule of a single neuron activation can be written with respect to the following relation (2), where $k = 0, 1, 2, \dots$ are later moments of time, while w_i is multiplicative weight assigned to the connection of the entry i with the membrane of a neuron [11] as follows:

$$y^{k+1} = \begin{cases} 1, & \text{when } \sum_{i=1}^N w_i x_i^k \geq T \\ 0, & \text{when } \sum_{i=1}^N w_i x_i^k < T. \end{cases} \quad (2)$$

Generally it is assumed that between moments k and $k + 1$, the individual time of delay is passing for excitatory synapses $w_i = +1$ and for inhibitory synapses $w_i = -1$, where T is the threshold value determining the neuron's reaction. As McCulloch-Pitts model has several crucial simplifications (only binary operations, discrete work time, synchronization of all neurons work in one network, and unchanging thresholds and weights), a general description of an artificial neuron is introduced. Thus, now more complex models of a neuron are used to realize a neural network. Each neuron consists of a transforming element connected with synaptic entries and one entry according to the model Figure 2.

The transfer of signals in all connections is one way, and the output signal y of a neuron is defined with respect to following relation (3), where \mathbf{w} is a weights vector, while \mathbf{x} is the input vector defined by relations (4):

$$y = f(\mathbf{w}^T \mathbf{x}) = f\left(\sum_{i=0}^N w_i x_i\right), \quad (3)$$

$$\mathbf{w} \stackrel{\text{def}}{=} [w_0, w_1, \dots, w_N]^T, \quad (4)$$

$$\mathbf{x} \stackrel{\text{def}}{=} [x_0, x_1, x_2, \dots, x_N]^T.$$

The function $f(\mathbf{w}^T \mathbf{x})$, whose domain is the set of all activation link connections of a neuron, is called the "activation function." Defining activation with the symbol net, the function can be defined as $f(\text{net})$, where the variable net is defined by the dot product of weights and input signals vector $\text{net} = \mathbf{w}^T \mathbf{x}$. Whole activation net is an equivalent of the activation potential in the biological neuron. Individual classes of neurons differ in the function of definition $f(\text{net})$. Due to (3), defining the output signal y appears the fact that firstly weighted input signals are added up in order to define the whole activation net. Then, the operation $f(\text{net})$ is done according to its own activation function. The most popular

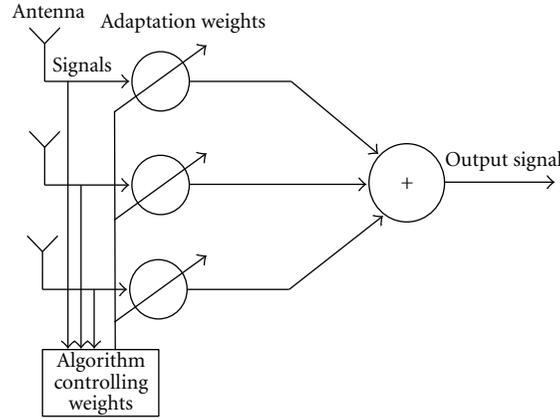


FIGURE 1: A simplified block schema of adaptation antenna system.

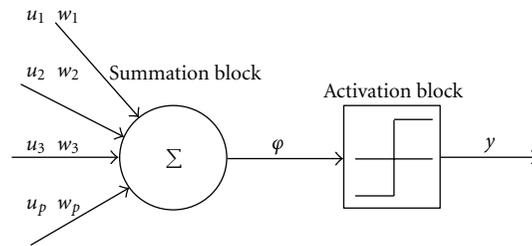


FIGURE 2: General model of an artificial neuron.

type of a neural network is one-way network example of which is a simple perceptron. As there are no connections between elements of the output layer, each element can be treated as a separate network with $N + 1$ inputs and one output. The perceptron network can be divided precisely into ordered classes and separated classes of elements called layers within which there is an input layer, an output layer, and hidden layers. Among layers of neurons which make up together a multilayered perceptron, the input layer with a linear activation function can be distinguished. The number of elements of this layer is precisely determined by the number of input data taken into account while doing the task of forming the microstrip antenna pattern.

3. Implementation of Artificial Neural Network Process

In the presented research the tool used to model the work of artificial neural networks was the modular programme Flex Tool. The implementation of the ANN process had several stages. Firstly, initial working conditions of the artificial neural network were defined and self-learning pairs were prepared. Secondly, the architecture optimization of this network was done. Later on, the process of learning and testing were conducted. What was next was defining initial conditions that concern aspects in which the ANN work will work. It was assumed that the network should form the beam pattern of a linear array consisting of six elements simultaneously following the emission source (ES) while

interference. The second condition was setting to zero the beam pattern on the direction of interference which changes its location. The second stage was preparing self-learning set. data that was used in the process of teaching the ANN was calculated for the array above ($n = 6$) with the use of the LMS algorithm. The algorithm used estimates the correlation vector \mathbf{S} of the coming signal \mathbf{x}_d with the desired reference signal \mathbf{r} . For the LMS algorithm which was used the general form of a signal and interference is presented in, the form of (5), where A_d and A_z define amplitudes of coming signal and interfering signal, θ_d and θ_z define the angle of coming signal and the angle of coming interference, and ϕ_d and ϕ_z define phasal shifts between elements of the antenna array as the following:

$$d_n = A_d \cdot e^{-in\phi_d}, \quad dz_n = A_z \cdot e^{-in\phi_z}. \quad (5)$$

Later on, signals are written in the form of $\mathbf{X}_d = [d_1, d_2, d_3, d_4, d_5, d_6]^T$ and $\mathbf{X}_z = [dz_1, dz_2, dz_3, dz_4, dz_5, dz_6]^T$, and then what is defined is the covariance matrix of a signal Φ_{sig} and the covariance matrix of interference Φ_{dis} according to the following relations:

$$\begin{aligned} \Phi_{\text{sig}} &= \bar{\mathbf{x}}_d \mathbf{x}_d^T, \\ \Phi_{\text{dis}} &= \bar{\mathbf{x}}_z \mathbf{x}_z^T. \end{aligned} \quad (6)$$

Next, the covariance matrix of both is calculated; the correlation vector \mathbf{S} and the weight vector \mathbf{W} are also estimated as follows:

$$\begin{aligned}\Phi &= \Phi_{\text{sig}} + \Phi_{\text{dis}}, \\ \mathbf{S} &= \bar{\mathbf{x}}_d \mathbf{r}, \\ \mathbf{W} &= \Phi^{-1} \mathbf{S}.\end{aligned}\quad (7)$$

Weights which are calculated with the use of the algorithm above are treated as model ones in the process of organizing. Then the architecture optimization of a network was prepared. It was done as a result of selecting the right number of hidden layers and neurons in these layers. Next, the defined network underwent the process of organizing and testing. Testing means drawing the power beam pattern, which weights were calculated with the use of LMS algorithm. When the results did not add up with the assumptions, other neurons were added to the network architecture. The process was carried on until the beam patterns of the microstrip antenna tallied with the model.

3.1. Self-Learning ANN Algorithm. To teach an artificial neural network, the Delta method was used which changes the learning error in the function of a number of self-learning epochs. The Delta rule is used with neurons which have constant activation function and is used as an equivalent of the perceptron rule which is also called the constant perceptron rule. This method can be easily introduced as a minimization of the square criterion error. Its general version can be used in multilayered networks. In the Delta rule each neuron after receiving particular signals on its inputs (from network inputs or other neurons which are earlier layers of the networks) defines its output signal using information in the form of weight values and (if necessary) thresholds, which were set earlier. The value of the output signal defined by the neuron in the particular learning stage of process is compared with the model reply given by the teacher in the learning sequence. If there is a disagreement that is when a neuron defines the difference between its output signal and the value of this signal, although which could be correct according to the teacher. Such difference is usually defined as δ (delta, a Greek letter) like the name of this method. The error signal (delta) is used by the neuron to correct its weight coefficients (and if necessary the threshold) applying following rules:

- (i) the more serious error was detected, the bigger change in weights is made,
- (ii) weights connected with the inputs on which there are high values of input signals are changed more than weights which input signal was not high.

With the use of Delta method in this work, the error function was defined with respect to the following relation (8), where d_{jp} is the model output signal j th element of the output layer while forcing with the learning signal p , and y_{jp} is the current

output signal j th element of the output layer while forcing with the learning signal p :

$$E_p = \frac{1}{2} \sum_{i=1}^n (d_{jp} - y_{jp})^2. \quad (8)$$

This method of calculating the function error is based on backpropagation algorithm where, first of all, errors in the last layer are calculated on the basis of comparison current and model output signals, and on such a ground connection weights are changed then in the layer before and previous ones until the first one. There are three stages in the backpropagation algorithm:

- (i) introduce the organizing signal \mathbf{x} on the input, and calculate current outputs \mathbf{y} ;
- (ii) compare the output signal \mathbf{y} with the model signal \mathbf{d} , and then calculate local errors for all network layers;
- (iii) do the weight adaptation.

For the energy function defined by (8) as a mean squared error, the updating weight method for the output layer can be defined with (9), where the index $(\cdot)^{[2]}$ is the output layer:

$$\Delta w_{ji}^{[2]} = -\eta \frac{\partial E_p}{\partial w_{ji}^{[2]}} = -\eta \frac{\partial E_p}{\partial u_j^{[2]}} \frac{\partial u_j^{[2]}}{\partial w_{ji}^{[2]}}. \quad (9)$$

Taking into account (10),

$$u_j^{[2]} = \sum_{i=1}^{n2} w_{ji}^{[2]} o_i^{[1]}, \quad (10)$$

and defining the local error with respect to the following equation (11):

$$\begin{aligned}\delta_j^{[2]} &= -\frac{\partial E_p}{\partial u_j^{[2]}} = -\frac{\partial E_p}{\partial e_{jp}} \frac{\partial e_{jp}}{\partial u_j^{[2]}} = -e_{jp} \frac{\partial (d_{jp} - y_{jp})}{\partial u_j^{[2]}} \\ &= e_{jp} \frac{\partial y_{jp}}{\partial u_j^{[2]}} = e_{jp} \frac{\partial \Psi}{\partial u_j^{[2]}},\end{aligned}\quad (11)$$

results in the relation defining updating weights in the output layer with respect to the following equation (12):

$$\begin{aligned}w_{ji}^{[2]}(t) &= w_{ji}^{[2]}(t-1) + \Delta w_{ji}^{[2]}, \\ \Delta w_{ji}^{[2]} &= \eta \delta_j^{[2]} o_i^{[1]}.\end{aligned}\quad (12)$$

More complicated problem is finding the local error for elements in hidden layers as the correct output signal from the elements of this layer is not known. Thus, available or

easily calculated data is used with respect to the formula (13) as follows:

$$\begin{aligned}\delta_j^{[1]} &= -\frac{\partial E_p}{\partial u_j^{[1]}} = -\frac{\partial E_p}{\partial o_{jp}^{[1]}} \frac{\partial o_{jp}^{[1]}}{\partial u_j^{[1]}} = -\frac{\partial E_p}{\partial o_{jp}^{[1]}} \frac{\partial \Psi}{\partial u_j^{[1]}}, \\ \frac{\partial E_p}{\partial o_j^{[1]}} &= \sum_{i=1}^{n_2} \frac{\partial E_p}{\partial u_i^{[2]}} \frac{\partial u_i^{[2]}}{\partial o_j^{[1]}} = \sum_{i=1}^{n_2} \left(\frac{\partial E_p}{\partial u_i^{[2]}} \right) \frac{\partial}{\partial o_j^{[1]}} \left(\sum_{k=1}^{n_2} w_{ik}^{[2]} o_k^{[1]} \right) \\ &= -\sum_{i=1}^{n_2} \delta_i^{[2]} w_{ij}^{[2]}.\end{aligned}\quad (13)$$

As a result; the formula for the local error was written according to (14), and allowance for weights was defined with the use of (15):

$$\delta_j^{[1]} = \frac{\partial \Psi}{\partial u_j^{[1]}} \sum_{i=1}^{n_2} \delta_i^{[2]} w_{ij}^{[2]}, \quad (14)$$

$$\Delta w_{ji}^{[1]} = \eta \delta_j^{[1]} x_i. \quad (15)$$

Firstly, the criterion finishing the process was the assumed number of learning epochs. However, analysing the graphs with the error change depending on the number of the epoch, this criterion does not react to the so-called effect of network overlearning which is decreasing the ability to generalize a net. More effective criterion finishing the process of learning artificial neural network is defining the learning error threshold after which the process is stopped.

3.2. Optimal Structure of ANN Selection. In the first stage of searching for an optimal structure of ANN was one hidden layer. The selection of number of neurons in the hidden layer starts with the least number and more of them are added gradually. The important role in this part of the process is testing the organized net and comparing the result with requirements defined in the assumption. The model beam pattern, marked with green colour, is compared with the results from artificial neural net simulation should be compared. The received ANN beam patterns are marked with black colour. These beam patterns were received with the use of an artificial neural net. The direction of the desired signal is marked with blue colour, and the direction of interfering signal is marked with red colour. The graphs which are presented are the patterns of power radiated from the array in the function of the radiating angle to the maximal values of radiated power.

4. Results of the Analysis

4.1. Optimal Architecture an ANN Selection for the Process of Following the Emission Source. The process of ANN learning in order to follow the emission source was started with ten neurons in the hidden layer. As a result of calculating weights; the ANN beam pattern was depicted on Figure 3. Influenced

by the level of side lobes, the decision about increasing the number of neurons was made. Next steps, which were finished with testing, show some shape improvement of the ANN beam pattern in comparison with the model pattern (marked with green colour). Figure 4 shows the beam pattern received with 14 neurons in the hidden layer and Figure 5 shows the beam pattern with 18 neurons in the hidden layer. With twenty neurons in the hidden layer, the shape of the ANN beam pattern and the level of side lobes were close enough to the model beam pattern (Figure 6).

4.2. Optimal Architecture an ANN Selection for the Process of Setting to Zero the Beam Pattern on the Expected Interference Direction. In case of optimal architecture, an artificial neural network selection with the setting to zero variant on the direction of expected interference the test of an ANN with twelve neurons in the hidden layer was started. On the basis of Figure 7, it can be said that the received results of an ANN beam pattern are far from expectations as on the direction of interference there is side lobe, which cannot be accepted. Gradual increasing of the number of neurons (Figures 8 and 9) improved the beam pattern making it closer to the shape of the assumed model beam pattern—marked with black colour. In this case as well as in the previous one, the minimal number of neurons in the hidden layer which meets the requirements provided in assumptions was defined as twenty neurons in the hidden layer presented on Figure 10.

4.3. Following the Emission Source with the Use of an ANN . The effect of further tests was receiving beam patterns with the use of an artificial neural network. Some models of ANN beam patterns were received with the use of an artificial neural network on Figures 11, 12, 13, and 14. The graphs presented here are patterns of radiation power from the antenna array in the function of radiation angle which were standardized to the maximal radiation power. In the process of following emission source, the beam patterns presented on Figures 11–14 are scanning a space beam and following the desired signal.

4.4. The Beam Pattern Setting to Zero on the Interference Direction with the Use of an ANN . The ANN beam patterns on Figures 15, 16, 17, and 18 present the process of elimination of interference (marked with red colour on the graphs) on particular directions taking into account the assumption that the desired signal comes from the direction perpendicular to the antenna array. The desired signal is marked with blue colour.

4.5. Following the Emission Source with Two Constant Interferences with the Use of an ANN . The ANN beam patterns presented on Figures 19, 20, 21, and 22 are showing the process of following the desired signal which was marked with blue colour on the interference background. The interference was marked with red colour on the models of interferences with 45 degrees and 135 degrees.

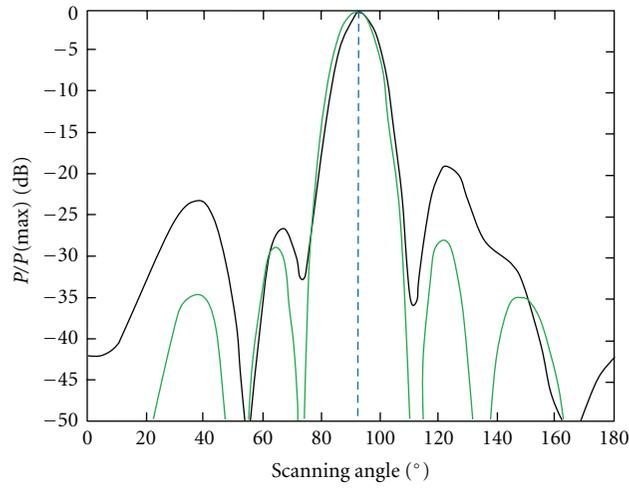


FIGURE 3: The beam pattern received with 10 neurons in the hidden layer—following ES.

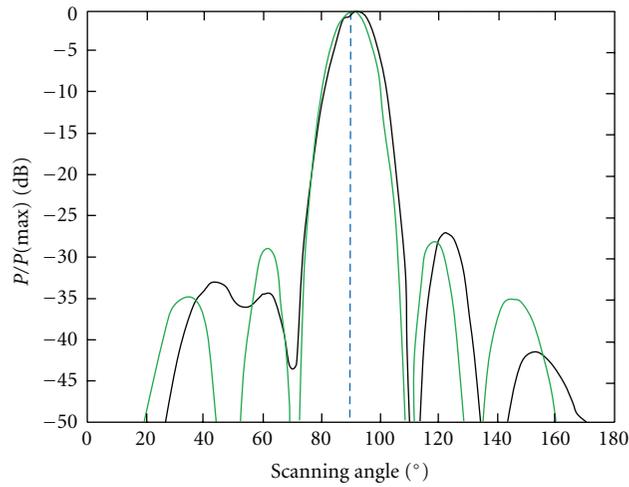


FIGURE 4: The beam pattern received with 14 neurons in the hidden layer—following ES.

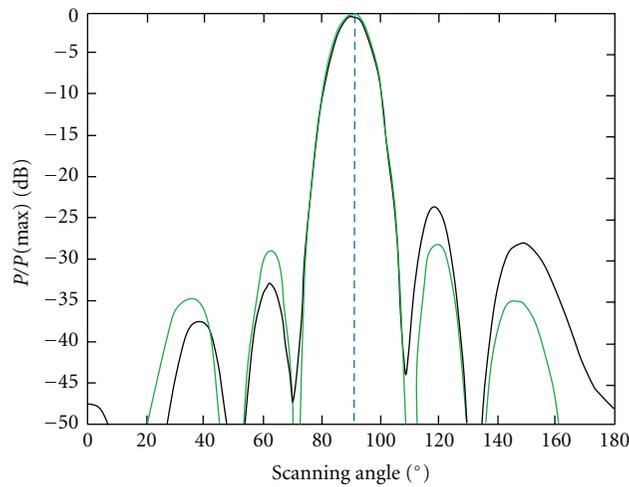


FIGURE 5: The beam pattern received with 18 neurons in the hidden layer—following ES.

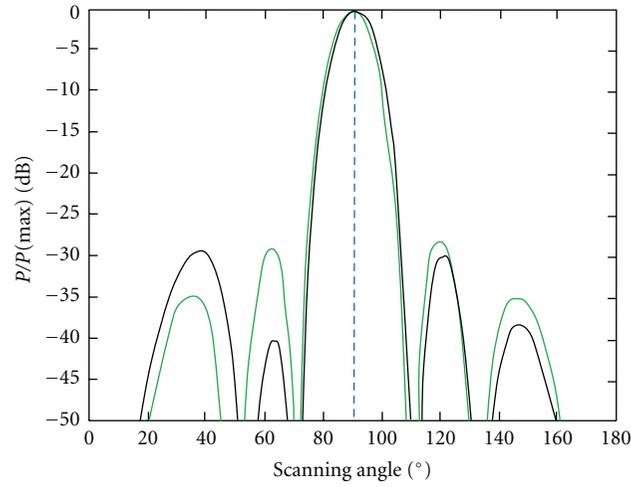


FIGURE 6: The beam pattern received with 20 neurons in the hidden layer—following ES.

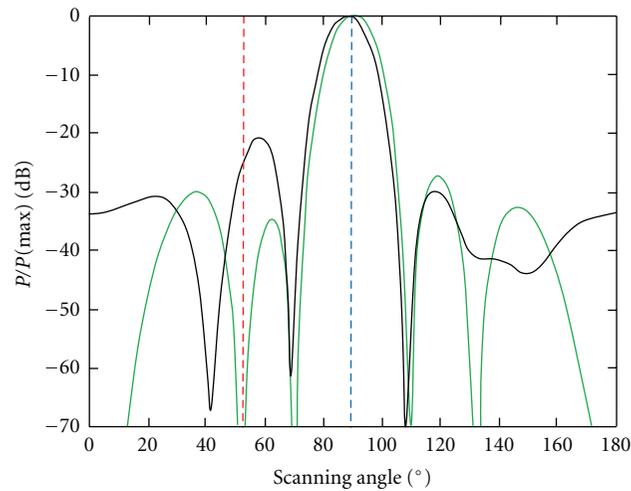


FIGURE 7: The ANN beam pattern received with 12 neurons in the hidden layer—the beam pattern setting to zero on the interference direction.

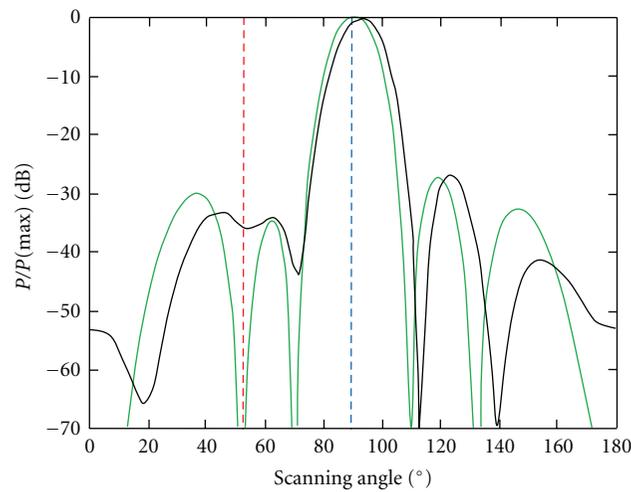


FIGURE 8: The ANN beam pattern received with 14 neurons in the hidden layer—the beam pattern setting to zero on the interference direction.

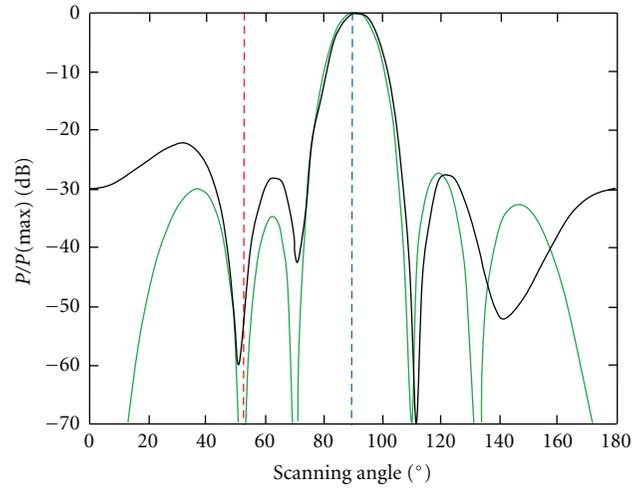


FIGURE 9: The ANN beam pattern received with 16 neurons in the hidden layer—the beam pattern setting to zero on the interference direction.

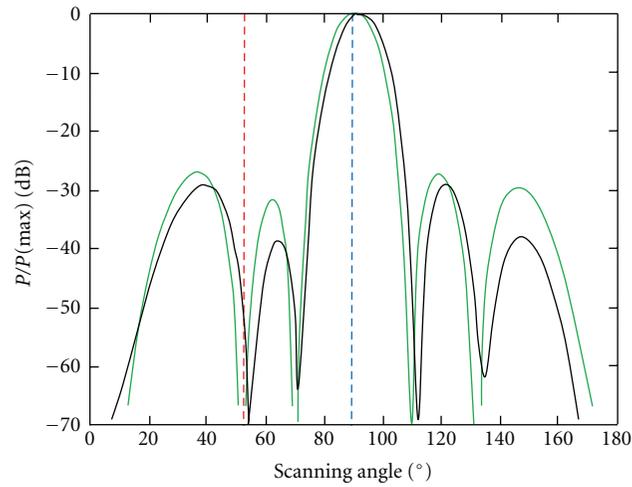


FIGURE 10: The ANN beam pattern received with 20 neurons in the hidden layer—the beam pattern setting to zero on the interference direction.

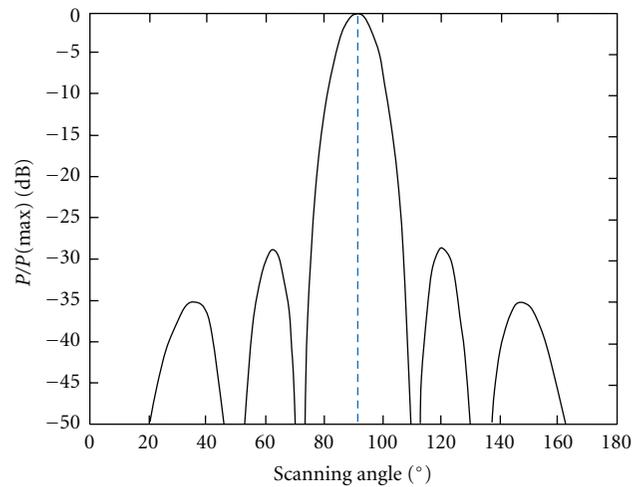


FIGURE 11: The beam pattern pointed towards the signal-glancing right.

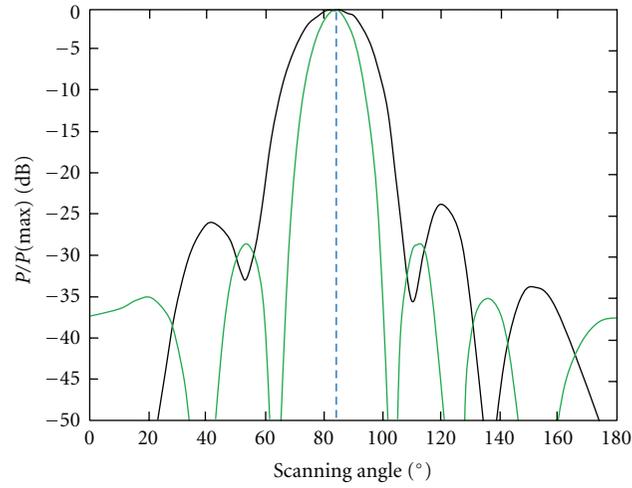


FIGURE 12: The beam pattern pointed towards the useful signal with the angle of 83 degrees.

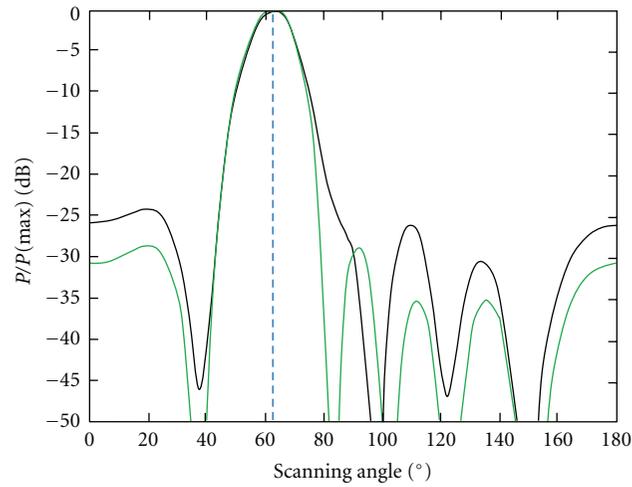


FIGURE 13: The beam pattern pointed towards the useful signal with the angle of 62 degrees.

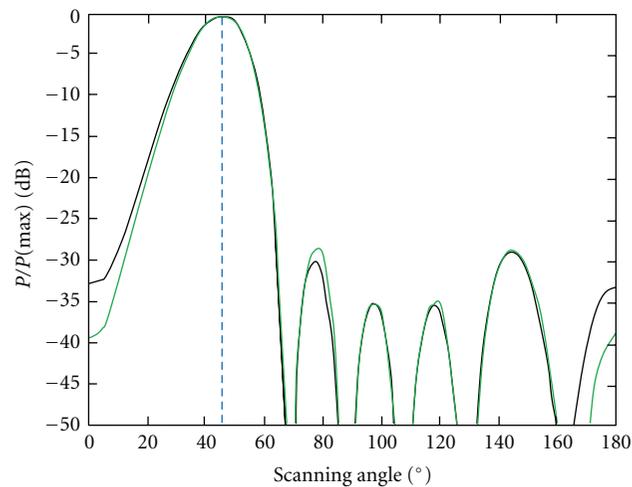


FIGURE 14: The beam pattern pointed towards the useful signal with the angle 45 degrees.

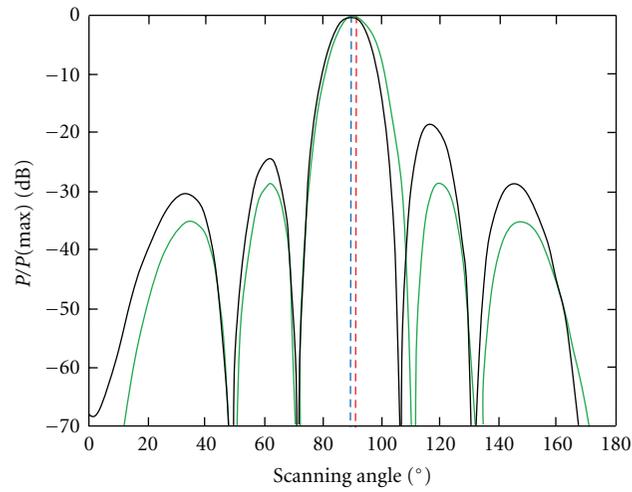


FIGURE 15: The beam pattern of the antenna array with 0 degree of interference.

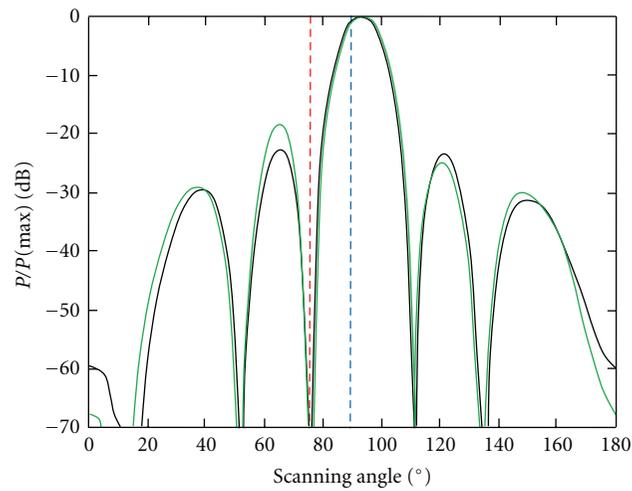


FIGURE 16: The beam pattern of the antenna array with 75 degrees of interference.

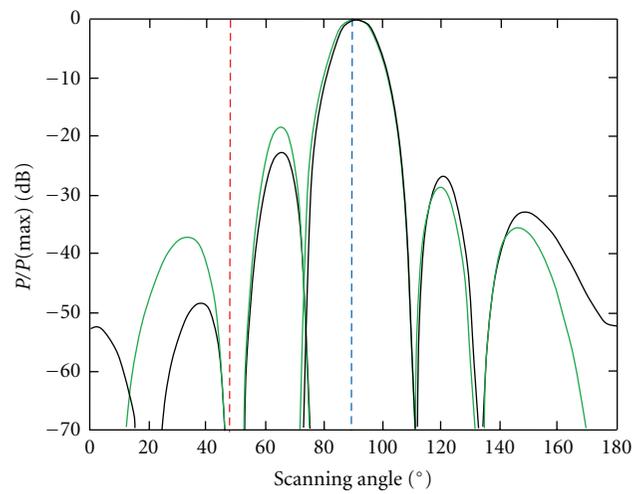


FIGURE 17: The beam pattern of the antenna array with the 47 degrees of interference.

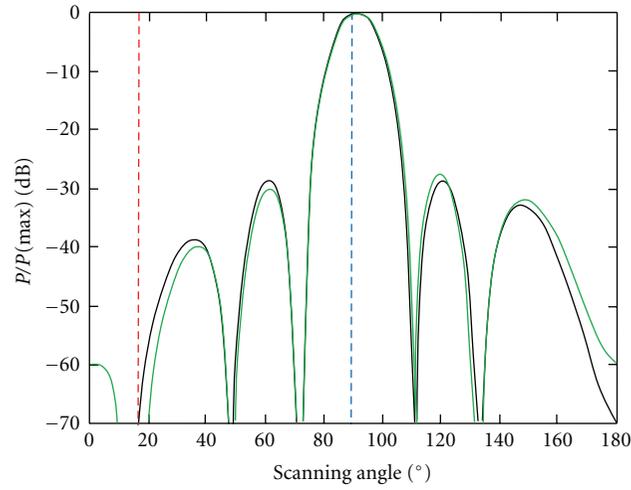


FIGURE 18: The beam pattern of the antenna array with the 15 degrees of interference.

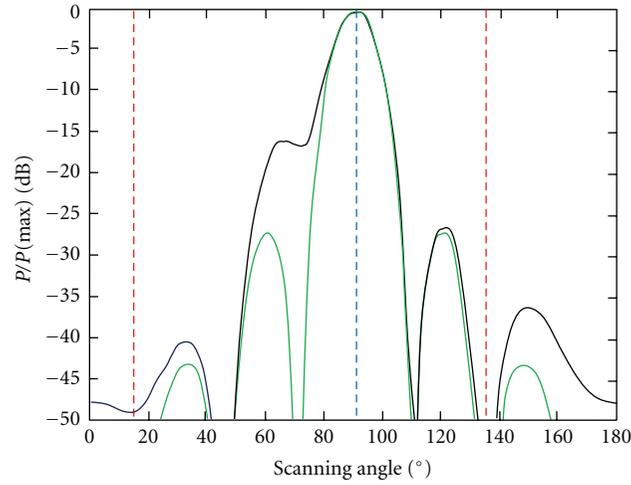


FIGURE 19: An ANN beam pattern pointed towards the signal with the right angle.

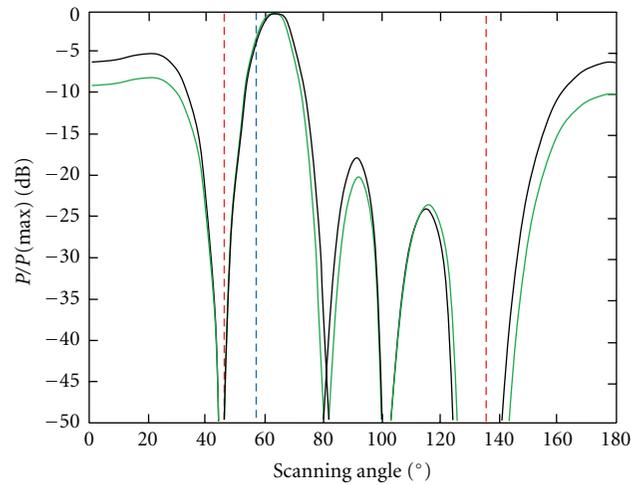


FIGURE 20: An ANN beam pattern pointed towards the signal with the angle of 55 degrees.

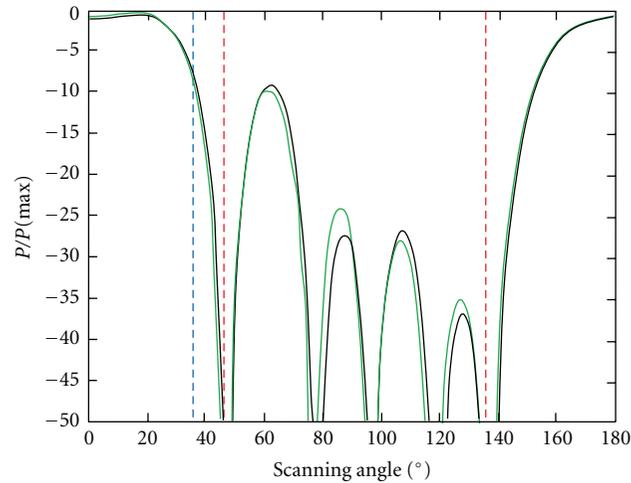


FIGURE 21: An ANN beam pattern pointed towards the useful signal with the angle of 35 degrees.

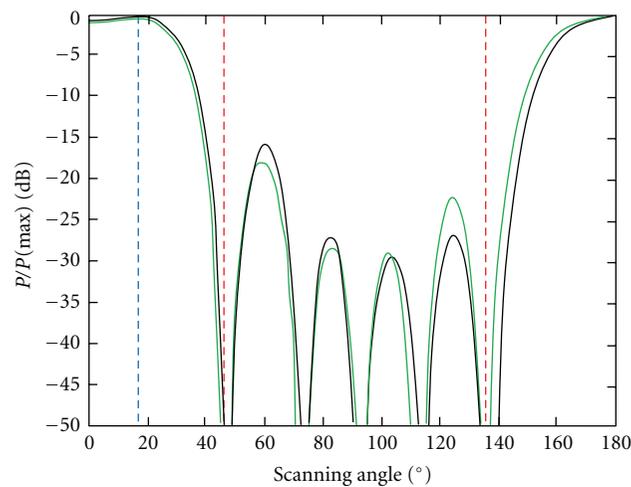


FIGURE 22: An ANN beam pattern pointed towards the useful signal with the angle of 15 degrees.

5. Conclusions

This paper presented the use of an artificial neural network to form a microstrip complex antenna array beam pattern. On the basis of the graph analysis of the ANN beam pattern received in the use of neural networks and comparing them with graphs received as a result of using traditional algorithms, it can be said that due to the adjustment of the neural network structure to the task (the appropriate number of hidden layers and the number of neurons in these layers), the results which were received agreed with the model beam pattern.

In case of an emission source following the main lobe of beam pattern goes after the signal. The only possible quibble about it may be the size of side lobes. Nevertheless, the main goal was achieved, and the case of side lobes minimization will probably be the topic of further tests in this aspect.

It was assumed that there will be an emission source of the desired signal on the direction perpendicular to the antenna array while setting to zero the beam pattern on

the direction of the expected interference. In comparison with the previous case, such an option increases the number of complications while controlling the pattern. However, even such an option provides satisfying results which are presented on Figures 15–18.

The most complicated issue was following the emission source while having simultaneously two interferences. This task required increasing the learning set which resulted in increasing the time of an ANN learning. Improvement of the shape of the beam pattern was made as a result of the extension of an artificial neural network in the layer. Some limitations are also introduced by the platform used in neural networks simulation which need to be considered while precisising the assumptions concerning the expected quality of the mapping pattern.

The artificial neural network tested here turned out to be a tool fair enough to make decisions about the signal which after learning is able to distinguish interference from the desired signal. In this case the criterion of making the

decision was connected only with the signal amplitude; however, further tests will be concentrated on increasing the parameters considered as the criterion. However, it must be emphasized that the model neural network requires another process of learning in every change of conditions resulting from the assumptions. This process of learning adjusts its weights to the ensuing situation, which complicates the use of an artificial neural network in cases where there are many changes. What should be also noticed is that work of an ANN is adding, multiplying, and comparing with the output signals' threshold which in comparison with mathematical operations used in typical algorithms (matrix inversion, correlation vectors calculation, and covariance matrix calculation) facilitates using the DSP technique.

Using complex microstrip antenna arrays decreases weights and production costs. The characteristic feature of microstrip antenna arrays is their simple change of the phase area decomposition in the aperture of the system, which makes them easy to use in electronic controlling the beam pattern.

Acknowledgment

This work was supported by the National Centre for Research and Development (NCBiR) from sources for science in the years 2010–2012 under Project O R00 0161 12.

References

- [1] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Array*, John Wiley & Sons, 1980.
- [2] B. Widrow, P. E. Mantey, L. J. Griths, and B. B. Goode, "Adaptive antenna systems," *Proceedings of the IEEE*, vol. 55, pp. 2143–2159, 1967.
- [3] S. P. Applebaum, "Adaptive Arrays," *IEEE Transactions on Antennas and Propagation*, vol. 24, no. 5, pp. 585–598, 1976.
- [4] S. W. W. Shor, "Adaptive technique to discriminate against coherent noise in a narrow-band systems," *Journal of the Acoustical Society of America*, vol. 34, no. 1, pp. 74–78, 1966.
- [5] R. L. Riegler and R. T. Compton, "Adaptive array for interference rejection," *Proceedings of the IEEE*, vol. 61, no. 6, pp. 748–758, 1973.
- [6] R. Garg, P. Bhartia, I. Bahl, and A. Ittipiboon, *Microstrip Antenna Design Handbook*, Artech House INC, London, UK, 2001.
- [7] J. F. Zurcher and F. E. Gardiol, *Broadband Patch Antennas*, Artech House INC, London, UK, 1995.
- [8] S. Osowski, *Neural Networks in Algorithms*, Scientifically Technical Press, Warsaw, Poland, 1996.
- [9] D. Rutkowska, M. Pilinski, and L. Rutkowski, *Neural Networks, Genetic Algorithms and Fuzzy Systems*, Scientifically Technical Press, Warsaw, Poland, 1997.
- [10] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to Neural Networks*, Scientifically Technical Press, Warsaw, Poland, 1993.
- [11] W. S. McCulloch and W. H. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [12] D. O. Hebb, *The Organization of Behaviour*, John Wiley & Sons, New York, NY, USA, 1949.
- [13] F. Rosenblatt, *Principles of Neurodynamics*, Spartan, Greece, 1962.
- [14] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *1960 IRE WESCON Convention Record*, pp. 96–104, New York, NY, USA, 1960.
- [15] M. L. Minsky and S. A. Papert, *Perceptrons*, MIT Press, 1969.
- [16] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [17] F. Fazayeli, L. Wang, and W. Liu, "Back-propagation with chaos," in *Proceedings of the IEEE International Conference Neural Networks and Signal Processing (ICNNSP' 08)*, pp. 5–8, Zhenjiang, China, June 2008.
- [18] V. Kecman, *Learning and Soft Computing, Support Vector machines, Neural Networks and Fuzzy Logic Models*, The MIT Press, Cambridge, Mass, USA, 2001.
- [19] L. P. Wang and X. J. Fu, *Data Mining with Computational Intelligence*, Springer, Berlin, Germany, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

