An introduction to soft computing

[1]RACHITA RAUL, Gandhi Institute of Excellent Technocrats, Bhubaneswar, India

[2]PRIYANKA MOHANTY, Indus College of Engineering, Bhubaneswar, Odisha, India

Abstract

As opposed to classical computing, soft computing deals with approximation models and provides answers to challenging real-world issues. Soft computing, in contrast to hard computing, is accepting of approximations, ambiguity, partial truth, and imprecision. In actuality, the human mind serves as a model for soft computing. Techniques like fuzzy logic, genetic algorithms, artificial neural networks, machine learning, and expert systems are the foundation of soft computing. Even though soft computing theory and methods were initially established in the 1980s, automatic control engineering is now focusing a lot of research and study on this field. Nowadays, numerous home, commercial, and industrial applications successfully employ soft computing approaches. Due to the development of inexpensive, extremely fast digital processors and the decrease in the price of *Keywords:*Soft computing; fuzzy logic; genetic algorithms; neural networks; expert system.

1.   Introduction

One of the problems in traditional control systems is that complex plants cannot be accurately described by mathematical models, and are therefore difficult to control using such existing methods. Soft computing on the other hand deals with partial truth, uncertainty, and approximation to solve complex problems. Dr Zadeh[1] who is the pioneer of fuzzy logic quoted that "the guiding principle of soft computing is to exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution cost, better rapport with reality". Because of its features such as intelligent control, nonlinear programming, optimization, and decision making support, soft computing has become popular and has drawn research interest from people with different backgrounds, Jang et al[2].

It is becoming difficult to control the growing complexity of modern machinery using traditional control systems techniques. For example, many nonlinear and time-variant plants with large time delays cannot easily be controlled and stabilized using traditional techniques. One of the reasons for this difficulty is the lack of an accurate model that describes the plant. Soft computing is proving to be an efficient way of controlling such complex plants.

Zadeh[3] pointed out that soft computing is not a single method, but instead it is a combination of several methods, such as fuzzy logic, neural networks, and genetic algorithms. All these methods are not competitive, but are complimentary to each other and can be used together to solve a given problem[4]. It can be said that soft computing aims to solve complex problems by exploiting the imprecision and uncertainty in decision making processes.

Fig. 1 shows the conventional and soft computing based problem solution principle as suggested by Gupta and Kulkarni[5]. The left diagram shows the traditional hard computing approach where an exact model of the plant under investigation is available and traditional mathematical methods are used to solve the problem. The right diagram shows soft computing approach where only an approximate model of the plant may be available, and the solution depends upon approximate reasoning techniques.
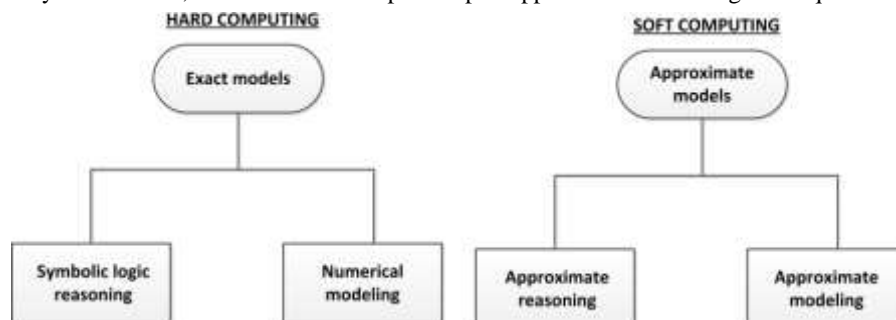
Fig. 1 Problem-solving approach

Fuzzy control has been in use for over two decades to solve complex control problems, Driankov et al[6]. In addition, many instrumentation problems are being solved using fuzzy logic principles as reported by Russo[7].

Neural networks, although a newer concept, have also been used by many people to solve complex automatic control problems, including the demanding servo problems[8].

In addition to solving automatic control problems, soft computing has also been used in diverse applications such as in intelligent speech recognition[9], communications[10], fields of signal processing[11], heavy current systems[12], design and manufacturing[13], pattern recognition[14], and many more applications.

This paper is an overview of soft computing techniques and describes some of the commonly used techniques to solve complex problems with soft computing methods, such as fuzzy logic, neural networks, genetic algorithms, andexpert systems.

2. Fuzzy logic

The concept of fuzz logic was introduced by Zadeh[3] as a method for representing human knowledge that is imprecise by nature. Fig. 2 shows the basic configuration of a fuzzy logic system.

The fuzzification interface transforms the crisp input value into a fuzzy linguistic value. The fuzzification is always necessary in a fuzzy logic system since the input values from existing sensors are always crisp numerical values. The inference engine takes the fuzzy input and the fuzzy rule base and generates fuzzy outputs. The fuzzy rule base is in the form of "IF-THEN" rules involving linguistic variables. The last processing element of a fuzzy logic system is the defuzzification which has the task of producing crisp output actions.

Perhaps one of the biggest advantage of fuzzy logic is that it offers a practical way for designing nonlinearcontrol systems which are difficult to design and stabilize using traditional methods.
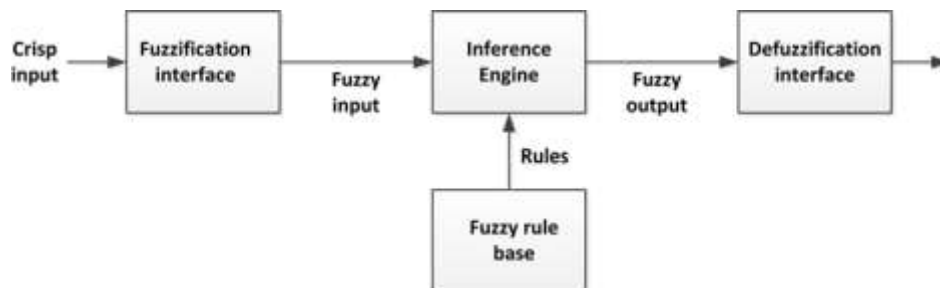

Fig. 2 Architecture of a fuzzy logic system

3. Artificial neural networks

Artificial neural networks (ANN), or neural computing is one of the rapidly growing fields of research, attracting researchers from a wide variety of engineering disciplines, such as electronic engineering, control engineering, and software engineering.

ANNs are information processing systems that are inspired by the way biological nervous system and the brain works. ANNs are usually configured for specific applications, such as pattern recognition, data recognition, image processing, stock market prediction, weather prediction, image compression, and security and loan applications. Neural networks aim to bring the traditional computers a little closer to the way human brain works. ANNS work best if the relationship between the inputs and outputs are highly non-linear. ANNs are highly suitable for solving problems where there are no algorithms or specific set of rules to be followed in order to solve the problem.

A neural network is a large network of interconnected elements, inspired by the human neurons. Each neuron performs a little operations and the overall operation is the weighted sum of these operations.

A neural network has to be trained so that a known set of inputs produces the desired outputs. Training is usually done by feeding teaching patterns to the network and letting the network to change its weighting function according to some previously defined learning rules. The learning can either be *supervised*, or *unsupervised*. In supervised learning the network under investigation is trained by giving it inputs and matching output patterns. i.e. the outcomes are known for specific inputs. In unsupervised learning the

output of the network is trained to respond to input patterns.

Some of the advantages and disadvantages of neural networks are:

- ANNs are not universal tools for solving problems as there is no methodology for training and verifying an ANN.
- The result of an ANN depends upon the accuracy of the available data
- Excessive training may be required in complex ANN systems
- ANNs can deal with incomplete data sets
- ANNs are successful in prediction and forecasting applications[15].

An ANN is basically composed of three layers: input, hidden layer, and output, where each layer can have number of nodes. Backpropagation algorithm[16,17] is used in most ANN networks as a method to train the network. Here, output of the neural network is evaluated against desired output, and if the results are not as expected, the weights between layers are modified and the process is repeated until a very small error remains.

4. Genetic algorithms

Genetic algorithms[18,19] are 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 cheaperdesign.

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:

- Initialization, where an initial population is created randomly.
- Evaluation, where each member of the population is evaluated and the fitness of the individuals are assessedbased on how well they fit the desired requirements.
- Selection, where only the ones that fit the desired requirements are selected.
- Crossover, 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.

5. 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 ahuman 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 and the future

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.

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.