SCIENCE BASED OPEN ELECTIVES EOE-031/EOE-041: INTRODUCTION TO SOFT COMPUTING (Neural Networks, Fuzzy Logic and Genetic Algorithm)

Course Objective

• Soft computing refers to principle components like fuzzy logic, neural networks and genetic algorithm, which have their roots in Artificial Intelligence.

• Healthy integration of all these techniques has resulted in extending the capabilities of the technologies to more effective and efficient problem solving methodologies

Learning Outcomes

Upon completion of the course, you should be able to:

• Identify and describe soft computing techniques and their roles in building intelligent machines

• Recognize the feasibility of applying a soft computing methodology for a particular problem

• Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems

• Apply genetic algorithms to combinatorial optimization problems

• Apply neural networks to pattern classification and regression problems

• Effectively use existing software tools to solve real problems using a soft computing approach

• Evaluate and compare solutions by various soft computing approaches for a given problem.

SYLLABUS

Unit-I

Neural Networks-1(Introduction & Architecture)

Neuron, Nerve structure and synapse, Artificial Neuron and its model, activation functions, Neural network architecture: single layer and multilayer feed forward networks, recurrent networks. Various learning techniques; perception and convergence rule, Auto-associative and hetro-associative memory.8

Unit-II

Neural Networks-II (Back propagation networks) Architecture: perceptron model, solution, single layer artificial neural network, multilayer perception model; back propogation learning methods, effect of learning rule co-efficient ;back propagation algorithm, factors affecting backpropagation training, applications. 8

Unit-III

Fuzzy Logic-I (Introduction)

Basic concepts of fuzzy logic, Fuzzy sets and Crisp sets, Fuzzy set theory and operations, Properties of fuzzy sets, Fuzzy and Crisp relations, Fuzzy to Crisp conversion. 8

Unit-IV

Fuzzy Logic –II (Fuzzy Membership, Rules) Membership functions, interference in fuzzy logic, fuzzy if-then rules, Fuzzy implications and Fuzzy algorithms, Fuzzyfications & Defuzzificataions, Fuzzy Controller, Industrial applications. 8

Unit-V

Genetic Algorithm(GA)

Basic concepts, working principle, procedures of GA, flow chart of GA, Genetic representations, (encoding) Initialization and selection, Genetic operators, Mutation, Generational Cycle, applications. 8

Text Books:

1. S. Rajsekaran & G.A. Vijayalakshmi Pai, "Neural Networks,Fuzzy Logic and Genetic Algorithm:Synthesis and Applications" Prentice Hall of India.
2. N.P.Padhy,"Artificial Intelligence and Intelligent Systems" Oxford University Press.

Reference Books:

3. Siman Haykin,"Neural Netowrks"Prentice Hall of India
4. Timothy J. Ross, "Fuzzy Logic with Engineering Applications" Wiley India.
5. Kumar Satish, "Neural Networks" Tata Mc Graw Hill

For UPTU Syllabus
goto: http://www.uptu.ac.in/academics/syllabus/science_based_open_electives_21_04_09.pdf

Contents

1 Soft Computing:

1.1 Introduction

1.1.1 What is Soft Computing?

The idea behind soft computing is to model cognitive behavior of human mind.

Soft computing is foundation of conceptual intelligence in machines.

Unlike hard computing, soft computing is tolerant of imprecision, uncertainty, partial truth, and approximation.

Soft computing (SC) is a branch, in which, it is tried to build intelligent and wiser machines. Intelligence provides the power to derive the answer and not simply arrive to the answer. Purity of thinking, machine intelligence, freedom to work, dimensions, complexity and fuzziness handling capability increase, as we go higher and higher in the hierarchy as shown in Fig. 1.1. The final aim is to develop a computer or a machine which will work in a similar way as human beings can do, i.e. the wisdom of human beings can be replicated in computers in some artificial manner.

Intuitive consciousness/wisdom is also one of the important area in the soft computing, which is always cultivated by meditation. This is indeed, an extraordinary challenge and virtually a new phenomenon, to include consciousness into the computers.

Soft computing is an emerging collection of methodologies, which aim to exploit tolerance for imprecision, uncertainty, and partial truth to achieve robustness, tractability and total low cost. Soft computing methodologies have been advantageous in many applications. In contrast to analytical methods, soft computing methodologies mimic consciousness and cognition in several important respects: they can learn from experience; they can universalize into domains where direct experience is absent; and, through parallel computer architectures that simulate biological processes, they can perform mapping from inputs to the outputs faster than inherently serial analytical representations. The trade off, however, is a decrease in accuracy. If a tendency towards imprecision could be tolerated, then it should be possible to extend the scope of the applications even to those problems where the analytical and mathematical representations are readily available. The motivation for such an extension is the expected decrease in computational load and consequent increase of computation speeds that permit more robust system.

(Jang et al. 1997).

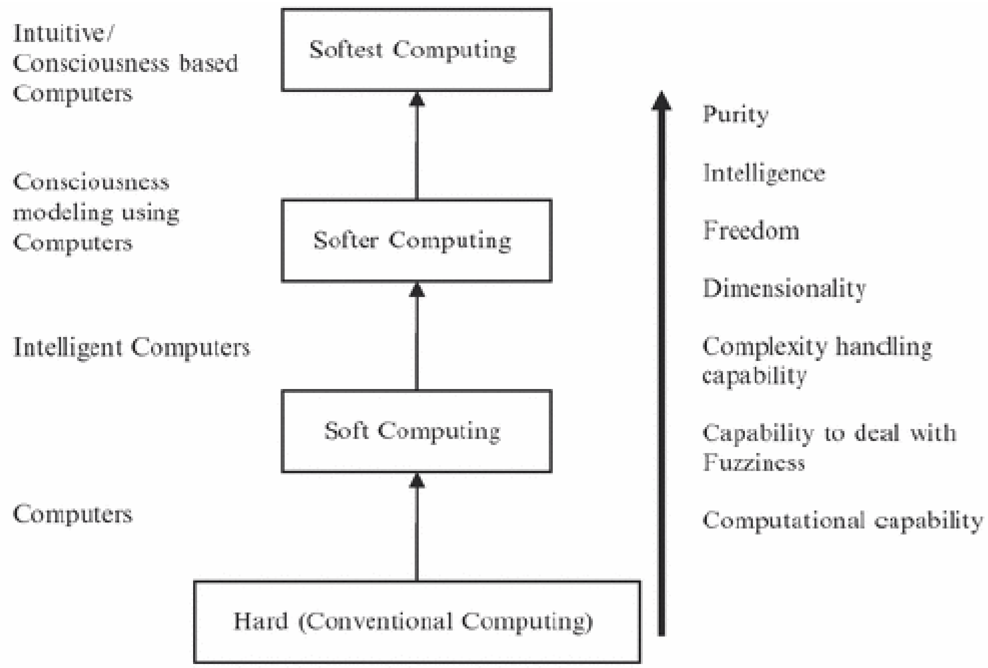

Fig. 1.1. Development soft computing

Soft Computing differs from conventional (hard) computing in many ways. For example, soft computing exploits tolerance of imprecision. uncertainty, partial truth and human mind.

"In effect the role model of soft computing is human mind."

Soft-computing is defined as a collection of techniques spanning many fields that fall under various categories in computational intelligence. Soft computing has three main branches: fuzzy Systems, evolutionary computation, artificial neural computing, machine learning (ML), Probablistic Reasoning (PR), belief networks, chaos theory, parts of learning theory and Wisdom based Expert System (WES), etc.

1.1.2 Hard Vs Soft Computing Paradigms

Hard computing

Based on the concept of precise modeling and analyzing to yield accurate results.
Works well for simple problems, but is bound by the NP-Complete set.

Soft computing

Aims to surmount NP-complete problems.
Uses inexact methods to give useful but inexact answers to intractable problems.
Represents a significant paradigm shift in the aims of computing - a shift which reflects the human mind.
Tolerant to imprecision, uncertainty, partial truth, and approximation.
Well suited for real world problems where ideal models are not available.

1.1.3 Difference b /w Soft and Hard Computing

Hard Computing	***Soft Computing***
Conventional computing requires a precisely stated analytical model.	Soft computing is tolerant of imprecision.
Often requires a lot of computation time.	Can solve some real world problems in reasonably less time.
Not suited for real world problems for which ideal model is not present.	Suitable for real world problems.
It requires full truth	Can work with partial truth
It is precise and accurate	Imprecise.
High cost for solution	Low cost for solution

1.1.4 Unique Features of Soft Computing

Soft Computing is an approach for constructing systems which are computationally intelligent,

possess human like expertise in particular domain,

can adapt to the changing environment and can learn to do better

can explain their decisions

1.1.5 Components of Soft Computing

Components of soft computing include:

Fuzzy Logic (FL)

Evolutionary Computation (EC) - based on the origin of the species
Genetic Algorithm
Swarm Intelligence
Ant Colony Optimizations
Neural Network (NN)
Machine Learning (ML)

1.2 IMPORTANCE OF SOFT COMPUTING

The complementarily of FL, NC, GC, and PR has an important consequence: in many cases a problem can be solved most effectively by using FL, NC, GC and PR in combination rather than exclusively.
A striking example of a particularly effective combination is what has come to be known as "neuro-fuzzy systems." Such systems are becoming increasingly visible as consumer products ranging from air conditioners and washing machines to photocopiers and camcorders. Less visible but perhaps even more important are neuro-fuzzy systems in industrial applications.
What is particularly significant is that in both consumer products and industrial systems, the employment of soft computing techniques leads to systems which have high MIQ (Machine Intelligence Quotient). In large measure, it is the high MIQ of **SC-** based systems that accounts for the rapid growth in the number and variety of applications of soft computing.

— Fuzzy Logic (FL), Neural Networks (NN), Support Vector Machines (SVM), Evolutionary Computation (EC), and — Machine Learning (ML) and Probabilistic Reasoning (PR)

It is widely accepted that the main components of Soft Computing are Fuzzy Logic, Probabilistic Reasoning, Neural Computing and Genetic Algorithms. This four constituents share common features and they are considered complementary instead of competitive. The mentioned technologies can be combined in models which exploit their best characteristics. As an important consequence, some real problems can be solved most effectively by using hybrid systems what is increasing the interest on them. The first and probably the most successful hybrid approach till now are the so-called neurofuzzy systems, although some other hybridations are being developed with great success as, for instance, the genetic fuzzy systems.

Soft computing replaces the traditional time-consuming and complex techniques of hard computing with more intelligent processing techniques. The key aspect for moving from hard to soft computing is the observation that the computational effort required by conventional approaches which makes in many cases the problem almost infeasible, is a cost paid to gain a precision that in many applications is not really needed or, at least, can be relaxed without a significant effect on the solution. A basic difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are fuzzy.

1.2.1 TECHNIQUEs IN SOFT COMPUTING

(Note: Details are found in later Units)

1.2.1.1 Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems. Such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANNs as well.

1.2.1.2 Fuzzy Logic (FL)

FL is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation- based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster.

1.2.1.3 Genetic Algorithns in Evolutionary Computation

A genetic or evolutionary algorithm applies the principles of evolution found in nature to the problem of finding an optimal solution to a Solver problem. in a "genetic algorithm." the problem is encoded in a series of bit strings that are manipulated by the algorithm: in an "evolutionary algorithm," the decision variables and problem functions are used directly. Most commercial Solver products are based on evolutionary algorithms. An evolutionary algorithm for optimization is different from "classical" optimization methods in several ways:

- Random Versus Deterministic Operation
  • Population Versus Single Best Solution
  • Creating New Solutions Through Mutation
  • Combining Solutions Through Crossover
  • Selecting Solutions Via "Survival of the Fittest"
  Randomness. First, it relies in part on random sampling. This makes it a nondeterministic method, which may yield somewhat different solutions on different runs -- even if you haven't changed your model. In contrast, the linear, nonlinear and integer Solvers also included in the Premium Solver are deterministic methods -- they always yield the same solution if you start with the same values in the decision variable cells.
  Population. Second, where most classical optimization methods maintain a single best solution found so far, an evolutionary algorithm maintains a population of candidate solutions. Only one (or a few, with equivalent objectives) of these is "best," but the other members of the population are "sample points" in other regions of the search space, where a better solution may later be found. The use of a population of solutions helps the evolutionary algorithm avoid becoming "trapped" at a local optimum, when an even better optimum may be found outside the vicinity of the current solution.
  Mutation. Third -- inspired by the role of mutation of an organism's DNA in natural evolution ..- an evolutionary algorithm periodically makes random changes or mutations in one or more members of the current population, yielding a new candidate solution (which may be better or worse than existing population members).
  There are many possible ways to perform a "mutation," and the Evolutionary

Solver actually employs three different mutation strategies. The result of a mutation may be an infeasible solution, and the Evolutionary Solver attempts to “repair" such a solution to make it feasible: this is sometimes, but not always, successful.

Crossover. Fourth -- inspired by the role of sexual reproduction in the evolution of living things -o- an evolutionary algorithm attempts to combine elements of existing solutions in order to create a new solution with some of the features of each “parent." The elements (e.g. decision variable values) of *existing* solutions are combined in a "crossover” operation, inspired by the crossover of DNA strands that occurs in reproduction of biological organisms.

As with mutation, there are many possible ways to perform a crossover operation -l- some much better than others and the Evolutionary Solver actually employs multiple variations of two different crossover strategies.

Selection. Fifth -- inspired by the role of natural selection in evolution -. an evolutionary algorithm performs a selection process in which the “most fit" members of the population survive, and the “least fit" members are eliminated. In a constrained optimization problem, the notion of “fitness" depends partly on whether a solution is feasible (i.e. whether it satisfies all of the constraints), and partly on its objective function value. The selection process is the step that guides lhe evolutionary algorithm towards ever-better solutions.

1.3 Applications of Soft Computing

- Handwriting Recognition
- Image Processing and Data Compression
- Automotive Systems and Manufacturing
- Soft Computing to Architecture
- Decision-support Systems
- Soft Computing to Power Systems
- Neuro Fuzzy systems
- Fuzzy Logic Control
- Machine Learning Applications
- Speech and Vision Recognition Systems
- Process Control and So On

1.4 FUTURE OF SOFT COMPUTING

Soft computing is likely to play an especially important role in science and engineering, but eventually its influence may extend much farther.

Soft computing represents a significant paradigm shift in the aims of computing. A shift which reflects the fact that the human mind, unlike present day computers, possesses a remarkable ability to store and process information which is pervasively imprecise, uncertain and lacking in categoricity.