SOFT COMPUTING

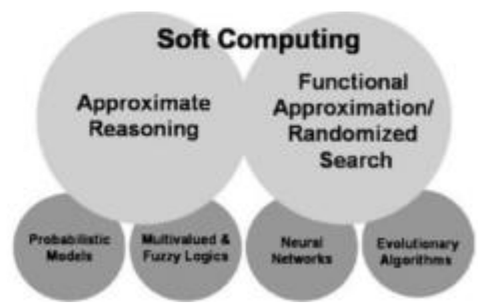
Safnas Saleem, varsha Valsan, Fathima Riswana, Haneena Yousaf
B.TECH 111year, Department of Computer Science and Engineering
KERALA TECHNOLOGICAL UNIVERSITY

Abstract-Soft computing, as opposed to traditional computing, deals with approximate models and gives solutions to complex real-life problems. Unlike hard computing, soft computing is tolerant of imprecision, uncertainty, partial truth, and approximations. In effect, the role model for soft computing is the human mind. Soft computing is based on techniques such as fuzzy logic, genetic algorithms, artificial neural networks, machine learning, and expert systems. Although soft computing theory and techniques were first introduced in 1980s, it has now become a major research and study area in automatic control engineering. The techniques of soft computing are nowadays being used successfully in many domestic, commercial, and industrial applications. With the advent of the low-cost and very high performance digital processors and the reduction of the cost of memory chips it is clear that the techniques and application areas of soft computing will continue to expand. This paper gives an overview of the current state of soft computing techniques and describes the advantages and disadvantages of soft computing compared to traditional hard computing techniques.

IndexTerms— Softcomputing, fuzzy logic,geneticalgorithms, neural networks,expert system

1.INTRODUCTION

Soft computing is an emerging approach to computing that gives the remarkable ability of the human mind to argue and learn in the atmosphere of uncertainty and distrust.Soft computing is based on some biological induced methods such as genetics, development, ant behavior, the warm of particles, the human nervous system, etc. Now SC is the only solution when we do not have any mathematical modeling of problem-solving (i.e., algorithm), in real-time, there is a need to solve a complex problem, adapt with the changed scenario and be implemented with parallel computing. It has massive applications in many application zones such as medical diagnosis, computer vision, machine intelligence, weather forecasting, network optimization, LSI design, pattern recognition, handwritten character improvement etc.



2.APPLICATION OF SOFTCOMPUTING

- Robotic works in the form of Emotional Pet robots.
- Food preparation devices are Microwave and Rice cookers.
- For amusing gaming playing product like Checker and Poker etc.
- Recognition for Handwriting.
- Data compression/Image Processing
- For Architecture
- System to Decision-suppor

3.IMPORTANCE OF SOFTCOMPUTING

The supplementation of FL, NC, GC, and PR is an important result: In many cases, any problem can be solved most effectively by using FL, NC, GC and PR rather than specially in combination. A great example of a particularly effective combination is known as “Neurofjie System”. Such systems are increasingly seen as a consumer product ranging from air conditioners and washing machines to photocopiers and camcorders. There are less visible but perhaps even more important Neurofjie systems in industrial applications. It is especially important that in both consumer products and industrial systems, the use of soft computing technologies leads to systems that have high MIQ (Machine Intelligence Quota). Artificial

4.SOFTCOMPUTING TECHNIQUES

1. Neural Networks (ANN):Human brains in a way describe the real world conditions, which computers cannot. In order to solve this issue, for the first time, neural networks were developed in the 1950s. An artificial neural network is an attempt to emulate a network of neurons that makes a human brain so that computers can be able to learn things and make decisions in a human way. ANN is made by regular computer programming, as if they are mutually associated with brain cells.

2. Fuzzy logic:Fuzzy logic is a mathematical logic, which attempts to solve problems with an open, imprecise spectrum of data that makes it possible to get an array of precise findings. Fuzzy logic is designed to be considered the best possible decision by considering all available information and looking an input.

3. Genetic Algorithm:Nature is and will always be an amazing source of inspiration for all of mankind. Genetic algorithms (GA) take all their inspiration from nature, and there are no less genetic algorithms based on search-based algorithms that find its roots in natural selection and concepts of genetics. The genetic algorithm is also a subset of a large branch of computation (also called evolutionary computation).

5.ARTIFICAL NEURAL NETWORK

An artificial neural network (ANN) is the piece of a computing system designed to simulate the way the human brain analyzes and processes information. It is the foundation of Ai.(AI) and solves problems that would prove impossible or difficult by human or statistical standards. ANNs have self-learning capabilities that enable them to produce better results as more data becomes available. Artificial neural networks are built like the human brain, with neuron nodes interconnected like a web. The human brain has hundreds of billions of cells called neurons. Each neuron is made up of a cell body that is responsible for processing information by carrying information towards (inputs) and away (outputs) from the brain.

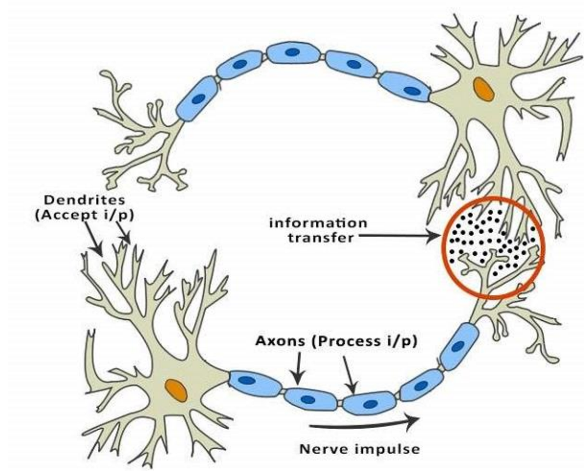
An ANN has hundreds or thousands of artificial neurons called processing units, which are interconnected by nodes. These processing units are made up of input and output units. The input units receive various forms and structures of information based on an internal weighting system, and the neural network attempts to learn about the information presented to produce one output report. Just like humans need rules and guidelines to come up with a result or output, ANNs also use a set of learning rules called backpropagation, an abbreviation for backward propagation of error, to perfect their output results.

An ANN initially goes through a training phase where it learns to recognize patterns in data, whether visually, aurally, or textually. During this supervised phase, the network compares its actual output produced with what it was meant to produce—the desired output. The difference between both outcomes is adjusted using backpropagation. This means that the network works backward, going from the output unit to the input units to adjust the weight of its connections between the units until the difference between the actual and desired outcome produces the lowest possible error.

During the training and supervisory stage, the ANN is taught what to look for and what its output should be, using yes/no question types with binary numbers. For example, a bank that wants to detect credit card on time may have four input units fed with these questions: (1) Is the transaction in a different country from the user's resident country? (2) Is the website the card is being used at affiliated with companies or countries on the bank's watch list? (3) Is the transaction amount larger than $2,000? (4) Is the name on the transaction bill the same as the name of the cardholder?

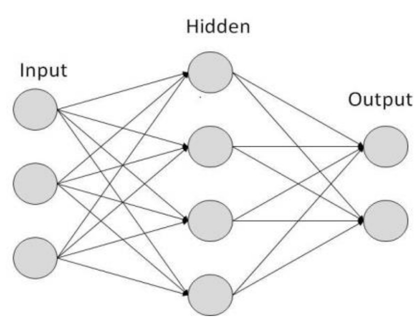
The bank wants the "fraud detected" responses to be Yes Yes Yes No, which in binary format would be 1 1 1 0. If the network's actual output is 1 0 1 0, it adjusts its results until it delivers an output that coincides with 1 1 1 0. After training, the computer system can alert the bank of pending fraudulent transactions, saving the bank lots of money. The idea of ANNs is based on the belief that working of human brain by making the right connections, can be imitated using silicon and wires as living **neurons** and **dendrites**.

The human brain is composed of 86 billion nerve cells called **neurons.** They are connected to other thousand cells by **Axons.** Stimuli from external environment or inputs from sensory organs are accepted by dendrites. These inputs create electric impulses, which quickly travel through the neural network. A neuron can then send the message to other neuron to handle the issue or does not send it forward.



ANNs are composed of multiple **nodes**, which imitate biological **neurons** of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its **activation** or **node value.**

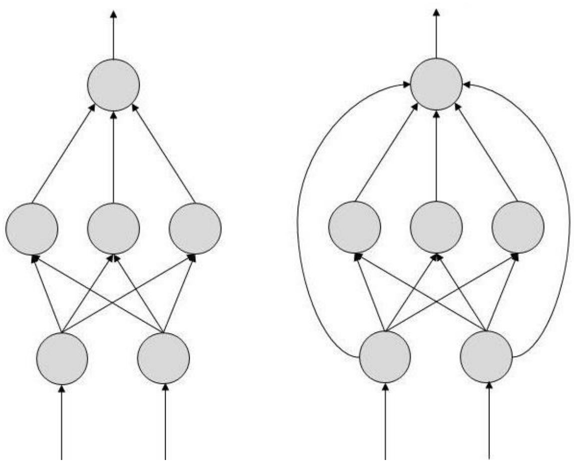
Each link is associated with **weight.** ANNs are capable of learning, which takes place by altering weight values. The following illustration shows a simple ANN −



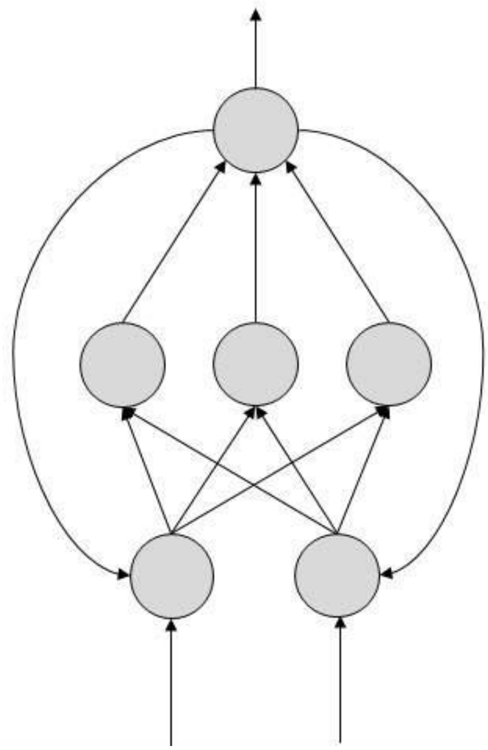
There are two Artificial Neural Network topologies − **FeedForward** and **Feedback.**

FeedForward ANN

In this ANN, the information flow is unidirectional. A unit sends information to other unit from which it does not receive any information. There are no feedback loops. They are used in pattern generation/recognition/classification. They have fixed inputs and outputs.



Here, feedback loops are allowed. They are used in content addressable memories.



Working of ANNs

In the topology diagrams shown, each arrow represents a connection between two neurons and indicates the pathway for the flow of information. Each connection has a weight, an integer number that controls the signal between the two neurons.

If the network generates a “good or desired” output, there is no need to adjust the weights. However, if the network generates a “poor or undesired” output or an error, then the system alters the weights in order to improve subsequent results.

Machine Learning in ANNs

ANNs are capable of learning and they need to be trained. There are several learning strategies −

- **Supervised Learning** − It involves a teacher that is scholar than the ANN itself. For example, the teacher feeds some example data about which the teacher already knows the answers.

For example, pattern recognizing. The ANN comes up with guesses while recognizing. Then the teacher provides the ANN with the answers. The network then compares it guesses with the teacher's "correct" answers and makes adjustments according to errors.

- **Unsupervised Learning** − It is required when there is no example data set with known answers. For example, searching for a hidden pattern. In this case, clustering i.e. dividing a set of elements into groups according to some unknown pattern is carried out based on the existing data sets present.
- **Reinforcement Learning** − This strategy built on observation. The ANN makes a decision by observing its environment. If the observation is negative, the network adjusts its weights to be able to make a different required decision the next time.

Back Propagation Algorithm

It is the training or learning algorithm. It learns by example. If you submit to the algorithm the example of what you want the network to do, it changes the network's weights so that it can produce desired output for a particular input on finishing the training.

Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks.

Bayesian Networks (BN)

These are the graphical structures used to represent the probabilistic relationship among a set of random variables. Bayesian networks are also called **Belief Networks** or **Bayes Nets.** BNs reason about uncertain domain.

In these networks, each node represents a random variable with specific propositions. For example, in a medical diagnosis domain, the node Cancer represents the proposition that a patient has cancer.

The edges connecting the nodes represent probabilistic dependencies among those random variables. If out of two nodes, one is affecting the other then they must be directly connected in the directions of the effect. The strength of the relationship between variables is quantified by the probability associated with each node.

There is an only constraint on the arcs in a BN that you cannot return to a node simply by following directed arcs. Hence the BNs are called Directed Acyclic Graphs (DAGs).

BNs are capable of handling multivalued variables simultaneously. The BN variables are composed of two dimensions −

- Range of prepositions
- Probability assigned to each of the prepositions.

Consider a finite set $X = \{X_1, X_2, \ldots, X_n\}$ of discrete random variables, where each variable X_i may take values from a finite set, denoted by $Val(X_i)$. If there is a directed link from variable X_i to variable, X_j, then variable X_i will be a parent of variable X_j showing direct dependencies between the variables.

The structure of BN is ideal for combining prior knowledge and observed data. BN can be used to learn the causal relationships and understand various problem domains and to predict future events, even in case of missing data.

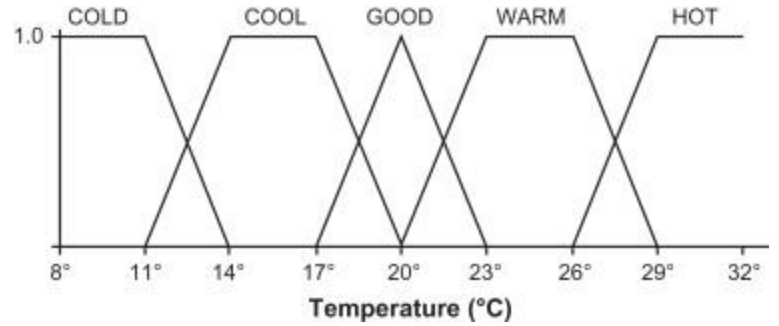
Building a Bayesian Network

A knowledge engineer can build a Bayesian network. There are a number of steps the knowledge engineer needs to take while building it.

Example problem − *Lung cancer.* A patient has been suffering from breathlessness. He visits the doctor, suspecting he has lung cancer. The doctor knows that barring lung cancer, there are various other possible diseases the patient might have such as tuberculosis and bronchitis.

6.FUZZIFICATION

Fuzzification is the process of decomposing a system input and/or output into one or more fuzzy sets. Many types of curves and tables can be used, but triangular or trapezoidal-shaped membership functions are the most common, since they are easier to represent in embedded controllers. Figure 7.18 shows a system of fuzzy sets for an input with trapezoidal and triangular membership functions. Each fuzzy set spans a region of input (or output) values graphed against membership. Any particular input is interpreted from this fuzzy set, and a degree of membership is obtained. The membership functions should overlap, in order to allow smooth mapping of the system. The process of fuzzification allows the system inputs and outputs to be expressed in linguistic terms to allow rules to be applied in a simple manner to express a complex system.



Consider a simplified implementation of an air-conditioning system with a temperature sensor. The temperature might be read by a microprocessor that has a fuzzy algorithm that processes output to continuously control the speed of a motor which keeps the room at a "good temperature"; it also can direct a vent upward or downward as necessary. Figure 7.18 illustrates the process of fuzzification of the air temperature.

There are five fuzzy sets for temperature: COLD, COOL, GOOD, WARM, and HOT.

The membership function for fuzzy sets COOL and WARM are trapezoidal, the membership function for GOOD is triangular, and those for COLD and HOT are half triangular, with shoulders indicating the physical limits for such a process (staying in a place with a room temperature lower than 8°C or above 32°C would be quite uncomfortable). The way to design such fuzzy sets depends solely on the designer's experience and intuition. The figure shows some non-overlapping fuzzy sets, which can indicate any in the modeling process. There an input temperature of 18°C would be considered COOL to a degree of 0.75 and would be considered GOOD to a degree of 0.25. To build the rules that will control the air-conditioning motor, we could watch how a human expert adjusts the settings to speed up and slow down the motor in accordance with the temperature, obtaining the rules empirically. For instance, if the room temperature is good, keep the motor speed medium; if it is warm, turn the knob of the speed to fast, and blast the speed if the room is hot. On the other hand, if the temperature is cool, slow down the speed, and stop the motor if it is cold. The beauty of fuzzy logic is the way it turns common sense, and linguistic descriptions, into a computer-controlled system. To complete this process it is necessary to understand how to use some logical operations to build the rules.

Boolean logic operations must be extended in fuzzy logic to manage the notion of partial truth – truth-values between "completely true" and "completely false". A fuzziness nature of a statement such as "X is LOW" might be combined with the fuzziness statement of "Y is HIGH" and a typical logical operation could be given as X is LOW AND Y is HIGH. What is the truth-value of this AND operation? Logic operations with fuzzy sets are performed with the membership functions. Although there are various other interpretations for fuzzy logic operations, the following definitions are very convenient in embedded control applications:

truth(X and Y)=Min(truth(X),truth(Y))
truth(X or Y)=Max(truth(X),truth(Y))
truth(not X)=1.0−truth(X)

5. GENETIC ALGORITHM

Genetic algorithm sare parts of artificial intelligence and fuzzy computing and they are mainly used to solve various optimization problems encountered in real-life applications. The basic idea of a genetic algorithm is to mimic the natural selection in nature in order to find a good selection for an application. Genetic algorithm is basically a model of machine learning inspired by the process of evolution in nature. A genetic algorithm can be used for finding solutions complex search problems found in engineering applications. For example, they can search through various designs and components to find the best combination that will result in overall better and cheaper design. Genetic algorithms are used in many diverse fields nowadays, such as climatology, biomedical engineering, code-breaking, control engineering, games theory, electronic design, and automated manufacturing and design. The basic processes in genetic algorithms are: xInitialization, where an initial population is created randomly. xEvaluation, where each member of the population is evaluated and the fitness of the individuals are assessed based on how well they fit the desired requirements. xSelection, where only the ones that fit the desired requirements are selected. xCrossover, where new individual are created by combining best aspects of the existing individuals. At the end of this it is expected to create individuals that are closer to the desired requirements. The process is repeated from the second step until a termination condition is finally reached.

6. EXPERT SYSTEMS

An expert system, also known as a knowledge based system, is a computer based system that can make intelligent decisions by emulating the decision making abilities of human experts. Expert systems are rule based systems and they are part of the artificial intelligence. Expert systems have the abilities that they can change their decisions and make new decisions based on the external factors. Some expert systems are designed to take place of a human in an application, while some others are designed to aid the human. Some application areas of expert systems are: online medical systems for diagnosing a problem, financial loan/credit decisions, legal matters, robotics, and engineering design. One of the main problems in expert systems is the knowledge acquisition. The main components of an expert system are: knowledge base, interface engine, and user interface. The knowledge base is probably the most important part of any expert system. This is where the intelligence of the system is stored. Expert systems in general can acquire new knowledge by their sensors or by training and extend their knowledge bases so that they can easily respond to new problems. The knowledge is stored in the form of IN-THEN-ELSE statements. The interface engine is between the knowledge base and the user. The interface engine makes decisions by following the conditions and the requirements before it comes to an outcome and presents a solution to the user. The user interface is usually in the form of natural language used daily by the user in everyday life. There are basically two types of programming languages: algorithmic and symbolic. Traditional programming languages such as Pascal, Basic C, and Fortran are algorithmic, also known as procedural languages, where it is difficult to implement logical inferences in these languages. Several symbolic languages have been developed over the years for expert systems development, such as Prolog, Lisp, Clips and so on.

6.CONCLUSIONS

Intelligent systems and hence soft computing techniques are becoming more important as the power of computer processing devices increase and their cost is reduced. Intelligent systems are required to make complex decisions and choose the best outcome from many possibilities, using complex algorithms. This requires fast processing power and large storage space which has recently become available in recent years to many research centres, universities, and technical colleges at a very low cost. With the power and the recognition of the Internet of Things (IoT) concept, the need for using soft computing techniques and building intelligent systems have become more important than ever. Nowadays, most soft computing applications can be handled efficiently by low-cost but super-fast microcontrollers. Already we see the use of fuzzy logic, artificial neural networks, and expert systems in many everyday domestic appliances, such as washing machines, cookers, and fridges. Many industrial and commercial applications of soft computing are also in everyday use and this is expected to grow within the next decade. It is the author's opinion that the soft computing theory and techniques and its applications will grow rapidly together with the use of IoT devices in future domestic, industrial and commercial markets

.

7. REFERENCES

1. Zadeh LA. Fuzzy logic, neural networks and soft computing. One-page course announcement of CS 294-4. Spring 1993. University of California at Berkeley; Nov. 1992. 2. Jang JSR, Sun CT, and Mizutani E.Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence.Upper Saddle River, NJ: Prentice-Hall; 1997.3. Zadeh LA. Fuzzy logic, neural networks, and soft computing. Communications of the ACM 1994; vol. 37. no. 3. pp. 77-84.4. Buckley JJ and Hayashi Y. Fuzzy neural networks. A survey. Fuzzy Sets and Systems 1994;66: 1-13. 5. Gupta P, Kulkarni N. An Introduction of Soft Computing Approach over Hard Computing. International Journal of latest Trends 1993.6. Driankov D, Reinfrank M, Hellendoorn H. An Introduction to Fuzzy Control. Berlin: Germany. Springer; 1993. 7. Russo F. Fuzzy systems in instrumentation: Fuzzy signal processing. IEEE Trans Instrumentation and Measurement. vol.45. no.2. 8. Gao XZ and Ovaska SJ. Friction compensation in servo motor systems using neural networks. in Proc.IEEE Midnight-Sun Workshop on Soft Computing Methods in industrial applications. Kuusamo: Finland; June 1999. 9. Komori Y. A neural fuzzy training approach for continue speech recognition improvement. in Proc. International Conference on Acoustics, Speech, and Signal Processing. San

Francisco: CA; 1992: 405-408. 10. Yuhas B, Ansari N. Neural Networks in Telecommunications. Boston: MA. Kluwer Academic; 1994. 11. Cichocki A, Unbehauen R. Neural Networks for Optimization and Signal Processing. West Sussex: UK. John Wiley & Sons; 1993. 12. Dote Y, Hoft RG. Intelligent Control:Power Electronic Systems. Oxford: UK. Oxford University Press; 1998. 13. Roy R, Furuhashi T, Chawdhry PK. Advances in Soft Computing: Engineering Design and Manufacture. London: UK. Springer; 1998.14. Pedrycz W. Fuzzy sets in pattern recognition: Methodology and methods, Pattern Recognition 1990; vol. 23. no. 1: 121-146.15. Bone R, Crucianu M. Multi-step-ahead Prediction with Neural Networks. A review. Publication de l'equipe RFAI. 9 emes Rencontre 689; April 1996.16. Goh ATC. Back-propagation neural networks for modelling complex systems, Artificial Intelligence in Engineering 1995;9(3):143-151. 17. Saerens M. Neural controller based on back-propagation algorithm. IE Proc. On Radar and Signal Processing 2002; 138(1): 55-62. 18. Whitley DA. Genetic algorithm tutorial. Statistics and Computing 1994; 4(2): 65-85. 19. Holland JH. Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. 2nd ed. Cambridge: MA. MIT Press;1992.