MULTI-SPECTRAL SATELLITE IMAGE CLASSIFICATION USING AN EVOLVING NEURAL NETWORK APPROACH

S. Suresh*, V. Mani**, S.N. Omkar** and N. Sundararajan*

Abstract

This paper investigates a new evolving neural network classifier based on real coded genetic algorithm for automatic multi-spectral satellite image classification (land cover mapping problem). The evolving neural network classifier is designed using hybrid genetic operators with classification accuracy as a measure of performance. The evolving neural network methodology is implemented in Pentium clusters. The proposed methodology searches for the best neural network architecture and its connection weights for a given set of training patterns. The performance of the proposed evolving neural network based classifier is evaluated for Level-II classifier model using the Landsat 7 Thematic Mapper high resolution imagery. After evolving the neural network at pixel level, the system performance is tested with sites not seen during training. Results are compared with maximum likelihood classifier, gradient based fully connected multilayer perceptron and growing and pruning radial basis function classifier. The proposed classifier is more accurate, robust with respect to the noise in the input spectrum and also overcomes the common limitations of the standard neural based classifier models.

Keywords: *Land Cover Mapping, Multi-spectral Classification, Multilayer Perceptron Network, Growing and Pruning Radial Basis Function Network, Maximum Likelihood Method, Genetic Algorithm*

Introduction

Land cover mapping has emerged as a subject in itself because of its importance in management of resources on the planet of Earth. In land cover mapping, classification is regarded as a fundamental process, which transforms the multi-spectral satellite or remotely sensed images to usable geographic products. So far, several pattern classification algorithms have been adopted in remote sensing land cover mapping problems [1]. The evolution of pattern classification algorithms for land cover mapping problems can be broadly divided in three categories [2]. In the first category, the classifier models are developed based on many supervised and unsupervised classification algorithms such as supervised maximum-likelihood approach [3], [4], artificial neural networks [5]-[15], [24], decision tree [16], and genetic algorithm [17]. In the second category, classifier models consist of many novel-system level approaches that integrate the outputs of the fundamental classifier algorithms [18], [19]. In the last category, the classifier models are developed by exploiting the multiple data types or ancillary information such as texture, structural and spatial information [20]-[23].

Among various classifier models, the neural network based classifier models are the most promising approach for land cover mapping problem due to their ability to approximate the complex nonlinear function accurately and also provide the information on probability distribution. The first research work on neural network based classifier model for remotely' sensed imagery appeared in 1990 [3]. Since then, many researchers developed different neural network classifier models for different applications and data sources [5]-[15]. In most of the cases, the neural network classifiers can improve the classification accuracy by 10-30% compared to the traditional classifiers [6], [11]. In literature, neural network architectures such as multilayer perceptron network [5]-[10], radial basis function network [22]-[26], binary diamond network [27] and fuzzy-neural networks [19], [28] have been developed to tackle the multi/hyper spectral satellite data classification problem. The performance of neural classifier depends on the finite training samples and number of samples per class. The small number of training samples and the presence of imbalance in training set result in not knowing the input distribution completely. This affects

* School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
** Department of Aerospace Engineering, Indian Institute of Science, Bangalore-560 012, India, Email : omkar@aero.iisc.ernet.in
Manuscript received on 27 Dec 2005; Paper reviewed, revised and accepted on 09 May 2006

the performance of neural classifier considerably. Also, finding an appropriate number of hidden neurons to approximate the input-output distributions affects the performance considerably.

In remote sensing literature, multilayer perceptron (MLP) network is widely used rather than any other neural network architecture. The multilayer perceptron network is a fully connected layered feed forward neural network. For a given fully connected neural network model, the weight connection between the neurons are calculated using error backpropagation learning algorithm. The learning algorithm is based on the gradient descent approach where the connection weights in the network are adjusted continuously in order to minimize an error function. The commonly used error function is mean square error between the target and network outputs over the entire training set. Since backpropagation uses a gradient-descent procedure, the network may converge to a local minima, which represent sub-optimal solutions. In literature, many modifications in learning algorithms are described to overcome the local minima problem [29]-[31]. However, these algorithms are computationally intensive and also do not solve the many existing problems in neural network design as described below [32].

Multilayer perceptron, trained using back-propagation algorithms learn the complex input-output relationship in stages. In a typical learning process, the mean-square error decreases with an increasing number of epoches. Initially, the reduction in mean- square error is rapid and later it decreases slowly as the network reaches the local minima. The stopping criteria, learning rate, weight initialization, selection of activation functions, network size and scaling of input/output data set affects the learning process and have to be selected properly [5], [32]. In neural network approach, the 'best' network architecture is not known a priori. The performance of network depends on the neural network architecture and also depends on the training data set and method of presenting the training data in training process. Hence, the neural network design (network architecture selection and weight parameter estimation) for any given realworld application is formulated as a search problem.

The Genetic Algorithm (GA) is perhaps the most well-known of all evolution based search techniques [33], [34]. Genetic algorithms are developed in an attempt to explain the adaptive process of natural systems and to design artificial systems based upon these natural systems. Genetic algorithms are widely used to solve complex optimization problems where the number of parameters and constraints are large and the analytical solutions are difficult to obtain [33], [34]. In recent years, many schemes for combining genetic algorithms and neural networks have been proposed and tested. The combination of these two techniques is either supportive or collaborative or both. Supportive combinations typically involve using one of these methods to prepare data for consumption by the other. For example, genetic algorithms can be used to select the features and neural network will generate the classifier based on the selected features [35], [36]. Collaborative combinations typically involve using the genetic algorithms to determine the neural network weights [37], [38] or the network topology [39], [40] or both [39]. A complete survey of evolving neural network using genetic algorithm can be found in 41].

In this paper, we present a evolving multilayer perceptron neural network design using a real coded genetic algorithm (RCGA). The evolving neural network has two different set of genetic operators. The first set of genetic operators control the neural network architecture and the second set of genetic operators evolve the connection weights. The proposed RCGA based evolving neural network design in turn reduces the cost of implementing the neural network, in terms of hardware and processing time. The parallel evolving neural network is implemented in Pentium cluster. The processor in the cluster exchange the intermediate population using message passing interface (MPI) routine [42]. The parallel evolving neural network reduces the computation time required to obtain optimal neural network for a given training data considerably. The performance of the neural classifier models depends on selection of training data, the network architecture, objective function and stopping criterion. In the proposed model, overall classification accuracy is used as a measure to evolve the optimal/best neural architecture and it's weights. Hence, in this study, the evolving neural network is used to develop Level-II classifier model from the multi-spectral satellite image. The inputs to the classifier model is the multi-spectral band data of Landsat 7 Thematic Mapper (TM) image from the southern part of India. The performance of the classifier is analyzed using different imagery from the same region. For comparison purpose, the same satellite image is classified using a maximum likelihood classifier, a gradient based fully connected multilayer perceptron and a growing and pruning radial basis function network [43]. Overall, the evolving neural network classifier outperformed the other models by 5-25% and obtains approximately 90% classification accuracy.

The contribution of the present work consist of (1) introducing evolving neural network approach for multi-spectral satellite image classification problem; (2) introducing the usage of classification accuracy as a measure for evolving the network architecture; and (3) compare the performance with other classifier models. In addition, the other important issues such as computational complexity, number of parameters, influence of training data and effect of noise in spectral bands are also analyzed.

This paper is organized as follows: In Section - **The Neural Network Model**, we introduce the multilayer perceptron network and backpropagation learning algorithm, Section - **Evolving Neural Network Design**, presents the real-coded genetic algorithm based neural network design, describing the solution representation, genetic operators and other details. In Section - **Experimental Results and Discussion**, we present satellite data description and the experimental results for multi-spectral satellite image classification problem. Finally, we discuss the results and conclude in **Conclusion Section**.

The Neural Network Model

In this study, we consider a three layered feed forward neural network as shown in Fig.1. The network consist of N_0 input neurons, N_1 hidden neurons and N_2 output neurons. Each neuron in the hidden layer uses bipolar sigmoidal function as its activation function ($f[x]$) and each neuron in the output layer uses pure linear function ($p[x]$) as its activation function. The output of the neurons in the hidden and output layer are expressed as

$$y_i^1 = f\left[\sum_{j=1}^{N_0} w_{ij}^1 u_j\right], \quad i = 1, 2..., N^1 \tag{1}$$

$$y_i^2 = p\left[\sum_{j=1}^{N_1} w_{ij}^2 y_j^1\right], \quad i = 1, 2..., N^2 \tag{2}$$

where y_i^1 is the output of ith hidden neuron, y_i^2 is the output of ith output neuron, the term w_{ij}^1 is the weight connection between the ith hidden neuron and jth input neuron, the term w_{ij}^2 is the weight connection between the ith output neuron and jth hidden neuron and the vector $U = [u_1, u_2 \ldots, u_N 1]$ is input to the network. The activation functions $f[x]$ and $p[x]$ are defined as:

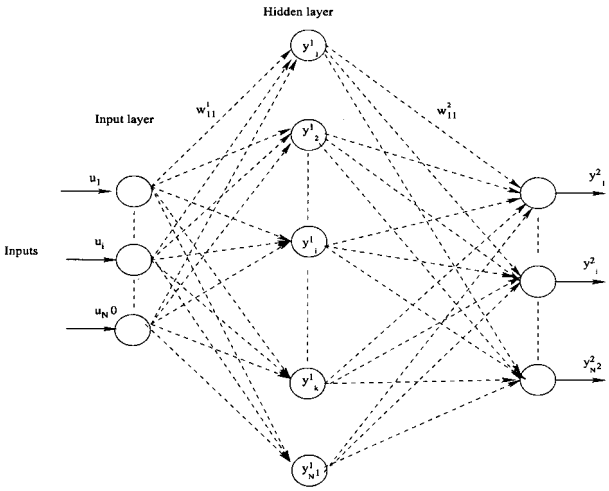

Fig.1 Architecture of three layered feed forward neural network

$$f[x] = \frac{1.0 - e^{-ax}}{1.0 + e^{-ax}} \tag{3}$$

$$p[x] = ax \tag{4}$$

where a is non-zero constant.

Let us assume that we have P patterns $\left(\{U^i, T^i\}\right)$ for training the neural network. Each sample in the training set has N^0 dimensional feature inputs and N^2 dimensional outputs. Let u_{ij}^2, $\forall\, j = 1, 2, \ldots, N^2$ be the outputs of neural network for input ith pattern and the target output vector be t_{ij}, $\forall, j = 1, 2 \ldots, N^2$. The training error for this neural network is defined as:

$$TE = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{N^2}\sum_{j=1}^{N^2}\left(t_{ij} - y_{ij}^2\right)^2 \tag{5}$$

The weights describing the input-output relationship between the training data set can be obtained by minimizing the training error. For a given fully connected neural network structure, there are number of learning algorithms available to determine the connection weights [29]-[31]. These algorithms are based on gradient descent approaches where the weights in the network are adjusted continuously in order to minimize the training error. The weight adaptation is stopped only when the training error is less than certain specified value. The trained network is tested with the new data set to study the generalization ability of the network. The selection of number of hidden

neurons and corresponding weight matrices affects the generalization performance considerably. The problem of finding optimal number of hidden neurons and weight matrices is difficult and is formulated as optimization problem.

Given: three layer perceptron network and input-output training data set.

To find: the optimal number of hidden neuron ($N^{1)}$ and the weight matrices (W^1, W^2) such that the training error is a minimum.

We can see that the above optimization problem is combinatorial (number of hidden neurons is integer and weight matrices are real number) in nature. Hence, in the next section, we present a real coded genetic algorithm based evolving neural network design.

Evolving Neural Network Design

Genetic Algorithms (GA's) are stochastic optimization algorithms based on the concepts of biological evolutionary theory [33], [34]. GA maintains a population of search nodes, which represents potential solutions to the optimization problem. Each search node in the population has an associated fitness value indicating the performance of the solution. In our problem, the search nodes are the neural network architectures. In general, GA, starts with a population of randomly generated search nodes and advances towards the better search nodes by applying genetic operators. During the successive generations, the search nodes with higher fitness (parents) are selected for genetic operations such as crossover and mutation. The new search nodes (offsprings) with higher fitness value will replace their parents in the next generation. The steps involved in the standard genetic algorithm are described below:

1. Randomly create initial population of search nodes.
2. Calculation of fitness for each search node.
3. Selection of the parents for genetic operations.
4. Generate new population of search nodes using genetic operators.
5. If termination criterion is satisfied then stop otherwise go to step 2.

A good representation scheme for solution is very important in obtaining best solution using GA for a given problem. The most commonly used solution representation is binary vector [33]. In most of the optimization problems, the design variables are in continuous domain (real number). Hence, it is natural to represent the solutions directly as real numbers since the representations of the solutions are very close to the natural formulation. This real number representation for solution is commonly referred as Real Coded Genetic Algorithm (RCGA). A detail discussion on advantage of representing solution using real number over binary variables is presented in [34].

A real coded genetic algorithm for any particular optimization problem must have the following components:

- String Representation
- Population Initialization
- Selection Function
- Genetic Operators
- Fitness Function
- Termination Function

Now, we describe these components of real coded genetic algorithm for evolving neural network design

String Representation

The string representation is the process of encoding a potential search node (solution) as a string. The string representation depends on structure of the problem in genetic algorithm framework and also depends on genetic operators used in the algorithms. In the earlier work on genetic algorithms [33], the string was restricted to only binary digits (0 and 1). It is shown in [34], that a natural representation of strings are more efficient and produce better results. Hence, in our studies, the string representation for search node, is a string of real numbers. The real numbers represent the weight connection between the neurons in the neural network. Each search node also has an unique integer number representing the number of hidden neurons (N^1) in the string. The search node is of length N^1 ($N^0 + N^2$).

For example, let us consider neural network architecture with single input neuron ($N^0 = 1$), five hidden neurons ($N^1 = 5$) and one output neuron ($N^2 = 1$). The string representing the above network architecture is

$$C_1 = \left[5, w_{11}^1, w_{12}^1, w_{13}^1, w_{14}^1, w_{15}^1, w_{11}^2, w_{21}^2, w_{31}^2, w_{41}^2, w_{51}^2\right] \quad (6)$$

The string representation and the corresponding weight connections are shown in Fig.2. In this string representation, we can represent all the combinations of the neural network architectures and the representation is unique.

Population Initialization

In genetic algorithm, initial population of N search nodes are generated using the most common random generation procedure. The size of population (N) and the method of initialization affects the convergence of the problem. Since genetic algorithm can iteratively improve the search nodes, some of the search nodes in the initial population can be potentially good solutions, with the remainder of the population being randomly generated. The population size (N) is typically problem-dependent and has to be determined through simulation. The initial population for evolving neural network is generated using N different neural networks trained for 50 epoches.

Selection Function

In genetic algorithm, the selection of a search node from the existing search nodes (population), to produce new search nodes for the next generations plays an important role. A probabilistic selection is performed based upon the fitness of search nodes, such that the better search nodes have a better chance of being selected for producing

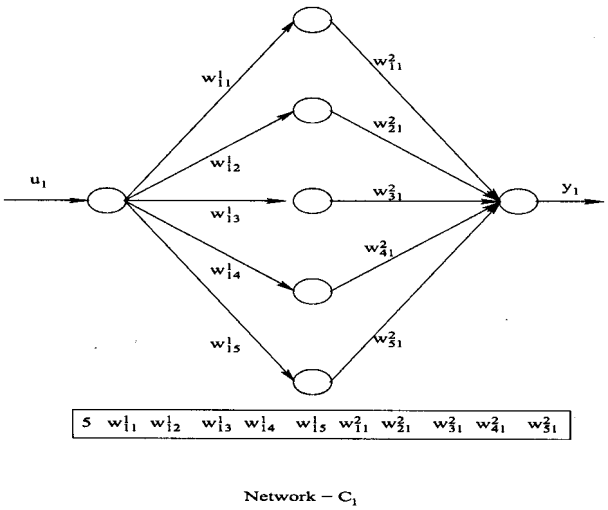

Fig.2 String representation for neural network architecture

new search nodes using genetic operators. It is possible that a search node in the population can be selected more than once for producing new search nodes. In literature [33], [34] several schemes such as roulette wheel selection and its extensions, scaling techniques, tournament and ranking methods are presented for the selection process. In our studies, normalized geometric ranking method given in [44] is used for the selection process.

Normalized Geometric Ranking Method: The search nodes are arranged in descending order of their fitness value. Let q be the selection probability for selecting best search node and r_j be the rank of jth search node in the partially ordered set. The probability of search node j being selected using normalized geometric ranking method is

$$s_j = q'(1-q)^{\tau_j - 1} \quad (7)$$

where $q' = \dfrac{q}{1-(1-q)^N}$ and N is the population size.

Genetic Operators

Genetic operators provide the basic search mechanism of the genetic algorithm. The operators are used to create new search nodes based on existing search nodes in the population. New search nodes are obtained by combining or rearranging parts of the old search nodes, and a new search node obtained may be a better solution to the optimization problem. These genetic operators are analogous to those which occur in the natural world: reproduction (crossover, or recombination) and mutation. The probability of these operators affects the efficacy of the genetic algorithm. The real-coded genetic operators used in our study are described below.

Crossover Operator: Crossover operator is a primary operator in genetic algorithm. The role of crossover operator is to recombine information from the two selected search nodes to produce two new search nodes. The crossover operator improves the diversity of the solution. In this paper, we present weight connection and network architecture based crossover operators. The operators which act on individual weight connections of the search nodes are called weight based crossover operators and operators which act on network architecture are called network based crossover operators. Now, we describe three weight connection based crossover operators and a network based crossover operator.

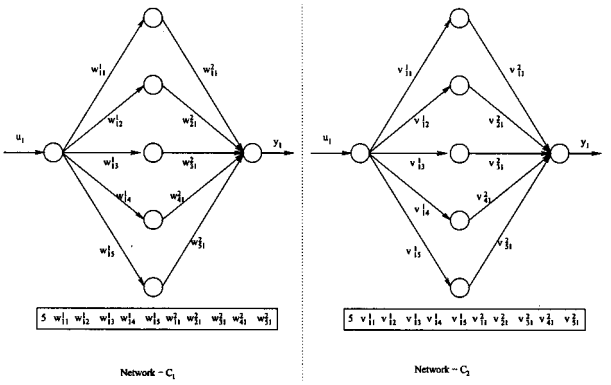

Fig.3 Selected string for crossover operation

Two Point Weight Crossover: Let C_1 and C_2 are the two search nodes selected for crossover operations as shown in Fig.3. This operator first selects two crossover points i and j randomly. Let $i < j$. The weights in between the crossover points are represented in bold faces.

$$C_1 = \left[5,\ w_{11}^1,\ \mathbf{w_{12}^1},\ \mathbf{w_{13}^1},\ \mathbf{w_{14}^1},\ \mathbf{w_{15}^1},\ \mathbf{w_{11}^2},\ w_{21}^2,\ w_{31}^2,\ w_{41}^2,\ w_{51}^2\right] \tag{8}$$

$$C_2 = \left[5,\ v_{11}^1,\ \mathbf{v_{12}^1},\ \mathbf{v_{13}^1},\ \mathbf{v_{14}^1},\ \mathbf{v_{15}^1},\ \mathbf{v_{11}^2},\ v_{21}^2,\ v_{31}^2,\ v_{41}^2,\ v_{51}^2\right] \tag{9}$$

The two new search nodes H_1 and H_2 are generated by exchanging the selected weights.

$$H_1 = \left[5,\ w_{11}^1,\ \mathbf{v_{12}^1},\ \mathbf{v_{13}^1},\ \mathbf{v_{14}^1},\ \mathbf{v_{15}^1},\ \mathbf{v_{11}^2},\ w_{21}^2,\ w_{31}^2,\ w_{41}^2,\ w_{51}^2\right] \tag{10}$$

$$H_2 = \left[5,\ v_{11}^1,\ \mathbf{w_{12}^1},\ \mathbf{w_{13}^1},\ \mathbf{w_{14}^1},\ \mathbf{w_{15}^1},\ \mathbf{w_{11}^2},\ v_{21}^2,\ v_{31}^2,\ v_{41}^2,\ v_{51}^2\right] \tag{11}$$

Uniform Weight Crossover: In this operator, the crossover sites are selected randomly. Let C_1 and C_2 are the two search nodes selected for crossover operations and the crossover points are represented in bold faces.

$$C_1 = \left[5,\ w_{11}^1,\ \mathbf{w_{12}^1},\ w_{13}^1,\ w_{14}^1,\ w_{15}^1,\ \mathbf{w_{11}^2},\ w_{21}^2,\ w_{31}^2,\ \mathbf{w_{41}^2},\ w_{51}^2\right] \tag{12}$$

$$C_2 = \left[5,\ v_{11}^1,\ \mathbf{v_{12}^1},\ v_{13}^1,\ v_{14}^1,\ v_{15}^1,\ \mathbf{v_{11}^2},\ v_{21}^2,\ v_{31}^2,\ \mathbf{v_{41}^2},\ v_{51}^2\right] \tag{13}$$

The two new search nodes H_1 and H_2 are generated by exchanging the selection weights.

$$H_1 = \left[5,\ w_{11}^1,\ \mathbf{v_{12}^1},\ w_{13}^1,\ w_{14}^1,\ w_{15}^1,\ \mathbf{v_{11}^2},\ w_{21}^2,\ w_{31}^2,\ \mathbf{v_{41}^2},\ w_{51}^2\right] \tag{14}$$

$$H_2 = \left[5,\ v_{11}^1,\ \mathbf{w_{12}^1},\ v_{13}^1,\ v_{14}^1,\ v_{15}^1,\ \mathbf{w_{11}^2},\ v_{21}^2,\ v_{31}^2,\ \mathbf{w_{41}^2},\ v_{51}^2\right] \tag{15}$$

Averaging Weight Crossover: It is a commonly used operator in real coded genetic algorithm and it generates new solutions by averaging the two parents. Two new solutions H_1 and H_2 are:

$$H_1 = C_1 + \beta\,(C_1 - C_2)$$

$$H_2 = C_2 + \beta\,(C_2 - C_1)$$

where β is a scalar value in the range of $(0 \le \beta \le 1)$. In our simulation studies, β is set to 0.3.

Heuristic Network Crossover: The operator randomly select hidden neurons as crossover points. The input and output weight connection between the selected hidden neurons are modified to generate the new search nodes. Let the second hidden neuron be selected as crossover point. The new weights generated are

$$H_1 = C_1 \pm \gamma w_m \frac{C_1 - C_2}{\|C_1 - C_2\|} \tag{16}$$

$$H_2 = C_2 \pm \gamma w_m \frac{C_2 - C_1}{\|C_2 - C_1\|} \tag{17}$$

where w_m range of the weight vectors and γ is positive constant. In our experiment, *range* and γ are set to 100 and 0.1 respectively.

Hybrid Crossover: We have presented four types of crossover operators. The performance of these operators in terms of convergence to optimal solution depends on the problem. One type of crossover operator which performs well for one problem may not perform well for another problem. Hence, many research works are carried out to study the effect of combining crossover operators in a genetic algorithm for a given problem [45]. Hybrid

crossovers are a simple way of combining different crossover operators. The hybrid crossover operator use different kinds of crossover operators to produce diverse offsprings from the same parents. The hybrid crossover operator presented in this study generates eight offsprings for each pair of parents by the four crossover operators. The most promising offsprings of the eight substitute their parents in the population.

Mutation Operators: The mutation operator alters one solution to produce a new solution. The mutation operator is needed to ensure diversity in the population, and to overcome the premature convergence and local minima problems. Similar to crossover operator, here also we have weight and network based mutation operators. Now, we describe different mutation operators used in this study.

Let us assume that C_1 is the parent selected for the mutation operation and w_{12}^1 is the mutation weight.

Random Weight Mutation: Let w_m be the range of weight. The new weight h_{12}^1 is a random number in the range w_m.

Non-uniform Weight Mutation: If this operator is applied in a generation t and G is the maximum number of generations, then

$$h_{12}^1 = w_{12}^1 + \Delta\left(t, w_{\min} - w_{12}^1\right), \quad if\ \gamma = 0 \tag{18}$$

$$h_{12}^1 = w_{12}^1 + \Delta\left(t, w_{\max} - w_{12}^1\right), \quad if\ \gamma = 1 \tag{19}$$

where $\gamma \in [0,1]$ and

$$\Delta(t, y) = y\left(1 - r^{1 - \frac{t}{G}^b}\right) \tag{20}$$

where r is the random number between the interval [0, 1] and b is the parameter that determines the degree of dependency. This function gives a value in the range [0, y] such that the probability of returning a number close to zero increases as the algorithm advances [34].

Add/Delete Network Mutation: This operator adds or deletes hidden neuron in the selected parent. In case of adding a hidden neuron, random numbers in the range w_m are assigned to the weights connecting the hidden neuron with input and output neurons. In case of deleting a hidden neuron, zero values are assigned to the weights connecting the hidden neuron with the input and output neurons.

Fitness Function

Fitness is the driving force in genetic algorithms. In the evolving neural network problem, fitness function assigns a fitness (value) to each of the search node in a generation. Fitness function must be capable of evaluating every search node in the search space. Genetic algorithm does not know anything about the problem domain or fitness function. The only information used in the execution of genetic algorithm is the observed values of the fitness function (i.e., fitness) of the individual search nodes actually present in the population. Genetic algorithm is guided by the fitness value to search for the most efficient search node to solve the given problem.

The fitness function in the land cover mapping problem is based on the overall classification efficiency. The objective of the evolving network is to maximize the classification accuracy (CA). Hence, the fitness function is

$$CA = \frac{N_p}{P} \times 100 \tag{21}$$

where N_p is the number of correctly classified patterns and P is the total number of training patterns.

Termination Function

In a genetic algorithm, in each generation, search nodes are selected on the basis of their fitness and subject to genetic operations such as crossover and mutation. The evolution process of successive generations continues until a termination criterion is satisfied. The most frequently used stopping criterion are population convergence and a specified maximum number of generations. Population convergence criteria for our problem is that, all the search nodes in the population are the same in four successive generations. This search node is the optimal or best search node obtained using the genetic algorithm. On the other hand, if the maximum number of generations is reached, then we have a population of search nodes with different fitness. The best search node is the one in the population with a maximum fitness.

We can see that the above evolving neural network use the survival of fittest strategy by passing the 'good' search nodes to the next generation of search nodes, and combining different search nodes to explore new search nodes. So we call the search node obtained upon termination from genetic algorithm as the 'best' search node. We know that the 'best' search node, is the best neural network architecture for a given land cover mapping problem. Best implies that for this remote sensing training data, the classification efficiency obtained is maximum.

Experimental Results and Discussion

In this section, we present the experimental results obtained for multi-spectral satellite image classification problem. First, we describe characteristic of the satellite data, method of obtaining the ground truth and training and testing data selection. Next, we present several experimental results obtained from different classifier models. Finally, we present the comparison among different classifier models by analyzing their performances.

Satellite Image Acquisition

In our experimental study, a portion of high resolution, multi-spectral Landsat 7 Thematic Mapper images acquired form southern region of India is used. The wavelength range for the thematic mapper sensor is from the visible, through the mid-IR, into the thermal-IR portion of the electromagnetic spectrum. The characteristic of the seven bands and their spectral resolution are given in Table-1 In our experimental study, the band 6 which has spatial resolution of 120 m is not used. The portion of Landsat image is 15 x 15.75 sq.km (500 x 526 pixels) and has 30 m spatial resolution, which corresponds to pixel spacing of 30 m. The satellite image is radiometrically and geometrically corrected using the satellite model and platform/ephemeris information. The image is further rotated and aligned to a user-defined map projection, using ground control points to improve the satellite model. The color composite image of the study area and corresponding ground truth are given in Fig.4. The satellite image was taken when there is no cloud. Even though the image was taken during no cloud day, small and minor cloud may

Table-1 : Thematic Mapper band resolution

Band No.	Resolution	Spectral Range	Type
1	30 m	0.45-0.52 μm	visible blue-green
2	30 m	0.45-0.52 μm	visible green
3	30 m	0.45-0.52 μm	visible red
4	30 m	0.45-0.52 μm	near infrared
5	30 m	0.45-0.52 μm	mid infrared
6	120 m	0.45-0.52 μm	far infrared
7	30 m	0.45-0.52 μm	mid infrared

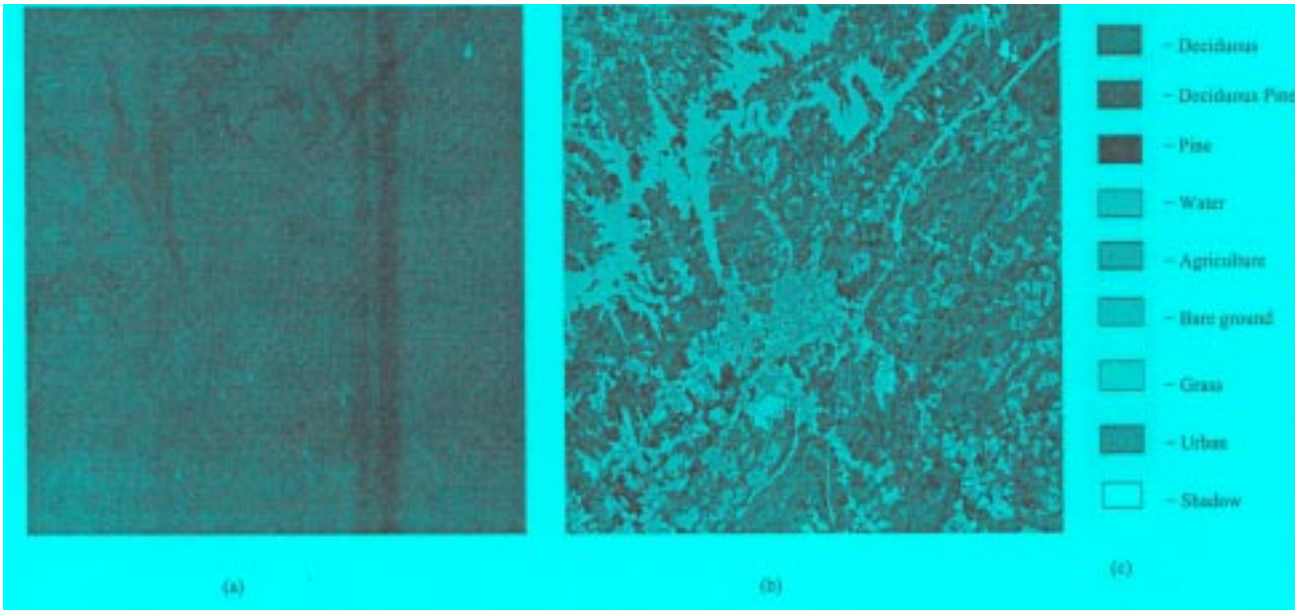

Fig.4 Landsat image : a) Colour composite image, b) Ground truth and c) Legend

present but not visible in the image. The presence of small and minor cloud may affect the performance of the classifier models. The presence of cloud is considered as a process noise in the spectral data. Also, in order to simulate the real situation such as spatial variability of the spectral signature, uncertainty on the ground truth, noise at the sensor, and observation noise, we introduce different amounts and sources of noise (Gaussian and uniform) in the test set (not training set). This study clearly indicates the robustness of trained classifier when subjected to change in environment.

Ground Truth Description and Training Pixels Selection

The aim of the study to develop neural network classifier to distinguish forest, vegetation, water and built-up land. The primary analysis on the satellite data indicated that these four classes are not sufficient. Therefore, the four classes are further divided into 9 sub-classes. The forest class is sub-divided into deciduous, deciduous-pine and pine. Vegetation is further divided into agriculture, bare ground (agriculture land without water) and grass. The built-up land is sub-divided into urban and shadow. The classes and sub-classes are defined based on the Level-II standards and the map obtained using this approach can be directly used for practical applications.

The ground truth is prepared for the Landsat image by visual inspection, auxiliary data (maps and aerial photography) and field work. The 9 classes are used to represent the ground truth and image with ground truth is shown in Fig.4(b). The legends for each classes are shown in Fig,4(c). For development of supervised neural classifiers, it is necessary to define the classes and select some of the region in the image for training of each classes. The training data for each class are selected from the region where the field work has been carried-out. The details description of the class, number of pixels obtained from field work and number of validation pixels are given in Table-2. The ground truth obtained for 10,000 pixels from field work are used to develop neural classifier models. The evolved classifier model is validated using the remaining pixels. The ground truth for these pixels are obtained from visual inspection and aerial photography.

Experimental Results

In this section, we discuss the experimental results with evolving multilayer perceptron network (EMLP) for multi-spectral satellite image classification problem and

Table-2 : Description of classes and ground truth available form field work and validation

Class No.	Class		Pixels form Field Work	Pixels for Validation
	Level-I	Level-II		
O1		Deciduous	1200	71228
O2	Forest	Deciduous-Pine	1300	80848
O3		Pine	1100	24911
O4	Water		1075	12518
O5			1100	23070
O6	Vegetation	Bare Ground	1100	26986
O7		Grass	1050	7400
O8		Urban	1075	11636
O9	Built-up	Shadow	1000	3547

compare the results with the maximum likelihood classifier (MLC), fully connected multilayer perceptron classifier (MLP) and growing and pruning radial basis function network classifier (GAP-RBFN). For multilayer perceptron based classifier models, three layer structure with one hidden layer are selected. The performance of the multilayer per- ceptron network depends on the training patterns [5]. Hence, we would like to study results in the following manner:

- **Case I: Effect of Training Data Selection**. In this case, we develop two fully connected multilayer perceptron classifier using randomly selected training data and training data selected using the procedure given in [5].
- **Case II: Effect of Optimal Neural Network Architecture**. In this case, we develop two multilayer perceptron classifier models using proposed evolving neural network technique with randomly selected training data and training data selected using the procedure given in [5].
- **Case III:** An optimal neural network classifier is developed using the growing and pruning radial. basis function network technique given in [43].

Finally, the performance of the classifier models are compared and also the effect of noise in spectral band is

analyzed. The following performance parameters are used to study the behavior of the classifier.

Performance Measures: The classification/confusion matrix (Q) is used to obtain the statistical measures for both the class-level and global performances of the classifier. Class-level performance is indicated by the percentage classification which tells us how many samples belonging to a particular class have been correctly classified. The percentage classification η_i for class c_i is

$$\eta_i = \frac{q_{ii}}{N_i^T} \tag{22}$$

where q_{ii} is the number of correctly classified samples and N_i^T is the number of samples for the class c_i in the testing data set.

The global performance measures are the average (η_a) and overall (η_o) efficiency, which are defined as

$$\eta_a = \frac{1}{C}\sum_{i=1}^{C}\eta_i \tag{23}$$

$$\eta_o = \frac{1}{N^T}\sum_{i=1}^{C} q_{ii} \tag{24}$$

where C the total number of classes and N^T is the number of samples in the testing set.

Case I. Effect of Training Data Selection: The six band Landsat data and a bias input are used to develop a 9 class classifier models. The general architecture of the multilayer perceptron network considered in this study is shown in Fig.1. In the hidden layer tangential sigmoid function is used as activation function while in the output layer linear function is used as activation function. To study the effect of training data selection, the learning rate, number of epochs, and number of hidden neurons are kept at 0.2, 200, and 14 respectively.

Random Selection Process: To develop a classifier model, 1000 patterns from each class are randomly selected. The neural network classifier is trained for 200 epochs with 0.2 learning rate. After training process is completed, the neural network classifier model is tested with the complete imagery. Table-3 shows the classification matrix for random selection process. The average accuracy and overall accuracy are 75.57% and 71.86% respectively. From the classification matrix, we can see that the following classes has high classification accuracy O3, O4 and O7, and the most troublesome classes are O2, O5 and O8. From Table-3, we can see that the deciduous-pine class (O2) has strong overlap with pine (O3) and vegetation classes (O5-O7). Similarly, the urban class (O8) has strong overlap with the vegetation classes. The above facts influence the performance of the MLP classifier model developed using random training data selection process.

Table-3 : The classification matrix for multilayer perceptron classifier with random selection process

		Classification Results									Actual No. Patterns
		O1	O2	O3	O4	O5	O6	O7	O8	O9	
Ground Truth	O1	**54769**	12490	1772	0	2029	2	165	0	1	71228
	O2	1081	**46539**	14303	4331	2875	1617	8934	104	1064	80848
	O3	589	1820	**22489**	2	0	0	0	0	11	24911
	O4	0	38	117	**21864**	1	22	63	4	961	23070
	O5	5475	799	13	1	**17776**	42	2880	0	0	26986
	O6	0	130	4	75	172	**5466**	741	796	1	7400
	O7	131	471	0	54	536	1036	**10211**	77	2	12518
	O8	0	127	16	3	1339	1308	2602	**6227**	14	11636
	O8	0	7	1	463	0	2	0	30	**3044**	3547

Table-4 : The classification matrix for multilayer perceptron classifier with Yoshida's selection process

		Classification Results									Actual No. Patterns
		O1	O2	O3	O4	O5	O6	O7	O8	O9	
Ground Truth	O1	**63805**	2452	1898	0	2841	0	149	79	4	71228
	O2	5265	**53674**	9146	169	1782	2523	5303	2630	356	80848
	O3	458	612	**23831**	0	0	0	0	10	0	24911
	O4	4	973	76	**20860**	3	226	73	9	846	23070
	O5	1139	244	31	0	**22740**	24	2017	576	215	26986
	O6	0	4	4	0	87	**5917**	422	954	12	7400
	O7	0	15	0	0	504	1583	**9940**	475	1	12518
	O8	0	12	2	0	255	983	170	**10202**	12	11636
	O8	0	34	0	72	0	13	3	4	**3421**	3547

Yoshida's Selection Process: The training patterns to develop the classifier are selected using the procedure presented in [5]. In this procedure, first the high resolution imagery is clustered using Kohonen's self-organizing map [5]. Then, using the ground truth and cluster information, 9000 training patterns are selected. The structure of the multilayer perceptron is the same as in the case of random selection process. The neural network is trained for 200 epochs with learning rate 0.2. After training process is completed, the neural network classifier model is tested with the complete imagery. Table-4 shows the classification matrix for Yoshida's selection process. The average and overall accuracy are 85.53% and 81.78% respectively. From the above results, it can be easily seen that the Yoshida's selection process has better performance than the random selection process. This is due to the fact that Yoshida's selection process, select the training patterns in the boundary between the classes. Hence, the MLP classifier model is able to capture the decision boundary between the classes accurately. From the classification matrix, we can see that the troublesome classes O5 and O8 in random selection process is improved considerably in Yoshida's selection. Even though there is an increase in the performance, the overall accuracy is not improving considerably. This is due to poor performance in the deciduous pine class (O2). The deciduous-pine has strong overlap between the pine and vegetation.

The performance of the multilayer perceptron classifier models not only depends on the training data but also depends on the stopping criteria, learning rate, weight initialization, selection of activation function, network size, error function and scaling of input/output data set.

Table-5 : Parameters used in simulation studies

Parameters	Description	Value
N	Polulation size	20
S_m	Mutation probability	0.05
S_c	Crossover probability	0.6
M	Maximum number of generations	500
q	Selection probability	0.08

Hence, in the next case, we present the results based on evolving multilayer perceptron network technique presented in this paper.

Case. II. Effect of Optimal Neural Network Architecture: The evolving neural network technique to obtain the best multilayer perceptron network classifier models for the training patterns selected using random and Yoshida's approach was implemented and tested in on a Pentium clusters. In parallel evolving neural network approach, the selected population (intermediate population) for genetic operation is divided into 4 sub-populations of equal sizes (4 is the processors in the cluster). After receiving the respective sub-populations, each processor in the cluster performs crossover, mutation operations. At the end of every generation, the processors in the cluster broadcast the new population to other processors using message passing interface sub-routine. The above steps are repeated, until the termination criterion is satisfied. The evolving neural network technique use the following parameters given in Table-5 in all our simulations. The convergence of the evolving neural network to the best

netWork is depends on these parameters and are usually determined by trail-and-error. The advantage with evolving neural network technique is that it starts with a random search nodes and modify the search nodes in successive generations, and the best search node is obtained. Search node is the best multilayer perceptron classifier model, for a given training patterns.

Random Selection Process: First, we have trained 20 different neural network architectures for 10 epochs using randomly selected training patterns. These networks forms the initial population for the evolving neural network. The best multilayer perceptron classifier model obtained in this evolution process is having 18 hidden neurons with partially connected network. The evolved classifier model is tested with the complete imagery. The classification matrix for random selection process is given in Table-6. The average and overall accuracy are 85.28% and 89.28% respectively. From the above results, it can be easily seen that the evolving multilayer perceptron network for random selection process has better performance than the fixed fully connected multilayer perceptron classifier models presented in the previous case. Even though there is a substantial increase in the overall accuracy, the average accuracy is not improving considerably. From classification matrix, it can be observed that the patterns in the urban and shadow classes are misclassified as deciduous-pine and water. These classes reduces the average classification accuracy.

Yoshida's Selection Process: As mentioned in the random selection process, here also 20 different neural network architectures are trained for 10 epochs. These networks forms the initial population for the evolving neural network. The best multilayer perceptron classifier model obtained in this evolution process is having ___hidden neurons with partially connected network. The evolved classifier model is tested with the complete imagery. The classification matrix is given in Table-7. The average and overall accuracy are 87.38% and 87.34% respectively. From Table-4 and 7, we can observed that the overlap between the deciduous, deciduous-pine, and pine classes are captured considerably in evolving multilayer perceptron network classifier model than the fixed fully connected MLP classifier model. Hence, the average and overall accuracy of the evolving network classifier model is better than the fixed fully connected classifier model. In the process of evolution, the patterns belongs to urban are misclassified as vegetation. This affect the overall accuracy of the classifier model.

Case III. Growing and Pruning Radial Basis Function Network: In this section, we develop a classifier model using growing and pruning radial basis function network (GAP-RBFN) [43]. In this approach, sequential learning algorithm is used to approximate the nonlinear relationship between the band data and the corresponding classes. In sequential learning algorithm, series of training data samples are presented one by one to the network. Initially, the network begin with no hidden neurons. Based on the training data, new hidden neurons are added/deleted based on growing and pruning criterion explained in [43]. After the learning process is completed, the network evolves with H hidden neurons and the output of the network is obtained as

Table-6 : The classification matrix for evolving neural network with random selection process

		Pattern Recognition Results									Actual No. Patterns
		O1	O2	O3	O4	O5	O6	O7	O8	O9	
Ground Truth	O1	**66668**	2771	319	0	1469	0	1	0	0	71228
	O2	2024	**74338**	1629	188	837	256	1300	44	232	80848
	O3	1016	2197	**21697**	0	0	0	0	1	0	24911
	O4	10	1297	5	**21068**	2	42	17	2	627	23070
	O5	2384	1067	1	0	**22930**	2	578	24	0	26986
	O6	1	253	0	0	142	**6004**	669	330	1	7400
	O7	0	448	0	0	1022	631	**10301**	116	0	12518
	O8	12	693	0	0	1287	888	790	**7963**	3	11636
	O8	11	122	0	157	0	116	29	32	**3080**	3547

Table-7 : The classification matrix for evolving neural network with Yoshida's selection process

		Pattern Recognition Results									Actual No. Patterns
		O1	O2	O3	O4	O5	O6	O7	O8	O9	
Ground Truth	O1	**62358**	2645	1800	0	4424	0	0	0	1	71228
	O2	1451	**68370**	3629	180	2825	1171	2742	118	362	80848
	O3	71	856	**23982**	0	0	0	0	1	1	24911
	O4	16	1076	10	**20943**	8	135	45	1	836	23070
	O5	519	269	1	0	**25003**	12	1156	26	0	26986
	O6	0	23	0	0	100	**6791**	291	190	5	7400
	O7	0	42	0	0	535	1214	**10617**	110	0	12518
	O8	2	198	0	0	1236	1854	724	**7616**	6	11636
	O8	5	22	0	77	0	108	22	27	**3286**	3547

Table-8 : Performance of different classifier models

CA in %	MLC		MLP		EMLP		GAP-RBFN	
	Random Selection	Yoshida's Selection	Random Selection	Yoshida's Selection	Random Selection	Yoshida's Selection	Random Selection	Yoshida's Selection
O1	87.83	86.73	76.83	89.50	**93.52**	87.47	77.23	90.12
O2	26.05	27.57	57.56	66.39	**91.95**	84.57	60.11	70.12
O3	92.22	91.65	90.28	95.66	87.10	**96.27**	89.91	94.23
O4	93.88	93.37	**94.77**	90.42	91.32	90.78	94.51	89.99
O5	76.75	80.15	65.87	84.27	84.97	**92.65**	67.12	80.12
O6	50.69	59.46	73.86	79.96	81.14	**91.77**	79.01	81.45
O7	74.92	67.69	81.57	79.41	82.29	**84.81**	82.23	82.27
O8	58.34	56.82	53.94	87.68	68.43	65.45	85.45	**90.92**
O9	10.77	11.64	85.82	**96.45**	86.83	92.64	88.91	91.23
Average	63.50	63.90	75.61	85.53	85.28	**87.38**	80.49	85.61
Overall	64.59	64.85	71.88	81.78	**89.28**	87.34	74.45	82.73

$$y_i = \sum_{i=1}^{H} w_{ij}\phi_i(X),\ j = 1, 2, ..., N^2 \qquad (25)$$

where w_{ij} is the weight connection between the ith hidden neuron and jth output neuron. The term $\phi_i(X)$ is the response of the ith hidden neuron for given band input data

$$\phi_i(X) = \exp\left(-\frac{||X - \mu_i||^2}{\sigma_i^2}\right) \qquad (26)$$

In this case also, we develop networks based on randomly selected training data and training data selected based on Yoshida's selection process. The selected training patterns are rearranged such that window of 100 samples have patterns from all classes. The following are the values of the parameters selected in our study are: $\in_{min} = 0.004$, $\in_{max} = 0.5$, $\kappa = 0.75$ and $\eta = 0.9$. The expected accuracy in the simulation study is 0.0001. The number of hidden neurons required to capture the relationship for random and Yoshida's selection process are 270 and 220. The performance of theGAP-RBFN classifier models are given in Table-8. From Table-8, we can see that the GAP-RBFN evolved using Yoshida's selection

process has higher classification accuracy than the random selection process.

Comparative Analysis and Discussion

For comparative study, we also develop classifier model using maximum likelihood approach [46]. For maximum likelihood approach, the training data define the parameters of the probability distribution of the classes. Similar to other models, maximum likelihood classifier is also developed using training data obtained from random and Yoshida's selection process. The results obtained using the classifier models are listed in the Table-8. From Table, one easily observe that there is slight improvement in the classification accuracy based on training data selection for maximum likelihood classifier. From the result, we can easily see that the performance of the maximum likelihood classifiers are approximately 10-25% less than the other models. By looking at the classification accuracy of individual classes, the maximum likelihood classifier are comparable with other classifier models for classes O1, O3, O4, O5 and O7, where as O2, O6, O8, and O9 are troublesome classes.

To analysis the performance of the classifier models, classification accuracy of individual classes, average and overall accuracy for maximum likelihood classifier models (MLC), fixed fully connected multilayer perceptron classifier models (MLP), evolving network classifier models (EMLP) and GAP-RBFN classifier models are given in Table-8. From the table, it can be clearly observed that the evolving neural network based classifier models have better performance over other classifiers.

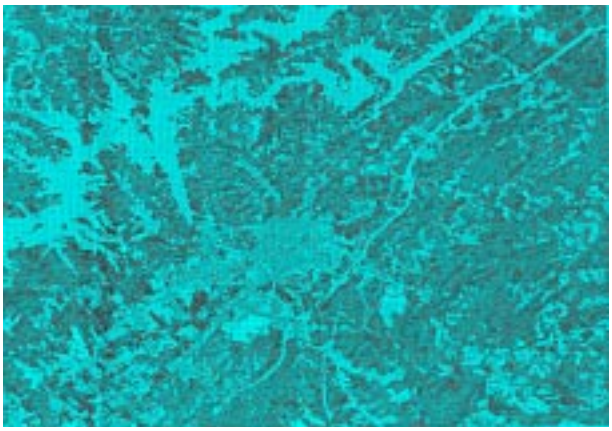

Fig.5 Classification results for evolving neural network classifier model using random selection process

The classifier model based on random selection shows better performance in classes O1, O2, O4 and O8 where as the Yoshida's selection shows better performance in classes O3, O5, O6, O7 and O9. Since, the number of patterns in first two classes are high and the evolving neural network with random selection has better classification accuracy in the first two classes. Hence, the overall accuracy is better than the Yoshida's selection. The map generated using the results obtained from the evolving network classifier based on random selection is shown in Fig.5. From Fig.s 4(b) and 5, one can easily observe that the land cover map obtained using evolving neural network maps are more suitable for practical applications.

Effect of Noise: Now, we analyze the robustness of the classifier models in the presence of additive gaussian noise to all spectral band. In this experiment, we add different amount of additive gaussian noises to the test patterns and study the behavior of the classifier models trained using clean patterns (with-out noise). These experiments provide real situations where the acquired satellite data subjected to sensor noise, observation noise, cloud correction, uncertainty in ground truth, uncertainty in spectral signature and etc. The overall accuracy of the different classifiers when subjected to different signal-to-noise ratios (SNR) are shown in Fig.6. For different signal-to-noise ratio (between 1-40 db), the experiments are carried out for several times and the average performance is shown in the Fig.6. From the figure, we can observe that theEMLP and GAP-RBFN classifiers have better performance than the MLP and MLC classifier models. This behavior is observed both for Gaussian and uniform additive noise, but it is more evident in the former. From the study,. it should be noted here, that the degradation in performance appear only at a very low SNR, i.e., when-

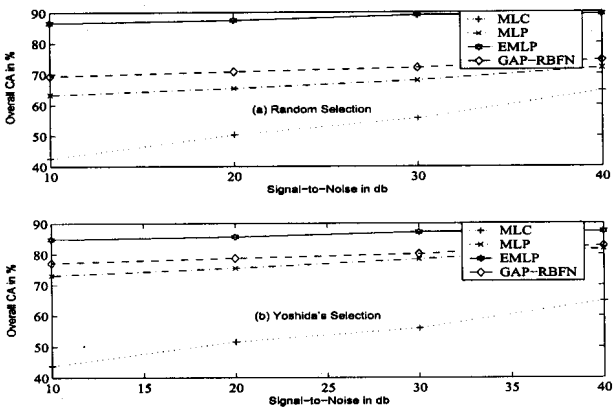

Fig.6 Variation in overall classification accuracy for different SNR

Table-9 : Complexities of the classifier models

Training Data Selection	Classifier Model	Hidden Neurons	Total No. Parameters	Training Time in Minute	Objective Function	Overall Accuracy
	MLC	-	387	4	-	64.59
	MLP	14	224	843	MSE	71.88
	EMLP	18	271	1235^{+}	CA	89.28
	GAP-RBFN	270	4320	1649	RMSE	74.45
	MLC	-	387	4	-	64.85
	MLP	14	224	845	MSE	81.78
	EMLP	19	289	1181^{+}	CA	87.34
	GAP-RBFN	220	3520	1512	RMSE	82.73

+ Time taken to evolve optimal neural network in single processor system

ever the amount of noise is extremely high, which does not represent realistic situations.

Complexity: The complexities of the different classifier models are analyzed based on number of hidden neurons, number of parameters and training time. The complexities of the MLC, MLP, EMLP and GAP-RBFN classifier models for random and Yoshida's method of training data selection process are given in Table-9. From the above table, we can observe that the training time for MLC classifier is less when compared to other classifier models. But, the performance of the MLC is approximately 10-25% less than the other classifier models. The number of parameters and training time for MLP based classifier models are less when compared to EMLP and GAP-RBFN. However, the overall performance of EMLP is better than the MLP classifier model. Since, the EMLP search for optimal/best number of hidden neurons for partially connected network architecture with classification accuracy (CA) as objective function, it's performance is better than the other classifier models.

Conclusion

In this paper, we have presented the evolving neural network based Level II classifier model for Landsat 7 Thematic Mapper high resolution imagery. The six band data are used as inputs to the neural classifier model and it is implemented in Pentium cluster environment. The evolving neural network adopts a hybrid real coded genetic algorithm methodology and the genetic operators are carefully designed to optimize the neural network structure and its connection weights. The proposed methodology avoids the premature convergence and permutation problems in multilayer perceptron network with back propagation learning algorithm. Since, the evolving neural network approach uses classification accuracy as objective function, its performance is better than the other classifier models. The experimental studies clearly indicate that the evolving neural network classifier model out perform the classifier models based on fixed multilayer perceptron network and growing and pruning radial basis function network. The robustness study of the classifier model shows that the evolving neural network classifier is not sensitive to additive gaussian noise up to 20db in all spectral band.

Acknowledgement

This work was in part supported by the grant of Indian Space Research Organization (ISRO), India and Satellite Technology Cell, Indian Institute of Science, Bangalore, India. Also, the last author acknowledges gratefully the support extended to him by the Department of Aerospace Engineering, Indian Institute of Science, Bangalore to visit the Institute as visiting Professor during which this study was undertaken.

References

1. Townshend, J., Justice, C., Li, W., Gurney, C. and McManus, J., "Global Land Cover Classification by Remote Sensing: Present Capabilities and Future Possibilities", Remote Sensing Environ., Vol. 35, pp. 243-255, 1991.

2. Wilkinson, G.G., "Results and Implications of a Study of Fifteen years of Satellite Image Classifica-

tion Experiments", IEEE. Trans. Geosci. Remote Sens., Vol. 43, No. 3, pp. 433-440, 2005.

3. Benediktsson, J.A., Swain, P.H. and Ersoy, O.K., "Neural Network Approaches Versus Statistical Methods in Classification of Multisource Remote Sensing Data", IEEE Trans. Geosci. Remote Sens., Vol. 28, No. 4, pp. 540-552,1990.

4. Frizzelle, B.G. and Moddy, A., "Mapping Continuous Distribution of Land Cover: A Comparison of Maximum-Likelihood Estimation and Artificial Neural Networks", Photogramm. Eng. Remote Sens., Vol. 67, No. 6, pp. 693-705, 2001.

5. Yoshida, T. and Omattu, S., "Neural Network Approach to Land Cover Mapping", IEEE Trans. Geosci. Remote Sens., Vol. 32, No. 5, pp. 1103-1109, 1994.

6. Kavzoglu, T. and Mather, P.M., "The Use of Backpropagating Artificial Neural Networks in Land Cover Classification", Int. Journal of Remote Sensing, pp. 1-32, 2002.

7. Foody, G.M., "Supervised Image Classification by MLP and RBF Neural Networks with and without an Exhaustively Defined Set of Classes," Int. J. Remote Sens., Vol. 25, No. 15, pp. 3091-3104, 2004.

8. Kanellopoulos, I. and Wilkinson, G.G., "Stratiges and Best Practice for Neural Network Image Classification", Int. J. Remote Sens., Vol.18, No. 4, pp. 711-725,1997.

9. Ji, C.Y., "Land-use Classification of Remotely Sensed Data Using Kohonen Self-organizing Feature Map Neural Networks", Photogramm. Eng. Remote Sens., Vol. 66, No. 12, pp. 1451-1460, 2000.

10. Bischof, H and Leona, A., "Finding Optimal Neural Networks for Land Use Classification", IEEE Trans. GeoSci. Remote Sens., Vol. 36, No. 1, pp. 337-341, January 1998.

11. Sunar F. Erbek., Zkan, C. and Taberner, M., "Comparison of Maximum Likelihood Classification Method with Supervised Artificial Neural Network Algorithms for Land use Activities", lnt. Journal of Remote Sensing, Vol. 25, No. 9, pp. 1733-1748, 2004.

12. Muchoney, D. and Williamson, J., "A Gaussian Adaptive Resonance Theory Neural Network Classification Algorithm Applied to Supervised Land Cover Mapping Using Multitemporal Vegetation Index Data", IEEE Trans. Geosci. Remote Sens., Vol. 39, No. 9, pp. 1967-1977, 2001.

13. Weiguo Liu., Seto, K.C., Wu, E.Y., Gopal, S. and Woodcock, C.E., "ART-MMAP: A Neural Network Approach to Subpixel Classification", IEEE Trans. Geosci. Remote Sens., Vol. 42, No. 9, pp. 1976-1983, 2004.

14. Bogdanov, A.V., Sandven, S., Johannessen, O.M., Alexandrov, V. and Bobylev, L.P.,"Multisensor Approach to Automated Classification of Sea Ice Image Data", IEEE Trans. Geosci. Remote Sens., Vol. 43, No. 7, pp. 1648-1664, 2005.

15. Gamba, P. and Houshmand, B., "An Efficient Neural Classification Chain of SAR and Optical Urban Images", Int. J. Remote Sens., Vol. 22, No. 8, pp.1535-1553, 2001.

16. Kumar, A.S. and Majumder, K.L., "Information Fusion in Tree Classifiers", Int. J. Remote Sens., Vol. 22, No. 5, pp. 861-869, 2001.

17. Tso, B.C.K. and Mather, P.M., "Classification of Multi Source Remote Sensing Imagery Using a Genetic Algorithm and Markov Random Fields", IEEE Trans. Geosci. Remote Sens., Vol. 37, No.3, pp.1255-1260, 1999.

18. Foody, G.M., "Hard and Soft Classifications by a Neural Network with a Nonechaustively Defined Set of Classes", Int. J. Remote Sensing, Vol. 23, No. 18, pp.3853-3864, 2002.

19. Zhang, J. and Foody, G.M., "A Fuzzy Classification of Sub-urban Land Cover from Remotely Sensed Imagery", Int. J. Remote Sensing, Vol. 19, No. 14, pp. 2721-2738, 1998.

20. Franklin, S.E., Maudie, A.J. and Lavigne, M.B., "Using Spatial Co-occurrence Texture to Increase Forest Structure and Species Composition Classification Accuracy", Photogramm. Eng. Remote Sens., Vol. 67, No.7, pp.849-855, 2001.

21. Gong, P. and Howarth, P.J., "The Use of Structural Informations for Improving Land-cover Classification Accuracies at Rural-urban Fringe", Photogramm. Eng. Remote Sens., Vol. 56, No.1, pp. 67-73, 1990.

22. Bruzzone, L., Conese, C., Maselli, F. and Roli, F., "Multisource Classification of Complex Rural Areas by Stastical and Neural Network Approaches", Photogramm. Eng. Remote Sens., Vol.63, No.5, pp.523-533, 1997.

23. Bruzzone, L., Prieto, D.F. and Serpico, S.B., "A Neural-statistical Approach to Multitemporal and Multisource Remote-sensing Image Classification", IEEE Trans. Geosci. Remote Sens., Vol. 37, No.11, pp.1350-1359, 1999.

24. Keramitsoglou, I., Sarimveis, H., Kiranoudis, C.T. and Sifakis, N., "Radial Basis Function Neural Networks Classification Using Very High Spatial Resolution Satellite Imagery: An Application to the Habitat Area of Lake Kerkini (Greece)", Int. J. Remote Sensing, Vol. 26, No.9, pp.1861-1880, 2005.

25. Bruzzone, L. and Cossu, R., "A Multiple-cascade-Classifier System for a Robust and Partially Unsupervised Updating of Land-cover Maps," IEEE Trans. Geosci. Remote Sens., Vol.40, No. 9, pp. 1984-1996, Sep. 2002.

26. Sarimveis, H., Alexandridis, A., Tsekouras, G. and Bafas, G., "A Fast and Efficient Algorithm for Training Radial Basis Function Neural Networks Based on a Fuzzy Partition of the Input Space", Industrial and Engineering Chemistry Research, Vol.41, pp.751-759, 2002.

27. Salu, Y. and Tilton, J., "Classification of Multispectral Image Data by the Binary Diamond Neural Network and by Nonparametric, Pixel-by-pixel Methods", IEEE Trans. Geosci. Remote Sens., Vol.30, pp. 606-618, 1993.

28. Baraldi, A. and Parmiggiani, F., "A Neural Network for Unsupervised Categorization of Multivalued Input Patterns: An Application of Satellite Image Clustering", IEEE Trans. Geosci. Remote Sensing, Vol.33, pp. 305-316,1995.

29. Moller, M.F., "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning", Neural Networks, Vol. 6, pp. 525-533,1993.

30. Charalambous, C., "Conjugate Gradient Algorithm for Efficient Training of Artificial Neural Networks", IEEE Proceedings, Vol.139, No.3, pp.301-310, 1992.

31. Hagan, M.T. and Menhaj, M., "Training Feed Forward Networks with the Marquardt Algorithm", IEEE Transactions on Neural Networks, Vol. 5, No.6, pp. 989-993, 1994.

32. Hush, D.R. and Horne, B.G., "Prograss in Supervised Neural Networks", IEEE Signal Process. Mag., Vol. 10, No. I, pp. 8-39,1993.

33. Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.

34. Michalewicz, Z., Genetic Algorithm + Data Structures = Evolution Programs, New York: Springer-Verlag, 1994.

35. Nygard, K.E. and Kadaba, N., "Modular Neural Networks and Distributed Adaptive Search for Traveling Salesman Algorithms", Proc. of the SPIE, Vol. 1294, pp. 442-451,1990.

36. Chang, E.J. and Lippmann, R.P.,"Using Genetic Algorithms to iMprove Pattern Classification Performance", Advances in Neural Information Processing, Vol. 3, pp. 797-803, 1991.

37. Weymaere, N. and Martens, J., "On the Initialization and Optimization of Multi Player Perceptrons", IEEE Trans. Neural Networks, Vol. 5, pp. 738-751, 1994.

38. Yao, X. and Liu, Y., "A New Evolutionary System for Evolving Artificial Neural Networks", IEEE Trans. Neural Networks, Vol. 8, pp. 694-713, 1997.

39. Lam, H.K., Ling, S.H., Leung, F.H.F. and Tam, P.K.S., "Tuning of the Structure and Parameters of Neural Network Using an Improved Genetic Algorithm", Proc. 27th Annu. Conf. IEEE Ind. Electron. Soc., pp. 2530, 2001.

40. Yao, X., "Evolving Artificial Networks", Proc. IEEE, Vol. 87, pp. 1423-1447, 1999.

41. Schaffer, J.D., Whitley, D. and Eshelman, L.J., "Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art", Proc. Int. Workshop Combinations Genetic Algorithms Neural Networks, Vol. 137,1992.

42. Gropp, W., Lusk, E. and Skjellum, A., Using MPI: Portable Parallel Programming with the Message-Passing Interface, MIT Press, 1999.

43. Guang-Bin Huang, Saratchandran, P. and Sundararajan, N., "A Generalized Growing and Pruning RBF (GGAP- RBF) Neural Network for Function Approximation", IEEE Trans. on Neural Networks, Vol. 16, No. 1, pp. 57-67, 2005.

44. Suresh, S., Mani, V., Omkar, S.N. and Kim, H. J., "Genetic Algorithms for Divisible Load Scheduling", Technical Report AF/GC/001/2004, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India, 2004.

45. Yoon, H.S. and Moon, B.R., "An Empirical Study on the Synergy of Multiple Crossover Operators", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 212-223, 2002.

46. Duda, R.O. and Hart, P.E., Pattern Classification and Scene Analysis, New York: Wiley and Sons, 1973.