ELSEVIER

FUZZY sets and systems

Fuzzy Sets and Systems 65 (1994) 285-296

Fuzzy MLP based expert system for medical diagnosis

Sushmita Mitra*

European Laboratory for Intelligent Techniques Engineering, Korneliuscenter, 52076 Aachen, Germany

Received December 1993; revised January 1994

Abstract

A fuzzy MLP model, developed by the author, is used as a connectionist expert system for diagnosing hepatobiliary disorders. It can handle uncertainty and/or impreciseness in the input as well as the output. The input to the network is modelled in terms of linguistic pi-sets whose centres and radii along each feature axis are determined from the distribution of the training data. In case of partial inputs, the model is capable of querying the user for the more important feature information, when required. Justification for an inferred decision may be produced in rule form. A comparative study of the performance of the model with other methods is also provided.

Key words: Fuzzy neural networks; Multilayer perceptron; Connectionist expert systems; Pattern classification; Inferencing; Rule generation

1. Introduction

Artificial neural networks [9, 17] are massively parallel interconnections of simple neurons that function as a collective system. They have been found to be proficient in solving various pattern recognition problems. Fuzzy sets [6, 19, 21], on the other hand, are capable of modelling uncertain or ambiguous data so often encountered in real life. Therefore, fuzzy neural networks [1, 7, 16] are designed to utilise a synthesis of the computational power of the neural networks along with the uncertainty handling capabilities of fuzzy logic. The multilayer perceptron (MLP) [17] is a feed-forward neural network model consisting of multiple layers of simple, sigmoid processing elements or neurons. A fuzzy version of the MLP (developed by the author [13, 14]) is used in this work for diagnosing hepatobiliary disorders.

An expert system [20] functions in a narrow domain dealing with specialized knowledge generally possessed by human experts. Traditional rule-based expert systems encode this information as If-Then rules while the connectionist expert system [2] uses the connection weights of a trained neural net model for this purpose. These are usually suitable in data-rich environments and avoid the time-consuming phase of knowledge base construction. Since fuzzy neural networks have been found to be better equipped in handling various forms of uncertainties generally associated with natural data,

* On leave from Machine Intelligence Unit, Indian Statistical Institute, 203, B. T. Road, Calcutta 700035, India. Direct all correspondence to this address.

0165-0114/94/$07.00
SSDI 0165-0114(94)00029-7

their use in designing fuzzy connectionist expert systems could alleviate several problems related to uncertainty or ambiguity.

The fuzzy MLP model [13, 14] has been used for classification and rule generation in [10] with some synthetic data consisting of fuzzy as well as linearly nonseparable, nonconvex and disjoint pattern classes. The model is extended here to automate the generation of the input description of the patterns (along each feature axis) at the input layer of the fuzzy neural network from the training data. The input vector (which can be in quantitative/linguistic/set forms) is represented in terms of the linguistic properties low, medium and high while the output decision is in terms of class membership values. The centres and radii of the pi-functions along each feature axis are determined automatically from the distribution of the training patterns. Initially, the fuzzy MLP is used for classifying the multi-class patterns.

Next, the trained network is used to generate rules. The connection weights in this stage constitute the knowlege base (in embedded form) for the classification problem under consideration. The model is then capable of inferring the output decision for complete and/or partial inputs along with a certainty measure and can query the user for the more essential missing input information. If asked by the user, it is capable of justifying its decision in If–Then rule form with the antecedent and consequent parts produced in linguistic and natural terms.

The effectiveness of the model as a neuro-fuzzy expert system is demonstrated on a set of medical data from the domain of hepatobiliary disorders. A comparative study is made with the classificatory performance of the fuzzy neural expert system by Hayashi [18, 4] and the more conventional statistical approach, viz., linear discriminant analysis.

2. The fuzzy MLP and rule generation

In this section we describe the fuzzy MLP model [13, 10] which is used for both classification and rule generation. Consider the layered network given in Figure 1. The output of a neuron in any layer other than the input layer is given as

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

where y_i^h is the state of the ith neuron in the preceding hth layer and w_{ji}^h is the weight of the connection from the ith neuron in layer h to the jth neuron in layer $h+1$. For nodes in the input layer, y_j^0 corresponds to the jth component of the input vector. The Least Mean Square error in output vectors is minimized by the backpropagation algorithm using gradient descent with a gradual decrease of the gain factor [13].

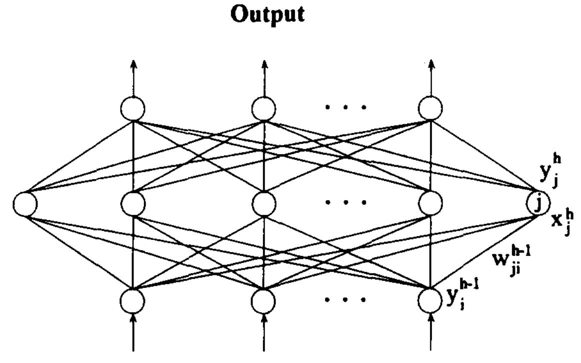

Fig. 1. The three-layered MLP model.

An n-dimensional pattern $\boldsymbol{F}_i = [F_{i1}, F_{i2}, \ldots, F_{in}]$ is represented as a $3n$-dimensional vector

$$\boldsymbol{F}_i = [\mu_{\text{low}(F_{i1})}(\boldsymbol{F}_i), \mu_{\text{medium}(F_{i1})}(\boldsymbol{F}_i), \mu_{\text{high}(F_{i1})}(\boldsymbol{F}_i), \ldots, \mu_{\text{high}(F_{in})}(\boldsymbol{F}_i)]$$
$$= [x_1^0, x_2^0, \ldots, x_{3n}^0] \tag{2}$$

where the μ values indicate the membership functions of the corresponding linguistic pi-sets along each feature axis.

For an l-class problem domain, the membership of the ith pattern in class k, lying in the range [0, 1], is defined as

$$\mu_k(\boldsymbol{F}_i) = \frac{1}{1 + (z_{ik}/F_d)^{F_e}} \tag{3}$$

where z_{ik} is the weighted distance between the ith pattern and the mean of the kth 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. Then, for the ith input pattern, the desired output of the jth output node is defined as

$$d_j = \mu_j(\boldsymbol{F}_i).$$

Note that according to this definition a pattern can simultaneously belong to more than one class. This is determined basically from the training set used during the learning phase.

Rules can be generated from the trained network. Figure 2 gives an overall view of the various steps involved in the process of inferencing and rule generation. The input for a test pattern can be in quantitative, linguistic or set forms or a combination of these. It is represented as memberships to the three primary linguistic properties low, medium and high as in (2), modelled as π-functions. The information can be in exact numerical form like F_j is r_1. It may be given as F_j is *prop*, where *prop* stands for any of the primary linguistic properties. The model can also handle the linguistic hedges *very, more or less* (*Mol*) and *not*. In case of set form usage, the input is a mixture of linguistic hedges and quantitative terms. 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 $k = (j-1)*3+1$. We use

$$\text{no information} \equiv \{0.5/L, 0.5/M, 0.5/H\} \tag{4}$$

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

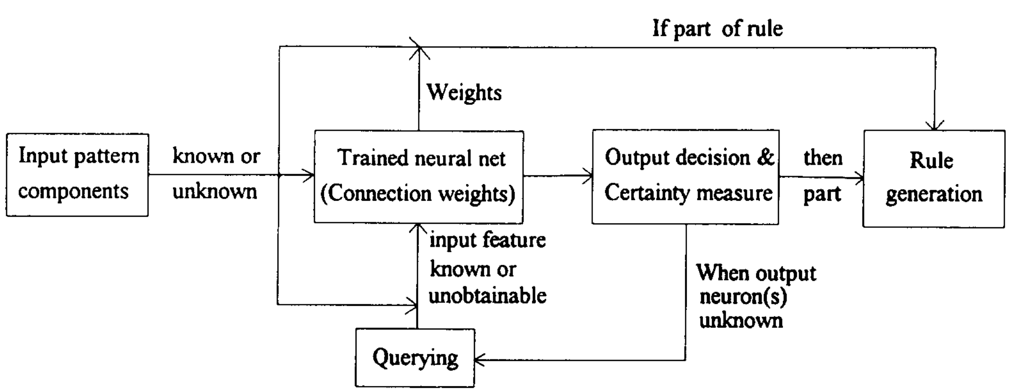

Fig. 2. Block diagram of the inferencing and rule generation phases of the fuzzy neural network.

2.1. Forward pass

Associated with each neuron j in layer h are its confidence estimation factor conf_j^h, a variable unknown_j^h providing a measure of the weighted information from the preceding ambiguous neurons i in layer $h-1$ (having $\text{noinf}_i^{h-1}=1$) and a variable known_j^h giving a measure of the weighted information from the remaining non-ambiguous preceding neurons (with $\text{noinf}_i^{h-1}=0$). For neuron j in layer $h>0$ we define

$$\text{unknown}_j^h = \sum_{\{i \mid \text{noinf}_i^{h-1}=1\}} w_{ji}^{h-1} y_i^{h-1}$$
$$\text{unden}_j^h = \sum_{\{i \mid \text{noinf}_i^{h-1}=1\}} |w_{ji}^{h-1}| \tag{5}$$

and

$$\text{known}_j^h = \sum_{\{i \mid \text{noinf}_i^{h-1}=0\}} w_{ji}^{h-1} y_i^{h-1} \tag{6}$$

where for neurons in layer $h>0$ we have

$$\text{noinf}_j^h = \begin{cases} 1 & \text{if } |\text{known}_j^h| \leqslant |\text{unknown}_j^h|, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Using (1), (5)–(7), we define

$$\text{conf}_j^h = \begin{cases} \left| \sum_i y_i^{h-1} w_{ji}^{h-1} / \text{unden}_j^h \right| & \text{if noinf}_j^h=1 \text{ and } h>0, \\ y_j^h & \text{otherwise.} \end{cases} \tag{8}$$

If there is no neuron j with $\text{noinf}_j^H = 1$, then the system finalises the decision inferred irrespective of whether the input information is complete or partial. A certainty measure (for each output neuron in output layer H) is defined as

$$\text{bel}_j^H = y_j^H - \sum_{i \neq j} y_j^H \tag{9}$$

where $\text{bel}_j^H \leqslant 1$. 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.

2.2. Querying

If there is any neuron j in the output layer H with $\text{noinf}_j^H = 1$ by (7), we begin the querying phase. We select the unknown output neuron j_1 from among the neurons with $\text{noinf}_j^H = 1$ such that $\text{conf}_{j_1}^H$ by (8) (among them) is maximum. Then 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. We select $i = i_1$ such that

$$|w_{j_1 i_1}^h * y_{i_1}^h| = \max_{\{i \mid \text{noinf}_i^{h-1}=1\}} |w_{j_1 i}^h * y_i^h|. \tag{10}$$

The process is repeated down to the input layer ($h=0$). For node i_1 in the input layer, the model queries the user for the value of the corresponding input feature u_1.

Note that if a missing input variable by (4) is queried for and found to be missing once again, we now tag it as unobtainable. 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 $\text{noinf}_{k_1}^0 = \text{noinf}_{k_1+1}^0 = \text{noinf}_{k_1+2}^0 = 0$, such that $k_1 = (u_1 - 1) * 3 + 1$. The response from an unobtainable input variable can allow the neuron activations in the following layers to become non-ambiguous with $\text{noinf}_j^h = 0$ such that an output decision may finally be inferred.

2.3. Justification

In this phase 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, in terms of the salient input features. Let the user ask for justification about a conclusion regarding class j. We choose neuron i from the hidden layer $H-1$ if $w_{ji}^{H-1} > 0$. Let a set of m_{H-1} neurons be so selected. For the remaining layers we obtain the maximum weighted paths through these neurons down to the input layer.

Let the set of the selected m_0 input neurons be given by $\{a_1^0, a_2^0, \ldots, a_{m_0}^0\}$ and their corresponding path weights (sum of the connection weights along the different layers) to neuron j in layer H be denoted as

$$\{\mathrm{wet}_{a_1^0}, \mathrm{wet}_{a_2^0}, \ldots, \mathrm{wet}_{a_{m_0}^0}\}.$$

We arrange these neurons in the decreasing order of their *net impacts*, where we define for neuron i

$$\text{net impact}_i = y_i^0 * \mathrm{wet}_{i^0}$$

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

$$\sum_{i_s} \mathrm{wet}_{i_s^0} > \sum_{i_n} \mathrm{wet}_{i_n^0} \tag{11}$$

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

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

$$\text{prop} = \begin{cases} \text{low} & \text{if } i_{s_1} - 3(u_{s_1} - 1) = 1, \\ \text{medium} & \text{if } i_{s_1} - 3(u_{s_1} - 1) = 2, \\ \text{high} & \text{otherwise.} \end{cases} \tag{12}$$

A linguistic hedge *very*, *more or less* or *not* may also be attached to the linguistic property obtained for the antecedent part. We use the mean square distance $d(u_{s_1}, \mathrm{pr}_m)$ between the 3-dimensional input values (components of the pattern vector) at the neurons corresponding to feature u_{s_1} and the linguistic property prop obtained from u_{s_1} by (12) represented as pr_m (with or without modifiers). The pr_m for which $d(u_{s_1}, \mathrm{pr}_m)$ is the minimum is selected as the antecedent clause corresponding to feature u_{s_1} (or neuron i_{s_1}). The procedure is repeated for all the input neurons selected by (11) to generate a set of antecedent clauses for the rule justifying the inference regarding output node j. All input features (of the test pattern) need not necessarily be selected for antecedent clause generation.

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 by (9). For the linguistic output form we use: *very likely* for $0.8 \le \mathrm{bel}_j^H \le 1$, *likely* for $0.6 \le \mathrm{bel}_j^H < 0.8$, *more or less likely* for $0.4 \le \mathrm{bel}_j^H < 0.6$, *not unlikely* for $0.1 \le \mathrm{bel}_j^H < 0.4$ and *unable to recognize* for $\mathrm{bel}_j^H < 0.1$.

3. Input vector representation

When the input feature is numerical, we use the π-fuzzy sets [15] (in the one-dimensional form), with range [0, 1], given as

$$\pi(F_j; c, \lambda) = \begin{cases} 2(1 - |F_j - c|/\lambda)^2, & \text{for } \tfrac{1}{2}\lambda \le |F_j - c| \le \lambda, \\ 1 - 2(|F_j - c|/\lambda)^2, & \text{for } 0 \le |F_j - c| \le \tfrac{1}{2}\lambda, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where $\lambda > 0$ is the radius of the π-function with c as the central point.

When the input feature is linguistic, its membership values for the π-sets low, medium and high are quantified from (13) as

$$\begin{aligned} \text{low} &\equiv \left\{ \frac{0.95}{L}, \frac{\pi(F_j(0.95/L); c_m, \lambda_m)}{M}, \frac{\pi(F_j(0.95/L); c_h, \lambda_h)}{H} \right\} \\ \text{medium} &\equiv \left\{ \frac{\pi(F_j(0.95/M); c_l, \lambda_l)}{L}, \frac{0.95}{M}, \frac{\pi(F_j(0.95/M); c_h, \lambda_h)}{H} \right\} \\ \text{high} &\equiv \left\{ \frac{\pi(F_j(0.95/H); c_l, \lambda_l)}{L}, \frac{\pi(F_j(0.95/H); c_m, \lambda_m)}{M}, \frac{0.95}{H} \right\} \end{aligned} \tag{14}$$

where c_l, λ_l, c_m, λ_m, c_h, λ_h refer to the centres and radii of the three linguistic properties and $F_j(0.95/L)$, $F_j(0.95/M)$, $F_j(0.95/H)$ refer to the corresponding feature values F_j at which the three linguistic properties attain membership values of 0.95.

3.1. Choice of parameters

Let $F_{j_{max}}$ and $F_{j_{min}}$ denote the upper and lower bounds of the dynamic range of feature F_j in all L pattern points, considering numerical values only. Let m_j be the mean of the pattern points along the jth axis. Then m_{j_l} and m_{j_h} are defined as the mean (along the jth axis) of the pattern points having co-ordinate values in the range $[F_{j_{min}}, m_j)$ and $(m_j, F_{j_{max}}]$ respectively. For the three linguistic property sets we define the centres as

$$c_{\text{medium}(F_j)} = m_j, \qquad c_{\text{low}(F_j)} = m_{j_l}, \qquad c_{\text{high}(F_j)} = m_{j_h}, \tag{15}$$

and the corresponding radii as

$$\begin{aligned} \lambda_{\text{low}(F_j)} &= 2(c_{\text{medium}(F_j)} - c_{\text{low}(F_j)}), \\ \lambda_{\text{high}(F_j)} &= 2(c_{\text{high}(F_j)} - c_{\text{medium}(F_j)}), \\ \lambda_{\text{medium}(F_j)} &= \text{fnos} * \frac{\lambda_{\text{low}(F_j)} * (F_{j_{max}} - c_{\text{medium}(F_j)}) + \lambda_{\text{high}(F_j)} * (c_{\text{medium}(F_j)} - F_{j_{min}})}{F_{j_{max}} - F_{j_{min}}} \end{aligned} \tag{16}$$

where fnos is a parameter controlling the extent of the overlapping. Note that this choice of parameters as well as the linguistic representation of features are different from that reported in [13, 10]. Here we take into account the distribution of the pattern points along each feature axis while choosing the corresponding centres and radii of the linguistic properties. This has been found to be more efficient in modelling skewed data distributions. Besides, the amount of overlap between the three linguistic properties can be different along the different axes, depending on the pattern set. We are also able to ensure that any feature value along the jth axis for pattern $\boldsymbol{F}_i$ is assigned membership value combinations in the corresponding 3-dimensional linguistic space of (2) in such a way that at least one of $\mu_{\text{low}(F_{ij})}(\boldsymbol{F}_i)$, $\mu_{\text{medium}(F_{ij})}(\boldsymbol{F}_i)$ or $\mu_{\text{high}(F_{ij})}(\boldsymbol{F}_i)$ is greater than 0.5. This allows a pattern $\boldsymbol{F}_i$ to have strong membership to at least one of the properties low, medium and high.

4. Implementation and results

The model was used on a set of 576 patient cases of various hepatobiliary disorders (already used by Hayashi et al. in [18, 4]). There were nine input features corresponding to 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 Haemoglobin (MCH, pg), Total Bilirubin (TBil, mg/dl) and Creatinine (CRTNN, mg/dl). These were represented in the 3n-dimensional linguistic form of (2). It is to be noted that the medical data under consideration had appreciably skewed distributions along most of the feature axes. The upper and lower bounds $F_{j_{max}}$, $F_{j_{min}}$ and the mean m_j along the jth axis for each of the nine features are indicated in Table 1. This sort of data distribution could be suitably handled by the choice of parameters given in (15)–(16) for the linguistic pi-sets used. The 10th feature corresponded to the sex of the patient and was represented in binary mode as (1, 0) or (0, 1). A diagrammatic representation of the data points along the 1st feature axis (corresponding to GOT) is depicted in Figure 3. The hepatobiliary disorders used for the four output classes were Alcoholic Liver Damage (ALD), Primary Hepatoma (PH), Liver Cirrhosis (LC) and Cholelithiasis (C). The network was trained using perc % samples from each representative pattern class of the data set. We selected $F_d = 5$, $F_e = 1$ in (3) and fnos = 1 in (16) after several experiments.

We used two measures of percent correct classification performance for the training set. The output, after a number of updating steps, was considered a perfect match p if the value of each output neuron y_j^H was within a margin of 0.1 from the desired membership value d_j. This was a stricter criterion than the best match b_1, where we tested whether the jth neuron output y_j^H had the maximum activation when the jth component d_j of the desired output vector also had the highest value. The factor b_2 corresponded to the performance of the model when one also considered the second best choice (i.e., the output neuron with second highest activation corresponded to the correct pattern class). Note that mse (mean square error), p, b_1, b_2 refer to the training set while mse$_t$, t_1 (best choice), t_2 (with second best choice) are indicative of the test set (remaining (100-perc)% samples). Here $b_2(t_2)$ corresponds to the sum of $b_1(t_1)$ with the score contributed by the neuron with the second highest activation. The individual classwise performance (with best choice) are also provided for the test set patterns for the four output classes.

During the rule generation phase, complete/partial sets of inputs were clamped at the input layer and the appropriate classification was inferred by the trained neural model. A measure of certainty was used and querying regarding unknown input feature values resorted to in case of some partial input sets. Justification in If–Then rule form, regarding a conclusion, could also be obtained when desired.

A comparison is provided with the results of Hayashi et al. [18, 4]. They used the Pocket algorithm for training the Distributed Single-layer Perceptron Network. The real-life fuzzy data were defuzzified using the Level Set representation to produce the crisp inputs $\{+1, -1, 0\}$ required by the algorithm. The various cut-off levels for receptor responses of each feature were set manually after consultation with domain experts. Generally 50 000 iterations were required for convergence. In our approach, the choice of the parameters for the linguistic features along each feature axis is automated, depending on the pattern set distribution. Besides, convergence is also achieved generally within 500 sweeps. The results of using the linear discriminant analysis on the same data were reported by Hayashi et al. in [18, 4].

Table 2 compares the classificatory performance of the fuzzy MLP based model (using three hidden layers having 40 nodes each) with those of Hayashi's model and the more conventional linear discriminant analysis method. All models used 70% of the pattern set as training data. Note that the classwise recognition score, using best choice, of our model is comparable to the scores, including the second best choice, of Hayashi's model. The overall superiority of our model in classifying the pattern sets can be easily observed.

In Table 3 we provide a study of the effect on the recognition score (%), using different numbers of hidden layers, nodes and training set size perc. The number of hidden nodes in each case corresponds to the network configuration (found experimentally) providing the best results with the given combination of number of layers and training set size. It is observed that better results are obtained in cases representing large training set size coupled with large network configuration (in terms of hidden layers and nodes). Small training set sizes usually resulted in poor generalization capabilities on the test set. However in such cases the performance deteriorated even further when an optimal number of

Table 1
The upper and lower bounds and the mean along each input feature axis for the data on hepatobiliary disorders

Feature	Unit	$F_{j_{min}}$	m_j	$F_{j_{max}}$
GOT	Karmen unit	8	113.0	4356
GPT	Karmen unit	3	54.5	1124
LDH	iu/l	179	476.3	6327
GGT	mu/ml	4	144.1	3075
BUN	mg/dl	3.3	17.2	91.0
MCV	fl	66.7	96.1	160.5
MCH	pg	20.3	32.1	52.5
TBil	mg/dl	0.1	3.2	37.0
CRTNN	mg/dl	0.4	1.1	4.3

hidden nodes were exceeded, but an increase in the number of hidden layers seemed to slightly enhance the efficiency of the model.

Table 4 depicts the rule generation and querying phases of the fuzzy MLP (trained using perc = 70 with 40 nodes in each of the three hidden layers) for a sample set of partially known input features. Columns 4 and 5 refer respectively to the input feature supplied by the user after querying and the resulting output membership value of the neuron corresponding to the hepatobiliary disorder supported by the Then part of the generated rule in column 7. The last column indicates the rules obtained from

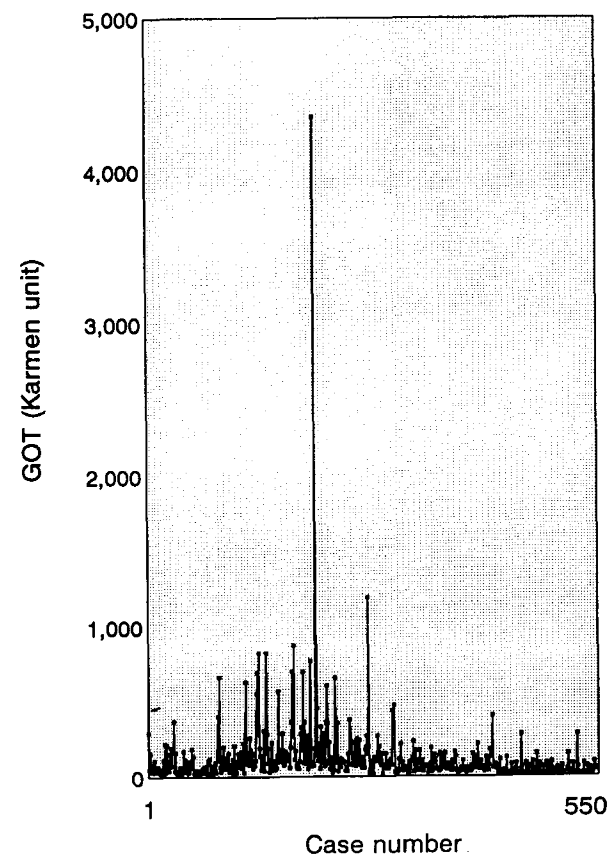

Fig. 3. Diagrammatic representation of the 576 data points for the feature Glutamic Oxalacetic Transaminate (GOT).

Table 2
Comparative study of the recognition scores (%) using the different models

Model	Linear discriminant analysis	Hayashi's Fuzzy neural network		Fuzzy MLP model	
		best choice	with 2nd best choice	best choice	with 2nd best choice
best match b_1	67.0	94.6	—	100.0	—
with 2nd best match b_2	—	—	100.0	—	100.0
perfect match p	—	—	—	97.4	—
ALD	57.6	54.5	69.7	65.7	88.6
PH	64.7	66.6	82.3	87.0	90.7
LC	65.7	40.0	71.4	65.7	89.4
C	63.6	59.1	81.8	80.5	86.1
best net score t_1	63.2	56.4	—	76.0	—
with 2nd best net score t_2	—	—	77.3	—	88.9

the initially supplied feature set in column 2 (from Hayashi, [22]). There were to types of such rules in [22], viz., the ones excluding a disease and the ones confirming a disease. In our model we resort to querying and further updating to obtain rules that are more specifically indicative of a disease. It may be mentioned that the input feature values of the sample test pattern as well as the connection weights learned during training contribute to the rule generation process. This helps extracting rules relevant to that region of the feature space which is local to the area pointed to by the feature values of the test pattern. Note that the certainty measure (and not the output membership value) is used in deciding whether querying should be resorted to at a particular stage and therefore querying is not required in all cases with partial set of input features (e.g., see row 1 of the table).

5. Conclusions and discussion

A neuro-fuzzy expert system has been described and its usefulness in diagnosing hepatobiliary disorders demonstrated. The model could handle uncertainty both at the input and the output. The input to the network was modelled in terms of the primary linguistic properties low, medium and high, using pi-functions. The centres and radii of these π-sets were automatically determined from the distribution of the training patterns. The model was capable of querying the user for the more important missing information, in case of partial inputs. Justification for an inferred decision could be produced in If–Then rule form. A comparative study with the methods of Hayashi et al. [18, 4] and the more conventional linear discriminant analysis was also provided.

It is to be noted that the connectionist expert system model by Gallant [2] uses crisp inputs/outputs

Table 3
Classification performance of the fuzzy MLP using different network configurations (with fnos = 1, $F_d = 5$, $F_e = 1$)

Layers	Hidden nodes	perc	b_1	b_2	p	mse	#sweeps	ALD	PH	LC	C	t_1	t_2	mse_t
3	20	10	98.1	98.1	56.9	0.006	350	31.4	44.7	56.2	80.3	52.3	71.3	0.112
	30	50	94.8	98.9	23.9	0.017	280	53.4	65.2	58.1	79.7	64.2	83.2	0.089
	10	70	79.0	88.7	9.2	0.056	200	42.8	81.4	55.2	83.3	67.4	80.9	0.075
4	15	10	100.0	100.0	98.1	0.0007	180	62.8	59.6	33.0	65.4	55.4	72.1	0.094
	15	50	96.6	98.5	26.9	0.012	370	56.9	77.5	56.4	89.8	70.9	83.2	0.08
	20	70	93.8	98.1	54.7	0.014	470	62.8	68.5	68.4	83.3	70.5	82.2	0.082
5	15	10	100.0	100.0	96.1	0.0006	210	44.7	55.2	46.4	68.2	53.8	75.2	0.109
	25	50	99.2	99.2	90.3	0.001	430	60.3	80.9	59.7	84.7	72.4	84.3	0.08
	40	70	100.0	100.0	97.4	0.0009	400	65.7	87.0	65.7	80.5	76.0	88.9	0.064

Table 4
Rule generation and querying phases with a sample set of partial input features

Sr No	Input features			Output memb. value	Rule generated		Rule from initial features
	Initially supplied	Initially unknown	Queried for		If parts	Then part	
1	GOT > 100 GPT < 40 LDH > 700 GGT < 60 15 < BUN < 20 30 < MCH < 36 male	MCV, TBil, CRTNN	—	0.77	GOT is medium GPT is low GGT is very low BUN is very medium	likely PH	ALD is completely false
2	MCV > 100 male	GOT, GPT, LDH, GGT, BUN, MCH, TBil, CRTNN	 GOT is very low GPT is very low MCH is Mol medium BUN is very low LDH is very low	0.0 0.1 0.07 0.26 0.74 1.0	GOT is very low GPT is very low LDH is very low MCV is very high MCH is Mol medium MCH is Mol high	very likely LC	PH is completely false
3	GOT < 40 female	GPT, LDH, GGT, BUN, MCV, MCH, TBil, CRTNN	 GPT is very low GGT is very low BUN is Mol medium LDH is low MCV is Mol medium	0.04 0.07 0.62 0.53 0.67 0.72	GOT is very low GPT is very low LDH is low GGT is very low BUN is Mol medium MCV is Mol medium	likely LC	PH is completely false
4	GOT < 40 40 < GPT < 100 MCV < 90 female	LDH, GGT, BUN, MCH, TBil, CRTNN	 MCH is Mol medium	0.75 0.8	GOT is very low MCV is very low MCH is Mol medium MCH is Mol low	likely C	PH is completely false
5	LDH > 700 MCV < 90 male	GOT, GPT, GGT, BUN, MCH, TBil, CRTNN	 GOT is not high GPT is very medium MCH is Mol medium GGT is not medium BUN is very medium	0.87 0.91 0.94 0.94 0.87 0.84	GOT is low GPT is Mol low BUN is very medium MCV is very low MCH is Mol medium	very likely PH	LC is completely false
6	BUN > 20 MCV > 100 male	GOT, GPT, LDH, GGT, MCH, TBil, CRTNN	 GOT is high GPT is high MCH is Mol medium	0.85 0.85 0.9 0.84	GOT is high BUN is Mol high MCV is high MCH is Mol medium MCH is Mol high	very likely PH	C is completely false
7	GOT < 40 40 < GPT < 100 male	LDH, GGT, BUN, MCV, MCH, TBil, CRTNN	 MCH is high CRTNN is low GGT is very high BUN is very low MCV is very high	0.01 0.59 0.77 0.88 0.78 0.89	GOT is very low BUN is very low MCV is very high MCH is high CRTNN is low CRTNN is very medium	very likely ALD	PH is completely false

and a linear discriminant network (with no hidden nodes) that is trained by the simple Pocket Algorithm. The absence of the hidden nodes and nonlinearity limits the utility of the system in modelling complex decision surfaces [9]. Dependency information regarding the variables, in the form of an adjacency matrix, are provided by the expert. Each variable (symptom, disease or treatment) corresponds to some node of the network. On the other hand, we use the fuzzy MLP-based model which is capable of automatically extracting the disease-symptoms dependency in the form of If–Then rules from the training data consisting of patient cases. These rules may be verified by the expert at a later stage or may be used to constitute the knowledge base of a traditional expert system.

Besides, medical information such as results of biochemical tests and/or the diagnosed disorder are often ambiguous and/or fuzzy [4]. Hence incorporation of fuzziness at input and output levels becomes more effective in modelling such problems. The skewness of the data set under consideration can be appropriately handled by the chosen input description that automatically determines the centres and radii of the linguistic pi-sets. It may be mentioned in this connection that other work on the choice of the appropriate input membership distribution for a fuzzy neural network for rule generation has been reported in [8, 3].

Fuzzy connectionist expert system models have also been designed by the author using logical operators based on And/Or functions [11, 12] in place of the sigmoid nonlinearities. However, it has been observed that the more conventional fuzzy version (reported here) performed more accurately. This result corroborates Keller's view in [5, p. 200]. It was found that the logical-operator-based version generated better rules for simpler problems.

Acknowledgements

I am grateful to Dr. Y. Hayashi of Ibaraki University, Japan for the data on hepatobiliary disorders and his personal communication regarding some of his results. I also thank Prof. Dr. Dr. h. c. H.-J. Zimmermann of RWTH, Aachen, for his interest in this study and his help in obtaining the data. I must also express my gratitude to the referees for their suggestions towards the improvement of this paper.

This work was supported by a German Academic Exchange Service (DAAD) Fellowship under Award No. 572/813-014-3.

References

[1] J.C. Bezdek and S.K. Pal, Eds., *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data* (IEEE Press, New York, 1992).
[2] S.I. Gallant, Connectionist expert systems, *Communications of the ACM* **31** (1988) 152–169.
[3] B. Grant and C. de Bruijn, Generation of fuzzy control rules and membership functions using neural nets, in: *Proceedings of 1st European Congerss on Fuzzy and Intelligent Technologies* (Aachen, 1993) 651–656.
[4] Y. Hayashi, Neural expert system using fuzzy teaching input and its application to medical diagnosis, in: *Proceedings of 2nd International Conference on Fuzzy Logic and Neural Networks* (Iizuka, 1992) 989–993.
[5] J.M. Keller, The inference process in fuzzy logic through artificial neural network structures, in: *Proceedings of 1st European Congress on Fuzzy and Intelligent Technologies* (Aachen, 1993) 195–201.
[6] G.J. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information* (Addison-Wesley, Reading, MA, 1989).
[7] B. Kosko, *Neural Networks and Fuzzy Systems* (Prentice Hall, Englewood Cliffs, NJ, 1991).
[8] R. Krishnapuram and F.C. Rhee, Compact fuzzy rule base generation methods for computer vision, in: *Proceedings of 2nd IEEE International Conference on Fuzzy Systems* (San Francisco, 1993) 809–814.
[9] R.P. Lippmann, An introduction to computing with neural nets, *IEEE Acoustics, Speech and Signal Processing Magazine* **61** (1987) 4–22.
[10] S. Mitra and S.K. Pal, Fuzzy multi-layer perceptron, inferencing and rule generation, *IEEE Transactions on Neural Networks* (in press).
[11] S. Mitra and S.K. Pal, Logical operation based fuzzy MLP for classification and rule generation, *Neural Networks* (in press).
[12] S. Mitra and S.K. Pal, Neuro-fuzzy expert systems: overview with a case study, in: S. Tzafestas and A.N. Venetsanopoulos, Eds., *Fuzzy Reasoning in Information, Decision and Control Systems* (Kluwer Academic Publishers, Boston, MA, in press).
[13] S.K. Pal and S. Mitra, Multi-layer perceptron, fuzzy sets and classification, *IEEE Transactions on Neural Networks* **3** (1992) 683–697.
[14] 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* (in press).
[15] S.K. Pal and P.K. Pramanik, Fuzzy measures in determining seed points in clustering, *Pattern Recognition Letters* **4** (1986) 159–164.
[16] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks* (Addison-Wesley, Reading, MA, 1989).

[17] D.E. Rumelhart and J.L. McClelland, Eds., *Parallel Distributed Processing* **1** (MIT, Cambridge, MA, 1986).
[18] K. Yoshida, Y. Hayashi, A. Imura and N. Shimada, Fuzzy neural expert system for diagnosing hepatobiliary disorders, in *Proceedings of 1990 International Conference on Fuzzy Logic and Neural Networks* (Iizuka, 1990) 539–543.
[19] L.A. Zadeh, Making computers think like people, *IEEE Spectrum* Aug. (1984) 26–32.
[20] H.-J. Zimmermann, *Fuzzy Sets, Decision Making and Expert Systems* (Kluwer Academic Publishers, Boston, MA, 1987).
[21] H.-J. Zimmermann, *Fuzzy Set Theory – and its Applications* (Kluwer Academic Publishers, Boston, MA, 1991).
[22] Y. Hayashi, Personal communication.