IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 7,Issue 3,July 2019

In-depth explanation of "soft computing"

V RADHIKA DEVI, B.SWATHI,G SURENDRA

Abstract

In contrast to classical computing, soft computing works with approximations of models and provides answers to difficult, real-world situations. Soft computing, in contrast to traditional computing, can tolerate fuzziness, doubt, half-truths, and approximations. The human brain serves as the inspiration for soft computing. Fuzzy logic, genetic algorithms, artificial neural networks, machine learning, and expert systems are all examples of the kinds of soft computing that form the basis of the field. A significant field of study in automated control engineering, soft computing has been around since the 1980s but has only recently been widely accepted as a viable theory and set of methods in its own right. These days, soft computing methods are effectively used in a wide variety of consumer, business, and manufacturing settings. It is certain that the methodologies and application areas of soft computing will continue to develop with the emergence of low-cost and extremely high performance digital processors and the drop in the cost of memory chips. This article provides an overview of contemporary soft computing approaches, contrasting them with more conventional hard computing methods and detailing the benefits and drawbacks of each.

Keywords: Fuzzy logic, genetic algorithms, neural networks, and expert systems are all examples of soft computing.

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 Complex plants are notoriously difficult to handle using present techniques of regulation because they defy proper mathematical modeling. Soft computing, on the other hand, is concerned with solving complicated problems in the face of partial truth, ambiguity, and approximation. For example, Dr. Zadeh1, the father of fuzzy logic, said, "The driving idea of soft computing is to leverage the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, resilience, cheap solution cost, greater rapport with users, and other desirable properties." reality". Jang et al.2 state that the rise in popularity of soft computing may be attributed to the fact that it facilitates study from a wide range of disciplines due to its intelligent control, nonlinear programming, optimization, and decision making support characteristics.

As contemporary equipment increases in complexity, it becomes more challenging to regulate using conventional control system methods. Typical methods of control and stabilization are inadequate for many complex systems, such as nonlinear and time-variant plants with significant time delays. The absence of a reliable model to explain the plant is contributing to the problem. Controlling such complex plants using soft computing is proven to be an effective method.

According to Zadeh3, soft computing is not a single technique but rather a collection of approaches, such as fuzzy logic, neural networks, and evolutionary algorithms. Each of these approaches stands on its own, but they also complement one another and may be used in tandem to address a specific issue4. The goal of soft computing is to find workable solutions to difficult issues by taking advantage of human fallibility and ambiguity throughout the decision-making process.

Asst. Professor
Department of cse
SRINIVASA INSTITUTE OF TECHNOLOGY AND SCIENCE , KADAPA
Chennai- Hyderabad Bypass Road, Ukkayapalli ,kadapa -516002

Solution principles proposed by Gupta and Kulkarni5 are shown in Fig. 1 for both traditional computing and soft computing. On the left, we see how the issue may be solved using the time-honored method of "hard computing," in which an accurate model of the plant under study is at hand and standard mathematical techniques are used. Soft computing, seen on the right, is used when just a rough model of the plant is available, necessitating the use of approximation reasoning methods in order to find a work

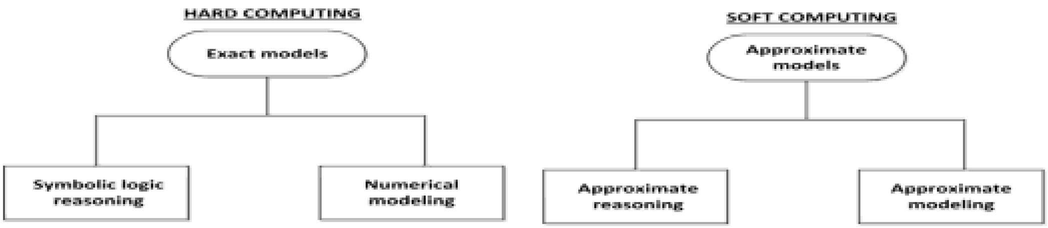

Fig. 1 Problem-solving approach

As Driankov et al.6 point out, fuzzy control has been used for over two decades to address difficult control issues. According to Russo7, fuzzy logic is also being used to provide answers to several issues related to instrumentation.
Although neural networks are a relatively recent idea, they have already been put to use in solving difficult automated control issues, such as the demanding servo problems8.
Soft computing has been employed in a wide variety of contexts outside automated control, including but not limited to: intelligent voice recognition9, communications10, areas of signal processing11, heavy current systems12, design and manufacturing13, pattern recognition14, and many more.

This article provides an introduction to the field of soft computing and covers some of the most popular approaches to solving complicated issues using these methods.

2. Fuzzy logic

Zadeh3 presented the idea of fuzz logic as a way to describe the imprecise nature of human knowledge. The fundamental structure of a fuzzy logic system is seen in Fig. 2.

The crisp input value is converted to a fuzzy linguistic value through the fuzzification interface. Since the input data from preexisting sensors are always precise numerical values, fuzzification is always required in a fuzzy logic system. Fuzzy outputs are produced by the inference engine once the input and the fuzzy rule base have been processed. The "IF-THEN" rules containing linguistic variables make up the fuzzy rule base. Defuzzification is the last step of a fuzzy logic system's processing, and its job is to provide crisp output actions. For nonlinear control systems, which are notoriously difficult to design and stable, fuzzy logic provides a realistic solution.

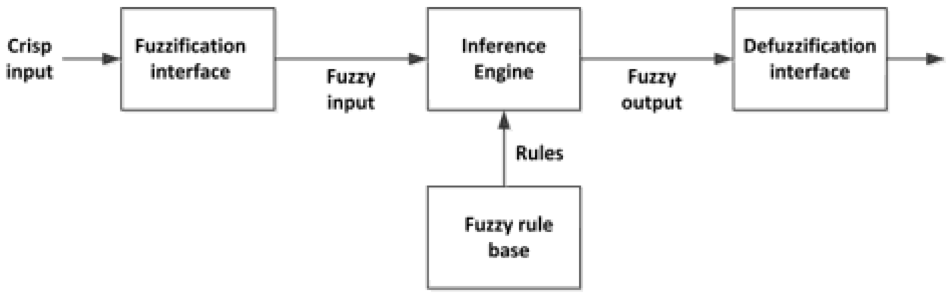

Fig. 2 Architecture of a fuzzy logic system

3. Artificial neural networks

Researchers in domains including electrical engineering, control engineering, and software engineering are flocking to the fast expanding subject of artificial neural networks (ANN), often known as neural computing.

Artificial neural networks (ANNs) are data processors that mimic the functionality of the human brain and nervous system. Pattern identification, data recognition, image processing, stock market prediction, weather prediction,

ISSN 2347–3657
Volume 7,Issue 3,July 2019

picture compression, and lending and security applications are just some of the many tasks for which ANNs are often trained. The purpose of neural networks is to make conventional computers perform more like the human brain. An very non-linear connection between inputs and outputs is ideal for ANNS. ANNs excel at tackling issues for which there are neither a predetermined set of instructions nor an established method.

A neural network is a complex system of linked nodes that mimics the organization of the human nervous system. Small operations are carried out by each neuron, and the total operation is the weighted sum of these small activities.

To get the intended results from a neural network given a certain set of inputs, the network must be trained. In most cases, a network is trained by providing it with training patterns and then allowing it to adjust its weighting function based on a set of predetermined rules. Supervised or unsupervised learning is possible. Supervised learning is a kind of network training in which the target network is given inputs and trained using patterns of expected outputs. That is, for a given set of inputs, the outputs are known in advance. In unsupervised learning, the network's output is modified without any human oversight.

There is no uniform approach for training and validating ANNs, therefore their results rely on the quality of the data fed into them. Extensive training may be necessary for complicated ANN systems. ANNs are able to handle insufficient input.

• ANNs perform well in contexts requiring prediction and forecasting15.

The input, the hidden layer, and the output are the three main parts of an ANN, and each of these layers may have any number of nodes. The majority of ANN networks use the backpropagation algorithm16,17 to educate the network. Here, the neural network's output is compared to the intended output, and if the results are off, the weights between layers are adjusted and the procedure is repeated until the error is very tiny.

4. Genetic algorithms

Artificial intelligence and fuzzy computing, of which genetic algorithms18,19 are subsets, are widely used to address a wide range of practical optimization issues. In order to identify a suitable option for an application, genetic algorithms attempt to replicate natural selection. To put it simply, a genetic algorithm is a model of machine learning that takes its cues from the natural process of evolution. In engineering applications, complicated search problems may be solved with the help of a genetic algorithm. By way of illustration, they may look through through a number of iterations of design and component testing in order to arrive at an optimal configuration that both improves upon and reduces the cost of the final product.

Climate science, biomedical engineering, code-breaking, control engineering, games theory, electrical design, automated manufacturing, and design are just some of the numerous areas that today make use of genetic algorithms.

In genetic algorithms, the fundamental steps are as follows:

The first step is called "initialization," and it entails the creation of a random population.

• Evaluating the population as a whole to determine which members are most desirable and how effectively they meet those needs.

• Selecting candidates based on how well they meet certain criteria.

• Crossover, in which superior qualities of several persons are combined to form a new one. People who better fit the ideal profile will hopefully emerge as a result.

From then, you'll repeat steps 2 through whichever circumstance triggers the end of the process.

5. Expert systems

Knowledge-based systems, often known as expert systems, are computer programs that may act as if they were human experts in a certain field. A subset of AI, expert systems are rule-based programs that can handle complex problems independently of human input. In response to new information or circumstances, expert systems might adjust their previous judgments or formulate brand-new ones. While some expert systems are designed to do the work of humans, others are intended to supplement them. Expert systems have several uses, including online medical diagnostics, loan/credit judgments, the law, robotics, and engineering design. Expert system knowledge acquisition is a key challenge.

Knowledge bases, interface engines, and user interfaces form the backbone of expert systems. When it comes to expert systems, the knowledge base is crucial. That's where all the brains of the machine are hidden. In general, expert systems may grow their knowledge bases via sensing or training, making it easier for them to deal with novel challenges. IN IF THEN ELSE statements are used to store the information. The interface engine is the link between the user and the underlying knowledge base. The interface engine considers the given circumstances and needs, then arrives at a conclusion and provides a solution for the user to review. Interfaces often take the shape of the user's ordinary speech. Languages that use algorithms and languages that use symbols are the two main categories of computer programming languages. Algorithmic, or procedural, languages are the foundation of traditional programming, but they make it challenging to include logical assumptions. Over the years, a number of symbolic languages—including Prolog, Lisp, Clips, and others—have been created specifically for use in creating expert systems.

6. Conclusions and the future

As the processing power of computers grows and their prices drop, the importance of intelligent systems and, by extension, soft computing approaches, rises. Intelligent

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 7,Issue 3,July 2019

systems are needed because of the complexity of the judgments they must make and the variety of outcomes they must choose from. This calls for fast processing power and ample storage capacity, both of which have been widely accessible at relatively cheap cost to a wide range of research institutions, universities, and technical schools in recent years.

The importance of adopting soft computing approaches and developing intelligent systems has increased dramatically with the rise in popularity of the idea of the Internet of Things (IoT). These days, low-cost yet lightning-fast microcontrollers can easily handle the majority of soft computing applications.

Many commonplace kitchen and laundry equipment now incorporate some kind of artificial intelligence, such as fuzzy logic, neural networks, or expert systems. Soft computing is already widely used in many commercial and industrial settings, and this trend is predicted to accelerate over the next decade.

The author predicts that the usage of IoT devices in future residential, industrial, and commercial industries will increase fast, and with it will come increased need for soft computing theory and methods and their implementations.

References

1. Fuzzy reasoning, neural networks, and the rise of "soft" computing. In the spring of 1993, a one-page course notice for CS 294-4 was sent. This was written in November of 1992 at the University of California, Berkeley.
2. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence by Jang JSR, Sun CT, and Mizutani E. Prentice-Hall, Upper Saddle River, NJ, 1997.
3. Fuzzy reasoning, neural networks, and the rise of soft computing. ACM Computing Communications, Issue 37, Pages 77–84, 1994.
4. Fuzzy neural networks., by John McCarthy Buckley and Yoshio Hayashi. To conduct a survey. Systems with Fuzzy Boundaries 1994;66: 1-13.
5. An Introduction to the Soft Computing Approach as an Alternative to Hard Computing. Review of Emerging Technologies and Innovations 1993.
6. The sixth group is Driankov, Reinfrank, and Hellendoorn. Fuzzy Control: A Primer. Berlin, Germany. Paperback edition published by Springer in 1993.
7. The use of fuzzy logic and fuzzy systems to instrumentation and signal processing by Frank Russo. The article may be found in issue 45-2 of the IEEE Transactions on Instrumentation and Measurement.
8. Number eight, Gao XZ and S.J. Ovaska. Neuro-fuzzy systems for servo-motor friction adjustment. Applied Soft Computing Methods at the IEEE Midnight-Sun Workshop Proceedings. In the month of June of 1999, Kuusamo, Finland was the location of the Olympic Winter Games.
9. 9 Komori, Y., "A neural fuzzy training strategy for continuing speech recognition enhancement," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. S.F., CA: 1992, pp. 405–408.
10. Yuhas B, Ansari N. Neural Networks in Telecommunications. 10. Kluwer Academic Publishers: Boston, MA, 1994.
11. Neural Networks for Optimization and Signal Processing, by Cichocki A. and Unbehauen R. UK; West Sussex. John Wiley & Sons, 1993.
12. Power Electronic Systems, 12th Edition, by Dote Y and Hoft RG. UK: Oxford University Press, 1998.
13. Roy R, Furuhashi T, and Chawdhry PK. Technical Development and Production of Soft Computing Systems. English capital city of London, United Kingdom. Publication date: Springer, 1998.
14. Fuzzy Sets in Pattern Recognition: Methodology and Techniques, by W. Pedrycz Pattern Recognition, volume 23, issue 1, 1990, pages 121-146.
15. R. Bone and M. Crucianu. Multi-step-ahead Neural network-based forecasting. A look back. Team RFAI Publication; April 1996; 9 emes Rencontre 689.
16. In the article "Back-propagation neural networks for modeling complex systems," published in the journal Artificial Intelligence in Engineering in 1995, page numbers 143-151, author Anthony T.C. Goh describes the use of such networks to model such systems.
17. Saerens M. 17. Back-propagation-based neural controller. Radar and Signal Processing, IE Proceedings, Vol. 138, No. 1, 2002, Pages 55–62.
18. Eighteenth Citation: Whitley DA. Genetic Algorithm Tutorial. The year 1994 saw the publication of Statistics and Computing Volume 4 Issue 2 Pages 65–85.
19. Adaptation in Natural and Artificial Systems. 19. Holland, J.H. An Overview Analyzed with Real-World Examples in Biology, Control, and AI. Cambridge, Mass. : MIT Press, 1992. Second edition.