Fuzzy Multi-Layer Perceptron, Inferencing and Rule Generation

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

***Abstract*— A connectionist expert system model, based on a fuzzy version of the multilayer perceptron developed by the authors, is proposed. It infers the output class membership value(s) of an input pattern and also generates a measure of certainty expressing confidence in the decision. The model is capable of querying the user for the *more important* input feature information, if and when required, in case of partial inputs. Justification for an inferred decision may be produced in rule form, when so desired by the user. The magnitudes of the connection weights of the *trained* neural network are utilized in every stage of the proposed inferencing procedure. The antecedent and consequent parts of the justificatory rules are provided in *natural* forms. The effectiveness of the algorithm is tested on the speech recognition problem, on some medical data and on artificially generated intractable (linearly nonseparable) pattern classes.**

I. Introduction

ARTIFICIAL neural networks [1, 2] are massively parallel interconnections of simple neurons that function as a collective system. An advantage of neural nets lies in their high computation rate provided by massive parallelism, so that real-time processing of huge data sets becomes feasible with proper hardware. Information is encoded among the various connection weights in a distributed manner. The utility of fuzzy sets [3, 4, 5] lies in their capability in modelling uncertain or ambiguous data so often encountered in real life. There have been several attempts recently [6, 7, 8] in making a fusion of fuzzy logic and neural networks for better performance in decision making systems. The uncertainties involved in the input description and output decision are taken care of by the concept of fuzzy sets while the neural net theory helps in generating the required decision regions.

An expert system [9, 10] is a computer program that functions in a narrow domain dealing with specialized knowledge generally possessed by human experts. It is expected to be able to draw conclusions without seeing all possible information and be capable of directing the acquisition of new information in an efficient manner. It should also be able to justify a conclusion arrived at. The major components of an expert system are the *knowledge base, inference engine* and *user interface*. Traditional rule-based expert systems encode the knowledge base in the form of *If-Then* rules while the connectionist expert system [11] uses the set of connection weights of the *trained* neural net model for this purpose. However, the knowledge base itself is a major source of uncertain information [10] in expert systems, the causes being unreliable information, imprecise descriptive languages, inferencing with incomplete information, and poor combination of knowledge from different experts.

In this work we consider an application of the fuzzy version of the MLP (already developed by the authors) [12] to design a connectionist expert system. The model is expected to be capable of handling uncertainty and/or impreciseness in the input representation, inferring output class membership value(s) for complete and/or partial inputs along with a certainty measure, querying the user for the *more essential* missing input information and providing justification (in the form of rules) for any inferred decision. Note that the input can be in quantitative, linguistic or set forms or a combination of these. The model is likely to be suitable in data-rich environments for designing *classification-type* expert systems.

Initially, in the learning phase the training samples are presented to the network in cycles until it finally converges to a minimum error solution. The connection weights in this stage constitute the knowledge base. Finally, in the testing phase the network infers the output class membership values for unknown test samples. When partial information about a test vector is presented at the input, the model either infers its category or asks the user for *relevant* information in the order of their relative importance (decided from the *learned* connection weights). A measure of confidence (certainty) expressing belief in the decision is also defined.

If asked by the user, the proposed model is capable of justifying its decision in rule form with the antecedent and consequent parts produced in linguistic and *natural* terms. The connection weights and the certainty measure are used for this purpose. It is expected that the model may be able to generate a number of such rules in *If-Then* form. These rules can then also be used to automatically form the knowledge base of a traditional expert system.

The effectiveness of the algorithm is demonstrated on the speech recognition problem, on some medical data and on artificially generated intractable (linearly nonseparable) pattern classes.

II. Fuzzy Version of the MLP

The MLP [2, 13, 14] consists of multiple layers of sigmoid processing elements or neurons that interact using weighted connections. Consider the network given in Fig. 1. The output of a neuron in any layer other than the input layer ($h > 0$)

Manuscript received November 10, 1992; revised July 26, 1993 and December 2, 1993.

S. Mitra and S. K. Pal are with Machine Intelligence Unit, Indian Statistical Institute, 203, B. T. Road, Calcutta - 700035, India.

IEEE Log Number 9400111.

1045–9227/95$04.00

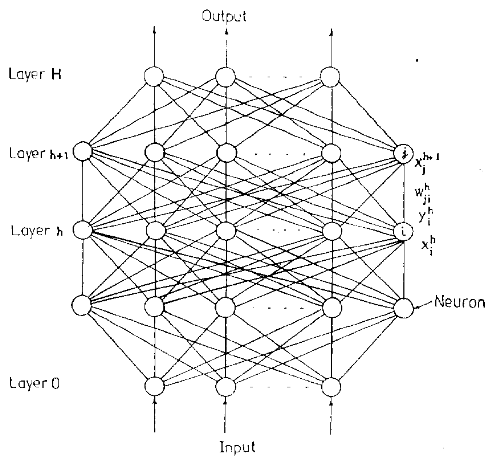


Fig. 1. The fuzzy neural network with three hidden layers.

is given as

$$y_j^{h+1} = \frac{1}{1+e^{-\sum_i y_i^h w_{ji}^h}} \quad (1)$$

where y_i^h is the state of the i^{th} neuron in the preceding h^{th} layer and w_{ji}^h is the weight of the connection from the i^{th} neuron in layer h to the j^{th} neuron in layer $h+1$. For nodes in the input layer we have $y_j^0 = x_j^0$, where x_j^0 is the j^{th} component of the input vector.

The Least Mean Square error in output vectors, for a given network weight vector w, is defined as

$$E(w) = \frac{1}{2}\sum_{j,c}\left(y_{j,c}^H(\mathbf{w}) - \mathbf{d_{j,c}}\right)^2 \quad (2)$$

where $y_{j,c}^H(\mathbf{w})$ is the state obtained for output node j in layer H in input-output case c and $d_{j,c}$ is its desired state specified by the *teacher*. One method for minimization of E is to apply the method of gradient-descent by starting with any set of weights and repeatedly updating each weight by an amount

$$\Delta w_{ji}^h(t) = -\epsilon\frac{\partial E}{\partial w_{ji}} + \alpha\Delta w_{ji}^h(t-1) \quad (3)$$

where the positive constant ϵ controls the descent, $0 \leq \alpha \leq 1$ is the momentum coefficient and t denotes the number of the iteration currently in progress. After a number of sweeps through the training set, the error E in (2) may be minimized.

The fuzzy version of the MLP, discussed here, is based on the model reported in [12] and is capable of classification of fuzzy patterns. Each input feature F_j is expressed in terms of membership values indicating a measure of belongingness to each of the linguistic properties *low, medium* and *high* modelled as π-sets [4]. An n-dimensional pattern $\vec{F_i} = [F_{i1}, F_{i2}, \ldots, F_{in}]$ is represented as a $3n$-dimensional vector

$$\vec{F_i} = \left[\mu_{\text{low}(F_{i1})}(\vec{F_i}), \mu_{\text{medium}(F_{i1})}(\vec{F_i}), \mu_{\text{high}(F_{i1})}(\vec{F_i}), \ldots, \mu_{\text{high}(F_{in})}(\vec{F_i})\right] \quad (4)$$

where the μ value indicates the membership to the corresponding linguistic π-set along each feature axis. The overlapping structure of the three π-functions for a particular input feature F_j (j^{th} axis) is the same as reported in [12].

It is to be noted here that an n-dimensional feature space is decomposed into 3^n overlapping sub-regions corresponding to the three primary properties. This enables the model to utilize more local information of the feature space [15] and is found to be more effective in handling linearly nonseparable pattern classes having nonconvex decision regions [16]. Therefore, numerical data are also fuzzified to enable a better handling of the feature space. Besides, this three-state structure of (4) helps in handling linguistic input suitably.

When the input feature is linguistic, its membership values for the π-sets *low, medium* and *high* are quantified as

$$\text{low} \equiv \left\{\frac{0.95}{L}, \frac{0.6}{M}, \frac{0.02}{H}\right\}$$
$$\text{medium} \equiv \left\{\frac{0.7}{L}, \frac{0.95}{M}, \frac{0.7}{H}\right\}$$
$$\text{high} \equiv \left\{\frac{0.02}{L}, \frac{0.6}{M}, \frac{0.95}{H}\right\} \quad (5)$$

When F_j is numerical we use the π-fuzzy sets [17] (in the one-dimensional form), with range [0, 1], given as

$$\pi(F_j; c, \lambda) = \begin{cases} 2\left(1 - \frac{|F_j - c|}{\lambda}\right)^2, & \text{for } \frac{\lambda}{2} \leq |F_j - c| \leq \lambda \\ 1 - 2\left(\frac{|F_j - c|}{\lambda}\right)^2, & \text{for } 0 \leq |F_j - c| \leq \frac{\lambda}{2} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $\lambda > 0$ is the radius of the π-function with c as the central point. The choice of λ's and c's for each of the linguistic properties *low, medium* and *high* are the same as reported in [12].

To model real-life data with finite belongingness to more than one class, we clamp the desired membership values (lying in the range [0, 1]) at the output nodes during training. For an l-class problem domain, the membership of the i^{th} pattern to class C_k is defined as

$$\mu_k\left(\vec{F_i}\right) = \frac{1}{1+\left(\frac{z_{ik}}{F_d}\right)^{F_e}} \quad (7)$$

where z_{ik} is the weighted distance between the i^{th} pattern and the mean of the k^{th} class (based on the training set) and the positive constants F_d and F_e are the denominational and exponential fuzzy generators controlling the amount of fuzziness in this class-membership set. For the i^{th} input pattern we define the desired output of the j^{th} output node as

$$d_j = \mu_j(\vec{F_i}) \quad (8)$$

where $0 \leq d_j \leq 1$ for all j. When the pattern classes are known to be nonfuzzy, z_{ik} of (7) may be set to 0 for a particular class and *infinity* for the remaining classes so that $\mu_k(\vec{F_i}) \in \{0, 1\}$.

The ϵ of (3) is gradually decreased in discrete steps, taking values from the chosen set {2, 1, 0.5, 0.3, 0.1, 0.05, 0.01, 0.005, 0.001}, while the momentum factor α is also decreased [12].

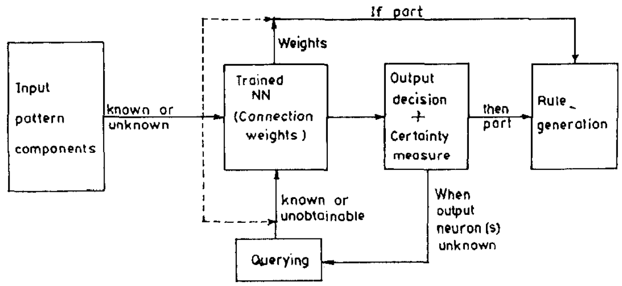


Fig. 2. Block diagram of the inferencing and rule generation phases of the model.

III. Inferencing in the Fuzzy Expert System Model

The most difficult, time-consuming and expensive task in building an expert system is constructing and debugging its knowledge base. In practice the knowledge base construction can be said to be the *only* real task in building an expert system, given the proliferating presence of *expert shells*. Several approaches have been explored for easing this knowledge-acquisition bottleneck. Connectionist expert systems offer an alternative approach in this regard. Rules are not required to be supplied by humans. Instead, the connection weights encode among themselves, in a distributed fashion, the information conveyed by the input-output combinations of the training set. Such models are especially suitable in data-rich environments and enable human intervention to be minimized. Moreover, using fuzzy neural nets for this purpose, helps one to incorporate the advantages of approximate reasoning [18] into the connectionist expert system. This leads to the design of fuzzy connectionist expert systems [19, 20]. A study of neuro-fuzzy expert systems may be found in [21].

In this work we consider an $(H+1)$-layered fuzzy MLP (as depicted in Fig. 1) with $3n$ neurons in the input layer and l neurons in the output layer, such that there are $H-1$ hidden layers. The input vector with components x_j^0 represented as $\vec{F}$ by (4) is clamped at the input layer while the desired l-dimensional output vector with components d_j by (8) is clamped during training at the output layer. At the end of the training phase the model is supposed to have encoded the input-output information distributed among its connection weights. This constitutes the *knowledge base* of the desired expert system. Handling of imprecise inputs is possible and natural decision is obtained associated with a certainty measure denoting the confidence in the decision. The model is capable of inferencing based on complete and/or partial information, querying the user for unknown input variables that are key to reaching a decision, and producing justifications for inferences in the form of *If-Then* rules. Fig. 2 gives an overall view of the various stages involved in the process of inferencing and rule generation.

A. Input Representation

The input can be in quantitative, linguistic or set forms or a combination of these. It is represented as a combination of memberships to the three primary linguistic properties *low, medium* and *high* as in (4), modelled as π-functions. When the information is in exact numerical form like F_j is r_1, say, we determine the membership values in the corresponding 3-dimensional space of (4) by the π-function using (6).

When the input is given as F_j is *prop* (say), where *prop* stands for any of the primary linguistic properties *low, medium* or *high*, the membership values in the 3-dimensional space of (4) are assigned using the π-sets of (5). The proposed model can also handle the linguistic hedges [15] *very, more or less (Mol)* and *not*. The sets *very low* and *low* or, say, *very high* and *high* are considered to be pairs of different but overlapping sets [15], such that the *minimum (maximum)* feature value has a higher membership to the set *very low (very high)* as compared to that in the set *low (high)*. Hence π-functions are found to be appropriate for modelling these linguistic sets. The hedge *not* is defined as

$$\mu_{\mathrm{Not}(A)} = 1 - \mu_A(x) \tag{9}$$

In the set form, the input is a mixture of linguistic hedges and quantitative terms. Since the linguistic term increases the impreciseness in the information, the membership value of a quantitative term is lower when modified by a hedge [15]. The modifiers used are *about, less than, greater than* and *between.*

If any input feature F_j is *not available* or *missing*, we clamp the three corresponding neurons $x_k^0 = x_{k+1}^0 = x_{k+2}^0 = 0.5$, such that $k = (j-1)*3+1$. Here $1 \le k \le 3n$ and $1 \le j \le n$, where n is the dimension of the input pattern vector. We use

$$\text{no information} \equiv \left\{\frac{0.5}{L}, \frac{0.5}{M}, \frac{0.5}{H}\right\} \tag{10}$$

as 0.5 represents the *most ambiguous* value in the fuzzy membership concept. We also tag these input neurons with $noinf_k^0 = noinf_{k+1}^0 = noinf_{k+2}^0 = 1$. Note that in all other cases the variable $noinf_k^0$ is tagged with 0 for the corresponding input neuron k, indicating absence of ambiguity in its input information.

The appropriate input membership values obtained by (4–6,10), with/without the hedges or modifiers, are clamped at the corresponding input neurons.

B. Forward Pass

The l-dimensional output vector with components y_j^H is computed using (1) in a single forward pass. This output vector, with components in the range [0, 1], gives the inferred membership values of the test pattern to the l output classes. Associated with each neuron j in layer $h+1$ are also

- its confidence estimation factor $conf_j^{h+1}$
- a variable $unknown_j^{h+1}$ providing the sum of the weighted information from the preceding *ambiguous* neurons i in layer h having $noinf_i^h = 1$
- a variable $known_j^{h+1}$ giving the sum of the weighted information from the (remaining) *non-ambiguous* preceding neurons with $noinf_i^h = 0$.

Note that for a neuron j in layer $h+1$ with no preceding neurons i tagged with $noinf_i^h = 1$, we have $unknown_j^{h+1} = 0$. For neuron j we define

$$unknown_j^{h+1} = \sum_i w_{ji}^h y_i^h$$

$$unden_j^{h+1} = \sum_i |w_{ji}^h| \tag{11}$$

for all i having $noinf_i^h = 1$ and

$$known_j^{h+1} = \sum_i w_{ji}^h y_i^h \tag{12}$$

for all i with $noinf_i^h = 0$, where for neurons in layer $h > 0$ we have

$$noinf_j^h = \begin{cases} 1 & \text{if } |known_j^h| \le |unknown_j^h| \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

For neuron j in the input layer ($h = 0$), the value of $noinf_j^0$ is assigned as explained earlier. Neuron j with $noinf_j^h = 1$ signifies the lack of meaningful information. For an input neuron this implies missing input information while for other neurons ($h > 0$) this is an indicator to the transmission of a larger proportion of weighted *ambiguous information* as compared to *more certain information* from the input layer. Using (1,11–13), we define

$$conf_j^h = \begin{cases} \left| \dfrac{\sum_i y_i^{h-1} w_{ji}^{h-1}}{unden_j^h} \right| & \text{if } noinf_j^h = 1 \text{ and } h > 0 \\ y_j^h & \text{otherwise} \end{cases} \tag{14}$$

Note that $conf_j^h$ is comparable either among the set of neurons having $noinf_j^h = 1$, or among those with $noinf_j^h = 0$, but not between the neurons belonging to these two different sets. In the output layer ($h = H$) if $noinf_j^H = 0$ then $conf_j^H$ is higher for neurons having larger y_j^H, implying a greater belongingness to output class j. Hence this is a measure of the confidence in the decision. However if $noinf_j^H = 1$ then $conf_j^H$ gives a measure of the confidence of the *ambiguous* neuron output. This is because as $unden_j^h$ by (11) (absolute sum of connection weights from *ambiguous* preceding layer neurons) increases, the confidence $conf_j^h$ decreases and vice versa.

If there is no output neuron j with $noinf_j^H = 1$, then the system finalizes the decision inferred irrespective of whether the input information is complete or partial. In case of partial inputs, this implies presence of all the *necessary* features required for taking the decision. It may be mentioned that the weights (learned during training), that constitute the *knowledge-base*, play an important part in determining whether a missing input feature information is *essential* to the final inferred decision or not. This is because these weights are used in computing the $noinf_j^h$'s for the neurons by (11–13) and these in turn determine whether the inferred decision may be taken.

It is to be noted that the difficulty in arriving at a particular decision in favor of class j is dependent not only on the membership value y_j^H but also on its differences with other class membership values y_i^H, where $i \ne j$. To take this factor into account, a certainty measure (for each output neuron) is defined as

$$bel_j^H = y_j^H - \sum_{i \ne j} y_i^H \tag{15}$$

where $bel_j^H \le 1$. The higher the value of $bel_j^H (> 0)$, the lower is the difficulty in deciding an output class j and hence the greater is the degree of certainty of the output decision.

C. Querying

If the system has not yet reached a conclusion at the output layer (as explained in Section III. B.) to complete the session, it must find an input neuron with *unknown* activation and ask the user for its value. If there is any neuron j in the output layer H with $noinf_j^H = 1$ by (13), we begin the querying phase.

We select the *unknown* output neuron j_1 from among the neurons with $noinf_j^H = 1$ such that $conf_{j_1}^H$ by (14) (among them) is maximum. This enables starting the process at an output neuron that is *most certain* among the *ambiguous* neurons. We pursue the path from neuron j_1 in layer H, in a *top-down* manner, to find the *ambiguous* neuron i_1 in the preceding layer ($h = H-1$) with the greatest absolute influence on neuron j_1. For this, we select $i = i_1$ such that

$$|w_{j_1 i_1}^h * y_{i_1}^h| = \max_i |w_{j_1 i}^h * y_i^h| \text{ where } noinf_i^h = 1 \tag{16}$$

This process is repeated until the input layer ($h = 0$) is reached. Then the model queries the user for the value of the corresponding input feature u_1 such that

$$u_1 = (i_1 - 1) \bmod 3 + 1 \tag{17}$$

where $1 \le i_1 \le 3n$, $1 \le u_1 \le n$ and n is the dimension of the input pattern vector.

When the user is asked for the value of a *missing* variable, she can respond in any of the forms stated in Section III.A. However if a *missing* input variable of (10) is found to be missing once again, we now tag it as *unobtainable*. This implies that the value of this variable will not be available for the remainder of this session. The inferencing mechanism treats such variables as *known* with values $x_{k_1}^0 = x_{k_1+1}^0 = x_{k_1+2}^0 = 0.5$ but with $noinf_{k_1}^0 = noinf_{k_1+1}^0 = noinf_{k_1+2}^0 = 0$ such that $k_1 = (u_1 - 1) * 3 + 1$. We now have

$$information \equiv \left\{ \frac{0.5}{L}, \frac{0.5}{M}, \frac{0.5}{H} \right\} \tag{18}$$

Note the difference from (10) in the value of $noinf_k^0$ and its effect in the confidence estimation by (11–14). The response from an *unobtainable* input variable might allow the neuron activations in the following layers to be inferred, unlike that of a *missing* variable. Besides, a *missing* variable has a temporary value of 0.5 that may be changed later in the session, whereas an *unobtainable* variable has a *known final* value of 0.5.

Once the requested input variable is supplied by the user, the procedure in Section III. B. is followed either to infer a decision or to again continue with further querying. On completion of this phase all neurons in the output layer have $noinf_j^H = 0$ by (13).

D. Justification

The user can ask the system why it inferred a particular conclusion. The system answers with an *If-Then* rule applicable to the case at hand. It is to be noted that these *If-Then* rules are not represented explicitly in the knowledge base; they are generated by the *inferencing system* from the connection weights as needed for explanations. As the model has already inferred a conclusion (in this stage), we take a subset of the currently known information to justify this decision. A particular conclusion regarding output j is inferred depending upon the certainty measure bel_j^H. It is ensured that output nodes j with $bel_j^H > 0$ (or, large y_j^H values) are chosen for obtaining the justification. This is because explanation becomes feasible only when the decision is not uncertain.

Output Layer: Let the user ask for the justification about a conclusion regarding class j. Starting from the output layer $h = H$, the process continues in a *top-down* manner until the input layer ($h = 0$) is reached. In the first step, for layer H, we select those neurons i in the preceding layer that have a positive impact on the conclusion at output neuron j. Hence we choose neuron i of layer $H-1$ if $w_{ji}^{H-1} > 0$. Let the set of m_{H-1} neurons of layer $H-1$, so selected, be $\left\{a_1^{H-1}, a_2^{H-1}, \ldots, a_{m_{H-1}}^{H-1}\right\}$ and let their connection weights to neuron j in layer H be given as $\left\{wet_{a_1^{H-1}} = w_{ja_1}^{H-1}, \ldots, wet_{a_{m_{H-1}}^{H-1}} = w_{ja_{m_{H-1}}}^{H-1}\right\}$. For the remaining layers we obtain the *maximum weighted* paths through these neurons down to the input layer.

Intermediate Layers: We select neuron i in layer $0 < h < H-1$ if

$$y_i^h > 0.5, \text{ and}$$
$$wet_{i^h} = \max_{a_k^{h+1}}[wet_{a_k^{h+1}} + w_{a_k i}^h] \tag{19}$$

such that $wet_{i^h} > 0$. Let the set of m_h neurons so chosen be given by $\{a_1^h, a_2^h, \ldots, a_{m_h}^h\}$ and their cumulative link weights to neuron j in layer H be $\left\{wet_{a_1^h}, wet_{a_2^h}, \ldots, wet_{a_{m_h}^h}\right\}$ respectively, by (19). Note that this heuristic ensures that each of the selected m_h neurons have a significant output response y_i^h. This implies choosing a path with neurons that are currently active for deciding the conclusion that is being justified. It also enables each neuron i to lie along one of the *maximum weighted* paths from the input layer ($h = 0$) to the output node j in $h = H$, by choosing only one of the m_{h+1} previously selected paths that provides the largest net weight wet_{i^h}.

Input Layer: Let the process of (19) result in m_0 chosen neurons (paths) in (from) the input layer ($h = 0$). These neurons indicate inputs that are *known* and have contributed to the ultimate positivity of the conclusion at neuron j in the output layer H. It may happen that $m_0 = 0$, such that no clear justification may be provided for a particular input-output case. This implies that no suitable path can be selected by (19) and the process terminates.

Let the set of the selected m_0 input neurons be $\{a_1^0, a_2^0, \ldots, a_{m_0}^0\}$ and their corresponding path weights to neuron j in layer H be $\left\{wet_{a_1^0}, wet_{a_2^0}, \ldots, wet_{a_{m_0}^0}\right\}$. We arrange these neurons in the decreasing order of their net impacts, where we define the net impact for neuron i as

$$net\ impact_i = y_i^0 * wet_{i^0}$$

Then we generate clauses for an *If-Then* rule from this ordered list until

$$\sum_{i_s} wet_{i_s^0} > 2 \sum_{i_n} wet_{i_n^0} \tag{20}$$

where i_s indicates the input neurons selected for the clauses and i_n denotes the input neurons remaining from the set $\{a_1^0, a_2^0, ..., a_{m_0}^0\}$ such that

$$|i_s| + |i_n| = m_0$$

and $|i_s|$, $|i_n|$ refer respectively to the number of neurons selected and remaining from the said set. This heuristic allows selection of those *currently active* input neurons contributing *the most* to the final conclusion (among those lying along the maximum weighted paths to the output node j) as the clauses of the antecedent part of a rule. Hence, it enables the *currently active* test pattern inputs (current evidence) to influence the generated *knowledge base* (connection weights learned during training) in producing a rule to justify the *current* inference.

Clause Generation: For a neuron i_{s_1} in the input layer ($h = 0$), selected for clause generation, the corresponding input feature u_{s_1} is obtained as in (17). The antecedent of the rule is given in linguistic form with the linguistic property being determined as

$$prop = \begin{cases} \text{low} & \text{if } i_s1 - 3(u_s1 - 1) = 1 \\ \text{medium} & \text{if } i_s1 - 3(u_s1 - 1) = 2 \\ \text{high} & \text{otherwise} \end{cases} \tag{21}$$

Here, the 3-dimensional components for the input feature u_{s_1} correspond to the appropriate part of the test pattern vector (given in quantitative, linguistic or set form and converted to the respective 3-dimensional space of (4)). Suppose that the relevant input feature had been initially supplied in linguistic form as *medium* with the individual components given by (5). The neuron i_{s_1} selected for clause generation by (19–20) can, however, result in feature u_{s_1} corresponding to any of the three properties *low, medium* or *high* by (21). This is because the path generated during backtracking is primarily determined by the connection weight magnitudes encoded during training. However, the test pattern component magnitudes at the input also play a part in determining whether the input neuron i_{s_1} can be selected or not. In the example under consideration, the input feature components being {0.7, 0.95, 0.7}, the linguistic property *prop* can be either *low* or *medium* or *high* and is not constrained to be *medium* only. Therefore, feature properties highlighted for the input pattern may not necessarily be reflected in a similar manner while selecting the value of *prop* in feature u_{s_1} for a clause of the rule. In fact, such an input feature component may also not be selected at all as an antecedent clause.

A linguistic hedge *very, more or less* or *not* may be attached to the linguistic property in the antecedent part, if necessary. We use the mean square distance $d(u_{s_1}, pr_m)$ between the 3-dimensional input values at the neurons corresponding to feature u_{s_1} and the linguistic property *prop* by (21), with or without modifiers, represented as pr_m. The corresponding 3-dimensional values of pr_m (without modifiers) for *prop* are given by (5). The incorporation of the modifiers *very, more or less* and *not* result in the application of different operators (as reported in [15]) to generate the corresponding modified values for pr_m. That value of pr_m (with/without modifiers) for which $d(u_{s_1}, pr_m)$ is the *minimum* is selected as the antecedent clause corresponding to feature u_{s_1} (or neuron i_{s_1}) for the rule justifying the conclusion regarding output neuron j.

This procedure is repeated for all the $|i_s|$ neurons selected by (20) to generate a set of conjunctive antecedent clauses for the rule regarding inference at output node j. All input features (of the test pattern) need not necessarily be selected for antecedent clause generation.

Consequent Deduction: The consequent part of the rule can be stated in quantitative form as membership value y_j^H to class j. However a more *natural* form of decision can also be provided for the class j, having significant membership value y_j^H, considering the value of bel_j^H of (15). For the linguistic output form, we use

1. *very likely* for $0.8 \leq bel_j^H \leq 1$
2. *likely* for $0.6 \leq bel_j^H < 0.8$
3. *more or less likely* for $0.4 \leq bel_j^H < 0.6$
4. *not unlikely* for $0.1 \leq bel_j^H < 0.4$
5. *unable to recognize* for $bel_j^H < 0.1$

In principle it should be possible to examine a connectionist network and produce every such *If-Then* rule. These rules can also be used to form the knowledge base of a traditional expert system.

An Example: Consider the simple 3-layered network given in Fig. 3 demonstrating a simple rule generation instance regarding class 1. Let the paths be generated by (19). A sample set of connection weights w_{ji}^h, input activation y_i^0 and the corresponding linguistic labels are depicted in the figure. The solid and dotted-dashed paths (that have been selected) terminate at input neurons i_s and i_n respectively, as determined by (20). The dashed lines indicate the paths not selected by (19), using the w_{ji}^h and y_i^h values in the process. Let the certainty measure for the output neuron under consideration be 0.7. Then the rule generated by the model in this case to justify its conclusion regarding class 1 would be

If F_1 is *very medium AND F_2 is high*
then *likely* class 1.

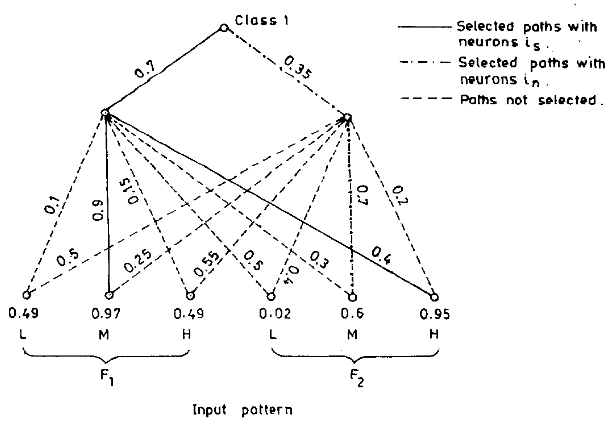


Fig. 3. An example to demonstrate the rule generation scheme by back-tracking.

In this case, the *net path weights* by (20) at the end of the clause selection process are found to be 2.7 (= 1.6 + 1.1) and 1.05 for the *selected* i_s and *not selected* i_n neurons respectively such that $2.7 > 2 * 1.05$. The modifier *very* is obtained by applying appropriate operators [15], and this is found to result in the *minimum* value for $d(u_{s_1}, pr_m)$.

To demonstrate querying, let us consider F_1 to be initially *unknown.* Then $y_1^0 = y_2^0 = y_3^0 = 0.5$, with the other values corresponding to those given in Fig. 3. From (11–13), we have $known_1^1 = 0.57$, $known_2^1 = 0.618$, $unknown_1^1 = 0.575$, $unknown_2^1 = 0.65$, and therefore $noinf_1^1 = noinf_2^1 = noinf_1^2 = 1$. As the system cannot reach any conclusion in this state, the querying phase is started. In this case, the only *unknown* input feature is F_1 and it can be supplied in any of the forms mentioned in Section III. A.

IV. Implementation and Results

The above-mentioned algorithm was first tested on 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. Thus the dimension of the input vector in (4) for the proposed model is 9. Note that the boundaries of the classes in the given data set are seen to be ill-defined (fuzzy). Fig. 4 shows a 2D projection of the 3D feature space of the six vowel classes $(\partial, a, i, u, e, o)$ in the $F_1 - F_2$ plane (for ease of depiction). The training data has the complete set of input features in the 9-dimensional form while the desired output gives the membership to the 6 vowel classes. The test set uses complete/partial sets of inputs and the appropriate classification is inferred by the trained neural model.

The model has also been implemented on a medical diagnosis problem that deals with *kala-azar* [22], a tropical disease, using a set of 68 patient cases. The input features are the symptoms while the output indicates the presence or absence of the disease. The symptoms are the measurements of *blood*

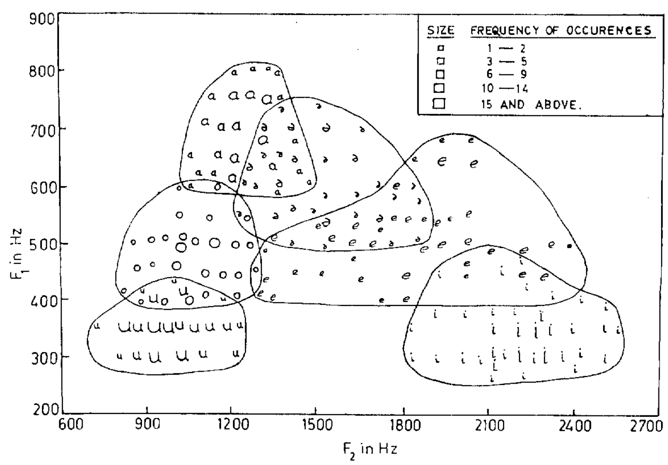


Fig. 4. Vowel diagram in the $F_1 - F_2$ plane.

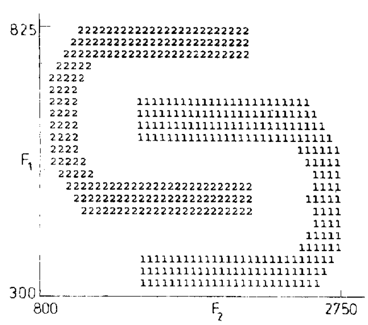


Fig. 5. Pattern Set A in the $F_1 - F_2$ plane.

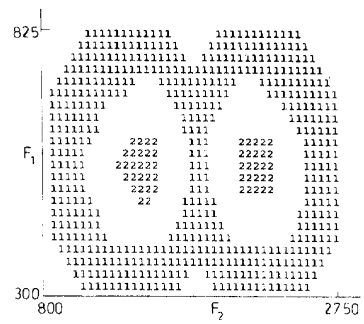


Fig. 6. Pattern Set B in the $F_1 - F_2$ plane.

urea (mg %), *serum creatinine* (mg %), *urinary creatinine* (mg %) and *creatinine clearance* (ml/min) indicated respectively as F_1, F_2, F_3 and F_4. These are represented in the linguistic form of (4). The training data has the complete set of symptoms with the desired classification indicating presence or absence of the disease.

Lastly, the model was used on two sets (A, B respectively) of artificially generated intractable (linearly nonseparable) pattern classes represented in the $2D$ feature space $F_1 - F_2$, each set consisting of 880 pattern points. These are depicted in Figs. 5–6. The training set consists of the complete pattern vectors in the 6-dimensional form of (4).

A. Vowel Data

The details regarding the classification performance on various training and test sets as well as the choice of the parameters for the said model (on the vowel data) have already been reported in [12]. Here we demonstrate a sample of the inferencing ability of a *trained neural model* (with five layers having 10 nodes per hidden layer) that functions as a *knowledge base* for the vowel recognition problem. It was trained using 50% samples from each representative class. The results are demonstrated in Tables I-III.

Table I illustrates the inferred output responses of the proposed model on a set of partial and complete input feature vectors. It is observed that often the two features F_1 and F_2 are sufficient for reaching a conclusion. This may easily be verified from the 2D representation of the vowel data in Fig. 4. Here the 2^{nd} entry corresponds to no particular vowel class and hence the certainty measure is appreciably low with both classes e and i registering *ambiguous* output membership values slightly less than 0.5. The $4a^{th}$ entry has only one accurate input value corresponding to F_1. Hence this maps to a line parallel to the F_2 axis at $F_1 = 700$ in Fig. 4. Note that both classes a and ∂ register positive belongingness, although class a is the *more likely* winner. On the other hand the 3^{rd} entry, with a complete feature vector, specifies a more certain decision in favor of class a. In entry $4b$, with a *certain* value for F_2, the decision shifts in favor of class e. The 5^{th} entry also possesses finite possibility of belongingness to classes e and i, as verified from the vowel diagram. However the certainty measure is indicative of the *uncertainty* in the decision. The *ambiguity* of the 6^{th} and 7^{th} entries are evident both from Fig. 4 as well as the two highest output membership values and the certainty measures. The $11a^{th}$ entry corresponds to a horizontal band across Fig. 4 around $F_1 = 350$. The classes e and i, having the two highest horizontal coverages in this region, correspond to the significant responses obtained. This may be contrasted with entry $4a$ where at least F_1 has a *definite* value 700. On the other hand, entry $11a$ corresponds to a

TABLE I
INFERRED OUTPUT RESPONSES AND CERTAINTY MEASURES FOR A SET OF VOWEL DATA, USING A FIVE-LAYERED FUZZY MLP HAVING 10 NODES PER HIDDEN LAYER WITH $perc = 50$

Sr.	Input features			Highest output		Significant 2^{nd} choice		Certainty
No.	F_1	F_2	F_3	Class j	Membership y_j^H	Class	Membership	bel_j^H
1	300	900	*missg.*	u	0.89	-	-	0.88
2	250	1550	*unobt.*	e	0.49	i	0.47	0.02
3	700	1000	2600	a	0.89	-	-	0.89
4a	700	*missg.*	*missg.*	a	0.85	∂	0.14	0.71
4b	700	2300	*missg.*	e	0.77	-	-	0.66
5	450	2400	*missg.*	e	0.70	i	0.11	0.47
6	600	1200	*missg.*	∂	0.71	o	0.27	0.39
7	*low*	*very low*	*missg.*	u	0.48	o	0.35	0.10
8	*high*	*Mol low*	*missg.*	a	0.91	-	-	0.91
9	*between* 500 & 600	1600	*missg.*	e	0.75	-	-	0.72
10	*greater than* 650	*high*	*missg.*	e	0.75	-	-	0.60
11a	*about* 350	*missg.*	*missg.*	e	0.70	i	0.10	0.50
11b	*about* 350	*high*	*missg.*	e	0.65	i	0.34	0.31

TABLE II
QUERYING MADE BY THE NEURAL NETWORK MODEL WHEN PRESENTED WITH A SAMPLE SET OF *PARTIAL* PATTERN VECTORS FOR VOWEL DATA

Serial	Input features			Query
No.	F_1	F_2	F_3	for
1a	700	*missing*	*missing*	F_2
1b	700	2300	*missing*	-
2a	*about* 350	*missing*	*missing*	F_2
2b	*about* 350	*high*	*missing*	-
3	400	800	*missing*	F_3
4	400	*missing*	*missing*	F_3
5	250	1550	*missing*	F_3

pattern point having relatively more uncertainty at all three frequency values. This results in the difficulty of decision as is evident from the value of the certainty measure. Besides, pattern class u (with a lower horizontal coverage *around* the *broader* band *about* 350) also does not figure among the top two significant responses. In entry $11b$, as F_2 becomes set at *high*, the response in favor of class i increases. However, the *ambiguity* in the decision is still evident.

In Table II we demonstrate a sample of the partial input feature combinations that are insufficient for inferring any particular decision. The *more essential* of the feature value(s) is queried for by (16, 17). The 3^{rd} and 5^{th} entries are seen to lack *essential* information in spite of having specific values corresponding to two features. This can be explained from the ambiguity of decision (w. r. t. a class) observed at these pattern points in the 2D projection in Fig. 4.

Table III shows the rules generated from the *knowledge base* by presenting a sample set of test patterns. The antecedent parts are obtained using (19–21) while the consequent parts are deduced from the values of the certainty measure bel_j^H. The rules obtained may be verified by comparing with Fig. 4. Note that the 5^{th}, 6^{th} and 9^{th} entries generate no justification.

B. Kala-azar Data

The model was next trained with the *kala-azar* data using 30 (20 diseased and 10 control/normal) cases. The test set consisted of 38 samples constituting the responses of the above-mentioned 20 diseased patients (over the next 20 days) to the ongoing treatment [22]. Some of these patients were cured while the conditions of a few others worsened, sometimes ultimately culminating in death. The instances of patients cured constituted the output class *normal/cured* while the remaining cases were clubbed under the output class *diseased*. The performance of various sizes of the proposed model on the *kala-azar* data with training set size $perc = 44.1 (= 38/68)$ is depicted in Table IV. Note that mean square error mse, *perfect match* p and *best match* b refer to the training set while mean square error mse_t and *overall score* t are indicative of the test set.

Then a trained three-layered neural network with 10 hidden nodes was used to demonstrate the inferencing ability (Tables V-VI) of the model on the *kala-azar* data. Table V shows the inferred output responses of the model for a sample set of test data. Here class 1 corresponds to *diseased* while class 2 refers to *cured*. The 1^{st} and 6^{th} entries correspond to patients experiencing speedy recovery during the course of treatment while the 2^{nd} entry refers to a patient who was gradually cured. The certainty measure and output membership values bear testimony to this. Note that the 1^{st} and 2^{nd} rows for each entry refer respectively to the status of the patient at the end of 10 and 20 days. The 3^{rd} and 4^{th} entries correspond

TABLE III
RULES GENERATED BY THE NEURAL NETWORK MODEL TO JUSTIFY ITS INFERRED DECISIONS FOR A SET OF PATTERN VECTORS FOR VOWEL DATA.

Serial No.	Input features F_1	F_2	F_3	Justification / Rule generation *If* clause	*Then* conclusion
1	300	900	*missing*	F_2 is *very low* and	
				F_1 is *very low*	*very likely* class *u*
2	250	1550	*unobtainable*	F_1 is *very low* and	
				F_2 is *Mol low*	*unable to recognize*
3	700	1000	2600	F_2 is *very low* and	
				F_1 is *Mol high* and	
				F_3 is *Mol high*	*very likely* class *a*
4	700	*unobtainable*	*missing*	F_1 is *Mol high*	*likely* class *a*
5	450	2400	*missing*	*no explanation*	-
6	700	2300	*missing*	*no explanation*	-
7	*high*	*Mol low*	*missing*	F_1 is *high* and	
				F_2 is *Mol low*	*very likely* class *a*
8	*between* 500 & 600	1600	*missing*	F_2 is *very medium* and	
				F_1 is *very medium*	*likely* class *e*
9	*greater than* 650	*high*	*missing*	*no explanation*	-
10	*about* 350	*high*	*missing*	F_2 is *high* and	
				F_1 is *very low*	*not unlikely* class *e*

TABLE IV
OUTPUT PERFORMANCE ON TRAINING AND TEST SET OF KALA-AZAR DATA BY THE FUZZY NEURAL NET MODEL FOR VARIOUS LAYERS $H+1$, WITH m NODES PER HIDDEN LAYER, USING $perc = 44.1$

Layers $H+1$	3		4
Nodes m	10	5	10
perfect p (%)	93.4	90.0	100.0
best b (%)	100.0	100.0	100.0
test t (%)	86.8	81.5	86.8
mse	0.002	0.004	0.001
mse_t	0.129	0.158	0.188

to patients who expired after 10 days of treatment. The 5^{th} and 7^{th} entries refer to patients whose conditions deteriorated during treatment. All these cases may be verified from the patient records listed in [22].

In Table VI we illustrate a few of the rules generated from the knowledge base. The serial nos. refer to the corresponding test cases reported in Table V. The antecedent and consequent parts are deduced as explained earlier.

C. Artificially Generated Data

Finally, the network was trained on the two sets of nonconvex pattern classes in succession. Two nonseparable pattern classes 1 and 2 were considered in each case. The region of *no pattern points* was modelled as class *none (no class)*. Table VII compares the performance of the three-layered fuzzy neural network model with that of the conventional MLP (*Vanilla MLP*), on the two sets of nonseparable patterns A, B, (depicted in Figs. 5–6 respectively) Training set size of $perc = 10$ was chosen from each representative pattern class. The number of hidden nodes used were $m = 11$ for Pattern Set A and $m = 13$ for Pattern Set B [16] for both the models. The *perfect match p*, *best match b* mean square error mse correspond to the training set while the remaining measures refer to the test set (classwise, corresponding to the three classes 1, 2, *none* and also overall, along with the mean square error mse_t).

In Tables VIII and X we demonstrate the inferred output responses of a five-layered model (with 10 nodes per hidden layer and trained with $perc = 50$) on some partial and complete input feature vectors for the two pattern sets. Tables IX and XI illustrate the generation of a few rules from the above-mentioned two *knowledge bases*. Verification regarding these tables may be made by examining the original patterns given in Figs. 5–6. The disjunctive *(Or)* terms in the antecedent parts are obtained by combining the various conjunctive clauses generated for the *same* feature corresponding to a single rule (produced to justify a single inferred decision). These disjunctive clauses result due to the concave and/or disjoint nature of the pattern class(es).

In Table VIII, the 1^{st}, 4^{th}, 5^{th} and 7^{th} entries correspond to horizontal bands across Fig. 5 showing Pattern Set A. Class *none*, having the largest horizontal coverage at $F_1 = low$ in entry 1, produces a *significant* response. Note that entry 4 (with $F_1 = medium$ and inferring class 1) and entry 5 (with $F_1 = Mol\ medium$ and inferring class *none*) denote *ambiguous* decisions as observed from the certainty measure. However entry 7 with $F_1 = Mol\ high$ produces a *more definite* response in favor of class 1. As F_2 becomes *known* as *low* in entry 2, the response changes from class 1 to class *none*. This is because of the fact that along the horizontal band at $F_1 = very\ low$, class 1 has the largest horizontal coverage. However when the smaller region of interest is specified at $F_2 = low$, the decision shifts in favor of class *none* and the *ambiguity* in decision decreases

TABLE V
INFERRED OUTPUT RESPONSES AND CERTAINTY MEASURES FOR A SET OF KALA-AZAR DATA

Serial No.	Input features				Highest output		Significant 2^{nd} choice	Certainty
	F_1	F_2	F_3	F_4	Class j	Membership y_j^H	Membership	bel_j^H
1	20.0	0.8	56.48	71.3	2	0.75	0.24	0.52
	22.5	0.87	61.21	60.5	2	0.91	-	0.83
2	26.0	0.9	51.45	76.6	1	0.50	0.49	0.01
	29.0	0.97	48.89	64.0	2	0.51	0.49	0.03
3	45.0	1.2	75.0	65.0	1	0.76	0.26	0.5
4	52.0	1.4	35.7	64.5	1	1.0	-	1.0
5	25.0	1.1	86.85	90.0	1	0.59	0.4	0.19
	27.0	1.3	117.27	89.3	1	1.0	-	1.0
6	18.0	0.83	78.8	65.5	2	0.75	0.25	0.5
	19.0	0.9	71.02	64.0	2	0.97	-	0.94
7	21.0	0.8	72.46	96.0	1	0.87	0.13	0.73
	30.0	1.1	96.4	85.0	1	1.0	-	1.0

TABLE VI
RULES GENERATED BY THE NEURAL NETWORK MODEL TO JUSTIFY ITS INFERRED DECISIONS FOR KALA-AZAR DATA

Serial No.	*If* clause	*Then* conclusion
1	F_3 is *very medium* and F_4 is *very low* and F_1 is *very low*	*more or less likely* cured
	F_3 is *very medium* and F_2 is *very medium* and F_1 is *Mol low*	*very likely* cured
2	F_4 is *very low* and F_2 is *very medium* and F_1 is *very medium* and F_3 is *low*	*unable to recognize*
	F_3 is *very medium* and F_2 is *Mol high*	*unable to recognize*
3	F_4 is *very low* and F_3 is *Mol high*	*more or less likely* diseased
4	F_4 is *very low* and F_3 is *very low*	*very likely* diseased
5	F_4 is *very medium* and F_1 is *very medium*	*not unlikely* diseased
	F_4 is *very medium* and F_1 is *Mol high*	*very likely* diseased
6	F_3 is *very medium* and F_2 is *very medium*	*more or less likely* cured
7	F_4 is *Mol high* and F_2 is *low*	*likely* diseased
	F_1 is *high* and F_4 is *Mol low*	*very likely* diseased

drastically as the certainty increases ($bel_j^H = 1$ here). In case of entries 6, 8 the corresponding responses in favor of classes 2 and *none* become *more certain* as F_2 becomes specified. All results of Tables VIII-IX may be verified by comparing with Fig. 5. Note that in Table IX, entries 2 and 4 generate no justification.

In Table X, entries 1, 2, 5 correspond to horizontal bands across Fig. 6 showing Pattern Set B. The 1^{st} and 5^{th} entries, for F_1 = *not low* and *very high* respectively, generate comparatively *less certain* decisions in favor of class 1. Entry 2 with F_1 = *medium* produces a *decisive* response in favor of class 2. As F_2 becomes *known* as *low* in entry 3, the response changes from class *none* to class 2 as the region of interest becomes more localized. But the *ambiguity* in decision is observed to be more in case of the complete input specification. All results of Tables X-XI may be verified by comparing with Fig. 6.

V. CONCLUSION AND DISCUSSION

In this work we considered a fuzzy neural net based expert system model. The *trained* neural network constituted the *knowledge base* for the application in hand. The network was capable of handling uncertainty and/or impreciseness in the input representation provided in quantitative, linguistic and/or set forms. The output decision was inferred in terms of membership values to one or more output classes. The user could be queried for the *more essential* feature information in case of partial inputs. Justification for the decision reached was generated in rule form. The antecedent and consequent parts of these rules were provided in linguistic and *natural* terms. The magnitudes of the connection weights of the *trained* neural net were used in every stage of the inferencing procedure. A measure of certainty expressing confidence (belief) in an output decision was also defined. The effectiveness of the algorithm was demonstrated on the vowel recognition problem, on some medical *kala-azar* data and on two sets of artificially generated nonconvex pattern classes.

Due to the limitations of the available medical data (on *kala-azar*), the proposed model could not be shown to sug-

TABLE VII
COMPARISON OF RECOGNITION SCORES OF THREE-LAYERED FUZZY NEURAL NET MODEL WITH THAT OF THE MORE CONVENTIONAL MLP, ON THE TWO NONSEPARABLE PATTERN SETS A, B

Pattern set		A		B	
Model		Fuzzy	Conventional	Fuzzy	Conventional
Test	1	78.6	47.7	83.9	87.1
	2	84.0	72.0	84.8	0.0
	none	84.9	82.0	59.5	51.6
	Overall	83.1	71.1	75.4	69.6
	mse_t	0.088	0.152	0.143	0.16
perfect p		62.1	1.2	32.2	8.1
best b		100.0	87.4	100.0	77.1
mse		0.007	0.078	0.008	0.104

TABLE VIII
INFERRED OUTPUT RESPONSES AND CERTAINTY MEASURES FOR A SAMPLE OF PATTERN SET A DATA, USING A FIVE-LAYERED FUZZY MLP HAVING $m = 10$ NODES IN EACH HIDDEN LAYER

Serial	Input features		Highest output		Significant 2^{nd} choice		Certainty
No.	F_1	F_2	Class j	Membership y_j^H	Class	Membership	bel_j^H
1	*low*	*missing*	*none*	1.0	-	-	1.0
2	*very low*	*missing*	1	0.60	*none*	0.41	0.19
	very low	*low*	*none*	1.0	-	-	1.0
3	*Mol low*	*low*	2	1.0	-	-	1.0
4	*medium*	*missing*	1	0.67	*none*	0.34	0.32
5	*Mol medium*	*missing*	*none*	0.72	1	0.32	0.4
6	*not medium*	*missing*	2	0.73	*none*	0.27	0.47
	not medium	*low*	2	1.0	-	-	1.0
7	*Mol high*	*missing*	1	1.0	-	-	1.0
8	*not high*	*missing*	*none*	0.8	1	0.2	0.62
	not high	*low*	*none*	1.0	2	0.01	0.99
9	*low*	*high*	*none*	1.0	-	-	1.0
10	*medium*	*medium*	*none*	1.0	-	-	1.0

TABLE IX
RULES GENERATED BY THE NEURAL NETWORK MODEL TO JUSTIFY ITS INFERRED DECISIONS FOR A SAMPLE OF INPUT VECTORS FOR PATTERN SET A DATA

Serial	Input features		Justification / Rule generation	
No.	F_1	F_2	*If* clause	*Then* conclusion
1	*Mol high*	*missing*	F_1 is *Mol high* or *very medium*	*very likely* class 1
2	*medium*	*missing*	*no explanation*	-
3	*medium*	*low*	F_1 is *medium* or *Mol high* and F_2 is *very medium*	*Mol likely* class 1
4	*Mol medium*	*missing*	*no explanation*	-
5	*medium*	*high*	F_1 is *medium* and F_2 is *high* or *very medium*	*very likely no class*
6	*high*	*medium*	F_2 is *medium* or *Mol high* and F_1 is *high*	*Mol likely no class*

TABLE X
INFERRED OUTPUT RESPONSES AND CERTAINTY MEASURES FOR A SAMPLE OF PATTERN SET B DATA, USING A FIVE-LAYERED FUZZY MLP HAVING $m = 10$ NODES IN EACH HIDDEN LAYER

Serial	Input features		Highest output		Significant 2^{nd} choice		Certainty
No.	F_1	F_2	Class j	Membership y_j^H	Class	Membership	bel_j^H
1	*not low*	*missing*	1	0.81	*none*	0.19	0.61
2	*medium*	*missing*	2	0.92	*none*	0.07	0.93
3	*Mol medium*	*missing*	*none*	0.97	2	0.07	0.90
	Mol medium	*low*	2	0.88	*none*	0.11	0.77
4	*not medium*	*low*	1	0.82	*none*	0.18	0.63
5	*very high*	*missing*	1	0.82	*none*	0.17	0.64
6	*medium*	*high*	2	1.0	-	-	1.0

TABLE XI
RULES GENERATED BY THE NEURAL NETWORK MODEL TO JUSTIFY ITS INFERRED DECISIONS FOR A SAMPLE OF INPUT VECTORS FOR PATTERN SET B DATA

Serial	Input features		Justification / Rule generation	
No.	F_1	F_2	*If* clause	*Then* conclusion
1	*very low*	*low*	F_1 is *very low* and	
			F_2 is *low* or *very medium*	*very likely no class*
2	*not low*	*missing*	F_1 is *very high*	*likely* class 1
3	*medium*	*missing*	F_1 is *medium* or *Mol low*	*very likely* class 2
4	*Mol medium*	*low*	F_1 is *Mol medium* or *Mol low* and	
			F_2 is *low*	*likely* class 2
5	*not medium*	*low*	F_2 is *low* or *very medium*	*likely* class 1
6	*high*	*low*	F_1 is *high* and	
			F_2 is *low*	*very likely no class*
7	*medium*	*high*	F_1 is *medium* or *Mol low* and	
			F_2 is *high*	*very likely* class 2

gest therapies and/or handle multiple diseases. However the suitability of the model in inferring correct decisions in the presence of overlapping disease categories may easily be gauged from its efficient handling of the fuzzy vowel data and the subsequent generation of appropriate justificatory rules. In the presence of suitable medical data, the therapies could be treated as output classes such that the certainty in favor of any such recommendation might be inferred. Any evaluation of the performance of the proposed model on the nonconvex Pattern Sets A and B should be made in the context of the *difficult* nature of the problem of class separability in these cases. This accounts for the relatively better performance of the model on the vowel data.

ACKNOWLEDGMENT

The authors gratefully acknowledge Mr. S. Chakraborty for drawing the diagrams. A part of this work was carried out when Ms. S. Mitra was supported by the German Academic Exchange Service Fellowship and Dr. Sankar K. Pal held the Jawaharlal Nehru Fellowship.

REFERENCES

[1] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoustics, Speech and Signal Processing Magazine*, 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] G. J. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information.* Reading, MA: Addison-Wesley, 1989.

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

[5] J. C. Bezdek and S. K. Pal, Eds., *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data.* New York: IEEE Press, 1992.

[6] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks.* Reading, MA: Addison-Wesley, 1989.

[7] *Proc. of Second IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, (California, USA), March 1993.

[8] *Proc. of Second International Conference on Fuzzy Logic and Neural Networks (IIZUKA 92)*, (Iizuka, Fukuoka, Japan), July 1992.

[9] H. J. Zimmermann, *Fuzzy Sets, Decision Making and Expert Systems.* Boston, MA: Kluwer Academic Publishers, 1987.

[10] K. Ng and B. Abramson, "Uncertainty management in expert systems," *IEEE Expert*, pp. 29–48, April 1990.

[11] S. I. Gallant, "Connectionist expert systems," *Communications of the ACM*, vol. 31, pp. 152–169, 1988.

[12] S. K. Pal and S. Mitra, "Multi-layer perceptron, fuzzy sets and classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 683–697, 1992.

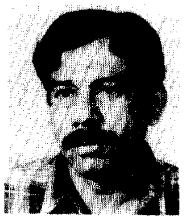
[13] F. Rosenblatt, *Principles of Neurodynamics, Perceptrons and the Theory of Brain Mechanisms.* Washington: Spartan, 1961.

[14] M. Minsky and S. Papert, *Perceptrons.* Cambridge, MA: MIT Press, 1969.

[15] S. K. Pal and D.P. Mandal, "Linguistic recognition system based on approximate reasoning," *Information Sciences*, vol. 61, pp. 135–161, 1992.
[16] S. K. Pal and S. Mitra, "Fuzzy versions of Kohonen's net and MLP-based classification: Performance evaluation for certain nonconvex decision regions," *Information Sciences*, vol. 76, pp. 297–337, 1994.
[17] S. K. Pal and P. K. Pramanik, "Fuzzy measures in determining seed points in clustering," *Pattern Recognition Letters*, vol. 4, pp. 159–164, 1986.
[18] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert systems," *Fuzzy Sets and Systems*, vol. 11, pp. 199–227, 1983.
[19] Y. Hayashi, J. J. Buckley and E. Czogala, "Approximation between fuzzy expert systems and neural networks," in *Proc. 2nd Intl. Conf. Fuzzy Logic and Neural Networks, Iizuka*, (Japan), pp. 135–139, 1992.
[20] E. Sanchez, "Fuzzy connectionist expert systems," in *Proc. 1990 Intl. Conf. Fuzzy Logic and Neural Networks, Iizuka*, (Japan), pp. 31–35, 1990.
[21] S. Mitra and S. K. Pal, "Neuro-fuzzy expert systems: overview with a case study," in *Fuzzy Reasoning in Information, Decision and Control Systems*, S. Tzafestas and A. N. Venetsanopoulos, Eds. Boston/Dordrecht: Kluwer Academic Publishers, pp. 121–143, 1994.
[22] D. K. Biswas, "Study of Kala-azar with special reference to renal function," Master's thesis, School of Tropical Medicine, University of Calcutta, Calcutta, India, 1989.

Sushmita Mitra completed her B.Sc. (Hons.) in physics at Calcutta University in 1984. She then obtained her B.Tech. and M.Tech. degrees in computer science from Calcutta University in 1987 and 1989, respectively. She was a Senior Research Fellow of the Council for Scientific and Industrial Research from 1989 to 1991, working for her Ph.D. degree.

She is a programmer in the Machine Intelligence Unit of the Indian Statistical Institute, Calcutta. Her research interests include pattern recognition, fuzzy sets, artificial intelligence, and neural networks. She was with ELITE, Korneliuscenter, Promenade 9, 52076 Aachen, Germany under the DAAD fellowship in 1993-1994.



Sankar K. Pal (M'81–SM'84–F'93) is a Professor and Founding Head of the Machine Intelligence Unit at the Indian Statistical Institute, Calcutta. He earned the B.Sc. (Hons.) in physics and the B.Tech., M.Tech., and Ph.D. in radiophysics and electronics from the University of Calcutta, India, in 1969, 1972, 1974, and 1979, respectively. In 1982, he received another Ph.D. in electrical engineering along with a DIC from Imperial College, University of London.

In 1986, he was awarded the Fulbright Post-Doctoral Visiting Fellowship to work at the University of California, Berkeley and the University of Maryland, College Park, MD, USA. In 1989, he received an NRC-NASA Senior Research Award to work at the NASA Johnson Space Center, Houston, TX, USA. He received the 1990 Shanti Swarup Bhatnagar Prize in engineering sciences (which is the most coveted and highest scientific award in India) for his contribution in pattern recognition. In 1993, he was awarded the Jawaharlal Nehru Fellowship, Vikram Sarabhai Research Award, and NASA Tech Brief Award. Along with S. Mitra, he received the 1994 Outstanding Paper Award from IEEE Neural Networks Council.

He served as a Professor-in-Charge of the Physical and Earth Sciences Division, Indian Statistical Institute during 1988–90. He was also a Guest Lecturer (1983–86) in computer science, Calcutta University. His research interests mainly include pattern recognition, image processing, artificial intelligence, neural nets, genetic algorithms, and fuzzy sets and systems. He is a co-author of the book *Fuzzy Mathematical Approach to Pattern Recognition* (New York: John Wiley & Sons [Halsted], 1986), 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* (New York: IEEE Press, 1992). He has to his credit more than one hundred and seventy-five research papers, including thirteen in edited books and more than a hundred in international journals. He is listed in *Reference Asia, Asia's Who's Who of Men and Women of Achievement*, and *Biography International*.

Dr. Pal is a Fellow of the IEEE, Indian National Science Academy, National Academy of Sciences, India, and the IETE. He is a Member of the Executive Advisory Editorial Board of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the *International Journal of Approximate Reasoning* (North Holland). He is an Associate Editor of the *Far-East Journal of Mathematical Sciences* and a Member of the Editorial Reviewing Board of the IEEE COMPUTER MAGAZINE, the *International Journal of Applied Intelligence*, and the *Mathematical Reviews*. He is 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).