World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:2, 2007

Complex-Valued Neural Network in Image Recognition: A Study on the Effectiveness of Radial Basis Function

Anupama Pande, and Vishik Goel

Abstract**—**A complex valued neural network is a neural network, which consists of complex valued input and/or weights and/or thresh olds and/or activation functions. Complex-valued neural networks have been widening the scope of applications not only in electronics and informatics, but also in social systems. One of the most important applications of the complex valued neural network is in image and vision processing. In Neural networks, radial basis functions are often used for interpolation in multidimensional space. A Radial Basis function is a function, which has built into it a distance criterion with respect to a centre. Radial basis functions have often been applied in the area of neural networks where they may be used as a replacement for the sigmoid hidden layer transfer characteristic in multi-layer perceptron.

This paper aims to present exhaustive results of using RBF units in a complex-valued neural network model that uses the back-propagation algorithm (called 'Complex-BP') for learning. Our experiments results demonstrate the effectiveness of a Radial basis function in a complex valued neural network in image recognition over a real valued neural network. We have studied and stated various observations like effect of learning rates, ranges of the initial weights randomly selected, error functions used and number of iterations for the convergence of error on a neural network model with RBF units. Some inherent properties of this complex back propagation algorithm are also studied and discussed.

Keywords**—**Complex valued neural network, Radial Basis Function, Image recognition.

I. Introduction

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Artificial Neural Networks, like people, learn by example. An Artificial Neural Network is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons.

A. Historical Background

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much. Neural network simulations appear to be a recent development. Many important advances have been boosted by the use of inexpensive computer emulations. During the period when funding and professional support was minimal, relatively few researchers made important advances. These pioneers were able to develop convincing technology, which surpassed the limitations. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

B. Back Propagation Algorithm in a Real Plane

Back propagation algorithm has been used extensively in neuron models. This algorithm is a development from the simple Delta rule in which extra hidden layers (layers additional to the input and output layers, not connected externally) are added. The network topology is constrained to be feed forward or loop-free - generally connections are allowed from the input layer to the first (and possibly only)

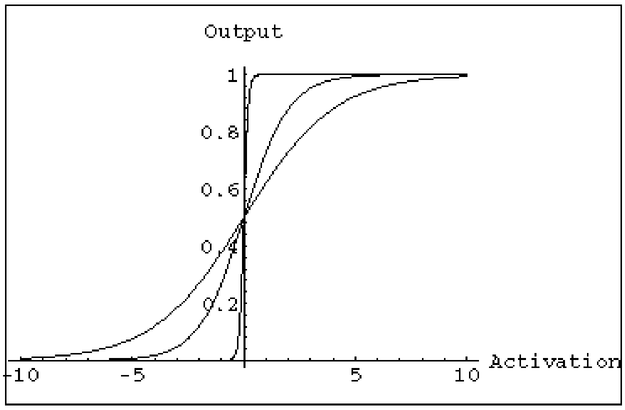

Fig. 1 Input/ output graph of a back-propagation unit

Manuscript received February 28th, 2007.

Anupama Pande is with IBM India Pvt Ltd as a software developer and analyst (phone: +91-9900212837; e-mail: anupamapande@ in.ibm.com).

Vishik Goel is with Computer Science Corporation (CSC) as a software developer and analyst (phone: +919910392055; fax: +91-80-41922150; e-mail: vgoel5@csc.com).

hidden layer; from the first hidden layer to the second and from the last hidden layer to the output layer. In a typical back propagation network, the hidden layer learns to recode (or to provide a representation for) the inputs. More than one hidden layer can be used. The architecture is more powerful than single-layer networks: it can be shown that any mapping can be learned, given two hidden layers (of units). The units are a little more complex than those in the original perceptron. Their input/ output graph is shown above. As a function:

$$\mathbf{Y = 1 / (1 + e^{(-k.(\Sigma W_{in} * X_{in}))})}$$

The graph shows the output for *k=0.5, 1, and 10,* as the activation varies from -10 to 10.The weight change rule is a development of the perceptron learning rule. Weights are changed by an amount proportional to the error at that unit times the output of the unit feeding into the weight [7].

Running the network consists of the forward pass and backward pass. In the forward pass the outputs are calculated and the error at the output units calculated. In the backward pass the output unit error is used to alter weights on the output units. The error at the hidden nodes is calculated (by back-propagating the error at the output units through the weights), and the weights on the hidden nodes are altered using these values. For each data pair to be learned, a forward pass and backwards pass is performed. This is repeated over and over again until the error is at a low enough level.

C. Radial Basis Function Networks

RBF hidden layer units have a receptive field which has a centre*:* that is, a particular input value at which they have a maximal output. Their output tails off as the input moves away from this point. Generally, the hidden unit function is a Gaussian. RBF networks are trained by deciding on how many hidden units there should be, deciding on their centers and the sharpnesses (standard deviation) of their Gaussians and then training up the output layer [8].

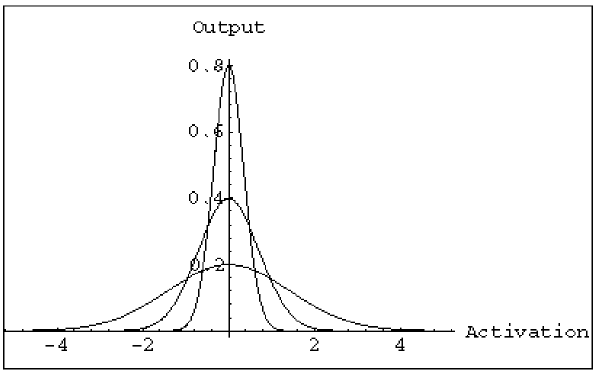

Fig. 2 Input/ output graph of a Radial Basis unit

Generally, the centers and standard deviations are decided on first by examining the vectors in the training data. The output layer weights are then trained using the Delta rule. Back Propagation is the most widely applied neural network technique. Radial Basis Functions have the advantage that one can add extra units with centers near parts of the input which are difficult to classify [15].

RBF networks have the advantage of not suffering from local minima in the same way as multi-layer perceptrons. This is because the only parameters that are adjusted in the learning process are the linear mapping from hidden layer to output layer. Linearity ensures that the error surface is quadratic and therefore has a single easily found minimum. In regression problems this can be found in one matrix operation. In classification problems the fixed non-linearity introduced by the sigmoid output function are most efficiently dealt with using iterated reweighed least squares [16].

RBF networks have the disadvantage of requiring good coverage of the input space by radial basis functions. RBF centers are determined with reference to the distribution of the input data, but without reference to the prediction task. As a result, representational resources may be wasted on areas of the input space that are irrelevant to the learning task [8].

II. Complex Valued Neural Network

A complex valued neural network is a neural network (of arbitrary topology), which consists of complex valued input and/or thresholds and/or activation functions. The need for such neural networks is widespread [3], [6].

For instance, in electrical engineering, signals are complex valued. The processing of such signals requires the design and implementation of new complex valued neural network architectures. This subject has been gaining increasing interest and significance in recent years. One of the most important characteristics of the Complex-valued neural networks is the proper treatment of complex-amplitude information, e.g., the treatment of wave-related / rotation-related phenomena such as electromagnetism, light waves, quantum waves, oscillatory phenomena even including traffic signal control, and color images processing based on adaptive signal rotation.

A. The Complex BP Algorithm

This algorithm is a complex valued version of a probabilistic-descent method [1]. It has been proved that the learning algorithm for the Complex adaptive Pattern Classifier Model converges. This algorithm states the following.

If **n** is a parameter representing discrete time then we can modify the complex valued parameter **w** as

$$\mathbf{w_{n+1} = w_n + \Delta w_n},$$

where $\mathbf{w_n}$ denotes a complex valued parameter at time **n.** We can also re-write the equation as follows:

$$\mathbf{Re[w_{n+1}] = Re[w_n] + Re[\Delta w_n]},$$
$$\mathbf{Im[w_{n+1}] = Im[w_n] + Im[\Delta w_n]},$$

where **Re[z], Im[z]** denotes the real and imaginary part of a complex number **z**, respectively. By definition we say that a parameter **w** is optimal if and only if the average error **R (w)** is local or global minimum. Then the following theorem holds.

Let **A** be a positive definite matrix. Then by the using the update rules:

$$\mathbf{Re}[\Delta w_n] = -\varepsilon A \Delta^{Re} r(z(w_n, x_n), y_n),$$
$$\mathbf{Im}[\Delta w_n] = -\varepsilon A \Delta^{Im} r(z(w_n, x_n), y_n), \quad n = 0, 1\ldots..$$

The (complex-valued) parameter **w** approaches the optimum as near as desired by choosing a sufficiently small learning constant $\varepsilon > 0$ (Δ^{Re} is a gradient operator with respect to the real part of **w**, and Δ^{Im} with respect to the imaginary part).

B. Generalization of Real Back Propagation Algorithm

The theory of Complex adaptive Pattern Classifier Model has been applied to a multi-layer (complex valued) neural network [1]. For complex valued back propagation model we used the following derived results [11], [12], [13], [14].

In a complex Back Propagation model all the input signals, weights, thresholds and output signals are complex numbers. The output activity (analogous to the activity of a real BP) for a neuron n is defined as:

$$Y_n = \sum W_{nm} X_m + V_n,$$

where W_{nm} is the (complex-valued) weight connecting neuron **n** and **m**, X_m is the (complex-valued) input signal from neuron **m**, and V_n is the (complex-valued) threshold value of neuron **n**. To obtain the (complex-valued) output signal, the activity Y_n is converted into its real and imaginary part

$$Y_n = x + iy = z$$

where i denotes √-1. Although various output functions can be considered, we used the output definition defined by

$$f_C(z) = f_R(x) + i\, f_R(y)$$

where $f_R(x)$ is a sigmoid function. Also $f_C(z)$ is not holomorphic.

Consider the following variables, w_{ml} is the weight between the input layer **m** and the hidden node **l**, v_{nm} is the weight between the hidden layer **n** and the output neuron **m**, **Θ** is the threshold of the **m** hidden neuron, **γ** is the threshold of the **m** output neuron. Let I_l, H_m, O_n denote the output values of the input neuron **l**, the hidden neuron **m** and the output neuron **n**, respectively. Let U_m and S_n denote the internal potentials of the hidden neuron **m** and the output neuron **n,** respectively. That is $U_m = \sum w_{ml} \; I_l + \Theta_m$, $S_n = \sum v_{nm} \; H_m + \gamma_n$, $H_m = f_C(U_m)$ and $O_n = f_C(S_n)$.

Let $\delta^n = T_n - O_n$ denote the error between the actual pattern O_n and the target T_n of output neuron **n**. The square error for the pattern **p** is $E_p = (1/2) \sum (T_n - O_n)^2$ where N is the number of output neurons. The learning rule for the complex-BP model described above is as follows. For a sufficiently small learning constant (learning rate) $\varepsilon > 0$ and a unit matrix **A**, using Theorem stated above, it has been shown that the weights and the threshold should be modified according to the following equations:

$$\Delta v_{nm} = \hat{H}_m \Delta\gamma_n$$

$$\Delta\gamma_n = \varepsilon(Re[\delta_n](1 - Re[O_n])Re[O_n] + i\, I_m[\delta_n](1 - Im[O_n])Im[O_n])$$

$$\Delta\omega_{mt} = \hat{I}_t \Delta\theta_m$$

$$\Delta\theta_m = \varepsilon[(1 - Re[H_m])Re[H_m]$$
$$\sum(Re[\delta n](1 - Re[O_n])Re[O_n])Re[O_n]Re[v_{nm}] + Im[\delta n](1 - Im[O_n])Im[O_n])Im[O_n]Im[v_{nm}])$$
$$- i(1 - Im[H_m])Im[H_m]$$
$$\sum(Re[\delta^n](1 - Re[O_n])Re[O_n])Re[O_n]Im[v_{nm}] - Im[\delta^n](1 - Im[O_n])Im[O_n])Im[O_n]Re[v_{nm}])]$$

where Â denotes the complex conjugate of A [4], [2].

III. Experiments

A. C – XOR Benchmark Problem

Multi-layer networks use a variety of learning techniques. We used one of the popular techniques of back-propagation. We compared the output values with the correct answer to compute the value of some predefined error-function. The error was then fed back through the network. Using this information, the algorithm adjusted the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles the network converged to some state where the error of the calculations is small. In this way the network has learnt a certain target function. To adjust weights properly we applied a general method for non-linear optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason back-propagation can only be applied on networks with differentiable activation functions. [2], [5]

To check the efficiency of a RBF unit in a complex plane, we first observed the effect of an RBF unit on a small model first. We researched on the results obtained with and without the RBF units. Our model took complex valued inputs of a C-XOR benchmark problem. To enhance our experiments results we varied the learning rates, the initial weights taken, and the iterations of the neural network. We also studied the effect of various error functions on neural networks with and without the RBF unit.

1. Model Description

Our model had two complex valued inputs which corresponded to the inputs of the CXOR benchmark problem. The model had two hidden layers. The first layer had six nodes in the first hidden layer. The nodes had the following functions: Two summation functions, two production function and two RBF units. The second hidden layer had a summation function. This was finally sent to the output layer which also had another summation function.

2. Input

We used f(x) (1/ (1+exp (-x))) as our activation function. The value of the learning rate was varied from 0.1 to 0.9 and all the results were carefully observed and thoroughly studied. The initial weights were chosen randomly by rand ( ) function. This was also varied to get the optimal results. The model was experimented with various error functions to see with which error function the error converged in minimum number of iterations. All the observations were extensively researched.

3. Results and Discussion

Various important results were deduced from the above experiments. They have been summarized as below:

The presence of a RBF unit improves the efficiency of a neural model to a great deal. With the same set of values, a NN with an RBF unit shall, converge the error in lesser number of iterations. In our experiment the optimization was not only in case of iterations but also with the number of nodes present in the model. We got an error of .00001 for an RBF Neural Network within 4000 iterations that is our model with 9 nodes (two as RBF units) converged the error within 4000 iterations. Keeping the other values at optimum levels, the model with 13 nodes (none as RBF) showed an error convergence within 10000 iterations.

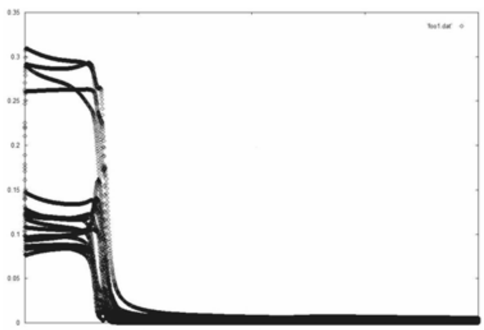

Fig. 3 Error convergence graph using complex quadratic error function

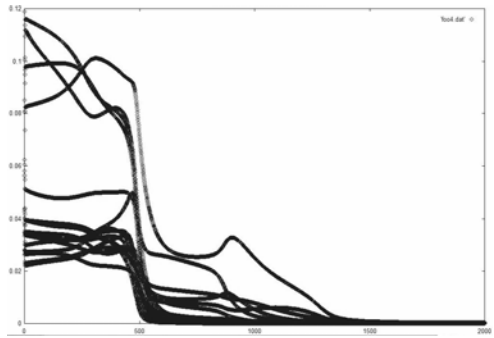

Fig. 4 Error convergence graph using complex fourth error function

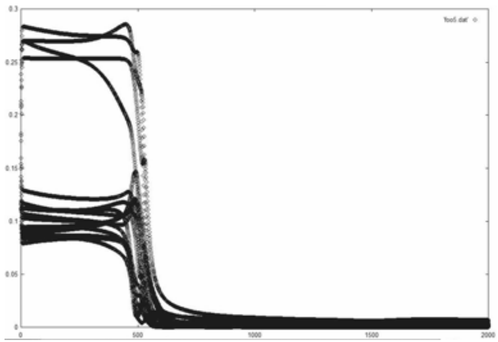

Fig. 5 Error convergence graph using complex mean-median error function

We experimented with various values of the learning rates. We varied the learning rates from 0.1 and 0.9 and observed the effect of the learning rates on the error convergence. For Complex Quadratic, Complex fourth, Complex Huber, complex Logcosh, Complex Welch, Complex Minkowski, Complex Fair error function the best error convergence rate was observed when the learning rate was 0.5 – 0.55. For mean-median error function the best convergence rate was observed when the learning rate was 0.45 – 0.5

B. Image Recognition on Real Life Example

1. Model Description

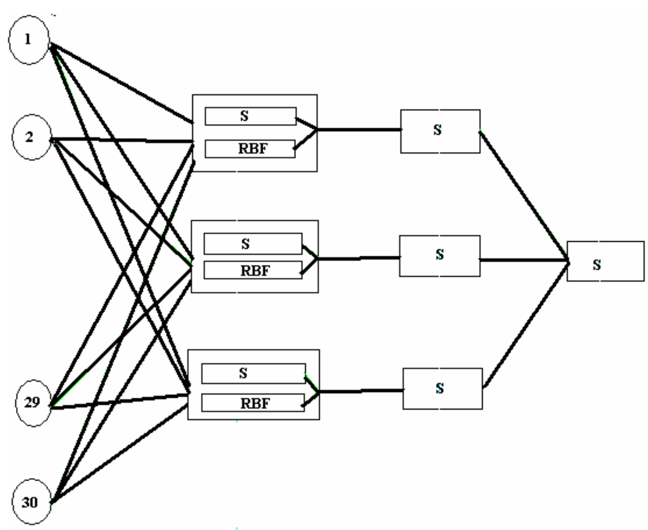

Fig. 6 Model Diagram

Our model took the eigen values matix of 30 * 30 as an input and processed them. The input had 30 nodes. The first hidden layer had three RBF units and three summation units as depicted. The output from these functions was then summed up further using another summation node. All the output from these summation nodes was further summed up to get the final output.

2. Sample Input

3. Methodology

The images were gathered to first train the neural network. Some of the images on which we trained our NN have been displayed above. Each image was read as a 2D array of pixel intensities for ORL database. Now the images read were converted to grayscale if they were in RGB format. Then masking was performed on the images. The grayscale images were masked off using a mast, an elliptical mask. This helped in extracting the oval face of a person and also removing the unnecessary background images.

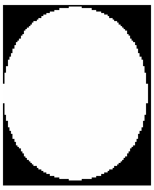

Fig. 7 Mask used to extract the face

All these masked images were of resolution 92*112 so we resize them to 30*30 using the bilinear interpolation method with no filter. The resized images were then used to calculate the complex eigenvalues of the respective images. Thus the image of size 30*30 gives an array of eigenvalues of size 30. These 30 complex values are used as inputs to the complex Back propagation algorithm. The network uses complex weights and has 30 complex inputs and one output (complex). Then the network was trained for a given complex value for a given person.

4. Result and Discussion

The results of the above experiments were thoroughly studied and documented. The presence of an RBF unit in pattern recognition system greatly enhanced the error convergence rate. The model achieved an error of 0.000001 within 12,000 iterations. This also depended on the value of learning rate, weights, resolution of picture and also number of inputs taken.

We observed that to get an optimum output the weights generated randomly should be between -0.01 and 0.01. To get a better error convergence rate the input should be fed again into the neural network during the training phase. All the inputs should be normalized between the values of -1 and +1. We tested our model on a number of error functions also. The error function that gave us the best result was the complex quadratic error function. The error convergence graph is shown below.

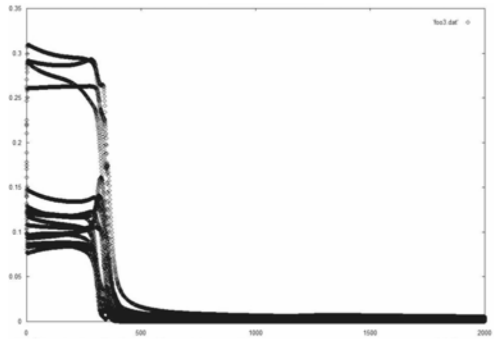

Fig. 8 Error convergence graph using complex quadratic error function

While testing if the values that are fed as inputs to the n/w yield output close to a given output value for which the n/w was earlier trained then that person is recognized. Our model identified the images with 90% to 95% accuracy. To improve the accuracy level even further the input points can be increased then while image resolution there will be a lesser loss while resizing.

IV. Conclusion and Future Work

We have shown the effectiveness of a complex valued version of back propagation model with RBF units for pattern recognition. Furturemore, we have investigated the fundamental characteristics of the complex- Back Propagation algorithm and found this algorithm better than general or real valued BP algorithm. The average convergence speed is much superior to that of a Real Back Propagation. [11] The updating rule of the Complex-Back Propagation is such that the probability for a standstill in learning is reduced. Also the presence of an RBF unit in a complex neural model is well suited for pattern recognition and learning complex-valued images [9], [10].

Indeed several interesting applications of the complex valued neural network architectures can be extended in the following areas:

Optoelectronics, Imaging, Optical computing, Remote sensing, Quantum Neural devices and systems, Intelligent transport systems, Spatiotemporal analysis of Physiological Neural Systems, Artificial Neural Information Processing, Communication system design (Mobile channel equalizer design), Direction of Arrival Estimation (Signal Processing), Traffic Control, Robotics, Neuron Dynamics, Chaos in the complex domain.

ACKNOWLEDGMENT

We wish to thank our mentor and project guide Mr. Bipin K. Tripathi for providing us with an opportunity for this study. We are also grateful to our colleagues for their valuable comments and suggestions.

REFERENCES

[1] Amari, S. (1967). “A theory of adaptive pattern classifier.” *IEEE Transactions on electronic computers*, EC-16 (3), 299 – 307.
[2] Benvenunto N. and Piazza F., “On the Complex Backpropagation Algorithm”, *IEEE Trans and Signal Processing*, Vol 40, No 4 pp. 967-969, 1992
[3] Derrick, W. R. (1984). “Complex Analysis and applications.” *Wadsworth*, Inc.
[4] Georgiou G.M. and Koutsougeras, “Complex Domain Backpropagation”, *IEEE Trans Circuits and System – II: Analog and Digital Processing*, Vol 39, No 5, pp. 330-334, 1992
[5] Hirose. A., “Proposal of fully Complex-valued Neural Networks”, *Proceedings of international Joint Conference on neural networks*, Vol 4, pp 152-157, 1992
[6] Kim, M.S., & Guest, C.C. (1990). “Modification of back propagation networks for complex-valued signal processing in frequency domain.” *Proceedings of the IJNN, San Diego*, June, 3, 27-31.
[7] Lipmann R. P., “An introduction to computing with neural nets”, *IEEE Acoustic, Speech and Signal Processing Magzine,* pp. 4-22, April 1987.
[8] Mark J. L. Orr (1966, April). “Introduction to Radial Basis Function Networks.” Available: http://www.anc.ed.ac.uk/rbf/intro/intro.html
[9] Miyauchi. M.. & Seki. M. (1992a). “Interpretaion of optical flow through neural network learning.” *Proceedings of IEEE international conference on communication systems,* international symposium on information theory and its applications, Singapore. Nov 1247 – 1251
[10] Miyauchi. M., Seki M., Watanabe. A. & Miyauchi. A. (1992 b). “Interpretation of optical flow through neural network learning”. *Proceedings of IAPR workshop on machine vision applications, Toky*o, Dec , 523-528
[11] Nitta T., "An Extension of the Back-Propagation Algorithm to Complex Numbers", *Neural Networks*, Vol.10, No.8, pp.1391-1415 (1997)
[12] Nitta T., "An Analysis of the Fundamental Structure of Complex-valued Neurons", *Neural Processing Letters*, Vol.12, No.3, pp.239-246
[13] Nitta T., "Redundancy of the Parameters of the Complex-valued Neural Networks", *Neurocomputing*, Vol.49, Issue 1-4, pp.423-428 (2002).
[14] Nitta T., "Solving the XOR Problem and the Detection of Symmetry Using a Single Complex-valued Neuron", *Neural Networks*, Vol.16, No.8, pp.1101-1105 (2003)
[15] P. Geitnner, D. Leers and H.-J. Hagemann: “Radial Basis Function for speech pattern recognition”
[16] R.A. Finan, A.T. Sapeluk and R.I. Damper, University of Southampton, “Comparison of Multilayer and Radial Basis Function Networks for Text-Dependent Speaker Recognition”

Miss Anupama Pande is a computer science engineer from Harcourt Butler Technological Institute, one of the premier colleges in India. She completed her Bachelors with honors as the topper of the Computer science and Information Technology department. During her college years she also published a paper in the International Academy of Business and Economics (IABE) in the journal Review of Business Research (RBR) in November 2006. She then joined India software Labs of IBM and is presently working under a software development team of DB2 as an analyst and a software engineer. This paper is the author's personal work product, and IBM India Pvt Ltd, its parent or any of its affiliates have not contributed to and do not own or have any rights to the contents disclosed in this paper.

Mr. Vishik Goel is a Computer Science Engineer from Harcourt Butler Technological Institute, Kanpur (INDIA). Apart from novel work in the field of neural networks he has also done numerous projects on compilers and database administration. He is presently working as a Software Engineer in Mainframe Technology in CSC India.