OCCLUDED OBJECT RECOGNITION BY NEURO-FUZZY APPROACH

by
L.N.NAGENDRA PRASAD

under the supervision of
PROF. KUMAR SANKAR RAY
Electronics & Communication Sciences Unit

a
DISSERTATION
submitted in the partial fulfillment for the degree of
Master of Technology
in
Computer Science

INDIAN STATISTICAL INSTITUTE
203, B.T ROAD.
CALCUTTA - 700035
1997

INDIAN STATISTICAL INSTITUTE
203, B.T ROAD.
CALCUTTA - 700035

Certificate of Approval

Certified that the dissertation titled "**Occluded Object Recognition By Neuro-Fuzzy Approach**" submitted by L.N.Nagendra Prasad, as partial fulfillment of the requirements for the degree of M.Tech(Computer Science) is a record of the work of the student, which has been carried out under my supervision.

(Prof. Kumar Sankar Ray)
ECSU ,ISI

Acknowledgements

I am happy to express my deepest sense of indebtness and gratitude to my guide Prof. Kumar Sankar Ray for his planning and timely guidance and regular supervision during the entire course of M.Tech dissertation.

I am thankful to Mr. Bijan Mukherjee, Mr.Tapan Kumar Dinda and Mr.Subrata Acharya for their affection,timely help and advice during my dissertation work.

I express a sense of gratitude to Head, ECSU for permitting me to avail the computer facilities for the execution of my project.

L.N.Nagendra Prasad

Contents

1 INTRODUCTION 1

1.1 NEURAL NETWORK AN OVERVIEW 1

1.2 COMPUTING WITH NEURAL NETWORKS 3

1.3 NEURO-FUZZY APPROACH 5

1.4 OCCLUDED OBJECT RECOGNITION : AN OVERVIEW 7

2 REVIEW OF FUZZY REASONING TECHNIQUES 10

3 MULTIDIMENSIONAL FUZZY REASONING (MFR) AND STATEMENT OF THE PROBLEM 15

4 REVIEW ON BACKPROPAGATION TYPE NEURAL NETWORK 19

5 IMPLEMENTATION OF NEW INTERPRETATION OF MULTIDIMENSIONAL FUZZY REASONING (MFR) AND BACK-PROPAGATION TYPE NETWORK 29

6 FORMULATION OF THE PROBLEM 31

6.1 LOCAL FEATURE EXTRACTION 31

6.2 PROCESS OF FUZZIFICATION 31

6.3 ASSIGNMENT OF THE MEMBERSHIP FUNCTION TO THE CONSEQUENT PART OF THE IF-THEN RULES 32

6.4 PROCESS OF DEFUZZIFICATION 35

6.5 GENERATION OF THE MODEL BASED OBJECT RECOGNITION SCHEME 35

7 CASE STUDY **39**

7.1 THE OUTPUT OF THE PROGRAM 41

8 CONCLUSION **48**

8.1 HARDWARE AND SOFTWARE REQUIREMENTS 48

1 INTRODUCTION

1.1 NEURAL NETWORK AN OVERVIEW

MOTIVATION

Since from the time the first primitive computing machines are invented, their designers and users are making efforts to push computers beyond the role of automatic calculators into the realm of "thinking machines" That, what is meant by thinking machine is debatable. The popular phrase "artificial intelligence" is given a variety of definitions. Hence methods for supposedly implementing humanlike thought process with a deterministic machine are varied. Neural networks represent one of these approaches.

BIOLOGICAL NEURAL NETWORKS

A single cell capable of a sort of crude computation is called a biological neuron. One or more inputs stimulate it and it generates an output that is sent to other neurons. The output is dependent on the strength each of the inputs and on the nature of the input connection (called a synapse). Some synapses may be such that an input there will tend to excite the neuron (increase the output).Rest may be inhibitory an input to them will tend to reduce the neuron's output. The actual relationship between the input and output is quite complex. Significant time delays may occur between the application of input stimulus and the generation of output response. Fatigue can set in, so that neuron does not always respond in the same way to same inputs. Even random events have an effect on the operation of the neuron. Fotunately, a large body of research indicates that simple models which account for only the most basic neural processes, can provide excellent solutions to practical problems.

NEURAL NETWORK CAPABILITIES

-Classification

Neural networks can be used to determine the crop types from satellite photographs, to distinguish a submarine from a boulder given its sonar return, and to identify the diseases of heart from electro cardiograms Any task that can be done by traditional discriminant analysis can be done atleast as well (and almost always much better) by neural network.

-Noise Reduction

An artificial neural network can be trained to recognise a number of objects. These objects may parts of time-series,images etc.. If a version of one of these objects, corrupted by noise , is presented to a properly trained neural network, the network can provide the original object on which it was trained. This technique has been used with great success in some image restoration problems.

-Prediction

A very common problem is that of predicting the value of a variable given the historic values of itself(and perhaps of other variables). It has been shown that the neural network, frequently outperform the traditional techniques like ARIMA and frequency domain analysis.

Artificial neural networks perform superior to other methods under the following conditions:

1. The data on which the conclusions are to be based is "fuzzy". If the input data is human opinions, ill defined categories or is subject to possibly large error then it is better to use artificial neural networks.

2. The objects important to the required decision are subtle or deeply hidden. One of the principal advantages of a neural network is its ability to discover objects in data which are so obscure as to be imperceptible to human researchers and standard statistical methods. One of the first major commercial uses of the neural networks was predicting the creditworthiness of loan applicants based on their spending and payment history. The correct decision depends on far more than simple factors like salary and debt level. Neural networks were shown to provide decisions superior to those made by trained humans.

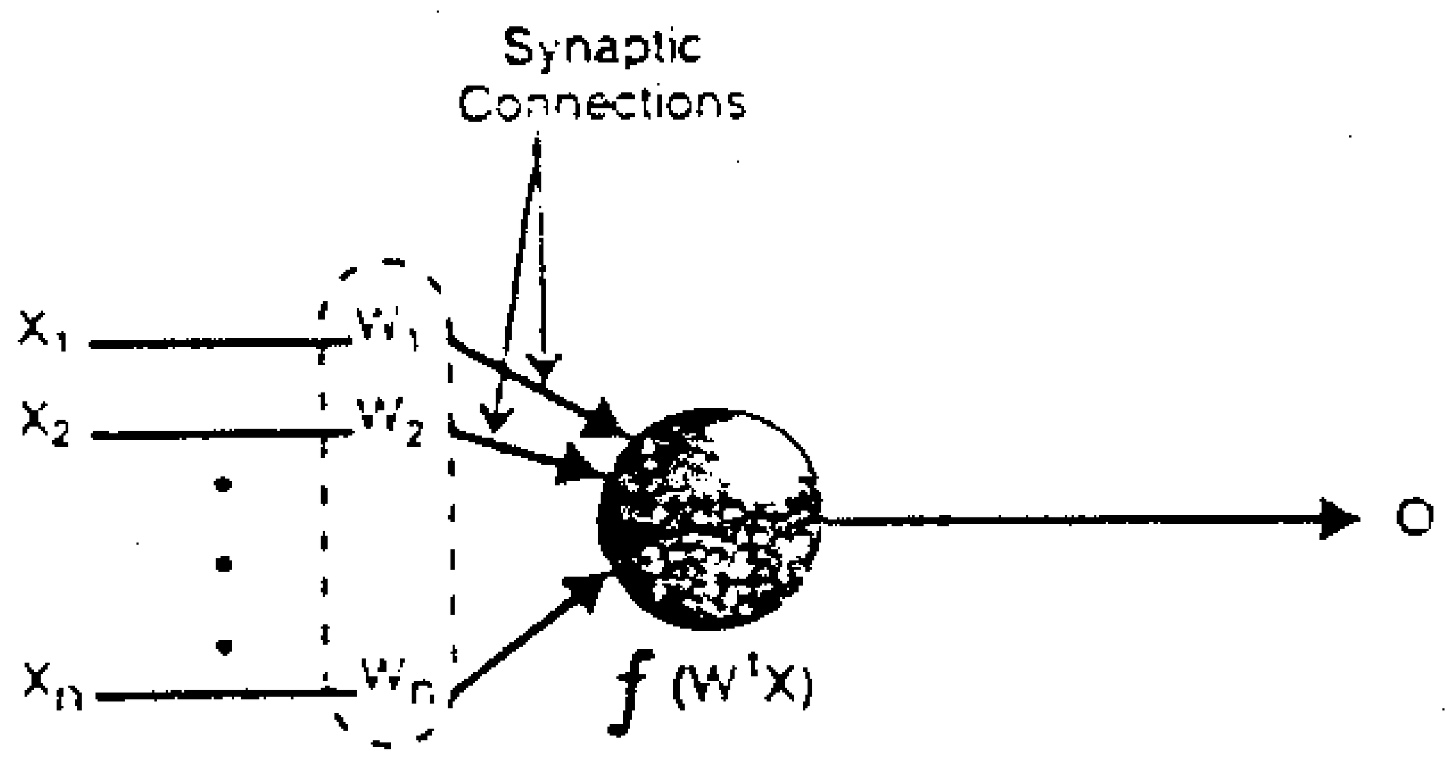
3. The data exhibits significant unpredictable nonlinearity.Traditional time series models for predicting future values, such as ARIMA and kalman filters, are based on strictly defined models. If the data does not fit the models, results will be useless. Neural networks marvelously adaptable.

4. The data is chaotic(in mathematical sense). Chaos can be found in telephone line noise, stock market prices and a host of other physical processes. Such behaviour is devastating to most other techniques, but neural networks are generally robust with inputs of this type. Many artificial neural networks both substantial theoritical foundations and practical utility. Any problem that can be solved with traditional modelling or statistical methods can be solved more effectively with neural network.

1.2 COMPUTING WITH NEURAL NETWORKS

The general structure of a neural network is as shown in fig.1. Neural net models explore many competing hypothesis simultaneously using massively parallel nets composed of computational elements connected by links with variable weights. Computational elements or nodes used in neural net models are nonlinear are typically analog and may be slow compared to modern electron digital circuitry. The simplest node sums N wifhted inputs and passes the result through a nonlinearity as shown in Fig.1. The node is charecterised by an internal threshold or offset Theta and by the type of nonlinearity Fig.2 illustrates three common types of nonlinearities; hard limiters, threshold logic elements and sigmoidal nonlinearities.More complex nodes may include temporal integration or other types of time dependencies and more complex mathematical operations than summation.

Neural net models are specified by the net topology, node charecteristics and training rules. These rules specify an initial set of weights and indicate how weights should be adapted during use to improve performance. Both design procedures and training rules are topic of current research. The potential benefits of neural nets extend beyond the high computation rates provided by massive parallelism. Neural nets typically provide a greater degree of robustness or fault tolerance than Von Neumann sequential computers because there are many more processing nodes, each with a primarily local connections. Damage to a few nodes or links thus need not impair overall performance significantly. Most neural net algorithms also adapt connection weights in time to improve performance based on current results. Application or learning is a major focus of neural net research. The ability to adapt and continue learning is essential in areas such as speech recognition where training data is limited and new talkers, new words, new dialects, new phrases and



$$O = f\,(W^tX) = f\left(\sum_{i=1}^{n} W_i X_i\right)$$

Figure 1: A general neuron capable of providing continuous-valued outputs

new environments are continuously encountered.

Adaptation also provides a degree of robustness by compensating for minor variables in characteristics of processing elements. Traditional statistical techniques are not adaptive but typically process all training data simultaneously before being used with new data. Neural net classifiers are also non parametric and make weaker assumptions concerning the shapes of underlying distributions than traditional statistical classifiers. They may prove to be more robust when distributions are generated by non linear processes and are strongly non-Gaussian. Designing artificial neural nets to solve problems and lead to new insights and algorithmic improvements.

Work on artificial neural net models has a long history. Development of mathematical models began more than 40 years ago with the work of McCulloch and Pitts,Hebb, Roenblatt, Widrow and others. More recent work by Hopfield, Rumelhart and McClelland, Sejnowski,Feldman,Grossberg and others has lead to a new resurgence of the field. This new interest is due to the development of new net topologies and algorithms, new analogy VLSI implementation techniques, and some intriguing demonstrations as well as

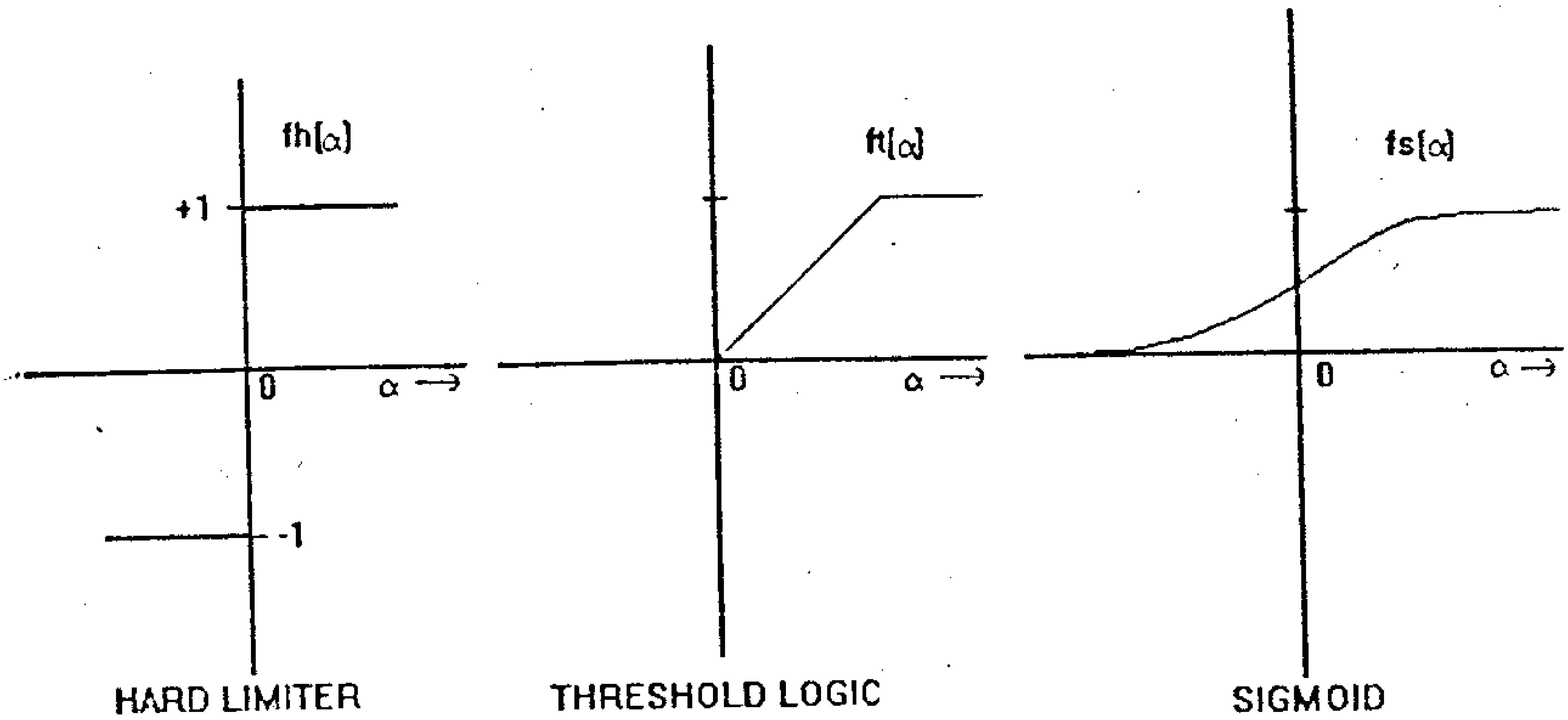


Figure 2: Three representative nonlinearities

by a growing fascination with the functioning like performance in areas of speech and image recognition will require enormous amounts of processing. Neural nets provide one technique for obtaining the required processing capacity using large number of simple processing elements operating in parallel. This paper provides an introduction to the field of neural nets by back propagation type neural models that can be used for occluded object classification.

1.3 NEURO-FUZZY APPROACH

Since Zadeh presented the compositional rule of inference,a large amount of literature is available on fuzzy reasoning and its applications(Zadeh , 1970; Fukami et al.,1990). However, there have been few discussions on fuzzy reasoning in the multidimensional case.(Tsukamoto,1979; Sugeno and Takagi, 1983). We give a new interpretation to the multi dimensional fuzzy reasioning (MFR). Subsequently we realise the new interpretation through back propogation type neural network. Off late, fuzzy reasoning and neural nets have independently received tremendous focus from the fields of applied sciences and engineering. Both can yield solutions for flexible information processing. Hence an nets have independently received tremendous focus from the fields of applied sciences and engineering. Both can yield solutions for flexible information processing. Hence an attempt has begun to generate fusion technology which will compensate the inherent demerits of one fields by the merits of the other. Recent results on fusion technology are reported in Takagi and Hayashi(1991) and Takagi et al.(1992). But these results show an implementation of the Sugeno and Kang model (1998) of approximate reasoning on a back propogation neural network. In course of developing such fusion technology the researchers that the existing fuzzy reasoning suffers from the following defeciencies.

1. Lack of a definite method to determine the membership function.

2. Lack of a learning function or adaptability which can be overcome by neural network driven fuzzy reasoning.

The aim of the present work is to generate an alternative fusion technology for occluded object recognition problems by realising the new interpretation of MFR through a back propagation type neural network. In doing this we state the following problems of the existing MFR.(Sugeno and Takagi, 1983):

1. Lack of a definite method to determine the appropriate translating rule for the law of implication.

2. Difficulty in handling the multi dimensional fuzzy relation.

3. Lack of a definite method to determine the appropriate compositional rule.

4. Computational complexity in applying any compositional rule over a multidimensional fuzzy relation.

5. Lack of learning function or adaptability.

In this connection, instead of questioning the lack of a definite method to determine the membership function of a fuzzy set we provide the following argument. According to Zadeh (1970) membership is the quantification of the human perception about the situation at hand. Normally mem- berships are assigned to the linguistic, by experienced operator/ domain expert. Hence it is quite natural to assume that representation of such perception through membership assignment is quite reasonable to deal with. For instance if we have a finite universe of discourse, say $X = \{1, 2, 3, 4\}$ and if state X is "BIG" then,

$$\mu_x = 0.2/1 + 0.5/2 + 0.7/3 + 1/4$$

where μ_x is the membership function of X.

Now alternatively we may have

$$\mu_x = 0.1/1 + 0.4/2 + 0.6/3 + 0.9/1,$$

etc. But we will never represent "BIG" X as the following

$$\mu_x = 0.1/1 + 0.7/2 + 0.5/3 + 0.2/4$$

which is a representation of a wrong perception about a particular situation. Thus the quantification human perception through the assignment of membership / membership function may vary (depending upon the level of the perception of the individual) within tolerable limit but can never be widely different. Sometimes to fit the situations more accurately we may need tune the memberhip function further through some heuristic means by tuning the shapes of the membership function. Therefore whatever may be methods of determining the membership function, we can successfully utilise them in many application domains of fuzzy reasoning, especially the fuzzy logic controller (Lee, 1990). But sometimes a large dimension of a fuzzy relational matrix, the lack of choice of appropriate translating rule and compositional rule of inference affect the success of fuzzy reasoning to great extent. Although there are thumb rules (Lee 1990) to select them (translating rule and compositional rule) , they are not well supported by some theories. Hence in the present work we try to avoid these two rules for fuzzy reasoning and try to introduce the feature of adaptability in fuzzy reasoning. We achieve these goals by realising the new interpretation of MFR through back propogation type neural networks

1.4 OCCLUDED OBJECT RECOGNITION : AN OVERVIEW

When objects are occluded, many shape recognition methods that use global information fails. To recognise the partially occluded objects we represent each object by a set of "landmarks". Shape recognition is an important task in object recognition. We will use the term shape to refer to the invariant geometrical properties of relative distances among a set of static spatial features of the object. These static spatial features are known as the shape features of the object. A shape feature is classified as either a global or a local representation. A global shape feature represents the entire object region, such as silhoutte or contour of the object , local shape features represent shape measure, quantifies the similarity between the portions of the objects. A global shape measure is derived from the global shape features of the objects ; a local shape measure is derived from the local shape features

For the purpose of recognition much of the visual data perceived by the human being is highly redundant. It has been suggested from the view point of the human visual system some dominant points along the object contour are rich in information content and are sufficient to charecterise the shape of the object. These dominant points of the objects are usually referred to as landmarks of the object. However, we will define the landmarks of an object as the points of interest of the object that have important shape attributes. Examples of landmarks are corners, holes, protrusions and high curvature points. They can be problem specific based on apriori knowledge. For example in medical imaging landmarks could be the location of important bone joints. Landmarks could be the location of important bone joints. Landmark based shape recognition is motivated by the above concept of dominant points.It uses landmarks as shape features to recognise objects in a scene. One of the merits of the landmark based shape recognition is that the extraction of the entire object contour is not required to acheive recognition. It only requires a landmark extractor that can detect and order the landmarks in a sequence that corresponds to consecutive points along the object boundary.

A landmark based object recognition system is shown in Fig.1.4 Landmarks extracted from the model object and from the scene are referred to as model landmarks and scene landmarks respectively. Properties of the landmarks of each model can be used to guide the extraction of the landmarks in the scene. The hypothesis of a model object in the

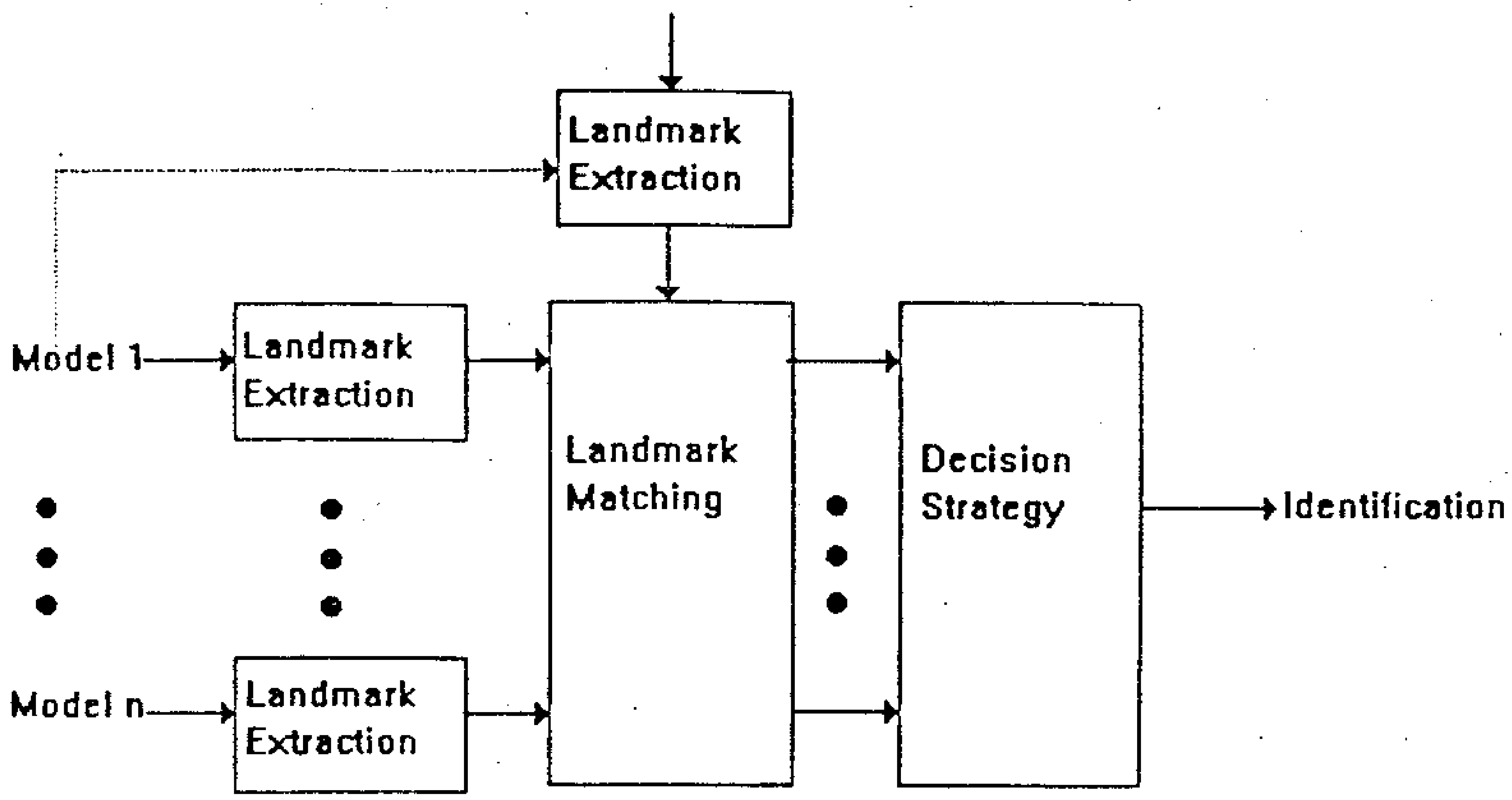


Figure 3: A Landmark based shape recognition system

scene is made by matching the model landmarks with the scene landmarks. Based on a decision strategy on how well the landmarks of each model are matched with those in the scene objects in the scene are identified.

2 REVIEW OF FUZZY REASONING TECHNIQUES

There are many successful applications of the method of approximate reasoning to design of the fuzzy logic controllers in different fields. But there is no systematic study about the application of the method of approximate reasoning to the object recognition problems. Here first of all the method of approximate reasoning under the law of fuzzy im plications and max-min compositional rule of inference to classify a set of synthetic data and subsequently compare the performance of the method of approximate reasoning.

We shall consider the following form of inference[1]

$$\left.\begin{array}{lll} \text{Premise 1 :} & \text{If } X \text{ is } A \text{ then} & Y \text{ is } B \\ \text{Premise 2 :} & X \text{ is } A' & \\ \hline \text{Consequence :} & & Y \text{ is } B' \end{array}\right\} \tag{1}$$

where A, A' are fuzzy sets in U and B, B' are fuzzy sets in V.The consequence B' is deduced from premise 1 and premise 2 by taking the max-min composition of the fuzzy set A' and the fuzzy relation $A \to B$ obtained from the fuzzy implication "if A then B". That means, we get,

$$B' = A' \circ (A \to B),$$
$$\mu_{B'}(v) = \bigvee_u \{\mu_{A'}(u) \wedge \mu_{A \to B}(u, v)\}$$

If the fuzzy set A' is a singleton u_0, that is, $\mu_{A'}(u_0) = 1$ and $\mu_{A'}(u) = 0$ for $u \neq u_0$, the consequence B' is simplified as,

$$\left.\begin{array}{rcl} \mu_{B'}(u) & = & \vee_u\{\mu_{A'}(u) \wedge \mu_{A \to B}(u, v)\} \\ & = & \vee_{u(\neq u_0)}\{0 \wedge \mu_{A \to B}(u, v)\} \vee \{1 \wedge \mu_{A \to B}(u_0, v)\} \\ & = & \mu_{A \to B}(u_0, v) \end{array}\right\} \tag{2}$$

If the fuzzy implication $A \to B$ isrepresented by the direct product $A \times B$ of fuzzy sets A and B as in case of Mandami's method, B' is given as, $\mu_B'(v) = \mu_A(u_0) \wedge \mu_B(v)$ at $A \to B = A \times B$. In the Table 1, several fuzzy implications $A \to B$ are listed which can be used in the approximate reasoning approach to object classification. Now we consider the follow- ing form of inference in which a fuzzy conditional proposition "if... then..." contains two fuzzy compositions "$X$is$A$" and "$Y$is$B$" combined using the connective "and".

$$\left.\begin{array}{l} \text{Premise 1 : If } X \text{ is } A \text{ and } Y \text{ is } B \text{ then } Z \text{ is } B \\ \text{Premise 2 : If } X \text{ is } A' \\ \hline \text{Consequence : } Z \text{ is } C' \end{array}\right\} \qquad (3)$$

where A, A' are fuzzy sets in U and B, B' are fuzzy sets in V and C, C' are fuzzy sets in W.

The consequence C' can be deduced from Premise 1 and Premise 2 by taking the **max-min** composition $\circ$ of a fuzzy set(A' and B') in $U \times V$ and a fuzzy relation (A and B) $\to C$ in $U \times V \times W$. That means, we get

$$\begin{array}{rcl} C' & = & (A' \text{ and } B') \circ [(A \text{ and } B) \to C] \\ \mu_{C'}(w) & = & \vee_{u,v}\{[\mu_{A'}(u) \wedge \mu_{B'}(v)] \wedge [(\mu_A(u) \wedge \mu_B(v)) \to \mu_C(w)]\} \end{array} \qquad (4)$$

In the case of Mandani's method R_c in Table 1, the fuzzy implication $[(A \text{ and} B) \to C]$ is translated into $\mu_A(u) \wedge \mu_B(v) \wedge \mu_C(w)$ by virtue of $a \to b = a \wedge b$. Thus the consequence C' is given by

$$\mu_{C'}(w) = \bigvee_{u,v}\{[\mu_{A'}(u) \wedge \mu_{B'}(v)] \wedge [(\mu_A(u) \wedge \mu_B(v)) \wedge \mu_C(w)]\} \qquad (5)$$

Let $R_c(A, B, C) = (A \text{ and } B) \to C, R_c(A; C) = A \to C$ and $R_c(B; C) = B \to C$ be fuzzy implications by Mamdani's method R_c. Then the consequenc C' of equation (5) reduces to

$$\mu_{C'}(w) = [\bigvee_u\{\mu_{A'}(u) \wedge \mu_A(u) \wedge \mu_C(w)\}] \wedge [\bigvee_v\{\mu_{B'}(v) \wedge \mu_B(v) \wedge \mu_C(w)\}]$$

$$= \bigvee_u \mu_{A'}(u) \wedge \mu_A(u) \wedge \mu_C(w) \wedge \mu_{B' \circ R_c(B;C)}(w)$$

$$= \mu_{A' \circ R_c(A;C)}(w) \wedge \mu_{B' \circ R_c(B;C)}(w)$$

Therefore the consequence $C' = (A' \text{ and } B') \circ R_c(A, B; C)$ can be obtained as the intersection of $A' \circ R_c(A; C)$ and $B' \circ R_c(B; C)$ for the Mamdani's implication R_c. That means we get,

$$C' = (A' \text{ and } B') \circ R_c(A, B; C)$$
$$= [A' \circ R_c(A; C)] \cap [B' \circ R_c(B; C)]$$

Similarly we can have,

$$(A' \text{ and } B') \circ [(A \text{ and } B) \to C] = [A' \circ (A \to C)] \cap [B' \circ (B \to C)] \tag{6}$$

for the fuzzy implications R_p, R_{pp} and R_{dp} in Table 1. Note that R_a, R_b, R^*, R_s, R_g and R_Δ in Table 1, for which the equality $(a \wedge b) \to c = (a \to b) \wedge (b \to c)$ holds, satisfy the following,

$$(A' \text{ and } B') \circ [(A \text{ and } B) \to C] = [A' \circ (A \to C)] \cup [B' \circ (B \to C)] \tag{7}$$

If the fuzzy sets A' and B' are singletons in (3),i.e., $A' = u_0$ and $B' = v_0$, the consequence C' of (4) is represented as,

$$\mu'_C(w) = \bigvee_{(u \neq u_0) \text{ or } (v \neq v_0)} \{0 \wedge [(\mu_A(u) \wedge \mu_B(v) \to \mu_C(w)]\} \vee \{1 \wedge [(\mu_A(u_0) \wedge \mu_B(v_0)) \to \mu_C(w)]\}$$

$$= [\mu_A(u_0) \wedge \mu_B(v_0)] \to \mu_C(w)$$

For examplein the case of R_c and R_a, we have consequences $C' at A' = u_0$ and $B' = v_0$ as follows,

$$R_c : [\mu_A(u_0) \wedge \mu_B(v_0)] \wedge \mu_C(w_0),$$
$$R_a : 1 \wedge [1 - (\mu_A(u_0) \wedge \mu_B(v_0) + \mu_c(w)]$$

Similar results can be obtained from other fuzzy implications in Table 1 In the above discussion, the operation $\wedge (= min)$ is used as the meaning of "and". It is possible to introduce other operations, say, the algebraic product "*" and more generally t-norms as "and".

As a generalised form of approximate reasoning we shall consider approx- mate reasoning with fuzzy conditional proposition combined with "else".

$$\left.\begin{array}{l}\text{Premise 1 : If } X \text{ is } A_1 \text{ and } Y \text{ is } B_1 \text{ then } Z \text{ is } C_1 \text{ else}\\ \text{Premise 2 : If } X \text{ is } A_2 \text{ and } Y \text{ is } B_2 \text{ then } Z \text{ is } C_2 \text{ else}\\ \ldots\\ \ldots\\ \text{Premise n : If } X \text{ is } A_n \text{ and } Y \text{ is } B_n \text{ then } Z \text{ is } C_n \text{ else}\\ \text{Premise n+1 : If } X \text{ is } A_{n+1} \text{ and } Y \text{ is } B_{n+1} \text{ then } Z \text{ is } C_{n+1} \text{ else}\\ \hline \text{Consequence : } ZisC'\end{array}\right\} \quad (8)$$

If we interpret "else" as union ($\bigcup$) which is valid for fuzzy implications R_c, R_p, R_{bp}, R_{dp} in Table 1, we can deduce the consequences C' (refer equation (6)) as

$$\begin{aligned}C' &= (A' \text{ and } B') \circ [(A_1 \text{ and } B_1) \to C_1) \cup \ldots\\ &= [(A' \circ A_1 \to C_1) \cap (B' \circ B_1 \to C_1)] \cup \ldots\\ &\quad \cup[(A' \circ A_n \to C_n) \cap (B' \circ B_n \to C_n)]\\ &= C_1{}' \cup C_2{}' \cup \ldots \cup C_n{}'\end{aligned}$$

whereas for the fuzzy implications R_a, R_m, R_b, R^* and R_Δ in table 1, "else" in (8) is represented as intersection ($\cap$). Thus the consequenc C' for these fuzzy implications are defined as,

$$C' = (A' \text{ and } B') \circ [((A_1 \text{ and } B_1) \to C_1) \cap \ldots \cap ((A_n \text{ and } B_n) \to C_n)]$$

$$\subseteq [(A' \circ A_1 \to C_1) \cup (B' \circ B_1 \to)] \cap \ldots \cap [(A' \circ A_n \to C_n) \cup (B \circ B_n \to C_n)]$$

It is noted that the consequence C' is not equal to but contained in the intersection of fuzzy inference results $[(A' \circ A_i \to C_i) \cup (B' \circ B_i \to C_i)], \forall i$. However, for simplicity of calculation C' will be represented as
$C' = C_1{}' \cap C_2{}' \cap \ldots \cap C_n{}'$

R_c	$: \mu_A(u_0) \wedge \mu_B(v)$	Mamdani
R_p	$: \mu_a(u_0) \bullet \mu_B(v)$	Larsen
R_{bp}	$: 0 \vee [\mu_A(u_0) + \mu_B(v) - 1]$	bounded product
R_{dp}	$: \begin{cases} \mu_A(u_0), & \mu_B(v) = 1 \\ \mu_B(v), & \mu_A(u_0) = 1 \\ 0, & \mu_A(u_0),\ \mu_B(v) < 1 \end{cases}$	drastic product
R_a	$: 1 \wedge [1 - \mu_A(u_0) + \mu_B(v)]$	Zadeh's arithmetic rule
R_m	$: [\mu_A(u_0) \wedge \mu_B(v)] \vee [1 - \mu_A(u_0)]$	Zadeh's maximum rule
R_b	$: [1 - \mu_A(u_0)] \vee \mu_B(v)$	Boolean implication
R_s	$: \{ 1, \ \mu_A(u_0) \preceq \mu_B(v)$	Standard sequence
R_g	$: \begin{cases} 1, & \mu_A(u_0) \preceq \mu_B(v) \\ \mu_B(v), & \mu_A(u_0) > \mu_B(v) \end{cases}$	Gödelian logic
R_Δ	$: \begin{cases} 1, & \mu_A(u_0) \preceq \mu_B(v) \\ \mu_B(v)/\mu_A(u_0), & \mu_A(u_0) > \mu_B(v) \end{cases}$	Gougen logic
$R^{\bullet}$	$: 1 - \mu_A(u_0)] + \mu_A(u_0) \bullet \mu_B(v)$ $[1 - \mu_A(u_0) \vee \mu_B(v)]$	Bandler logic
$R_{\#}$	$: \wedge [\mu_A(u_0) \ \vee \ 1 - \mu_A(u_0)]$ $\wedge [\mu_B(v) \ \vee \ 1 - \mu_B(v)]$	Bandler logic

Table 1: Fuzzy implications $\mu_{A \to B}(u_0, v) = \mu_A(u_0) \to \mu_B(v)$

3 MULTIDIMENSIONAL FUZZY REASONING (MFR) AND STATEMENT OF THE PROBLEM

If we have a multidimensional fuzzy implication (MFI) such as "if (x is A , y is B) then z is C" where A, B and C are fuzzy sets, we can have the following interpretation taken in the multidimensional case. For example in the compositional rule of inference, the above two dimensional implication can be translated into

if x is A and y is B then z is C

or

if x is A then if y is B then z is C

According to Tsukamato(1979) the above multidimensional fuzzy implication can be represented as

$$\begin{array}{l} if \quad \mathrm{x} \quad is \quad \mathrm{A} \quad then \quad \mathrm{z} \quad is \quad \mathrm{C} \\ and \\ if \quad \mathrm{y} \quad is \quad \mathrm{B} \quad then \quad \mathrm{z} \quad is \quad \mathrm{C} \end{array} \tag{9}$$

and the intersection $C' \cap C''$, where C' is the inferred value from the first implication and C'' is the inferred value from the second implication is taken for the consequence of reasoning.

To tackle the object recognition problems using MFR, the following new interpretation of the multidimensional fuzzy implication is provided.

$$\left.\begin{array}{c} \text{if } x \text{ is } A \text{ then } z \text{ is } C_1 \\ \text{and} \\ \text{if } y \text{ is } B \text{ then } z \text{ is } C_2 \end{array}\right\} \tag{10}$$

where $C = C_1 \cap C_2$ and the intersection $C_1' \cap C_2'$ where C_1' is the inferred value from the first implication and C_2' is the inferred value from the second implication, is taken as the consequence of reasoning.

The essential difference between the Tsukamoto model and the newly proposed model occurs at the interpretation of the consequent part of each of the decomposed fuzzy implicationd (DFI) of the multidimensional fuzzy implication (MFI). According to Tsukamato, the consequent parts parts of the DFIs of an MFI are same as the consequent part of MFI. Whereas in the newly proposed model the consequent part of the MFI is the intersection of the consequent parts of the DFIs. Thus the linguistic connective "and" has more meaningful logical interpretation than of the "$\cap$".

For object recognition, there are basically two approaches, namely the decision theoritic and the syntactic. Since the MFR approach to the object recognition is similar to the decision theoritic method of object recognition, the basic concept of the decision theoritic approach to object recognition will briefly described first and then we try to establish the similarity between the decision theoritic approach and the MFR approach.

Under decision theoritic approach, each pattern is represented by a vector of features. The feature space is divided into a number of regions, each of which represents a prototype pattern or a cluster of patterns. A decision function maps the given pattern to previously determined regions.

In the MFR approach to pattern recognition and occluded object recognition , each element of the (Local) feature vector is represented by the fuzzy linguistic variable instead of a real number. For instance,suppose we have a (2×1) feature vector

$F = (F_1, F_2)^T$,

where T represents transpose, F_1 is the internal angle at the significant point on the boundary of the object and F_2 is the curvature at that significant point. In the decision theoritic approach to pattern recognition ,F_1 and F_2 are represented by real numbers, where as in MFR approach to the occluded object recognition F_1 and F_2 are local features represented by the fuzzy linguistic variables, F_1 is small and F_2 is medium. The elements of the feature vector (F_f) which are represented by fuzzy linguistic variables are characterised by their membership functions. Thus in the present approach, instead of a single pattern, a population of patterns is represented by the feature vector F(see Fig. 4) The elements of the feature vector F_f actually constitute the antecedent parts of the MFI. The consequent part of the fuzzy implication represents the possibility of occurence of each class on the feature space.

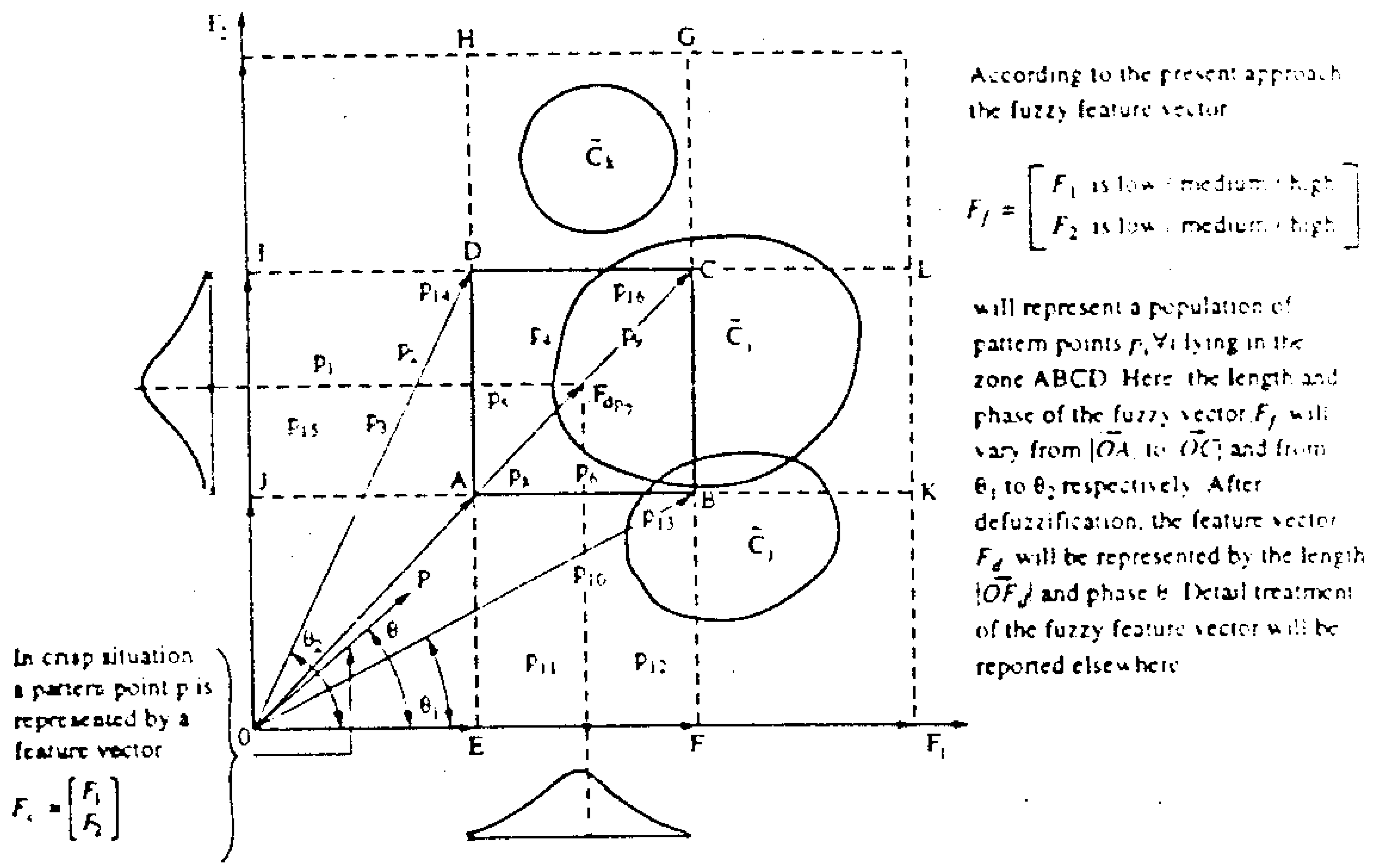


Figure 4: Representation of Feature vectors F_c, F_i, F_d

The process of recognition is divided into two steps, learning and classification. The main stages of learning are : feature extraction, feature selection, clustering and determination of the appropriate fuzzy if-then rules which will constitute a decision function.

The main stages of classification are : are extraction of a selected set of features, application of if - then rules and decision making based on the results obtained by the application of the if-then rules. In the classification process, first an unknown object is presented to the system, then a set of predetermined features is extracted from the scene. Finally a set of if-then rules determines the possibility of occurence of the object in the scene.

4 REVIEW ON BACKPROPAGATION TYPE NEURAL NETWORK

It is essential that the system be able to learn from experience and to infer descriminants autonomously. We consider the semilinear Feed forward network for the occluded object recognition. We apply the generalised delta rule for the Net with Backpropagation of the error. The semilinear feedforward net as reported by Rumelhart, Hinton and Williams [1986] has been found to be an effective system for learning descriminants for objects from a body of examples. It has been shown that the multilayer perceptron networks with a single hidden layer and a nonlinear activation function are universal classifiers.

The knowledge required to map input into an appropriate classification is embodied by the weights. Initially the weights appropriate to the given problem domain are unknown. Until a set of applicable weights is found the network has no ability to deal with the problem to be solved. The process of finding a useful set of weights is called training. Training begins with a training set consisting of specimen inputs with associated outputs that represent a correct classification.

Training the network involves moving from the training set, to a set of weights which correctly classifies the training set vectors atleast to within some defined error limit. In effect the network learns what the training set has to teach it. If the training set is good and the training algorithm is effective, the network should then be able to correctly classify input not belonging to the training set. This sometimes termed generalisation. The application of the neural network to recognition problem involves two phases. During the training phase the network weights are adapted to reflect the problem domain as shown in Fig. 5. In the operational phase, the weights have been frozen and the network when presented with test data or real world data will predict a classification. This is shown in fig 6

To train the network, we must recognise the need for a measure of how close the network has come to an established desired value. This measure is the network error. Since we are dealing with supervised training, the desired value is known to us for the given training set. The overall process of Backpropagation including forward and backward passes is shown in the flowchart. (fig 8).

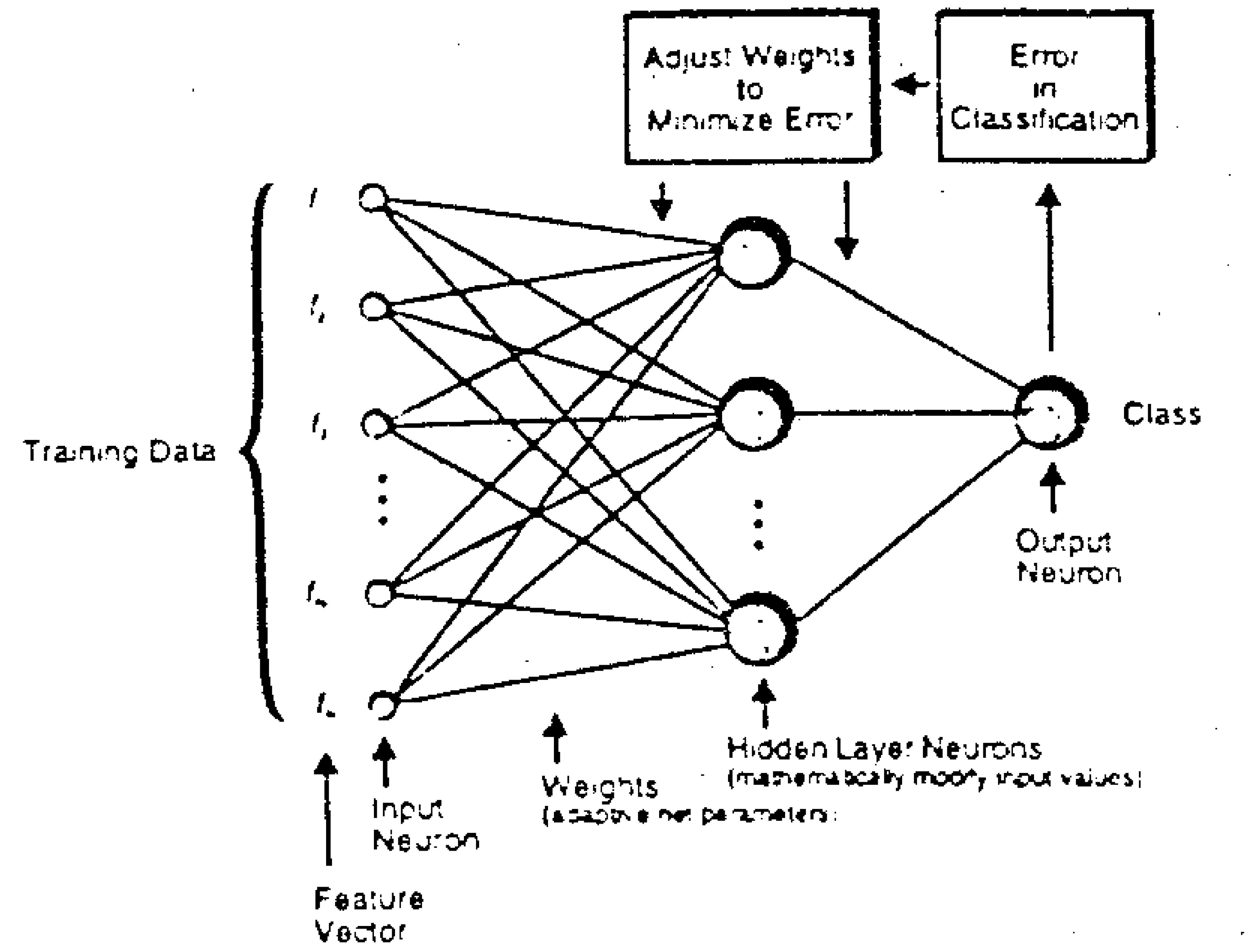


Figure 5: Training phase

In general such a net is made up of set of nodes arranged in layers. The output of the nodes one layer is transmitted to the nodes of the other layer through the links that amplify or attenuate such output through the weighting factors. I have used and implemented the CONVENTIONAL approach to the back propagation type neural network.

4.1 CONVENTIONAL APPROACH

In this approach, except for the input layer nodes, the net input to each node is the sum of the weighted outputs of the nodes in the prior layer. Each node is activated in accordance with the input to the node, the activation function of the node and the bias of the node.

Thus in Fig.7 the components of an input pattern constitute the inputs to the nodes of the layer i. The output of the nodes in that layer may be taken equal to the inputs or they can be normalised such that they can be scaled to fall betweeen the values of -1 and +1.

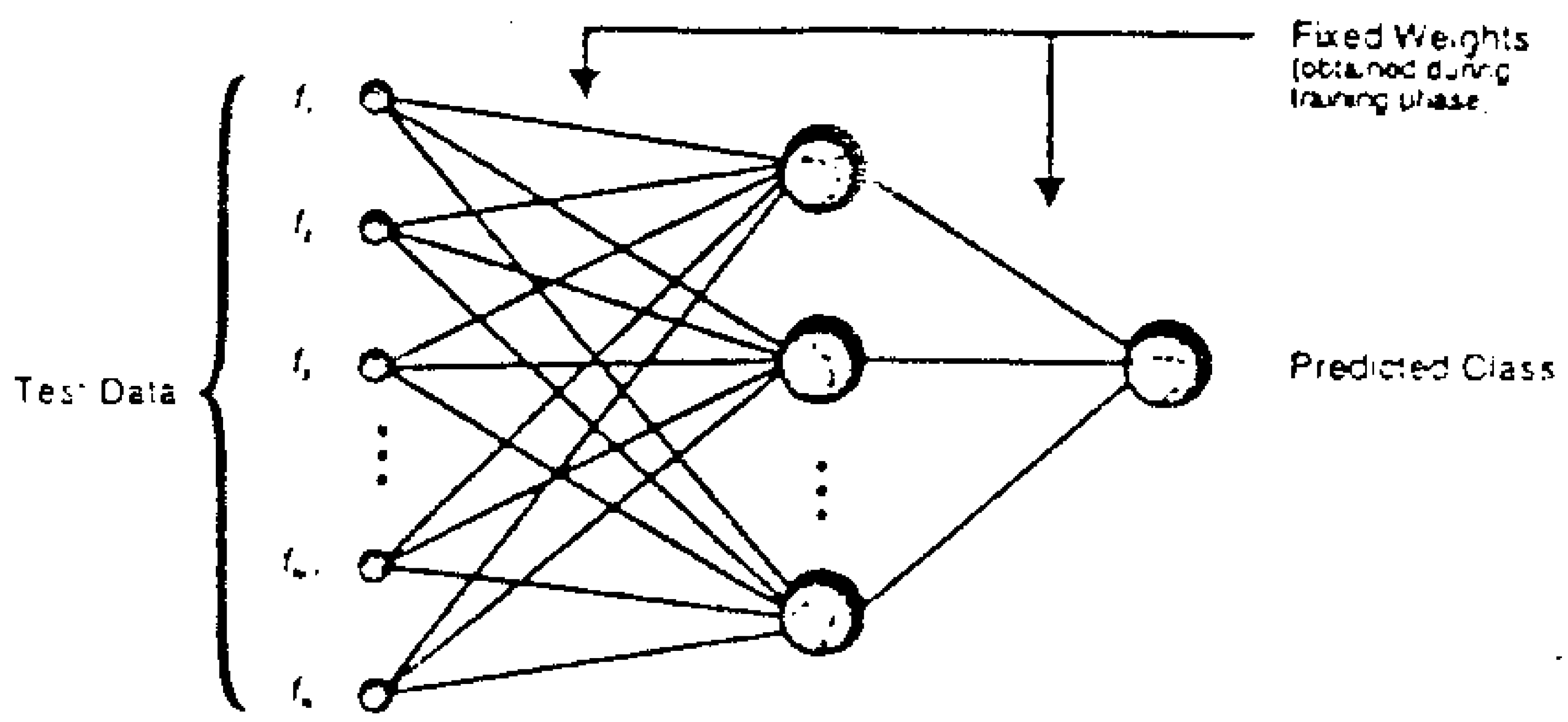


Figure 6: Prediction phase

The net input to a node in layer j is

$$net_j = \sum w_{ji} o_i$$

The output of the node j is

$$o_j = f(net_j),$$

where f is the activation funcition.

For a sigmoidal activation function, we have

$$o_j = \frac{1}{1 + e^{-(net_j + \theta_j)/\theta_0}}$$

In the above expression the parameter θ_j serves as the threshold or bias. The effect of a positive θ_j is to shift the activation function to the left along the horizontal axis, and the effect of θ_0 is to modify the shape of the sigmoid. A low value of θ_0 tends to make the sigmoid take on the charecteristics of the threshold logic unit (TLU),whereas a high value of θ_0 results in a more gently varying function. Continuing the description of the computational process, we have for the nodes in layer k the input

$$net_k = \sum w_{kj} o_j$$

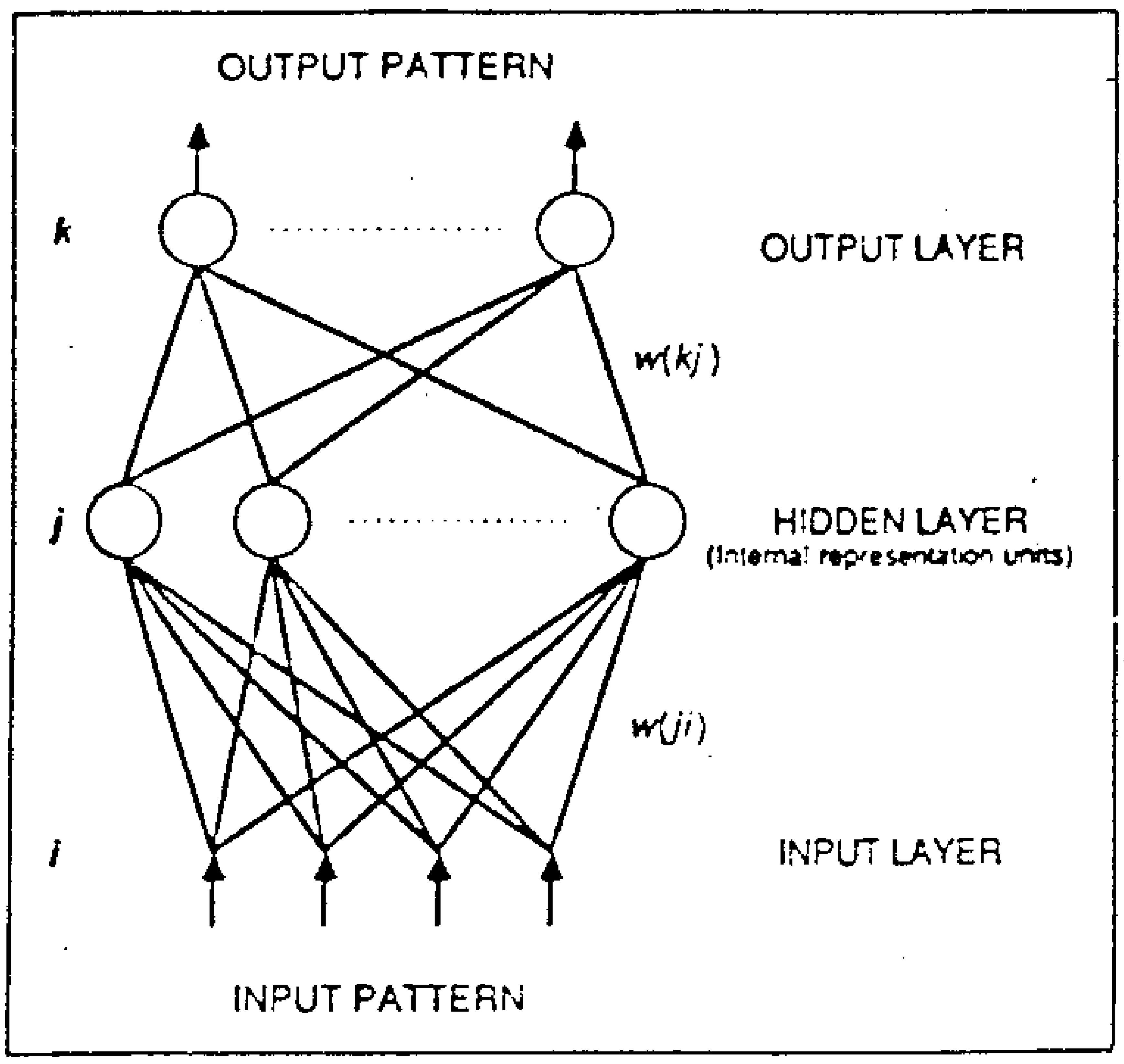


Figure 7: A schematic depiction of a nonlinear feedforward neural network with one hidden layer

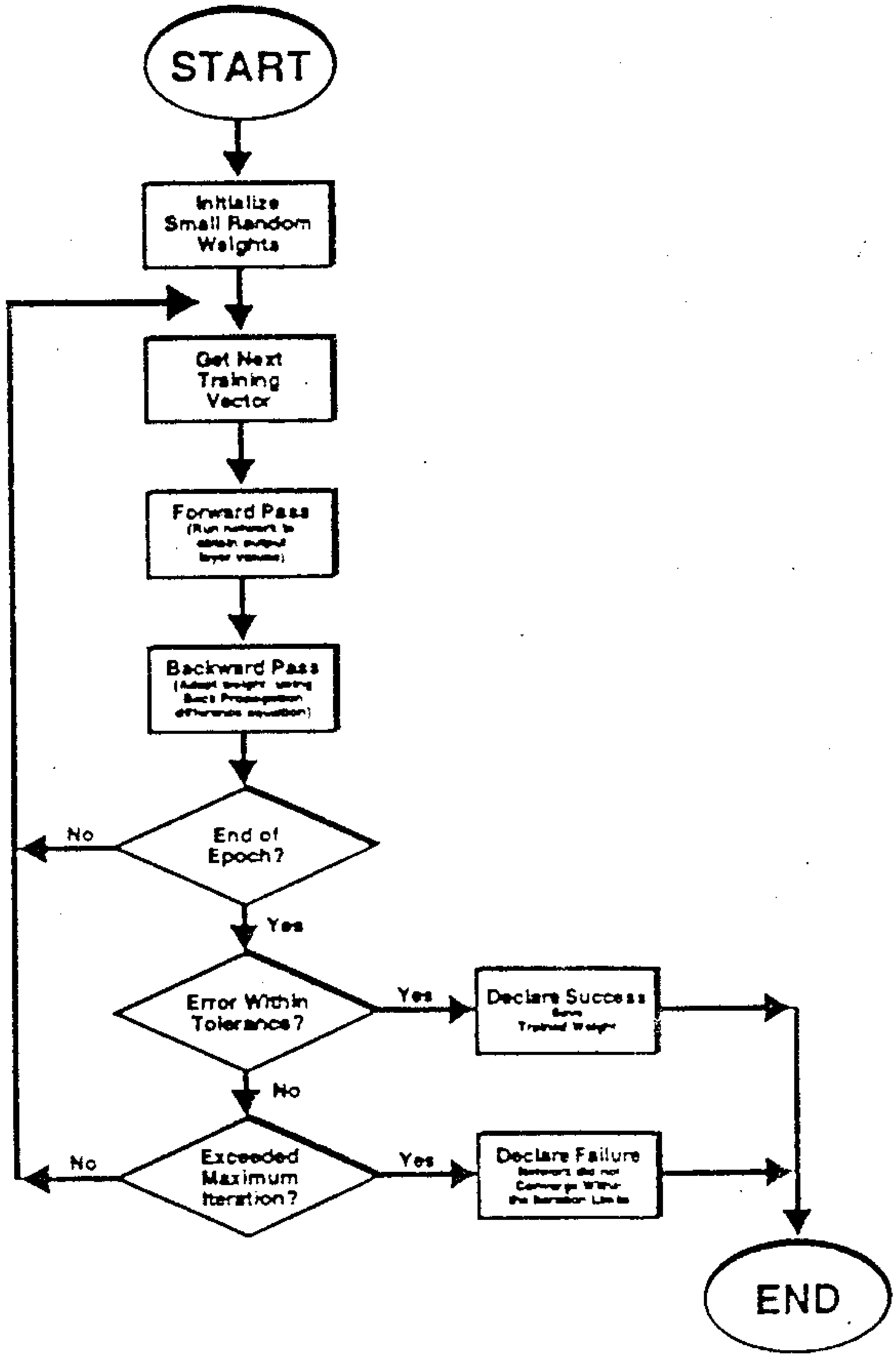


Figure 8: Flow chart for backpropagation.

and the corresponding outputs

$$o_k = f(net_k)$$

In the learning phase of training such a net, we represent the pattern $I_p = \{i_{pi}\}$ as input and ask that the net adjust the set of weights in all connecting links and also all the thresholds in the nodes such that the desired output $\{t_{pk}\}$ are obtained at the output nodes. Once this adjustment has been accomplished by the net, we represent another pair of I_p and $\{t_{pk}\}$,and ask that the net learn that association also.In fact, we ask the net to find a single set of weights and biases that will satisfy all the (input, output) pairs presented to it. This process can pose a very strenuous learning task and is not always readily accomplished.

In general, the outputs o_{pk} will not be same as the target or desired values t_{pk}. For each pattern, the square of the error is

$$E_p = \frac{1}{2}\sum_k (t_{pk} - o_{pk})^2$$

and the average system error is

$$E = \frac{1}{2P}\sum_p \sum_k (t_{pk} - o_{pk})^2$$

where P is the number of objects used for training the net. Backpropagation is one of the simpler members of the family of training algorithms collectively termed gradient. Our goal is to minimise the network total error by adjusting weights. Gradient descent, is sometimes known as the method of steepest descent.

In generalised delta rule, we achieve convergence towards the improved values for weights and threshold/biases by taking incremental changes $\triangle w_{kj}$ proportional to $-\partial E_p/\partial w_{kj}$ that is,

$$\triangle_p w_{kj} = -\eta \frac{\partial E_p}{\partial w_{kj}}$$

where η is the learning coefficient. However E_p the error, is expressed in terms of the outputs o_k, each of which is a nonlinear output of node k. That is,

$$o_k = f(net_{pk})$$

where net_{pk} is the input to the kth node and by definition is the weighted linear sum of all the outputs from the previous layer:

$$net_{pk} = \sum w_{kj} o_j$$

The partial derivative $\partial E_p / \partial w_{kj}$ can be evaluated using the chain rule

$$\frac{\partial E_p}{\partial w_{kj}} = \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial w_{kj}}$$

Hence we can obtain

$$\frac{\partial net_{pk}}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} \sum w_{kj} o_{pj} = o_{pj}$$

we now define

$$\delta_{pk} = -\frac{\partial E_p}{\partial net_{pk}}$$

and write

$$\triangle_p w_{kj} = \eta \delta_k o_j$$

To compute $\delta_{pk} = -\partial E_p / \partial net_{pk}$,we can use the chain rule to express the partial derivative as follows,

$$\delta_{pk} = -\frac{\partial E_p}{\partial net_{pk}} = -\frac{\partial E_p}{\partial o_{pk}} . \frac{\partial o_{pk}}{\partial net_{pk}}$$

The two factors are obtained as follows,

$$\frac{\partial E_p}{\partial o_{pk}} = -(t_{pk} - o_{pk})$$

and,

$$\frac{\partial o_{pk}}{\partial net_{pk}} = f_k'(net_{pk})$$

From which we obtain $\delta_k = (t_{pk} - o_{pk}) f_k'(net_{pk})$ for any ouput-layer node k and we have,

$$\triangle_p w_{kj} = \eta \delta_{pk} o_{pj} = \eta (t_{pk} - o_{pk} f_k'(net_{pk}) o_{pj}$$

We still write,

$$\triangle_p w_{ji} = -\eta \frac{\partial E_p}{\partial w_{ji}}$$

$$= -\eta \frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}}$$

$$
\begin{aligned}
&= -\eta \frac{\partial E_p}{\partial net_{pj}} o_j \\
&= \eta \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} o_{pj} \\
&= \eta \frac{\partial E_p}{\partial o_{pj}} f_j^{'}(net_{pj}) o_{pi} \\
&= \eta \delta_{pj} o_{pi}
\end{aligned}
$$

However, the factor $\frac{\partial E_p}{\partial o_{pj}}$ can not be evaluated directly. Instead we write it in terms of the quantities that are known and other quantities that can be evaluated. Hence we need to apply the chain rule again.

$$
\frac{\partial E_p}{\partial o_{pj}} = -\sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}}
$$

$$
\begin{aligned}
&= -\sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial}{\partial o_{pj}} \sum_m w_{km} o_m \\
&= \sum_k -\frac{\partial E_p}{\partial net_{pk}} w_{kj} \\
&= \sum_k \delta_k w_{kj}
\end{aligned}
$$

Now we can represent the error in the hidden layer as:

$$
\delta_{pj} = f^{'}(net_{pj}) \sum_k \delta_{pk} w_{kj}
$$

Thus the deltas at an internal node can be evaluated in terms of the deltas at an upper layer. Thus, starting at the highest layer, the output layer we can evaluate δ_{pk} and we can propagate the "errors" backward to lower layers. In particular, if

$$
o_{pj} = f(net_{pj}) = \frac{1}{1 + e^{(-net_{pj} + \theta)}}
$$

and then

$$
\frac{\partial o_{pj}}{\partial net_{pj}} = f^{'}(net_{pj}) = o_{pj}(1 - o_{pj})
$$

and the deltas are given by the following expressions:

$$
\delta_{pk} = (t_{pk} - o_{pk}) o_{pk} (1 - o_{pk})
$$

$$
\delta_{pj} = o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{kj}
$$

for the output and hidden layers respectively. Note that the threshold/bias θ_j are learned in the same manner as the other weights are learned. We simply imagine that θ_j is the weight from a unit that always has an output value of unity. It may be mentioned at this point that the generalised delta rule used in the learning includes a momentum term and is stated as follows :

$$\triangle_p w_{ji}(n+1) = \eta(\delta_{pj} o_i) + \alpha \triangle_p w_j i(n),$$

where the quantity (n+1) is used to indicate the (n+1)th step, and α is a proportionality constant. The second term in the above expression is used to specify that the change in w_{ij} at the (n+1)th step should be somewhat similar to the change undertaken at the nth step.

5 IMPLEMENTATION OF NEW INTERPRETATION OF MULTIDIMENSIONAL FUZZY REASONING (MFR) AND BACK-PROPAGATION TYPE NETWORK

Let us consider the Eq.9 of chapter-3, which is stated earlier. It has two expressions each of them is a law of implication. We implement each of these laws through back propagation type neural networks which are basically three layered perceptrons (refer Fig.9). The antecedent part of the If-Then rules is fed as input to the neural networks. The antecedent part is represented by the fuzzy membership function. The consequent part of the rule forms reference output. The consequent part is also represented by the fuzzy membership function which basically represents the possibility of occurence of each class. Other features of the network is same as the conventional back-propagation network.

Each network is trained independently by a set of fuzzy if-then statements using generalised delta rule (Pao, 1989). Once the network is trained we can combine the output of each network by the intersection ($\cap$) operator (See Fig.9). Thus, if we have a two dimensional law of fuzzy implication as shown in Eq.10 , we can realise the MFR through a network configuration shown in Fig.9. If we have an n - dimensional law of fuzzy implication, We can realise the MFR through a network which consists of n - number of independent three layered perceptrons which are trained using the principle of back-propagation type neural networks. The output of n networks can be combined through an intersection ($\cap$) operator. Here the object classification on R^2 is considered and we always consider the network structure of Fig.9.

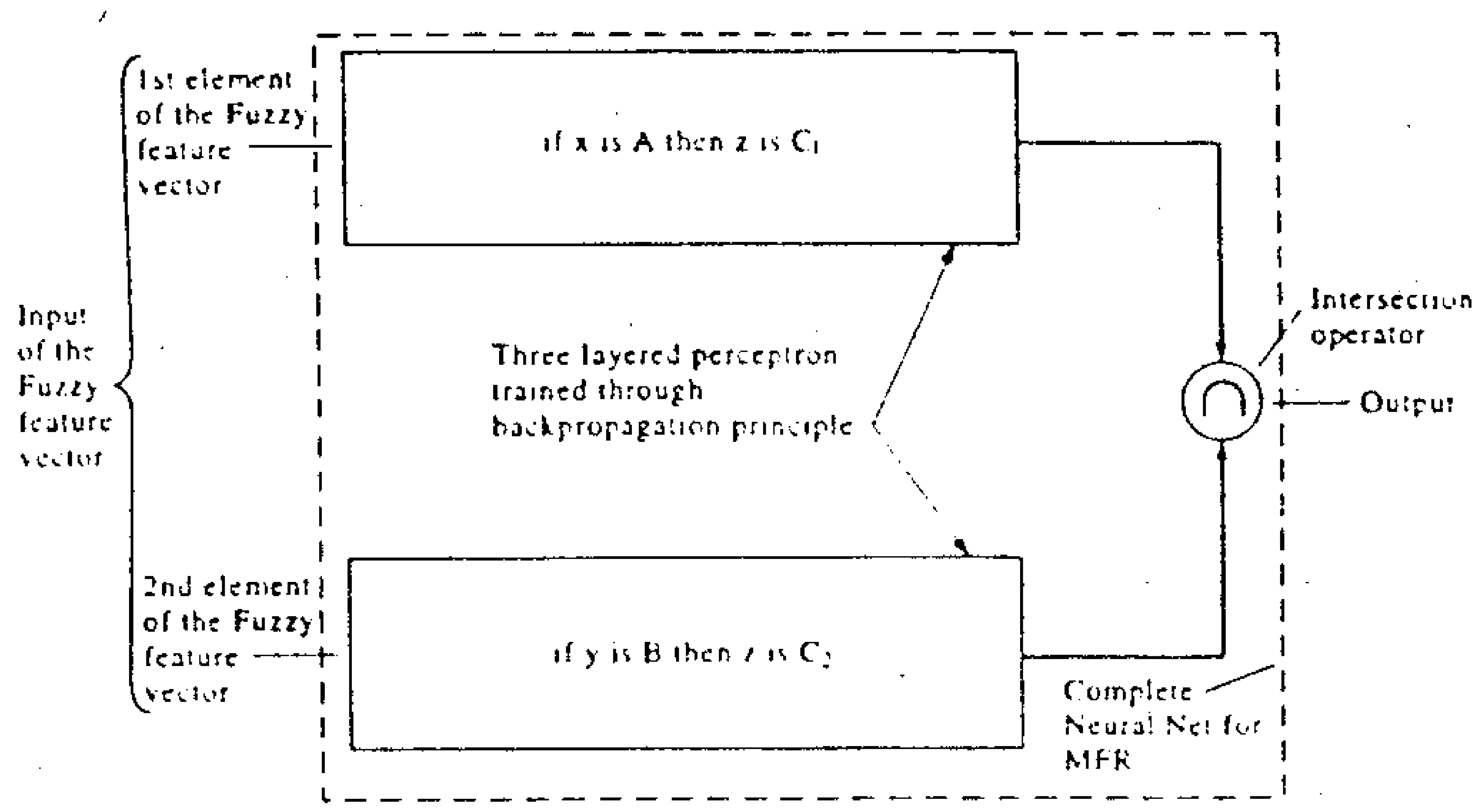


Figure 9: Realization of two dimensional MFR through MLP type neural network

6 FORMULATION OF THE PROBLEM

6.1 LOCAL FEATURE EXTRACTION

One class of techniques dealing with the problem of object recognition uses the global object features such as area and perimeter of boundary curve, centroid and shape moments. However, this type of approach is not suited for the recognition of partially occluded objects whose global features are totally destroyed. Some of the local object attributes may be subjected to change, such as slope and position, but some of them will remain invariant, such as curvature and internal angle of a curved object. The invariance property of such object features is used to generate $F_1 - F_2$ feature space. In our case, we have considered the internal angle of significant points on a curve as feature F_1 and corresponding curvature as the feature F_2. Now, curvature is a measure of the rate of change of orientation per unit arc length. The geometric interpretation for the curvature is depicted in the Fig 10. Let P be a point on the curve, T be the tangent at that point and A be a neighbouring point on the curve. Let α denote the angle between the line AP and T, and arc(AB), the arc length between A and B. The curvature k at P is the ratio $\alpha/$ arc(AB).

6.2 PROCESS OF FUZZIFICATION

At the learning stage we discretise the universe of discourse of the features F_1 (internal angle) and F_2 (curvature). Discretization is often referred to as quantization which discretizes the universe into a certain number of segments (quantization levels). Each segment is labelled as a generic element and forms a discrete universe. A fuzzy set is then defined by assigning a grade of membership values to each generic element of the new discrete universe. At the classification stage, the selected features are fuzzified using the concept fuzzy singleton.

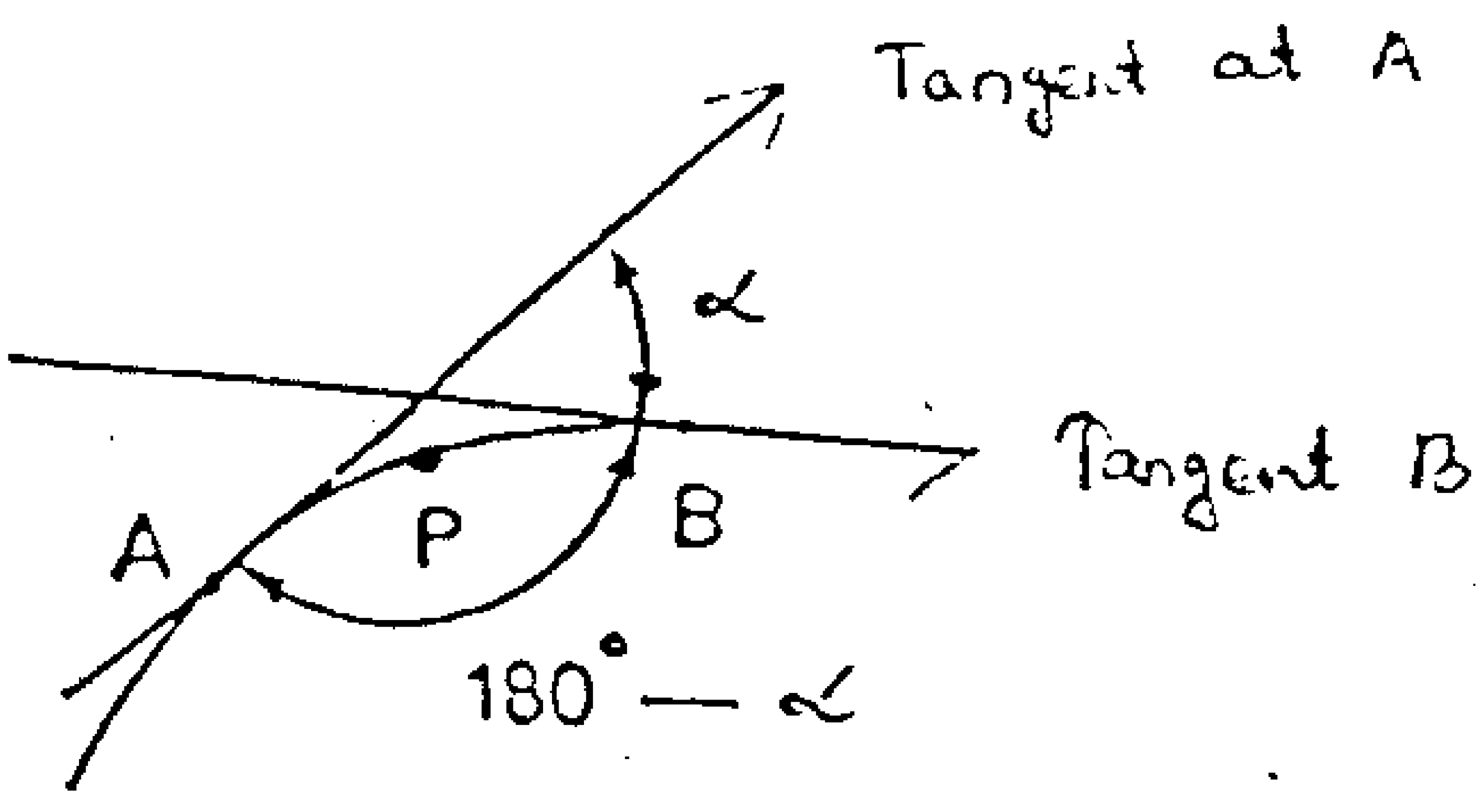


Figure 10: The internal angle at point P with respect to its neighbourhood, is $180 - \alpha$

6.3 ASSIGNMENT OF THE MEMBERSHIP FUNCTION TO THE CONSEQUENT PART OF THE IF-THEN RULES

The membership function of the consequent part of the **if-then** rule represents the possibility of occurence of each class in the fuzzily partitioned feature space(see Fig. 11). The membership function of the consequent part of the **if-then** rules are calculated as follows:

At the learning stage of the classifier, we depend on the expert's experience which is captured through fuzzy **if-then** rules. For this purpose we form a Query Table (QT). We explain this table with help of an example. Let F_1 and F_2 be names of the two linguistic variables and **High, Medium and Low** are the three primary terms. Assume that F_1 and F_2 form the essential input features to the classifier. Let there be two classes $C_1 and C_2$. Suppose, the output linguistic variables are formed with the help of two primary terms **Low and high**. Now we construct the **QT** as shown in the Table 2.

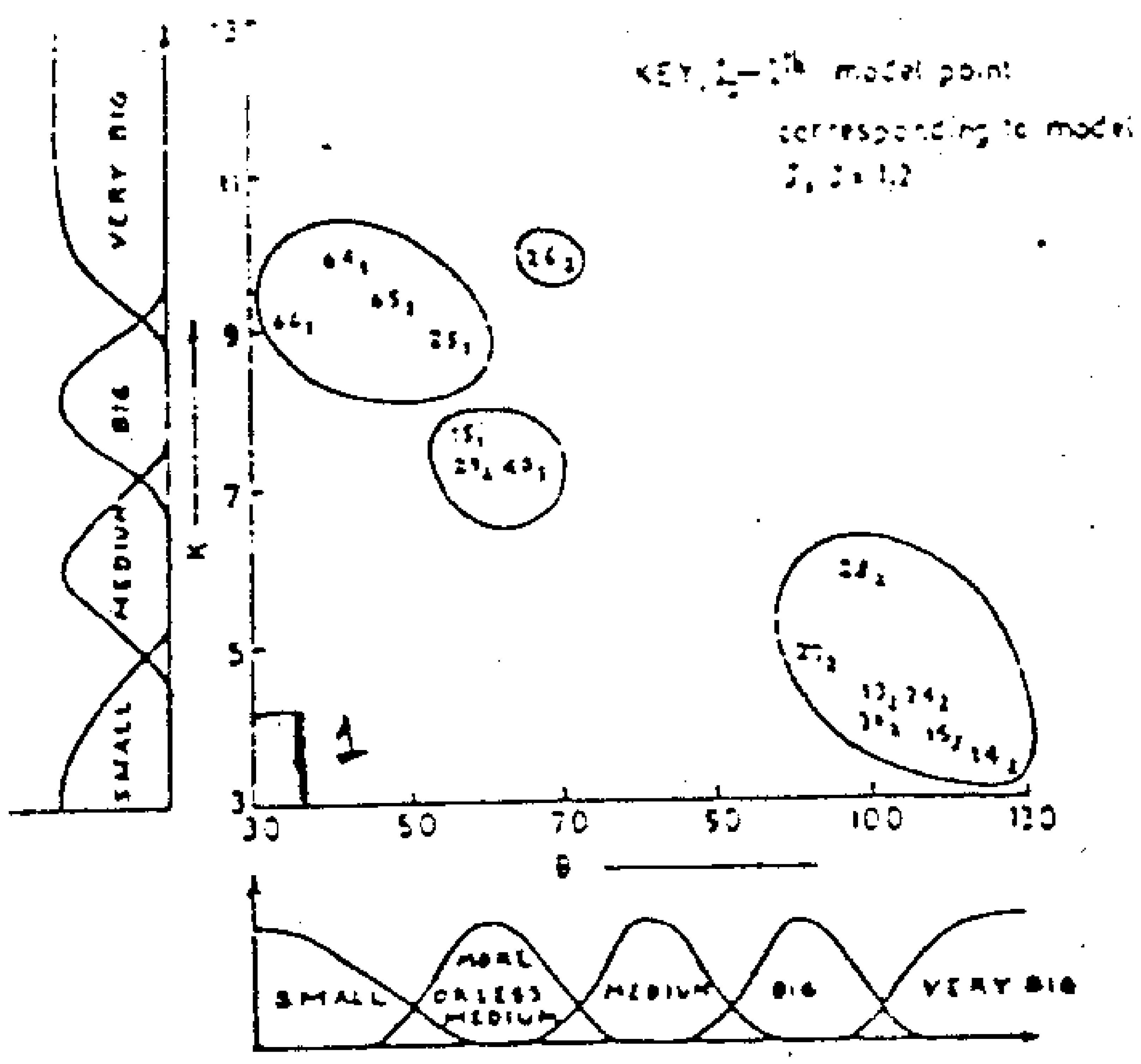


Figure 11: Significant points of pliers and wrench

If we consider Table 2 and read any of the rows first and then any of the columns of the QT, a question will be formed. The answer to this question is put in the corresponding elemental positon of the matrix. If the answer is "yes" we put "+", if "no", a "-" and if not answerable a "$\sim$". As for illustration, if we read the first row and then third column, we will have the quesion:"**if F_1 is Low will C_2 be High ?** ". If the answer is "yes" we put a "+" in (1.3) position of the **QT**. Now, we set **"Very High" = 1. "High" = 0.9. "More or Less High" = 0.7, "Medium" = 0.6, "Very Low" = 0.01, "Low" = 0.1, "More or Less Low" = 0.2** to assign membership values to the consequent part of the **if-then** rules.

Such assignment of membership function to the consequent part of the fuzzy **if-then** rules does not follow any conventional approach to fuzzification. Rather we use the concept of Scaling[23] which is basically derived from the scale proposed by Saaty[16] and the exponential scales proposed by Lootsma[7]. For any practical application,there is no single scale which can outperform the other scales [23]. Therefore depending upon the application we have to select the appropriate scale. In [7] there are some discussion on scales which are similar to the one used in this work. There are two critical problems raised in scaling. One is how to quantify (numerically) the linguistic choices selected by the decisionmaker and the second is how to process the numerical values. Both the problems have been nicely tackled in the present scaling process using fuzzy set.

Depending on the nature of the membership function of the antecedent part of the rule, we fuzzily partition the feature space and generate the **if-then** rules to classify the local features of the model objects. After the initial generation of the **if-then** rules we test the validity of the rules by classifying some unknown features of the model objects. If we get the satisfactory classification results, we proceed further, otherwise we tune the rules by changing (change around the initial value) the grades of the membership function of the antecedent parts and consequent parts. As off-line generation of fuzzy **if-then** rules for pattaern classification basically deals with a static situation, the tuning of the grade of the membership functions of the antecedent part and consequent part does not take a very long time which is very common phenomenon for tuning fuzzy control rules of the dynamical system. In the present situation, tuning of the consequent part of the rule is primarily guided by the seed points of the clusters of the features and that of the antecedent part of the rule is guided by the error generated in the classification. After tuning, the net is relearned by the refined set of rules.

6.4 PROCESS OF DEFUZZIFICATION

At the time of taking non fuzzy decision out of the fuzzy classification (i.e., defuzzification) we can go by selecting the class having the highest membership value. In case of a tie situation, which normally occurs for the features lying in overlapped zone we have to state the equal possibility of a feature belonging to both the classes. And such a conclusion is quite natural which normally does not exist in conventional classification approach. In some cases features in the over lapped zones are classified with "almost equal" possibility of occurence for more than one class. If such situation is treated as tie situation mentioned above, we have to select an appropriate threshold which entirely depends on the need of the problem.

Thus from the graded consequence, when we select a single class having the highest membership value, we consider the hard partitioning of the feature space. Whereas from the graded sequence, when we consider multiple classifications occurring in the overlapped zone , we consider the fuzzy partitioning of the feature space.

6.5 GENERATION OF THE MODEL BASED OBJECT RECOG-NITION SCHEME

The values of the internal angle and the curvature of the models are plotted on $F_1 - F_2$ plane. F_1 axis represents the internal angle F_2 axis represents the curvature axis(see Fig 11). Based on the values we cluster the features(see Fig.11). F_1 and F_2 axes are then fuzzily partitioned and several **if - then** rules are constructed to capture the information about the local features of the model objects. These rules form the knowledge base for the model based object recognition. The antecedent part of the rule is fed as input to the neural net and the consequent part represents the target value with which the individual output of the network is compared and using the generalised delta the network is trained. After the training is over if the local features of an unknown scene consisting of model objects are calculated and injected as fuzzy singleton to the input of the network, then the network can classify which particular feature belongs to which particular class.

The number of inputs to the network depends on the fuzzification process discussed in Section 6.2. and the number of outputs equals the number of model objects to be classified.

In the scene the model objects may have been placed in different orient ations and may partially occlude each other. In case of a partial occluded object the number of local features visible is less than the number of local features visible in the corresponding model object. We assume under occlusion atleast "50% of the local features $\pm\sigma$". That means, a maximum "50% (approx.)" occlusion of the model object is tolerable to the to the recognition scheme. Let a model object has a total 10 features. Under maximum conclusion(i.e., under 50% occlusion) 5 features are expected to be visible in the scene consisting of the model and other objects. During the recognition process out of the 5 features of the said model either
(i) some of the features(say one or two) may not be visible due to some error in computation or noise or
(ii) some additional features(say one or two) which are newly generated due to occlusion etc. may be recognised as the features of the said model.

After the recognition of the local features of the scene if the recognition scheme produces the number of model features visible in the scene (by checking the highest output of the network) for each object, then the following decision rule of of Table 3.. will help us to state the degree of possibilty of each model object to be present in the scene.

Now, for each object 'i' in the scene, a rough estimate of the number of the number of model features not visible in the scene is represented by the difference between the number of features(for the occluded object i) correctly reecognised (by checking the highest output of the neural net) minus $k * TNMF$ (TNMF is considered for the model object i). In the occluded environment under hard partitioning this said difference is non positive. And for asserting the possibility of presence of an object in the scene we safely go by the rules of the Table 3.. But under hard partitioning if the difference is positive then it becomes an alarming situation indicating that some of the features of the model object 'i' are recognised as the features of the model object 'j' where $i \neq j$ In that case the vision

Table 2: QT matrix

IF	*Will C_1 be*		*Will C_2 be*	
	Low	High	Low	High
F_1 is Low	$\sim$	$\sim$	$\sim$	$\sim$
F_1 is High	-	+	+	-
F_1 is Medium and F_2 is High	-	+	-	+
F_1 is High and F_2 is Medium	+	-	-	+

should be intervened in an interactive mode. In the occluded environment under hard partitioning, the value of k is experimentally chosen as 1.5. Thus the vision system is given a freedom to recognise the features of the newly generated points which are either formed due to overlap of different objects in a scene or created due to some environmental uncertainities. As because the newly generated points were not present at the time the net was trained, it is quite obvious that it is only the strength of approximate reasoning which can correctly recognise these additional points.

In this context, I like to state the basic design philosophy of the vision system which says "We can recognise objects which we have seen earlier". That means in case of machine vision, the machine can recogn- ise the objects which it has seen (or about which it has learned) earlier. But under occlusion it is quite obvious, that some of the model features which were seen earlier, may not be visible in the scene due to significant changes in the feature values and hence may not be detectable by the recognition scheme. Conversely, some of the newly generated points which were not earlier seen by the machine and which have feature values close to the seen points of the model objects, are recognised correctly by the machine. Thus the proposed vision system has reasonable tolerance around the basic design philosophy.

In case of fuzzy partitioning, which is very essential under uncertain environment, the value of k is experimentally chosen as 2. Thus we have given some added flexibility in the design of the vision system at the cost of some risks. But the achievements due to this additional flexibility is very significant in comparision to the risks involved and hence very much acceptable for practical use.

Table 3: Decision Table

number of model features visible in the scene	possibility of presence	degree of prescence
EQ 0	Nil	0
GT 0 but LT(50% of TNMF- σ)	poor	0.3
EQ (50% of $TNMF \pm \sigma$)	fair	0.7
GT (50% of $TNMF+ \sigma$) but $LT(k^{\circ}TNMF)$	Good	1

Table 4: Internal angle and Curvature of the model object 1 (see Fig.12)

Model point	Internal angle	Curvature
15	56.07	7.74
25	74.44	8.96
40	63.98	7.25
64	39.07	10.06
65	46.89	9.50
66	32.64	9.20

7 CASE STUDY

Here we consider a scene which consists of two objects. The objects are pliers and wrench (see Fig 12 and 13). We calculated the internal angle θ (Feature F_1) and radius of curvature k (Feature F_2) at some significant points of the model objects. (Refer Tables 4 and 5). These local features are plotted as in Fig 11. and clusters are formed. The features are then fuzzified as discussed in section 6.2. Next we generate the fuzzy **if-then** rules (see Tables 6 and 7) The network of the Fig 9 is trained by the said rules and is made intelligent to recognise the objects in an unknown scene. The output of the program is given and the results are shown in the table 8. The dynamic ranges of the features of both model and scenes are $30 - 130$ deg (internal angle) and 3-13 (curvature) I have considered 6 significant points of pliers (refer table 4) and 9 significant points(Refer table 5) of the wrench. But the total number of significant points in the scene need not necessarily be 15 (6+9). In case of fig 14. we have a total of 15 significant points of which 6 points come from model 1, 5 points come from model 2 and the rest are newly generated due to the overlapping of the objects. The newly generated points (formed due to the overlapping of the objects) may be classified in any one of the model classes or may be classified in more than one model class.

The newly generated points which are basically the junction points in the scene and the significant points whose feature values are not exactly same as the feature values of the model objects are satisfactorily classified by the strength of Approximate Reasoning. In the present approach, the dynamic range of features vary from $30 - 130$ deg and $3 - 13$, we take σ equal to one. The choice of σ is subjective and problem dependent.

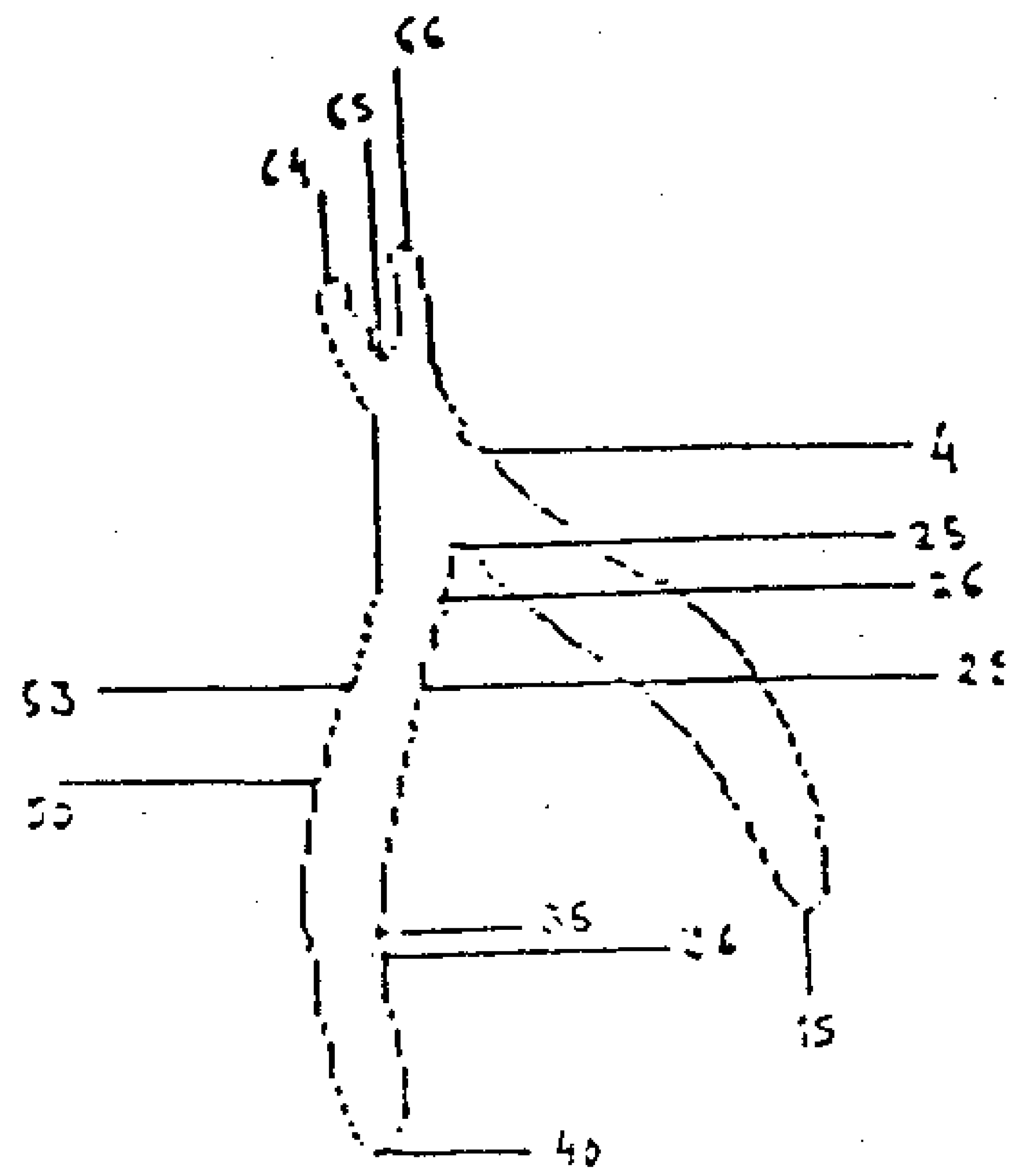


Figure 12: Model object Pliers

7.1 THE OUTPUT OF THE PROGRAM

Select L(earning) or O(utput generation):

L

START OF LEARNING PROCESS

Enter the task name : trngftr1

How many features in the input pattern : 20

Table 5: Internal angle and curvature of the model object 2 (see Fig.13)

Model point	Internal angle	curvature
13	109.39	4.41
14	123.64	3.52
15	118.36	3.85
24	112.78	4.20
26	60.77	9.93
27	101.40	4.91
28	107.05	6.07
29	57.97	7.62
30	110.00	4.47

Table 6: Training rules for the First feature of Fig. 11

Antecedent	Consequent (possibility of occurence)	
	Model 1	Model 2
If θ is small then	1	0
If θ is more or less medium then	0.9	0.1
If θ is Medium then	0	1
if θ is Big then	0	1
If θ is very Big then	0.1	0.9

Table 7: Training rules for the Second feature of Fig.11

Antecedent	Consequent (possibility of occerence)	
	Model 1	Model 2
if k is small then	0.1	0.9
if k is medium then	0.5	0.5
if k is big then	0.9	0.1
if k is very big then	0.5	0.5

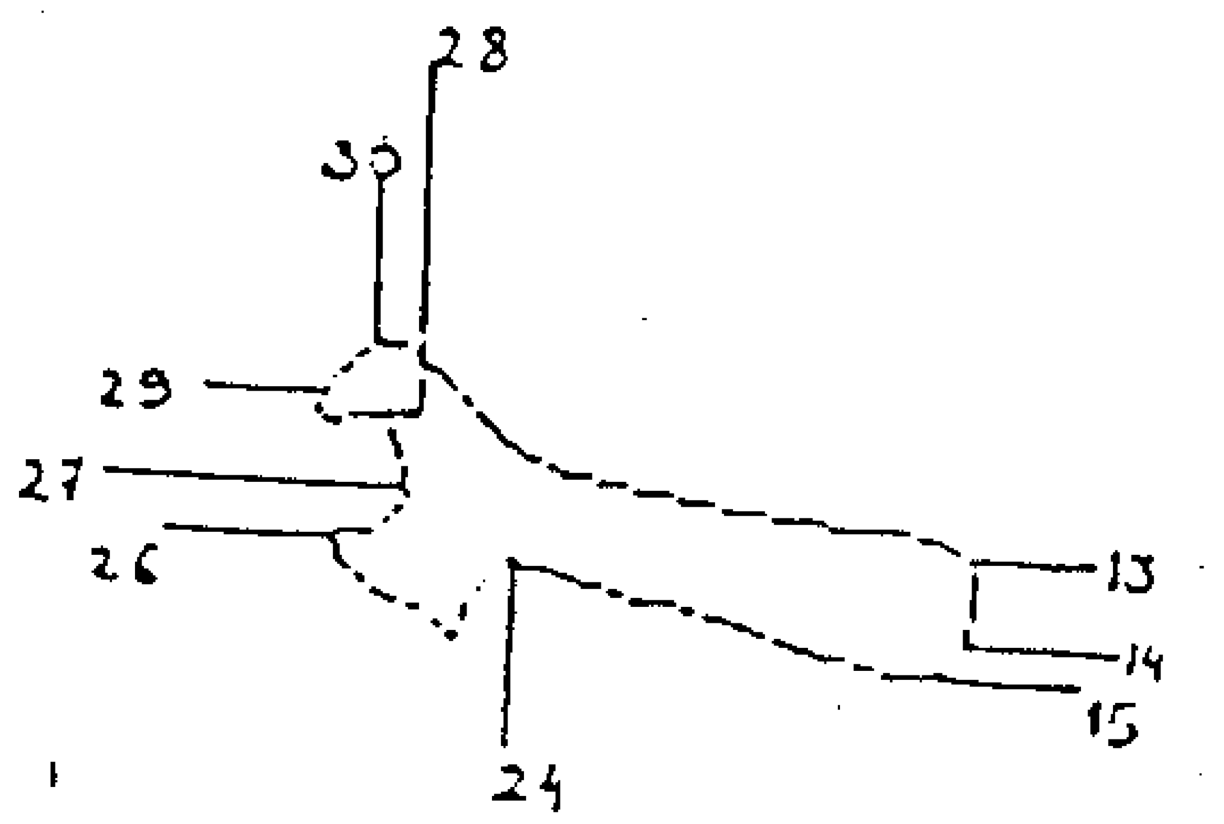


Figure 13: Model object Wrench

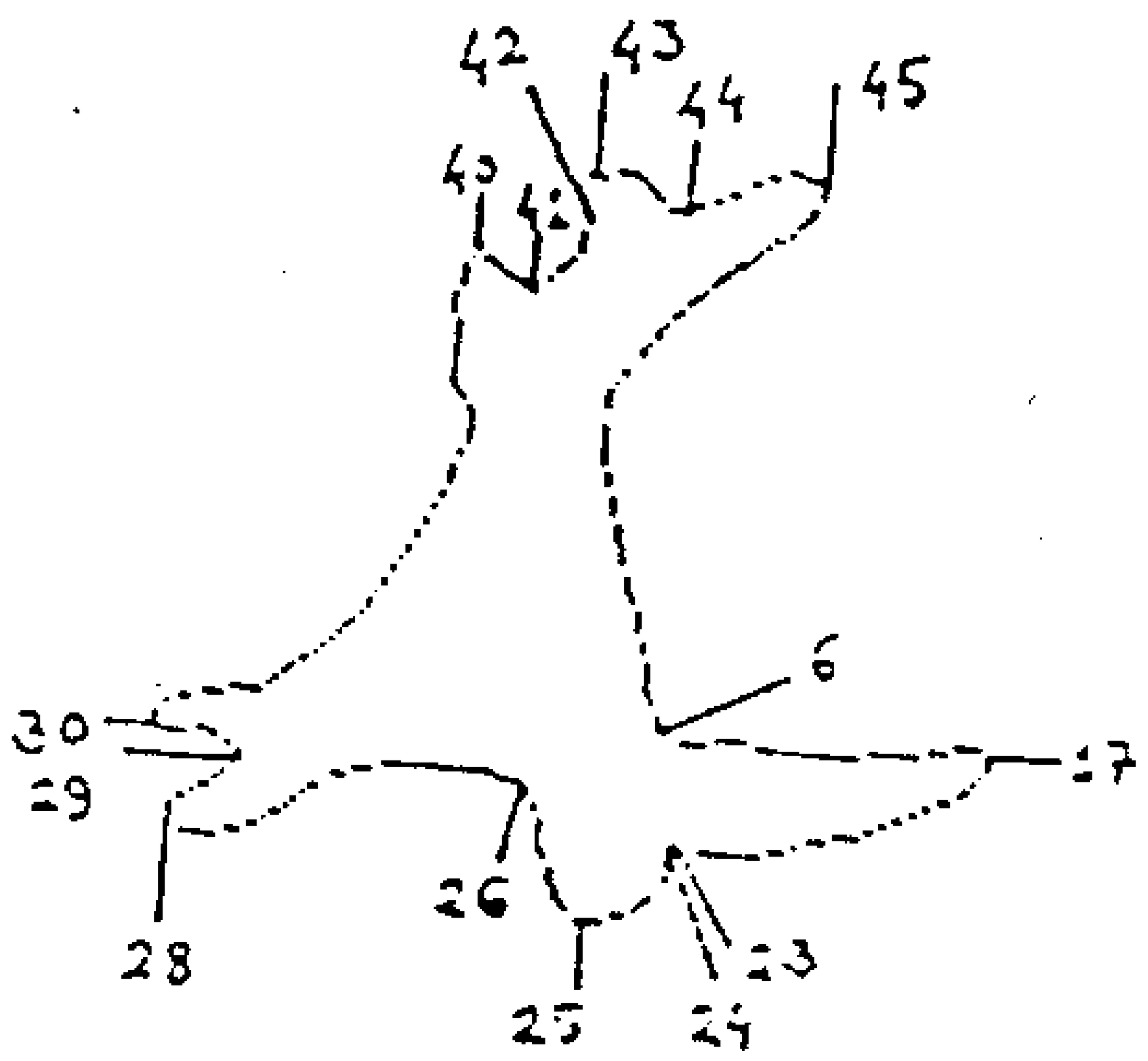


Figure 14: Occlusion

```
How many output units: 2
Total no of input samples : 5
Input file name is trngftr1.data
Do you want to look at the data just read
Answer Yes or No : n
Momentum rate eta (default = 0.9) ? : 0.9
Learning rate alpha (default = 0.7) ? : 0.7
Maximum total error (default = 0.01) ?: 0.000001
Maximum inidividual error (default = 0.001) ?: 0.0000001
Max number of iterations(default = 1000) ?: 15000
Number of hidden layers ?: 1
Number of units for the hidden layer 1 : 10
Create error file ? If so type 1 or 0: 1

sample 0 output 0 = 0.995305 target 0 =1.000000

sample 0 output 1 = 0.004660 target 1 =0.000000

sample 1 output 0 = 0.900008 target 0 =0.900000

sample 1 output 1 = 0.099997 target 1 =0.100000

sample 2 output 0 = 0.002170 target 0 =0.000000

sample 2 output 1 = 0.997837 target 1 =1.000000

sample 3 output 0 = 0.001989 target 0 =0.000000

sample 3 output 1 = 0.998021 target 1 =1.000000

sample 4 output 0 = 0.099992 target 0 =0.100000

sample 4 output 1 = 0.900005 target 1 =0.900000

total number of iterations is 15000
Normalised system error is 0.000012
```

Do you want to continue

y

Select L(earning) or O(utput generation)

o

Enter 1 for the top network and 2 for the bottom network

1

Enter the no of patterns for processing: 15

Do you want to continue:

yes

Select Learning or O(utput) generation :

L

START OF LEARNING PROCESS

Enter the task name : trngftr2

How many features in the input pattern: 10

How many output units: 2

Total no of input samples : 4

Input file name is trngftr2.data

Do you want to look at the data just read

Answer Yes or No ; no

Momentum rate eta (default = 0.9) ? : 0.9

Learning rate alpha (default = 0.7) ? : 0.7

Maximum total error (default = 0.01) ?: 0.000001

Maximum inidividual error (default = 0.001 ?: 0.0000001

Max number of iterations(default = 1000) ?: 15000

Number of hidden layers ?: 1

Number of units for the hidden layer 1 : 5

Create error file ? If so type 1 or 0: 1

sample 0 output 0 = 0.100569 target 0 =0.100000

sample 0 output 1 = 0.899295 target 1 =0.900000

sample 1 output 0 = 0.500325 target 0 =0.500000

sample 1 output 1 = 0.499909 target 1 =0.500000

sample 2 output 0 = 0.898902 target 0 =0.900000

sample 2 output 1 = 0.101264 target 1 =0.100000

sample 3 output 0 = 0.499833 target 0 =0.500000

sample 3 output 1 = 0.499923 target 1 =0.500000

total number of iterations is 198
Normalised system error is 0.000001

Do you want to continue
yes
Select L(earning) or O(utput generation)
o
Enter 1 for the top network and 2 for the bottom network
2
Enter the no of patterns for processing: 15

outputs have been generated

Do you want to continue
no
It is all finished
Good Bye

Table 8: Recognition scores for the scene occlusion

scene points	Membership values		Results	Remarks		Possibility of presence as per Table 2			
	Model1	Model2		Model1	Model2	Model1		model2	
						Lingustic value	Possibility degree	Linguistic value	possibility degree
6	0.21	0.76	Jn-point						
17	0.82	0.09	Correct	5 features	7 features	Good	1	Good	1
23	0.04	0.30	Wrong	out of	out of				
24	0.04	0.18	Jn-point	6 Model	9 model				
25	0.07	0.76	Correct	features	features				
26	0.15	0.66	Correct	have	have				
28	0.70	0.24	Correct	been	been				
29	0.69	0.30	Correct	correctly	correctly				
30	0.82	0.06	Correct	classified	classified				
40	0.10	0.57	Correct						
41	0.06	0.70	Correct						
42	0.41	0.18	Correct						
43	0.03	0.70	Correct						
44	0.24	0.73	Jn-point						
45	0.46	0.30	Jn-point						

REFERENCES

1. Y.H.Pao "Adaptive Pattern Recognition and Neural Networks". Addison Wesley.

2. Timothy Masters, "Practical Neural Networks Recipes". Academic Press Inc.1993.

3. K.S.Ray "Method of Approximate Reasoning"

4. K.S.Ray & J.Ghoshal "Fuzzy Sets and Systems". vol 2251,1996. ELSEVIER publication.

5. W.Pedrycz "IEEE TRANSACTIONS ON PATTERN RECOGNITION AND MACHINE INTELLIGENCE" VOL. 13, No.3, March 1991.

6. NORWAN ANSARI and EDWARD J.DELP "IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE".

7.Richard P. Lippman "An introduction to computing with Neural Nets". IEEE ASSP MAGAZINE,APRIL 1987.