IETE Journal of Research
Vol 44, Nos 1 & 2, January-April 1998, pp 61-87

Soft Computing Tools and Pattern Recognition*

SANKAR K PAL, FIETE
Machine Intelligence Unit, Indian Statistical Institute, 203, B T Road, Calcutta 700 035, India.

Relevance of different soft computing tools e.g., fuzzy sets, artificial neural networks and genetic algorithms to pattern recognition problems is explained. Their distinguishing characteristics and roles in the soft computing framework are stated. Genetic algorithmic approach, being relatively new to the pattern recognition community, is paid more attention. A classification methodology based on this approach is described in detail along with its different features and a comparison in performance with the related methods. The effect of incorporating variable string length and differentiation in chromosome is discussed. Relation with Bayes decision boundary and analogy with multilayer perceptron based classification are explained. Finally, the merits of integrating the different soft computing tools for designing an efficient decision making system are stated with some application specific examples on neuro-fuzzy and neuro-genetic approaches. Scope for further research is outlined. An extensive bibliography is also provided.

1 Introduction

Soft computing is a consortium of methodologies which work synergetically and provides in one form or another flexible information processing capabilities for handling real life ambiguous situations. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning and partial truth in order to achieve *tractability, robustness and low cost solution*. In other words, it provides the foundation for the conception and design of high MIQ (Machine IQ) systems, and therefore forms the basis of future generation computing systems. At this juncture, Fuzzy Logic (FL), Artificial Neural Networks (ANN) and Genetic Algorithms (GAs) are the three principal components where FL provides algorithms for dealing with imprecision and uncertainty, ANN the machinery for learning and adaptation, and GAs for optimization and searching [1,2].

The present article deals with the relevance along with applications of soft computing in the area of pattern recognition. Pattern recognition [3,4] and machine learning form a major area of research and development that encompasses the processing of pictorial and other non-numerical information obtained from interaction between science, technology and society. A motivation for this spurt of activity in this field is the need for the people to communicate with computing machines in their natural mode of communication. Another important motivation is that scientists are also concerned with the idea of designing and making intelligent machines that can carry out certain tasks as we human beings do. The most salient outcome of these is the concept of future generation computing systems.

The ability to recognize a pattern is the first requirement for any intelligent machine. Pattern recognition is a must component of the so-called "Intelligent Control Systems" which involve processing and fusion of data from different sensors and transducers. It is also a necessary function providing "failure detection", "verification", and "diagnosis task". Machine recognition of patterns can be viewed as a two-fold task, consisting of learning the invariant and common properties of a set of samples characterizing a class, and of deciding that a new sample is a possible member of the class by noting that it has properties common to those of the set of samples. Therefore, the task of pattern recognition by a computer can be described as a transformation from the measurement space M to the feature space F and finally to the decision space D.

When the input pattern is a gray tone image, the measurement space involves some processing tasks such as enhancement, filtering, noise reduction, segmentation, contour extraction and skeleton extraction, in order to extract salient features from the image pattern. This is what is basically known as image processing [5,6]. The ultimate aim is to make its understanding, recognition and interpretation from the processed information available from the image pattern. Such a complete image recognition/interpretation system is called a vision [7,8] system which may be viewed as consisting of three levels namely, low level, mid level and high level.

* The article is dedicated to his students and colleagues who worked with him during the last 10 years; without their sincere contribution it would have been difficult to win the *20th IETE-Ram Lal Wadhwa Gold Medal*.

Delivered on 1 October '97 during the 40th IETE Annual Technical Convention at Mumbai.

In a pattern recognition or vision system, uncertainties can arise at any phase of the aforementioned tasks resulting from incomplete or imprecise input information, ambiguity or vagueness in input images, ill-defined and/or overlapping boundaries among the classes or regions, and indefiniteness in defining/extracting features and relations among them. Any decision taken at a particular level will have an impact on all higher level activities. It is therefore required for a recognition system to have sufficient provision for representing the uncertainties involved at every stage, so that the ultimate output (results) of the system can be associated with the least uncertainty (and not be affected or biased very much by the earlier or lower level decisions).

The utility of fuzzy set theory [9-16] in handling uncertainty [17-19], arising from deficiencies of information available from a situation (as mentioned above) in pattern recognition problems, has adequately been addressed in the literature [11-13, 16,17, 20-22]. This theory provides an approximate, yet effective and more flexible means of describing the behavior of systems which are too complex or too ill-defined to admit precise mathematical analysis by classical methods and tools. Since the theory of fuzzy sets is a generalization of classical set theory, it has greater flexibility to capture faithfully the various aspects of incompleteness or imperfection (i.e., deficiencies) in information of a situation. This theory is also reputed to mimic human reasoning process for decision making.

Again, for any pattern recognition, image analysis or vision system, one desires to achieve robustness of the system with respect to random noise and failure of components, and to obtain output in real time. Moreover, a system can be made artificially intelligent if it is able to emulate some aspects of human information processing system. Neural network (*NN*) [23-34] based approaches are attempts to achieve these goals. A neural network can formally be defined as : *a massively parallel interconnected network of simple (usually adaptive) processing elements which is intended to interact with the objects of the real world in the same way as biological systems do*. The architecture of the network depends on the goal one is trying to achieve. The massive connectivity among the neurons usually makes the system fault tolerant (with respect to noise and component failure) while the parallel processing capability enables the system to produce output in real time. Moreover, most of the image analysis operations are co-operative in nature and the tasks of recognition mostly need formulation of complex decision regions. Neural network models have the capability of achieving these properties. All these characteristics, therefore, suggest that image processing and pattern recognition problems can be considered as prospective candidates for neural network implementation. Many efficient methodologies have been developed, specially during the last decade, based on this realization.

Thus we see that fuzzy set theoretic models try to mimic human reasoning and uncertainty handling capabilities, whereas neural network models attempt to emulate the architecture and information representation schemes of the human brain. Integration of the merits of these two technologies therefore promises to provide, to a great extent, more intelligent systems (in terms of parallelism, fault tolerance, adaptivity and uncertainty management) to handle real life recognition problems. A large number of researchers have now concentrated on exploiting these modern concepts during the past seven to ten years to solve complex problems in various fields under a new branch called *neuro-fuzzy computing*.

One may note that the methods developed for pattern recognition and image processing are usually problem dependent. Moreover, many tasks involved in the process of analyzing/identifying a pattern need appropriate parameter selection and efficient search in complex spaces in order to obtain optimal solutions. This makes the process not only computationally intensive, but also leads to a possibility of losing the exact solution.

Genetic algorithms (GAs) [35-39], another biologically inspired technology, are randomized search and optimization techniques guided by the principles of evolution and natural genetics. They are efficient, adaptive and robust search processes, producing near optimal solutions and have a large amount of implicit parallelism. Therefore, the application of genetic algorithms for solving certain problems of pattern recognition, which need optimization of computation requirements, and robust, fast and close approximate solution, appears to be appropriate and natural [40]. Based on this realization it also appears justified to apply GAs to overcome some of the drawbacks (limitations) of fuzzy set theory (e.g., tuning of membership functions) and artificial neural networks (e.g., determining optimum network architectures) under hybrid frameworks. Research articles on the integration of genetic algorithms and pattern recognition have started to come out. Therefore, this area of soft computing is relatively much newer than those using fuzzy sets and artificial neural networks.

Our discussion in this article will give more emphasis, among others, towards the development of classification methodology with GAs. The rest of the article is organized as follows. In Section 2, the relevance of fuzzy set theoretic methods for pattern recognition/image analysis is described. The relevance of neural network based techniques in this context is described in Section 3. In Section 4 we provide, first of all, a brief description of the characteristics of GAs, and then explain a methodology for designing a classifier in Section 5. This includes incorporation of ancestor's influence in fitness function, sexual discrimination in chromosome, variable strings in encoding and comparison with other classification techniques. Various integrations of the soft computing tools such as neuro-fuzzy and neuro-genetic approaches for designing efficient hybrid systems are discussed in Section 6. Concluding remarks can be found in Section 7.

2 Relevance of fuzzy set theory in pattern recognition

Fuzzy sets were introduced in 1965 by Zadeh [9] as a new way to represent vagueness in everyday life. They are generalizations of conventional (crisp) set theory. Conventional sets contain objects that satisfy precise properties required for membership. Fuzzy sets, on the other hand, contain objects that satisfy imprecisely defined properties to varying degrees. A fuzzy set A of the universe X is defined as a collection of ordered pairs

$$A = \{(x\,\mu_A(x), x), \forall x \in X\} \tag{1}$$

where $\mu_A(x)$, $(0 \leq \mu_A(x) \leq 1)$ gives the degree of belonging of the element x to the set A or the degree of possession of an imprecise property represented by A. Since the theory of fuzzy sets is a generalization of classical set theory, it has greater flexibility to capture faithfully the various aspects of incompleteness or imperfection in information of a situation. The flexibility of fuzzy set theory is associated with the elasticity property of the concept of its membership function. The grade of membership is a measure of the compatibility of an object with the concept represented by a fuzzy set. The higher the value of membership, the lesser will be the amount (or extent) to which the concept represented by a set needs to be stretched to fit an object. Different aspects of fuzzy set theory including membership functions, basic operations and uncertainty measures can be found in [9-22].

In this section we explain some of the uncertainties which one often encounters while designing a pattern recognition system and the relevance of fuzzy set theory in handling them. Let us consider, first of all, the case of processing and recognition of a gray-tone image pattern. A gray tone image possesses ambiguity within each pixel because of the possible multi-valued levels of brightness. This pattern uncertainty is due to inherent vagueness rather than randomness. If the gray levels are scaled to lie in the range [0,1], we can regard the gray level of a pixel as its degree of belonging (membership) in the set of high-valued ('bright') pixels; thus a gray tone image can be viewed as a fuzzy set. Regions, features, primitives, properties, and relations among them that are not crisply defined can similarly be regarded as fuzzy subsets [41-46]. Basic principles and operations of image processing and recognition in the light of fuzzy set theory are available in [16,46].

Uncertainty in an image pattern may he explained in terms of grayness ambiguity or spatial (geometrical) ambiguity or both. Grayness ambiguity means 'indefiniteness' in deciding whether a pixel is white or black. Spatial ambiguity refers to 'indefiniteness' in the shape and geometry of a region within the image. For example, grayness ambiguity measures are reflected by index of fuzziness and entropy [16, 47-54], whereas spatial ambiguity measures are represented by fuzzy geometrical properties [42,45, 55-57].

Conventional approaches to image analysis and recognition [5-8] consist of segmenting the image into meaningful regions, extracting their edges and skeletons, computing various features (e.g., area, perimeter, centroid etc) and primitives (e.g., line, corner, curve etc) of and relationships among the regions, and finally, developing decision rules and grammars for describing, interpreting and/or classifying the image and its sub-regions. In a conventional system each of these operations involves crisp decisions (i.e., yes or no, black or white, 0 or 1) to make regions, features, primitives, properties, relations and interpretations crisp.

Since the regions in an image are not always crisply defined, uncertainty can arise within every phase of the aforesaid tasks. Any decision made at a particular level will have an impact on all higher level activities. An image recognition system should have sufficient provision for representing and manipulating the uncertainties involved at every processing statge; i.e., in defining image regions, features and relations among them, so that the system retains as much of the 'information content' of the data as possible. If this is done, the ultimate output (result) of the system will possess minimal uncertainty (and unlike conventional systems, it may not be biased or affected as much by lower level decision components).

For example, consider the problem of object extraction from a scene. Now, the question is 'how can one define exactly the target or object region in a scene when its boundary is ill-defined?' Any hard thresholding made for the extraction of the object will propagate the associated uncertainty to subsequent stages (e.g., thinning, skeleton extraction, primitive selection, etc) and this might, in turn, affect feature analysis and recognition. A similar case arises with the task of skeletonization and contour detection of a region. Thus, it is convenient, natural and appropriate to avoid committing ourselves to a specific (hard) decision (e.g., segmentation, edge detection and skeletonization), by allowing the segments or skeletons or contours to be fuzzy subsets of the image, the subsets being characterized by the possibility (degree) to which each pixel belongs to them. Prewitt [41] first suggested that the results of image segmentation should be fuzzy subsets, rather than ordinary subsets. Similarly, while describing relations among different components and features or classifying the sub-regions it is necessary to make the decision-making algorithms flexible by providing soft decisions. *In Short, gray information is expensive and informative. Once it is thrown away, there is no way to get it back. Therefore one should try to retain this information as long as possible throughout the decision making tasks for its full use. When it is required to make a crisp decision at the highest level one can always through away or ignore this information.*

Some of the areas of image analysis where the theory of fuzzy sets has been adequately applied are:

(*i*) computation of fuzzy geometric properties [42,45,55,57]

(*ii*) fuzzy segmentation [56, 58-68]

(*iii*) evaluation of image quality [16,48,58, 69-77]

(*iv*) image operations, like thinning and edge detection [16,43, 78-80]

(*v*) fuzzy primitives (or features) from fuzzy edges and segmented regions for shape analysis, matching, and recognition [79, 81-84].

Let us now consider the case of a decision-theoretic approach to pattern classification. With the conventional probabilistic and deterministic classifiers [3,4], the features characterizing the input patterns are considered to be quantitative (numeric) in nature. The patterns having imprecise or incomplete information are usually ignored or discarded from their designing and testing processes. The impreciseness (or ambiguity) may arise from various causes. For example, instrumental error or noise corruption in the experiment may lead to only partial or partially reliable information being available on a feature measurement F, e.g., F is about 500 (say), or F is between 400 and 500 (say). Again, in some cases the expense incurred in extracting the exact value of a feature may be high, or it may be difficult to decide on the actual salient features to be extracted. Sometimes, it may become convenient to use linguistic variables and hedges, e.g., small, medium, high, very, more or less etc in order to describe the feature information (e.g., F is very small). In such cases, it is not appropriate to give exact representation to uncertain feature data. Rather, it is reasonable to represent uncertain feature information by fuzzy subsets.

Again, uncertainty in classification or clustering of patterns may arise from the overlapping nature of the various classes. This overlapping may result from fuzziness or randomness. Moreover, the concept of clustering, in practice, is a fuzzy notion (because the technique is unsupervised and we do not have any information on the class structures or labeled samples). In the conventional technique, it is usually assumed that a pattern may belong to only one class, which is not necessarily true in real life applications. A pattern can and should be allowed to have degrees of membership in more than one class. It is, therefore, necessary to convey this information while classifying a pattern or clustering a data set.

Similalry, consider the problem of determining the boundary or shape of a class from its training set. Classical approaches attempt to estimate the exact shape for the class by determining the boundary which contains or passes through some or all of the sample points; which in a practical case may not be the right one. It may be necessary to extend the boundaries to some extent to cover the possible portions uncovered by the sample points. The extended portions should have lower possibility to be in the class than the portions explicitly highlighted by the sample points. The size of the extended regions should also decrease with an increase in the number of sample points. This leads one to determine a fuzzy shape and boundary of a pattern class.

From the aforementioned examples, we see that the concept of fuzzy sets can be used at the *feature level* in representing input data as an array of membership values denoting the degree of possession of certain properties, in representing linguistically phrased input features for their processing, in weakening the strong commitments for extracting ill-defined image regions, properties, primitives, and relations among them, and at the *classification level*, for representing class membership of objects, and for providing an estimate (or a representation) of missing information in terms of membership values. *In other words, fuzzy set theory provides a notion of embedding: We find a better solution to a crisp problem by looking in a large space at first, which has different (usually less) constraints and therefore allows the algorithm more freedom to avoid errors forced by commission to hard answers in intermediate stages.*

The capability of fuzzy set theory in pattern recognition problems has been reported adequately since late sixties. For example, feature extraction is dealt with in [85-88]. Classification of patterns including linguistic representation of inputs [89-92] can be found in [13,16, 93-97]. Clustering techniques and their validations are discussed in [11, 98-104]. Some applications to real life problems are reported in [16,60,82,84,105-111]. A review showing the development of this area has been provided in [17].

3 Relevance of neural network approaches

Neural network (NN) models [23-24] try to emulate the biological neural network/nervous system with electronic circuitry. NN models have been studied for many years with the hope of achieving human-like performance (artificially), particularly in the field of pattern recognition, by capturing the key ingredients responsible for the remarkable capabilities of the human nervous system. Note that these models are extreme simplifications of the actual human nervous system.

ANNs are designated by the network topology, connection strength between pairs of neurons (called weights), node characteristics and the status updating rules. Node characteristics mainly specify the primitive types of operations it can perform, like summing the weighted inputs coming to it and then amplifying it or doing some fuzzy aggregation operations. The updating rules may be for weights and/or states of the processing elements (neurons). Normally an objective function is defined which represents the complete status of the network and the set of minima of it corresponds to the set of stable states of the network. Since there are interactions among the neurons the collective computational property inherently reduces the computational task and makes the system fault tolerant. Thus NN models are also suitable for tasks where collective decision making is required. Hardware implementations of neural networks are attempted in [112-115].

Artificial neural networks have become a technical folk legend of late. The market is flooded with new, more technical software (hardware) and products; many more are sure to come. Some of the popular networks are Hopfield Net (HN), Multilayer Perceptron (MLP), Self-Organizing Feature Map (SOFM), Learning Vector Quantization (LVQ), Radial Basis function Network, Cellular Neural Network (CNN) and Adaptive Resonance Theory (ART) network.

Neural network based systems are usually reputed to enjoy the following major characteristics:

- adaptivity: adjusting the connection strengths to new data/information,
- speed: due to massively parallel architecture,
- robustness: to missing, confusing, ill-defined/noisy data,
- ruggedness: to failure of components,
- optimality: as regards error rates in performance.

For any pattern recognition system, one desires to achieve the above mentioned characteristics. More over, there exists some direct analogy between the working principles of many pattern recognition tasks and neural network models. For example, image processing and analysis in the spatial domain mainly employ simple arithmetic operations at each pixel site in parallel. These operations usually involve information of neighboring pixels (co-operative processing) in order to reduce the local ambiguity and to attain global consistency. An objective measure is required (representing the overall status of the system), the optimum of which represents the desired goal. The system thus involves collective decision. On the other hand, we notice that neural network models are also based on parallel and distributed working principles (all neurons work in parallel and independently). The operations performed at each processor site are also simpler and independent of the others. The overall status of a neural network can also be measured.

Let us consider, in particular, the case of pixel classification. A pixel is normally classified into different classes depending on its gray value, positional information and contextual information (collected from the neighbors). Pixels at different sites can be classified independently. The mathematical operations needed for this task are also simple. A neural network architecture in which a single neuron is assigned to a pixel and is connected to its neighbors can therefore be applied for this task. The neurons operate in parallel and are independent of each other. The local interconnections provide the contextual information (which can be adaptive or dynamic also) for classification.

Again, the task of recognition in a real-life problem involves searching a complex decision space. This becomes more complicated particularly when there is no prior information on class distribution. Neural network based systems use adaptive learning procedures, learn from examples and attempt to find a useful relation between input and output, however complex it may be, for decision-making problems. Neural netowrks are also reputed to model complex non-linear boundaries and to discover important underlying regularities in the task domain. These characteristics demand that methods are needed for constructing and refining neural network models for various recognition tasks. For example, consider the case of supervised classification. Here each pattern is characterized by a number of features. Different features usually have different amounts of weight in characterizing the classes. A collective decision, taking into account all the features, is made for assignment of class labels to an input. A multi-layer perceptron in which the input layer has neurons equal to the number of features and the output layer has neurons equal to the number of classes, can therefore be used to tackle this classification problem. Here the importance of different features will automatically be encoded in the connecting links during training. The non-linear decision boundaries are modeled and class labels are assigned by taking collective decisions. *In short, neural networks are natural classifiers having resistance to noise, tolerance to distorted images / patterns (ability to generalize), superior ability to recognize partially occluded or degraded images/overlapping pattern classes or classes with highly nonlinear boundaries, and potential for parallel processing.*

Major areas in which neural networks have been applied in order to exploit the computational power, and to make robust decisions are:

i: feature selection and pattern classification [88,116-128],

ii: image preprocessing and scene analysis [129-141],

iii: text processing [117,142],

iv: expert system design/rule generation [143-149],

v: controller design [150-152],

vi: natural language processing [153,154],

vii: approximate reasoning and speech recognition [116,155-157].

As mentioned before, GA is relatively a newer technology as compared to fuzzy logic and artificial neural networks. Therefore, its application to pattern recongition problem will be discussed in detail. Before that we describe in the next section the basic principles and features of GA for the convenience of readers.

4 Genetic algorithms: basic principles and features

Genetic algorithms (GAs) [35-39] are adaptive computational procedures modeled on the mechanics of natural genetic systems. They express their ability by efficiently exploiting the historical information to speculate on new offspring with expected improved performance [36]. GAs are executed iteratively on a set of coded solutions, called population, with three basic operators: *selection/reproduction, crossover* and *mutation*. They use only the payoff (objective function) information and probabilistic transition rules for moving to the next iteration. They are different from most of the normal optimization and search procedures in four ways:

- GAs work with the coding of the parameter set, not with the parameter themselves.
- GAs work simultaneously with multiple points, and not a single point.
- GAs search via sampling (a blind search) using only the payoff information.
- GAs search using stochastic operators, not deterministic rules.

Since a GA works simultaneously on a set of coded solutions it has very little chance to get stuck at local optima when used as optimization techniques. Again, it does not need any sort of auxiliary information, like derivative of the optimizing function. Moreover, the resolution of the possible search space is increased by operating on coded (possible) solutions and not on the solutions themselves. Further this search space need not be continuous. Recently, GAs are finding widespread applications in solving problems, requiring efficient and effective search, in business, scientific and engineering circles like synthesis of neural network architectures [158-164], travelling salesman problem [165], graph coloring, scheduling [38], numerical optimization [166], and pattern recognition and image processing [40,167-169].

GAs are intended to mimic some of the processes observed in natural evolution. The evolution starts from a set of individuals (assumed solution set for the function to be optimized) and proceeds from generation to generation through genetic operations. Replacement of an old population with a new one is known as generation when generational replacement technique (replace all the members of old population with the new ones) is used. Another reproduction technique, called steady state reproduction, replaces one or more individuals at a time instead of the whole population [38]. GAs require only a suitable objective function which acts as the environment in order to evaluate the suitability of the derived solutions (chromosomes). A schematic diagram of the basic structure of a genetic algorithm is shown in Fig 1.

A GA typically consists of the following components:

- A population of binary strings or coded possible solutions (biologically referred to as chromosomes)
- A mechanism to encode a possible solution (mostly as a binary string)
- Objective function and associated fitness evaluation techniqes
- Selection/reproduction procedure
- Genetic operators (crossover and mutation)
- Probabilities to perform genetic operations

Let us briefly describe these components.

Population: To solve an optimization problem, GAs start with the chromosomal (structural) representation of a parameter set $\{x_1, x_2, ..., x_p\}$. The parameter set is to be

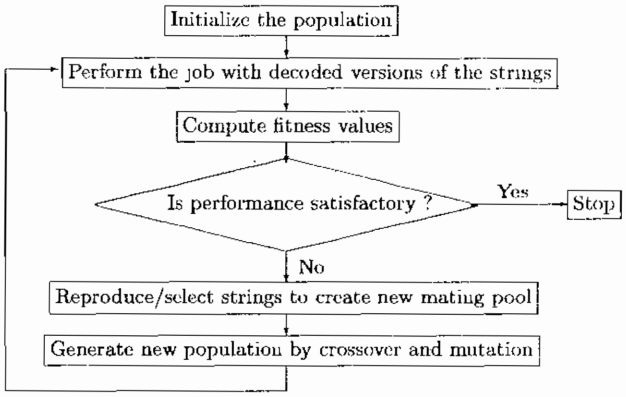

Fig 1 Basic steps of a genetic algorithm

coded as a finite length string over an alphabet of finite length. Usually, the chromosomes are strings of 0's and 1's. For example, let $\{a_1, a_2, ..., a_p\}$ be a realization of the parameter set and the binary representation of $a_1, a_2, ..., a_p$ be 10110, 00100, ..., 11001, respectively. Then the string 10110 00100 .... 11001 is a chromosomal representation of the parameter set $\{x_1, x_2, ..., a = x_p\}$. It is evident that the number of different chromosomes (strings) is 2^l, where l is the string length. Each chromosome actually refers to a coded possible solution. A set of such chromosomes in a generation is called a population. The size of a population may vary from one generation to another or it can be constant. Usually, the initial population is chosen randomly.

Encoding/decoding mechanism: It is the mechanism to convert the parameter values of a possible solution into binary strings resulting into chromosomal representation. If the solution of a problem depends on p parameters and if we want to encode each parameter with a binary string of length q, then the length of each chromosome will be $p \times q$. Decoding is just the reverse of encoding.

Objective function and associated fitness evaluation techniques: The fitness/objective function is chosen depending on the problem. It is chosen in a way such that highly fitted strings (possible solutions) have high fitness values. It is the only index to select a chromosome to reproduce for the next generation.

Selection/reproduction procedure: The selection/ reproduction process copies individual strings (called parent chromosomes) into a tentative new population (known as mating pool) for genetic operations. Number of copies reproduced for the next generation by an individual is expected to be directly proportional to its fitness value; thereby mimicking the natural selection procedure to some extent. Roulette wheel parent selection [36] and linear selection [38] are the most frequently used selection procedures.

Genetic operators are applied on parent chromosomes and new chromosomes (called offspring) are generated. Frequently used genetic operators are described below.

Crossover: The main purpose of crossover is to exchange information between randomly selected parent chromosomes by recombining parts of their corresponding strings. Actually, it recombines genetic material of two parent chromosomes to produce offspring for the next generation. The crossover may proceed in two steps. First, members of the reproduced strings in the mating pool are mated at random. Second, each pair of strings undergoes crossing over as follows: an integer position k is selected uniformly at random between 1 and $l - 1$, where l is the string length greater than 1. Two new strings are created by swapping all characters from position $(k + 1)$ to l. Let

a = 11000 10101 01000 ... 01111 10001

b = 10001 01110 11101 ... 00110 10100

be two strings (parents) selected for the crossing over operation and the generated random number be 11 (eleven). Then the newly produced offspring (swapping all characters after position 11) will be

a' = 11000 10101 01101 ... 00110 10100

b' = 10001 01110 11000 ... 01111 10001.

Some common crossover techniques are: one point crossover, multiple point crossover, shuffle-exchange crossover, uniform crossover [38] etc.

Mutation: The main aim of mutation is to introduce genetic diversity into the population. Sometimes, it helps to regain the information lost in earlier generations. In case of binary representation it negates the bit value and is known as bit mutation. Like natural genetic systems, mutation in GAs are also made occasionally. A random bit position of a random string is selected and is replaced by another character from the alphabet. For example, let the third bit of string a, given above, be selected for mutation. Then the transformed string after mutation will be

11100 10101 01000 ... 01111 10001.

Mutation is not always worth performing. High mutation rate can lead the genetic search to a random one. It may change the value of an important bit, and thereby affect the fast convergence to a good solution. Moreover, it may slow down the process of convergence at the final stage of GAs.

Probabilities to perform genetic operations: The probability to perform crossover operation is chosen in a way so that recombination of potential strings (highly fitted chromosomes) increases without any disruption. Generally, the crossover probability lies in between 0.6 and 0.9 [36,38].

Since mutation occurs occasionally, it is clear that the probability of performing mutation operation will be very low. Typically the value lies between 0.001 and 0.01 [36,38].

Elitism: In standard GA (SGA), we do not preserve the best possible solution obtained so far; thereby increasing the chance of losing the obtainable best possible solution. Elitist strategy overcomes this problem by copying the best member of each generation into the next one. Though this strategy may increase the speed of dominance of a population by a potential string (string with high fitness value), it enhances the performance of a GA using generational replacement. Concepts of distributed GAs and parallel GAs are also introduced recently [158,162].

Some improved genetic operations and developments: Several investigations into the theoretical aspects of genetic algorithms are available in the literature. The earliest one is Holland's Schema theorem [36]. Others dealt with schema analysis and deceptive problems. The issue of convergence of GAs to the globally optimal solution has been pursued in [170].

In [171], Murthy *et al* provided a stopping criterion, called ε-optimal stopping criterion, for the elitist model of the GAs. Subsequently, they have derived the ε-optimal stopping time for GAs with elitism under a 'practically valid assumption'. Note that, any string can be generated from any given string by mutation operation.

Let us now mention here some of the genetic operations recently developed for enhancing the performance of GAs. To sustain diversity (which may be lost due to crossover and very low mutation rate) into the population, Whitley *et al* [158] proposed a technique called adaptive mutation; where instead of fixed mutation rate the probability to perform mutation operation is made to increase with increase of genetic homogeneity in the population.

In another investigation to improve the performance of GAs, concepts of adaptive probabilities of crossover and mutation based on various fitness values of the population are recommended in [172]. Here, high solutions are protected and the subaverage solutions are completely disrupted; thereby preventing to get stuck at local optima.

Three new methods of selection of mating pairs for GAs are introduced recently [173] by De, Pal and Ghosh where the partners are chosen based on either their genotypic similarity (called genotypic assortative mating) or their phenotypic similarity (called phenotypic assortative mating). These methods not only help in exploiting the current search space properly before exploring the new one, but also enable one to mimic inbreeding of natural genetics. The superiority of this method over the conventional GA and the incesr prevention [174] algorithm is established on some problems of optimizing complex functions and selecting optimal neural network parameters.

Bhandari *et al* [175] have proposed a new mutation operator known as *directed mutation* which follows from the concept of induced mutation in biological systems. This operation does not involve probabilistic decision rules but the information acquired in the previous generations. Again, it does not alter the probabilistic nature of the search technique. In certain environment this operation will deterministically introduce a new point in the population. The new point is directed (guided) by the solutions obtained earlier and therefore it is called *directed mutation.* Directed mutation operation thus exploits the merits of both gradient search and genetic search.

GAs consider only the fitness value of the chromosome under consideration for measuring its suitability for selection for the next generation i.e., the fitness of a chromosome is a function of the functional value of the objective function. Fitness of a chromosome $x = g(f(x))$, where $f(x)$ is the objective function and g is another function which by operating on $f(x)$ gives the fitness value. Hence, a GA does not discriminate between two identical offspring, one produced from better (highly fit) parents and the other from comparatively weaker (low fit) parents. In nature, normally an offspring is more fit (suitable) if its ancestors (parents) are better i.e., an offspring possess some extra facility to exist in its environent if it belongs to a better family (ancestors are higbly fitted). In other words, the fitness of an individual depends also on the fitness of its ancestors in addition to its own fitness. It therefore appears to give more weightage to highly fitted chromosome (due to better ancestors) in order to generate offspring for the next generation. Based on this concept, De, Ghosh and Pal [176] defined fitness of a chromosome $x = g(f(x), a_1, a_2, ..., a_n)$ where a_is are the fitness values of its ancestors. The function g may be of various types. The weightage given to the fitness of the ancestors can be assigned heuristically at the beginning and kept fixed througbout the procedure or it may be varying/adaptive. Heuristics are given for choosing the weighting factors manually; and a procedure for evolving them automatically has also been developed. The merit of the proposed algorithm as compared to that of the GA was explained using the Schema theorem.

5 Pattern classification with genetic algorithms

Relevance of GAs to pattern recognition and image processing problems is described in Section 1. In this section we provide some results [177-182] of investigation demonstrating an application of GAs for pattern classification (supervised) in N dimensional space. Here classification is viewed as a problem of generating decision boundaries that can successfully distinguish the various classes in the feature space. In real life problems, the boundaries between the different classes are usually non-linear. The characteristics of GAs have been exploited in searching for a number of hyperplanes which can approximate the non-linear boundaries in order to provide minimum misclassification.

The feature space is generally unbounded and continous in nature. However, if bounding information can be derived from the training patterns and the space is discretized to sufficiently small intervals in each dimension, then the classification problem can be handled within the framework of Genetic Algorithms. A distinguishing feature of this approach is that the boundaries (approximated by piecewise linear segments) are generated explicitly for making decisions. Note that in the conventional methods or in multilayered perceptron based approaches to pattern classification, the generation of boundaries is a consequence of the respective decision making processes.

5.1 The methodology

In the realm of pattern classification in N dimensions, we consider a fixed number (H) of hyperplanes to constitute the decision boundary. Each chromosome encodes the parameters of these H hyperplanes. Note that since each hyperplane provides two regions, H hyperplanes provide a maximum of 2^H regions. Hence for a k class problem, $H \geq \log_2 k$. The search space for the hyperplanes (which may be considered as candidates for the formation of the decision boundary) is restricted to the hyper rectangle formed around the training pattern points.

A. String representation [177]

From elementary geometry, the equation of a hyperplane in N dimensional space ($X_1 - X_2 - \ldots - X_N$) is given by

$$x_N \cos \alpha_{N-1} + \beta_{N-1} \sin \alpha_{N-1} = d \tag{2}$$

where $\beta_{N-1} = x_{N-1} \cos \alpha_{N-2} + \beta_{N-2} \sin \alpha_{N-2}$

$$\beta_{N-2} = x_{N-2} \cos \alpha_{N-3} + \beta_{N-3} \sin \alpha_{N-3}$$

$$\vdots$$

$$\beta_1 = x_1 \cos \alpha_0 + \beta_0 \sin \alpha_0$$

The various parameters are as follows:

X_i : the ith feature of the sample points.

$(x_1, x_2, \ldots, x_N)$: a point on the hyperplane

α_{N-1} : the angle that the unit normal to the hyperplane makes with the X_n axis.

α_{N-2} : the angle that the projection of the normal in the $X_1 - X_2 - \ldots - X_{N-1}$ space makes with the X_{N-1} axis.

$\vdots$

α_1 : the angle that the projection of the normal in the $(X_1 - X_2)$ plane makes with the X_2 axis.

α_0 : the angle that the projection of the normal in the (X_1) plane makes with the X_1 axis = 0.

d : the perpendicular distance of the hyperplane from the origin.

Thus the N tuple $< \alpha_1, \alpha_2 \ldots, \alpha_{N-1}, d >$ specifies a hyperplane in N dimensional space. Each angle α_j, $j = 1, 2, \ldots, N-1$ is allowed to vary in the range of 0 to 2π. If b_1 bits are used to represent an angle, then the possible values of α_j are

$$0, \delta * 2\pi, 2\delta * 2\pi, 3\delta * 2\pi, \ldots, (2^{b_1} - 1)\, \delta * 2\pi$$

where $\delta = \frac{1}{2^{b_1}}$. Consequently, if the b_1 bits contain a binary string having the decimal value v_1, then the angle is given by $v_1 * \delta * 2\pi$.

Once the angles are fixed, the orientation of the hyperplane becomes fixed. Now only d must be specified in order to specify the hyperplane. For this purpose the hyper rectangle enclosing the sample points is considered. Let x_i^{min}, x_i^{max} be the minimum and maximum values of feature X_i as obtained from the training sample points. Then the vertices of the enclosing hyper rectangle are given by

$$(x_1^{ch_1}, x_2^{ch_2}, \ldots, x_N^{ch_N})$$

where each ch_i , $i = 1,2, \ldots, N$ can be either *max* or *min*. (Note that there will be 2^N vertex points). Let *diag* be the length of the diagnonal of this hyper rectangle given by

$$diag = \sqrt{(x_1^{max} - x_1^{min})^2 + (x_1^{max} - x_1^{min})^2 + \ldots + (x_N^{max} - x_N^{min})^2}. \tag{3}$$

A hyperplane having the given orientation and passing through one of the vertices of the hyper rectangle such that its perpendicular distance from the origin is minimum (among the hyperplanes passing through other vertices) is designated as the *base hyperplane*. Let the distance be d_{min}. If b_2 bits are used to represent d, then a value of v_2 in these bits represent a hyperplane with the given orientation and for which d is given by $d_{min} + \frac{diag}{2^{b_2}} * v_2$.

Thus each chromosome is of length of $L = H((N-1) * b_1 + b_2)$. These are initially generated randomly for a population of size Pop. A fitness function described in the next subsection is associated with each string.

B. Fitness computation

A chromosome encodes the parameters of H hyperplanes as described earlier. Using these parameters, the region in which each training pattern point lies is determined from equation 2. A region is said to provide the demarcation for class i, if maximum number of points that lie in this region belong to class i. Other points that lie in this region are considered to be misclassified. The misclassifications associated with all the regions (for these H hyperplanes) are summed up to provide the total misclassification, *miss*, for the string. Its fitness is defined as (n − *miss*), n being the total number of training samples. For this problem, the fitness function is identical to the objective function.

C. Genetic operators

Roulette wheel selection [36] is adopted to implement the *proportional selection strategy*. Elitism is incorporated by replacing the worst string of the current generation with the best string seen upto the last generation. *Single point* crossover is applied with a fixed crossover probability value. The mutation operation is performed on a bit by bit basis for a mutation probability value which is initially high, then decreased gradually to a prespecified minimum value and then increased again in the later stages of the algorithm. This ensures that in the initial stage, when the algorithm has very little knowledge about the search domain, it performs a random search through the feature space. This randomness is gradually decreased with the passing of generations so that now the algorithm performs a detailed search in the vicinity of promising solutions obtained so far. Inspite of this, the algorithm may still get stuck at a local optima. This problem is overcome by increasing the mutation probability to a high value, thereby making the search more random once again. The algorithm is terminated if the population contains at least one string with zero misclassification of points and there is no significant improvement in the average fitness of the population over subsequent generations. Otherwise, the algorithm is executed for a fixed number of generations.

5.2 Classification performance

Here we provide the implementation aspect of the classifier, its relation with Bayes decision boundary, improvement of the performance through variable string length and incorporating chromosome differentiation, and its analogy with multilayer perceptron. Finally some other attempts in this line are mentioned.

A. Implementation and Results

The effectiveness of the methodology has been demonstrated [177-182] on non linearly separable artificial data sets, Iris data and two other real life data such as speech sound and remote sensing image. Their performance is compared with Bayes classifier [3,4] (assuming normal distributions and unequal *a priori* probabilities), k-NN classifier [3,4] and multilayer perceptron (MLP) [23-24]. Here we provide some of those results for an artificial data ADS1 (Figure 2) and the speech (vowel) data (Figure 3). The artificial data set consists of 557 samples and has two classes, 1 and 2. X and Y coordinates represent the two features in an Euclidian space. The vowel data set corresponds to 871 Indian Telugu vowel sounds [16]. These were uttered in a consonant-vowel-consonant context by three male speakers in the age group of 30-35 years. The data set has three features corresponding to the first, second and third vowel formant frequencies (F_1, F_2, F_3), and six classes $\{\delta, a, i, u, e, o\}$.

A fixed population size of 20 was chosen. The crossover probability μ_c was fixed at some value. A

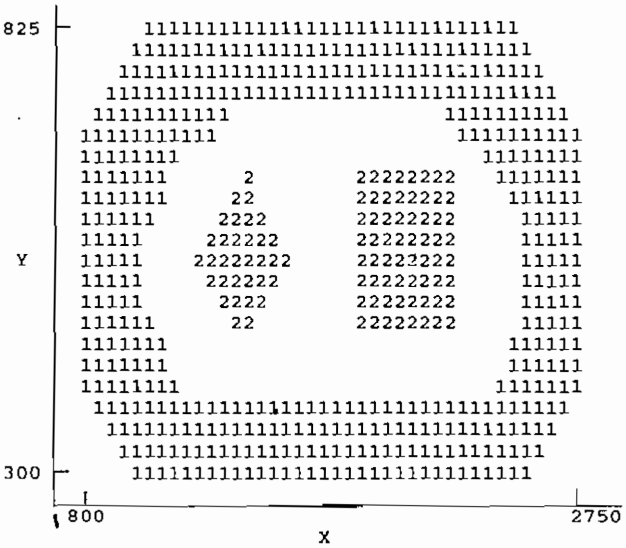

Fig 2 Data ADS1

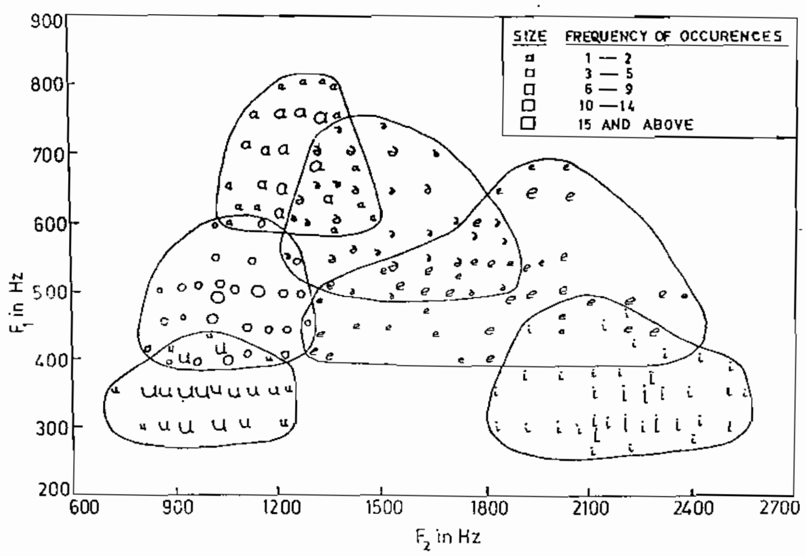

Fig 3 Vowel diagram in F_1, F_2 plane

variable value of *mut-prob* was selected from the range [0.015, 0.333]. Initially it had a high value, gradually decreasing at first, and then increasing again in the later stages of the algorithm. 100 iterations were performed with each value of the mutation probability. The process was executed for a maximum of 1500 iterations in case it did not attain a solution with zero misclassification. The experimental results are described when the size of the training set is considered to be 10%, i.e., *perc* = 10.

Table 1 shows the comparative classwise recongition scores and the overall recognition score for the artificial data for crossover probability $\mu_c = 0.8$. Different values of H (viz. 7, 6 and 4) are considered for the GA-hased classifier. Since this data set has a very small class totally surrounded by a larger class, the classwise recognition score in this case is of greater importance than the overall score. Figure 4 shows the decision boundaries obtained from 10% training data (for $H = 6$) and their ability in classifying the remaining 90% test data. The training pattern points are underscored in the figure. This boundary was obtained on termination of training after 538 iterations when a string with no misclassified points had been found.

A comparison of the performance of the proposed alogrithm (for *perc* = 10) with the Bayes classifier, k-NN

TABLE 1 Classification performance (%) of GA-classifier for ADS1, $\mu_c = 0.8$

Class	GA			Bayes	k-NN	MLP
	$H = 7$	$H = 6$	$H = 4$			
1	93.71	94.44	93.96	100.00	96.85	100
2	77.27	87.50	57.95	18.18	59.09	0
Overall	90.83	93.22	87.65	85.65	90.23	82.47

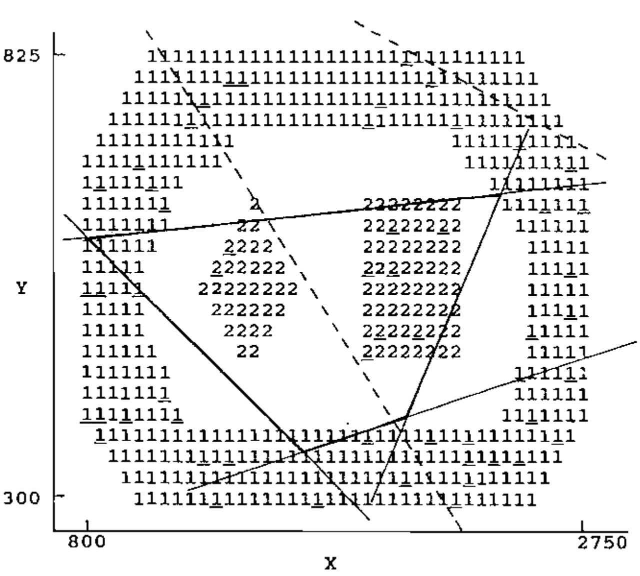

Fig 4 ADS1 data set with class boundaries generated by *GA classifier*

classifier (for $k = \sqrt{n}$) and MLP is also shown in Table 1. Both k-NN and MLP are capable of generating piecewise linear boundaries. The results show that the performance of the GA based algorithm for $H = 6$ is the best. As expected the Bayes classifier performs poorly.

Table 2 shows the comparative results on the vowel data. In each case, class δ is classified very poorly as this is the class with maximum overlap. This was also found in [16,183] where a fuzzy set theoretic classifier and Bayes classifier were used for vowel classification problem. The results show that the Bayes classifier performs best for this data set. This conform to an earlier finfings [184]. The results of the proposed algorithm are seen to be comparable to those of the Bayes classifier.

An observed feature of the methodology is that increasing the number of lines (for approximating the decision boundaries) does not necessarily result in an increase of the classification performance. The reason behind this is that increasing the number of lines means tuning more and more to the peculiarities in the training set, which may not necessarily be beneficial to the test set. A point to be noted here is that although two lines are redundant in Figure 4, for $H = 6$, yet assuming a lower value produces an inferior result as seen from Table 1. A reason behind this may be the insufficient knowledge about the termination of the algorithm. Note that the problem of ascertaining H is analogous to that of fixing the number of nodes in MLP. An overestimation of this number may lead to poor generalization capability of the classifier. A comparison between GAs and simulated annealing with respect to the design of such a classifier is provided in [180].

B. Relation with Bayes classifier

It is proved [181] that for $n \rightarrow \infty$ (i.e., for sufficiently large training data set) and for a sufficiently large number of iterations, the probability of error provided by the *GA-classifier* during training is less than or equal to that of the Bayes classifier. However, one may note that the Bayes classifier with known class distributions and *a priori*

TABLE 2 Classification performance (%) of GA-classifier for Vowel (F_1, F_2 features), $\mu_c = 0.8$

Class	GA			Bayes	k-NN	MLP
	$H = 7$	$H = 6$	$H = 4$			
δ	20.00	61.53	12.30	46.15	35.38	23.07
a	93.82	76.54	59.25	85.18	81.48	66.67
i	82.58	72.25	85.16	81.93	85.80	84.50
u	94.11	86.02	92.64	89.70	76.47	91.17
e	74.86	77.54	82.35	83.42	75.40	69.51
o	67.90	72.22	78.39	72.83	77.16	14.80
Overall	75.69	75.44	75.69	79.13	75.31	60.81

probabilities always provides the best generalization capability. Extensive experimental results on overlapping data sets following triangular and normal distributions with both linear and nonlinear class boundaries conform to these claims [181]. The claims also hold good when circular surfaces are considered as constituting elements/ segments of boundaries. It is observed that as n increases, the probabilities of error of the *GA-classifier* during both training and testing approach those of the Bayes classifier. Figure 5 shows that as n increases, the boundary provided by the GA classifier approaches the Bayes boundary. Here two class problem with triangular distribution of the data is considered [181].

It is shown that the optimum number of hyperplanes generated by the GA based classifier is equal to that required to model the Bayes decision boundary when there exists only one partition of the feature space that provides the Bayes error probability. The variation of recognition score with *a priori* class probability is also shown to be similar for both the classifiers.

C. Using variable string length

One can see that an a *priori* knowledge of a proper value of H is difficult to estimate for the *GA-classifier*. Thus, it is usually overestimated, leading to the problem of over fitting of the data along with an associated reduction in the generalization capability of the *GA-classifier*. Additionally, it often results in the presence of some redundant surfaces in the final decision boundary.

In order to overcome these limitations, the concept of variable string lengths in GA (termed as VGA) has been used for the problem of constructing class boundaries through the placement of a variable number of hyperplanes so as to classify patterns in N dimensional feature space. The parameters of a number of hyperplanes, whose value may now vary, are encoded in a chromosome or string. Specially designed genetic operators are applied on these strings over a number of generations till a termination criterion is attained. The fitness function incorporates a weighting coefficient on the number of hyperplanes which ensures, primarily, the minimization of the number of misclassified samples, as also the reduction of the number of hyperplanes. The *GA-classifier* utilizing variable string lenghts is called the *VGA-classifier* [182].

As mentioned in Section 5.2.B, it has been theoretically established here that as the number of iterations and the size of the training data go to infinity, the performance of the classifier approaches that of the Bayes classifier. At the same time, the number of hyperplanes provided by the classifier will he the optimum.

Investigations into the effectiveness of the aforesaid *VGA-classifier*, for two sets of artificial data, Iris data and speech data, demonstrate that the concept of VGA, besides being able to evolve the number of hyperplanes from a given range, helps in improving its recognition capability. Table 3 provides a few such results. It is seen that the VGA-classifier is able to determine H automatically (e.g., 3 for ADS1 and 6 for vowel) starting from a maximum

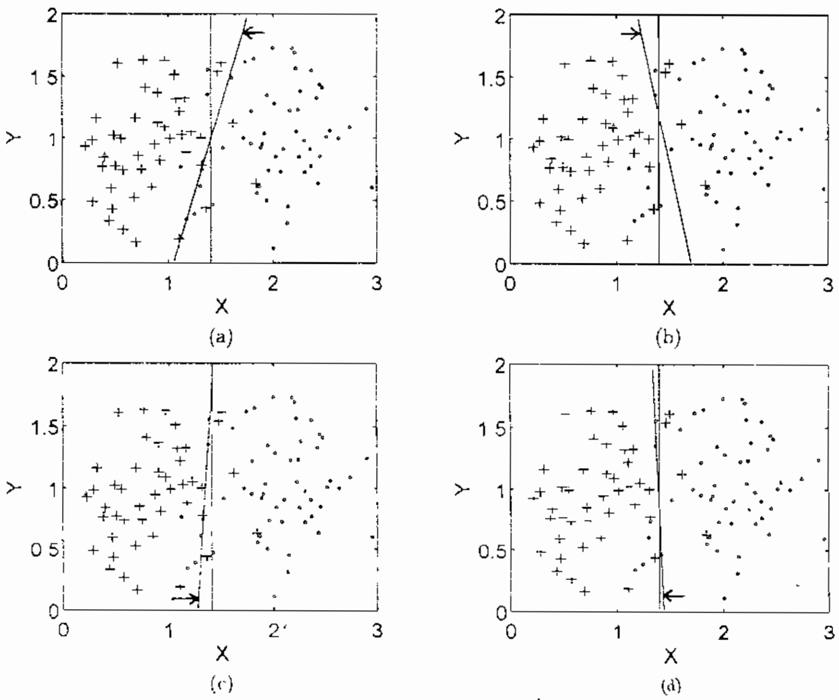

Fig 5 Triangular distribution of points with Bayes-line and GA-line (arrow) for 1500 iterations (*a*) n = 100, (*b*) n = 1000, (*c*) n = 2000, (*d*) n = 4000

number of hyperplanes $H_{max} = 10$. The recognition scores are also improved.

D. Incorporating chromosome discrimination

The effect of incorporating some common forms of differentiation as found in natural systems (e.g., sexual differentiation), in GAs is investigated in [178]. A genetic algorithmic methodology, termed GACD, is described, which incorporates chromosome differentiation for evolutionary process.

In GACD, chromosomes are distinguished into two categories of population, termed as M and F populations, based on the value contained in the two class bits. Initially, the two populations are of the same size, and are generated based on maximum Hamming distance between them. Crossover (mating) is allowed only between individuals belonging to these categories. As a results of crossover, two offspring are created, whose classes are determined stochastically, depending on the parent chromosomes. The offspring are accordingly put into the different populations. Thus the number of chromosomes in the individual populations may vary, although the total number is kept constant.

Schema analysis of GACD shows that the basic tenet of genetic algorithms holds for GACD as well; above average, short, low order schema will recieve increasing number of trials (or instances) in subsequent generations. It is also shown that in certain situations, the lower bound of the number of instances of a schema sampled by GACD is greater than or equal to that of the conventional genetic algorithms.

Implementation of the *GA-classifier* by using GACD instead of GA provides the *GACD-classfier*. Its subsequent application to the aforesaid two artificial, Iris and speech data sets is found to generate better recognition scores. The number of iterations required to attain the termination criterion is also found to be less than that required by conventional GA. It is to be noted that the concept of GACD enriches the GA literature in general, and that of the classifier in particular.

Tables 4 and 5 show some results for two different values of crossover probabilities μ_c with $H = 6$. It is clear that the GACD classifier is superior to GA classifier both in terms of recognition score and rate of convergence.

E. Analogy between VGA classifier and MLP based classifier

An analogy between the principles of *VGA-classifier*, mentioned in Section 5.2.C, and multilayer perceptron (with hard limiters) based classifier is described in [185]. Both of them basically model the class boundaries using a

TABLE 3 Classification performance (%) of *VGA-classifier* for $H_{max} = 10$ and *GA-classifier* for $H = 6$, $\mu_c = 0.8$. Vowels have features F_1, F_2, F_3.

Data set	*VGA-classifier*			Test score for
	Training score (%)	Test score (%)	H_{VGA}	*GA-classifier*, $H = 6$
ADS1	100.00	95.62	3	93.22
Vowel	80.00	73.66	6	71.99

TABLE 4 Classification performance (%) of *GACD-classifier* and *GA-classifier* for $\mu_c = 0.8$, H = 6. Vowels have features F_1, F_2, F_3.

Data set	Avg no. of generations		Training (avg recog score)		Testing (avg recog score)	
	GACD	GA	GACD	GA	GACD	GA
ADS1	415.2	512.6	100.00	100.00	94.25	93.22
Vowel	1500	1500	94.12	82.35	75.19	71.99

TABLE 5 Classification performance (%) of *GACD-classifier* and *GA-classifier* for $\mu_c = 0.5$, H = 6. Vowels have features F_1, F_2, F_3.

Data set	Avg no. of generations		Training (avg miss)		Testing (avg recog score)	
	GACD	GA	GACD	GA	GACD	GA
ADS1	520.2	905.5	100.00	98.71	95.02	94.22
Vowel	1500	1500	96.47	88.23	76.71	71.37

number of hyperplanes. Additionally, it is established that the number of hyperplanes provided by the *VGA-classifier* for constituting the decision boundary will be the optimum. Based on these, a method called the *network construction algorithm* (NCA) for determining the MLP architecture automatically is described. It is shown that the architecture would need atmost two hidden layers, the neurons of which are responsible for generating hyperplanes and regions.

The architecture derived using NCA has the following specifications. The input layer has N neurons, where N is the dimensionality of the feature space. These simply transmit the input signals to their outputs. The first hidden layer has H_o neurons, where H_o is the number of hyperplanes provided by the *VGA-classifier*. These are responsible for generating the equations of the H_o hyperplanes. The neurons in the second hidden and output layers perform the AND & OR functions respectively. The NCA also includes a post processing step which automatically removes any redundant neuron in the hidden/output layer [185].

Superiority of the MLP thus derived using NCA, over its more conventional counterparts, is extensively established over different data sets [185]. Tables 6 and 7 demonstrate some of the results. As seen, the architecture for the MLP is found to be 2:3:5:2 for ADS1 and 3:6:7:6 for vowel. MLP with this architecture trained with back

TABLE 6 MLP structure and classification performance (%) for ADS 1

Class	*VGA-classifier* $H_o = 3, r = 5$		MLP SIGMOID Arch = 2:3:5:2		MLP Hard Limiter Arch=2:3:5:2	
	Training	Testing	Training	Testing	Training	Testing
1	100.00	95.89	100.00	100.00	100.00	95.89
2	100.00	94.31	0.0	0.0	100.00	94.31
Overall	100.00	95.62	83.63	82.47	100.00	95.62

TABLE 7 MLP structure and classification performance (%) for Vowel (F_1, F_2, F_3 features)

Class	*VGA-classifier* $H_o = 3, r = 5$		MLP SIGMOID Arch = 3:6:7:6		MLP Hard Limiter Arch=3:6:7:6	
	Training	Testing	Training	Testing	Training	Testing
δ	10.30	8.21	85.71	6.15	10.30	8.21
a	100.00	91.35	87.50	13.58	100.00	91.35
i	94.11	84.51	100.00	78.70	94.11	84.51
u	73.33	66.91	100.00	91.91	73.33	66.91
e	89.99	85.56	80.00	59.35	89.99	85.56
o	83.33	75.92	77.78	43.21	83.33	75.92
Overall	80.00	73.66	88.23	56.36	80.00	73.66

Fig 6 Classified SPOT image with *GA classifier*

propagation is seen to perform worse than that derived using NCA. A significance of this investigation is that the said analogy between the *VGA-classifier* and MLP will augment the application domain of *VGA-classifier* to those areas where MLP has widespread use.

Finally, we provide Fig 6 as an example of pixel classification on a SPOT image representing a part of the city of Calcutta by the said methodology for $H = 15$. Here the classes considered are pure water, turbid water, concrete, vegetation, habitation, open space and roads (including bridges), and unclassified region. The different regions like Hooghly river, Howrah bridge (Rabindra Setu), race course, Gardenreach dock yard, and a few other important roads and canals are found to be detected well. A comparison of the output with those of Bayes rule, k-NN rule and a multivalued recognition method [90,91] is provided in [186]. Note that the information in bands near infra red, red and green are considered as input features.

Before leaving this section, let us mention a few attempts made by others in this line. Srikanth *et al* [187] described a genetic algorithmic approach to classification, both crisp and fuzzy where clusters in pattern space are approximated by ellipses or sets of ellipses in two dimensions and ellipsoids in general. A variable number of ellipsoids is searched for, which collectively classify a training set of objects. Since the number of ellipsoids is not fixed a priori, a variable length chromosome is considered. The performance of the classifier is illustrated using two two-dimensional data sets. Performance on a four-dimensional data set is compared with that of various neural network architectures. The comparison is marginally in favour of the genetic technique presented.

The problem of editing for the k-NN rule is addressed by Kuncheva [188]. Results in a problem with highly overlapping classes are amongst others compared to those obtained with the Multiedit scheme. Although the differences are not statistically significant, the author presents use of a GA as an alternative to other techniques.

Piper has described [189] the use of GA in chromosome classification. The fitness function takes into account constraints imposed by the context of a metaphase cell, as well as similarity of homologous. Comparison with previous classification methods demonstrate equal classification accuracy. Incorporation of the homologue similarity constraint does not substantially improve the error rate.

Dev and Murthy [190] presented a GA solution to the problem of optimal assignment of production rules in a knowledge base to a variable number of partitions. Partitioning of a knowledge base is an important problem since an efficient solution can increase its performance both in the compilation phase and in the execution phase. It is beneficial to the maintainability of the knowledge base. The authors demonstrate that the solutions obtained using a GA compare favourably with those obtained with a clustering algorithm.

Selection of a subset of principal components for classification using GA is made in [191]. Since the search space depends on the product of the number of classes and the number of original features, this selection process by conventional means may be very computationally expensive. Results on two data sets with small and large cardinalates are presented.

Recently, Murthy and Chowdhury [192] have used GAs for finding optimal clusters, without the need for searching all possible clusters. Experimental results show that the GA based scheme may improve the final output of the K-means algorithm where an improvement is possible. GA is also used for image enhancement [193] through selection of approximate operators, fractal image compression and magnification [194,195]. It is found that the method greatly decreases the search space for finding the self similarities in an image. A new method for feature selection using a self cross-over operator is reported in [196].

6 Integration of the soft computing tools and hybrid systems

In this section we describe some ways how the various soft computing tools can be made to work synergetically (not competitively) to build efficient hybrid systems [2, 197]. Let us first of all consider the problem of integrating fuzzy sets and artificial neural networks. As mentioned before, fuzzy set theory provides an approximate but effective and flexible way of representing, manipulating and utilizing vaguely-defined data and information, and of describing the behaviors of systems which are too complex or too ill-defined to admit of precise mathematical analysis by classical methods and tools. Successful use of fuzzy logic to create many commercial products has been made recently in Japan. This, in turn, has increased interest among engineers, researchers and company executives to understand and explore further this technology. Though the approach tries to model the human thought process in a decision-making system, it has no relation with the architecture of the human neural information processing system, nor does it make into consideration the information storage technique of human beings, and some times it is computationally intensive.

Human intelligence and discriminating power, on the other hand, are mainly attributed to the massively connected network of biological neurons in the human brain. We mentioned earlier that attempts have recently been made to emulate electronically the architecture and information representation scheme of human neural network under the name *artificial neural network* models. The collective computational abilities of the densely interconnected nodes or processors may provide a material technique, at least to a great extent, for solving highly complex real life problems in a manner such as a human being does.

It, therefore, appears that integration of the merits of these two technologies can provide more intelligent systems (in terms of parallelism, fault tolerance, adaptivity and uncertainty management) to handle real life recognition problems. These promises have motivated (during the last 7-10 years) a large number of researchers to exploit these modern concepts for solving real world problems, leading to the development of a new paradigm called *neuro-fuzzy* computing. Besides the generic advantages of parallelism, fault-tolerance and uncertainty handling, the neuro-fuzzy paradigm some times provides some application specific advantages. The fusion or integration made so far, can be categorized as follows [198].

(*i*) Incorporating Fuzziness into Neural Network Frameworks: fuzzyifying the input data, assigning fuzzy labels to training samples, and obtaining outputs of neural networks in terms of fuzzy sets [143,147,148,199-203].

(*ii*) Designing Neural Networks Guided by Fuzzy Logic Formalism: designing neural networks to implement fuzzy logic and fuzzy decision making, and to realize membership functions representing fuzzy sets [155,204-213].

(*iii*) Changing the Basic Characteristics of the Neurons: neurons are designed to perform various operations used in fuzzy set theory (like fuzzy union, intersection, aggregation) instead of doing the standard multiplication and addition operations [146,214-216].

(*iv*) Making the Individual Neurons Fuzzy: the input to a neuron is a fuzzy set and the output also is a fuzzy set. The activity of networks involving fuzzy neurons is a fuzzy process [217,218].

(*v*) Measures of Fuzziness as Error or Instability of a Network: using the fuzziness/uncertainty measures of a fuzzy set to model the error or instability or energy function of a neural network based system [138,219,220].

As an example of the application specific merit of neuro-fuzzy computing, let us consider the work of Pal and Mitra [184,201], which falls under category (*i*). Here, the concept of fuzzy sets is introduced in various stages of multi-layer perceptrons and Kohonen's model for designing both supervised and unsupervised fuzzy classifiers for uncertainty analysis and recognition of patterns. The self-organizing netwrok developed for fuzzy partitioning of patterns takes membership values corresponding to linguistic properties (e.g., small, medium and high) along with some contexual class information as input. An index of disorder based on mean square distance between input and weight vectors has been defined in order to provide a quantitative measure for the ordering of the output space. The method based on the multi-layer perceptron, on the other hand, involves assignment of appropriate weights to the back-propagated errors depending on the membership values at the corresponding outputs. Its input can also be in terms of linguistic properties. Incorporation of fuzziness also makes the system less oscillatory in addition to providing superior performance for overlapping classes. The system is found to be robust with respect to fuzzification of input properties. These modified versions also provided better performance for certain non-convex decision regions [200] as compared to the classical methods and conventional connectionist approaches as they incorporate more local information of the feature space by decomposing it into 3^N (N being the dimension of feature space) sub-regions through the properties *small, medium and high*. Similar concept of fuzzy labels has also been utilized in [202] for learning in a network.

An application of the fuzzy multi-layer perceptron and Kohonen's model for designing classificatory connectionist expert systems is described in [147,148]. These models can handle impreciseness in the input representation and provide output decisions as class labels with certainty factors. The input can be in quantitative, linguistic or set form. These models have been found to be useful in querying, inferencing and generating if-then rules for medical diagnosis with incomplete symptoms. An attempt has also been made in [221] to build an expert system which can handle both numerical and linguistic medical data.

Another application of the aforesaid fuzzy MLP is described recently in [222] for designing a knowledge-based system for classification and rule generation. Here the knowledge collected from a data set is initially encoded among the connection weights in terms of class *a priori* probabilities. This encoding also includes incorporation of hidden nodes corresponding to both the pattern classes and their complementary regions. The network is then refined during training. Negative rules corresponding to a pattern not belonging to a class can also be obtained. These are useful for inferencing in ambiguous cases. Speed of learning is also improved. Both convex and concave decision regions are considered in the process. Recently, an application of fuzzy integral for combining an ensemble of networks for multispectral data classification is reported in [223].

A fuzzy neural network using logical operations, viz, *max-min* and *product-probabilistic sum* has also been developed by Mitra and Pal [146] under category (*iii*) for both classification and rule generation with linguistic properties as input. For the purpose of rule generation and inferencing, the user could be queried for more essential feature information in the case of partial input. The model is likely to be suitable for data-rich environments. The use of logical neurons helps in generating rules in more appropriate forms and makes its hardware realization easier. The system is robust with respect to input fuzzification.

Finally for an example in the category (v), we consider the work of Ghosh *et al* [138] which incorporated various fuzziness measures in a multi-layer network to make it able to perform (unsupervised) self-organizing tasks in image processing, in general, and object extraction in particular. The network architecture is basically a feed-forward one with a feed-back path. In each layer every neuron corresponds to an image pixel. Each neuron is connected to the corresponding neuron in the previous layer and its neighbors. The status of neurons in the output layer is described as the memberhsip value to a *fuzzy set* representing *object regions*. A fuzziness measure (e.g., index of fuzziness and entropy [16] of this set is used to quantify system error (instability of the network) and it is back-propagated to correct weights.

After the weights have been adjusted the output of the neurons in the output layer is fed back to the corresponding neurons in the input layer. The second pass is then continued with this as input. The iteration (updating of weights) is continued until the network stabilizes, i.e., the error value (measure of fuzziness) becomes negligible. This integration makes it possible for a layered network (which is usually used as a supervised classifier) to act as an unsupervised one, in addition to providing a robust and noise-insensitive segmentation algorithm. Here the neuro-fuzzy integration also provides an application specific advantage. As such, we cannot always generalize this concept to use an MLP as an unsupervised classifier.

Recently, a similar concept of using fuzziness measure as an objective function of a network is utilized [219,220] for feature selection and extraction problem under both supervised and unsupervised modes. Here a fuzzy feature evaluation index is minimized with a layered network using gradient descent technique such that the network parameters determine the importance of different input features or the extracted features.

Note that the aforesaid attempts of integration can also be broadly divided in two ways:

- a neural network equipped with the capability of handling fuzzy information for extending its application domain, and
- a fuzzy system augmented by neural networks to enhance its flexibility, speed and adaptability.

The former may be called *fuzzy neural networks*, while the latter *neural fuzzy systems*. Further, these integrations are mainly made in the field of pattern recognition and to some extent in fuzzy logic control. Literature on neuro-fuzzy image processing is not adequate at this moment. More details on neuro-fuzzy computing to pattern recognition are available in [198].

Let us now discuss some of the recent attempts for designing *genetic-neural* systems for the synthesis of artificial neural networks architecture using GAs, as an example of another kind of integration under the framework of soft computing. Methods for designing neural network architectures using GAs are primarily divided into two parts: in one part the GA replaces the learning method to find appropriate connection weights of some predefined architecture. In another part, GA itself is used to find the architecture (connectivity) and it is evaluated using some learning algorithm.

Whitley, *et al* [158] suggested a way to apply GAs to neural network problems and they showed how genetic search can be improved to achieve more accurate and consistent performance. The method is able to optimize the weighted connections for small neural network problems as follows:

(*a*) For the purpose of weight optimization, neural networks with prefixed connectivity and predefined range of weight values were taken. Each weight value was encoded into 8 bits (−127 to 127 with 0 twice). For each feed-forward net, sum square error was calculated for the training pattern set (weight values of each net was obtained using GA). This methdology was tested for signal detection.

(*b*) To optimize the connectivity of a NN from a predefined maximally connected feed forward net, the method finds out combination of connections (using GAs) which can quickly and accurately learn (learning took place using BP) using GENITOR. An explicit mechanism to reward small, fast, and well-trained network was adopted. To reduce the learning time, instead of assigning random weights initially, the weight values of already trained fully connected net were assigned to each connection. The resultant net has direct i/o connection (which has less probability to get stuck at local optima).

In [163] Bhandari *et al* incorporated GAs to find out the optimal set of weights (biases) in a layered network. Weighted mean square error over the training examples was used as the fitness measure. They introduced a new concept of selection, called *non-linear selection*, which enhances genetic homogeneity into the population and speeds up searching. They experimented on both linearly separable and non-linearly separable pattern sets.

Harp and Samad also suggested an approach [161] to evolve the architecture of neural networks and their learning parameters using GAs. The network was trained separately by backpropagation algorithm. Each chromosome contains two parts: (1) *fixed length area parameter specification* which corresponds to layer number, the number of units in it, how the units are organized, learning parameters and threshold values; (2) *projection specification field* which shows the way of connection between layers. Since the total number of layers may vary, the chromosomal length was also variable. They also introduced a modified crossover operator which exchanges homologous segments of two chromosomes or networks. Architecture, connection strengths and thresholds of individual neurons of feedforward neural networks were simultaneously found out in a recent investigation [160] without using any error propagation algorithm.

An attempt is also made by Pal *et al* to evolve Hopfield type optimum neural network architecture for extracting object regions from gray images using GAs [164]. Each chromosome represents a network architecture. For an $m \times n$ image, each pixel (neuron) being connected to at most k of its neighbours, the length of the chromosome is $m \times n \times k$. If a neuron is connected to any of its neighbours, the corresponding bit of the chromosome is set to 1, else 0. Initial population of the GA is generated randomly. Each network is then allowed to run for object extraction as in [137]. The energy value obtained at the converged state of a network is taken as the index of fitness of the corresponding chromosome (minimum energy corresponds to maximum fitness) for its selection for the next generation. Crossover and mntation operations are performed on these selected chromosomes to get new offspring (new architectures). The whole process is continued for a number of generations until the GA converges. The best chromosome evoluted at the final population represents the optimum architecture for object extraction.

This GA based technique has been able to evolve network architectures whose connectivity is about two-third of the requirement of the corresponding fixed fully connected ones in order to produce comparable segmented output. It is also found from the output images that, for highly corrupted image, this method provides better results than the corresponding fixed fully connected architecture.

Similarly, consider cellular neural networks (CNN) where setting up a CNN needs a proper selection of circuit parameters of cells. The set of circuit parameter is called cloning template which determines the dynamics of the network and its application domain. There is no standard method for selecting cloning template automatically for a CNN, although some heuristics have been suggsted [31] for this task. GAs have also been used recently [224,225] with fuzzy fitness function for classification of objects and background by CNNs. The grayness and spatial ambiguity measures have been used as the basis of the fitness function. It has been found from the results that the template selected by GAs performs consistently well and is more immune to noise as compared to the heuristically selected template. This investigation can be classified under the category of *neuro-fuzzy-genetic* integration. Another typical example of integrating fuzzy logic, ANN and GAs is shown in Fig 7. Here the different parameters (e.g., membership functions for low (L), medium (M), and High (H) at the input, the connection weights and biases of ANN, and the boundaries of the output classes or decisions) of a fuzzy neural network are adjusted (tuned) by GAs for its optimum performance.

There are many such attempts being reported in conference proceedings and journals [226-228]. Note that, the tuning of fuzzy membership functions with GAs can be classified under the framework of *fuzzy-genetic* integration [2,40]. Recently, the theory of rough sets [229] is also seen to draw the attention of researchers as a soft computing tool for building intelligent systems in conjunction with others. The work of Banerjee *et al* [230,231] is an example in this regard where a Rough-Fuzzy-MLP model is designed for knowledge encoding and classification. Rough set theory is exploited for efficient knowledge encoding. Crude domain knowledge is extracted from the data set in the form of rules. The syntex of these rules automatically determines the

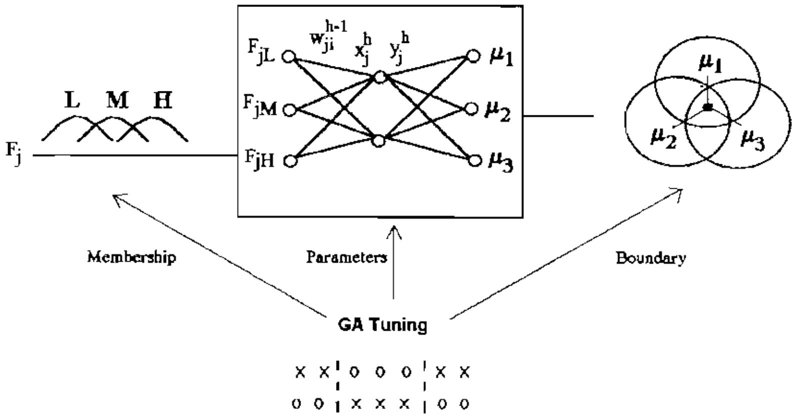

Fig 7 Tuning of parameters of a fuzzy neural network with GA

TABLE 8 Recognition scores (%) for Vowel (F_1, F_2, F_3 features)

Attributes for		Rough-Fuzzy MLP						Fuzzy
c_1, c_3, c_4			$H_1 \wedge L_2 \wedge L_3$;	$M_3 \vee H_3$;		$M_3 \vee H_3$		M
c_2		$M_1 \vee L_3$				$M_1 \vee M_3$		L
c_5			M_3				H_3	P
c_6		$M_1 \vee M_3$	$M_1 \vee M_3 \vee H_3$	$M_1 \vee M_3$	$M_1 \vee H_3$	$M_1 \vee M_3$	$M_1 \vee M_3 \vee H_3$	
# links		18	19	18	18	18	19	90
Training		85.48	80.65	81.11	80.19	83.18	82.72	80.88
T	∂	51.4	21.6	56.8	43.2	54.1	59.5	21.6
e	a	84.4	88.9	82.2	88.9	82.2	75.6	82.2
s	i	94.1	85.9	95.3	85.9	94.1	87.1	94.1
t	u	90.2	86.6	87.8	87.8	90.2	87.8	87.8
s	e	84.0	97.2	78.3	93.4	82.1	84.9	88.7
e	o	93.9	93.9	95.1	93.9	93.9	93.9	95.1
t	Net	86.27	85.13	85.13	86.27	85.81	84.44	84.44

appropriate number of hidden nodes while the dependency factors are used in the initial weight encoding. To demonstrate its effectiveness, consider the vowel data (Fig 3). Twelve sets of rules for six vowel classes (c_1, c_2, ... , c_6, *say*) were generated. This led to twelve possible fuzzy-MLP encoding. Table 8 demonstrates the performance of some of such networks along with their comparison with that of the fuzzy-MLP (with no initial knowledge weight encoding). A sample set of dependency rules generated for the six classes is

$$H_1 \wedge L_2 \wedge L_3 \rightarrow c_1, M_1 \vee L_3 \rightarrow c_2, M_3 \vee H_3 \rightarrow c_3,$$

$$M_3 \vee H_3 \rightarrow c_4, M_3 \rightarrow c_5 \text{ and } M_1 \vee M_3 \rightarrow c_6.$$

The network generated from this set of rules is shown in Fig 8. Here L, M, and H denote the fuzzy linguistic terms low, medium and high respectively. The subscripts 1,2,3, associated with them, represent the three features of vowel. Some more attempts of integrating rough sets with fuzzy sets are available in [232-234].

7 Conclusions

We have discussed on some of the recent aspects of soft computing paradigm, its primary constituting tools and relevance to pattern recognition. The role of these tools for exploiting the tolerance for imprecision, uncertainty, approximate reasoning and partial truth in order to achieve robustness, tractability and low cost solutions in pattern recognition problems is described. As the significance of fuzzy set theory to pattern recognition problems is adequately established since late sixties, we have given here sufficient references for the convenience of readers. Research in artificial neural networks both in theory and applications is in full swing and has reached almost its peak stage. This is evident through publications of several journals, special issues and books. The issue of its hybridization with fuzzy logic and GAs is addressed in [17,30,40,235-238] reasonably. Since GA is relatively a much newer technology to pattern recognition scientists, more emphasis is given in this article in demonstrating its effectiveness for designing a non-parametric classification methodology. Various enhanced versions of the classifier are mentioned.

The applicability of the classifier to different kinds of data sets with nonlinear boundaries between classes is explained. Some of the application specific advantages of integrating fuzzy logic with artificial neural networks, and genetic algorithms with artificial neural networks, besides the generic merits are provided. Research is going on extensively towards developing various hybrid systems involving the merits of the individual technology

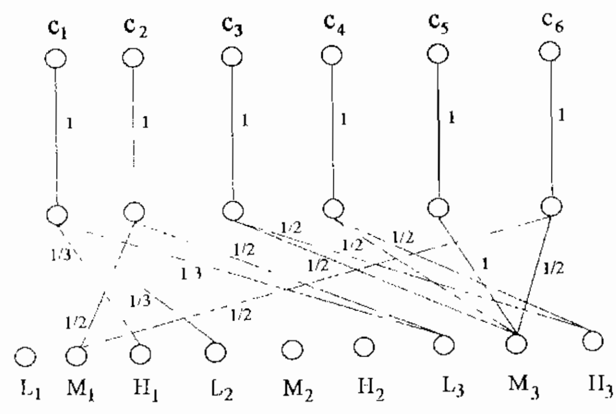

Fig 8 Network encoding

synergetically. It needs many more investigations and research work to keep the promise of GAs in helping other tools for hybridization. For example, operators emulating the human genetic systems need to be defined with schema analysis. Similarly, the problem of determining the genetic parameters (which is done mostly heuristically, at present) automatically needs to be addressed extensively. Hybridization should ensure that it provides application specific merits, besides the generic advantages. The theory of rough sets (a tool for handling uncertainties arising from granularity in the domain of discourse i.e., from the indiscernibility between objects in a set) is seen recently to draw the attention of researchers with a hope to build more efficient intelligent systems in the framework of soft computing. However, this is still in its infancy for pattern recognition problems.

It may be noted that soft computing can be viewed as the key ingredient of real world computing (RWC), which is capable of distributed representation of information, massively parallel processing and learning and adaptability in order to achieve flexibility in information processing. Therefore the growth of information technology in terms of computing power ranges from conventional computing (whose kernel is data processing), fifth generation computing systems (whose kernel is knowledge based information processing) to RWC (whose kernel is flexible information processing). *Thus, as it stands, the soft computing research will not only continue to remain in the forefront line for the next decade, but also will play a key role in the development of future technology including sixth generation computing systems.*

ACKNOWLEDGEMENTS

The author expresses his heartfelt thanks to the distinguished members of the IETE awards committee for selecting him for the *20th Ramlal Wadhwa Gold Medal* considering his contributions during the last ten years. Typing of this manuscript by Mr I Dutta is also acknowledged.

REFERENCES

1. L A Zadeh, Fuzzy Logic, Neural Networks and Soft Computing, *Communications of the ACM*, vol 37, pp 77-84, 1994.

2. S K Pal & N R Pal, Soft Computing: Goals, Tools and Feasibility, *J Inst Electron and Telecom Engrs (Spl issue on neural networks)*, vol 42, pp 335-347, 1996.

3. R O Duda & P E Hart, *Pattern Classification and Scene Analysis*, John Wiley, NY, 1973.

4. J T Tou & R C Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, Reading, MA, 1974.

5. A Rosenfeld & A C Kak, *Digital Picture Processing*, Academic Press, NY, 1982.

6. R C Gonzalez & P Wintz, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1987.

7. D H Ballard & C M Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

8. D Marr, *Vision*, W H Freeman, NY, 1982.

9. L A Zadeh, Fuzzy Sets, *Information and Control*, vol 8, pp 338-353, 1965.

10. D Dubois & H Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, NY, 1980.

11. J C Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, NY, 1981.

12. A Kandel, *Fuzzy Techniques in Pattern Recognition*, Wiley Interscience, NY, 1982.

13. A Kandel, *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley, Reading, Massachusetts, 1986.

14. A Kaufmann, *Fuzzy Subsets-Fundamental Theoretical Elements*, Academic Press, NY, 1980.

15. A Kaufmann & M Gupta, *Introduction to Fuzzy Mathematics*, Van Nostrand Reinhold, NY, 1985.

16. S K Pal & D Dutta Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*, Wiley (Halsted), NY, 1986.

17. J C Bezdek & S K Pal, (Eds), *Fuzzy Models for Pattern Recognition : Methods that Search for Structures in Data*, IEEE Press, NY, 1992.

18. G J Klir & T Folger, *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, NJ, 1988.

19. G J Klir & B Yuan, *Fuzzy Sets and Fuzzy Logic - Theory and Applications*, Prentice Hall, NY, 1995.

20. L A Zadeh, K S Fu, K Tanaka & M Shimura (Eds), *Fuzzy Sets and Their Applications to Cognitive and Decision Process*, Academic Press, London, 1975.

21. W Pedrycz, Fuzzy Sets in Pattern Recognition : Methodology and Methods, *Pattern Recognition*, vol 23, pp 121-146, 1990.

22. R R Yager & L A Zadeh (Eds), *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, Kluwer Academic, Boston, Massachusetts, 1992.

23. D E Rumelhart, J McClelland & the PDP Research Group, *Parallel Distributed Processing: Explorations in the Micro-structure of Cognition*, vol 1, MIT Press, Cambridge, Massachusetts, 1986.

24. T Kohonen, *Self-organization and Associative Memory*, Springer Verlag, Berlin, 1989.

25. J J Hopfield, Neurons with Graded Response have Collective Computational Properties like those of Two-state Neurons, *Proc National Academy of Sciences*, vol 81, pp 3088-3092, 1984.

26. K Fukushima, Neocognitron : A Self-organizing Multilayer Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, *Biological Cybernetics*, vol 36, pp 193-202, 1980.

27. S Grossberg (Ed), *Neural Networks and Natural Intelligence*, MIT Press, Reading, MA, 1988.

28. Y H Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley, Reading, Massachusetts, 1987.

29. P D Wassermann, *Neural Computing : Theory and Practice*, Van Nostrand Reinhold, NY, 1990.

30. B Kosko, *Neural Networks and Fuzzy Systems*, Prentice-Hall, NJ, 1992.

31. L O Chua & L Yang, Cellular Neural Network : Thoery, *IEEE Trans Circuits and Systems*, vol 35, pp 1257-1272, 1988.

32. A Ghosh, N R Pal & S K Pal, Neural Computing: An Introduction and Some Applications, *Students' Jounral Inst of Electron and Telecom Engrs* (Invited Paper), vol 35, pp 105-125, 1994.

33. B D Ripley, *Pattern Recognition and Neural Network*, Cambridge University Press, NY, 1996.

34. S Haykin, Neural Networks : *A Comprehensive Foundation*, Macmillan College Publishing Co, NY, 1994.

35. L Davis (Ed), *Genetic Algorithms and Simulated Annealing*, Pitman Publishing, London, 1987.

36. D E Goldberg, *Genetic Algorithms : Search, Optimization and Machine Learning*, Addison Wesley, Reading, Massachusetts, 1989.

37. Z Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Springer Verlag, Berlin, 1992.

38. L Davis (Ed), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, NY, 1991.

39. J H Holland, *Adaptation in Neural and Artificial Systems*, The University of Michigan Press, Ann Arbor, 1975.

40. S K Pal & P P Wang (Eds), *Genetic Algorithms for Pattern Recognition*, CRC Press, Boca Raton, 1996.

41. J M S Prewitt, Object Enhancement and Extraction, in *Picture Processing and Psycho-Pictorics*, (Eds B S Lipkin & A Rosenfeld), Academic Press, NY, 1970.

42. A Rosenfeld, Fuzzy Geometry of Image Subsets, *Pattern Recognition Letters*, vol 2, pp 311-317, 1984.

43. S K Pal & R A King, On Edge Detection of X-ray Images Using Fuzzy Set, *IEEE Trans Patt Anal & Mach Intell*, vol 5, pp 69-77, 1983.

44. S K Pal, Fuzzy Sets in Image Processing and Recognition, *Proc Fuzz-IEEE '92 (The 1st IEEE Int Conf on Fuzzy Systems)*, San Diego, pp 119-126, 1992.

45. S K Pal & A Ghosh, Fuzzy Geometry in Image Analysis, *Fuzzy Sets and Systems*, vol 48, pp 23-40, 1992.

46. S K Pal, Fuzzy Set Theoretic Tools for Image Analysis, in *Advances in Electronics and Electron Physics*, vol 88, (Eds P Hawkes & B Kazan), Academic Press, NY, pp 247-296, 1994.

47. A De Luca & S Termini, A Definition of a Non Probabilistic Entropy in the Setting of Fuzzy Set Theory, *Information and Control*, vol 20, pp 301-312, 1972.

48. S K Pal, A Note on the Quantitative Measure of Image Enhancement Through Fuzziness, *IEEE Trans Patt Anal & Mach Intell*, vol 4, pp 204-208, 1982.

49. C A Murthy, S K Pal & D Dutta Majumder, Representation of Fuzzy Operators Using Ordinary Sets, *IEEE Trans Syst, Man and Cyberns*, vol 17, pp 840-847, 1987.

50. N R Pal & S K Pal, Entropic Thresholding, *Signal Processing*, vol 16, pp 97-108, 1989.

51. N R Pal & S K Pal, Object-Background Segmentation Using New Definitions of Entropy, *IEE Proceedings-E*, vol 136, pp 284-295, 1989.

52. N R Pal & S K Pal, Image Model, Poisson Distribution and Object Extraction, *Int J Patt Recog and Artificial Intell*, vol 5, pp 459-483, 1991.

53. N R Pal & S K Pal, Entropy : A New Definition and its Applications, *IEEE Trans Syst, Man and Cyberns*, vol 21, pp 1260-1270, 1991.

54. B Kosko, Fuzzy Entropy and Conditioning, *Information Sciences*, vol 40, pp 165-174, 1986.

55. S K Pal & A Ghosh, Index of Area Coverage of Fuzzy Image Subsets and Object Extraction, *Pattern Recognition Letters*, vol 11, pp 831-841, 1990.

56. S K Pal & A Rosenfeld, Image Enhancement and Thresholding by Optimization of Fuzzy Compactness, *Pattern Recognition Letters*, vol 7, pp 77-86, 1988.

57. D Dubois & M C Jaulent, A Generalized Approach to Parameter Evaluation in Fuzzy Digital Pictures, *Pattern Recognition Letters*, vol 6, pp 251-259, 1987.

58. S K Pal, R A King & A A Hashim, Automatic Gray Level Thresholding Through Index of Fuzziness and Entropy, *Pattern Recognition Letter*, vol 1, pp 141-146, 1983.

59. T L Huntsberger, C L Jacobs & R L Cannon, Iterative Fuzzy Image Segmentation, *Pattern Recognition*, vol 18, pp 131-138, 1985.

60. M Trivedi & J C Bezdek, Low-level Segmentation of Aerial Images with Fuzzy Clustering, *IEEE Trans Syst, Man and Cyberns*, vol 16, pp 589-598, 1986.

61. R L Cannon, R Dave, J C Bezdek & M Trivedi, Segmentation of a Thematic Mapper Image Using the Fuzzy c-means Clustering Algorithm, *IEEE Trans Geoscience and Remote Sensing*, vol 24, pp 400-407, 1986.

62. L O Hall, A M Bensaid, L P Clarke, R P Velthuizn, M S Silbiger & J C Bezdek, A Comparison of Neural Networks and Fuzzy Clustering Techniques in Segmenting Magnetic

Resonance Images of the Brain, *IEEE Trans Neural Networks*, vol 3, pp 672-682, 1992.

63. S K Pal & A Ghosh, Image Segmentation Using Fuzzy Correlation, *Information Sciences*, vol 62, pp 223-250, 1992.

64. C A Murthy & S K Pal, Fuzzy Thresholding : Mathematical Framework, Bound Functions and Weighted Moving Average Technique, *Pattern Recognition Letters*, vol 11, pp 197-206, 1990.

65. C A Murthy & S K Pal, Histogram Thresholding by Minimizing Gray Level *Fuzziness, Information Sciences*, vol 60, pp 107-135, 1992.

66. S K Pal, Fuzziness, Image Information and Uncertainty Management in Pattern Recognition (Invited Paper), *J Sci Indust Research*, vol 51, pp 71-98, 1992.

67. S K Pal & A Dasgupta, Spectral Fuzzy Sets and Soft Thresholding, *Information Sciences*, vol 65, pp 65-97, 1992.

68. N R Pal & S K Pal, A Review on Image Segmentation Techniques, *Pattern Recognition*, vol 26, pp 1277-1294, 1993.

69. X W Xie, An Information Measure for a Color Space, *Fuzzy Sets and Systems*, vol 36, pp 157-165, 1990.

70. S K Pal, Image Enhancement and a Quantitative Index Using Fuzzy Sets, *Int J Syst Sci*, vol 18, pp 1783-1797, 1987.

71. S K Pal & N R Pal, Segmentation Based on Measures of Contrast, Homogeneity, and Region Size, *IEEE Trans Syst, Man and Cyberns*, vol 17, pp 857-868, 1987.

72. M K Kundu & S K Pal, Automatic Selection of Object Enhancement Operator with Quantitative Justification Based on Fuzzy Set Theoretic Measures, *Pattern Recognition Letters*, vol 11, pp 811-829, 1990.

73. S K Pal, Fuzzy Tools for the Management of Uncertainty in Pattern Recognition, Image Analysis, Vision and Expert System, *Int J Syst Sci*, vol 22, pp 511-549, 1991.

74. N R Pal & S K Pal, Higher Order Fuzzy Entropy and Hybrid Entropy of a Set, *Information Sciences*, vol 61, pp 211-231, 1992.

75. C A Murthy & S K Pal, Bounds for Membership Function: Correlation Based Approach, *Information Sciences*, vol 65, pp 143-171, 1992.

76. N R Pal & S K Pal, Some Properties of the Exponential Entropy, *Information Sciences*, vol 66, pp 119-137, 1992.

77. K Tanaka & M Sugeno, A Study on Subjective Evaluations of Printed Color Images, *Int J Approx Reason*, vol 5, pp 213-222, 1991.

78. S K Pal & A Rosenfeld, A Fuzzy Medial Axis Transformation Based on Fuzzy Disk, *Pattern Recognition Letters*, vol 12, pp 585-590, 1991.

79. S K Pal & L Wang, Fuzzy Medial Axis Transformation (FMAT): Practical Feasibility, *Fuzzy Sets and Systems*, vol 50, pp 15-34, 1992.

80. S K Pal, Fuzzy Skeletonization of Image, *Pattern Recognition Letters*, vol 10, pp 17-23, 1989.

81. E T Lee, Shape-oriented Chromosome Classification, *IEEE Trans Syst, Man and Cyberns*, vol 5, pp 629-632, 1975.

82. A Pathak & S K Pal, Fuzzy Grammar in Syntactic Recognition of Skeletal Maturity from X-ray, *IEEE Trans Syst, Man and Cyberns*, vol 16, pp 657-667, 1986.

83. S K Pal & A Bhattacharyya, Pattern Recognition Techniques in Analyzing the Effect of Thiourea in Brain Neurosecretory Cells, *Pattern Recognition Letters*, vol 11, pp 443-452, 1990.

84. S K Pal, Uncertainty Management in Space Station Autonomous Research : Pattern Recognition Perspective, *Information Sciences*, vol 72, pp 1-63, 1993.

85. J C Bezdek & P F Castelez, Prototype Classification and Feature Selection with Fuzzy Sets, *IEEE Trans Syst, Man and Cyberns*, vol 7, pp 87-92, 1977.

86. S K Pal & B Chakraborty, Fuzzy Set Theoretic Measures for Automatic Feature Evaluation, *IEEE Trans Syst, Man and Cyberns*, vol 16, pp 754-760, 1986.

87. S K Pal, Fuzzy Set Theoretic Measures for Automatic Feature Evaluation-II, *Information Sciences*, vol 64, pp 165-179, 1992.

88. R K De, N R Pal & S K Pal, Feature Analysis : Neural Network and Fuzzy Set Theoretic Approaches, *Pattern Recognition*, vol 30, pp 1579-1590, 1997.

89. A K Nath, S W Liu & T T Lee, On Some Properties of Linguistic Variables as Input, *Fuzzy Sets and Systems*, vol 17, pp 297-311, 1985.

90. D P Mandal, C A Murthy & S K Pal, Formulation of a Multivalued Recognition System, IEEE *Trans Syst, Man and Cyberns*, vol 22, pp 607-620, 1992.

91. D P Mandal, C A Murhty & S K Pal,Theoretical Performance of a Multivalued Recognition System, *IEEE Trans Syst, Man and Cyberns*, vol 24, pp 1001-1021, 1994.

92. S K Pal & D P Mandal, Linguistic Recognition System Based on Approximate Reasoning, *Information Sciences*, vol 61, pp 135-161, 1992.

93. M Roubens, Pattern Classification Problems and Fuzzy Sets, *Fuzzy Sets and Systems*, vol 1, pp 239-253, 1978.

94. J M Keller, M R Gray & J A Givens, A Fuzzy *k*-nearest Neighbour Algorithm, *IEEE Trans Syst, Man and Cyberns*, vol 15, pp 580-585, 1985.

95. R Bellman, R Kalaba & L Zadeh, Abstraction and Pattern Classification, *J Mathematical Analysis Applications*, vol 13, pp 1-7, 1966.

96. B B Devi & V V S Sarma, A Fuzzy Approximation Scheme

for Sequential Learning in Pattern Recognition, *IEEE Trans Syst, Man and Cyberns*, vol 16, pp 668-679, 1986.

97. D P Mandal, C A Murthy & S K Pal, Determining the Shape of a Pattern Class From Sampled Points in R^2, *Int J General Systems*, vol 20, pp 307-339, 1992.

98. E Ruspini, A New Approach to Clustering, *Information and Control*, vol 15, pp 22-32, 1969.

99. M Windham, Geometric Fuzzy Clustering Algorithms, *Fuzzy Sets and Systems*, vol 10, pp 271-279, 1983.

100. E Backer & A K Jain, A Clustering Performance Measure Based on Fuzzy Set Decomposition, *IEEE Trans Patt Anal & Mach Intell*, vol 3, pp 66-75, 1981.

101. M Roubens, Fuzzy Clustering Algorithms and their Cluster Validity, *European J Operation Research*, vol 10, pp 294-301, 1982.

102. X L Xie & G Beni, A Validity Measure for Fuzzy Clustering, *IEEE Trans Patt Anal & Mach Intell*, vol 13, pp 841-847, 1991.

103. R Dave, Fuzzy *c*-shell Clustering and Applications to Circle Detection in Digital Images, *Int J General Systems*, vol 16, pp 343-355, 1990.

104. S K Pal & S Mitra, Fuzzy Dynamic Clustering Algorithm, *Pattern Recognition Letters*, vol 11, pp 525-535, 1990.

105. D P Mandal, C A Murthy & S K Pal, Utility of Multiple Choices in Detecting Ill-defined Roadlike Structures, *Fuzzy Sets and Systems*, vol 64, pp 213-228, 1994.

106. D P Mandal, C A Murthy & S K Pal, A Remote Sensing Application of a Fuzzy Classifier, *Int J Uncertainty, Fuzziness and Knowledge-Based Systems*, vol 2, pp 287-295, 1994.

107. S K Pal & A B Leigh, Motion Frame Analysis and Scene Abstraction : Discrimination Ability of Fuzziness Measures, *Journal of Intelligent & Fuzzy Systems*, vol 3, pp 247-256, 1995.

108. D P Mandal, C A Murthy & S K Pal, Analysis of IRS Imagery for Detecting Man-made Objects with a Multivalued Recognition System, *IEEE Trans Syst, Man and Cyberns*, Part A, vol 26, pp 241-247, 1996.

109. S K Pal & S Mitra, Noisy Fingerprint Classification using Multilayered Perceptron with Fuzzy Geometrical and Textual Features, *Fuzzy Sets and Systems*, vol 80, pp 121-132, 1996.

110. S K Pal & S N Sarbadhikari, Classification of Distorted Overlapping Fingerprints with Fuzzy Geometrical Features, *Int J Knowledge-based Intelligent Engg Syst*, vol 1, pp 120-137, 1997.

111. A Malaviya & L Peters, Fuzzy Feature Description of Handwriting Patterns, *Pattern Recognition*, vol 30, pp 1591-1609, 1997.

112. Special Issue on Neural Network Hardware Design, *IEEE Trans Neural Networks*, vol 3, 1992.

113. U Ramacher & U Ruckert (Eds), *VLSI Design of Neural Networks*, Kluwer Academic Press, Massachusetts, 1991.

114. M E Robinson, H Yoneda & E Sanches-Sinencio, A Modular CMOS Design of a Hamming Network, *IEEE Trans Neural Networks*, vol 3, pp 444-456, 1992.

115. H Harrer, J A Nosset & R Stelzt, An Analog Implementation of Discretetime Cellular Neural Networks, *IEEE Trans Neural Networks*, vol 3, pp 466-476, 1992.

116. R P Lippmann, Pattern Classification using Neural Networks, *IEEE Communications Magazine*, pp 47-64, 1989.

117. D J Burr, Experiments on Neural Net Recognition of Spoken and Written Text, *IEEE Trans Acoustics, Speech, and Signal Processing*, vol 36, pp 1162-1168, 1988.

118. J Lee, R C Weger, S K Sengupta & R M Welch, A Neural Network Approach to Cloud Classification, *IEEE Trans Geoscience and Remote Sensing*, vol 28, pp 846-855, 1990.

119. Y Hirari & Y Tsukui, Position Invariant Pattern Matching by Neural Network, *IEEE Trans Syst, Man and Cyberns*, vol 20, pp 816-825, 1990.

120. T Kanaoka, R Chellapa, M Yoshitaka & S Tomita, A Higher-order Neural Network for Distortion Invariant Pattern Recognition, *Pattern Recognition*, vol 13, pp 837-841, 1992.

121. J Basak, S Chaudhury, S K Pal & D Dutta Majumder, Matching of Structural Shape Descriptions with Hopfield Net, *Int J Patt Recog and Arti Intell*, vol 7, pp 377-404, 1993.

122. S Mitra, S K Pal & M K Kundu, Fingerprint Classification Using Fuzzy Multilayer Perceptron, *Neural Computing and Applications*, vol 2, pp 227-233, 1994.

123. J Basak, N R Pal & S K Pal, A connetionist System for Learning and Recognition of Structures : Application to Handwritten Characters, *Neural Networks*, vol 8, pp 643-657, 1995.

124. J Basak & S K Pal, X-tron : An Incremental Connectionist Model for Category Perceptron, *IEEE Trans Neural Networks*, vol 6, pp 1091-1108, 1995.

125. J Basak & S K Pal, PsyCOP : A Psychologically Motivated Connectionist System for Object Perception, *IEEE Trans Neural Networks*, vol 6, pp 1337-1354, 1995.

126. J Basak, C A Murthy & S K Pal, A Self-organizing Network for Mixed Category Perception, *Neurocomputing*, vol 10, pp 341-358, 1996.

127. S Mitra, S N Sarbadhikari & S K Pal, An MLP-based Model for Identifying qEEG in Depression, *Int J Biomedical Computing*, vol 43, pp 179-187, 1996.

128. A K Jain, J Mao & K M Mohiuddin, Artificial Neural Networks : A Tutorial, *IEEE Computer*, pp 31-44, March 1996.

129. G W Cottrel & P Munro, Principal Component Analysis of Images via Back Propagation, *SPIE : Visual Communication and Image Processing*, vol 1001, pp 1070-1077, 1988.

130. C T Chen, E C Tsao & W C Lin, Medical Image Segmentation by a Constraint Satisfaction Neural Network, *IEEE Trans Nuclear Science*, vol 38, pp 678-686, 1991.

131. S P Luttrell, Image Compression Using a Multilayer Neural Network, *Pattern Recognition Letters*, vol 10, pp 1-7, 1989.

132. B S Manjunath, T Simchony & R Chellappa, Stochastic and Deterministic Networks for Texture Segmentation, *IEEE Trans Acoustics, Speech and Signal Processing*, vol 38, pp 1039-1049, 1990.

133. R H Silverman & A S Noetzel, Image Processing and Pattern Recognition in Ultrasonograms by Backpropagation, *Neural Networks*, vol 3, pp 593-603, 1990.

134. J Basak, B Chanda & D Dutta Majumder, On Edge and Line Linking in Graylevel Images with Connectionist Models, *IEEE Trans Syst, Man and Cyberns*, vol 24, pp 413-428, 1994.

135. A Ghosh, N R Pal & S K Pal, Image Segmentation Using Neural Networks, *Biological Cybernetics*, vol 66, pp 151-158, 1991.

136. A Ghosh & S K Pal, Neural Network, Self-Organisation and Object Extraction, *Pattern Recognition Letters*, vol 11, pp 387-397, 1992.

137. A Ghosh, N R Pal & S K Pal, Object Background Classification Using Hopfield Type Neural Network, *Int J Patt Recog and Arti Intell*, vol 6, pp 989-1008, 1992.

138. A Ghosh, N R Pal & S K Pal, Self-organization for Object Extraction using Multilayer Neural Network and Fuzziness Measures, *IEEE Trans Fuzzy Systems*, vol 1, pp 54-68, 1993.

139. A Ghosh, N R Pal & S K Pal, Modeling of Component Failure in Neural Networks for Robustness Evaluation : An Application to Object Extraction, *IEEE Trans Neural Networks*, vol 6, pp 648-656, 1995.

140. N M Nasrabadi & W Li, Object Recognition by a Hopfield Neural Network, *IEEE Trans Syst, Man and Cyberns*, vol 21, pp 1523-1535, 1991.

141. N M Nasrabadi & C Y Choo, Hopfield Network for Stereo Vision Correspondence, *IEEE Trans Neural Networks*, vol 3, pp 5-13, 1992.

142. T J Sejnowski & C R Rosenberg, Parallel Networks that Learn to Pronounce English Text, *Complex Systems*, vol 1, pp 145-168, 1987.

143. S I Gallant, Connectionist Expert System, *Communication of the Association for Computing Machinery*, vol 31, pp 152-169, 1988.

144. L Shastri, A Connectionist Approach to Knowledge Representation and Limited Inference, *Cognitive Science*, vol 12, pp 331-392, 1988.

145. H Narazaki & L Ralescu, A Connectionist Approach for Rule-based Inference Using an Improved Relaxation, *IEEE Trans Neural Networks*, vol 3, pp 741-751, 1992.

146. S Mitra & S K Pal, Logical Operation Based Fuzzy MLP for Classification and Rule Generation, *Neural Networks*, vol 7, pp 353-373, 1994.

147. S Mitra & S K Pal, Fuzzy Self Organization, Inferencing and Rule Generation, *IEEE Trans Syst, Man and Cyberns, Part A : Systems and Humans*, vol 26, pp 608-620, 1996.

148. S Mitra & S K Pal, Fuzzy Mulit-layer Perceptron, Inferencing and Rule Generation, *IEEE Trans Neural Networks*, vol 6, pp 51-63, 1995.

149. S Mitra & S K Pal, Neuro-fuzzy Expert Systems : Relevance, Features and Methodologies, *J Inst Electron and Telecom Engrs (Spl issue on neural networks)*, vol 42, pp 335-347, 1996.

150. D H Nguyen & B Widrow, Neural Networks for Self-learning Control Systems, *IEEE Trans Control Systems Magazine*, vol 10, pp 18-23, 1990.

151. R R Yager, Fuzzy Logic Controllers Using Neural Network, *Fuzzy Sets and Systems*, vol 48, pp 53-64, 1992.

152. H R Berenji & P Khedkar, Learning and Tuning of Fuzzy Logic Controllers through Reinforcement, *IEEE Trans Neural Networks*, vol 3, pp 724-740, 1992.

153. A F Rocha, I R Guilherme, M Theoto, A M K Miyadahira & M S Koizumi, A Neural Net for Extracting Knowledge from Natural Language Data Base, *IEEE Trans Neural Networks*, vol 3, pp 819-828, 1992.

154. J L McClelland, Putting Knowledge in its Place : A Scheme for Parallel Processing Structures on the Fly, *Cognitive Science*, vol 9, pp 113-146, 1985.

155. H Takagi, N Suzuki, T Koda & Y Kojima, Neural Networks Designed on Approximate Reasoning Architecture and Their Applications, *IEEE Trans Neural Networks*, vol 3, pp 752-760, 1992.

156. A Weibel, T Hanazawa, G Hinton, K Shikano & K J Lang, Phoneme Recognition Using Time Delay Neural Networks, *IEEE Trans Acoustics, Speech and Signal Processing*, vol 37, pp 328-339, 1989.

157. M A Franzani, Speech Recognition with Back-propagation, *Proc 9th Annual Conference of the Engineering in Medicine and Biology Society*, pp 1702-1703, 1987.

158. D Whitley, T Starkweather & C Bogart, Genetic Algorithms and Neural Networks : Optimizing Connections and Connectivity, *Parallel Computing*, vol 14, pp 347-361, 1990.

159. J D Schaffer, R A Caruana & L J Eshelman, Using Genetic Search to Exploit the Emergent Behavior of Neural Networks, *Physica D*, vol 42, pp 244-248, 1990.

160. S Saha & J P Christensen, Genetic Design of Sparse Feedforward Neural Networks, *Information Sciences*, vol 79, pp 191-200, 1994.

161. S A Harp & T Samad, Genetic Synthesis of Neural Network Architecture, in *Handbook of Genetic Algorithms*, (Ed L Davis), Van Nostrand Reinhold, NY, pp 202-221.

162. V Maniezzo, Genetic Evolution of the Topology and Weight Distribution of Neural Networks, *IEEE Trans Neural Networks*, vol 5, pp 39-53, 1994.

163. S K Pal & D Bhandari, Selection of Optimal Set of Weights in a Layered Network Using Genetic Algorithms, *Information Sciences*, vol 80, pp 213-234, 1994.

164. S K Pal, S De & A Ghosh, Designing Hopfield Type Networks Using Genetic Algorithms and its Comparison with Simulated Annealing, *Int J Patt Recog Art Intell*, vol 11, pp 447-461, 1997.

165. S Forrest, (Ed), Proceedings of *5th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, California, 1993.

166. Z Michalewicz & C Z Janikow, Genetic Algorithms for Numerical Optimization, *Statistics and Computing*, vol 1, pp 75-91, 1991.

167. Special Issue on Genetic Algorithms, (Ed E S Gelsema), *Pattern Recognition Letters*, vol 16, no 8, 1995.

168. B P Buckles & F E Petry (Eds), *Genetic Algorithms*, IEEE Computer Society Press, Los Alamitos, 1994.

169. W Pedrycz (Ed), *Fuzzy Evolutionary Computation*, Kluwer Academic, Boston, 1997.

170. D Bhandari, C A Murthy & S K Pal, Genetic Algorithm with Elitist Model and its Convergence. *Int J Pat Recog & Arti Intell*, vol 10, pp 731-747, 1996.

171. C A Murthy, D Bhandari & S K Pal, Optimal Stopping Time for Genetic Algorithm with Elitist Model, *Fundamenta Informaticae*, (communicated).

172. M Srinivas & L M Patnaik, Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms, *IEEE Trans Syst, Man, and Cybern*, vol 24, pp 656-667, 1994.

173. S De, S K Pal & A Ghosh, Genotypic and Phenotypic Assortative Mating in Genetic Algorithm, *Information Sciences*, (accepted).

174. L J Eshelman, Preventing Premature Convergence in the Genetic Algorithms by Preventing Incest, *Proc 4th Int Conf Genetic Algorithms*, San Diego, July 1991, pp 115-122.

175. D Bhandari, N R Pal & S K Pal, Directed Mutation in Genetic Algorithms, *Information Sciences*, vol 79, pp 251-270, 1994.

176. S De, A Ghosh & S K Pal, Fitness Evaluation in Genetic Algorithms with Ancestors' Influence, in *Genetic Algorithm for Pattern Recognition*, (Eds S K Pal and P P Wang), CRC Press, Boca Raton, pp 1-23, 1996.

177. S Bandyopadhyay, C A Murthy & S K Pal, Pattern Classification with Genetic Algorithms, *Pattern Recognition Letters*, vol 16, pp 801-808, 1995.

178. S Bandyopadhyay & S K Pal, Pattern Classification with Genetic Algorithms : Incorporation of Chromosome Differentiation, *Pattern Recognition Letters*, vol 18, pp 119-131, 1997.

179. S K Pal, S Bandyopadhyay & C A Murthy, Genetic Algorithms for Generation of Class Boundaries, *IEEE Trans Syst, Man and Cyberns*, (accepted).

180. S Bandyopadhyay, S K Pal & C A Murthy, Simulated Annealing Based Pattern Classification, *Information Sciences* (communicated).

181. C A Murthy, S Bandyopadhyay & S K Pal, Genetic Algorithm Based Pattern Classification : Relationship with Bayes Classifier, in *Genetic Algorithm for Pattern Recognition*, (Eds S K Pal & P P Wang), CRC Press, Boca Raton, pp 127-144, 1996.

182. S Bandyopadhyay, C A Murthy & S K Pal, Pattern Classification using Genetic Algorithms : Determination of *H*, *Pattern Recognition Letters* (communicated).

183. S K Pal & D Dutta Majumder, Fuzzy Sets and Decision Making Approaches in Vowel and Speaker Recognition, *IEEE Trans Syst, Man and Cyberns*, vol 7, pp 625-629, 1977.

184. S K Pal & S Mitra, Multi-layer Perceptron, Fuzzy Sets and Classification, *IEEE Trans Neural Networks*, vol 3, pp 683-697, 1992.

185. S Bandyopadhyay & S K Pal, Relation Between VGA Classifier and MLP : Determination of Network Architecture, *Pattern Recognition*, (comunicated).

186. S Bandyopadhyay, *Pattern Classificatian using Genetic Algorithms*, PhD Thesis, Indian Statistical Institute, Calcutta, 1998.

187. R Srikanth, R George, N Warsi, D Prabhu, F Petry & B Buckles, A Variable-length Genetic Algorithm for Clustering and Classification, *Pattern Recognition Letters*, vol 16, pp 789-800, 1995.

188. L I Kuncheva, Editing the k-nearest Neighbors Rule by a Genetic Algorithm, *Pattern Recognition Letters*, vol 16, pp 809-814, 1995.

189. J Piper, Genetic Algorithm for Applying Constraints in Chromosome Classification, *Pattern Recognition Letters*, vol 16, pp 857-864, 1995.

190. K Dev & C R Murthy, A Genetic Algorithm for the Knowledge Base Partitioning Problem, *Pattern Recognition Letters*, vol 16, pp 873-879, 1995.

191. M Prakash & M N Murty, A Genetic Approach for (near) Optimal Subsets of Principal Components for Discrimination, *Pattern Recognition Letters*, vol 16, pp 781-787, 1995.

192. C A Murthy & N Chowdhury, In Search of Optimal Clusters using Genetic Algorithms, *Pattern Recognition Letters*, vol 17, pp 825-832, 1996.

193. S K Pal, D Bhandari & M K Kundu, Genetic Algorithms for Optimal Image Enhancement, *Pattern Recognition Letters*, vol 15, pp 261-271, 1994.

194. S K Mitra, C A Murthy & M K Kundu, Technique for Fractal Image Compression using Genetic Algorithm, *IEEE Trans Image Processing* (to appear).

195. S K Mitra, C A Murthy & M K Kundu, Fractal Image Magnification Technique, *Proc 9th International Symposium on System-Modelling-Control*, Zakopane, Poland, 1998 (accepted).

196. N R Pal, S Nandi & M K Kundu, Self-Cross-over : A New Genetic Operator and its Application to Feature Selection, *Int J Syst Sci*, (accepted).

197. S K Pal & P K Srimani, Neurocomputing : Motivation, Models and Hybridization, *IEEE Computer*, vol 29, pp 24-28, 1996.

198. S K Pal & A Ghosh, Neuro-fuzzy Computing for Image Processing and Pattern Recognition, *Int J Syst Sci*, vol 27, pp 1179-1193, 1996.

199. J M Keller & D J Hunt, Incorporating Fuzzy Membership Functions into the Perceptron Algorithm, *IEEE Trans Patt Anal & Mach Intell*, vol 7, pp 693-699, 1985.

200. S K Pal & S Mitra, Fuzzy Versions of Kohonen's Net and MLP Based Classification : Performance Evaluation for Certain Non-convex Decision Regions, *Information Sciences*, vol 76, pp 297-337, 1994.

201. S Mitra & S K Pal, Self-Organizing Neural Network as a Fuzzy Classifier, *IEEE Trans Syst, Man and Cyberns*, vol 24, pp 385-399, 1994.

202. L O Hall, Learning on Fuzzy Data with a Back propagation Seheme, *Proceedings NAFIPS*, Missouri-Columbia, 1991, pp 329-332.

203. B R Kammerer, Incorporating Uncertainty in Neural Networks, *Int J Patt Recog and Arti Intell*, vol 6, pp 179-192, 1992.

204. J M Keller & H Tahani, Implementation of Conjunctive and Disjunctive Fuzzy Logic Rules with Neural Network, *Int J Approx Reason*, vol 6, pp 221-240, 1991.

205. J M Keller, R R Yagar & H Tahani, Neural Network Implementation of Fuzzy Logic, *Fuzzy Sets and Systems*, vol 45, pp 1-12, 1992.

206. J M Keller &.R Krishnapuram, Evidence Aggregation Networks for Fuzzy Logic Inference, *IEEE Trans Neural Networks*, vol 3, pp 761-769, 1992.

207. H Takagi & I Hayashi, Artificial Neural Network Driven Fuzzy Reasoning, *Int J Approx Reason*, vol 5, pp 191-212, 1991.

208. T L Huntsberger & P Ajjimarangsee, Parallel Self-organizing Feature Maps for Unsupervised Pattern Recognition, *Int J Gen Syst*, vol 16, pp 357-372, 1990.

209. S C Newton, S Pemmaraju & S Mitra, Adaptive Fuzzy Leader Clustering of Complex Data Sets in Pattern Recognition, *IEEE Trans Neural Networks*, vol 3, pp 974-800, 1992.

210. P K Simpson, Min-max Neural Network - Part I: Classification, *IEEE Trans Neural Networks*, vol 3, pp 776-786, 1992.

211. P K Simpson, Min-max Neural Network - Part II : Clustering, *IEEE Trans Fuzzy Systems*, vol 1, pp 32-45, 1993.

212. E C K Tsao, J C Bezdek & N R Pal, Fuzzy Kohonen Clustering Networks, *Pattern Recognition*, vol 27, pp 757-764, 1994.

213. N R Pal, J C Bezdek & E Tsao, Generalized Clustering Networks and Kohonen's Self-organizing Scheme, *IEEE Trans Neural Networks*, vol 4, pp 549-558, 1993.

214. R Krishnapuram & J Lee, Fuzzy Connective Based Hierachical Aggregation Networks for Decision Making, *Fuzzy Sets and Systems*, vol 46, pp 11-27, 1992.

215. R Krishnapuram & J Lee, Fuzzy Set Based Hierarchical Networks for Information Fusion in Computer Vision, *Neural Networks*, vol 5, pp 335-350, 1992.

216. W Pedrycz & H C Card, Linguistic Interpretation of Self-organizing Maps, *Proc 1st IEEE Int Conf Fuzz Syst*, San Diego, California, pp 371-378, 1992.

217. S C Lee & E T Lee, Fuzzy Neural Networks, *Mathematical Biosciences*, vol 23, pp 151-177, 1975.

218. T Yamakawa & S Tomada, A Fuzzy Neuron and its Applications to Pattern Recognition, *Proc 3rd Int Fuzzy Syst Assoc Cong*, Seattle, Washington, pp 30-38, 1989.

219. S K Pal, J Basak & R K De, Fuzzy Feature Evaluation Index and Connectionist Realization, *Information Sciences*, (accepted).

220. J Basak, R K De & S K Pal, Fuzzy Feature Evaluation Index and Connectionist Relization-II : Theoretical Analysis, *Information Sciences* (accepted).

221. M B Gorzalczay & M D McLeish, Fuzzy Neural Network Methodology Applied to Medical Diagnosis, *Proc NAFIPS '92*, Puerto Vallarta, Mexico, 1992, pp 266-275.

222. S Mitra, R K De & S K Pal, Knowledge Based Fuzzy MLP for Classification and Rule Generation, *IEEE Trans Neural Networks*, vol 8, pp 1338-1350, 1997.

223. A Senthil Kumar, S K Basu & K L Majumder, Robust Classification of Multispectral Data Using Multiple Neural Networks and Fuzzy Integral, *IEEE Trans Geoscience Remote Sensing*, vol 35, pp 787-790, 1997.

224. S K Pal, D Bhandari, P Harish & M K Kundu, Cellular Neural Networks, Genetic Algorithms and Object Extraction, *Far East Journal of Mathematical Sciences*, vol 1, pp 139-155, 1993.

225. S K Pal & D Bhandari, Genetic Algorithms with Fuzzy

Fitness Function for Object Extraction Using Cellular Neural Networks, *Fuzzy Sets and Systems*, vol 65, pp 129-139, 1994.

226. M Sarkar & B Yegnanarayana, Feedforward Neural Networks Configuration Using Evolutionary Programming, *Proc IEEE Int Conf Neural Networks*, Houston, TX, 1997, pp 438-443.

227. M Sarkar & B Yegnanarayana, An Evolutionary Programming-based Probabilistic Neural Network Construction Technique, *Proc IEEE Int Conf Neural Networks*, Houston, TX, 1997, pp 456-461.

228. N Kasabov & M Watts, Genetic Algorithms for Structural Optimization, Dynamic Adaptation and Automated Design of Fuzzy Neural Networks, *Proc IEEE Int Conf Neural Networks*, Houston, TX, 1997, pp 2546-2549.

229. Z Pawlak, *Rough Sets, Theoretic Aspects of Reasoning about Data*, Kluwer Academic, Dordrecht, 1991.

230. M Banerjee, S Mitra & S K Pal, Rough Fuzzy MLP : Knowledge Encoding and Classification. *IEEE Trans Neural Networks*, (revised and communicated).

231. M Banerjee, S Mitra & S K Pal, Knowledge-Based Fuzzy MLP with Rough Sets, *Proc Int Conf Neural Networks*, Houston, TX, 1997, pp 499-504.

232. M Banerjee & S K Pal, Roughness of a Fuzzy Set, *Information Sciences*, vol 93, pp 235-246, 1996.

233. M Sarkar & B Yegnanarayana, Rough-fuzzy Set Theoretic Approach to Evaluate the Importance of Input Features in Classification, *Proc IEEE Int Conf Neural Networks*, Houston, TX, 1997, pp 1590-1595.

234. S K Pal & A Skowron (Eds), *Fuzzy Sets, Rough Sets and Decision Making Processes*, Springer Verlag, Singapore (to appear).

235. Special Issue on Neural Networks : Theory and Application, (Eds S K Pal & P K Srimani), *IEEE Computer*, vol 29, no 3, 1996.

236. Special Issue on Neural Networks, (Ed S K Pal), *J Inst Electron Telecom Engrs*, vol 42, no 3 & 4, 1996.

237. Special Issue on Fuzzy Systems, (Ed J C Bezdek), *IEEE Trans Neural Networks*, vol 3, no 5, 1992.

238. Special Issue on Soft Computing, (Ed S K Pal), *Fundamenta Informaticae* (to appear).

AUTHOR

Sankar K Pal is a Professor and Founding Head of Machine Intelligence Unit at the Indian Statistical Institute, Calcutta. He obtained MTech and PhD in Radiophysics and Electronics in 1974 and 1979 respectively, from the university of Calcutta, India. In 1982 he received another PhD in Electrical Engineering along with DIC from Imperial College, University of London. In 1986 he was awarded Fulbright Post-doctoral Visiting Fellowship to work at the University of California, Berkeley and the University of Maryland, College Park, USA. In 1989 he received an NRC-NASA Senior Research Award to work at the NASA Johnson Space Center, Houston, Texas, USA . He is appointed a Distinguished Visitor of IEEE Computer Society (USA) for the Asia-Pacific Region.

He received the 1990 Shanti Swarup Bhatnagar Prize in Engineering Sciences, 1993 Jawaharlal Nehru Fellowship, 1993 Vikram Sarabhai Research Award, 1993 NASA Tech Brief Award, 1994 IEEE Transaction on Neural Networks outstanding paper Award, 1995 NASA Patent Application Award, and 1997 IETE - Ram Lal Wadhwa Gold Medal. His research interests mainly include Pattern Recogntion, Image Processing, Neural Nets, Genetic Algorithms, and Fuzzy Sets and Systems. He is a co-author of the book Fuzzy Mathematical Approach to Pattern Recognition, John Wiley & Sons (Halsted), NY, 1986 and a co-editor of three books Fuzzy Models for Pattern Recognition, IEEE Press, NY, 1992, Genetic Algorithms for Pattern Recognition, CRC Press, Boca Raton, 1996, and Rough-Fuzzy Hybridization : New Trend in Decision Making, Springer Verlag, Signapore 1998.

Prof. Pal is a Fellow of the IEEE, USA, Indian National Science Academy, Indian Academy of Sciences, National Academy of Sciences, India, Indian National Academy of Engineering, Institute of Engineers, India, and the IETE, a Member, Executive Advisory Editorial Board, IEEE Trans Fuzzy Systems and International Journal of Approximate Reasoning, North Holland, and an Associate Editor, IEEE Trans Neural Networks, Pattern Recognition Letters, Neurocomputing, Applied Intelligence, Information Sciences: Applications, Int. J. Knowledge Based Intell Engg Syst., and Far-East Journal of Mathematical Sciences. He was also the guest editor of IEEE Computer special issue on Neural Networks : Theory and Practice, March 1996, and JIETE special issue on Neural Networks, July-October, 1996.