Module-1

Subject: Soft Computing

Content: Basic tools of soft Computing: Fuzzy logic, Neural Networks and Evolutionary Computing, Approximations of Multivariate functions, Non - linear Error surface and optimization., Fuzzy Logic Systems: Basics of fuzzy logic theory. Crisp and fuzzy sets; Basic set operations, Fuzzy relations, Composition of Fuzzy relations, Fuzzy inference, Zadeh's compositional rule of inference, Defuzzification, Fuzzy logic control, Mamdani and Takagi and Sugeno architectures, Applications to pattern recognition.

Prepared by

Dr. Subhranshu Sekhar Dash, Professor
&
Dr. Sangram Keshori Mohapatra, Associate professor

Department of Electrical Engineering,
Government College of Engineering, Keonjhar, Odisha, India

MODULE-I

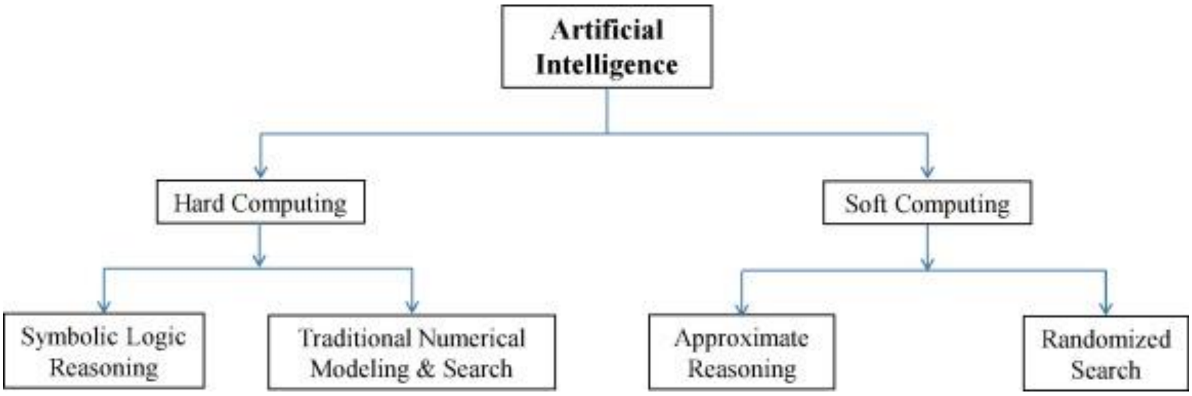
SOFT COMPUTING: -

Soft computing is a branch of computer science and artificial intelligence that deals with the development of intelligent systems that can handle imprecise, uncertain, or incomplete information. Soft computing techniques are used when traditional methods fail to provide satisfactory solutions, and when the available data is uncertain, noisy, or inconsistent.

Soft computing methods include fuzzy logic, neural networks, evolutionary algorithms, and probabilistic reasoning. These techniques can be used separately or in combination to solve complex problems that cannot be solved using traditional computing methods.

Fuzzy logic, for example, is a mathematical framework that deals with uncertainty and imprecision by assigning degrees of truth to statements. Neural networks, on the other hand, are biologically inspired models that can learn from data and make predictions. Evolutionary algorithms are optimization techniques that mimic the process of natural selection to find the best possible solution to a problem. Finally, probabilistic reasoning is a method for reasoning about uncertain events and making decisions based on probabilities.

Soft computing techniques have been applied to a wide range of applications, including pattern recognition, data mining, control systems, robotics, and decision making. They are particularly useful in situations where there is a high degree of uncertainty or ambiguity, and where human expertise and intuition are required to make informed decisions.



APPLICATIONS OF SOFT COMPUTING: -

Soft computing techniques are used in a wide range of applications, including:

1. Pattern recognition: soft computing techniques such as artificial neural networks, fuzzy logic, and genetic algorithms can be used for pattern recognition tasks, including image and speech recognition.
2. Control systems: soft computing techniques can be used to design control systems that can operate effectively in uncertain and dynamic environments.
3. Robotics: Soft computing techniques can be used to develop intelligent robots that can navigate and perform tasks in complex environments.
4. Data mining: soft computing techniques can be used to analyse large datasets and extract useful information.
5. Financial forecasting: soft computing techniques can be used to predict financial markets, stock prices, and other economic indicators.
6. Medical diagnosis: soft computing techniques can be used to diagnose diseases and medical conditions, including cancer and heart disease.
7. Natural language processing: soft computing techniques can be used to develop natural language processing systems that can understand and respond to human language.
8. Gaming: Soft computing techniques can be used to develop intelligent game agents that can learn and adapt to the player's behaviour.

Overall, soft computing techniques are used in a wide range of applications where traditional computing techniques may not be sufficient due to the complexity and uncertainty of the data involved.

SOFT COMPUTING VERSUS HARD COMPUTING: -

Soft computing and hard computing are two distinct approaches to solving computational problems.

Hard computing typically involves the use of precise mathematical models and algorithms to obtain exact solutions to problems. This approach is commonly used in applications where accuracy and reliability are critical, such as in scientific research, engineering, and finance. Hard computing methods include traditional numerical analysis, optimization techniques, and symbolic reasoning.

In contrast, soft computing emphasizes the use of approximate reasoning and uncertain information to solve problems. Soft computing methods include fuzzy logic, neural networks, genetic algorithms, and machine learning. These techniques are particularly useful in applications where the data is noisy or incomplete, and where exact solutions are difficult to obtain.

Overall, while hard computing is more focused on exact solutions and precise mathematical modelling, soft computing is more focused on approximations and adaptable, flexible problem-solving methods that can handle uncertainty and incomplete data. Both approaches have their strengths and weaknesses, and the choice of which to use depends on the specific problem being addressed and the available resources.

BASIC TOOLS OF SOFT COMPUTING: -

Soft computing is a field of computer science that focuses on creating intelligent systems that can learn from data and make decisions based on uncertain or incomplete information. The basic tools of soft computing include:

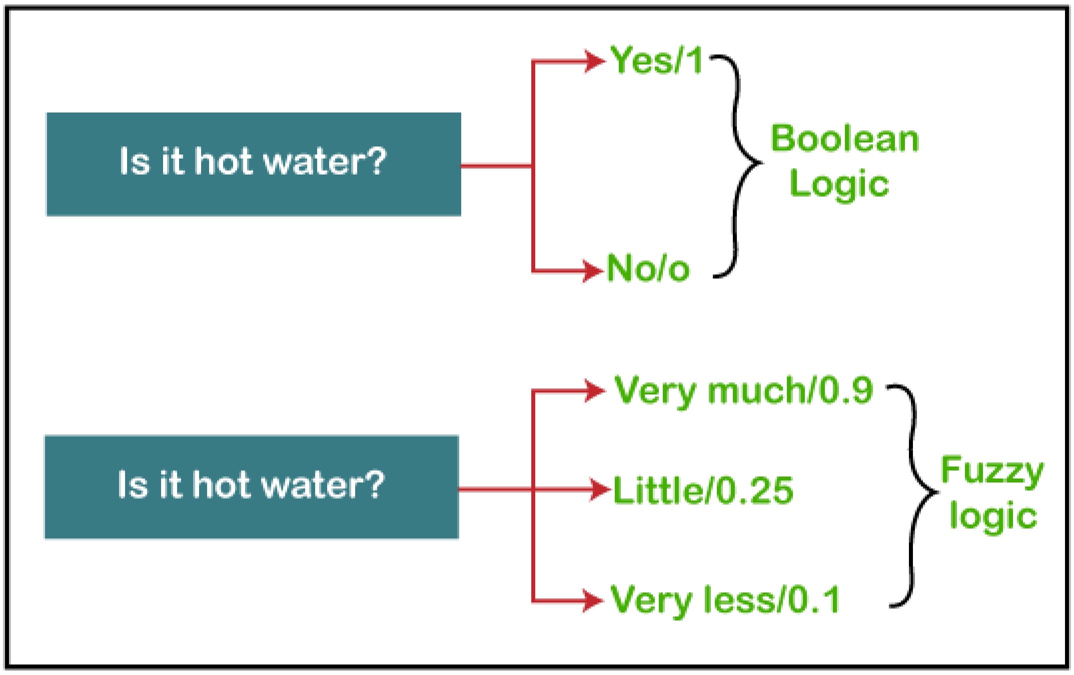
1. Fuzzy logic: Fuzzy logic is a mathematical framework that deals with reasoning under uncertainty. It uses a system of fuzzy sets and fuzzy rules to represent and manipulate uncertain or ambiguous data.
2. Neural networks: Neural networks are a type of machine learning algorithm inspired by the structure and function of the human brain. They consist of interconnected nodes, or neurons, that process and transmit information.
3. Evolutionary algorithms: Evolutionary algorithms are a family of optimization algorithms that use principles of natural selection and genetics to solve complex problems. They include genetic algorithms, evolutionary programming, and evolutionary strategies.
4. Probabilistic reasoning: Probabilistic reasoning is a mathematical framework for dealing with uncertainty that uses probability theory to represent and manipulate uncertain or incomplete data.
5. Swarm intelligence: Swarm intelligence is a collective behaviour exhibited by groups of simple agents, such as insects or birds, that work together to solve complex problems. It can be used to create algorithms for optimization, clustering, and classification.
6. Rough sets: rough sets are a mathematical framework that deals with uncertainty and vagueness in data. They are used to identify relationships between different attributes or features of a dataset.

These tools can be used in combination or individually to create intelligent systems that can learn, adapt, and make decisions based on uncertain or incomplete information.

FUZZY LOGIC: -

- Fuzzy logic is a heuristic approach that allows for more advanced decision-tree processing and better integration with rules-based programming.
- Fuzzy logic is a generalization from standard logic, in which all statements have a truth value of one or zero. In fuzzy logic, statements can have a value of partial truth, such as 0.9 or 0.5.
- Theoretically, this gives the approach more opportunity to mimic real-life circumstances, where statements of absolute truth or falsehood are rare.
- Fuzzy logic may be used by quantitative analysts to improve the execution of their algorithms.

- Because of the similarities with ordinary language, fuzzy algorithms are comparatively simple to code, but they may require thorough verification and testing.



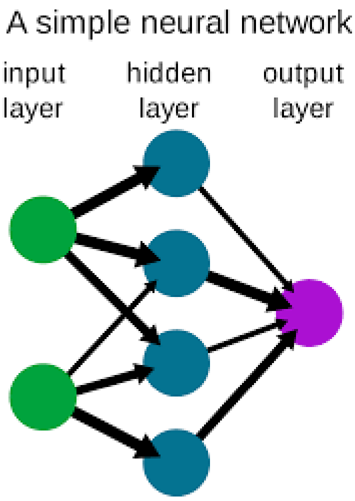
NEURAL NETWORKS AND EVOLUTIONARY COMPUTING: -

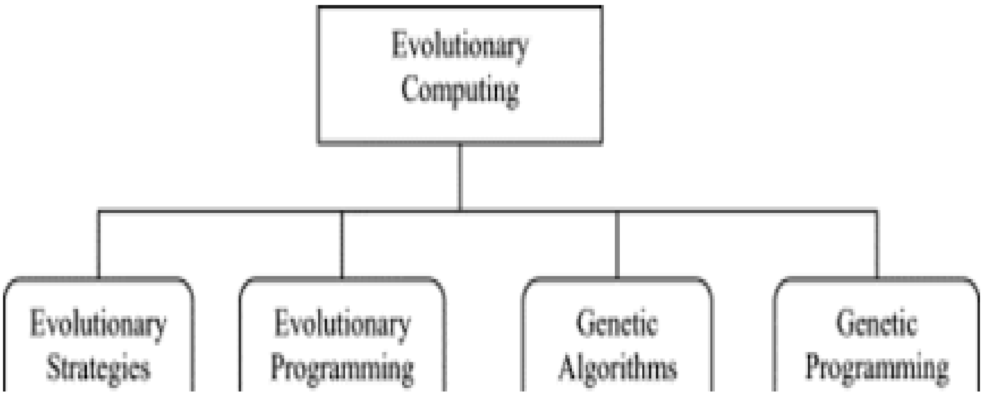
Neural networks and evolutionary computing are two different approaches to machine learning.

Neural networks are a type of machine learning algorithm that are modelled after the structure of the human brain. They consist of layers of interconnected nodes, which are capable of learning and adapting to input data. Neural networks are trained by adjusting the weights of the connections between nodes, using a process called backpropagation, until the network produces the desired output for a given input.

Evolutionary computing, on the other hand, is a family of optimization algorithms that are based on the principles of natural selection and evolution. Evolutionary computing algorithms work by generating a population of candidate solutions to a problem, and then using a process of selection, crossover, and mutation to evolve the population towards better solutions.

The two approaches can be combined in a technique known as neuro evolution, where neural networks are evolved using evolutionary computing algorithms. In this approach, a population of neural networks is generated, and then evolved over multiple generations using selection, crossover, and mutation to produce increasingly better-performing networks. Neuro evolution has been used to train neural networks for a wide range of applications, including game playing, robotics, and optimization problems.





APPROXIMATIONS OF MULTIVARIATE FUNCTIONS: -

Multivariate functions are functions that have multiple independent variables. Approximating such functions can be challenging, but there are several techniques that can be used to obtain approximate solutions.

One common technique for approximating multivariate functions is to use Taylor series expansion. Taylor series expansion provides a way to express a function as an infinite sum of terms that involve the function's derivatives evaluated at a particular point. By truncating the series at a certain order, we can obtain an approximate solution to the function.

Another technique for approximating multivariate functions is to use interpolation. Interpolation involves constructing a function that passes through a set of given data points. There are many interpolation methods available, including polynomial interpolation, spline interpolation, and radial basis function interpolation.

In addition, numerical methods such as finite element analysis and finite difference methods can be used to approximate multivariate functions. These methods involve discretizing the domain of the function and approximating the function values at the discretized points.

Overall, the choice of approximation technique depends on the specific problem at hand and the properties of the function being approximated. It is important to carefully consider the trade-offs between accuracy, computational cost, and complexity when selecting an approximation technique.

NON - LINEAR ERROR SURFACE AND OPTIMIZATION FUZZY LOGIC SYSTEMS: -

Non-linear error surfaces can arise in optimization problems when the relationship between the objective function and the decision variables is not linear. In such cases, traditional optimization techniques such as gradient descent may not be effective in finding the optimal solution. Fuzzy logic systems can be useful in dealing with such non-linear error surfaces.

Fuzzy logic is a mathematical framework that deals with uncertainty and imprecision. Fuzzy logic systems use fuzzy sets and fuzzy rules to describe relationships between variables. In an optimization problem, fuzzy logic can be used to create a fuzzy model of the error surface, which can then be used to guide the optimization process.

The basic steps in using fuzzy logic for optimization are as follows:

1. Define the problem: Define the optimization problem and the variables involved.
2. Create a fuzzy model: Create a fuzzy model of the error surface by defining fuzzy sets for each variable and creating fuzzy rules that describe the relationships between the variables.
3. Optimize: Use the fuzzy model to guide the optimization process. This can be done using a variety of techniques, such as genetic algorithms or particle swarm optimization.
4. Evaluate the solution: Evaluate the solution obtained by the optimization process to determine whether it meets the requirements of the problem.

Fuzzy logic systems can be particularly useful in optimization problems where the error surface is complex and non-linear. By creating a fuzzy model of the error surface, fuzzy logic can guide the optimization process in a more efficient manner than traditional optimization techniques. However, it should be noted that fuzzy logic systems are not a panacea for all optimization problems, and may not always be the most effective approach.

BASICS OF FUZZY LOGIC THEORY: -

Fuzzy logic is a mathematical framework for dealing with uncertainty and imprecision in data analysis and decision making. It was introduced by Lotfi Zadeh in the 1960s as an extension of traditional binary logic.

At its core, fuzzy logic is based on the idea that things can have varying degrees of truth or membership in a particular set. Instead of assigning a binary value of either 0 or 1 to a proposition, fuzzy logic allows for values between 0 and 1, representing degrees of truth.

In fuzzy logic, propositions are represented as fuzzy sets, which are defined by a membership function that assigns a degree of membership to each element in the set. The membership function can take many different forms, but the most common is a triangular or trapezoidal function that assigns a high degree of membership to elements that are close to the centre of the set and a low degree of membership to elements that are far from the centre.

Fuzzy logic also allows for the use of fuzzy rules, which are statements of the form "if A then B," where A and B are fuzzy sets. Fuzzy rules can be combined using fuzzy reasoning to make decisions or draw conclusions in situations where traditional logic would be inadequate due to uncertainty or imprecision.

Fuzzy logic has many applications in fields such as artificial intelligence, control systems, and decision making. It has proven to be particularly useful in situations where precise measurements or definitions are difficult to obtain or where expert knowledge is required to make decisions.

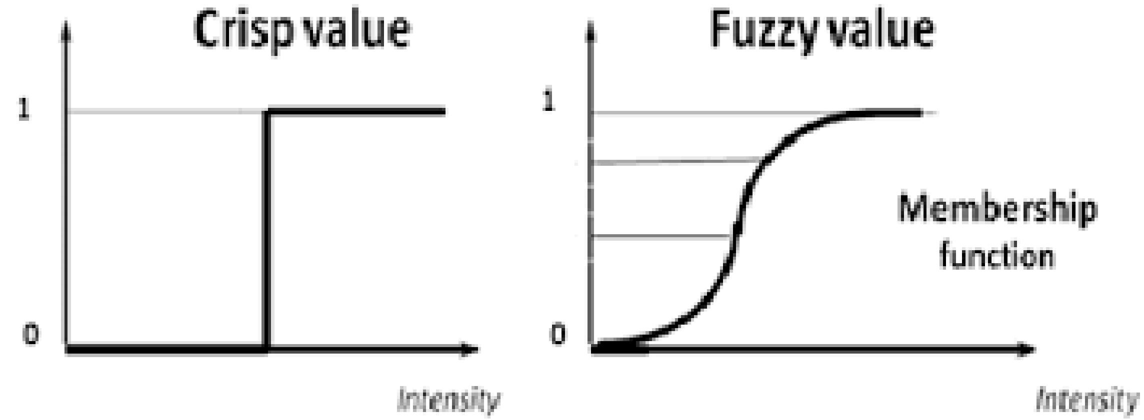
CRISP AND FUZZY SETS: -

Crisp sets and fuzzy sets are both concepts used in the field of mathematics and computer science to describe different types of sets.

A crisp set is a traditional type of set, where every element belongs to the set or does not belong to the set. For example, the set of all even numbers is a crisp set, where 2, 4, 6, and so on, are members of the set, while 1, 3, 5, and so on, are not.

On the other hand, a fuzzy set is a more flexible type of set, where each element can have a degree of membership, represented by a number between 0 and 1. For example, the set of tall people can be represented as a fuzzy set, where each person has a degree of membership based on how tall they are. A person who is 6 feet tall might have a membership degree of 0.8, while a person who is 5 feet tall might have a membership degree of 0.2.

Fuzzy sets are useful in situations where there is uncertainty or imprecision in defining the boundaries of a set. They are widely used in artificial intelligence, expert systems, control theory, and other fields. The concept of fuzzy sets was introduced by Lotfi Zadeh in 1965, and since then, it has become an important tool in various areas of research and development.



BASIC SET OPERATIONS: -

Basic set operations refer to fundamental operations that can be performed on sets. These operations include union, intersection, difference, and complement.

1. Union: The union of two sets A and B is a set that contains all the elements of A and all the elements of B, with no duplicates. The union is denoted by the symbol ∪.

Example: Let A = {1, 2, 3} and B = {3, 4, 5}. Then A ∪ B = {1, 2, 3, 4, 5}.

2. Intersection: The intersection of two sets A and B is a set that contains all the elements that are in both A and B. The intersection is denoted by the symbol ∩.

Example: Let A = {1, 2, 3} and B = {3, 4, 5}. Then A ∩ B = {3}.

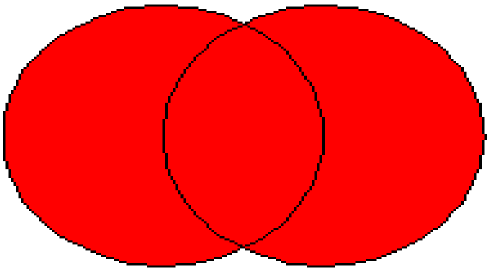
3. Difference: The difference between two sets A and B is a set that contains all the elements of A that are not in B. The difference is denoted by the symbol \ or -.

Example: Let A = {1, 2, 3} and B = {3, 4, 5}. Then A \ B = {1, 2}.

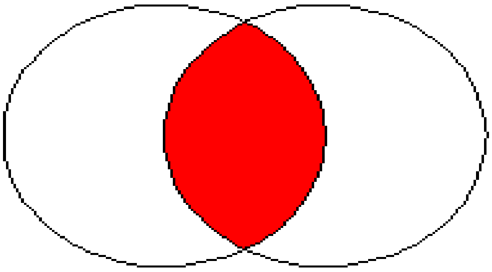
4. Complement: The complement of a set A with respect to a universal set U is the set of all elements in U that are not in A. The complement is denoted by the symbol Ac or A'.

Example: Let U = {1, 2, 3, 4, 5} and A = {1, 2, 3}. Then A' = {4, 5}

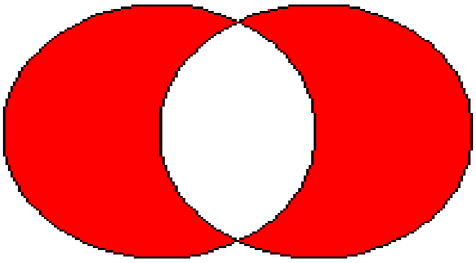
Union



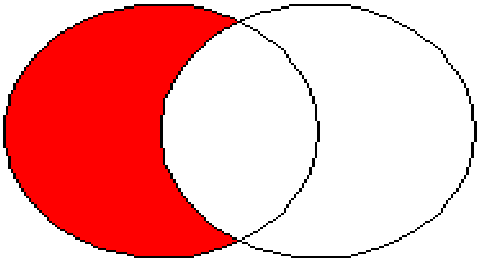
Intersection



Exclusive Or



Subtraction



FUZZY RELATIONS: -

Fuzzy relations are a mathematical concept used in fuzzy logic and fuzzy set theory. A fuzzy relation is a generalization of a crisp binary relation, where the truth values are replaced with degrees of membership in a fuzzy set.

In a crisp binary relation, each element is either related to another element or not related at all. In a fuzzy relation, each element can have a degree of membership in the relation, which can range from 0 (not related at all) to 1 (completely related).

Fuzzy relations can be represented as matrices, where the degree of membership of each element in the relation is represented by a value between 0 and 1. Fuzzy relations can be composed, inverted, and transposed, just like crisp relations.

Fuzzy relations have many applications, such as in decision-making, expert systems, and control systems. They are particularly useful in situations where there is uncertainty or imprecision in the data, as they can handle these situations more effectively than crisp relations.

COMPOSITION OF FUZZY RELATIONS: -

Fuzzy relations are a type of mathematical framework that allows us to reason about uncertainty in relationships between different entities. In a fuzzy relation, the degree of membership of each element of one set to another set is defined by a degree of truth or degree of possibility, which ranges between 0 and 1.

The composition of fuzzy relations is an operation that combines two fuzzy relations to create a new fuzzy relation that represents the relationship between the elements of the two original relations. The composition of fuzzy relations is a fundamental operation in fuzzy set theory and plays a crucial role in many applications, including control systems, decision-making, and artificial intelligence.

The composition of two fuzzy relations A and B can be defined as follows:

$R = A \circ B$

where R is the resulting fuzzy relation, A and B are the two fuzzy relations being composed, and o denotes the composition operator.

There are several different approaches to defining the composition operator for fuzzy relations, including the min-max composition, max-min composition, and max-product composition. The choice of composition operator depends on the specific application and the nature of the fuzzy relations being composed.

In the min-max composition, the composition operator is defined as follows:

$R(x, z) = \min\{\max[A(x, y), B(y, z)]\}$

where x, y, and z are elements of the sets associated with the fuzzy relations A and B, and R(x, z) is the degree of membership of the pair (x, z) in the resulting fuzzy relation R.

In the max-min composition, the composition operator is defined as follows:

$R(x, z) = \max\{\min[A(x, y), B(y, z)]\}$

where x, y, and z are elements of the sets associated with the fuzzy relations A and B, and R(x, z) is the degree of membership of the pair (x, z) in the resulting fuzzy relation R.

In the max-product composition, the composition operator is defined as follows:

$R (x, z) = \max [A (x, y) * B (y, z)]$

where x, y, and z are elements of the sets associated with the fuzzy relations A and B, and R (x, z) is the degree of membership of the pair (x, z) in the resulting fuzzy relation R.

Each of these composition operators has its strengths and weaknesses, and the choice of operator depends on the specific application and the properties of the fuzzy relations being composed. The composition of fuzzy relations is a powerful tool for reasoning about uncertainty and is widely used in many fields, including artificial intelligence, decision-making, and control systems.

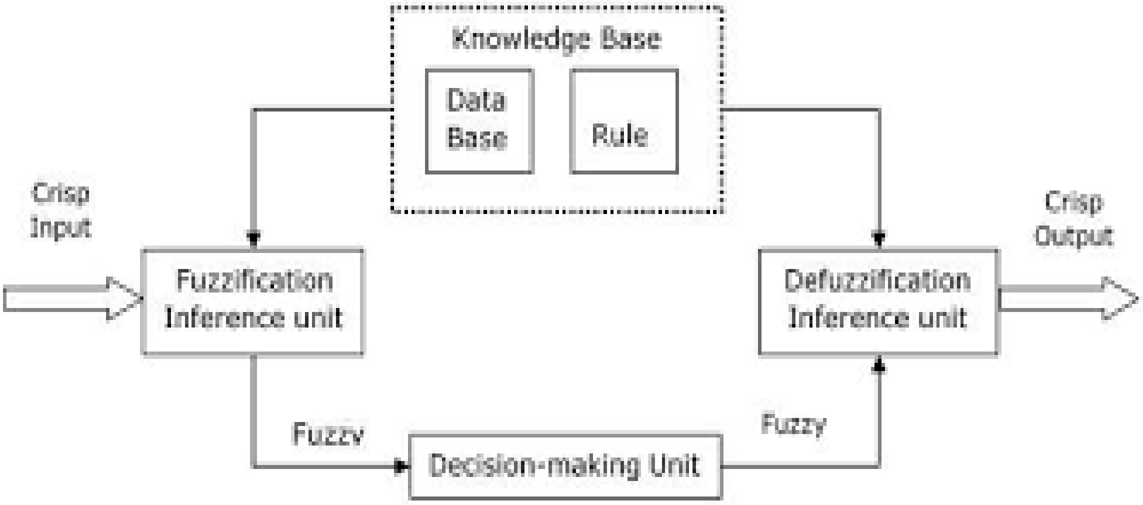
FUZZY INFERENCE: -

Fuzzy inference is a process of making decisions based on uncertain or incomplete information using fuzzy logic. Fuzzy logic is a type of mathematical logic that allows for reasoning with imprecise or uncertain data.

In a fuzzy inference system, inputs are first fuzzified by mapping them onto fuzzy sets. Then, rules are defined using fuzzy logic to make inferences based on the inputs. The rules define how the inputs are combined and how the output is determined.

The fuzzy inference process involves several steps, including fuzzification, rule evaluation, aggregation, and defuzzification. Fuzzification involves mapping the inputs onto fuzzy sets. Rule evaluation determines the degree to which each rule applies to the input. Aggregation combines the outputs of each rule into a single output. Defuzzification converts the fuzzy output into a crisp value.

Fuzzy inference is used in many applications, such as control systems, decision-making systems, pattern recognition, and artificial intelligence. It is particularly useful in situations where data is uncertain or incomplete, such as in natural language processing and image recognition.



ZADEH'S COMPOSITIONAL RULE OF INFERENCE: -

Zadeh's compositional rule of inference, also known as the compositional rule of inference for fuzzy logic, is a mathematical framework for reasoning with fuzzy sets. It was developed by Lotfi Zadeh, the founder of fuzzy logic, in the 1960s.

The compositional rule of inference states that given two fuzzy sets A and B, their composition (also known as their intersection or AND operation) is given by the equation:

$\mu (A \cap B) = \min (\mu A(x), \mu B(x))$

where μ A(x) and μ B(x) are the membership functions of A and B, respectively, at a particular value x. The output of this equation is a new fuzzy set C, which represents the degree to which x belongs to both A and B.

The compositional rule of inference is an important tool in fuzzy logic, as it allows for reasoning with uncertain or imprecise data. It can be used in a variety of applications, including control systems, pattern recognition, and decision making.

DEFUZZIFICATION: -

Defuzzification is the process of converting the fuzzy output of a fuzzy logic system into a crisp value that can be used in the real world. In fuzzy logic, the output of a system is a fuzzy set that represents the degree of membership of an input value in the output set.

Defuzzification is necessary because the output of a fuzzy logic system is a fuzzy set, which is not directly useful for making decisions or taking actions. Defuzzification transforms the fuzzy set into a single value that can be used as an output for the system.

There are several methods for defuzzification, including the centroid method, the maximum value method, and the height method. The most commonly used method is the centroid method, which calculates the centre of gravity of the output fuzzy set.

In the centroid method, the output fuzzy set is weighted by its degree of membership, and the weighted average of the set is calculated. This weighted average is the crisp value that represents the output of the fuzzy logic system.

Defuzzification is an important step in the application of fuzzy logic, as it enables the use of fuzzy logic systems in real-world applications.

FUZZY LOGIC CONTROL: -

Fuzzy logic control is a type of control system that uses fuzzy logic to make decisions based on uncertain or ambiguous information. Fuzzy logic is a mathematical framework that allows for reasoning with imprecise or uncertain information by using degrees of membership in a set. In a fuzzy logic control system, input variables are converted into fuzzy sets, which are then processed using a set of fuzzy rules to determine the output variable.

Fuzzy logic control is particularly useful in systems where traditional control methods may not be effective, such as in systems with non-linear dynamics, imprecise or uncertain inputs, or complex and changing operating conditions. Fuzzy logic controllers have been used in a wide range of applications, including process control, robotics, and automotive control systems.

One advantage of fuzzy logic control is its ability to handle non-linear and complex systems, which may be difficult to model and control using traditional methods. Additionally, fuzzy logic controllers are often more robust and adaptive than traditional control systems, as they can handle changing and uncertain environments more effectively. However, designing and implementing a fuzzy logic control system can be challenging, as it requires a detailed understanding of the system being controlled and the ability to design effective fuzzy rules and membership functions.

MAMDANI AND TAKAGI AND SUGENO ARCHITECTURES:

Mamdani, Takagi-Sugeno, and Sugeno are all architectures used in fuzzy logic systems, a type of mathematical logic that deals with reasoning that is approximate rather than precise.

1. Mamdani Architecture:

The Mamdani architecture is a type of fuzzy logic system that uses a set of fuzzy rules to map input variables to output variables. The input variables are first fuzzified, i.e., converted into fuzzy sets, which are then used to activate the fuzzy rules. The rules are evaluated using a fuzzy inference engine, and the resulting fuzzy outputs are then aggregated to obtain a crisp output.

2. Takagi-Sugeno Architecture:

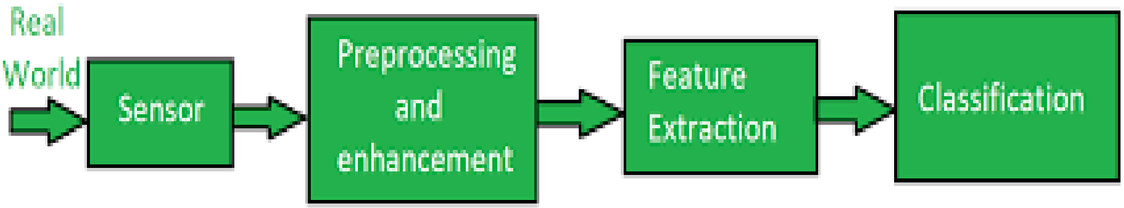
The Takagi-Sugeno architecture is a type of fuzzy logic system that uses a set of linear functions to map input variables to output variables. Unlike the Mamdani architecture, the Takagi-Sugeno architecture does not use fuzzy sets to represent input variables. Instead, it uses crisp input values to activate the linear functions, which are defined for each rule. The resulting linear outputs are then aggregated to obtain a crisp output.

3. Sugeno Architecture:

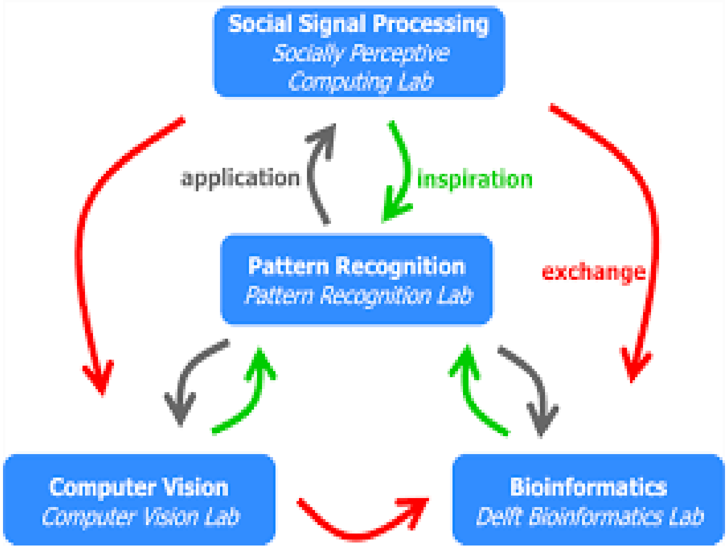
The Sugeno architecture is a type of fuzzy logic system that is similar to the Takagi-Sugeno architecture but uses a set of nonlinear functions instead of linear functions. The nonlinear functions are defined for each rule, and the crisp output is obtained by taking a weighted average of the nonlinear outputs.

In summary, Mamdani, Takagi-Sugeno, and Sugeno architectures are all used in fuzzy logic systems, but they differ in how they represent input variables and how they map them to output variables. The choice of architecture depends on the specific problem being addressed and the requirements of the system.

APPLICATIONS TO PATTERN RECOGNITION: -



Pattern Recognition System



Pattern recognition is a field of study that involves the automated identification of patterns in data. Applications of pattern recognition are diverse, and can be found in fields such as computer vision, speech recognition, natural language processing, and bioinformatics, among others. Some specific applications of pattern recognition are:

1. Object recognition: Object recognition involves identifying objects in images or videos. It has numerous applications in fields such as surveillance, robotics, and autonomous vehicles.
2. Handwriting recognition: Handwriting recognition involves the identification of handwritten text. It is used in applications such as automatic form processing and signature verification.
3. Speech recognition: Speech recognition involves converting spoken words into text. It is used in applications such as voice assistants and speech-to-text transcription.
4. Face recognition: Face recognition involves identifying individuals based on facial features. It is used in applications such as security systems, social media tagging, and mobile device authentication.
5. Image classification: Image classification involves assigning labels to images based on their content. It is used in applications such as medical diagnosis, quality control in manufacturing, and online image search.

6. Natural language processing: Natural language processing involves the analysis of human language. It is used in applications such as sentiment analysis, language translation, and chatbots.
7. Bioinformatics: Bioinformatics involves the application of computer science to biological data. Pattern recognition techniques are used to analyse DNA sequences and identify patterns that can provide insights into genetic disorders and diseases.

These are just a few examples of the many applications of pattern recognition. As the field of artificial intelligence continues to evolve, we can expect to see even more innovative applications of pattern recognition in the future.