UNIT 3 FUZZY PATTERN RECOGNITION

Structure **Page No**

3.1 INTRODUCTION

Pattern Recognition (PR) form a major area of research and development that encompasses the processing of pictorial, numeric and non-numeric data obtained from the nature. A pattern can be either a physical object (e.g., a human being or a car) or an abstract notion (e.g., a style of writing).

A pattern is represented as a measurement or attribute vector, where each dimension corresponds to a feature of the pattern. These features can be either quantitative or qualitative. For example, if *mass* and *color* are the two features used, then object with 20 units of mass and black in color, can be represented as an ordered two-tuple (20, black). One of the most difficult and challenging problems in pattern recognition is to decide on the ideal set of features for representing a domain. There is no guideline which suggests the appropriate features to use in representing patterns. A major problem is that representation is a highly subjective activity and so a general theory is unlikely to emerge. For example, given a collection of human beings represented using the *mass* attribute alone, it is not possible to separate these patterns into groups corresponding to *tall* and *short* people.

Pattern Recognition techniques can be classified into two broad categories: *unsupervised* techniques and *supervised* techniques. In supervised techniques, a set of human-annotated training samples is used to train the pattern recognition system. During the process of training, the system learns to distinguish between the different classes of samples by associating a distinctive set of feature values to each class. The resulting knowledge is stored as a classifier which can be a set of rules, a mathematical function or a set of weights, depending on the type of system implemented. The accuracy of the classifier is computed by testing it on a set of unseen patterns with known classes. We shall discuss unsupervised pattern recognition and supervised pattern recognition in Section 3.2 and 3.3, respectively.

In traditional pattern recognition systems, an object is classified into one known class. The main disadvantage of traditional (crisp) set theory is that it implies an aura of precision and definiteness for a decision that may not be warranted. In real-world systems, it is difficult to arrive at a crisp distinction always. Often data in a "boundary" condition can contain samples that could be said to be a member of more than one class. It can very well happen that a boundary condition sample X may be assigned to class A using crisp technique T1, but to class B using another crisp technique T2, and there is no way to gauge from the results which assignment is more appropriate.

Fuzzy pattern recognition systems on the other hand assign fuzzy membership values to all classes for all objects. Let us consider an example where existing bank customers have to be potentially classified into two groups – loan-worthy or non loan-worthy customers, based on their past transaction patterns. While there may be a well-defined set of features that can decide whether a customer is loan-worthy or not, it is not likely that a customer will necessarily satisfy all the properties of any one or the other class. In such a scenario, it is more practical to go for a fuzzy classification and exploit the fuzzy membership values to the two classes as some kind of confidence measure.

An unsupervised technique, on the other hand, does not use a given set of pre-classified data points. Since there is no pre-defined set of classes to associate patterns, this process is usually applied for grouping similar patterns together. Clustering is one of the primary techniques for grouping similar patterns together. We have already discussed Fuzzy clustering in Unit 2. In this unit, we will discuss some applications of clustering to unsupervised pattern recognition. Supervised pattern recognition and knowledge-based pattern recognition using fuzzy logic will follow in Section 3.4. Finally, we describe hybrid approaches and an application of hybrid fuzzy systems in medical image segmentation in Section 3.5. At the end we shall discuss multivalued recognition system in Section 3.6.

Objectives

After studying this unit, you should be able to:

- know the essence of the concept of pattern recognition;
- know what different types of PR systems are;
- differentiate between the supervised and unsupervised pattern recognition systems;
- know the essence of the hybrid system to solve complex PR problems;
- apply fuzzy logic to solve real-life PR problems.

3.2 UNSUPERVISED PATTERN RECOGNITION

Unsupervised pattern recognition is motivated by the need to find interesting patterns or groupings in a given set of data, the class labels of which are not known. For example, the manager of a supermarket may be interested in collecting data about a group of customers and analyzing it to find interesting grouping of customers on the basis of their buying patterns. The result of such an analysis can be used by a company to develop its marketing strategy or to develop a family of product lines that are targeted toward various customer groupings.

Image segmentation refers to partitioning pixels on an image into regions that correspond to different objects or different faces of objects in the images. Image segmentation can be viewed as data clustering problem where each datum is described by a set of image features like intensity, color, texture, etc. of a pixel. Conventional clustering algorithm generally finds a "hard partition" of a given dataset but in many real-world clustering problems, some data points partially belong to multiple clusters, rather than to a single cluster exclusively. For example, a pixel in a Magnetic Resonance Image (MRI) may correspond to a mixture of two different types of tissues. Another area which uses a lot of fuzzy clustering is that of customer behavior analysis. Clustering aims at finding similar groups of customers to target a particular product or promotional campaign. However, a particular customer may be a "borderline case" between two groups of customers (e.g., between moderate conservatives and moderate liberals). Fuzzy clustering provides a method of handling such borderline or ambiguous cases with ease. "Fuzzy clustering" algorithms attempt to find a "soft partition" of a given dataset in which a datum can partially belong to multiple clusters. One of the most popular fuzzy clustering algorithms, Fuzzy c-Means algorithm, has been already discussed in Unit-2. In the following sub-section an application of this algorithm is presented.

Applications of Fuzzy Clustering to Image Segmentation

One of the oldest applications of fuzzy clustering techniques has been to Image Segmentation. Algorithms such as fuzzy c-means (Bezdek, 1981) have been used to build clusters (segments). The class membership of pixels can be interpreted as similarity or compatibility with an ideal object or a certain property. Prior to applying fuzzy clustering techniques for image segmentation, it is necessary to perform image processing using fuzzy logic. This occurs in three stages (http://pami.uwaterloo.ca/tizhoosh/segment.htm). The first step is known as *image fuzzification,* and this is used to modify the membership values of a specific data set or the image. This step results in the transformation of the image data from gray-level plane to the membership plane. In the second step, appropriate fuzzy techniques are applied to modify the membership values. These techniques could be based on fuzzy clustering, a fuzzy rule-based approach, or a fuzzy integration approach. The final stage comprises of decoding of the results, called *defuzzification*. This step results in an output image. The main power of fuzzy image processing lies in modification of the fuzzy membership values. Fig. 1 below shows the steps involved. The roles of these steps are further explained through the following example.

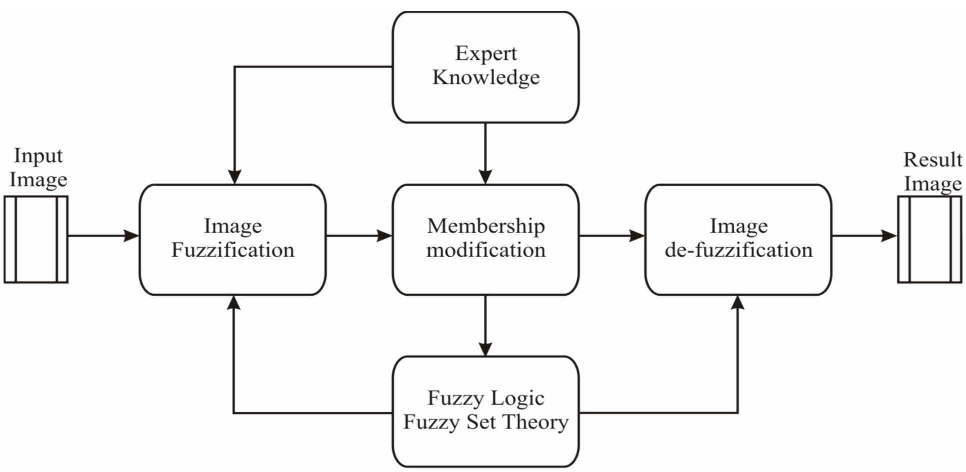

Fig. 1: Steps in Fuzzy Image Processing.

Example 1: Let us suppose an image comprises of grayscale values ranging from 0 to 255. It is required to segment the image into two regions – one consisting of the darker regions in the image and the other containing the comparatively lighter one. The image has different levels of darkness, and in classical set theory, a threshold such as the gray level 100 has to be set, so that if a pixel's gray-level value is less than 100 it will fall in the dark region, otherwise in the lighter region. But since the darkness of a particular pixel is a matter of degree, a fuzzy set (or subset to be precise) can model this property much better. To define this subset, two thresholds, say gray levels 50 and 150 are required. Then all the gray levels that are less than 50 are full member of the set dark, and all gray levels greater than 150 are definitely not members of the set. Gray levels between 50 and 150, however, have a partial membership to the set dark.

In fig. 2, a test image containing a few machine parts are shown. It is required to segment these parts out from the background. It is obvious that since the parts have different levels of grayness, it is difficult to choose a threshold value by inspection, which can separate the black background from the foreground objects. Fuzzy image processing can be used here to do the job of separating the background from the foreground objects successfully.

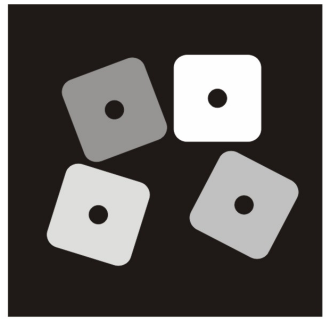

Fig. 2: Test Image for segmenting into background and foreground objects.

Fig. 3 below shows how fuzzy membership using standard S function can be used to decide the threshold for segmentation. The membership function, S function in this case, is moved pixel by pixel over the existing range of gray levels. In each position, a measure of fuzziness is calculated. The S function operates on each pixel with co-ordinate (m,n), and gray-scale value μ_{mn} as follows:

$$\mathbf{S}(\boldsymbol{\mu}_{\mathbf{mn}}) = -\boldsymbol{\mu}_{\mathbf{mn}} \, \mathbf{ln}(\boldsymbol{\mu}_{\mathbf{mn}}) - (\mathbf{1} - \boldsymbol{\mu}_{\mathbf{mn}}).$$

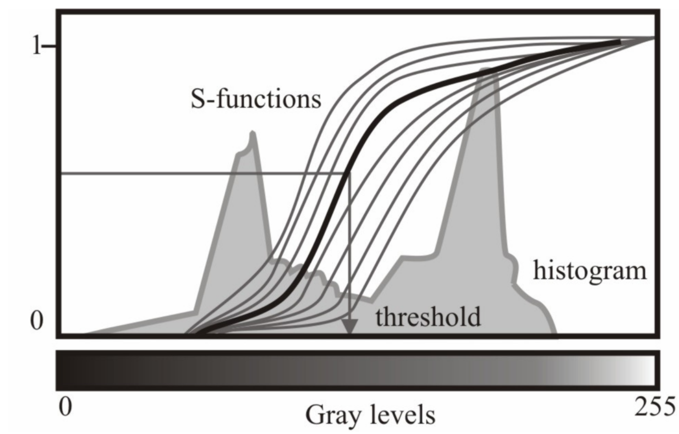

Fig. 3: Choosing threshold using S function for Image fuzzification.

The position with a minimum amount of fuzziness, as shown in the figure, is then chosen as the threshold. All pixels with fuzzy value greater than this threshold is assigned the value 255, while all pixels with fuzzy value less than this is assigned the value 0. The resulting image is shown in fig. 4. The image is clearly segmented into the background and the foreground objects.

The resulting image is shown in fig. 4. The image is clearly segmented into the background and the foreground objects.

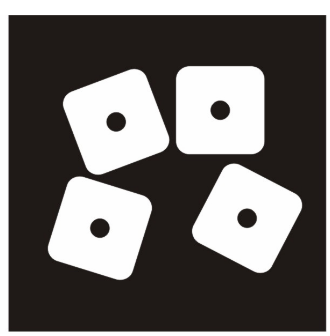

Fig. 4: Segmented image obtained using fuzzy techniques.

Now, let us focus on supervised pattern recognition.

3.3 SUPERVISED PATTERN RECOGNITION

In the previous section, we have discussed unsupervised pattern recognition technique which does not require points in the dataset to be labeled with their correct classification. In contrast, supervised pattern recognition uses data with known classifications, which are also called labeled data, to determine the classification of new data.

Supervised and unsupervised pattern recognition techniques differ in their advantages and limitations. The main benefit of unsupervised pattern recognition technique is that they do not require training data; however, their computation time is relatively high due to a large number of iterations needed before the algorithm converges. In contrast, a supervised pattern recognition technique is relatively fast because it does not need to iterate; however, it cannot be applied to a problem unless training data or relevant knowledge are available.

Now, let us discuss fuzzy k-nearest neighborhood.

Nearest neighbour (NN) classification techniques classify an unknown sample by comparing it to its nearest neighbors among a set of known samples. Any distance metric can be used for computing the nearest neighbours of an object, as long as it applies consistently to all samples in the set. An unknown sample is assigned membership to the class most represented by its K nearest neighbours.

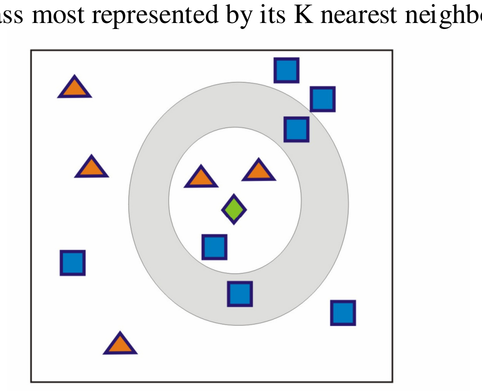

Fig. 5: Classification problem–Two classes of objects are known: triangles and squares. Find the class of the unknown object diamond shown in the centre.

Fig. 5 shows how a classification task is posed and solved using K–NN. Let us suppose, the system is trained to recognize two classes of objects–triangles and squares. In order to determine the class of an unknown object, the diamond, using K– NN and setting K to 3, it can choose three of its closest neighbours as shown by the inner circle. In this case, the class assigned to the diamond would be triangle. However, if K is chosen to be 5, then the diamond will be classified as a square.

Crisp K-NN classification techniques can be computationally quite simple and can run in O(N) time. However, crisp K–NN classification techniques suffer from major drawbacks as elaborated below:

a) An object can be classified into only one class based on the class of majority of its neighbours. Noise in the data can really lead to a very erroneous classification in this case, since all neighbours are given equal weights.

b) In case of a tie among more than one class, decision has to be arbitrarily taken in favour of one class or the other without any rationale.

For fuzzy K-NN, an unknown sample's membership to each class is assigned based on the neighbour's class and the distance of the neighbour from the unknown sample.

In essence, membership of object ***x*** to class ***i*** is given by the following equation, where K is the total number of neighbours considered.

$$\mu_i(x) = \frac{\sum_{j=1}^{k} \mu_{ij} \left(\frac{1}{\| x - x_j \|^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^{k} \left(\frac{1}{\| x - x_j \|^{\frac{2}{m-1}}} \right)}$$

The denominator in the above equation acts as a normalizing factor. This factor decides how the distance of a neighbour from the sample should influence the classification decision. When m is chosen to be greater than 1, the farther a neighbour is from an object, the less is its influence on the classification decision. For m greater than 1, this effect reduces exponentially. Though there is no theoretical study on the appropriate value of m to be chosen, m is usually chosen to be 2.

In the above example, the squares and the triangles would contribute almost equally to the class of the unknown sample diamond, and the object would have memberships to both classes, which is more sensible in the given scenario.

In the following section, we shall discuss knowledge-based pattern recognition.

3.4 KNOWLEDGE-BASED PATTERN RECOGNITION

A knowledge-base comprises an information store and an inference engine for making decisions. Fuzzy pattern recognition techniques have been very successful in implementing Knowledge-based systems for analyzing sensor data, where the nature of the data is such that non-fuzzy techniques cannot isolate patterns effectively. Sensor data is usually noisy. In this section, we shall discuss an application where fuzzy techniques have been successfully employed for real-time decision making in an automated manufacturing unit (Singh and Steinl, 1996).

Automation of factories is principally related to the automation of individual processes and decision making related to their operation. Decision making systems are typically designed as knowledge-based systems, where making decisions is based on a set of conditions and rules. The conditions are affected by the sensor data. These systems can become very complicated as their size increases. The increase in size is directly proportional to the number of components, the number of sensor measurements per component and the number of sensors. Large knowledge-bases are expensive to manage and modify. Decision-making in a large set-up is also complex, and the task of real-time decision making is challenging since decisions need to be made in such a manner that there are no bottlenecks on the production-line. Fuzzy search technique is related to the concept of possibility measurements in fuzzy logic, which can handle incomplete and imprecise data and perform fast searches over a large database.

In order to help real-time decision making, the first step is to reduce the amount of information needed for decision making. Since sensor data is noisy, the first step is to reduce the dataset into bandwidths rather than deal with a large number of unique sensor outputs. For example, if A and B are two sensors, whose outputs can be any real value in the range of 1 to 10 with resolution 0.1, then rather than have rules like *If* $(A = 1 \;\&\; B = 2)$ *OR* $A = 1.1 \;\&\; B = 2)$. *Then Component* 1, a rule which states *If* $(.9 < A < 1.1 \;\&\; 1.9 < B < 2.1)$. *Then Component* 1, is obviously more compact and useful. This is a simplistic representation which can be further improved as using the concept of possibility distribution as explained below.

The basic concept of a possibility distribution derives from the fact that the mean of a sample has the maximum possibility of occurrence, and this possibility decreases quadratically as one moves away from the mean. For example, let us suppose that for a total of 100 men, their heights are stored in a set H. Let H_{min} represent the minimum height, H_{max} represent the maximum height and H_{mean} the mean height of this set. If it was required to guess a new person's height, based on experience about men's height from the set H, then it is possible to make an assessment in a way such that the assessed height has a membership value which lies between 0 and 1, by making use of possibility distribution function. The possibility distribution function looks similar to a normal curve. Fig. 6 below illustrates the possibility distribution function along with membership function formulae.

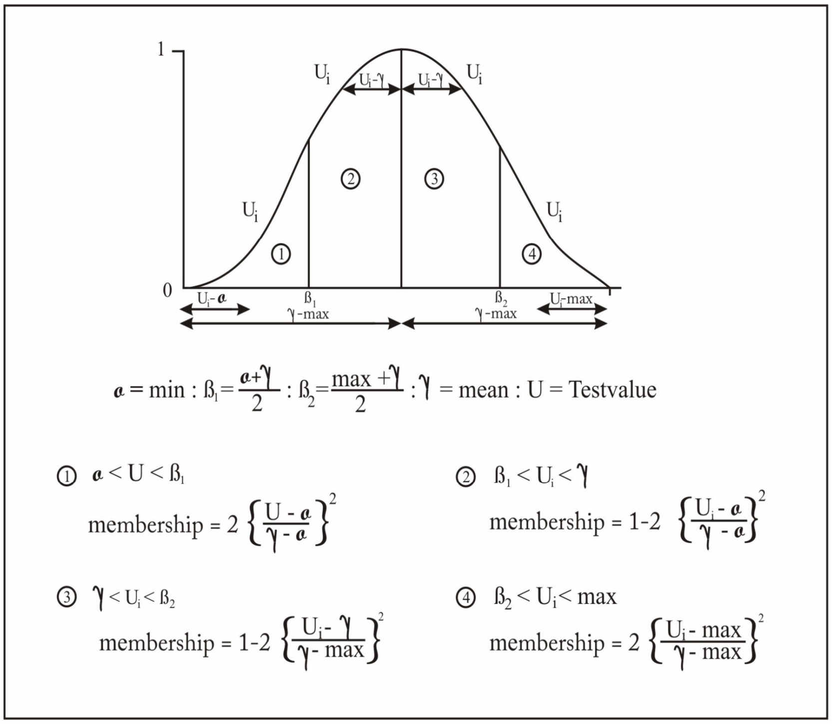

Fig. 6: Possibility Distribution Function for calculating memberships for new values.

The possibility distribution function can be computed to interpret sensor data in real-time, where each sensor has a minimum, mean and a maximum that is stored in the knowledge base. U_i represents a new sensor data, which has to be interpreted by the inference engine.

Let us suppose a manufacturing unit has **N** components ($C_1, C_2, \ldots, C_N$). Let each component record $\mathbf{k_{max}}$ readings for each sensor to account for sensor reliability, uncertainty, noise, etc. Thus sensor reading for component C_1 has (C_{11}, C_{12}, C_{1kmax}). Each reading consists of past $\mathbf{i_{max}}$ sensor measurements. A new reading $X = (X_1, X_2, \ldots, X_{imax})$ has to be interpreted by the system for taking a decision.

Initially, the system computes for column **i** of component **j**, the minimum, maximum and mean of its k_{max} readings and stores them as min_{ij} , max_{ij} and $mean_{ij}$, respectively. Since this is all that it needs for computing the possibility of a new reading, the actual size of the original data set has been reduced by a factor of $k_{max}/3$. For the individual measurements of X, the possibility of occurrence of each value is computed with respect to each component, is computed as a membership value between (0,1). The membership of new reading X_i with respect to the sensor **i** of component **j** is computed as $M(X_i, C_{ij})$, using the possibility distribution function.

In order to determine the component that readings of vector X represent, the overall possibility for each row has to be computed. This is achieved by calculating a cumulative index of possibility for each component by combining the individual values using the following function:

$$I_j = e^{M(x_1, C_{1j})} + \cdots + e^{M(x_1, C_{1j})} + \cdots + e^{M(x_{i_{max}}, C_{i_{max}})}$$

Index I_j is computed for each row in the knowledge base and the component corresponding to the highest index is assumed to represent X.

Now, let us discuss hybrid pattern recognition systems.

3.5 HYBRID PATTERN RECOGNITION SYSTEMS

The field of pattern recognition has seen enormous progress since its beginning more than 5 decades ago. Over the years various approaches have been emerged, based on statistical decision theory, structural matching and parsing, neural networks, fuzzy logic, artificial intelligence, evolutionary computing and others. Obviously, these approaches are characterized by a high degree of diversity. In order to combine their strengths and avoid their weaknesses, hybrid pattern recognition systems that combine several techniques together have come into existence. The major applications areas of hybrid pattern recognition system include on-line and off-line handwriting recognition, remotely sensed image interpretation, fingerprint identification, and automatic text categorization. In the following sub-section we will present neuro-fuzzy systems that have been successfully applied in medical domain. A neuro-fuzzy system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples.

3.5.1 Combining Fuzzy Systems with Neural Networks

Both neural networks and fuzzy systems have some things in common. They can be used for solving a problem (e.g. pattern recognition, regression or density estimation) if there does not exist any mathematical model of the given problem. They solely do have certain disadvantages and advantages which almost completely disappear by combining both concepts.

Neural networks can only come into play if the problem is expressed by a sufficient amount of observed examples. These observations are used to train the *black box*. On the one hand no prior knowledge about the problem needs to be given. On the other hand, however, it is not straightforward to extract comprehensible rules from the neural network's structure.

On the contrary, a fuzzy system demands linguistic rules instead of learning examples as prior knowledge. Furthermore the input and output variables have to be described linguistically. If the knowledge is incomplete, wrong or contradictory, then the fuzzy system must be tuned. Since there is not any formal approach for it, the tuning is performed in a heuristic way. This is usually very time consuming and error-prone.

Table 1 highlights the relative similarities and dissimilarities of the two approaches. Neural networks can learn from data, but cannot be interpreted - they are black boxes to the user. Fuzzy Systems consist of interpretable linguistic rules, but they cannot learn. It is desirable for fuzzy systems to have an automatic adaption procedure which is comparable to neural networks. We use learning algorithms from the domain of neural networks to create fuzzy systems from data. The learning algorithms can learn both fuzzy sets, and fuzzy rules, and can also use prior knowledge.

Table 1: Comparison of Neural and Fuzzy systems

Neural Networks	Fuzzy Systems
no mathematical model necessary	no mathematical model necessary
learning from scratch	a-priori knowledge essential
several learning algorithms	not capable to learn
black-box behaviour	simple interpretation and implementation

3.5.2 Neuro-Fuzzy System or Fuzzy Neural Network

In the field of artificial intelligence, *neuro-fuzzy* refers to combinations of artificial neural networks and fuzzy logic. Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Neuro-fuzzy hybridization is widely termed as Fuzzy Neural Network (FNN) or Neuro-Fuzzy System (NFS) in the literature. Neuro-fuzzy system (the more popular term is used henceforth) incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. The main strength of neuro-fuzzy systems is that they are universal approximators with the ability to solicit interpretable IF-THEN rules.

The strength of neuro-fuzzy systems involves two contradictory requirements in fuzzy modeling: interpretability versus accuracy. In practice, one of the two properties prevails. The neuro-fuzzy in fuzzy modeling research field is divided into two areas: linguistic fuzzy modeling that is focused on interpretability, mainly the Mamdani model; and precise fuzzy modeling that is focused on accuracy, mainly the Takagi-Sugeno-Kang (TSK) model.

Characteristics of NFS

Compared to a common neural network, connection weights and propagation and activation functions of fuzzy neural networks differ a lot. Although there are many different approaches to model a fuzzy neural network, most of these agree on certain characteristics such as the following:

- A neuro-fuzzy system based on an underlying fuzzy system is trained by means of a data-driven learning method derived from neural network theory. This heuristic only takes into account local information to cause local changes in the fundamental fuzzy system.
- It can be represented as a set of fuzzy rules at any time of the learning process, i.e., before, during and after. Thus the system might be initialized with or without prior knowledge in terms of fuzzy rules.
- The learning procedure is constrained to ensure the semantic properties of the underlying fuzzy system.
- A neuro-fuzzy system approximates a n-dimensional unknown function which is partly represented by training examples. Fuzzy rules can thus be interpreted as vague prototypes of the training data.
- A neuro-fuzzy system is represented as special three-layer feedforward neural network as it is shown in Fig. 7.
    - ♦ The first layer corresponds to the input variables.
    - ♦ The second layer symbolizes the fuzzy rules.
    - ♦ The third layer represents the output variables.
    - ♦ The fuzzy sets are converted as (fuzzy) connection weights.

- Some approaches also use five layers where the fuzzy sets are encoded in the units of the second and fourth layer, respectively. However, these models can be transformed into a three-layer architecture.

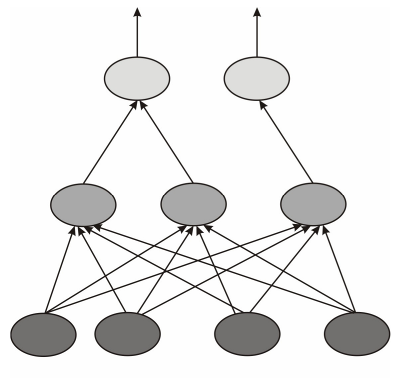

Fig. 7: The layers in a neuro-fuzzy system.

3.5.3 Hybrid Neuro-Fuzzy System or Fuzzy Neural Network

Hybrid neuro-fuzzy systems are homogeneous and usually resemble neural networks. Here, the fuzzy system is interpreted as a special kind of neural network. The advantage of such hybrid NFS is its architecture since both fuzzy system and neural network do not have to communicate any more with each other. They are one fully fused entity. These systems can learn online and offline. Fig. 8 shows such a hybrid FNN.

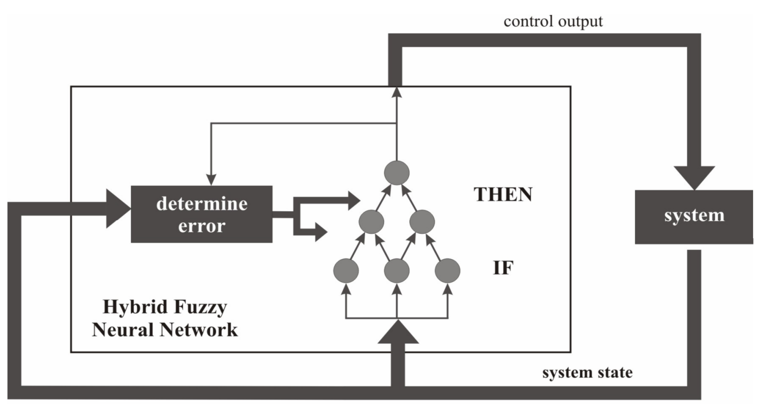

Fig. 8: The architecture of a hybrid fuzzy neural network.

The rule base of a fuzzy system is interpreted as a neural network. Fuzzy sets can be regarded as weights whereas the input and output variables and the rules are modeled as neurons. Neurons can be included or deleted in the learning step. Finally, the neurons of the network represent the fuzzy knowledge base. Obviously, the major drawbacks of both underlying systems are thus overcome.

In order to build a fuzzy controller, membership functions which express the linguistic terms of the inference rules have to be defined. In fuzzy set theory, there does not exist any formal approach to define these functions. Any shape (e.g., triangular, Gaussian) can be considered as membership function with an arbitrary set of parameters. Thus, the optimization of these functions in terms of generalizing the data is very important for fuzzy systems. Neural networks can be used to solve this problem.

By fixing a distinct shape of the membership functions, say triangular, the neural network must optimize their parameters by gradient descent. Thus, apart from the information about the shape of the membership functions, training data must also be available.

Another approach is to group the training data $\{(X_i, y_i) | x_i \in \chi,\ y_i \in Y,\ i = 1, 2, \cdots, l\}$ into M clusters. Every cluster represents a rule R_m where m = 1, 2 ,…, M. Hence these rules are not defined linguistically but rather by crisp data points X = $(x_1, x_2, \ldots, x_n)$ $X = (x_1, x_2, \cdots, x_n)$. Thus, a neural network with n input units, hidden layers and M output units might be applied to train on the pre-defined clusters. For testing, an arbitrary pattern x is presented to the trained neural network. Every output unit m will return a degree to which extent x may fit to the antecedent of rule R_m.

To guarantee the characteristics of a fuzzy system, the learning algorithm must enforce the following mandatory constraints:

- Fuzzy sets must stay normal and convex.
- Fuzzy sets must not exchange their relative positions (they must not *pass* each other).
- Fuzzy sets must always overlap.

Additionally there do exist some optional constraints like the following:

- Fuzzy sets must stay symmetric.
- The membership degrees must sum up to 1.

The ARIC (Approximate Reasoning-based Intelligent Control) is a fuzzy neural network model where a prior defined rule base is tuned by updating the network's prediction. Thus the advantages of fuzzy systems and neural networks are easily combined. ARIC is represented by two feed-forward neural networks, the Action-state Evaluation Network (AEN) and the Action Selection Network (ASN). ASN is a multilayer neural network representation of a fuzzy system. It consists of two separate modules. The first one represents the fuzzy inference and the second one computes a confidence measure based on the current and next system state. Both parts are eventually combined to the ASN's output.

As it is shown in Fig. 8, the first layer represents the rule antecedents, whereas the second layer corresponds to the implemented fuzzy rules and the third layer symbolized the system action. The network flow is as follows. In the first layer the system variables are fuzzified. In the next step these membership values are multiplied by the attached weights of the connections between the first and second layer. In the latter layer, every rule's input corresponds to the minimum of its input connections.

A rule's conclusion is installed as membership function. This function maps the inverse rule input value. Its output values are then multiplied by the weights of the connections between second and third layer. The final output value is eventually computed by the weighted average of all rule's conclusions.

The AEN (which is as three-layer feed-forward neural network as well) aims to forecast the system behaviour. The hidden layer obtains as input both the system state and an error signal from the underlying system. The output of the networks shall represent the prediction of the next reinforcement which depends on the weights and the system state. The weights are changed by a reinforcement procedure which takes into consideration the outputs of both networks ASN and AEN. ARIC was successfully applied to the cart-pole balancing problem.

Whereas the ARIC model can be easily interpreted as a set of fuzzy-if-then rules, the use of ASN network to adjust the weights is rather difficult to understand. It is a working neural network architecture that utilizes aspects of fuzzy systems. However, a semantic interpretation of some learning steps is not possible.

3.5.4 Applications in Medical Image Segmentation

Image segmentation is defined as the partitioning of an image into non-overlapping regions that are homogeneous with respect to some characteristic such as intensity, which does not easily happen in the medical image because most of medical images have a tissue (overlapping region). Image segmentation has its root in many real-life problems, especially medical applications. The imaging modalities can be divided into two global categories: anatomical and functional. Anatomical modalities, depicting primarily morphology, include X-ray, CT (Computed Tomography), MRI (Magnetic Resonance Imaging), US (ultrasound), portal images, and (video) sequences. Image segmentation algorithms play a role in biomedical imaging applications such as the quantification of tissue volumes diagnosis, localization of pathology study of anatomical structure, treatment planning, partial volume correction of functional imaging data, and computer integrated surgery. The methods for performing segmentations vary widely depending on the specific application. For example, the segmentation of brain tissue has different requirements from the segmentation of the liver. In general, there are two main approaches to clustering–crisp clustering and fuzzy clustering. One of the popular fuzzy clustering techniques proposed in literature for medical image segmentation is fuzzy c-means (FCM) algorithm. The neural network based approaches have been also introduced in literature for medical image segmentation. The use of fuzzy c-means algorithm with Kohonen neural network for MRI medical image segmentation can help to recognize the tumor region and its characteristics. Tsao et al., 1994 have extended the ideas of FCM and Kohonen neural network to a new family of algorithms called Fuzzy Kohonen Clustering Network (FKCN). The Kohonen Clustering Network (KCN) clustering is closely related to the Fuzzy C-Means (FCM) algorithms. Since Fuzzy C-Means algorithms are optimization procedure because the objective function is approximately minimized, the integration of FCM and KCN is one way to address several image segmentation problems. They combine the ideas of fuzzy membership values for learning rates, the parallelism of Fuzzy C-Means, and update rules of KCN. FKCN is a self-organizing algorithm, since the "size" of the updated neighbourhood is automatically adjusted during learning, and FKCN usually terminates in a way of minimized objective function of FCM. Fig. 9 illustrates the application of FKCN to identify brain tumor in a magnetic resonance image of a brain.

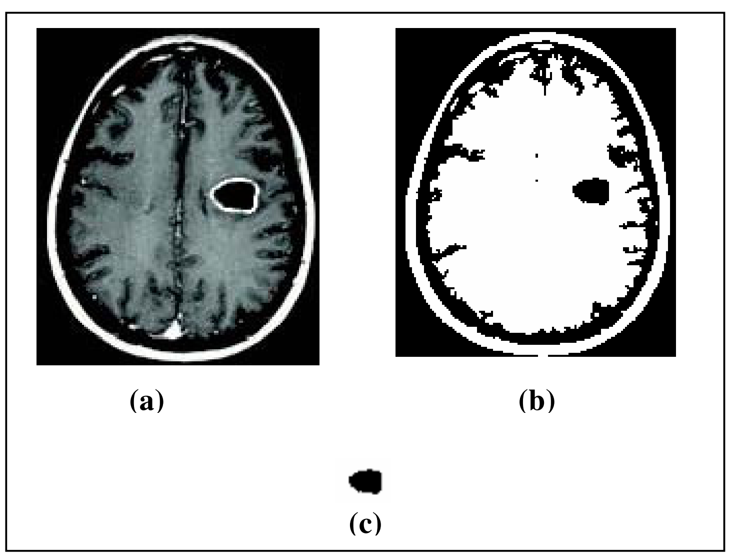

Fig. 9: (a) Magnetic Resonance Image (MRI) of a brain; (b) MRI of the brain after applying Fuzzy Kohonen neural network; and (c) Identified tumor.

Now, let us focus on multivalued recognition systems.

3.6 MULTIVALUED RECOGNITION SYSTEMS

Fuzzy logic is a form of multi-valued logic that is derived from fuzzy set theory. In fuzzy logic the degree of truth of a statement is represented by a value in the range of (0,1) both inclusive. This is in contrast to predicate logic which allows for only two truth values (TRUE, FALSE). Fuzzy logic allows for reasoning that is approximate rather than precise.

To begin with, let us consider designing a system based on anti-lock brakes. The anti-lock brakes are controlled by temperature. The anti-lock brakes have membership functions that control its function depending on different temperature ranges. Fig. 10 below defines how temperature ranges can be used to define membership values to three categories – cold, warm and hot. Fig. 10 states that, low temperature values indicate high membership values for cold and low membership values for hot and warm, medium temperatures indicate high membership value for warm and low membership values for both hot and cold, and high temperature indicates high membership value to hot and low membership values to cold and warm.

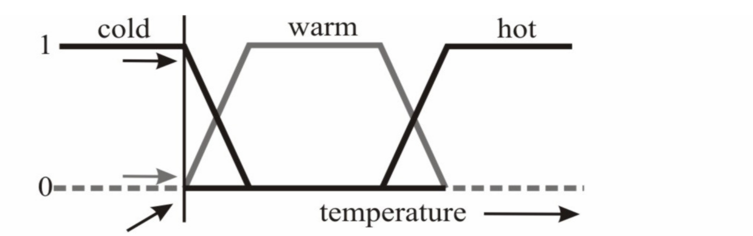

Fig. 10: Fuzzy membership functions to ascertain cold, hot and warm depending on temperature.

Fuzzy logic based systems are usually implemented using IF-THEN rules, though they can be implemented in other ways also. A rule is written in the form of "IF *variable* IS *property* THEN *action"*. It may be noted that these rules do not have any ELSE condition.

A simple temperature regulator connected to a fan may be designed using Fuzzy logic using the following rules:

IF *temperature* IS **COLD** then *step down speed of fan*

IF *temperature* IS **WARM** then *maintain normal speed of fan*

IF *temperature* IS **HOT** then *speed up fan*

Given a particular value of temperature, it is possible that more than one antecedent is fired with value greater than 0. For example if the temperature is 15 degrees, both cold and warm have non-zero membership values. Thus, the antecedents of both the first and the second rules have the potential to be fired. The final action will depend on the maximum truth value that the system attains.

The AND, OR and NOT operators are defined in fuzzy logic using *min*, *max* and *complement*. For fuzzy variables *x* and *y*, the truth values of combinations of *x* and *y* are defined as follows:

Truth(x AND y) = min (truth(x), truth(y))

Truth(x OR y) = max (truth(x), truth(y))

Truth(NOT(x)) = 1 – truth(x)

Fuzzy logic has been successfully used in edge detection for medical images. Lalande et al. [2] described a system for automatic detection of cardiac contours in MRI images using fuzzy logic. Since the heart is not an isolated organ, traditional edge detection operators do not perform well in such images.

The system uses two parameters. The first parameter ***p*** is estimated from the image using the logic that all pixels located on the cardiac contour will have roughly the same gray-value. To determine the gray level value of the contour pixels, the user indicates a point near the centre of the cardiac left ventricle. Then, a series of radial lines is automatically drawn. Along each line, a Gaussian edge operator (Lalandel et al., 1997) is used to detect the first edge between a high-intensity area and a low-intensity area (i.e. contour between the cardiac cavity and the heart muscle). Along each line the pixel where the Gaussian operator yields the highest value is noted and the gray level value of this pixel is retrieved. The extracted gray level value ***p*** of the endocardial contour is the maximum of the histogram of these local gray level values obtained along all radial lines.

Once the value ***p*** is estimated, all other pixels are compared to this value. The second parameter is computed based on the presence of edges, using fuzzy logic. For a given gray level, the closer the value is to the estimated contour gray value, the greater the likelihood that the component belongs to the myocardium. Fig. 11 describes the membership functions used for this purpose. The edge contour is finally found by linking the edge pixels and detecting the best path. Fig. 12 highlights the results.

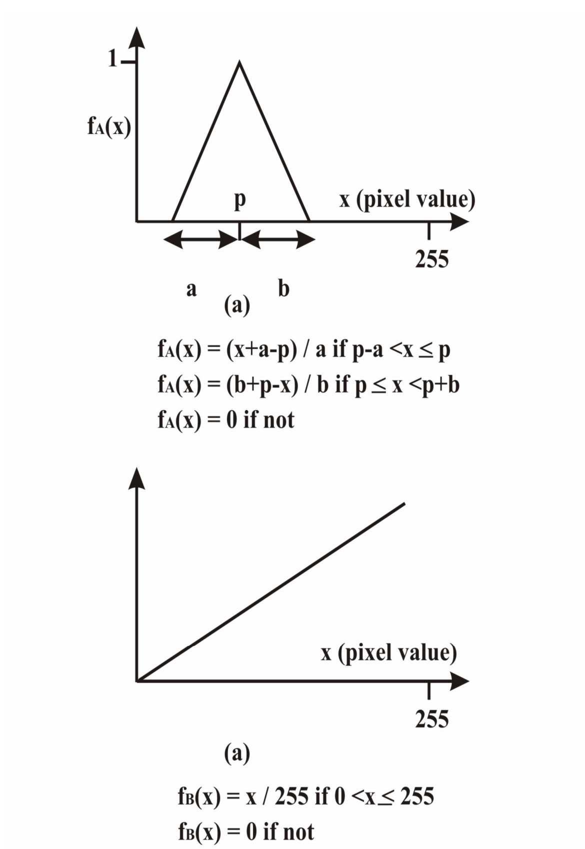

Fig. 11: (a) Membership function to determine whether a pixel belongs to the myocardium (A) ; and (b) Membership function to determine whether a pixel belongs to the cardiac contour.

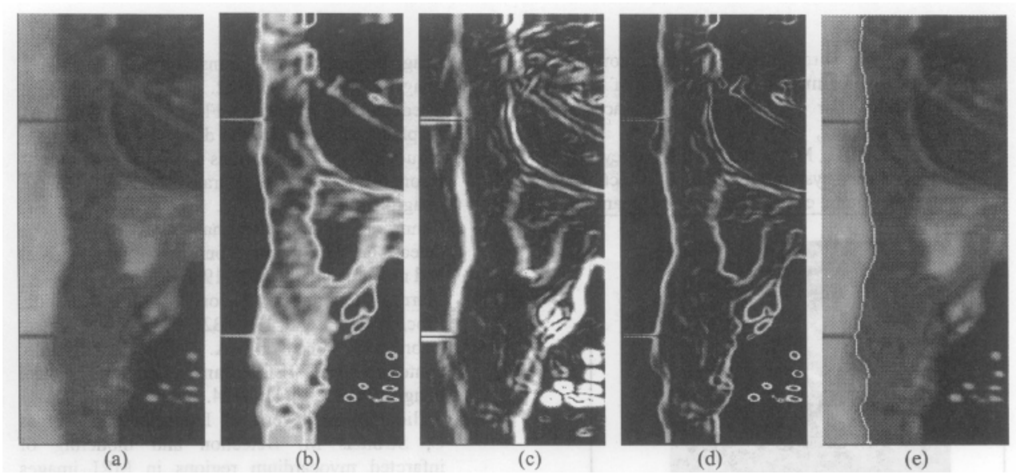

Fig. 12: (a) Initial image; (b) Membership degrees associated with gray level pixels; (c) Membership degrees associated with edges; (d) Edge pixels; and (e) Final edge contour after linking.

Now try the following exercises.

E1) John Doe is a medical doctor interested in statistics. He uses a certain test to determine if his patient Mr. Saha has cancer. However, the test is not perfectly reliable. In fact, the test succeeds to reveal cancer only with probability 0.98. In addition, when the patient does not have cancer, the test has a probability of 0.03 to be positive. It is also known that the probability of cancer in the overall population is 0.008.

What is the probability that Mr. Saha has cancer when the test is positive? After the positive result, John Doe decides to perform the test once more. The second test also results positive. What is the probability of cancer after the second test? Assume that the results of the tests are independent.

E2) Consider two normally distributed probability distributions

$$p(x|w_i) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma}\right)^2\right]$$

with equal deviations $\sigma = 1$ and a priori probabilities $P(\omega_1) = P(\omega_2)$. Determine a classifier with a minimum classification error.

Now, let us summarize the unit.

3.7 SUMMARY

In this unit, we have discussed the fundamental concepts of the fuzzy pattern recognition and its various real-life especially medical applications. We have also discussed how the supervised pattern recognition differs with unsupervised pattern recognition process. Specifically we have covered the following:

1. Elaborated the essence of pattern recognition and its applications to solve real life complex problems.
2. Explained both supervised and unsupervised processes with examples for pattern recognition.

3. Discussed knowledge-based pattern recognition system that uses domain knowledge to improve the quality of classification process.
4. Discussed how fuzzy system and neural networks can be combined into a hybrid pattern recognition to combine the strengths of both approaches for solving complex pattern recognition problems.
5. Provided details about multivalued recognition system with an example from biomedical domain to highlight its essence.

3.8 SOLUTIONS/ANSWERS

E1) Use Bayes formula to calculate the a posterior probabilities.

E2) Using the above formula for normal distribution, derive the equation for the posterior probability $P(\omega_i \mid x)$. Calculate then the logarithmic discriminant function $g_i(x) = \log P(\omega_i \mid x)$

Use the result to determine the decision boundary for the classifier. That is, determine x where $g_1(x) = g_2(x)$.

Verify the result using values $\mu_1 = 2$ and $\mu_2 = 4$.

3.9 PRACTICAL ASSIGNMENTS

Session 3

Write a program in C language to construct a function *C = knn(trainclass, traindata, data, k)* that performs k-nearest-neighbour classification. Matrix *traindata* contains training examples so that each column is a single example. Row vector *trainclass* contains the classes of the examples, so that element i of *trainclass* is the class of the example in column i of *traindata*. Matrix data contains samples to be classified, one in each column, and **k** is a parameter which tells number of nearest neighbours used in classification. Return value **C** is a row vector of classes and it should include one value for each column in data. Use Euclidean distance as distance measure.

Algorithm for k-nearest-neighbour classification can be described as follows:

- Find k nearest neighbours from the training set for a sample to be classified.
- Classify the sample to the class which has most training samples among the k nearest neighbours.

Session 4

Write a program in C language to implement a function c = cmeans (data, k) that performs c-means clustering for given data. Matrix data contains training examples so that each column is a single example. Scalar **k** is a parameter which tells the desired number of clusters. Return value c is a matrix which contains means of clusters, one in each column.

One of the simplest, most well-known, and most used clustering techniques is *c-means* which is also known as ISODATA and k-means. The c-means algorithm is as follows:

1. Choose the number of clusters.
2. Initialize the seed vectors for clusters.
3. Cluster the data according to the shortest distance to cluster means.
4. Calculate a mean vector for each cluster and set them as new mean vectors.

5. If there are changes in the mean vectors between iterations i and i-1, then goto 3. Means of clusters $\mu_1, \mu_2, ..., \mu_C$ must have some initial values. These can be obtained by randomly choosing c samples from the data, and assigning $\mu_1, \mu_2, ..., \mu_C$ to the values of these samples.

Session 5

Visualization of C-means clustering

Write a program in C language to modify your cmeans function created in session 4 so that progress of clustering process can be seen. This means that your function should plot original data and original guesses for means of clusters. After each iteration your function should show samples belonging to each cluster with different symbols and/or colors, and your function should also show trajectories of means of clusters starting from initial guesses and ending to final values.

3.10 REFERENCES

1. J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York.

2. S. Singh and M. Steinl (1996): Fuzzy Search Techniques in Knowledge-Based Systems, Proc. 6th International Conference on Data and Knowledge Systems for Manufacturing and Engineering (DKSME'96), Tempe, Arizona, USA, pp. 1-10.

3. A. Lalandel, L. Legrand, P. Walker, M. Jaulent, F. Guy, Y. Cottin, F. Brunotte (1997): Automatic detection of cardiac contours on MR Images using fuzzy logic and dynamic programming, Proc AMIA Annual Fall Symposium, pp. 474–478.

4. Eric Chen-Kuo Tsao, Jamec C.Bezdek and Nikhil R.Pal (1994): "Fuzzy Kohonen Clustering Networks, Pattern Recognition," Vol.27, No.5 pp.757-764.

5. S. Dick, S.K. Pal and S. Mitra (1999), Neuro-Fuzzy Pattern Recognition Methods in Soft Computing, IIE Transactions, John Wiley & Sons, New York, 375 pp ., ISBN 0-471-34844-9.