

Self-Organizing Neural Network As A Fuzzy Classifier

Sushmita Mitra, *Student Member, IEEE*, and Sankar K. Pal, *Fellow, IEEE*

Abstract—This paper describes a *self-organizing artificial neural network*, based on Kohonen's model of self-organization, which is capable of handling fuzzy input and of providing fuzzy classification. Unlike conventional neural net models, this algorithm incorporates fuzzy set-theoretic concepts at various stages. The input vector consists of membership values for linguistic properties along with some contextual class membership information which is used during self-organization to permit efficient modeling of fuzzy (ambiguous) patterns. A new definition of gain factor for weight updating is proposed. An index of disorder involving mean square distance between the input and weight vectors is used to determine a measure of the ordering of the output space. This controls the number of sweeps required in the process. Incorporation of the concept of fuzzy partitioning allows natural self-organization of the input data, especially when they have ill-defined boundaries. The output of unknown test patterns is generated in terms of class membership values. Incorporation of fuzziness in input and output is seen to provide better performance as compared to the original Kohonen model and the hard version. The effectiveness of this algorithm is demonstrated on the speech recognition problem for various network array sizes, training sets and gain factors.

1. INTRODUCTION

ARTIFICIAL NEURAL nets [1]–[5] are highly parallel interconnections of simple processing elements or *neurons* that function as a collective system with neurons interacting via feedback connections. There exist various problems in pattern recognition and image processing that humans seem more efficient in solving as compared to computers. Neural nets may be seen perhaps as an attempt to emulate such human performance. These networks can be broadly categorized as those that learn adaptively by updating their connection weights during training and those whose parameters are time-invariant. We consider a network of the first kind here.

Self-organization [4] refers to the ability of a neural net to elucidate or reproduce some fundamental organizational property of the input data without benefit of supervised training procedures. In Kohonen's model, the network automatically performs a mapping transformation from an input space to generally a lower-dimensional output space such that the latter acquires the same topological ordering as the former.

The benefit of neural nets [1]–[6] lies in the high computation rate provided by their inherent massive parallelism. This allows real-time processing of huge data sets with proper hardware backing. All information is stored in distributed

form among the various connection weights. The redundancy of interconnections produces a high degree of robustness resulting in a *graceful degradation* of performance in the case of damage to a few nodes or links.

It should be mentioned that human reasoning is somewhat fuzzy in nature. The utility of fuzzy sets [7]–[9] lies in their ability, to a reasonable extent, to model the *uncertain* or ambiguous data so often encountered in real life. Hence, to enable the system to deal with the ambiguous (ill-defined) data in an effective manner, one may incorporate the concept of fuzzy sets into the neural network.

The present work discusses a self-organizing neural network model that performs fuzzy classification. It is an attempt to extend Kohonen's model [4] by incorporating fuzzy set-theoretic concepts [7]–[10] at various stages. In the process, a separate testing phase is added to evaluate the performance of the proposed classifier in recognizing a separate set of test patterns. We consider a single layer two-dimensional rectangular array of neurons with short range lateral feedback interconnections between neighboring units.

The network under consideration passes through two stages, *viz.*, self-organization and testing. In the first stage a set of training data is used by the network to initially self-organize the connection weights and finally *calibrate* the output space. During this stage the weight vector most similar to the input pattern vector is rotated toward the latter. The neighboring weight vectors are also rotated, but by a lower amount. After a number of sweeps through the training set the output space becomes appropriately organized. An index of disorder is computed to provide an evaluation of this ordering. The network is now supposed to encode the input space information among its connection weights. By calibration we refer to the labeling of the neurons, after self-organization, relative to the training pattern classes. This procedure also provides some qualitative assessment of the topological ordering of the output space as compared to the input data space.

During training, the input vector also includes some contextual information regarding the finite output membership of the pattern to one or more classes. Compared to the conventional two-state system, which assigns membership to one class only and uses no class information in the input, the proposed technique produces a more efficient modeling in cases where the feature space has overlapping or ill-defined clusters. However, during self-organization, this part of the input vector is assigned a lower weight to allow the linguistic and/or quantitative input properties to dominate.

During calibration, only the class membership information in the input vector is used (in *crisp* form) while the input feature information is kept clamped at zero. In the conventional Kohonen's model, after self-organization, the training pattern vectors are used to label the neurons to which they are mapped. This gives the ordering of the pattern classes in the output space. In the proposed model, the labeling of the output neurons is determined solely by the contextual class information associated with the training pattern vectors. This is termed *calibration* of the neurons. Each neuron is labeled by the pattern class for which it generates the highest response. This corresponds to a *hard* partitioning of the neurons. A *fuzzy* partitioning of the output space is also generated to produce an appropriate topological ordering with fuzzy data.

In the second stage a separate set of test patterns is supplied to the network and the resulting neuronal outputs verified against the calibrated output map. This is an extension to the conventional Kohonen's model which basically performs a clustering operation. The proposed model, on the other hand, is designed to be a classifier. The *calibrated* neurons, *self-organized* by the training set, are used to evaluate the recognition capability (using *best match*) of the said trained neural net on the test set. Now the input vector contains only the feature information. A confusion matrix is generated to evaluate the classification performance (based on *best match*) of the network on the test set. The output is generated in terms of fuzzy class membership values.

The proposed fuzzy neural network model is capable of handling input features presented in quantitative and/or linguistic form. The components of the input vector may consist, for instance, of the membership values to the overlapping partitions of linguistic properties *low*, *medium*, and *high* corresponding to each input feature. This creates the possibility of incorporating linguistic information into the model, if necessary, and enhances its robustness in handling imprecise or uncertain input specifications.

The effectiveness of the proposed model is demonstrated on the speech recognition problem where the classes have ill-defined, *fuzzy* boundaries. Comparison is made with the standard Bayes' classifier and the conventional Kohonen's net, and the performance of the proposed model is found to be quite satisfactory.

Given the burgeoning interest in fuzzy self-organizing maps [11]–[13], it is worth highlighting the major contribution of the proposed work. Basically, the Kohonen clustering network is used here as a symbol map. There are phenomena which are inherently fuzzy but which are associated with physical manifestations that can be characterized quite precisely by physical measurements. Clustering or classifying solely on the basis of these physical measurements is not useful, however, because meaningful clusters can be constructed only with the assistance of additional factors which cannot be elucidated directly from these physical measurements. Human language, probably at all levels but especially in the area of phonology, is perhaps the best example of such a phenomenon. Thus, while a listener recognizes a phoneme from physical cues alone, exactly which phoneme class a particular conflation of physical features is assigned to by a listener depends

on factors which are not inherent in these physical features (e.g., the formant values used here), but which depend on physically extraneous factors such as (in particular but not limited to) the language the listener assumes is being spoken. There are also, for many reasons, variations among speakers such as are evident in the data used in this paper. Thus, assignment of speech sounds to phonemes yields clusters which are fuzzy at the very least in the sense that different listeners may disagree on what they believe themselves to be hearing and that different speakers may produce different physical manifestations of the same phoneme. The essential properties of phoneme clusters, therefore, must be elucidated by appeal to essentially psycholinguistic experimentation of one kind or another. Now, how can one build a self-organizing network which can perform this same classification? Simply by doing exactly what we have done, which is to replace the arbitrary encoding of the abstract portion of the data vectors with fuzzy class memberships. Note that this violates Ritter and Kohonen's "no information about similarities between the items" condition ([14], p. 247), but it does not matter, because a kind of orthogonality is maintained by the fact that x_a (attribute part) and x_s (symbol part) of the data vectors here are characterized by different "levels" of description (phonetic and phonemic). The value of this approach is manifested in calibration (clustering, labeling) and in classification, since the organized network yields a good fuzzy clustering of the neurons after calibration and functions as an effective fuzzy classifier. Thus, where there is reason to believe that the elements of x_a and x_s relate to each other not so much as purely arbitrary and purely physical (or at least less arbitrary, in some sense) but rather as two levels of abstraction, and where there is reason to believe that at least one of the levels (the "higher" one) is fuzzy, the fuzzification of the x_a is justifiable and yields excellent results. Attempts at crisp calibration and/or the use of purely arbitrary class labels (as in the pure Ritter and Kohonen approach, where the labels (the semantic concepts) are not connected to each other except through the data vectors they label) in such cases will prove to be fruitless. Note that this does indeed amount to a kind of partial supervision as we have suggested, but it is an extremely interesting kind of partial supervision in that it arises from reasonable assumptions about the nature of human language itself (i.e., its multi-level properties) and not directly from expert intervention (i.e., the learning is guided not by intelligence but by intuition)!

II. KOHONEN'S NEURAL NETWORK MODEL

The essential constituents of Kohonen's neural network model are as follows [3], [4], [15]–[17]:

- an array of neurons receiving coherent inputs and computing a simple output function,
- a mechanism for comparing the neuronal outputs to select the neuron producing maximum output,
- a local interaction between the selected neuron and its neighbors,
- an adaptive mechanism that updates the interconnection weights.

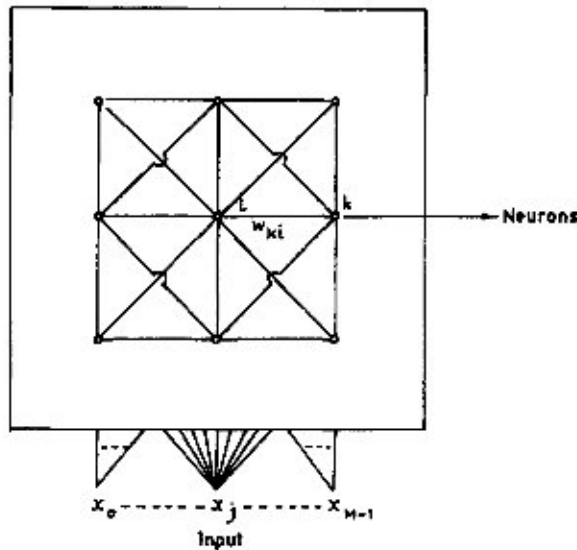


Fig. 1. Kohonen's neural network model [1]. M inputs connected to an $N \times N$ array of neurons via variable connection weights. The fixed connection weight between neurons i and k is w_{ki} .

Consider the self-organizing network given in Fig. 1. Let M input signals be simultaneously incident on each of an $N \times N$ array of neurons. The output of the i th neuron is defined as

$$\eta_i(t) = \sigma \left[\mathbf{m}_i(t)^T \mathbf{x}(t) + \sum_{k \in S_i} w_{ki} \eta_k(t - \Delta t) \right] \quad (1)$$

where \mathbf{x} is the M -dimensional input vector incident on the neuron along the connection weight vector \mathbf{m}_i , k belongs to the subset S_i of neurons having interconnections with the i th neuron, w_{ki} denotes the fixed feedback coupling between the k th and i th neurons, $\sigma[\cdot]$ is a suitable sigmoidal output function, t denotes a discrete time index and T stands for the transpose.

If the best match between vectors \mathbf{m}_i and \mathbf{x} occurs at neuron c , then we have

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \|\mathbf{x} - \mathbf{m}_i\|, \quad i = 0, 1, \dots, N^2 \quad (2)$$

where $\|\cdot\|$ indicates the Euclidean norm.

The weight updating rule is given by [4,15] as

$$\mathbf{m}_i(t+1) = \begin{cases} \mathbf{m}_i(t) + \alpha(t)(\mathbf{x}(t) - \mathbf{m}_i(t)) & \text{for } i \in N_c \\ \mathbf{m}_i(t) & \text{otherwise} \end{cases} \quad (3)$$

where $\alpha(t)$ is a positive constant that decays with time and N_c defines a topological neighborhood around the maximally responding neuron c , such that it also decreases with time. (Note that $\alpha(t)$ is a particular case of the more general Gaussian term $h(x, t)$ [16]). Different parts of the network become selectively sensitized to different inputs in an ordered fashion so as to form a continuous map of the signal space. After a number of sweeps through the training data, with weight updating at each iteration obeying (3), the asymptotic values of \mathbf{m}_i cause the output space to attain proper topological ordering. This

is basically a variation of *unsupervised* learning. The self-organization using training patterns enables the ordering of the output neurons. These may then be calibrated with the class information by applying labeled training patterns at the input.

Kohonen's net has already been applied to a phonemic recognition problem [15] and in image compression [18].

III. PATTERN REPRESENTATION IN LINGUISTIC FORM

In conventional statistical or syntactic classifiers, the input patterns are quantitative (exact) in nature. The patterns possessing imprecise or incomplete input features (say, due to instrumental error or noise corruption) are generally ignored or discarded while designing these classifiers. Besides, the cost of extracting the exact value of a feature may sometimes be too high. In such cases it may become convenient to use linguistic variables and hedges [19] like *low*, *medium*, *high*, *very*, *more or less*, etc., to describe input feature information.

The proposed (fuzzy neural network model) is capable of handling both exact and inexact forms of the input features. Since it is easier to convert exact information into linguistic form than vice versa, we consider the major linguistic properties *low*, *medium*, and *high* as input. Any input feature value can be described in terms of some combination of membership values for these properties. Hence any imprecise input may also be assigned a set of membership values according to this concept.

Fuzzy Sets

In traditional two-state classifiers [20], [21] an element x either belongs or does not belong to a given class A ; thus, the characteristic function is expressed as

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

In real-life problems, however, the classes are often ill-defined, overlapping, or fuzzy, and a pattern point may belong to more than one class; in such situations, fuzzy set-theoretic techniques [7]-[10] can be very useful. In a fuzzy context, the pattern point x , belonging to the universe X , may be assigned a characteristic function value or grade of membership value $\mu_A(x)$ ($0 \leq \mu_A(x) \leq 1$) which represents its degree of membership in the fuzzy set A . This may be represented as

$$A = \{(\mu_A(x), x)\}, \quad x \in X \quad (4)$$

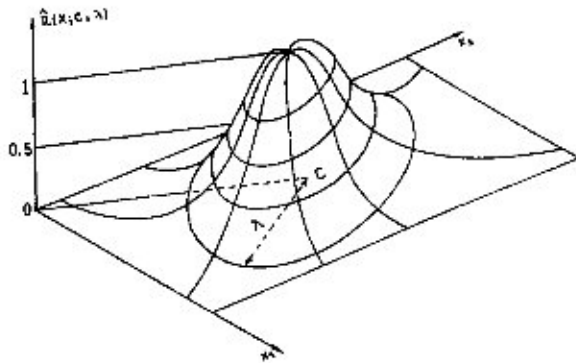
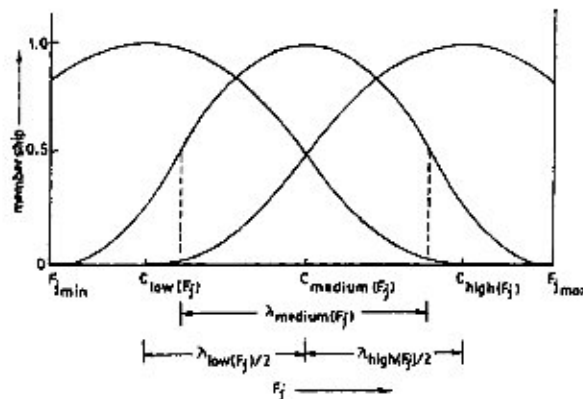
π Membership Function

The π -function, lying in the range [0,1], with $x \in \mathbb{R}^n$ is defined as [22]

$$\pi(\mathbf{x}; \mathbf{c}, \lambda) = \begin{cases} 2 \left(1 - \frac{\|\mathbf{x} - \mathbf{c}\|}{\lambda}\right)^2 & \text{for } \frac{\lambda}{2} \leq \|\mathbf{x} - \mathbf{c}\| \leq \lambda \\ 1 - 2 \left(\frac{\|\mathbf{x} - \mathbf{c}\|}{\lambda}\right)^2 & \text{for } 0 \leq \|\mathbf{x} - \mathbf{c}\| \leq \frac{\lambda}{2} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\lambda > 0$ is the radius of the π -function with \mathbf{c} as the central point at which $\pi(\mathbf{c}; \mathbf{c}, \lambda) = 1$. This is shown in Fig. 2 for $\mathbf{x} \in \mathbb{R}^2$.

A fuzzy set with membership function $\pi(\mathbf{x}; \mathbf{c}, \lambda)$ therefore represents a set of points clustered around \mathbf{c} . In the proposed

Fig. 2. π -function when $x \in \mathbb{R}^2$.Fig. 3. Coexistence structure of the compatibility functions for the linguistic properties *low*, *medium*, and *high*.

model we use the π -function (in the one-dimensional form) to assign membership values for the input features.

Incorporation of the Linguistic Concept

Each input feature F_j (in quantitative and/or linguistic form) can be expressed in terms of membership values indicating a degree of belonging to each of the linguistic properties *low*, *medium*, and *high*. Therefore an n -dimensional pattern $X_i = [F_{i1}, F_{i2}, \dots, F_{in}]$ may be represented as a $3n$ -dimensional [19] vector

$$X_i = [\mu_{low}(F_{i1})(X_i), \mu_{medium}(F_{i1})(X_i), \mu_{high}(F_{i1})(X_i), \dots, \mu_{high}(F_{in})(X_i)]. \quad (6)$$

Hence in trying to express an input X_i through its linguistic properties we are effectively dividing the dynamic range of each feature into three overlapping partitions. The sets *low*, *medium*, and *high* for each feature are represented by the π -function (5). Fig. 3 shows the coexistence structure of the various compatibility functions (π -functions) for a particular input feature F_j .

Choice of Parameters for the π -Functions: Let F_{jmax} and F_{jmin} denote the upper and lower bounds of the dynamic range of feature F_j considering all L pattern points. Then for the

three linguistic property sets we have

$$\lambda_{medium}(F_j) = \frac{1}{2}(F_{jmax} - F_{jmin})$$

$$c_{medium}(F_j) = F_{jmin} - \lambda_{medium}(F_j) \quad (7)$$

$$\lambda_{low}(F_j) = \frac{1}{f_{denom}}(c_{medium}(F_j) - F_{jmin})$$

$$c_{low}(F_j) = c_{medium}(F_j) - 0.5 * \lambda_{low}(F_j) \quad (8)$$

$$\lambda_{high}(F_j) = \frac{1}{f_{denom}}(F_{jmax} - c_{medium}(F_j))$$

$$c_{high}(F_j) = c_{medium}(F_j) + 0.5 * \lambda_{high}(F_j) \quad (9)$$

where $0.5 \leq f_{denom} \leq 1.0$ is a parameter controlling the extent of overlapping.

Unlike in [19], this combination of choices for the λ 's and c 's ensures that each quantitative input feature value x_j along the j th axis for pattern X_i is assigned membership value combinations in the corresponding 3-dimensional linguistic space of (6) in such a way that at least one of $\mu_{low}(F_{i1})(X_i)$, $\mu_{medium}(F_{i1})(X_i)$ or $\mu_{high}(F_{i1})(X_i)$ is greater than 0.5. This enables a more compact and meaningful representation of each pattern point in terms of its linguistic properties and ensures better handling both during the training and testing phases of the proposed neural network model.

IV. INCORPORATION OF CLASS INFORMATION IN INPUT VECTOR DURING TRAINING

The input to the proposed neural network model consists of two portions. In addition to the linguistic properties discussed in Section III, there is also some contextual information [14] regarding the fuzzy class membership [7] of each pattern used as training data during self-organization of the network.

In the traditional Kohonen's net model [3], [4], the input vector consists of quantitative information only regarding the patterns. Generally the training patterns used during self-organization are also used later for calibrating the output space. This refers to a *hard* labeling of the output neuron by the pattern class corresponding to a training pattern for which it elicits the maximum response. A qualitative measure of the topological ordering of the output space may be obtained from calibration. Note that during self-organization the model clusters the training patterns, whereas during calibration it labels these clusters with some additional class information. So the training phase is completely unsupervised while calibration is not. Then, we could add a testing phase to obtain a hard classification of a set of test data by assigning a membership value of 1 to only that class corresponding to the partition of the neuron (labeled during calibration) eliciting the maximum response.

In many real-life problems, the data are generally ill-defined with overlapping or fuzzy class boundaries. Each pattern used in training may possess finite membership in more than one class. To model such data, it often becomes necessary to incorporate some contextual information regarding class membership as part of the input vector. However during self-organization this part of the input vector is assigned a lower weight so that the linguistic properties dominate in determining the ordering of the output space. During calibration we use

the contextual class membership information part of the input vector (in *crisp* form as in (15)) only for determining the *hard* labeling of the output space. A separate fuzzy partitioning that allows scope for producing overlapping clusters is also introduced. It has been observed that the inclusion of this contextual class membership information produces more efficient self-organization and is necessary in handling fuzzy or imprecise data. This is perhaps because in addition to the associated higher input space dimensionality, some sort of partial supervision is used here instead of the completely unsupervised functioning of the more conventional Kohonen's model.

While the traditional Kohonen's model was used for clustering purposes, the proposed model has been extended to function as a fuzzy classifier, i.e., as a mechanism for assigning input vectors to known output classes. We use partial supervision in the form of assigning a *lower weight* to contextual class membership information during self-organization. We also use a testing phase to evaluate the recognition performance of the *calibrated* neurons on a separate set of test data.

Class Membership as Contextual Information

The pattern \mathbf{X}_i is considered to be presented as a concatenation of the linguistic properties in (6) and the contextual information regarding class membership. Let the input vector be expressed as

$$\mathbf{x} = [\mathbf{x}', \mathbf{x}']^T = [\mathbf{x}', \mathbf{0}^T \cup \mathbf{0}, \mathbf{x}']^T \quad (10)$$

where \mathbf{x}' contains the linguistic information in the $3n$ -dimensional space of (6) and \mathbf{x}'' covers the class membership information in an l -dimensional space for an l -class problem domain. So the input vector \mathbf{x} lies in an $(3n + l)$ -dimensional space. Both \mathbf{x}' and \mathbf{x}'' are expressed as membership values. The representation of \mathbf{x}' has been discussed in Section III. Here we consider the definition of \mathbf{x}'' .

Weighted distance: Let the n -dimensional vectors \mathbf{O}_k and \mathbf{V}_k denote the mean and standard deviation respectively of the training data (used during self-organization) for the k th class. The weighted distance of a training pattern $\mathbf{X}_i = [F_{i1}, F_{i2}, \dots, F_{in}]^T$ from the k th class is defined as

$$z_{ik} = \sqrt{\sum_{j=1}^n \left[\frac{F_{ij} - \theta_{kj}}{v_{kj}} \right]^2} \quad \text{for } k = 1, \dots, l \quad (11)$$

where F_{ij} is the value of the j th component of the i th pattern point \mathbf{X}_i . The weight $\frac{1}{v_{kj}}$ is used to take care of the variance of the classes so that a feature with higher variance has less weight (significance) in characterizing a class. Note that when all the feature values of a class are the same, then the standard deviation will be zero. In that case, we consider $v_{kj} = 1$ such that the weighting coefficient becomes one. This is obvious because any feature occurring with identical magnitudes in all members of a training set is certainly an *important* feature of the set and hence its contribution to the membership function should not be reduced [7], [23].

Membership Function: The membership of the i th pattern to class C_k is defined as [7]

$$\mu_k(\mathbf{X}_i) = \frac{1}{1 - \left(\frac{z_{ik}}{F_d} \right)^{F_e}} \quad (12)$$

where z_{ik} is the weighted distance from (11) and the positive constants F_d and F_e are the denominational and exponential fuzzy generators [7], [24] controlling the amount of fuzziness in this class-membership set. Obviously $\mu_k(\mathbf{X}_i)$ lies in the interval [0,1]. Here (12) is such that the higher the distance of a pattern from a class, the lower is its membership value to that class. It is to be noted that when the distance is zero, the membership value is one (maximum) and when the distance is infinite, the membership value is zero (minimum).

It should be mentioned that as the training data have fuzzy class separation, a pattern point \mathbf{X}_i may correspond to one or more classes in the input feature space. So a pattern point belonging to two classes (say, C_{k_1} and C_{k_2}) corresponds to two *hard* labels in the training data, with \mathbf{X}_i tagged to classes C_{k_1} and C_{k_2} respectively. In other words, there are two or more occurrences of point \mathbf{X}_i in the training set such that sometimes it is tagged to class C_{k_1} and sometimes to class C_{k_2} . In this case \mathbf{X}_i is used in computing \mathbf{O}_{k_1} , \mathbf{O}_{k_2} , \mathbf{V}_{k_1} , and \mathbf{V}_{k_2} only. Here the l -dimensional vector $\mu(\mathbf{X}_i)$ has only two non-zero components corresponding to z_{ik_1} and z_{ik_2} . However in the *hard* case \mathbf{X}_i corresponds to only one *hard* label in the training data, say C_{k_1} , such that \mathbf{X}_i is used in computing \mathbf{O}_{k_1} and \mathbf{V}_{k_1} only. Note that $\mu(\mathbf{X}_i)$ has l non-zero components in the *fuzziest* case and only one non-zero component in the *hard* case.

Fuzzy Modifier: In the *fuzziest* case, we may use the fuzzy modifier INT to enhance *contrast* in class membership [7]. We have

$$\mu_{INT(k)}(\mathbf{X}_i) = \begin{cases} 2[\mu_k(\mathbf{X}_i)]^2 & \text{for } 0 \leq \mu_k(\mathbf{X}_i) \leq 0.5 \\ 1 - 2[1 - \mu_k(\mathbf{X}_i)]^2 & \text{otherwise.} \end{cases} \quad (13)$$

This is needed to increase the contrast within class membership values, i.e., to decrease the ambiguity in making a decision.

Applying the Membership Concept: For the i th pattern we define

$$\mathbf{x}'' = \begin{cases} s * [\mu_{INT(1)}(\mathbf{X}_i), \dots, \mu_{INT(l)}(\mathbf{X}_i)]^T & \text{in the fuzziest case} \\ s * [\mu_1(\mathbf{X}_i), \dots, \mu_l(\mathbf{X}_i)]^T & \text{otherwise} \end{cases} \quad (14)$$

where $0 < s \leq 1$ is the scaling factor. To ensure that the norm of the linguistic part \mathbf{x}' predominates over that of the class membership part \mathbf{x}'' in (10) during self-organization, we choose $s < 0.5$.

Note that unlike the model in [14], we define the part \mathbf{x}'' of the input vector \mathbf{x} in terms of membership functions that attain values in the interval [0,1] and provide a measure of belonging

to the corresponding fuzzy set. During self-organization we allow partial supervision involving $s (< 0.5)$ times the class membership information, such that this knowledge may also be incorporated into the connection weight values. This enables a training pattern with membership, say, 0.9 in class C_{k_1} to be mapped perhaps to a neuron that is not the same as that to which another training pattern with membership, say, 0.5 to class C_{k_1} or, say, 0.5 to class C_{k_2} is mapped.

Modification of Input During Calibration

During calibration of the output space the input vector chosen is $\mathbf{x} = [0, \mathbf{x}']$, where \mathbf{x}' is given by (14) such that

$$\mu_q(\mathbf{X}_i) = \begin{cases} 1 & \text{if } q = k \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

for $k \in \{1, \dots, l\}$ and $s = 1$. The N^2 neuron outputs v_{ki} are calibrated w.r.t. the l classes. Here the class information of the training patterns is given full weight while the input feature information is suppressed. The primary objective of this stage is to label each neuron by the class (partition) for which it elicits the maximum response. The resulting *hard* (labeled) partitioning of the output space may be used to qualitatively assess the topological ordering of the pattern classes w.r.t. the input feature space. Note that while \mathbf{x}' contains class membership information during self-organization, we use binary \mathbf{x}' at the input during calibration. We also introduce a *fuzzy* partitioning of the output space by labeling the output neurons with the fuzzy membership values of their output responses. This helps generate overlapping partitions of the output space which are thereby closer to the input feature space representation in case of fuzzy data. This concept is explained in detail in Section V-C.

Let us consider the following situation. A pattern having class memberships of, say, 0.52 to class C_{k_1} and 0.48 to class C_{k_2} may be mapped to a neuron i (eliciting maximum response) that is calibrated as belonging to the *hard* partition of class C_{k_1} . However we should note that the lower yet significant membership of this pattern to class C_{k_2} ought not be ignored. Herein lies the utility of the *fuzzy* partitioning. By this, the particular neuron i may be calibrated as belonging to both the classes C_{k_1} and C_{k_2} , albeit with different membership values.

It should be noted that the traditional Kohonen's net model uses unsupervised learning during self-organization. During calibration, the training patterns or some reference vectors (in case of known samples) are used for the *hard* labeling of the neurons. This provides some insight into the topological ordering of the output space thus partitioned. In the semantic maps [14], on the other hand, the class information is used in this stage to generate the *hard* labeling of the partitions during calibration. We introduce a separate testing phase where a different set of fuzzy test patterns (kept aside from the original data set while randomly selecting the training set for self-organization) are classified using the input feature information of the test vector along with the above-mentioned fuzzy partitioning information. This procedure is explained in detail in Section V-D.

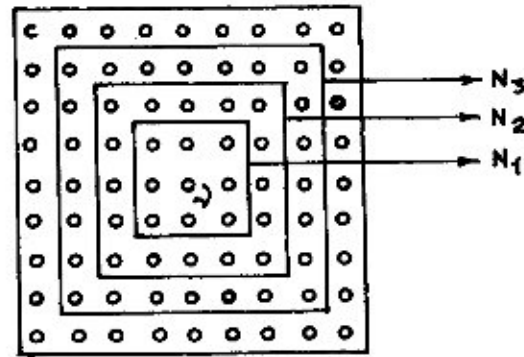


Fig. 4. Topological r -neighborhoods [N_r] as feature maps are formed. The neighborhood starts large and slowly decreases in size over time from $r = 3$ to $r = 1$.

V. FUZZY EXTENSION TO KOHONEN'S ALGORITHM

Consider an $(3n+l)$ -dimensional input space with the input vector $\mathbf{x} = [\mathbf{x}', \mathbf{x}'']^T$ of (10) being incident simultaneously on the $N \times N$ array of neurons.

Concept of r -Neighborhood: Each neuron $\nu(ii, jj)$ has a topological r -neighborhood $N_r(ii, jj)$, as depicted in Fig. 4, where ii, jj denote the row and column numbers respectively of the neuron. We have

$$N_r(ii, jj) = \{\nu(u, v) \mid \max\{|u - ii|, |v - jj|\} = r\} \\ 1 \leq u, v \leq N \quad (16)$$

where $r = 0, 1, \dots, 3$. Note that the indices ii and jj will be omitted in future reference to avoid clutter.

Output of a Neuron: The output of the i th neuron is computed using (1), with the subset S_i of neurons being defined as its r -neighborhood N_r . We choose

$$\sigma(q) = \begin{cases} 0 & \text{if } q < 0 \\ q & \text{otherwise.} \end{cases} \quad (17)$$

This transformation ensures that $\sigma(q) \geq 0$. We also use

$$w_{ki} = \begin{cases} b & \text{for } r = 1 \\ -\frac{b}{2} & \text{for } r = 2 \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Here b is the mutual interaction weight for the lateral coupling w_{ki} .

Weight Updating

Initially the components of the \mathbf{m}_i 's are set to small random values lying in the range $[0, 0.5]$. Let the best match between vectors \mathbf{m}_c and \mathbf{x} , selected using (2), occur at neuron c . Using (3), the weight updating expression may be stated as

$$\mathbf{m}_i(t+1) = \begin{cases} \mathbf{m}_i(t) + h_{ci} \cdot (\mathbf{x}(t) - \mathbf{m}_i(t)) & \text{for } i \in N_r, r = 0, 1, \dots, 3 \\ \mathbf{m}_i(t) & \text{otherwise} \end{cases} \quad (19)$$

where N_r defines a r -neighborhood by (16) around neuron c such that r decreases with time. Here the gain factor h_{ci}

$$msd = \frac{1}{|trainset|} \sum_{\mathbf{x} \in trainset} \left[\sum_{r=0}^3 \left\{ \left(\frac{1}{|N_r|} \sum_{i \in N_r} \|\mathbf{x} - \mathbf{m}_i\|^2 \right) * (1 - r * f) \right\} \right] \quad (21)$$

is considered to be bell-shaped like the π -function, such that $|h_{ci}|$ is the largest when $i = c$ and gradually decreases to zero with increasing distance from c . Besides, $|h_{ci}|$ also decays with time.

Gain Factor: We define

$$h_{ci} = \frac{(1 - r * f)}{1 + \left(\frac{nt}{edenom} \right)^2} \quad (20)$$

where nt is the number of sweeps already made through the entire set of training samples at any point of time, $edenom$ is a positive constant (scaling factor) suitably chosen and $0 < f < 1$. The decay of $|h_{ci}|$ with time is controlled by nt . The slowly decreasing radius of the bell-shaped function h_{ci} and the corresponding change in $|h_{ci}|$ are controlled by the parameters r and f . Due to the process of self-organization, the randomly chosen initial \mathbf{m}_i 's gradually attain new values according to (2), (19) such that the output space acquires appropriate topological ordering.

Index of Disorder

An index of disorder D may be defined to provide a measure of this ordering. Let msd denote the mean square distance between the input vector and the weight vectors in the r -neighborhood of neuron c . We define (21) [see top of page] where $|trainset|$ refers to the number of input pattern vectors in the training set. This definition ensures that neurons nearer c (smaller r) contribute more to msd than those farther away. Also

$$f = \begin{cases} 1/4, & 0 \leq r \leq 3 & \text{for } nent = 1 \\ 1/3, & 0 \leq r \leq 2 & \text{for } nent = 2 \\ 1/2, & 0 \leq r \leq 1 & \text{otherwise.} \end{cases} \quad (22)$$

Here $|N_r|$ denotes the number of neurons in the r -neighborhood of neuron c such that $|N_1| \leq 8$, $|N_2| \leq 16$ and $|N_3| \leq 24$ depending upon the position of c in the two-dimensional array. Note that N_0 implies neuron c itself.

The expression for the index of disorder is given as

$$D = msd(nt - kn) - msd(nt) \quad (23)$$

where $msd(nt)$ denotes the mean square distance by (21) at the end of the nt th sweep through the training set and kn is a suitable positive integer such that D is calculated relative to an interval of kn sweeps. Initially $nent$ is set to 1. Then

$$nent = \begin{cases} nent + 1 & \text{if } D < \delta \\ nent & \text{otherwise} \end{cases} \quad (24)$$

where $0 < \delta < 0.001$. The process is terminated when $nent > 3$, so that in (22) we always have $r \geq 1$. For good self-organization, the value of msd and therefore D should gradually decrease. It should be noted that the r and f parameters of (20) are determined by (22) and thus depend

on the $nent$ parameter; $nent$, in turn, is itself determined by (24) and thus depends on D .

Partitioning During Calibration

During calibration the input vector $\mathbf{x} = [0, \mathbf{x}']$ of (10) is applied to the neural network. Let the $(i1)_k$ th neuron generate the highest output η_{f_k} for class C_k . We define a membership value for the output of neuron i when calibrated for class C_k simply as

$$\mu_k(\eta_i) = \frac{\eta_{ik}}{\eta_{f_k}} \text{ for } i = 1, \dots, N^2, \text{ and } k = 1, \dots, l \quad (25)$$

such that $0 \leq \mu_k(\eta_i) \leq 1$ and $\mu_k(\eta_i) = 1$ for $i = (i1)_k$.

Each neuron i may be marked by the output class C_k , among all l classes, that elicits the maximal response η_{f_k} . This generates a hard partitioning of the output space and is used in the more conventional model [14].

Fuzzy Partitioning: On the other hand, each neuron i has a finite membership $\mu_k(\eta_i)$ to class C_k by (25). We may generate the crisp boundaries for the fuzzy partitioning of the output space by considering for each of the l classes the α -cut set $\{i | \mu_k(\eta_i) > \alpha'\}$, $0 < \alpha' \leq 1$, where α' is a suitably chosen value. This is done solely for the ease of depiction of the various partitions in the output space. Note that the generation of overlapping fuzzy partitions for the fuzzy input data demonstrates the utility of the process.

An ordered and unbroken map of the output space indicates good self-organization and hence grouping of the patterns according to similarity. In cases where the data are fuzzy and overlapping classes exist, the hard partitioning contains apparent disorder and/or discontinuity; the incorporation of the fuzzy membership concept alleviates this problem. The utility of the fuzzy approach may be appreciated by considering a point lying in a region of overlapping classes in the feature space. In such cases its membership to each of these classes may be nearly equal, and to follow the hard approach of calibrating relative only to the neuron for which the point elicits the maximum response is to ignore a significant property of the data.

Testing Phase

After self organization, the proposed model encodes all input data information distributed among its connection weights. The class membership of the training patterns is also learned due to the partial supervision used in that stage. During calibration, the neurons are labeled by the pattern classes and the corresponding membership values are assigned. This is the desired fuzzy classifier. In the final stage, a separate set of test patterns is supplied as input to the neural network model and its performance evaluated.

During this phase input test vectors $\mathbf{x} = [\mathbf{x}', 0]^T$, consisting of only the linguistic information in the $3n$ -dimensional space

defined by (6), is applied to the network. Let the p 1th and p 2th neurons generate the highest and second highest outputs η_{fp} and η_{sp} respectively, for test pattern p . Furthermore, let $\mu_{k_1}(\eta_{fpm})$ and $\mu_{k_2}(\eta_{spm})$ be the highest and second highest output membership values generated during testing, with respect to classes C_{k_1} and C_{k_2} respectively. It is to be noted that $k_1 = k_2$ for both choices for pattern points not lying in regions of overlapping classes and there is no ambiguity of decision in such cases. We define

$$\begin{aligned}\mu_{k_1}(\eta_{fpm}) &= \mu_{k_1}(\eta_{p1}) \\ \mu_{k_2}(\eta_{spm}) &= \frac{1}{\eta_{fp}} \mu_{k_2}(\eta_{p2}) * \eta_{sp},\end{aligned}\quad (26)$$

and $k_1 = k1$, $k_2 = k2$, if $\mu_{k_1}(\eta_{p1}) \geq \frac{1}{\eta_{fp}} \mu_{k_2}(\eta_{p2}) * \eta_{sp}$. Otherwise,

$$\begin{aligned}\mu_{k_1}(\eta_{fpm}) &= \frac{1}{\eta_{fp}} \mu_{k_2}(\eta_{p2}) * \eta_{sp}, \\ \mu_{k_2}(\eta_{spm}) &= \mu_{k_1}(\eta_{p1})\end{aligned}\quad (27)$$

such that $k_1 = k2$ and $k_2 = k1$. Here $k1$ and $k2$ refer to the output classes (hard partitions) C_{k1} and C_{k2} that elicited maximal strength responses at the p 1th and p 2th neurons respectively during calibration. C_{k1} and C_{k2} are dependent both on the actual output responses during testing and the membership values evaluated during calibration w.r.t. classes C_{k1} and C_{k2} . The membership values on the right-hand side of (26), (27) are defined as

$$\mu_{k_1}(\eta_{p1}) = \frac{\eta_{(p1)k_1}}{\eta_{fk_1}} \quad (28)$$

from (25), where η_{fk_1} and $\eta_{(p1)k_1}$ are obtained during calibration for class C_{k_1} . Hence pattern p may be classified as belonging to class C_{k_1} with membership $\mu_{k_1}(\eta_{fpm})$ lying in the interval $[0,1]$, using the first choice and to class C_{k_2} with membership $\mu_{k_2}(\eta_{spm})$ using the second choice. It is to be noted that classes C_{k_1} and C_{k_2} are determined from classes C_{k1} and C_{k2} by (26), (27). A confusion matrix [7] may be generated to evaluate the performance of this fuzzy classifier on the set of test patterns.

It is worth noting that if we consider the calibrated membership values instead of the calibrated strength values on the r.h.s. of (28) for substitution into (26), (27), then we get membership-based recognition instead of the strength-based recognition scheme just described.

Mean Square Distance for Test Set: The mean square distance for test patterns is defined as

$$msd_t = \frac{1}{|testset|} \sum_{p \in testset} \|p - m_{p1}\|^2 * \frac{3n+1}{3n} \quad (29)$$

where $|testset|$ corresponds to the number of pattern vectors used during testing, and m_{p1} consists of the first $3n$ components only of the weight vector of the neuron $p1$ generating the highest output response η_{fp} for test pattern p . This is a measure of the amount of mismatch between the two vectors while classifying pattern p . The factor $\frac{3n+1}{3n}$ is used to make the value of msd_t comparable to that of msd of (21).

VI. IMPLEMENTATION AND RESULTS

The neural network described in the previous sections was tested using a set of 871 Indian Telugu vowel sounds collected by trained personnel [24]. These were uttered in a Consonant-Vowel-Consonant context by three 30-35 year old male speakers. The simulation was in C on a VAX-8650 computer. Figure 5 shows the feature space of six vowel classes (θ, a, i, u, e, o) in the $F_1 - F_2$ plane (for ease of depiction); the actual data set has three features F_1, F_2 , and F_3 corresponding to the first, second, and third vowel formant frequencies obtained through spectrum analysis of the speech data. The dimension of the input vector is 15. Note that the boundaries of the classes in the given data set are seen to be ill-defined, overlapping, and fuzzy.

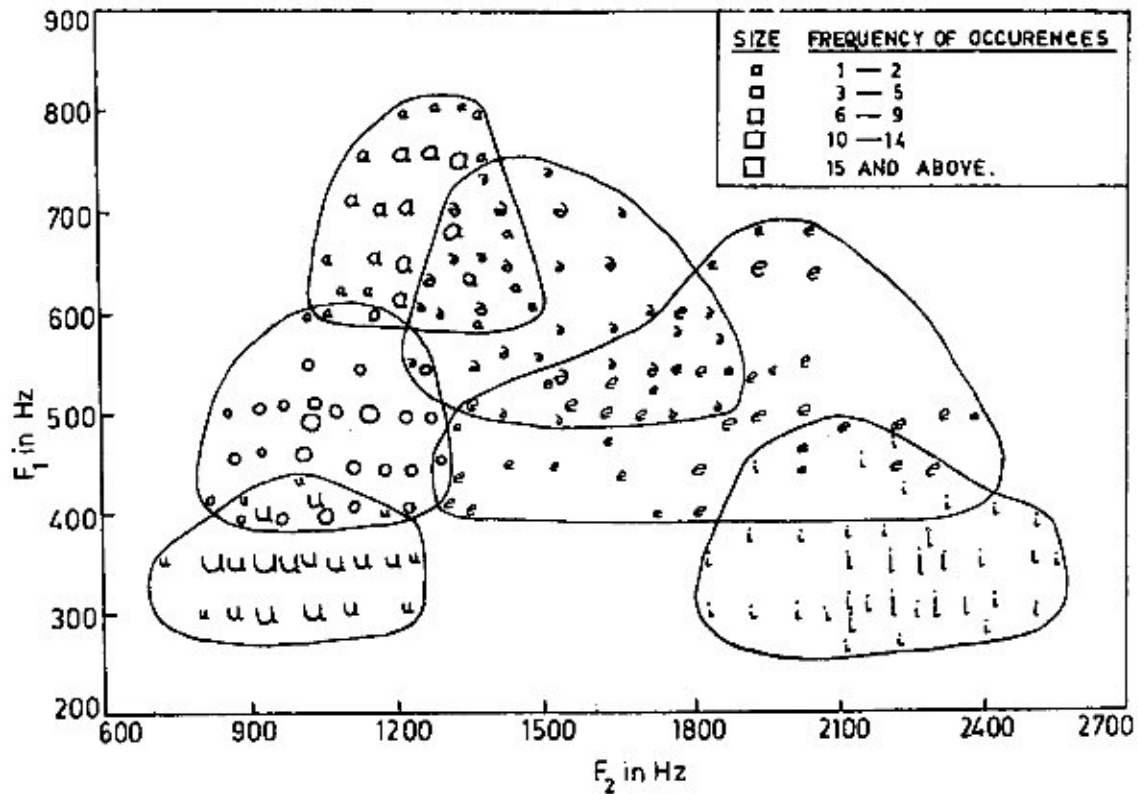
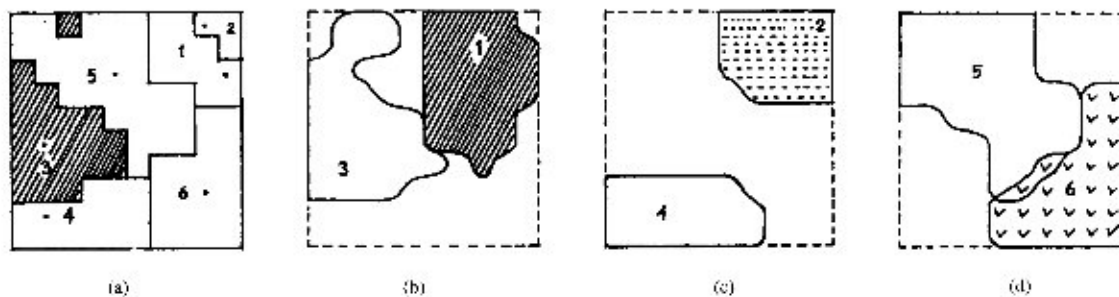
The model has been tested for two-dimensional networks with varying numbers of neurons. During self-organization, different sizes of training sets have been used by randomly choosing *perc* % samples from each representative vowel class. The remaining $(100 - perc)$ % samples from the original data set were used as the test set in each case. We selected $fdenom = 0.8$ in (8)-(9), $t_d = 5$ and $F_s = 1$ in (12), $s = 0.2$ in (14), $b = 0.02$ in (18) and $\delta = 0.0001$ in (24) after several experiments.

Output Map

After self-organization and calibration the resulting output map is plotted using both hard and fuzzy partitioning. In Figs. 6 and 7, (a) corresponds to the hard partitioning obtained by mapping each neuron to the vowel class to which it is most sensitive. The class number k (1 for θ , 2 for a , 3 for i , 4 for u , 5 for e , 6 for o) marks the neuron eliciting the maximum response η_{fk} for that class C_k while the neighboring dot indicates the neuron generating the second highest response. Parts (b)-(d) of the same figures indicate the boundaries for the fuzzy partitioning of the output space by (25) for the three pairs (chosen to render the displays as clear as possible) of the six classes using $\alpha' = 0.1$. It is to be noted that the topological ordering of the vowel classes in the two-dimensional output space (considering fuzzy partitioning) bears much similarity, including the amount of overlapping, to the original Fig. 5 in the two-dimensional feature space. The use of fuzzy partitioning is found to help in faithfully preserving the mapping of fuzzy or overlapping pattern classes.

Figure 6 shows the output map generated for an 10×10 array of neurons with *perc* = 15. The hard partitioning illustrates one discontinuous mapping for class 3. However the incorporation of fuzzy partitioning alleviates this problem and we find overlapping between classes 1,2; 1,5; 2,5; 2,6; 3,5; 4,5; 4,6; and 5,6. This compares favorably with the overlapping observed in the feature space of Fig. 5. It is to be noted that, unlike in Fig. 5, the classes 3 and 4 are seen to be adjacent in (a) here. This is because there exist no pattern points between these two classes in the input feature space and in this sense they may be termed *adjacent*.

Figure 7 shows the output for the conventional Kohonen's net model (using the same parameters as in Fig. 6) with $s = 0$ in (14) but also incorporating the fuzzy partitioning

Fig. 5. Vowel diagram in $F_1 - F_2$ plane.Fig. 6. 10×10 array of neurons with $perc = 15$ and $cdnorm = 100$. (a) Hard partitioning. (b) (c) (d) Fuzzy partitioning of the output space.

concept as extension. The input feature information part x' of (10) is in the fuzzy linguistic form of (6) for ease of comparison with the proposed model while demonstrating the utility of the inclusion of the contextual class membership part x'' in the input vector. Note the discontinuities among the hard partitions for classes 1 and 3. We also observe incorrect topological ordering of the vowel classes (as compared to Fig. 5). In (a), contrary to the desired situation, the partitions for classes 2,5 and 3,6 are adjacent, while classes 2,6 and 4,5 are separated. Furthermore, the neurons eliciting the highest and second highest responses have been observed to lie in the wrong calibrated *hard* partitions for classes 3 and 4. This has an adverse effect on the recognition performance over the

test set by (26)–(28). The use of fuzzy partitioning introduces discontinuities for class 6 in (d) while eliminating the problems for classes 1 and 3 in (b). However classes 1,3 and 2,4 are found to be adjacent in (b) and (c), unlike the case in Fig. 5. A comparison of Figs. 6 and 7 should make apparent the value of incorporating contextual information into the neural network.

Performance on Test Set

As a final step, a separate set of test patterns was applied to the model under consideration and its performance evaluated. In Figs. 8–10, (a) plots the percentage correct classification

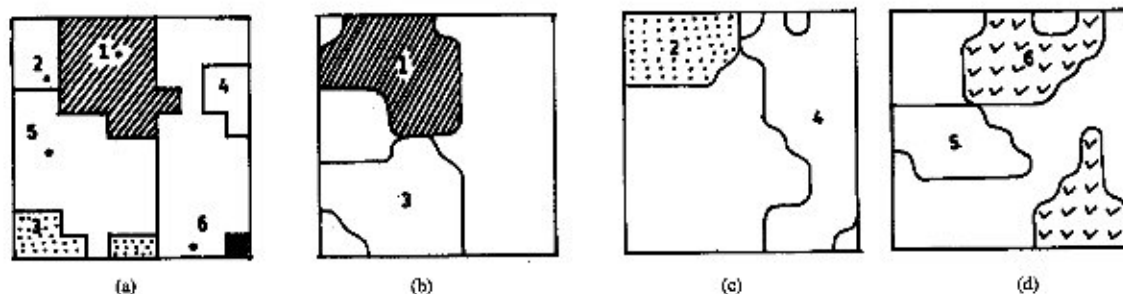


Fig. 7. Conventional Kohonen's model with $perc = 15$ and $cdenom = 100$. (a) Hard partitioning. (b)–(d) Fuzzy partitioning of the 10×10 output space.

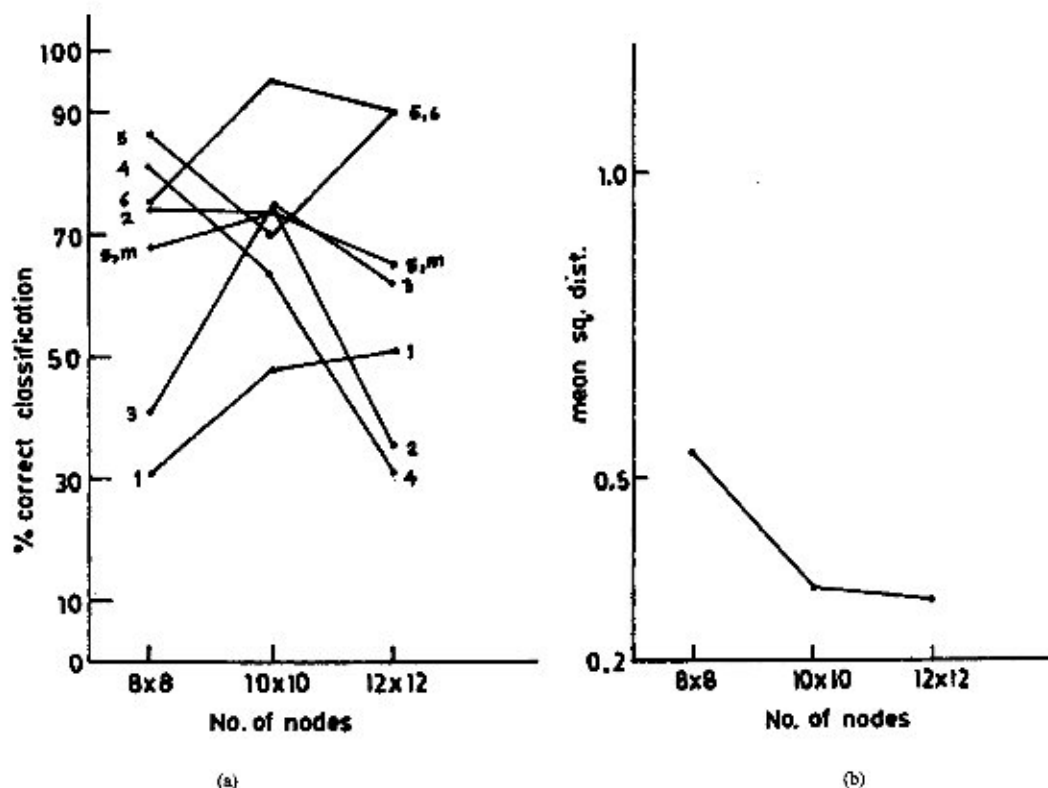


Fig. 8. Neural net model with $perc = 10$ and $cdenom = 100$. (a) Correct classification (percentage) versus size of neural network array. (b) Mean square distance versus size of neural network array, using test patterns.

while (b) shows the variation of the mean square distance msd_t of (29) along the ordinate. In (a), the class numbers ($k = 1, \dots, 6$) indicate the class-wise correct classification of the test set. The variables s and m correspond to the overall correct classification of the entire test set using the strength-based recognition by (26), (27), and the related membership-based recognition schemes, respectively.

Figure 8 illustrates the effect of varying the size of the network. The 10×10 array is observed to give best recognition rates in (a). A smaller size of the network is seen to be incapable of handling all the information required while a larger size may result in poor performance over the test

set. However the msd_t curve in (b) demonstrates that the 8×8 array results in a much poorer topological ordering as compared to the other two network sizes while the 12×12 array yields a *slightly lower* value of msd_t as compared to the 10×10 network.

Figure 9 demonstrates the effect of using the index of disorder D of (21)–(24) to control the number of sweeps through the training samples during self-organization. This is marked as "usual iterations" (i.e., controlled iteration count) on the abscissa of the figure. In the traditional Kohonen's model, the network goes through a larger number of sweeps. The effect of using 200 iterations without considering the

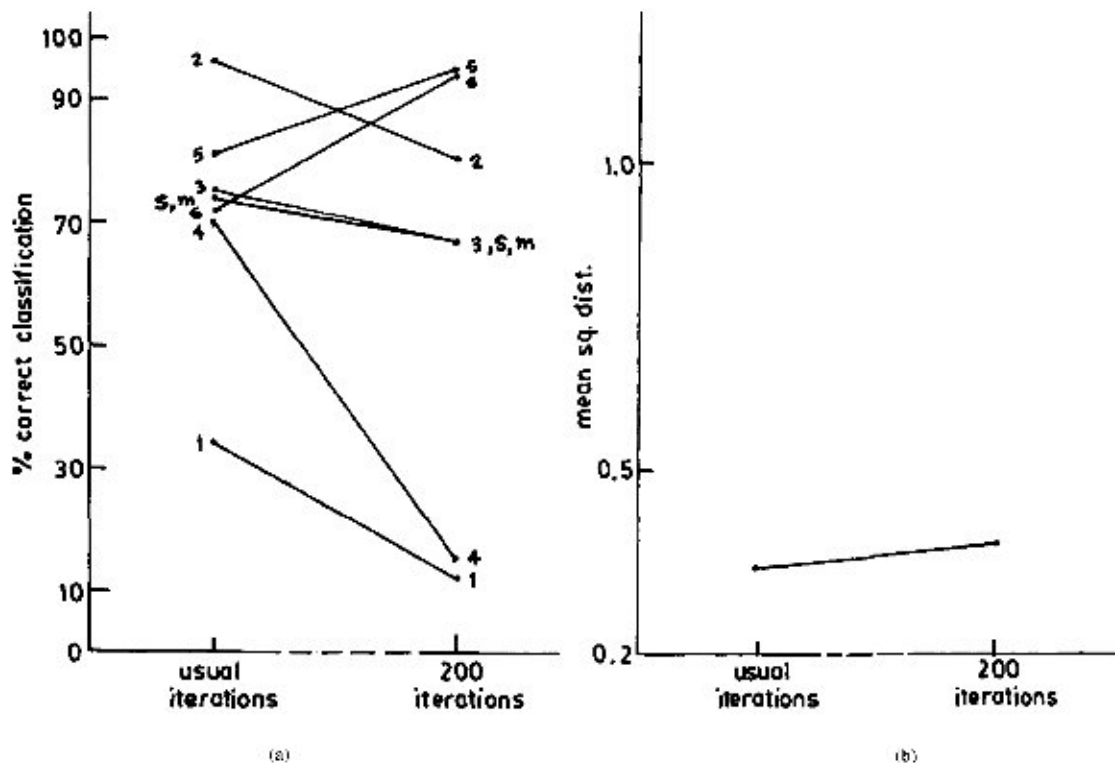


Fig. 9. Comparison between proposed neural net model using index of disorder D marked *usual iterations* and the more conventional model using 200 iterations without D . (a) Correct classification (percentage). (b) Mean square distance, with $perc = 10$ and $ednorm = 100$ for test patterns using 10×10 network.

influence of D is also plotted. The proposed model is found to yield an improved performance (with only 90 iterations) over the more conventional design.

In Fig. 10 we compare between (i) the proposed model (marked "usual" along the abscissa), (ii) the "hard" version using a *crisp* linguistic representation for the input vector with $s > 0$ and (iii) the "original" Kohonen's model with $s = 0$ in (14) but using *fuzzy* linguistic representation for the input vector along with the fuzzy partitioning concept as an extension. The different features of these models are listed in Fig. 11. In the hard model, the input feature information part in the $3n$ -dimensional space is assigned *crisp* values such that corresponding to a pattern X_i , along the j th axis, we clamp the highest of $\mu_{low}(F_{1j})(X_i)$, $\mu_{medium}(F_{1j})(X_i)$, and $\mu_{high}(F_{1j})(X_i)$ of (6) to 1 while the remaining two are kept clamped at 0. The gain factor h_{ci} from (20) is not *bell-shaped* and its *hard* version is defined as $h_{ci} = 1/[1 + (\frac{c_i}{ednorm})^2]$. The contextual class information, though present, is not in the form of graded membership values but is expressed in *crisp* terms giving a membership of 1 to only one class. The original model (method (iii)) is used with the $3n$ -dimensional fuzzy linguistic representation for the input feature information and the *bell-shaped* gain factor h_{ci} of (20).

Note that the hard model is seen to have the worst recognition rate, while the proposed model yields the best overall classification efficiency. Inclusion of fuzzy concepts (as in-

roduced in methods (i) and (iii)) is found to enhance the performance w.r.t. the *hard* version (method (ii)). On the other hand, the incorporation of class information with $s > 0$ enables the proposed model (method (i)) to score over the more conventional *original* version (method (iii)). This underscores the utility of involving fuzzy concepts in conjunction with partial class membership information in the proposed model.

It is observed that the msd_i curve in (b) exhibits better resultant topological ordering for the *hard* version as compared to the *original* model. This is in contrast to the findings for the recognition rate (%) in (a) of the figure where it is seen to have poorer performance. We should note that the *hard* version uses partial supervision ($s > 0$) although with *crisp* input, output and partitioning. This contextual class information generates a better ordering of the output space (along with a lower msd_i value) although the recognition rate is poorer due to the *hard* representation used. However the proposed model has a superior performance w.r.t. both the recognition rate and msd_i , as it incorporates both fuzziness and partial supervision.

Table I compares the recognition score (on test set) of the proposed neural net model to that of the Bayes' classifier [20], [21] and the standard fully supervised fuzzy approach [24]. We have used the Bayes' classifier for multivariate normal patterns with the *a priori* probabilities $p_i = \frac{|C_i|}{N}$ where $|C_i|$ indicates the number of patterns in class C_i and N is the total number of pattern points. The dispersion matrices are different for each

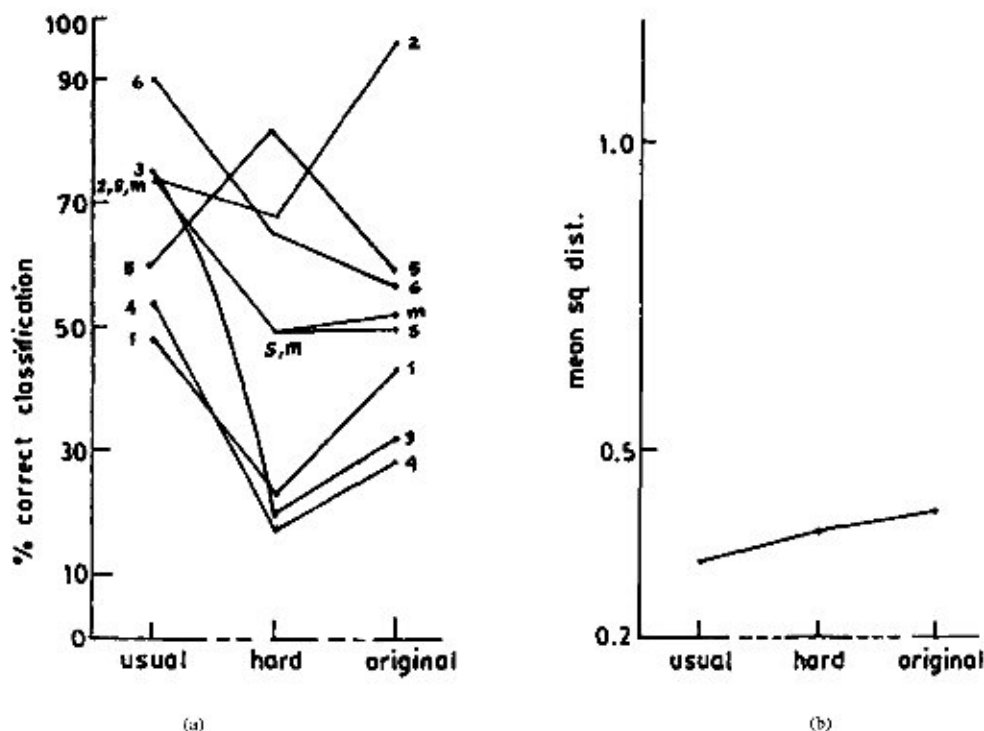


Fig. 10. Comparison between proposed neural net model (marked *usual*), its *hard* version, and the *original* Kohonen's model. (a) Correct classification (percentage). (b) Mean square distance, with $perc = 10$ and $cdenom = 100$ using 10×10 network for test patterns.

Model	Input Feature Information	Class Information Scale Factor	Membership Factor	Gain	Output Partitioning
Proposed (<i>usual</i>)	fuzzy linguistic	$s = 1$ for calibration $0.5 > s > 0$ for self-calibration	fuzzy	bell-shaped	fuzzy
Hard	crisp linguistic	$s = 1$ for calibration $0.5 > s > 0$ for self-calibration	crisp	pulse	fuzzy
Original	fuzzy linguistic	<i>nil</i>	<i>nil</i>	bell-shaped	fuzzy

Fig. 11. The different features of the three models, viz., *proposed (usual)*, the *hard* version, and the *original* Kohonen's network.

pattern class. The overall performance of the proposed model is found to be quite satisfactory. It is to be noted that the Bayes' classifier is the best that is theoretically possible and neural nets should not do better. A good statistical classifier, however, requires a lot of sequential computation and a large number of reference vectors. The value of the proposed approach resides in the fact that a neural network is massively parallel and can generalize well with a smaller set of training patterns.

As a rule, test patterns are misclassified by the network only into one of the neighboring classes in the vowel triangle (Fig. 5). The correct classification rate for an 10×10 network considering both the first and second choices by (26)–(27) is illustrated in Table II. The confusion matrix for this particular set of parameters, as shown in Table III, also supports this claim.

TABLE I
COMPARISON OF RECOGNITION SCORE (%) BETWEEN BAYES' CLASSIFIER, STANDARD SUPERVISED FUZZY CLASSIFIER, AND THE PROPOSED NEURAL NET MODEL WITH $perc = 10$. NEURAL NETWORK IS OF SIZE 10×10 WITH $cdenom = 20$

Class	Bayes' Classifier	Standard Fuzzy Classifier	Proposed Neural Model
\emptyset	44.6	51.4	23.0
<i>a</i>	83.9	81.7	97.5
<i>i</i>	81.9	78.0	74.8
<i>u</i>	88.9	67.6	73.5
<i>e</i>	82.8	77.7	88.7
<i>o</i>	77.7	78.8	92.6
Overall	79.6	73.4	79.6

TABLE II
RECOGNITION SCORE (%) WITH $cdenom = 60$ AND $perc = 10$ FOR A 10×10 NETWORK

Class	First choice	Second choice	Net score
\emptyset	53.8	7.7	61.5
<i>a</i>	76.5	21.0	97.5
<i>i</i>	79.3	3.9	83.2
<i>u</i>	66.9	13.2	80.1
<i>e</i>	64.7	23.0	87.7
<i>o</i>	90.1	1.9	92.0
Overall	73.5	11.2	84.7

In Table IV we compare the performance of the proposed model for various choices of the parameters r and f used in the computation of the gain factor h_{oi} of (20). Model *A*

TABLE III
CONFUSION MATRIX WITH $cdnorm = 60$
AND $perc = 10$ FOR A 10×10 NETWORK

	θ	α	i	u	e	o
θ	35	16	0	0	13	1
α	18	62	0	0	0	1
i	0	0	123	0	32	0
u	4	0	0	91	3	38
e	57	0	7	1	121	1
o	4	9	0	0	3	146

TABLE IV
COMPARISON OF RECOGNITION SCORE (%) FOR VARIOUS CHOICES OF
PARAMETERS r AND f IN THE GAIN FUNCTION h_{ci} FOR A 10×10
NEURAL NET MODEL WITH $cdnorm = 100$ AND $perc = 10$

Class	Model			
	A	B	C	D
θ	7.7	0.0	1.5	47.7
α	51.8	98.7	100.0	74.0
i	69.0	89.6	68.3	74.8
u	24.2	80.8	66.9	63.9
e	94.6	73.8	74.8	70.0
o	88.8	63.3	58.0	94.4
Overall	64.6	72.5	65.2	73.5

refers to the case where $f = \frac{1}{4}$ in (22) for all values of $ncnt$. Although both $|h_{ci}|$ and r decay with time, this constitutes a slight variation of the proposed neural net model D (due to the constant value of f). Network B uses $0 < r \leq 3$ and $f = \frac{1}{4}$ for all values of $ncnt$ in (22). Note that here only $|h_{ci}|$ decays with time by (20) while its radius remains constant. In model C the term $(1 - r * f)$ is eliminated from the numerator of (20) and the radius ($0 < r \leq 3$) of the gain function is kept constant (as in B). Here the function h_{ci} is no longer bell-shaped and only $|h_{ci}|$ decays with time. The significance of the proposed gain factor in model D , where both $|h_{ci}|$ and r decay with time, is obvious from the results.

Table V illustrates a comparison in the performance on the test set (using first choice) of the proposed model with the more conventional Kohonen's model (with fuzzy linguistic feature information x' of (6), (10) only at the input) for various sizes of training data set $perc$. This is to demonstrate the necessity of incorporating the contextual class membership information x'' into the input of the proposed network for modeling fuzzy data. We observe that the proposed model has a superior recognition score compared to its more conventional counterpart. Note that an increase in the size of the training set (abundance of attribute data) for the vectors under analysis has no appreciable impact on the performance of the conventional model. On the other hand, the incorporation of the contextual class membership information, with $s > 0$, seems to boost the efficiency of the proposed model (with identical parameter values) in classifying the same fuzzy data. This further demonstrates the utility of using class membership information in the input vector.

In Table VI we demonstrate the effect (on the recognition efficiency) of using various numbers of input attributes (dimensions) on the standard Kohonen's net (with fuzzy lin-

TABLE V
COMPARISON OF RECOGNITION SCORE (%) BETWEEN PROPOSED NEURAL NET
MODEL AND THE CONVENTIONAL KOHONEN'S NET FOR VARIOUS SIZES OF
TRAINING SET $perc$ USING 10×10 NETWORK ARRAY WITH $cdnorm = 100$

Model	Conventional Kohonen's Net					Proposed Neural Net					
	$perc$	10	20	30	40	50	10	20	30	40	50
θ	43.0	44.8	39.2	77.3	80.5	47.7	43.1	43.1	61.3	61.3	47.2
α	96.3	65.2	90.4	40.7	0.0	74.0	59.7	47.6	0.0	37.7	37.7
i	31.6	26.8	53.7	36.5	34.8	74.8	78.9	57.0	78.8	62.8	62.8
u	27.9	44.6	0.0	39.5	31.5	63.9	72.7	49.0	39.5	48.6	48.6
e	59.3	83.7	76.5	79.2	77.8	70.0	74.1	91.0	88.0	98.0	98.0
o	56.8	79.8	57.9	47.2	92.2	94.4	62.5	95.2	95.3	93.3	93.3
Overall	50.3	59.8	53.2	53.2	56.5	73.5	68.3	69.4	68.0	71.1	71.1

TABLE VI
COMPARISON OF RECOGNITION SCORE (%) BETWEEN PROPOSED
NEURAL NET MODEL AND THE CONVENTIONAL KOHONEN'S
NET FOR VARIOUS NUMBER OF INPUT ATTRIBUTES USING
 10×10 NETWORK ARRAY WITH $cdnorm = 100$ AND $perc$

Model	Conventional Kohonen's Net			Proposed Neural Net
	Input Vector	fuzzy linguistic features with $s = 0$	fuzzy linguistic features with $0.5 > s > 0$	
Components	9	18	27	15
Dimension	9	18	27	15
θ	43.0	52.3	100.0	47.7
α	96.3	0.0	0.0	74.0
i	31.6	69.0	51.6	74.8
u	27.9	71.3	0.0	63.9
e	59.3	75.9	0.0	70.0
o	56.8	97.5	0.0	94.4
Overall	50.3	68.4	18.4	73.5

guistic input feature information as an extension) and compare with the proposed model using contextual class membership information at the input. A very high input feature space dimensionality with too many attributes is found to hinder the efficiency of the conventional network. Partitioning the primary linguistic properties among *low*, *medium*, and *high* yields nine attributes for the given data set. Incorporation of the hedge *very* (for each of the three linguistic terms) yields 18 attributes while further addition of the hedge *more or less* leads to 27 attributes for the conventional model. The latter version is seen to be incapable of classifying the given pattern set. Note that the incorporation of the contextual class membership information (with $s > 0$) in the proposed model results in the best performance, both overall and class-wise.

VII. CONCLUSIONS AND DISCUSSION

A neural network model based on self organization and capable of performing fuzzy classification was presented. Basically, the Kohonen clustering network is used here as a semantic map. The algorithm passed through two stages, viz. self-organization and testing. The model had the flexibility of accepting linguistic input and could provide output decision in terms of membership values. The input vector incorporated partial class membership information during self-organization. An index of disorder was used to determine a measure of the ordering of the output space and control the number of sweeps required in the process. Unlike Kohonen's conventional model, the proposed net was capable of producing fuzzy

partitioning of the output space and could thereby provide a more faithful representation for ill-defined or fuzzy data with overlapping classes. Incorporation of fuzziness in the input and output of the proposed model was seen to result in better performance as compared to the original Kohonen's model and the hard version. The problem of vowel recognition was used to demonstrate the effectiveness of the proposed model for various network array sizes, training sets and gain factors.

It should be noted that only three linguistic properties *low*, *medium*, and *high* were used here. Incorporation of additional input feature information in the form of fuzzy hedges like *more or less*, *very*, *nearly*, etc., may improve the performance of the proposed model, due to the resulting more detailed input description, but then the cost of nodes and interconnections would also increase.

Representation of input in terms of π -sets *low*, *medium*, and *high* also enables the system to accept imprecise/vague features F_j in various forms, namely, F_j is about 500, F_j is between 400 and 500, F_j is low, medium, very low, more or less low or F_j is missing etc. In these cases F_j needs to be transformed into 3-dimensional vector consisting of membership values corresponding to the primary properties *low*, *medium*, and *high*. A convenient heuristic method for the determination of these membership values may be found in [19].

Neural net performance in fuzzy classification of the speech data was found to compare favorably with that of the Bayes' classifier trained on the same data. In the model described here, massively parallel interconnection links with simple processing elements (neurons) permit the computational complexity of standard statistical techniques to be avoided. Therefore with the necessary parallel hardware backing the proposed model should be able to perform much faster and hence more efficiently.

It has been observed that a *critical* size of the network was required for satisfactory performance. The fact that a larger size resulted in poorer recognition of the test patterns was favorable in the sense that more neurons would lead to an increased cost.

ACKNOWLEDGMENT

The authors gratefully acknowledge the referee for his elaborate constructive criticism, Prof. D. Dutta Majumder for his interest in this work, Mr. S. Chakraborty for drawing the diagrams and Mr. A. Ghosh for typing the final version. Ms. S. Mitra is grateful to the CSIR for providing her financial assistance in the form of a fellowship. A part of the work was completed while Prof. S. K. Pal held an NRC-NASA Senior Research Award at the Johnson Space Center, Houston.

REFERENCES

- [1] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoustics, Speech and Signal Processing*, vol. 61, pp. 4-22, 1987.
- [2] D. E. Rumelhart and J. L. McClelland, eds., *Parallel Distributed Processing*, vol. 1. Cambridge, MA: MIT, 1986.
- [3] T. Kohonen, "An introduction to neural computing," *Neural Networks*, vol. 1, pp. 3-16, 1988.
- [4] T. Kohonen, *Self-Organization and Associative Memory*. Berlin: Springer-Verlag, 1989.
- [5] J. C. Bezdek and S. K. Pal, eds., *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*. NY: IEEE Press, 1992.
- [6] 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.
- [7] S. K. Pal and D. Dutta Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*. New York: Wiley (Halsted Press), 1986.
- [8] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- [9] G. J. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information*. Reading, MA: Addison Wesley, 1989.
- [10] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [11] T. L. Huntberger and P. Ajjimarangsee, "Parallel self-organizing feature maps for unsupervised pattern recognition," *Int. J. General Syst.*, vol. 16, pp. 357-372, 1989.
- [12] J. C. Bezdek, E. C. Tsao, and N. R. Pal, "Fuzzy Kohonen clustering networks," in *Proc. 1st IEEE Conf. on Fuzzy Systems*, San Diego, pp. 1035-1043, 1992.
- [13] W. Pedrycz and H. C. Carl, "Linguistic interpretation of self-organizing maps," in *Proc. 1st IEEE Conf. on Fuzzy Systems*, San Diego, pp. 371-378, 1992.
- [14] H. Ritter and T. Kohonen, "Self-organizing semantic maps," *Biological Cybern.*, vol. 61, pp. 241-254, 1989.
- [15] T. Kohonen, "The neural phonetic typewriter," *IEEE Computer*, pp. 11-22, Mar. 1988.
- [16] H. Ritter and K. Schulten, "On the stationary state of Kohonen's self-organizing sensory mapping," *Biological Cybern.*, vol. 54, pp. 99-106, 1986.
- [17] T. Kohonen, "Analysis of a simple self-organizing process," *Biological Cybern.*, vol. 44, pp. 135-140, 1982.
- [18] S. P. Luttrell, "Image compression using a multilayer neural network," *Pattern Recog. Lett.*, vol. 10, pp. 1-7, 1989.
- [19] S. K. Pal and D. P. Mandal, "Linguistic recognition system based on approximate reasoning," *Information Sci.*, vol. 61, pp. 135-161, 1992.
- [20] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
- [21] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. London: Addison-Wesley, 1974.
- [22] S. K. Pal and P. K. Pramanik, "Fuzzy measures in determining seed points in clustering," *Pattern Recog. Lett.*, vol. 4, pp. 159-164, 1986.
- [23] G. S. Sebestyen, *Decision Making Processes in Pattern Recognition*. NY: Macmillan, 1962.
- [24] S. K. Pal and D. Dutta Majumder, "Fuzzy sets and decision making approaches in vowel and speaker recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, pp. 625-629, 1977.



Sushmita Mitra (S'91) received the B.Sc.(Hons.) degree in physics and the B.Tech and M. Tech. degrees in computer science from Calcutta University in 1984, 1987, and 1989, respectively.

She was a Senior Research Fellow of the Council for Scientific and Industrial Research from 1989 to 1991. She is a programmer in the Electronics and Communications Sciences Unit of the Indian Statistical Institute, Calcutta. Her research interests include pattern recognition, fuzzy sets, artificial intelligence, and neural networks. She is a student member of the INNS. Currently she is at ELITE (European Laboratory for Intelligent Techniques Engineering), Aachen, Germany, on a German Academic Exchange Service Fellowship.



Sankar K. Pal (M'81 SM'84 F'93) received the B.Sc.(Hons.) degree in physics and the B.Tech., M.Tech., and Ph.D. degrees in radiophysics and electronics, from the University of Calcutta, in 1969, 1972, 1974, and 1979, respectively. In 1982 he received the Ph.D. degree in electrical engineering along with DIC from Imperial College, University of London.

He is a Professor in the Electronics and Communication Sciences Unit at the Indian Statistical Institute, Calcutta. In 1986 he was awarded a Fulbright Postdoctoral Visiting Fellowship to work at the University of California, Berkeley, and the University of Maryland, College Park. In 1989 he received an NRC-NASA Senior Research Award to work at the NASA Johnson Space Center, Houston, TX. He received the 1990 Shanti Swarup Bhatnagar Prize in Engineering Sciences ("the most coveted and highest award to a scientist in India") for his contribution in pattern recognition. He served as a Professor-in-Charge of the Physical and Earth Sciences Division, Indian Statistical Institute, during 1988-1990. He was also a Guest Lecturer in computer science at Calcutta University from 1983-1986. His research interests mainly include pattern recognition, image processing, artificial intelligence, neural nets, and fuzzy sets and systems. He is a co-author of the book *Fuzzy Mathematical Approach to Pattern Recognition*, which received the Best Production Award in the 7th World Book Fair, New Delhi, and a co-editor of the book *Fuzzy Models for Pattern Recognition*. He has more than one hundred fifty research papers—including ten in edited books and more than ninety in international journals—to his credit. He has also lectured, on his research, at different U.S. and Japanese universities and laboratories. He is listed in *Reference Asia: Asia's Who's Who of Men and Women of Achievements*.

Dr. Pal is a Fellow of both the IEEE and the IETE. He is a member of the Editorial Boards of *IEEE Transactions on Fuzzy Systems*, *International Journal of Approximate Reasoning*, and the *Far East Journal of Mathematical Sciences*. He is a member of the Reviewing Board for *IEEE Computer* and *Mathematical Reviews* magazines, and an Executive Committee member of the ISFUMIP and IUPRAI. He is also a Permanent Member of the Indo-US Forum for Cooperative Research and Technology Transfer (IFCRTT).