ROHINI COLLEGE OF ENGINEERING AND TECHNOLOGY

24AI572-SOFT COMPUTING AND EVOLUTIONARY AI

Department of Computer Sciences and Engineering

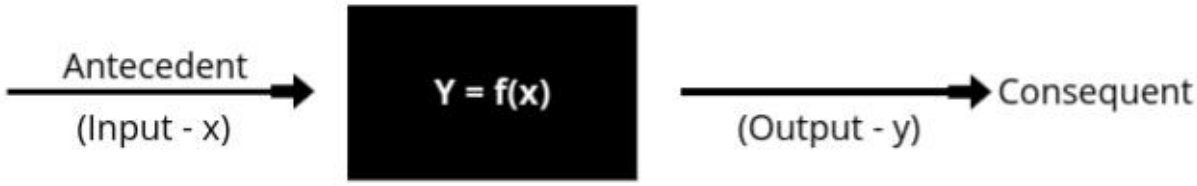
UNIT I – FOUNDATIONS OF SOFT COMPUTING AND COMPUTATIONAL INTELLIGENCE

1.1 Introduction to Soft Computing

Computing

Before diving into soft computing, it's essential to first understand what computing is and familiarize ourselves with the core terminologies related to computing.

- The following diagram represents the basic process that occurs in computing:



- In this diagram, we observe that there is a function, **f**, which takes some input, **x**, and produces an output, **y**. The input is referred to as the **Antecedent**, while the output is called the **Consequent**.
- The function **f** can also be called a **formal method**, **algorithm**, or a **mapping function**. It represents the logic or steps used to process the input and generate the output.
- The middle part of this diagram is known as the **computing unit**, where the function resides. This is where we feed the input, and a process occurs to convert that input into the desired output **y**.
- In the computing process, the steps that guide how the input is manipulated are called **control actions**. These control actions ensure that the input gradually approaches the desired output. The moment the process completes, we obtain the final result, known as the **Consequent**.

Basic Characteristics Related to Computing:

1. **Precise Solution:** Computing is used to produce a solution that is exact and definitive.
2. **Unambiguous & Accurate:** The control actions within the computing unit must be unambiguous, meaning they must have only one interpretation. Each control action should also be accurate to ensure that the process is valid and reliable.
3. **Mathematical Model:** A well-defined algorithm is a requirement for solving any problem through computing. This algorithm is essentially a mathematical model that represents the problem and its solution process.

- **In computing, there are two main types**:

    1. Hard Computing
    2. Soft Computing

Hard Computing

Hard computing refers to a traditional computing approach where the results are always precise and exact. This is because hard computing relies on well-defined mathematical models and algorithms to solve problems. There is no room for approximation, as every calculation or action leads to a deterministic output.

Main Features of Hard Computing:

- **Exact Results:** The results in hard computing are always exact. This means, whenever we solve a problem using hard computing methods, the answer will always be the same and correct. There is no room for uncertainty or approximation.
- **Clear Control Actions:** In hard computing, the steps or actions we take to solve a problem are clear and have only one meaning. For example, when following an algorithm, each step must be clearly understood and should only have one possible result. There should be no confusion or multiple meanings for a single step. This is called being "unambiguous."
- **Based on Mathematical Models:** All the actions in hard computing are based on well-defined mathematical rules or formulas. This means that each step we take while solving a problem has a proper mathematical explanation behind it. These steps follow a fixed pattern or method, which helps in solving the problem in a precise way.

- **Examples of Hard Computing:**

1. **Numerical Problems:** Problems that require accurate mathematical calculations, like solving equations or doing arithmetic, are examples of hard computing.

2. **Searching and Sorting Algorithms:** In computer science, methods like searching for an item in a list or sorting a list in a particular order are solved using hard computing techniques because they always give an exact answer.

3. **Computational Geometry Problems:** These are problems related to shapes, distances, and areas in geometry. For example, calculating the shortest distance between two points or finding the area of a triangle requires exact methods, which is where hard computing comes in.

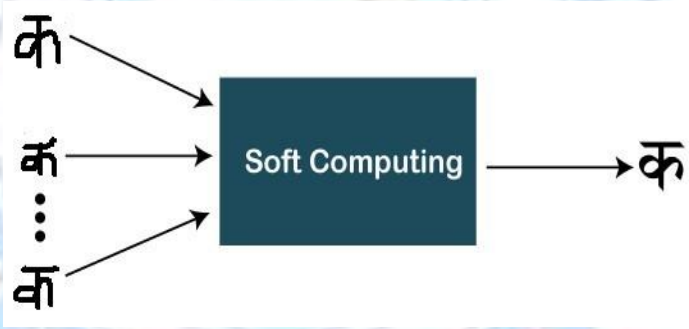
Soft Computing

Soft Computing is a collection of computational techniques that imitate human reasoning and decision-making to solve complex real-world problems involving uncertainty, ambiguity, and imprecision.

Definition of Soft Computing:

Lotfi A. Zadeh, 1992 : "Soft Computing is an emerging approach to computing which parallel the remarkable ability of the human mind to reason and learn in a environment of uncertainty and imprecision".

The Soft Computing consists of several computing paradigms mainly: Fuzzy Systems, Neural Networks, Evolutionary computation.These methodologies form the core of SC. Hybridization of these three creates a successful synergic effect; that is, hybridization creates a situation where different entities cooperate advantageously for a final outcome. Soft Computing is still growing and developing.



Why Soft Computing approach?

Mathematical model & analysis can be done for relatively simple systems. More complex systems arising in biology, medicine and management systems remain intractable to conventional mathematical and analytical methods. Soft computing deals with imprecision, uncertainty, partial truth and approximation to achieve tractability, robustness and low solution cost. It extends its application to various disciplines of Engineering and science.

Goals of Soft Computing

Soft Computing is a new multidisciplinary field, to construct new generation of Artificial Intelligence, known as Computational Intelligence.

The main goal of Soft Computing is to develop intelligent machines to provide solutions to real world problems, which are not modeled, or too difficult to model mathematically. Its aim is to exploit the tolerance for Approximation, Uncertainty, Imprecision, and Partial Truth in order to achieve close resemblance with human like decision making.

- Approximation: The model features are similar to the real system, but not the exactly the same. It is a simplified version that closely represents reality.
  Example: Using a straight line to represent a slightly curved road on a map. The road is not exactly straight, but it is close enough for most purposes.
- Uncertainty: Uncertainty means we do not know the exact value or feature of the real system because of incomplete information or lack of knowledge. Example: A weather forecast predicts a 70% chance of rain. We are not certain whether it will actually rain.
- Imprecision: Imprecision means the value in the model is not exactly equal to the real value, but it is close enough. The model features (quantities) are not the same as that of the real ones, but close to them. Example: The actual height of a person is 170.3 cm, but it is recorded as 170 cm. The value is not exact, but it is very close.

Importance of Soft Computing

- **Handles Uncertainty and Approximation:**
  - Traditional algorithms often require exact inputs, but real-world problems don't always have precise data. Soft computing methods can handle uncertainty, making them more suitable for practical applications.
  - Example: In medical diagnosis, data such as symptoms might be uncertain or incomplete, and soft computing helps make decisions despite that.
- **Flexible and Adaptive:**
  - Soft computing systems are highly adaptive to changes and can learn from new data. This makes them ideal for dynamic environments where conditions or inputs change over time.
  - Example: In stock market predictions, a neural network can learn from past trends and adapt to new market conditions.
- **Tolerance to Noise and Incompleteness:**
  - Soft computing techniques can work with noisy data (data that is random or disturbed) and incomplete information. They don’t require the exact and clean data that traditional methods do.
  - Example: Image recognition systems in self-driving cars can still work even when the images are slightly unclear or distorted.
- **Ease of Implementation:**
  - Soft computing techniques like fuzzy logic and genetic algorithms are often easier to implement for complex, real-world problems compared to traditional methods. They don’t require a precise model of the system but instead learn from examples.

- Example: In control systems, fuzzy logic can be quickly applied to model human-like decision-making without needing complex mathematical models.

- **Incorporates Human-Like Reasoning:**
  - Soft computing allows for reasoning based on approximate data, mimicking human decision-making. This is especially useful in applications where exact answers aren't available.
  - Example: In customer service, fuzzy logic can simulate human-like reasoning in responding to customer queries.

Various Techniques of Soft Computing

1. **Fuzzy Logic**: Mimics human reasoning, allowing for more than just "true" or "false" outcomes. It's widely used in control systems, such as thermostats or automated vehicle systems.
2. **Neural Networks**: Modeled after the human brain, these networks learn from data and improve over time. They are used in areas like image recognition, speech processing, and autonomous systems.
3. **Genetic Algorithms**: Inspired by natural selection, these algorithms "evolve" solutions over generations. They are useful for optimization problems, such as route planning or complex decision-making.
4. **Probabilistic Reasoning**: Deals with uncertainty using probabilities. It is useful in situations where outcomes are not deterministic, like predicting stock market trends or diagnosing diseases.

Applications of Soft Computing

Soft computing techniques are applied in a variety of fields, including:

- **Artificial Intelligence (AI)**: Many AI systems rely on neural networks and fuzzy logic to emulate human decision-making.
- **Pattern Recognition**: Soft computing helps identify patterns in complex data, such as handwriting recognition or facial recognition.
- **Robotics**: Soft computing enables robots to navigate environments with uncertain data, making them more adaptable.
- **Data Mining**: Soft computing techniques are used to discover patterns and relationships in large datasets.

- **Control Systems**: Fuzzy logic controllers are used in industrial systems, like temperature control in manufacturing plants.
- **Medicine**: Soft computing helps in diagnosing diseases, predicting patient outcomes, and optimizing treatment plans.
- **Economics and Finance**: Used in predicting market trends, risk analysis, and portfolio optimization.

HARD COMPUTING

Hard Computing is a conventional computing method that requires a precisely stated analytical model and often a lot of computational time. In 1996, L.A. Zade introduced the term hard computing. According to Zade, we term a computing as hard computing if precise result is guaranteed, control action is unambiguous, control action is pre-defined with mathematical model or algorithm and truthness value is 100% or 1.

Examples of Hard Computing are the calculation of done by the calculator 25 + 15 provides exact 40 as its answer and similarly the ATM machine gives the exact amount of money you request.

Difference between hard computing and soft computing

Hard Computing	Soft Computing
Conventional computing method that requires a precisely stated analytical model and often a lot of computational time.	Tolerates imprecision, uncertainty, partial truth, and approximation.
Based on binary logic, crisp systems, numerical analysis, and crisp software.	Based on fuzzy logic, neural networks, and probabilistic reasoning.
Programs must be explicitly written by the programmer.	Can learn and adapt by evaluating or improving its own program.
Uses two-valued (binary) logic (True/False, 0/1).	Uses multi-valued logic (degrees of truth).
Deterministic in nature.	Stochastic (probabilistic) in nature.
Requires exact and precise input data.	Can work with ambiguous, noisy, or incomplete data.
Performs strictly sequential computation.	Supports parallel computation.
Produces precise and exact answers.	Produces approximate but acceptable answers.