Eng Int Syst (2000) 2: 103–108

Engineering
Intelligent Systems

Pattern classification using fuzzy sets and neural nets: a case-based approach

Rajat K De and Sankar K Pal

Machine Intelligence Unit, Indian Statistical Institute, Calcutta 700035, INDIA. Email: {rajat,sankar}@isical.ac.in

The present article describes a case-based pattern classification method in connectionist framework using fuzzy set theory. Some labeled samples from each class are selected automatically as *cases* based on the concept of fuzzy similarity, and are represented as hidden nodes. The number of nodes is determined by the extent of fuzzy regions around *cases*. These are determined adaptively through *growing* and *pruning* under supervised training of the network. The effectiveness of the system, along with comparisons, has been demonstrated on various synthetic and real life pattern recognition problems for different extent of fuzzy regions.

Keywords: pattern classification, fuzzy sets, neural nets

1. INTRODUCTION

Case-based reasoning [1–3] may be defined as a model of reasoning that incorporates problem solving, understanding, and learning, and integrates all of them with memory processes. These tasks are performed using some typical situations, called *cases*, already experienced by the system. Systems based on this concept are finding widespread applications in various decision making processes e.g., medical diagnosis, law interpretation where the knowledge available is incomplete and/or evidence is sparse.

Artificial Neural Networks (ANNs), having the capability of fault tolerance, adaptivity and generalization, and scope for massive parallelism, are widely used in dealing with learning and recognition tasks. Fuzzy set theory enables one to handle uncertainties arising from deficiency (e.g., vagueness, incompleteness) in information, in different tasks of a pattern recognition system, in an efficient manner. *Neuro-fuzzy computing* which deals with the integration of the merits of artificial neural nets and fuzzy sets, has drawn the attention of researchers for the last few years, for designing artificially more intelligent systems. For the last few years, attempts are being made for combining the notion of case-based reasoning with the theories of ANN [4,5], fuzzy sets [6,7] and also with neuro-fuzzy computing [8,9], in order to develop artificially intelligent systems.

The present article is an attempt, in this regard, for developing a case-based pattern classification system using fuzzy set theory in connectionist framework. *Cases* are typically labeled patterns which represent different regions of the classes. Incorporation of fuzzy set theory helps in selecting the *cases* from ambiguous/overlapping regions. The concept of fuzzy similarity is used, in terms of distance measure, to determine the degree of similarity between two patterns. These tasks (e.g., computation of degree of similarity, selection of *cases* etc.) are performed in a connectionist framework whose architecture is determined adaptively through *growing* and *pruning* of nodes under supervised mode of training. The performance of the algorithm is compared with k-NN and Bayes maximum likelihood classifiers on various artificial and real life data sets.

2. METHODOLOGY

The methodology of the proposed case-based pattern recognition system involves the task of *selection of few samples from each class as cases*, followed by *classifica-*

tion of an unknown sample based on the *cases*. (For the sake of convenience, the samples which are not selected as *cases*, are referred to as patterns in the subsequent discussion.) For performing these tasks, let us define a concept of similarity between a pattern and a *case* using the notion of a fuzzy set representing a region around a *case*. The membership function characterizing this fuzzy set is such that higher its value, higher is the degree of similarity between a pattern and a *case*.

Let $\mathbf{x} = [x_1, x_2, ..., x_i, ..., x_n]$ be a pattern vector of known classification in an n-dimensional feature space containing M classes. $\xi_{l_k} = [\xi_{l_k 1}, \xi_{l_k 2}, ..., \xi_{l_k i}, ..., \xi_{l_k n}]$ denotes the l_kth *case* from kth class C_k. $\mu_{l_k}(\mathbf{x})$ is the degree of belonging of $\mathbf{x}$ to the fuzzy set R_{l_k} representing a region with ξ_{l_k} as its center. $d_{l_k}(\mathbf{x})$ stands for the distance between $\mathbf{x}$ and ξ_{l_k}.

The membership function denoting the degree of similarity between a pattern $\mathbf{x}$ and a *case* ξ_{l_k} is defined as

$$\begin{aligned} \mu_{l_k}(\mathbf{x}) &= 1 - 2\left(\frac{d_{l_k}(\mathbf{x})}{\lambda}\right)^2, & 0 \le d_{l_k}(\mathbf{x}) < \frac{\lambda}{2} \\ &= 2\left[1 - \frac{d_{l_k}(\mathbf{x})}{\lambda}\right]^2, & \frac{\lambda}{2} \le d_{l_k}(\mathbf{x}) < \lambda \\ &= 0, & \textit{otherwise} \end{aligned} \quad (1)$$

where λ is the bandwidth of the membership function, i.e. the separation between its two cross-over points where $\mu_{l_k} = 0.5$. The distance $d_{l_k}(\mathbf{x})$ may be expressed in many ways. Considering the Euclidian norm, we have

$$d_{l_k}(\mathbf{x}) = \left[\sum_i (x_i - \xi_{l_k i})^2\right]^{\frac{1}{2}} \quad (2)$$

It is clear from Eq. (1) that $\mu_{l_k}(\mathbf{x})$ decreases with the increase in $d_{l_k}(\mathbf{x})$ and *vice versa*. It is maximum (=1.0), iff $d_{l_k}(\mathbf{x})$ is zero (i.e. if a pattern ($\mathbf{x}$) and the l_kth *case* are identical). The value of $\mu_{l_k}(\mathbf{x})$ is minimum (= 0.0), if $d_{l_k}(\mathbf{x}) \ge \lambda$. When $d_{l_k}(\mathbf{x}) = \lambda/2$, $\mu_{l_k}(\mathbf{x})$ is 0.5, i.e. an ambiguous situation arises. Note that, one may define $\mu_{l_k}(\mathbf{x})$ in a different way satisfying the above-mentioned characteristics.

2.1 Selection of cases and class representation

First of all, a pattern $\mathbf{x}$ is selected randomly from any class C_k. It is considered as the first *case* if the *case-base* B_k corresponding to class C_k is empty. Otherwise, its membership values $\mu_{l_k}(\mathbf{x})$ (Eqs. (1–2)) corresponding to the fuzzy sets around the *cases* ξ_{l_k} in the *case-base* B_k, is computed. $\mathbf{x}$ is selected as a new *case*, if

$$\mu_{l_k}(\mathbf{x}) \le 0.5 \quad \textit{for all } l_k,$$

i.e. $\mathbf{x}$ does not fall within 0.5-cut of μ_{l_k}. When a *case* is selected, it is *inserted* into the *case-base*. After repeating this process over all the training patterns, a set of *cases* constituting the *case-base* for each class is obtained. The *case-base* B for the entire training set is the union of all B_ks, i.e. $B = \cup_{k=1}^{M} B_k$.

After the formation of this *case-base* B, *case* ξ_{l_k} for which $\mu_{l_k}(\mathbf{x}) \le 0.5$ is minimum, is *deleted* from B, if the number of patterns with $\mu_{l_k}(\mathbf{x}) > 0.5$ (or with $d_{l_k}(\mathbf{x}) < \lambda/2$) i.e. the number of patterns within 0.5-cut is less than some pre-specified number. The processes of *insertion* and *deletion* are repeated until the *case-base* becomes stable, i.e. the set of *cases* does not change further. This *deletion* process reduces the possibility of a spurious pattern being considered as a *case*.

Therefore, the class C_k can be viewed as a union of all fuzzy sets R_{l_k} representing the regions around its different *cases*, i.e.

$$C_k = \cup_{l_k=1}^{s_k} R_{l_k}$$

where s_k is the number of *cases* in class C_k. Note that as the value of λ increases, the extent of R_{l_k}s representing different regions around ξ_{l_k}s increases, and therefore, the number of *cases* s_k decreases.

Effect of λ

As λ increases, the extent of the region around a *case* increases, and therefore the number of *cases* required for representing a class decreases. This implies that the generalization capability of an individual *case* increases with increase in λ. Initially, although the number of *cases* decreases with the increase in λ, the generalization capability of individual *cases* dominates. For further increase in λ, the number of *cases* becomes so low that the generalization capability of the individual *cases* may not cope with proper representation of the class structures. As a result, the recognition score decreases. In other words, as λ increases, the possibility of wrong samples in the class increases. The ratio between the number of correct and incorrect samples falling within the class decreases with the increase in λ.

2.2 Classification

As described in Section 2.1, each class is modeled with a few membership functions defined around the *cases* selected from that class. The degree of belonging of a pattern $\mathbf{x}$ to class C_k is defined as

$$g_k(\mathbf{x}) = \max_{l_k}\{\mu_{l_k}(\mathbf{x})\} \quad (3)$$

or

$$g_k(\mathbf{x}) = \min_{l_k}\{\mu_{l_k}(\mathbf{x})\} \quad (4)$$

using connective property, or

$$g_k(\mathbf{x}) = \frac{1}{s_k}\sum_{l_k=1}^{s_k} \mu_{l_k}(\mathbf{x}) \quad (5)$$

using collective property, depending on the problem. Therefore, decide $\mathbf{x} \in C_k$ if

$$g_k(\mathbf{x}) = \max_t\{g_t(\mathbf{x})\}, \quad k, t = 1, 2, ... M \quad (4)$$

3. CONNECTIONIST REALIZATION

The methodology described in Section 2 is implemented in a layered network whose architecture is determined adaptively through *growing* and *pruning* of hidden nodes. Note that these *growing* and *pruning* phenomena correspond to the tasks of *insertion* and *deletion* of *cases*. These are described below.

3.1 Architecture

The connectionist model (Figure 1) consists of three layers, namely, input, hidden and output. The input layer represents the set of input features, i.e. for each feature there is a node (called input node) in the input layer. Similarly, for each *case* there is a node in the hidden layer. For each hidden node, there is an auxiliary node which makes the hidden node ON or OFF. An auxiliary node corresponding to a hidden node sends back signal to the input layer only when it sends a signal to the hidden node for making it ON. The hidden nodes are made ON one at a time keeping the remaining OFF. Finally, each node in the output layer functions as a Winner-Take-All network and represents a class.

The input nodes are connected to the hidden and auxiliary nodes by feedforward and feedback links respectively. The weight of the link connecting ith input node and l_kth hidden node is

$$w_{l_k i}^{(0)} = 1, \quad \forall l_k, i \tag{6}$$

The weight $w_{l_k i}^{(fb)}$ of a feedback link connecting the auxiliary node corresponding to a l_kth hidden node and ith input node is the same as the ith feature value of the l_kth case ($\xi_{l_k i}$). That is,

$$w_{l_k i}^{(fb)} = \xi_{l_k i} \tag{7}$$

The hidden layer is connected to the output layer via feedforward links. The weight ($w_{kl_k}^{(1)}$) of the link connecting the l_kth hidden node and the kth output node is 1, if the *case* corresponding to the hidden node belongs to class C_k. Otherwise, there is no such links between hidden nodes and output nodes. That is,

$$\begin{aligned} w_{kl_k}^{(1)} &= 1, \quad \text{if } \xi_{l_k} \in C_k \\ &= 0, \quad \text{otherwise} \end{aligned} \tag{8}$$

At the beginning, since the *case-base* is empty, there is no hidden node. Hence, the connectivity between the layers is not established. When there is at least one hidden node, a pattern $\mathbf{x}$ is presented to the input layer of the network. The activation of the ith input node, when the l_kth hidden node is ON, is given by

$$v_{l_k i}^{(0)} = (u_{l_k i}^{(0)})^2 \tag{9}$$

$u_{l_k}^{(0)}$ is the total input received by the ith input node when the l_kth hidden node is ON, and is given by

$$u_{l_k i}^{(0)} = x_{pi} - u_{l_k i}^{(fb)} \tag{10}$$

where $u_{l_k i}^{(fb)} = (-1) * w_{l_k i}^{(fb)}$ (−1 being the feedback activation of the auxiliary node corresponding to the l_kth hidden node is the feedback input received by the input node. The total input received by l_kth hidden node, when it is made ON, is

$$u_{l_k}^{(1)} = \sum_i v_{l_k i}^{(0)} * - w_{l_k i}^{(0)} \tag{11}$$

The activation function of the l_kth hidden node is the same as in Eq. (1). Thus, the activation $v_{l_k}^{(1)}$ of the l_kth hidden node is given by

$$\begin{aligned} v_{l_k}^{(1)} &= 1 - 2\left(\frac{(u_{l_k}^{(1)})^{\frac{1}{2}}}{\lambda}\right)^2, \quad 0 \le (u_{l_k}^{(1)})^{\frac{1}{2}} < \lambda/2 \\ &= 2\left[1 - \frac{(u_{l_k}^{(1)})^{\frac{1}{2}}}{\lambda}\right]^2, \quad \lambda/2 \le (u_{l_k}^{(1)})^{\frac{1}{2}} < \lambda \\ &= 0, \quad \text{otherwise.} \end{aligned} \tag{12}$$

Here the value of λ is stored in all the hidden nodes. The kth output node receives activations only from the hidden nodes corresponding to the *cases* in C_k. That is, the activation received by the kth output node from the l_kth hidden node is

$$u_{kl_k}^{(2)} = v_{l_k}^{(1)} * w_{kl_k}^{(1)} \tag{13}$$

Activation function of the kth output node has the same form as in Eq. (3). That is,

$$v_k^{(2)} = \max_{l_k} \{u_{kl_k}^{(2)}\} \tag{14}$$

Note that, here we have considered maximum of $u_{kl_k}^{(2)}$ as the activation value of the output node. However, minimum (Eq. (4)) or average (Eq. (5)) values could have also been used for this purpose depending upon the problem.

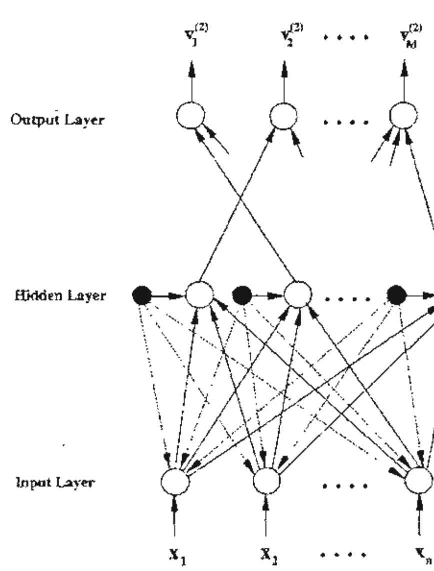

Figure 1 Schematic diagram of the neural network model for case-based classification. Black circles represent auxiliary nodes, and white circles represent input, hidden and output nodes

3.2 Training and formation of the network

The network whose architecture is described in Section

3.1, is formed through *growing* and *pruning* of the hidden nodes during the training phase under supervised mode. Initially there is only input and output layers. The patterns are presented in a random sequence to the input layer of the network. The first pattern that is presented to the network, is considered as a *case*. A hidden node along with its auxiliary node representing this *case* is added to the network. The connections of these auxiliary and hidden nodes with the input and output layers are established as described by Eqns. (6–8).

For the remaining patterns, their degrees of similarity with the *cases* represented by existing hidden nodes are computed, and if they are decided to be new *cases* (Section 2.1), hidden nodes are added through *growing* operation. After the process of addition is over, it is checked if there is any redundant hidden node. This is done through *pruning* operation depending on the criterion mentioned in Section 2.1. (In this connection, one may note the effect of λ in changing the number of *cases* and hence the hidden nodes.) These two operations, which together constitute a single iteration, are described below. These iterations are continued until the structure of the network becomes stable, i.e. till the set of hidden nodes representing the *cases* does not change.

Growing of hidden nodes:
For a pattern $\mathbf{x} \in C_k$, if $v_{l_k}^{(1)} \le 0.5$ and $\mathbf{w}_{l_k}^{(fb)} = \xi_{l_k} \in C_k$ for all the hidden nodes, $\mathbf{x}$ is selected as a *case*. A hidden node along with its auxiliary node is added to the network for representing this *case* and the links are established accordingly, using Eqns. (6–8). This process is called *growing of hidden nodes*. Note that the task 'insertion' of *cases* described in Section 2.1, is performed through this process.

Pruning of hidden nodes:
The l_kth hidden node is deleted, if

$$v_{l_k}^{(1)} = \min_{\xi_{l_k} = \mathbf{w}_{l_k}^{(fb)} \in C_k} v_{l_k}^{(1)} \le 0.5$$

and the number of training samples for which $V_{l_k} > 0.5$, is less than a pre-defined value. In this way, the network is pruned. Note that the task 'deletion' of *cases* described in Section 2.1, is performed through this process. ♣

During testing, an unknown pattern $\mathbf{x}$ is said to be in class C_k if the value of $v_k^{(2)}$ is the maximum of all such activation values of the output nodes.

4. EXPERIMENTAL RESULTS

The classification performance of the case-based system is compared with those of k-NN and Bayes maximum likelihood classifiers using both synthetic and real-life data. Here we have included the results of speech (vowel) [10] and medical data [11] only, in order to restrict the size of the article. In both the cases, 30% samples are considered during training, while the remaining 70% is used for testing.

The vowel data consists of a set of 871 Indian Telugu vowel sounds. These were uttered in a consonant-vowel-consonant context by three male speakers in the age group of 30 to 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. Figure 2 shows the overlapping nature of the six 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 [10]. This vowel data is being extensively used for two decades in the area of pattern recognition.

The medical data consisting of nine input features and four pattern classes, deals with various *Hepatobiliary disorders* [11] 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.

Tables 1 and 2 depict some of the results obtained with the above data sets for different values of λ. The first column of these tables indicates the number of iteration(s) required by the network until it stabilizes during training. It is found from these tables that the recognition scores on the training set, as expected, are higher than those on the test set. The recognition score during training decreases with the increase in the value of λ. On the other hand, for the test data, the recognition score increases with λ up to a certain value, beyond which it decreases. This can be explained as follows.

During training, the recognition score increases with decrease in λ due to better abstraction capability. While for the test data, as λ decreases, the modeling of class structures improves because of the increase in the number of *cases*, and therefore, the recognition score increases up to a certain value of λ. Beyond that, as mentioned in Section 2.1, the number of *cases* with poor generalization capability (i.e. membership functions with very small bandwidth) increases. As a result, the recognition score decreases due to *overlearning*.

As mentioned in Section 3.2, the number of hidden nodes of the network decreases with the increase in λ, for all the cases (Tables 1 and 2). Since class 'e' of vowel data

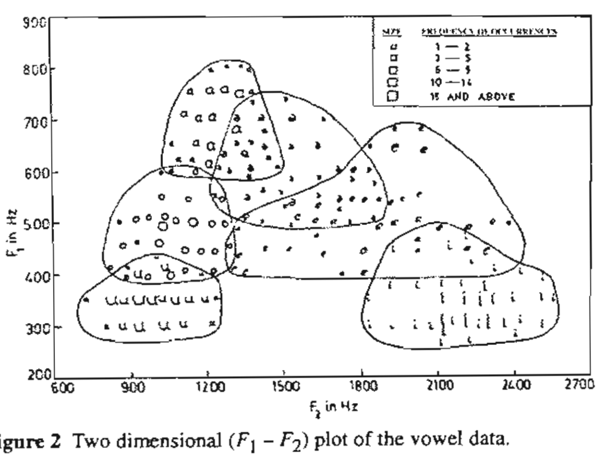

Figure 2 Two dimensional ($F_1 - F_2$) plot of the vowel data.

is most sparse as compared to the other classes (Figure 2), it needs more *cases* (and hence more hidden nodes) for its representation. This is reflected in Table 1. Similar observation holds good for medical data where class PH being most sparse, have the maximum number of hidden nodes. Note from Tables 1 and 2 the data sets, the stability of the architecture of the networks was achieved with a very few iterations. It is also to be noted that the pre-specified number (Section 2.1) for both the data sets was taken as 2.

In a part of the experiment, the performance of the *case-based* system (CBNN) is compared with those of k-NN and Bayes maximum likelihood classifiers. k-NN algorithm is executed taking k equal to $\sqrt{s}$, s being the number of training samples, as it is known that k-NN classifier approaches the optimal Bayes classifier [12, 13] for $k = \sqrt{s}$. For the Bayes maximum likelihood classifier, we assume a multivariate normal distribution of the samples with different dispersion matrices and *a priori* probabilities for different classes.

Table 3 depicts that CBNN performs better than both k-NN and Bayes maximum likelihood classifiers for vowel data. In the case of medical data, while the performance of CBNN on the training set is better than those obtained by the other two classifiers, the reverse is true on the test samples.

Table 1 Classification performance for different λ using vowel data

Number of iterations	λ	Class	Number of hidden nodes	Recognition score (%) Training set	Testing set
3	100.0	∂	21	95.24	41.18
		a	22	100.0	84.13
		i	42	98.04	72.73
		u	31	97.78	81.13
		e	53	98.39	67 59
		o	38	94.44	89.68
		Overall	207	97.30	75.0
3	150.0	∂	18	95.24	64.71
		a	13	96.15	93.65
		i	23	96.08	86.78
		u	20	88.89	86 79
		e	37	96.77	80.0
		o	26	92.59	85.71
		Overall	137	94.21	83.82
3	200.0	∂	16	80.95	64.71
		a	13	92.31	90.48
		i	21	98.04	87.60
		u	19	91.11	85.85
		e	36	93.55	81.38
		o	25	90.74	86.51
		Overall	130	92.28	83.99
1	250.0	∂	12	71.43	58.82
		a	9	88.46	80.95
		i	11	92.16	85.95
		u	9	84.44	72.64
		e	20	91.94	80.69
		o	14	81.48	74.60
		Overall	75	86.49	77.29
1	300.0	∂	10	57.14	52.94
		a	8	92.31	80.95
		i	10	92.16	86.78
		u	8	97.78	83.96
		e	20	88.71	80.69
		o	11	64.81	59.52
		Overall	67	83.78	75.82
3	350.0	∂	8	52.38	52.94
		a	7	92.31	95.24
		i	9	94.12	90.08
		u	8	97.78	89.62
		e	13	70.97	66.21
		o	8	46.30	42.86
		Overall	53	75.68	72.06
1	400.0	∂	8	57.14	56.86
		a	7	96.15	95.24
		i	7	88.24	86.78
		u	6	97.78	84.91
		e	10	69.35	65.52
		o	8	72.22	64.29
		Overall	46	80.31	75.16
1	450.0	∂	7	71.43	70.59
		a	5	84.62	68.25
		i	5	58.82	61.16
		u	6	93.33	83.02
		e	9	83.87	76.55
		o	6	68.52	67.46
		Overall	38	76.45	71.41

Table 1 Classification performance for different λ using medical data

Number of iterations	λ	Class	Number of hidden nodes	Recognition score (%) Training set	Testing set
1	150.0	ALD	17	61.76	32.93
		PH	30	81.13	48.80
		LC	19	91.89	73.56
		C	13	42.86	27.71
		Overall	79	71.07	46.42
1	160.0	ALD	20	71.43	56.79
		PH	35	70.37	29.84
		LC	17	78.95	66.28
		C	8	58.33	57.32
		Overall	80	69.94	50.13
1	170.0	ALD	20	71.43	58.02
		PH	34	68.52	29.03
		LC	17	78.95	66.28
		C	8	55.56	54.88
		Overall	79	68.71	49.60
1	180.0	ALD	20	68.57	56.79
		PH	34	72.22	31.45
		LC	16	71.05	61.63
		C	8	55.56	54.88
		Overall	78	67.48	49.06
1	190.0	ALD	19	80.00	61.73
		PH	33	77.78	36.29
		LC	14	76.32	68.60
		C	6	8.33	12.20
		Overall	72	62.58	43.97
7	200.0	ALD	15	76.47	58.54
		PH	25	71.70	45.60
		LC	14	72.97	68.97
		C	11	25.71	9.64
		Overall	137	62.89	45.89

Table 3 Comparative recognition score of various classifiers on the two sets of data

Data set	Class	Recognition score (%) CBNN Training	CBNN Testing	Bayes Training	Bayes Testing	k-NN Training	k-NN Testing
vowel	∂	80.95	64.71	38.10	43.14	23.81	33.33
	a	92.31	90.48	88.46	85.71	80.77	85.71
	i	98.04	87.60	90.20	85.12	88.24	85.12
	u	91.11	85.85	91.11	90.57	86.67	76.42
	e	93.55	81.38	75.81	80.69	75.81	77.93
	o	90.74	86.51	92.59	85.71	92.59	88.89
	Overall	92.28	83.99	83.01	81.70	79.92	78.43
medical	ALD	71.43	56.79	61.76	50.00	52.94	46.34
	PH	70.37	29.84	54.72	64.80	69.81	77.60
	LC	78.95	66.28	51.35	36.78	21.62	29.89
	C	58.33	57.32	91.43	75.90	54.29	61.45
	Overall	69.94	50.13	63.52	57.56	51.57	56.23

5. CONCLUSIONS

We have described a fuzzy case-based system for pattern classification in connectionist framework. Different *cases* have been selected during training of the network, and are stored as its parameters. The architecture of the network is determined adaptively through *growing* and *pruning* of hidden nodes. The number of hidden nodes increases with the decrease in extent λ (Eq. (1)) of the fuzzy region around a *case*. As λ decreases, the performance during training increases because of the higher number of representative *cases*. On the other hand, during testing, it increases with the decrease in λ up to a certain value, beyond which the performance deteriorates because of *overlearning* (poor generalization capability of the *cases*). It has been found that the *case-based* system performs better than both *k*-NN and Bayes maximum likelihood classifiers for vowel data. However, for medical data, the generalization capability of CBNN is seen to be poorer than *k*-NN and Bayes classifiers.

REFERENCES

1 Kolodner, J L. *Case-Based Reasoning*, Morgan Kaufmann, San Mateo (1993).
2 Aamodt, A and Plaza, E. "Case-based reasoning: Foundational issues, methodological variations and system approaches", *AI Com – Artificial Intelligence Communications*, Vol 7 (1994) pp 39–59.
3 Watson, I. *Applying Case-based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufmann, Boston, USA (1997).
4 Domeshek, E. "A case study of case indexing: Designing index feature sets to suit task demands and support parallelism", in *Advances in Connectionist and Neural Computation Theory* (Barnden, J and Holyoak, K eds.), Vol 2: *Analogical Connections*, Ablex, Norwood (1993)
5 Malek, M. *A connectionist indexing approach for CBR systems in Case-Based Reasoning Research and Development* (Veloso, M and Aamodt, A eds.). Springer-Verlag, Berlin, Germany (1995).
6 Petersen, J. "Similarity of fuzzy data in a case-based fuzzy system in anaesthesia", *Fuzzy Sets and Systems*, Vol 85 (1997) pp 247–62
7 Dubois, D and Prade, H. "Fuzzy set modeling in case-based reasoning", *International Journal of Intelligent Systems*, Vol 13 (1998) pp 345–73
8 Main, J, Dillon, T S and Khosla, R "Use of fuzzy feature vectors and neural vectors for case retrieval in case based systems" in Proc. *NAFIPS 1996 Biennial Conference of the North American Fuzzy Information Processing Society*, NY (1996) pp 438–43.
9 Liu, Z-Q and Yan, F. "Fuzzy neural network in case-based diagnostic system" *IEEE Trans. on Fuzzy Systems*, Vol 5 (1997) pp 209–22
10 Pal, S K and Dutta Majumder, D. *Fuzzy Mathematical Approach to Pattern Recognition*. John Wiley (Halsted Press), NY, USA (1986)
11 Hayashi, Y. "A neural expert system with automated extraction of fuzzy if-then rules and its application to medical diagnosi" in *Advances in Neural Information Processing Systems* (Lippmann, R P, Moody, J E and Touretzky, D S eds.), Morgan Kaufmann, Los Altos (1991) pp 578–84
12 Cover, T M and Hart, P E. "Nearest neighbor pattern classification", *IEEE Trans. on Information Theory*, Vol IT-13 (1967) pp 21–7
13 Fukunaga, K. *Introduction to Statistical Pattern Recognition*. Academic, NY, USA (1972)