BIOINFO Soft Computing
Volume 1, Issue 1, 2011, pp-05-07
Available online at: http://www.bioinfo.in/contents.php?id=304

Soft Computing and its Applications

[1]Kurhe A.B., [2]Satonkar S.S., [3]Khanale P.B. and [4]Shinde Ashok
[1]Department of Computer Science, Shri Guru Buddhiswami College, Purna, (M.S)
[2]Arts, Commerce and Science College, Gangakhed, (M.S.), India
[3]Dnyanopasak College, Parbhani, (M.S.)
[4]N.S.B College, Nanded
e-mail: ajay.kurhe@rediffmail.com, prb_suhas@rediffmail.com, prakash_khanale@hotmail.com

Abstract— Recently new technique is available for computation known as Soft computing. Soft computing is based on natural as well as artificial ideas. Soft Computing techniques are Fuzzy Logic, Neural Network, Support Vector Machines, Evolutionary Computation and Machine Learning and Probabilistic Reasoning. The present paper shows the techniques, applications and future of soft computing. The Soft Computing Techniques & applications is also highlighted in the paper.

Keywords: Soft Computing, Neural Network, FL, GA

I. INTRODUCTION

Soft Computing (SC) is an emerging field that consists of complementary elements of fuzzy logic, neural computing, evolutionary computation, machine learning and probabilistic reasoning. Due to their strong learning and cognitive ability and good tolerance of uncertainty and imprecision, soft computing techniques have found wide applications.

II. WHAT IS COMPUTING?

The discipline of computing is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application. The fundamental question underlying all computing is 'What can be (efficiently) automated?

III. WHAT IS HARD COMPUTING?

- Hard computing, i.e., conventional computing, requires a precisely stated analytical model and often a lot of computation time.
- Many analytical models are valid for ideal cases.
- Real world problems exist in a non-ideal environment

1. *Hard Computing*

- Premises and guiding principles of Hard Computing are
- Precision, Certainty, and rigor.
- Many contemporary problems do not lend themselves to precise solutions such as
- Recognition problems (handwriting, speech, objects, images
- Mobile robot coordination, forecasting, combinatorial problems etc.

IV. WHAT IS A SOFT COMPUTING?

Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect, the role model for
Soft computing is the human mind.

The principal constituents, i.e., tools, techniques, of Soft Computing (SC) are – Fuzzy Logic (FL), Neural Networks (NN), Support Vector Machines (SVM), Evolutionary Computation (EC), and – Machine Learning (ML) and Probabilistic Reasoning (PR)

2. *Premises of soft computing*

- The real world problems are pervasively imprecise and uncertain
- Precision and certainty carry a cost

3. *Principles of soft computing*

Exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness and low solution cost.

4. *Implications of soft computing*

- Soft computing employs NN, SVM, FL etc, in a complementary rather than a competitive way.
- One example of a particularly effective combination is what has come to be known as "neurofuzzy systems."
- Such systems are becoming increasingly visible as consumer products ranging from air conditioners and washing machines to photocopiers, camcorders and many industrial applications.

5. *Unique Property of Soft Computing*

- Learning from experimental data
- Soft computing techniques derive their power of generalization from approximating or interpolating to produce outputs from previously unseen inputs by using outputs from previous learned inputs

BIOINFO Soft Computing
Volume 1, Issue 1, 2011, pp-05-07
Available online at: http://www.bioinfo.in/contents.php?id=304

- Generalization is usually done in a high dimensional space.

V. Techniques in Soft Computing

A. 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 (neurones) 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 neurones. This is true of ANNs as well.

B. Fuzzy Logic

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.

C. Support Vector Machines

Are a set of related supervised learning methods used for classification and regression. In simple words, given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. More formally, a support vector machine constructs a hyperplane set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training datapoints of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

D. Genetic Algorithms 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 -- 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 -- 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 the evolutionary algorithm towards ever-better solutions.

VI. 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 "neurofuzzy 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 neurofuzzy 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.

VII. 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

VIII. 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.

IX. CONCLUSION AND SCOPE FOR FUTURE RESEARCH

In this paper we have given an Soft Computing Techniques, applications and future of Soft Computing.

The present paper can provide the readers a better understanding about Soft Computing and techniques and the topic is open to further research.

REFERENCE

[1] Onsen TOYGAR, Adnan ACAn, “ Face Recognition using PCA, LDA and ICA approaches on colored images”.
[2] Kailash J. Karande, Sanjay N. Talbar, “Face recognition under Variation of Pose and Illumination using Independent Component Analysis”.
[3] Shang-Hung Lin, “An Introduction to Face Recogntion Technology”.
[4] Fromherz T., Stucki P., Bichsel M., “A survey of Face Recognition”.
[5] Calvo R., Partridge M., and Jabri M., “A Comparative Study of Principal Components Analysis Techniques”.
[6] Yambor W.S.,”Analysis of PCA- Based and Fisher Discriminant- Based Image Recognition
[7] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins,” Digital Image Processing Using MATLAB”
[8] S. Jayaraman *et. al*.,” Digital Image Processing”.
[9] Anil K. Jain, “Fundamental of Digital Image Processing”.
[10] Digital Image-processing fundaments are given in this website: http://www.ph.tn.tudelft.nl/courses/FIP/frames/fip.html
[11] S.N. Sivanandan, ”Principals of Soft Computing”