Unsupervised feature extraction using neuro-fuzzy approach

Rajat K. De[a], Jayanta Basak[b, 1], Sankar K. Pal[a, *]

[a] *Machine Intelligence Unit, Indian Statistical Institute, 203, Barrackpore Trunk Road, Calcutta 700035, India*
[b] *IBM India Research Lab, Block-I, Indian Institute of Technology, Hauz Khas, New Delhi 110016, India*

Received 14 October 1998; received in revised form 15 November 2000; accepted 16 January 2001

Abstract

The present article demonstrates a way of formulating a neuro-fuzzy approach for feature extraction under unsupervised training. A fuzzy feature evaluation index for a set of features is newly defined in terms of degree of similarity between two patterns in both the original and transformed feature spaces. A concept of flexible membership function incorporating weighted distance is introduced for computing membership values in the transformed space that is obtained by a set of linear transformation on the original space. A layered network is designed for performing the task of minimization of the evaluation index through unsupervised learning process. This extracts a set of optimum transformed features, by projecting n-dimensional original space directly to n'-dimensional ($n' < n$) transformed space, along with their relative importance. The extracted features are found to provide better classification performance than the original ones for different real life data with dimensions 3, 4, 9, 18 and 34. The superiority of the method over principal component analysis network, nonlinear discriminant analysis network and Kohonen self-organizing feature map is also established.

1. Introduction

Feature selection or extraction is a process of selecting a map of the form $\mathbf{x}' = f(\mathbf{x})$ by which a sample $\mathbf{x}(x_1, x_2, \ldots, x_n)$ in an n-dimensional measurement space ($\Re^n$) is transformed into a point $\mathbf{x}'(x'_1, x'_2, \ldots, x'_{n'})$ in an n'-dimensional ($n' < n$) feature space ($\Re^{n'}$). The problem of feature selection deals with choosing some of x_i's from the measurement space to constitute the feature space. On the other hand, the problem of feature extraction deals with generating new x'_j's (constituting the feature space) based on some x_i's in the measurement space. The main objective of these processes is to retain the optimum salient characteristics necessary for the recognition process and to reduce the dimensionality of the measurement space so that effective and easily computable algorithms can be devised for efficient categorization. The present article concerns with feature extraction.

Different useful classical techniques for feature extraction are available in [3,6]. Some of the recent attempts made for this task in the framework of ANN are mainly based on multilayer feedforward networks [5,11,12,18,19] and self-organizing networks [12,10,9]. The methods based on multilayer feedforward networks include, among others, development

[*] Corresponding author. Tel.: +91-33-577-8085; fax: +91-33-577-3035.

E-mail addresses: sankar@isical.ac.in (S.K. Pal), jayantabasak2000@hotmail.com (J. Basak), rajat@isical.ac.in (R.K. De).

[1] The work was done when the author was with Machine Intelligence Unit, Indian Statistical Institute, Calcutta 700035, India.

of principal component analysis (PCA) network [17,5,1], nonlinear discriminant analysis network [22], Sammon's projection, linear discriminant analysis (LDA) network [12], whereas those based on self-organizing networks include development of nonlinear projection based Kohonen's self-organizing feature map (SOM) [12,8], distortion tolerant Gabor transformations followed by minimum distortion clustering by multilayer self-organizing maps [10] and a nonlinear projection method based on Kohonen's topology preserving maps [9]. Note that, depending on whether the class information of the samples are known or not, these methods are classified under supervised or unsupervised mode. For example, the algorithms described in [11,18,22] fall under supervised category whereas those in [10,17,8] are in unsupervised mode.

Recently, attempts are being made to integrate the merits of fuzzy set theory and ANN under the heading 'neuro-fuzzy computing' with an aim of making the systems artificially more intelligent. Incorporation of fuzzy set theory enables one to deal with uncertainties in different tasks of pattern recognition system, arising from deficiency (e.g., vagueness, incompleteness, etc.) in information, in an efficient manner. ANNs, having the capability of fault tolerance, adaptivity and generalization, and scope for massive parallelism, are widely used in dealing with learning and optimization tasks. In the area of pattern recognition, neuro-fuzzy approaches have been attempted mostly for designing classification/clustering methodologies. The problem of feature selection/extraction, particularly the later task, has not been addressed much in neuro-fuzzy framework.

The present article is an attempt in this regard and provides a neuro-fuzzy approach for feature extraction under unsupervised training. The methodology involves connectionist minimization of a fuzzy feature evaluation index. The feature evaluation index is defined based on the membership functions denoting the degrees of similarity between two patterns in both the original and transformed feature spaces. The lower the value of the index, the higher is the importance of the transformed features in characterizing/discriminating various clusters. The transformed space is obtained through a set of linear transformations. Computation of the membership values in the transformed space involves a set of weighting coefficients which provides flexibility in modeling various clusters and reflects the degree of individual importance of the transformed features. A layered network is designed for performing the task of minimization of the said index through unsupervised learning process; thereby extracting the optimum transformed space along with the weighting coefficients. This is described in Fig. 1. The algorithm considers interdependence of the original features. The architecture of the network is such that the number of nodes in its second hidden layer determines the desired number of extracted features.

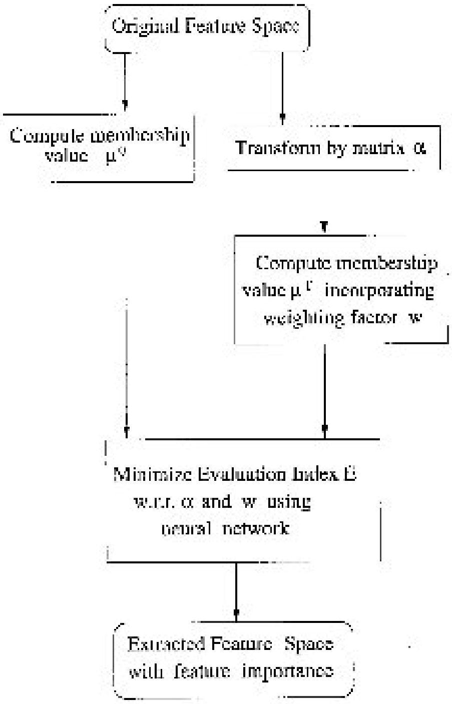
Original Feature Space
Compute membership value μ^O
Transform by matrix α
Compute membership value μ^T incorporating weighting factor w
Minimize Evaluation Index E w.r.t. α and w using neural network
Extracted Feature Space with feature importance

Fig. 1. Schematic description of the neuro-fuzzy method for feature extraction.

The effectiveness of the algorithm is demonstrated on five different real-life data sets, namely, Iris [4], vowel [16,15], medical [7,13], mango-leaf [14] and an ionospheric data [20]. The superior discrimination ability of the extracted features over the original ones is established using k-NN classifier for different values of k. The algorithm is also compared with both supervised and unsupervised methods including nonlinear discriminant analysis network (NDAN) [22], principal component analysis network (PCAN) [17] and Kohonen self-organizing feature map (SOM) [8].

2. Feature evaluation index

In this section we first of all provide a definition of the fuzzy feature evaluation index. The membership function for its realization is then defined in terms of distance measure and weighting coefficients.

2.1. Definition

Let, μ_{pq}^{O} be the degree that both the pth and qth patterns belong to the same cluster in the n-dimensional original feature space, and μ_{pq}^{T} be that in the n'-dimensional ($n' < n$) transformed feature space. μ values determine how similar a pair of patterns are in the respective features spaces. That is, μ may be interpreted as the membership value of a pair of patterns belonging to the fuzzy set "similar". Let, s be the number of samples on which the feature evaluation index is computed.

The feature evaluation index for a set (Ω) of features is defined as

$$E = \frac{2}{s(s-1)} \sum_{p} \sum_{q \neq p} \frac{1}{2} [\mu_{pq}^{\mathrm{T}}(1 - \mu_{pq}^{\mathrm{O}}) + \mu_{pq}^{\mathrm{O}}(1 - \mu_{pq}^{\mathrm{T}})]. \tag{1}$$

It has the following characteristics.

(i) If $\mu_{pq}^{\mathrm{O}} = \mu_{pq}^{\mathrm{T}} = 0$ or 1, the contribution of the pair of patterns to the evaluation index E is zero (minimum).

(ii) If $\mu_{pq}^{\mathrm{O}} = \mu_{pq}^{\mathrm{T}} = 0.5$, the contribution of the pair of patterns to E becomes 0.25 (maximum).

(iii) For $\mu_{pq}^{\mathrm{O}} < 0.5$ as $\mu_{pq}^{\mathrm{T}} \to 0$, E decreases.
For $\mu_{pq}^{\mathrm{O}} > 0.5$ as $\mu_{pq}^{\mathrm{T}} \to 1$, E decreases.

Therefore, the feature evaluation index decreases as the membership value representing the degree of belonging of pth and qth patterns to the same cluster in the transformed feature space tends to either 0 (when $\mu^{\mathrm{O}} < 0.5$) or 1 (when $\mu^{\mathrm{O}} > 0.5$). In other words, the feature evaluation index decreases as the decision on the similarity between a pair of patterns (i.e., whether they lie in the same cluster or not) becomes more and more crisp. This means, if the intercluster/intracluster distances in the transformed space increase/decrease, the feature evaluation index of the corresponding set of features decreases. Therefore, our objective is to extract those features for which the evaluation index becomes minimum; thereby optimizing the decision on the similarity of a pair of patterns with respect to their belonging to a cluster.

2.2. Computation of membership function

In order to satisfy the characteristics of E (Eq. (1)), as stated in the previous section, the membership function (μ) in a feature space may be defined as

$$\mu_{pq} = 1 - \frac{d_{pq}}{D} \qquad \text{if } d_{pq} \leqslant D,$$
$$= 0, \qquad \text{otherwise.} \tag{2}$$

Here d_{pq} is a distance measure which provides similarity (in terms of proximity) between the pth and qth patterns in the feature space. Note that, the higher the value of d_{pq}, the lower is the similarity between pth and qth patterns, and vice versa. D is a parameter which reflects the minimum separation between a pair of patterns belonging to two different clusters. When $d_{pq} = 0$ and $d_{pq} = D$, we have $\mu_{pq} = 1$ and 0, respectively. If $d_{pq} = D/2$, $\mu_{pq} = 0.5$. That is, when the similarity between the patterns is just in between 0 and D, the difficulty in making a decision, whether both the patterns are in the same cluster or not, becomes maximum; thereby making the situation most ambiguous.

The term D (in Eq. (2)) may be expressed as

$$D = \beta d_{\max}, \tag{3}$$

where $d_{\max}$ is the maximum separation between a pair of patterns in the entire feature space, and $0 < \beta \leqslant 1$ is a user defined constant. β determines the degree of flattening of the membership function (Eq. (2)). The higher the value of β, more will be the degree, and vice versa.

The distance d_{pq} (Eq. (2)) can be defined in many ways. Let this, for example, be the Euclidian distance. Then,

$$d_{pq} = \left[\sum_{i} (x_{pi} - x_{qi})^2 \right]^{1/2}, \tag{4}$$

where x_{pi} and x_{qi} are values of ith feature (in the corresponding feature space) of pth and qth patterns, respectively. $d_{\max}$ is defined as

$$d_{\max} = \left[\sum_{i} (x_{\max\, i} - x_{\min\, i})^2 \right]^{1/2}, \tag{5}$$

where $x_{\max i}$ and $x_{\min i}$ are the maximum and minimum values of the ith feature in the corresponding feature space.

2.2.1. Incorporating weighting coefficients

In the above discussion, we have measured the similarity between two patterns in terms of proximity, as conveyed by the expression for d_{pq} (Eq. (4)). Since, d_{pq} is an Euclidian distance, the methodology implicitly assumes that the clusters are hyperspherical. But in practice, this may not necessarily be the case. To model the practical situation we have introduced the concept of weighted distance such that

$$d_{pq} = \left[\sum_i w_i^2 (x_{pi} - x_{qi})^2\right]^{1/2} = \left[\sum_i w_i^2 \chi_i^2\right]^{1/2}, \quad \chi_i = (x_{pi} - x_{qi}), \tag{6}$$

where $w_i \in [0, 1]$ represents weighting coefficient corresponding to ith feature.

The membership value μ_{pq} is now obtained by Eqs. (2), (5) and (6), and becomes dependent on w_i. The values of w_i (<1) make the μ_{pq} function of Eq. (2) flattened along the axis of d_{pq}. The lower the value of w_i, the higher is extent of flattening. In the extreme case, when $w_i = 0$, $\forall i$, $d_{pq} = 0$ and $\mu_{pq} = 1$ for all pair of patterns, i.e., all the patterns lie on the same point making them indiscriminable.

The weight w_i (in Eq. (6)) reflects the relative importance of the feature x_i in measuring the similarity (in terms of distance) of a pair of patterns. The higher the value of w_i, the more is the importance of x_i in characterizing a cluster or discriminating various clusters. $w_i = 1\ (0)$ indicates most (least) importance of x_i.

Note that, one may define μ_{pq} in a different way satisfying the above mentioned characteristics. The computation of μ_{pq} in Eq. (2) does not require class information of the patterns, i.e., the algorithm is unsupervised. It is also to be noted that, the algorithm does not explicitly provide clustering of the feature space.

3. Feature extraction

In the process of feature extraction, the input feature space ($\mathbf{x}$) is transformed to $\mathbf{x}'$ by a matrix α ($= [\alpha_{ji}]_{n' \times n}$), i.e.,

$$\mathbf{x} \xrightarrow{\alpha} \mathbf{x}'.$$

The jth transformed feature is therefore,

$$x'_j = \sum_i \alpha_{ji} x_i, \tag{7}$$

where α_{ji} ($j = 1, 2, \ldots, n'$, $i = 1, 2, \ldots, n$ and $n > n'$) is a set of coefficients. The membership values (μ) are computed using Eq. (2) based on the derived feature values. The distance d_{pq} between pth and qth patterns in the transformed space is, therefore,

$$\begin{aligned} d_{pq} &= \left[\sum_j w_j^2 \left(\sum_i \alpha_{ji}(x_{pi} - x_{qi})\right)^2\right]^{1/2} \\ &= \left[\sum_j w_j^2 \left(\sum_i \alpha_{ji}\chi_i\right)^2\right]^{1/2}, \quad \chi_i = x_{pi} - x_{qi}, \\ &= \left[\sum_j w_j^2 \psi_j^2\right]^{1/2}, \quad \psi_j = \sum_i \alpha_{ji}(x_{pi} - x_{qi}) \end{aligned} \tag{8}$$

and the maximum distance $d_{\max}$ as

$$\begin{aligned} d_{\max} &= \left[\sum_j \left(\sum_i |\alpha_{ji}|(x_{\max i} - x_{\min i})\right)^2\right]^{1/2} \\ &= \left[\sum_j \phi_j^2\right]^{1/2}, \ \phi_j = \sum_i |\alpha_{ji}|(x_{\max i} - x_{\min i}). \end{aligned} \tag{9}$$

Weighting coefficients (w_j) representing the importance of the transformed features, make the shape of clusters in the transformed space hyperellipsoidal instead of hyperspherical.

The membership μ^{T} is computed using d_{pq} and $d_{\max}$ (Eqs. (2), (8) and (9)), while μ^{O} is done by Eqs. (2)–(5). The problem of feature extraction therefore reduces to finding a set of α_{ji} and w_j for which E (Eq. (1)) becomes a minimum. This is schematically explained in Fig. 1. The task of minimization has been performed under unsupervised mode by gradient-descent technique in a connectionist framework. This is described below.

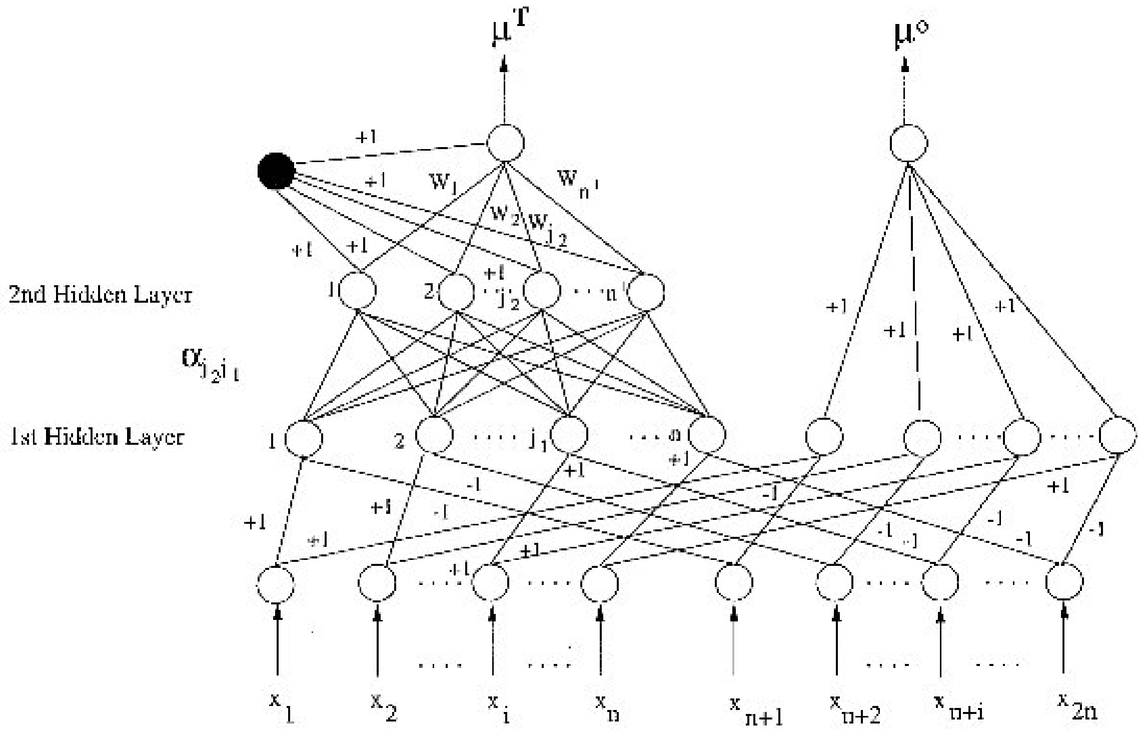

Fig. 2. A schematic diagram of the proposed neural network model.

3.1. Connectionist model

The network (Fig. 2) consists of an input, two hidden and an output layers. The input layer consists of a pair of nodes corresponding to each feature. The first hidden layer consists of $2n$ (for n-dimensional original feature space) number of nodes. Each of the first n nodes computes the part χ_i of Eq. (8) and the rest compute χ_i^2. The value of $(x_{\max i} - x_{\min i})$ is stored in each of the first n nodes. The number of nodes in the second hidden layer is taken as n', in order to extract n' number of features. Each of these nodes has two parts; one of which computes ψ_j^2 of Eq. (8) and the other ϕ_j^2 of Eq. (9). The output layer consists of two nodes which compute μ^{T} and μ^{O} values. There is a node (represented by black circle) in between the output node computing μ^{T}-values and the second hidden layer. This node computes $d_{\max}$ (Eq. (9)) in the transformed feature space and sends it to the output node for computing μ^{T}. The value of β is stored in both the output nodes. The feature evaluation index E (Eq. (20)) is computed from these μ-values off the network.

Input nodes receive activations corresponding to feature values of each pair of patterns. A j_1th node in the first hidden layer is connected to an ith $(1 \leqslant i \leqslant n)$ input node via connection weight +1, and to the $(i+n)$th $(1 \leqslant i \leqslant n)$ input node via connection weight -1. A j_2th node in the second hidden layer is connected to a j_1th node in the first hidden layer via connection weight $\alpha_{j_2 j_1}$. The output node computing μ^{T}-values is connected to a j_2th node in the second hidden layer via connection weight W_{j_2} $(= w_{j_2}^2)$, and that computing μ^{O}-values is connected to a j_1th $(n+1 \leqslant j_1 \leqslant 2n)$ node in the first hidden layer via connection weights +1 each. The node represented by the black circle is connected via weights +1 with the second hidden layer and also with the output node computing μ^{T}-values.

During training, each pair of patterns are presented to the input layer and the evaluation index is computed. The weights $\alpha_{j_2 j_1}$ and W_{j_2}'s are updated using gradient-descent technique in order to minimize the index E. When pth and qth patterns are presented to the input layer, the activation produced by ith $(1 \leqslant i \leqslant 2n)$ input node is

$$v_i^{(0)} = u_i^{(0)}, \tag{10}$$

where

$$\begin{aligned} u_i^{(0)} &= x_{pi} \quad \text{for } 1 \leqslant i \leqslant n \quad \text{and} \\ u_{(i+n)}^{(0)} &= x_{qi} \quad \text{for } 1 \leqslant i \leqslant n. \end{aligned} \tag{11}$$

$u_i^{(0)}$ $(1 \leqslant i \leqslant 2n)$ is the total activation received by an ith input node. The total activation received by j_1th node in the first hidden layer (connecting ith and $(i+n)$th input nodes) is given by

$$u_{j_1}^{(1)} = 1 \times v_i^{(0)} + (-1) \times v_{i+n}^{(0)} \quad \text{for } 1 \leqslant i \leqslant n, \tag{12}$$

and the activation produced by it is

$$\begin{aligned} v_{j_1}^{(1)} &= (u_{j_1}^{(1)}) \quad \text{for } 1 \leqslant j_1 \leqslant n, \\ &= (u_{j_1}^{(1)})^2 \quad \text{for } n+1 \leqslant j_1 \leqslant 2n. \end{aligned} \tag{13}$$

The total activation received by j_2th node in the second hidden layer is given by

$$u_{j_2}^{(2)} = \sum_{j_1} \alpha_{j_2 j_1} v_{j_1}^{(1)}. \tag{14}$$

The activation produced by j_2th node in the second hidden layer is given by

$$v_{j_2}^{(2)} = (u_{j_2}^{(2)})^2. \tag{15}$$

The total activation received by the output node which computes μ^{T}-values is

$$u_{\mathrm{T}}^{(3)} = \sum_{j_2} W_{j_2} v_{j_2}^{(2)}, \tag{16}$$

and that received by the other output node computing μ^{O}-values is

$$u_{\mathrm{O}}^{(3)} = \sum_{j_2} v_{j_2}^{(2)}. \tag{17}$$

Therefore, $u_{\mathrm{T}}^{(3)}$ and $u_{\mathrm{O}}^{(3)}$ represent d_{pq}^2 as given by Eqs. (8) and (4), respectively. The activations, $v_{\mathrm{T}}^{(3)}$ and $v_{\mathrm{O}}^{(3)}$, of the output nodes represent μ_{pq}^{T} and μ_{pq}^{O} for pth and qth pattern pair, respectively. Thus,

$$v_{\mathrm{T}}^{(3)} = 1 - \frac{(u_{\mathrm{T}}^{(3)})^{1/2}}{D} \tag{18}$$

and

$$v_{\mathrm{O}}^{(3)} = 1 - \frac{(u_{\mathrm{O}}^{(3)})^{1/2}}{D}. \tag{19}$$

The evaluation index, in terms of these activations, can then be expressed as (from Eq. (1))

$$E(\alpha, \mathbf{W}) = \frac{2}{s(s-1)} \sum_p \sum_{q \neq p} \frac{1}{2} [v_{\mathrm{T}}^{(3)}(1 - v_{\mathrm{O}}^{(3)}) + v_{\mathrm{O}}^{(3)}(1 - v_{\mathrm{T}}^{(3)})]. \tag{20}$$

The task of minimization of $E(\alpha, \mathbf{W})$ (Eq. (20)) with respect to $\alpha_{j_2 j_1}$ and W_{j_2}, for all j_1 and j_2 is performed using simple gradient-descent technique where the changes in $\alpha_{j_2 j_1}$ $(\Delta\alpha_{j_2 j_1})$ and W_{j_2} (ΔW_{j_2}) are computed as

$$\Delta\alpha_{j_2 j_1} = -\eta_1 \frac{\partial E}{\partial \alpha_{j_2 j_1}} \quad \forall j_1, j_2 \tag{21}$$

and

$$\Delta W_{j_2} = -\eta_2 \frac{\partial E}{\partial W_{j_2}} \quad \forall j_2, \tag{22}$$

where η_1 and η_2 are the learning rates.

For computation of $\partial E/\partial \alpha_{j_2 j_1}$ and $\partial E/\partial w_{j_2}$, the following expressions are used.

$$\frac{\partial E}{\partial \alpha_{j_2 j_1}} = \frac{2}{s(s-1)} \sum_p \sum_{q \neq p} \frac{1}{2} [1 - 2v_{\mathrm{O}}^{(3)}] \frac{\partial v_{\mathrm{T}}^{(3)}}{\partial \alpha_{j_2 j_1}}, \tag{23}$$

$$\frac{\partial v_{\mathrm{T}}^{(3)}}{\partial \alpha_{j_2 j_1}} = -\frac{D\frac{1}{2}(u_{\mathrm{T}}^{(3)})^{-1/2} \partial u_{\mathrm{T}}^{(3)}/\partial \alpha_{j_2 j_1} - (u_{\mathrm{T}}^{(3)})^{1/2} \partial D/\partial \alpha_{j_2 j_1}}{D^2}, \tag{24}$$

$$\frac{\partial u_{\mathrm{T}}^{(3)}}{\partial \alpha_{j_2 j_1}} = W_{j_2} \frac{\partial v_{j_2}^{(2)}}{\partial \alpha_{j_2 j_1}}, \tag{25}$$

$$\frac{\partial v_{j_2}^{(2)}}{\partial \alpha_{j_2 j_1}} = 2u_{j_2}^{(2)} \frac{\partial u_{j_2}^{(2)}}{\partial \alpha_{j_2 j_1}}, \tag{26}$$

$$\frac{\partial u_{j_2}^{(2)}}{\partial \alpha_{j_2 j_1}} = v_{j_1}^{(1)}, \tag{27}$$

$$\frac{\partial D}{\partial \alpha_{j_2 j_1}} = \frac{\beta}{d_{\max}} \left(\sum_i |\alpha_{j_2 i}|(x_{\max i} - x_{\min i}) \right) (x_{\max j_1} - x_{\min j_1}), \tag{28}$$

$$\frac{\partial E}{\partial W_{j_2}} = \frac{2}{s(s-1)} \sum_p \sum_{q \neq p} \frac{1}{2}[1 - 2v_{\mathrm{O}}^{(3)}] \frac{\partial v_{\mathrm{T}}^{(3)}}{\partial W_{j_2}}, \tag{29}$$

$$\frac{\partial v_{\mathrm{T}}^{(3)}}{\partial W_{j_2}} = -\frac{\frac{1}{2}(u_{\mathrm{T}}^{(3)})^{-1/2} \partial u_{\mathrm{T}}^{(3)} / \partial W_{j_2}}{D} \tag{30}$$

and

$$\frac{\partial u_{\mathrm{T}}^{(3)}}{\partial W_{j_2}} = v_{j_2}. \tag{31}$$

3.2. Algorithm for learning α and $\mathbf{W}$

- Calculate $d_{\max}$ (Eq. (5)) from the unlabeled training set and store it in the output node computing μ^{O} values. Store β (user specified) in both the output nodes.
- Initialize $\alpha_{j_2 j_1}$ and W_{j_2} with small random values in $[0, 1]$.
- Repeat until convergence, i.e., until the value of E becomes less than or equal to certain predefined small quantity, or number of iterations attains certain predefined number of iterations:
    - ○ For each pair of patterns:
        - – Present the pattern pair to the input layer.
        - – Compute $\Delta\alpha_{j_2 j_1}$ and ΔW_{j_2} for each j_1 and j_2, using the updating rules in Eqs. (21) and (22).
    - ○ Update $\alpha_{j_2 j_1}$ and W_{j_2} for each j_1 and j_2 with the average values of $\Delta\alpha_{j_2 j_1}$ and ΔW_{j_2}.

After convergence, $E(\alpha, \mathbf{W})$ attains a local minimum. Then the extracted features are obtained by Eq. (7) using the optimum α-values. The weights of the links connecting the output node computing μ^{T}-values, to the nodes in the second hidden layer indicate the order of importance of the extracted features.

4. Results

Here we demonstrate the effectiveness of the abovementioned algorithm on five data sets, namely, Iris [4], vowel [16,15], medical [7,13], mango-leaf [14] and an ionospheric data [20].

Anderson's Iris data [4] set contains three classes, i.e., three varieties of Iris flowers, namely, *Iris setosa*, *Iris versicolor* and *Iris virginica* consisting of 50 samples each. Each sample has four features, namely, sepal length (SL), sepal width (SW), petal length (PL) and petal width (PW). Iris data has been used in many research investigations related to pattern recognition and has become a sort of benchmark-data.

The vowel data consists of a set of 871 Indian Telugu vowel sounds collected by trained personnel. 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, 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. Fig. 3 shows a 2-D projection of the 3-D feature space of the six overlapping vowel classes (∂, a, i, u, e, o) in the F_1–F_2 plane (for ease of depiction). The details of the data and its extraction procedure are available in [16]. This vowel data is being extensively used for more than two decades in the area of pattern recognition.

The medical data consisting of 9 input features and 4 pattern classes, deals with various *Hepatobiliary disorders* [7,13] of 536 patient cases. The input features are the results of different biochemical tests, viz., glutamic oxalacetic transaminate (GOT, Karmen unit), glutamic pyruvic transaminase (GPT, Karmen Unit), lactate dehydrase (LDH, iu/l), gamma glutamyl transpeptidase (GGT, mu/ml), blood urea nitrogen (BUN, mg/dl), mean corpuscular volume of red blood cell (MCV, fl), mean corpuscular hemoglobin (MCH, pg), total bilirubin ($TBil$, mg/dl) and creatinine ($CRTNN$, mg/dl). The hepatobiliary disorders alcoholic liver damage (ALD), primary hepatoma (PH), liver cirrhosis (LC) and cholelithiasis (C), constitute the four output classes.

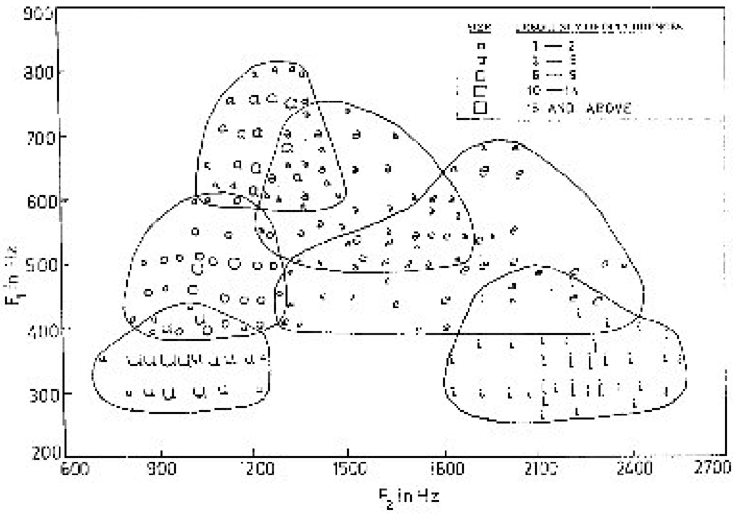

Fig. 3. Two dimensional (F_1–F_2) plot of the vowel data.

Mango-leaf data [14], on the other hand, provides information on different kinds of mango leaf with 18 features, (i.e., 18-dimensional data) for 166 patterns. It has three classes representing three kinds of mango. The feature set consists of measurements like Z-value (Z), area (A), perimeter (Pe), maximum length (L), maximum breadth (B), petiole (P), K-value (K), S-value (S), shape index (SI), $L + P$, L/P, L/B, $(L + P)/B$, A/L, A/B, A/Pe, upper midrib/lower midrib (UM/LM) and perimeter upper half/perimeter lower half (UPe/LPe). The terms 'upper' and 'lower' are used with respect to maximum breadth position.

The ionospheric data was collected by a system in Goose Bay, Labrador [20]. The system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kW. The targets were free electrons in the ionosphere. The data set consists of 351 instances. Each data point has 34 features and may be either "good" or "bad". "Good" data points are those which show evidence of some type of structure in the ionosphere. On the other hand, "bad" points do not show such structure; their signals pass through the ionosphere. The signals received by the radar were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Each instance in this database is described by two attributes per pulse number corresponding to the complex values returned by the function resulting from the complex electromagnetic signal; thereby resulting in 34 ($= 17 \times 2$) features for an instance.

As mentioned in Section 3, the number of nodes in the second hidden layer determines the desired number of extracted features. That is, in order to extract n' number of features, one needs to employ exactly n' nodes in the second hidden layer. For each data set, we performed experiments for different number of nodes in the second hidden layer for finding different sets of extracted features. The particular set for which E-value is minimum in a fixed number of iterations is considered to be the best set of extracted features.

Let us consider the case of Iris data. Table 1 shows the values of α_{ji} (in Eq. (7)) for different sets of extracted features along with their E-values. The extracted features are obtained by Eq. (7). Note that, the set containing two extracted features results in minimum E-value, and therefore, is considered to be the best of all. The expressions for these two extracted features are then written, from Eq. (7), as

$$I_1 = 0.040649SL - 0.000405SW + 0.168035PL + 0.164546PW$$

Table 1
α-values corresponding to different sets of extracted features with their E-values for Iris data

Extracted feature set containing	Coefficients (α) of				E (Eq. (1))
	SL	SW	PL	PW	
One feature	0.071854	−0.028614	0.195049	0.139982	0.102437
Two features	0.040649	−0.000405	0.168035	0.164546	0.099286
	−0.118670	−0.000103	−0.012020	−0.123748	
Three features	−0.017140	0.005148	−0.123089	−0.152892	
	−0.003976	−0.024542	−0.005904	−0.084350	0.104762
	0.023984	−0.004368	0.237469	0.199510	

Table 2
α-values corresponding to the best set of extracted features with their w-values for vowel data

Extracted features	Coefficients (α) of			w	Rank
	F_1	F_2	F_3		
V_1	−0.005676	0.050687	0.000573	0.710050	2
V_2	0.000755	−0.159839	0.000934	0.737597	1

Table 3
α-values corresponding to the best set of extracted features with their w-values for medical data

Extracted features	Coefficients (α) of									w	Rank
	GOT	GPT	LDH	GGT	BUN	MCV	MCH	$TBil$	$CRTNN$		
H_1	−0.193	−0.020	−0.155	−0.059	−0.081	0.096	0.135	0.193	−0.096	0.705	4
H_2	−0.046	0.097	0.035	−0.045	0.070	0.082	0.042	0.088	−0.136	0.711	1
H_3	−0.163	0.102	−0.122	−0.124	−0.155	−0.106	−0.110	0.101	−0.004	0.703	6
H_4	0.123	−0.170	−0.028	−0.107	0.142	0.043	−0.194	0.162	0.035	0.706	3
H_5	0.142	0.173	0.132	0.073	−0.045	−0.177	−0.188	−0.032	−0.030	0.705	4
H_6	−0.208	−0.003	0.083	0.102	0.013	−0.030	0.132	0.032	−0.081	0.707	2
H_7	−0.160	0.116	−0.163	0.082	−0.146	0.094	0.052	−0.142	−0.078	0.704	5
H_8	0.137	0.002	0.125	0.047	−0.078	−0.047	−0.164	0.125	0.053	0.707	2

and

$$I_2 = -0.118670SL - 0.000103SW - 0.012020PL - 0.123748PW.$$

w-values representing the importance of the features I_1 and I_2 are found to be 0.992983 and 0.744317 respectively.

Similarly, the dimension of the best extracted feature space is found to be 2 for vowel data, 8 for both medical and mango-leaf data, and 10 for the ionospheric data. Tables 2–4 show α and w-values for the best extracted feature sets corresponding to vowel, medical and mango-leaf data. (In order to restrict the size of the article, we have not included the table for the ionospheric data.) Note that, in these experiments the values of β are found to be 0.33, 0.16, 0.25, 0.33 and 0.5 for Iris, vowel, medical, mango-leaf and the ionospheric data, respectively.

In order to demonstrate the effectiveness of the feature extraction method, we have compared the discriminating capability of the extracted features with that of the original ones, using k-NN classifier for $k = 1,\ 3$ and 5. For Iris and vowel data, Tables 5 and 6 demonstrate the percentage classification using the extracted feature set and all possible subsets of the

Table 4
α-values corresponding to the best set of extracted features with their w-values for mango-leaf data

Extracted features	Coefficients (α) of																		w	Rank
	Z	A	Pe	L	B	P	K	S	SI	$(L+P)$	L/P	L/B	$(L+P)/B$	A/L	A/B	A/Pe	UM/LM	UPe/LPe		
M_1	−0.171	−0.001	−0.168	−0.075	−0.079	0.095	0.135	0.192	−0.094	−0.039	0.010	0.001	0.001	0.003	0.047	−0.001	−0.003	0.002	0.710	5
M_2	−0.147	0.085	−0.133	−0.168	−0.155	−0.109	−0.111	0.100	−0.005	0.102	−0.097	−0.001	−0.070	0.024	0.004	−0.134	0.027	−0.002	0.713	4
M_3	0.126	0.141	0.126	0.100	−0.047	−0.177	−0.188	−0.031	−0.031	−0.200	0.001	0.002	0.040	0.001	0.001	0.069	−0.002	−0.001	0.708	6
M_4	−0.127	0.099	−0.200	0.098	−0.148	0.093	0.052	−0.142	−0.078	0.124	−0.001	0.021	0.001	0.003	−0.002	−0.086	−0.004	−0.002	0.716	1
M_5	−0.158	−0.065	−0.070	0.073	−0.011	−0.126	0.185	−0.170	−0.021	−0.081	−0.114	−0.001	−0.138	−0.030	0.112	−0.000	0.002	0.048	0.708	6
M_6	−0.047	0.106	0.119	0.110	−0.153	−0.164	−0.122	0.142	0.179	0.129	0.001	−0.002	−0.020	0.001	−0.127	0.042	−0.003	−0.003	0.707	7
M_7	0.084	0.116	−0.223	−0.079	0.024	−0.001	0.145	0.103	−0.096	0.096	0.002	−0.030	0.001	−0.002	0.079	−0.025	−0.003	0.003	0.714	3
M_8	−0.149	0.085	0.234	−0.089	0.166	−0.067	0.174	−0.068	0.031	−0.151	−0.020	−0.027	0.024	−0.001	0.057	−0.035	0.003	0.003	0.715	2

original feature set. In the case of Iris data, the recognition score using the extracted feature set is found to be greater than or equal to that obtained using any set of the original features, except for one case (e.g., the set $\{SL, SW, PL, PW\}$ with $k = 5$). Similar is the case with the vowel data, where the extracted feature pair performs better than any other set of original features, except the set $\{F_1, F_2, F_3\}$.

For medical, mango-leaf and the ionospheric data, comparison is made only between the extracted feature set and the entire original feature set (Tables 7–9). Tables 8 and 9 show that the classification performance in the 8 and 10-dimensional extracted feature space of mango-leaf and the ionospheric data are better than those of the 18 and 34-dimensional original feature space for all values of k. Similar finding is obtained in the case of medical data, except for $k = 1$ (Table 7).

In a part of the experiment, the neuro-fuzzy method for feature extraction is compared with the well-known principal component analysis (PCA) and nonlinear discriminant analysis (NDA) in connectionist framework, called principal component analysis network (PCAN) [17] and nonlinear discriminant analysis network (NDAN) [22], respectively. (For the convenience of readers, PCAN and NDAN are described briefly in Appendices A and B, respectively.) The method is also compared with Kohonen self-organizing feature map (SOM) [8]. For all these cases, we provide the comparative results, using k-NN classifier and scatter plots, on Iris data only. As far as classification ability is concerned, the neuro-fuzzy method has extracted much stronger features than both PCAN and SOM, but slightly weaker features than NDAN (Tables 10 and 5). Note that unlike the proposed method, PCAN and SOM, NDAN is supervised.

Scatter plots in Figs. 4–7 show the class structures in the two-dimensional extracted planes obtained by the proposed neuro-fuzzy method, PCAN, NDAN and SOM, respectively. From these figures, it is observed that the extracted plane (Fig. 4) obtained by the proposed neuro-fuzzy method is much better than those of others (Figs. 5–7) in terms of cluster separability. Note that, an array of 100×100 nodes was considered in the output layer of SOM.

In order to compare the said class structures of the extracted planes (Figs. 4–7) with that of the

Table 5
Recognition score with k-NN classifier for different feature sets of Iris data

Data set	Feature set	% classification		
		$k=1$	$k=3$	$k=5$
Original	$\{SL\}$	48.67	66.67	67.33
	$\{SW\}$	55.33	52.67	52.67
	$\{PL\}$	93.33	95.33	95.33
	$\{PW\}$	89.33	96.00	96.00
	$\{SL, SW\}$	74.67	76.67	76.00
	$\{SL, PL\}$	95.33	93.33	95.33
	$\{SL, PW\}$	94.67	94.00	94.00
	$\{SW, PL\}$	94.67	92.00	93.33
	$\{SW, PW\}$	90.67	94.00	94.67
	$\{PL, PW\}$	93.33	96.00	96.00
	$\{SL, SW, PL\}$	94.00	94.00	94.00
	$\{SL, SW, PW\}$	93.33	93.33	92.00
	$\{SL, PL, PW\}$	96.00	96.67	96.00
	$\{SW, PL, PW\}$	94.00	96.67	95.33
	$\{SL, SW, PL, PW\}$	95.33	96.00	96.67
Extracted	$\{I_1, I_2\}$	96.00	96.67	96.00

Table 6
Recognition score with k-NN classifier for different feature sets of vowel data

Data set	Feature set	% classification		
		$k=1$	$k=3$	$k=5$
Original	$\{F_1\}$	26.52	27.21	27.21
	$\{F_2\}$	38.58	38.23	47.76
	$\{F_3\}$	26.06	33.41	33.87
	$\{F_1, F_2\}$	56.37	68.20	76.35
	$\{F_1, F_3\}$	44.32	46.84	55.80
	$\{F_2, F_3\}$	58.21	63.03	63.95
	$\{F_1, F_2, F_3\}$	78.42	81.29	82.43
Extracted	$\{V_1, V_2\}$	74.63	75.78	76.35

Table 7
Recognition score with k-NN classifier for extracted (obtained by the neuro-fuzzy feature extraction) and original feature sets of medical data

Feature set	% classification		
	$k=1$	$k=3$	$k=5$
Extracted	53.92	56.34	59.89
Original	55.22	56.16	59.14

Table 8
Recognition score with k-NN classifier for extracted (obtained by the neuro-fuzzy feature extraction) and original feature sets of mango-leaf data

Feature set	% classification		
	$k=1$	$k=3$	$k=5$
Extracted	85.71	88.10	92.86
Original	71.69	68.67	70.48

Table 9
Recognition score with k-NN classifier for extracted (obtained by the neuro-fuzzy feature extraction) and original feature sets of the ionospheric data

Feature set	% classification		
	$k=1$	$k=3$	$k=5$
Extracted	85.23	85.80	85.23
Original	84.66	84.66	82.95

original feature space, we provide the scatter plot for PL–PW in Fig. 8. (Note that $\{PL, PW\}$ is known to be the best feature pair [21,2] for Iris data.) The extracted feature plane I_1–I_2 (Fig. 4) is seen to have more resemblance with that in Fig. 8, as compared to Figs. 5–7.

Table 10
Recognition score with k-NN classifier for various extracted feature sets of Iris data

Feature set obtained by	% classification		
	$k=1$	$k=3$	$k=5$
PCAN	92.00	92.00	92.00
NDAN	98.67	97.33	96.00
SOM	66.67	68.00	72.00

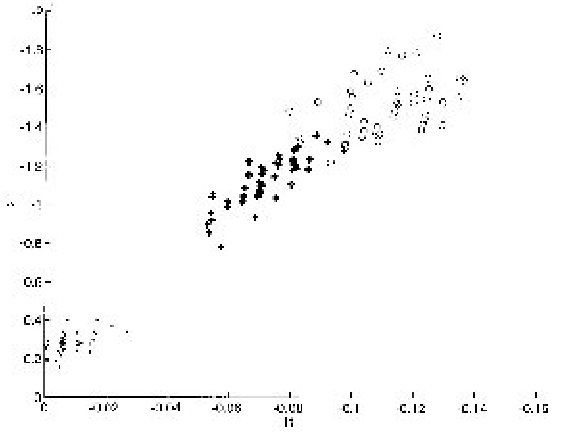

Fig. 4. Scatter plot I_1–I_2, in the extracted plane obtained by the neuro-fuzzy method, of Iris data. Here '.', '+' and '∘' represent classes Iris Setosa, Iris Versicolor and Iris Virginica, respectively.

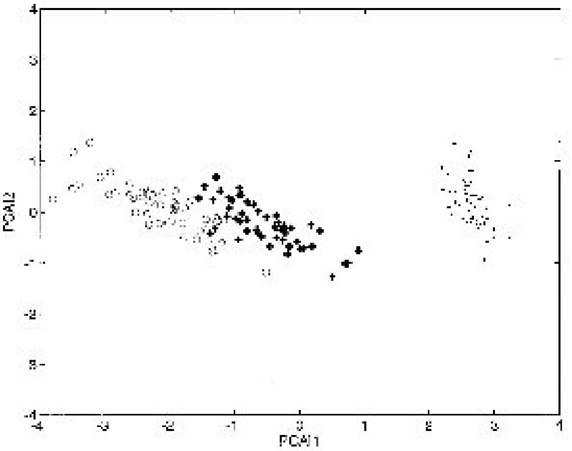

Fig. 5. Scatter plot PCA_1–PCA_2, in the extracted plane obtained by PCAN, of Iris data. Here '.', '+' and '∘' represent classes Iris Setosa, Iris Versicolor and Iris Virginica, respectively.

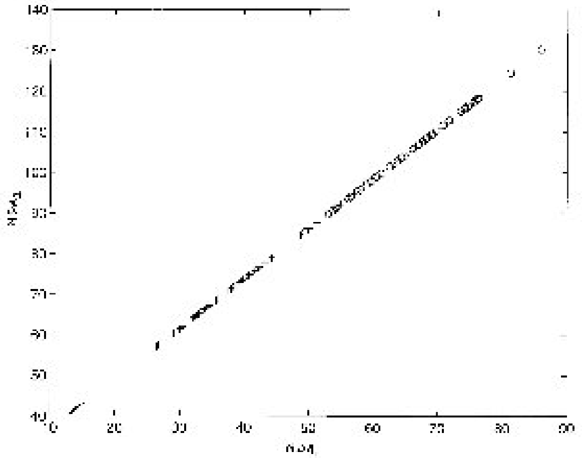

Fig. 6. Scatter plot NDA_1–NDA_2, in the extracted plane obtained by NDAN, of Iris data. Here '.', '+' and '∘' represent classes Iris Setosa, Iris Versicolor and Iris Virginica, respectively.

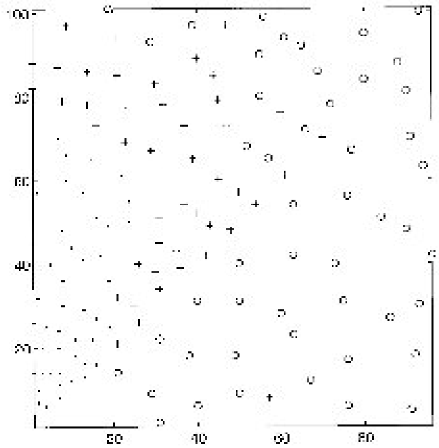

Fig. 7. Two-dimensional feature map obtained by SOM, of Iris data. Here '.', '+' and '∘' represent classes Iris Setosa, Iris Versicolor and Iris Virginica, respectively.

5. Conclusions and discussion

In this article we have demonstrated how the concept of neuro-fuzzy computing can be exploited for developing a methodology for feature extraction under unsupervised mode. The methodology developed involves connectionist minimization of a fuzzy feature evaluation index; thereby extracting an optimum transformed feature space along with the importance

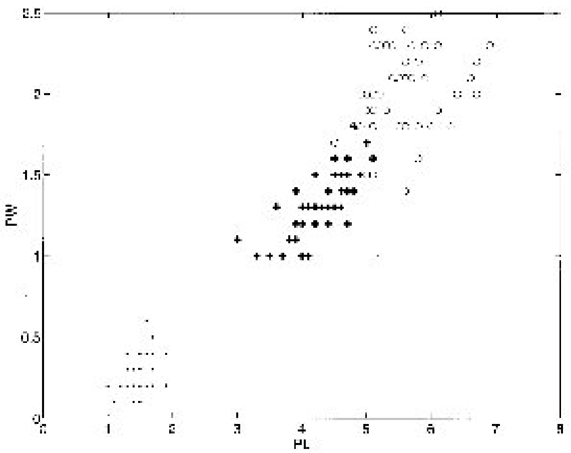

Fig. 8. Scatter plot *PL–PW* of Iris data. Here '.', '+' and 'o' represent classes Iris Setosa, Iris Versicolor and Iris Virginica, respectively.

of various features. The algorithm considers interdependence of the original features.

Although, the method is unsupervised, the extracted feature space has been able to provide better classification performance than the original ones for all the data sets. Results are compared with both unsupervised (PCAN and SOM) and supervised (NDAN) methods. It has been observed that the extent of overlapping region in the feature plane extracted by the neuro-fuzzy method is less than those obtained by the PCAN, NDAN and SOM. The classification ability of the extracted features obtained by the neuro-fuzzy method is much more than PCAN and SOM, but is slightly less than NDAN. Moreover, the neuro-fuzzy feature extraction preserves the data structure, cluster shape and inter pattern distances better than PCAN, NDAN and SOM.

Unlike NDAN and SOM, both neuro-fuzzy method and PCAN extract features without clustering/classifying the feature space explicitly. The neuro-fuzzy method, PCAN and SOM do not require to assume the class information of the patterns as well as the number of clusters. It is to be noted that the task of feature extraction by both the neuro-fuzzy method and NDAN involves projection of an n-dimensional original space directly to an n'-dimensional $(n' < n)$ transformed space. On the other hand, in the case of PCAN, this task involves projection of an n-dimensional original space to an n-dimensional transformed space, followed by selection of the best n' number of transformed components. Since the transformed features with low variances are ignored, there will be a loss of information in the resulting extracted space. This is also true for all the statistical feature extraction methods based on the K–L transformation.

In the present method, we have assumed linear transformation, as in the case of principal component analysis. However, this does not preclude the possibility of inclusion of nonlinear transformation by increasing the number of hidden layers in the network. It may be mentioned that Foley and Sammon [6] derived a set of discriminant vectors by selecting the projection axes one at a time under an orthogonality constraint. On the other hand, the present neuro-fuzzy method, as mentioned above, determines the extracted features simultaneously by minimizing the feature evaluation index. Regarding the time complexity of the neuro-fuzzy algorithm, we can say that it will be $O(Ts^2)$, where T is the number of iterations required for training the network, and s is the number of training samples.

In order to validate the results quantitatively, we have used a standard supervised classifier, viz., k-NN classifier, as an example; and the comparison is made in terms of % recognition score. Similar validation could also be done with an unsupervised classifier.

Acknowledgements

Dr. Rajat K. De is grateful to the Department of Atomic Energy and Council of Scientific & Industrial Research, Government of India for providing him a Dr. K.S. Krishnan Senior Research Fellowship and Research Associateship (Grant No. 9/93(58)/98-EMR.I), respectively.

Appendix A Principal component analysis network (PCAN) [17]

Principal component analysis is a well-known statistical method for feature extraction. It involves a linear orthogonal transform from an n-dimensional feature space to an n'-dimensional space, $n' \leqslant n$, such that the features in the new n'-dimensional space are uncorrelated and maximal amount of variance of the original data is preserved by only a small number of features.

The principal component analysis network (PCAN) architecture proposed by Rubner and Tavan [17] performs principal component analysis in a connectionist framework. It consists of n input and n' output nodes. An ith input node is connected to a jth output node with connection weight w_{ij}. All the output nodes are hierarchically organized in such a way that an lth output node is connected to a jth output node via connection weight $w_{lj}^{(\text{lat})}$ if and only if $l<j$. The training algorithm of the network is summarized below.

- Initialize all connection weights to small random values and choose the values of learning parameters.
- Repeat the following steps until all the lateral weights are sufficiently small for a given number of presentations (i.e., until their absolute values are below some threshold).
  - Randomly select an n-dimensional pattern $\mathbf{x}_p$ and present it to the input layer of the network. Compute the output ($\mathbf{x}'_p$) of the network, representing the corresponding pattern in n'-dimensional transformed space, using the equation

$$x'_{pj} = \mathbf{w}_j \cdot \mathbf{x}_p + \sum_{l<j} w_{lj}^{(\text{lat})} x'_{pl}, \quad j = 1,2,\ldots,n'. \tag{A.1}$$

  - Update $w_{ij}, \forall i,j$ following the Hebbian rule,

$$\Delta w_{ij} = \eta_1 x_{pi} x'_{pj}, \tag{A.2}$$

    where $\eta_1 > 0$ is the learning rate.
  - Normalize w_{ij} in such a way that $\|\mathbf{w}_j\| = 1$.
  - Update $w_{lj}^{(\text{lat})}$ by the anti-Hebbian rule,

$$\Delta w_{lj}^{(\text{lat})} = -\eta_2 x'_{pl} x'_{pj}, \tag{A.3}$$

    where η_2 is a positive learning parameter.

Appendix B. Nonlinear discriminant analysis network (NDAN) [22]

Nonlinear discriminant analysis network (NDAN) [22] is a multilayer feedforward network and is used to realize a nonlinear discriminant analysis. The main objective of the method is to project higher dimensional data set to a lower dimensional one under supervised mode of learning. The network consists of an input, one or more hidden and an output layers. The role of hidden layers is to implement a nonlinear transformation which projects input patterns in the original space to a space in which patterns are easily separated by the output layer.

The number of nodes in the input layer is the same as the number of features, and that in the output layer is equal to the number of pattern classes. We fix the number of nodes in the final hidden layer to n', the dimensionality of the projected space. The activation functions of the hidden nodes are nonlinear (sigmoid) and those of the input and output nodes are linear. The backpropagation learning algorithm is used to train the network which minimizes the squared error between its desired and actual outputs. After training, the outputs of nodes in the final hidden layer provide the feature values in the projected space.

References

[1] P. Baldi, K. Hornik, Neural networks and principal component analysis: learning from examples without local minima, IEEE Trans. Neural Networks 2 (1989) 53–58.

[2] J. Basak, R.K. De, S.K. Pal, Unsupervised feature selection using neuro-fuzzy approach, Pattern Recognition Lett. 19 (1998) 997–1006.

[3] P.A. Devijver, J. Kittler, Pattern Recognition, A Statistical Approach, Prentice-Hall, Inc., Englewood Cliffs, 1982.

[4] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (1936) 179–188.

[5] P. Foldiak, Adaptive network for optimal linear feature extraction, Proceedings IEEE International Joint Conference on Neural Networks, Vol. 1, Washington, DC, 1989, pp. 401–405.

[6] D.H. Foley, J.W. Sammon, An optimal set of discriminant vectors, IEEE Trans. Comput. C-24 (1975) 281–289.

[7] Y. Hayashi, A neural expert system with automated extraction of fuzzy if–then rules and its application to medical diagnosis, in: R.P. Lippmann, J.E. Moody, D.S. Touretzky (Eds.), Advances in Neural Information Processing Systems, Morgan Kaufmann, Los Altos, 1991, pp. 578–584.

[8] T. Kohonen, Self-organized network, Proc. IEEE 43 (1990) 59–69.

[9] M.A. Kraaijveld, J. Mao, A.K. Jain, A nonlinear projection method based on Kohonen's topology preserving maps, IEEE Trans. Neural Networks 6 (1995) 548–559.

[10] J. Lampinen, E. Oja, Distortion tolerant pattern recognition based on self-organizing feature extraction, IEEE Trans. Neural Networks 6 (1995) 539–547.

[11] D. Lowe, A.R. Webb, Optimized feature extraction and Bayes decision in feed-forward classifier networks, IEEE Trans. Pattern Anal. Machine Intelligence 13 (1991) 355–364.

[12] J. Mao, A.K. Jain, Artificial neural networks for feature extraction and multivariate data projection, IEEE Trans. Neural Networks 6 (1995) 296–317.

[13] S. Mitra, Fuzzy MLP based expert system for medical diagnosis, Fuzzy Sets and Systems 65 (1994) 285–296.

[14] S.K. Pal, Fuzzy set theoretic measures for automatic feature evaluation: II, Inform. Sci. 64 (1992) 165–179.

[15] S.K. Pal, B. Chakraborty, Fuzzy set theoretic measures for automatic feature evaluation, IEEE Trans. Systems, Man, Cybernet. 16 (1986) 754–760.

[16] S.K. Pal, D. Dutta Majumder, Fuzzy Mathematical Approach to Pattern Recognition, Wiley (Halsted Press), New York, 1986.

[17] J. Rubner, P. Tavan, A self-organizing network for principal component analysis, Europhys. Lett. 10 (1989) 693–698.

[18] E. Saund, Dimensionality-reduction using connectionist networks, IEEE Trans. Pattern Anal. Machine Intelligence 11 (1989) 304–314.

[19] W.A.C. Schmidt, J.P. Davis, Pattern recognition properties of various feature spaces for higher order neural networks, IEEE Trans. Pattern Anal. Machine Intelligence 15 (1993) 795–801.

[20] V.G. Sigillito, S.P. Wing, L.V. Hutton, K.B. Baker, Classification of radar returns from the ionosphere using neural networks, Johns Hopkins APL Tech. Dig. 10 (1989) 262–266.

[21] J.M. Steppe, K.W. Bauer Jr., Improved feature screening in feedforward neural networks, Neurocomputing 13 (1996) 47–58.

[22] A.R. Webb, D. Lowe, The optimized internal representation of multilayer classifier networks performs nonlinear discriminant analysis, Neural Networks 3 (1990) 367–375.