An Introduction to Soft Computing — A Tool for Building Intelligent Systems

B Azvine, N Azarmi and K C Tsui

Intelligent Systems Research, Advanced Applications & Technology Department,
BT Laboratories, Martlesham Heath, Ipswich, Suffolk, IP5 7RE, UK.
E-mail: ben.azvine@bt-sys.bt.co.uk
azarmin@info.bt.co.uk
tsuikc@info.bt.co.uk

> "The essence of soft computing is that unlike the traditional, hard computing, soft computing is aimed at an accommodation with the pervasive imprecision of the real world. Thus, the guiding principle of soft computing is: '...exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness, low solution cost and better rapport with reality'. In the final analysis, the role model for soft computing is the human mind." [1]

In this paper terms associated with soft computing are defined and its main components are introduced. It is argued, using a number of practical applications, that the hybrid approach of soft computing can provide a methodology for increasing machine intelligence.

1. Introduction

One of the primary issues in artificial intelligence (AI) has been the choice between two fundamentally different (and often viewed as competing) approaches to building intelligent systems — traditional symbolic AI and numeric (sub-symbolic) artificial neural networks (ANNs). This has been an issue engaging the AI community for three decades, and there have been attempts to bridge the gap between these two paradigms in order to take advantage of the relative merits of each [1].

In an attempt to model the human mind/brain it has been necessary to oversimplify the structure (resulting in ANNs) and the function (resulting in precisely defined symbolic- AI programmes) of the brain. Symbolic AI attempts to pre-program intelligence into a deterministic algorithm. On the other hand, most ANNs are equipped with relatively weak forms of learning (i.e. tuning a fixed set of parameters or weights). It has been argued [2] that despite the seemingly different approaches that symbolic AI and ANNs take to building intelligent systems, they both share common origins, and are both based on the hypothesis that cognition can be modelled by computation. Tasks performed by ANNs can be performed by symbolic AI and vice versa as both paradigms rely on different but essentially equivalent models of computation [2]. Decades of collective experience by theoreticians and practitioners in several areas of computer science have shown that it is efficiency, robustness and elegance that determine the best approach. Hybrid systems resulting from the integration of concepts and technologies drawn from both traditional AI systems and ANNs clearly demonstrate the potential benefits for the design of truly robust, flexible and adaptive intelligent systems in a wide application domain. This paper concentrates on one of many promising approaches for developing hybrid intelligent systems known as soft computing (SC). SC is not a single methodology, rather it is a partnership. The principal partners at this juncture are fuzzy logic (FL), neuro-computing (NC), and probabilistic reasoning (PR) which subsumes genetic algorithms (GA), chaotic systems, belief networks and parts of learning theory.

The term soft computing was coined by Zadeh, the inventor of fuzzy set theory, to be an extension to fuzzy logic by merging it mainly with neural networks and evolutionary computing. A concise definition for SC is: 'A term that describes a collection of techniques capable of dealing with imprecise, uncertain or vague information'. Zadeh advocates that SC has the means to extend what conventional AI has achieved in the past 40 years, and that a prerequisite to building an intelligent machine is a model of human cognitive capability [3]. The philosophical argument for SC is stated elegantly in Mamdani [4].

It is obvious that humans deal with uncertain and imprecise information everyday and are remarkably consistent in processing such information. This is the primary aim of SC — to exploit the tolerance of imprecision, uncertainty and partial truth associated with almost every aspect of real-world problems.

The aim of this paper is to describe SC in terms of the techniques associated with it, and how they are being combined to produce hybrid systems. There are three sections describing three techniques accepted as the main (but by no means the only) constituents of SC — fuzzy logic, neural networks and genetic algorithms. The final section describes hybrid techniques and their applications in the industry.

2. Intelligent Systems from a Soft Computing Perspective

There are many features that can be attributed to an intelligent system. Among them one can mention robustness, adaptivity, autonomy and the ability to communicate, including man/machine communication in multiple modalities. Dealing with real-world uncertainty or robustness is one of the most important characteristics of an intelligent system (Fig 1).

Uncertainty arises from many sources among which are nonlinear behaviour, time-varying behaviour (e.g. degradation over time) and interaction with uncertain environments. All of these features are present in human behaviour and therefore are important in the context of machines that interact with, co-operate with or replace humans in certain tasks. Humans do not seem to be as affected by uncertainty in sensory data as present-day computing machines. One explanation is that humans do

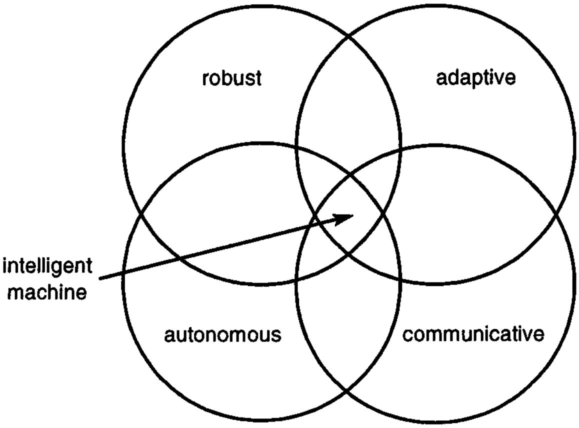

Fig. 1. Essential features of an intelligent machine.

not rely directly on raw data for decision making but on abstract, uncertain rules, e.g. in the rule 'if it is cold, put on an extra jumper', the actual temperature is not important, neither is the season nor the time of day. A definite advantage of using abstract rules is that a large amount of irrelevant information can be filtered out and the decision-making process is simplified. This is particularly important in the context of machines that rely on search techniques. It can be argued that in many real applications the relevant information belongs to a class that is not well defined, and its membership changes from time to time. To use the above example 'cold' means one thing today and its meaning may change next week, or next month. A fixed rule may be able to deal with this particular task, but it lacks the degree of adaptability required to work in changing environments. Humans seem to have the ability to change with their environment. Adaptive behaviour can be captured in a machine by using symbolic meta-level rules. For example, a rule can be defined that adjusts other rules according to a mean temperature based on the season, such that 'if it is winter then the mean temperature is 10°C', and 'in summer the mean temperature is 30°C', and define cold relative to these. This provides a fixed meta-level rule and an adaptive base-level rule. This is a partial solution, but what if there is an exceptional circumstance such as a particularly cold winter. This highlights one of the shortcomings of such an approach, namely that fixed symbolic meta-level rules can be restrictive in some circumstances.

The question that arises is how are these rules derived? Humans develop general rules from specific observations and then generalise from specific instances to new situations. For example, it can be seen that touching a specific hot object will result in pain and personal injury, so a general rule is developed — 'if an object is hot, do not touch it'. In this case it is assumed that we have some sensory information received by one or several of our five senses as to the temperature of an object. On the other hand, generally, humans do not develop rules for recognising friends' faces. Picking a familiar face in a crowd is performed instantaneously. Humans recognise vast numbers of patterns and exhibit many skills without having to develop rules for them or even know the rules that would result in such behaviours. Studies of the human brain have shown that the pre-attentive processing of stimuli is carried out in as few as 70 to 100 ms. We look, see, pay attention and then recognise without using rules. Then we address the higher cognitive functions such as reasoning, decision making, planning and control using rules. An intelligent machine therefore should be able to combine signal-level (sub-symbolic) intelligence with symbolic, more abstract level intelligence (rules). In this sense intelligence can become a property of a hybrid dynamical system.

SC enables the pre-processing of sensory information, reasoning using symbolic rules, and learning directly from observations. Adaptive systems must be able to develop rules for themselves and update the rules in view of new sensory data, i.e. learn from their experience. Learning (adaptability) is the second important feature of an intelligent system (Fig 1). Learning can be viewed as change in system behaviour based on experience. From a dynamical system's point of view, learning is the rate of change of an analytic function describing the system's behaviour [5]. As the analytic function is a mapping from the input to the output space, it can therefore be a collection of rules or a mathematical function. Clustering techniques are one aspect of learning addressed by SC.

Application of SC to real-world problems has been aimed at increasing machine intelligence quotient (MIQ). MIQ is measured by the level of control that a system can have over its own operation (autonomy) (Fig 1). For example a robot that can navigate its way around obstacles has a certain MIQ, another that can navigate and cope with unforeseen moving obstacles has a higher MIQ. Another measure for MIQ is the degree to which the machine assists humans in a particular task, e.g. a washing machine that chooses its own program has a certain degree of MIQ, another one that can program itself and use cheap electricity has a higher MIQ.

3. Fuzzy Logic in Brief

There are two reasons for using fuzzy logic in real applications. Firstly, in certain circumstances the definition of the problem is vague and uncertain. The information available does not lend itself readily to precise mathematical reasoning as in rule-based systems. A second class of applications are well defined but a precise solution is not necessary; the tolerance for imprecision can be exploited to simplify the solution. Most of the applications of fuzzy logic today fall into the second category.

Fuzzy logic can be viewed as a superset of Boolean logic in the sense that it can handle the concept of partial truth [6]. This concept has been used to develop more general extensions such as fuzzy calculus and fuzzy differential equations. Fuzzy logic is based on the principle of fuzzy subsets. In classical set theory, based on Boolean logic, membership of a subset U can be defined as a mapping from the elements of a set S to the elements of another set L with two members: 0 and 1. So an element is mapped to 0 if it is not a member of S and to 1 if it is. In fuzzy sets, a similar mapping exists to a set F with the difference that F contains all values between 0 and 1. This gives rise to the concept of degree of truth. A mapping to 0.3 is less true than a mapping to 0.6. The degree of membership of a set is defined by a membership function μ. Boundaries of fuzzy subsets are not sharp but fuzzy and overlapping. This implies that a particular entity A could be a member of two subsets with different degrees of membership — $\mu_1(A)$ and $\mu_2(A)$.

There are two important concepts that are central to the application of fuzzy logic:

- a linguistic variable;
- fuzzy if-then rules.

A linguistic variable is a variable that takes linguistic values such as height, age, speed, quality, etc. Such variables can take linguistic values like tall, young, fast, good, etc. A linguistic value is a label for a fuzzy set. Within fuzzy sets, degree of membership is characterised by membership functions, e.g. a membership function 'tall' determines the degree of tallness of someone of a certain height as shown in Fig 2. For example, it depicts a person who is 180 cm as tall with a degree of membership 0.45, while someone who is 185 cm tall is 'tall' with a degree of membership 0.6.

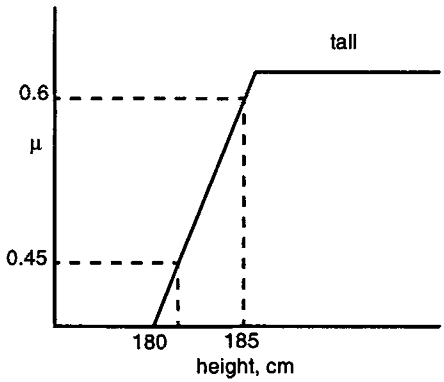

Fig. 2 Membership function 'tall' in a fuzzy set height.

A linguistic variable can be defined as a micro-language with context-free grammar and attributed-grammar semantics [3]. The context-free grammar defines the legal linguistic values for the variable and the grammar semantics defines the membership functions for any value within the linguistic variable, using the membership functions of primary terms. For a linguistic variable height, the legal values could be tall, short, very tall, not very tall, almost tall. The grammar semantics in this case define the membership functions for all values in terms of two primary membership functions — tall and short. For example:

$$\mu_{\text{not very tall}}(A) = 1 - (\mu_{\text{tall}}(A))^2$$

This introduces the idea of linguistic hedges or modifiers such as very, more or less and almost. Such terms are used extensively in natural language in a purely subjective way and therefore do not have a universal definition within different applications.

However, once they have been defined, consistency can be ensured within a particular application. Some common definitions are:

$$\mu_{\text{very}}(A) = \mu(A)^2$$

$$\mu_{\text{more or less}}(A) = \mu_{\text{ tall}}(A)^{\frac{1}{2}}$$

The reason fuzzy logic includes such terms is that linguistic terms are essential to the way humans perceive, reason and communicate. By using words, people compress data to achieve economy of communication. Fuzzy logic aims to exploit this important feature of natural language combined with the consistency of a logical approach.

Another way of looking at the degree of membership of a value in a fuzzy set is a possibility distribution, e.g. $\mu(A) = 0.45$ is equivalent to the statement: 'the possibility that A is tall is 0.45'. It should be noted that this is quite different to the statement: 'the probability of A being tall is 0.45', since probability represents randomness and depends on the frequency of observations, while the possibility depends on uncertainty and vagueness and remains the same irrespective of the number of observations. As long as the definition of tall is fixed by the membership function, A would always be tall to the same degree.

Fuzzy rules in their most simple form can be expressed as if X is A then Y is B, where A and B are fuzzy values. This can be represented by the cartesian product of A and B i.e. $A \times B$, so that the membership function of the above rule can be written as:

$$\mu_{A \times B}(X,Y) = \mu_A(X) \wedge \mu_B(Y)$$

where $\wedge$ is the conjunction operator usually defined as $min(A{\times}B)$. For example, a fuzzy rule can express a simple rule of thumb. If X is tall then X is a good basketball player'.This rule can be visualised in terms of two membership functions defining 'tall' and 'good_basketball_player' as shown in Fig 3. Any player of any height has a certain degree of being a good basketball player between 0 and 1.

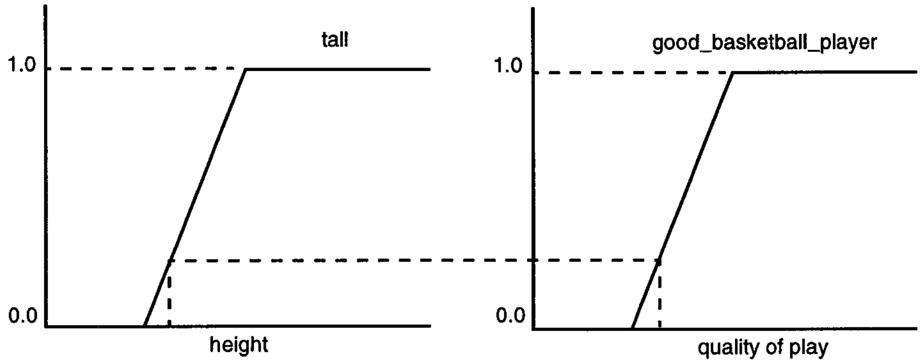

Fig. 3. Visualisation of a fuzzy rule.

A collection of fuzzy rules can be represented as a fuzzy graph. Fuzzy rules can be written as if *X* is A_i then *Y* is B_i, $i = 1...n$. For example, consider the following simple rules:

- if *X* is small then *Y* is large;
- if *X* is medium then *Y* is small;
- if *X* is large then *Y* is large.

This system can be represented by a fuzzy graph, f^* as shown in Fig 4. A fuzzy graph represents a coarse characterisation of functional dependency between *X* and *Y*. In this context, interpolation of rules becomes an important issue, i.e. what value of *Y* results if the input *X* is not a perfect match with any of the antecedent variables defined. This is carried out by considering each fuzzy rule and its degree of truth. Then defuzzification of the outputs is performed using one of many available techniques. The most widely used is the centre of gravity method. Interpolation is one of the most important features of fuzzy systems which can be exploited in situations where complex functional relationships are to be represented by a small number of fuzzy rules. This has been demonstrated in a number of complex industrial problems where the number of fuzzy rules have been typically between 10 and 20 [7].

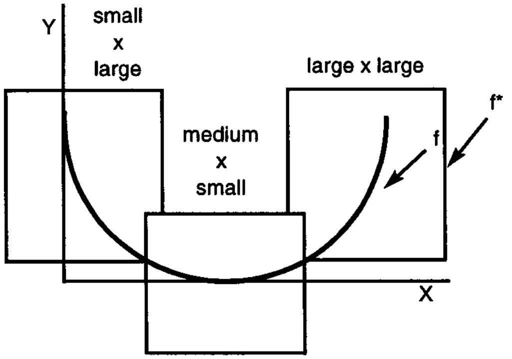

Fig. 4. Visualisation of a fuzzy graph.

One of the central issues in fuzzy logic is how to induce rules from observations. This is the problem of obtaining deep structure from surface structure [8]. It is relatively easy to write down a set of fuzzy rules to describe a particular behaviour. However, to calibrate these rules, i.e. to choose the type and characterisation of the membership functions, is not a trivial problem. A number of techniques have been used to solve this problem, such as dynamic and gradient programming (developed for multi-stage optimisation), genetic algorithms, reinforcement learning, and trial and error [8].

4. Artificial Neural Networks

The structure of the brain has been the subject of intense research in the past several decades. Many of the pioneers of AI drew upon biological inspiration for their work. Analogies were established between artificial processing elements and real neurons, between network connection and axions, and between connection strengths and synapses. A key aspect of the brain that ANNs try to imitate is its parallelism. ANNs achieve this by using densely interconnected simple processing units to store and process information. Each aspect of the neuron is represented mathematically by real numbers. The basic processing unit, or artificial neuron, is characterised by a set of input connections, a set of output connections, an activation level, an output level, and a bias value (Fig 5).

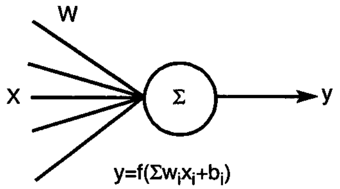

Fig. 5. A simple artificial neuron.

The output level of a neuron is determined according to a function of the activation level, which is a weighted sum of the signal from the input connections. ANNs have many characteristics such as nonlinear mapping, self-organisation and learning. Learning in ANNs is viewed as the problem of finding a set of connection weights so that given a set of inputs the desired outputs are generated. ANNs effectively perform a parallel version of curve fitting and their capabilities should be assessed as adaptive function approximators [5]. When viewed in this manner they are powerful tools that can be used in an intelligent system to give it the learning capability. Many learning algorithms have been proposed, mostly network-architecture-specific. Supervised-learning algorithms rely on a teacher module to provide a set of training data which contains the input and the associated expected output. The learning algorithm then minimises the difference between the network output and the expected output. A possible application is a function- learning task. Unsupervised-learning algorithms take only the input patterns as training data and try to organise the neurons which best classify the data [9]. With reinforce-ment-learning algorithms, instead of providing a desired output for each input as in supervised learning, only a scalar reinforcement signal is used, which may be available only occasionally. Typical application of reinforcement learning algorithms is in automatic control applications like the pole balancing problem [9].

A most common ANN architecture, called the feedforward net, arranges neurons into layers, namely input, hidden and output layers. Connections are restricted to the area between neurons in different layers. Many learning algorithms have been developed to train such neural networks. Most applications that employ feedforward net use the backpropagation algorithm for learning. In a multi-layer network, the input is coded into an internal representation, and it is this internal representation that generates the outputs. Given a large enough set of hidden units, it is possible to perform any mapping from the input set to the output set.

Another common type of ANN that allows connections between any two neurons is called the recurrent net. This allows complex interaction between the neurons. An example is the Hopfield net which has been used as an associative memory. There are connections between any two neurons in a Hopfield net, and some neurons are designated as input while others, not necessarily different, as output. Upon completion of training, a Hopfield net is capable of retrieving the stored information when a part of the string used during training is presented to the network. Relaxing the restriction of having only inter-layer feedforward connections leads to the development of a network architecture called the Elman net. For each neuron in the hidden layer, there are backward connections to some neurons in the input layer. These extra connections allow the network to include temporal information during its course of deriving a solution [10].

It is clear that the most important contribution that ANNs make to a smart system is that they make it adaptive. ANNs can automatically adjust their weights using the learning algorithm to optimise the system behaviour. This capability allows ANNs to continually track solutions in changing environments. However, it is not true to say that ANNs would be able to compete with conventional techniques at performing well-defined, precise, numerical calculations such as matrix inversion. They have produced the best results in problems that not only involve ambiguity but are also difficult to model, such as pattern recognition. It is this key characteristic of ANNs that is exploited within the framework of SC.

5. Genetic Algorithms

Genetic algorithms (GAs) are search techniques which derive their inspiration from biological natural selection and genetics. The starting point is a population of individuals, each representing a possible solution to a problem. Each individual is allocated a fitness measure according to the quality of the solution it produces. The fittest individuals survive to the next generation while the individuals that produce unsatisfactory solutions are eliminated. This represents survival of the fittest. The transition from one generation to the next is by means of reproduction among the survived individuals only. The reproduction results in new individuals as offspring who share some features taken from each parent.

A basic GA processes a finite population of binary strings. There are three basic operations — selection, crossover and mutation. Selection chooses two individuals to produce offspring. The primary objective of selection is to produce a partial ranking of the population so that fitter individuals will have a higher chance to reproduce. Crossover takes the two selected individuals and divides randomly their binary representation into two sections, called heads and tails. The two tails are then swapped to produce new individuals. For example 11111 and 00000 can produce two new strings 11000 and 00111, or 11110 and 00001. Mutation is applied to each offspring after crossover. It is an occasional alteration of a bit (gene) position. The quality of the offspring is evaluated in the same way as the parents.

The GA's application domain is wide. It can be applied to any optimisation problem and has produced impressive results in a number of applications [11]. General Electric developed a CAD system that combined expert systems with genetic algorithms. This system was used on a 100-variable portion of a gas-turbine design and produced a 92% increase in efficiency over human designers.

Variations to the basic form of the GA described above include real number and integer representation, different selection schemes that give various reproduction advantage to fitter individuals, and crossover operators that divide a string into more than two sections. Common to various forms of GA is their robustness in reaching an optimal solution in the presence of minimal, if any, prior knowledge of the problem at hand. It is also best used in situations which involve a large number of parameters. As a result, the search conducted by a GA is very computationally intensive. Recent research has produced an analytical theory of GAs based on stochastic differential equations which may further establish GAs as an efficient tool for optimisation and simulation of distributed systems.

6. Soft Computing and Hybrid Systems

In the past decade a number of hybrid techniques have been developed that take advantage of the relative merits of fuzzy systems, ANNs and GAs [12]. In the previous sections, these merits were discussed and can be summarised as shown in Table 1.

There are five categories in Table 1 used for the comparison — learning and optimisation refer to sub-symbolic learning and optimisation. (The techniques have been assessed on the basis of whether learning and optimisation are implicit features or have to be built in.) Knowledge extraction refers to symbolic knowledge extraction as defined in conventional AI systems. Real-time operation is linked with implementation issues, i.e. whether each method lends itself readily to hardware implementation or not. Knowledge representation is either symbolic or numeric. The entries for fuzzy systems, ANNs, GAs and conventional systems are as shown. It should be noted that Table 1 also assumes a simplistic binary set of entries — yes and no. In reality, they could themselves be fuzzy.

Two observations can be made from Table 1. Firstly, there is a good case for combining fuzzy, ANNs and GAs for building intelligent systems because each method can complement the other. Secondly, such a combination can enhance the capabilities of conventional AI systems. Some proponents of SC [13] use this as a strong argument for developing new ways of producing hybrid systems to address the shortcomings of conventional AI.

It has been argued [3] that the success of SC (as indicated by an explosion of applications in the present decade) is due to its emphasis on computational intelligence (CI) which is for the most part numeric rather than symbolic. CI is defined by Bezdek

[14] as the first step of achieving biological, or human-level, intelligence (BI), and it is purely based on numerical computation using sensory signals. AI lies somewhere between CI and BI and can be achieved by extending CI with symbolic representation and manipulation of non-numeric data (see Table 2). Fuzzy models are particularly suitable for a smooth transition between CI and AI because they can accommodate both numeric and symbolic information in a common framework.

Table 1. Relative merits of Fuzzy, ANN, GA and conventional AI systems.

	Sub-symbolic learning	Symbolic knowledge extraction	Real-time operation	Knowledge repre-sentation	Optim-isation
Fuzzy system	no	yes	yes	symbolic/ numeric	no
ANN	yes	no	yes	numeric	no
GA	yes	no	no	numeric	yes
Conven-tional AI systems	no	yes	no	symbolic/ number	no

Table 2. The ABC of intelligence [14].

Complexity

Complexity

Biological Human knowledge + sensory inputs	BNN	BI
Artificial Symbolic + numeric + sensor data	ANN	AI
Computational Numeric	CNN	CI

Bezdek presents the case for combining symbolic and sub-symbolic techniques by introducing a distinction between computational neural networks (CNNs) and artificial neural networks (ANNs). He argues [14] that ANNs result from the combination of CNNs, which are based purely on numerical processing of sensory data, and some knowledge usually in the form of non-numeric rules.

The most widely researched hybrid system in the area of SC is neuro-fuzzy systems. Here the learning capabilities of ANNs are exploited within the framework of fuzzy logic. In some systems, ANNs can be used to generate and tune the membership functions in a fuzzy system (Fig 6). A number of models have been suggested for such hybrid systems, e.g. fuzzy ART [15], Fuzzy LVQ [16] and radial basis functions [17]. The process of obtaining and tuning the fuzzy rules is one that is particularly suitable for ANNs, resulting in substantial reductions in cost and development time. Here, gradient descend methods have been used to define the shape and position of membership functions. This hybrid method has been used to design triangular, Gaussian, sigmoidal and bell-shaped membership functions [18].

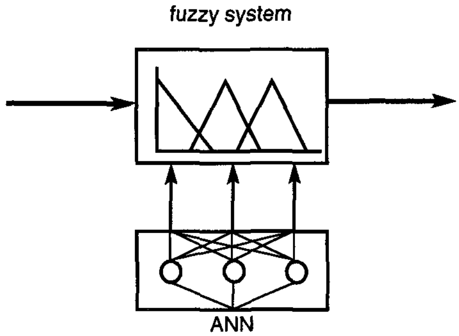

Fig. 6. *ANN for tuning membership functions.*

A number of applications have been reported in which fuzzy systems and ANNs are employed in series [19]. In such situations, either the sensor output is not suitable for direct input to the fuzzy system in which case an ANN pre-processes the input to the fuzzy system (Fig 7(a)), or the output of the fuzzy system is not suitable for direct interface with the external devices and an ANN is used as a post-processor to perform a mapping or conversion not easily achievable by other analytical techniques (see Fig 7(b)). For example, Toshiba's microwave-oven-cum-toaster estimates the temperature and the number of pieces of bread using an ANN and decides the optimum toasting time by using fuzzy reasoning, i.e. its model resembles closely Fig 7(a).

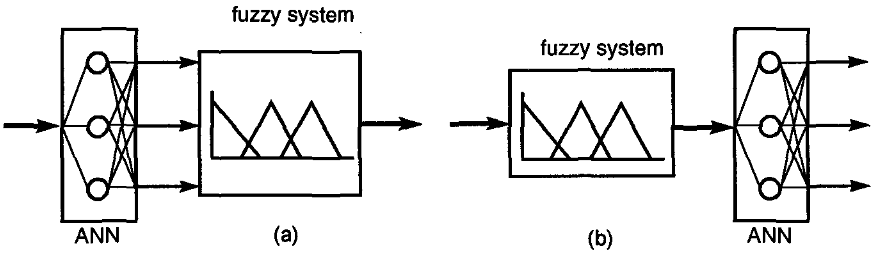

Fig. 7. Serial hybrid ANN-fuzzy systems.

In some applications ANN and fuzzy systems are used in parallel. One possible configuration uses a fuzzy system as the main system and an ANN to fine-tune the output to suit users' personal preferences. The ANN learns from the fine adjustments made by the user and corrects the output of the fuzzy system (Fig 8).

Another class of systems, known as neural fuzzy systems, have been used for knowledge acquisition and learning. In such systems, experts' knowledge in symbolic form is used to initialise a structured ANN. The ANN is then trained using the input/

output from an actual system. Symbolic knowledge is then acquired from the trained ANN in fuzzy logic representation (Fig 9).

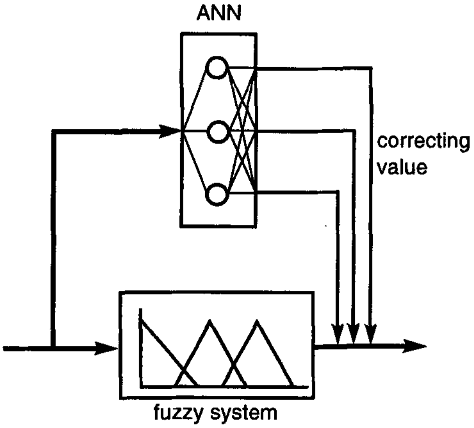

Fig. 8. Parallel hybrid ANN/fuzzy systems.

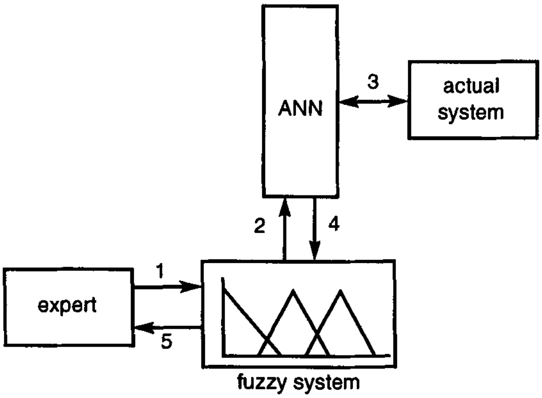

Fig. 9. Neural/fuzzy systems.

A significant number of researchers have concentrated their efforts on implementing fuzzy systems on a neural network representation. These include fuzzy weights, fuzzy neurons and fuzzy neural networks in which the neural network layers perform the fuzzification and defuzzification on crisp input/output data. The structure of fuzzy neural networks is shown in Fig 10. There are three groups of layers each performing one of the three functions of a fuzzy system.

The initial layers process crisp input data by assigning groups of nodes to the labels of linguistic variables and implementing membership functions in these nodes. The output of these layers goes to layers that function as fuzzy rules operating on fuzzy input. The final layer aggregates the results of applying the rules and defuzzifies the results [19]. GARIC [20] is an example of such a scheme which uses a five-layer neural network. It has been used in the space shuttle orbital operations control by NASA.

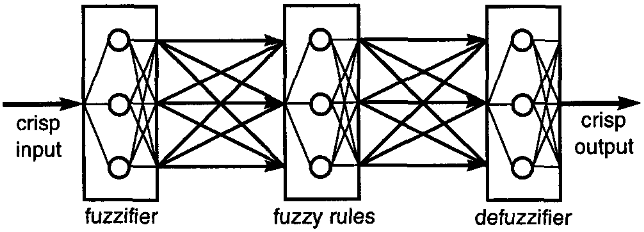

Fig. 10. Fuzzy/neural networks.

Development tools are becoming available for integrating fuzzy and neural networks which should pave the way for the exploitation of the available architectures in information systems applications. Two such tools are NEFCON-I [21] and O'INCA [19] by Intelligent Machines Inc.

Fuzzy-GA systems combine the optimisation capabilities of GAs with fuzzy logic (Fig 11). Such systems can develop the best possible set of rules for use by a fuzzy inference engine. They can also be used to optimise the choice of the membership functions.

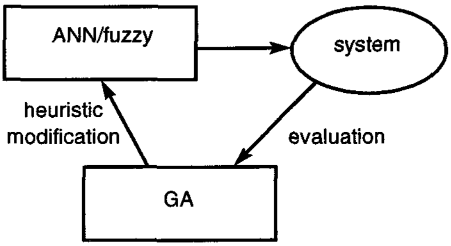

Fig. 11. GA for optimisation of ANN/fuzzy hybrid systems.

The applications of hybrid genetic and fuzzy systems are in adaptive process control, pattern recognition, robot trajectory generation and face recognition. GAs can be used to improve the performance of neural networks by changing their parameters, topologies or both. Applications include structure organisation of fuzzy neural networks, evolving ANNs and self-organising maps.

There are many other possible combinations of FL, ANN and GA. However, this paper has only presented the most widely published ways of combining these techniques within the context of SC. The references provide a more comprehensive list for further reading.

7. Some Applications of Soft Computing Techniques

Although SC is a relatively new field of research it has already been established as one of the fastest growing areas of AI technology in terms of consumer and industrial applications. Japanese companies lead the way both in research and development, although many European and American companies are allocating large resources to this technology and the products have already appeared in the market. The majority of the applications are in the areas of expert systems, control, pattern recognition, clustering and image processing. However, active research is being carried out in many other areas such as decision support systems, user interfaces, speech recognition, face recognition and natural language systems. This section gives a brief description of a small number of recent applications of this technology. The emphasis here will be on applications resulting from the fusion of SC techniques rather than applications of individual techniques.

7.1 A Smart Washing Machine

Hitachi has produced a washing machine that uses the parallel structure of Fig 8 [22]. The smart controller inside the washing machine determines the washing programme automatically by measuring the amount and type of clothes placed inside the machine. A fuzzy rule-based system then uses this information to control the water flow and the programme parameters such as washing time, rinsing time and spinning time. A neural network monitors the operation of the machine during a wash and uses the quality of the water inside the drum to fine-tune the output parameters of the fuzzy system. In this way, it acts as an adaptive correcting mechanism for the fixed fuzzy rules.

7.2 A User-seeking Electric Fan

The electric fan of Sanyo is designed to face the user as the user moves inside a room [23]. To solve this problem accurately a very sophisticated and expensive system is required, which is not suitable for a relatively cheap consumer product. The main problem is computing the distance of the user from the fan. Sanyo have designed a fuzzy system to estimate the distance given readings from an infra-red sensor. A neural network is then trained to use this information to compute the required turning angle of the fan. This technique has not only produced a financially viable solution but it has 2.5 times better accuracy compared to statistical regression methods.

7.3 A Photocopier with a 'Brain'

The Matsushita Electric photocopier machine controls its operation with a set of fuzzy rules [24]. All photocopier machines operate with a set of fixed parameters which can be adjusted manually by an engineer or the user. However, the quality of the copies can deteriorate with time or can be dependent upon the type of the original document. A fuzzy rule-based system can use the information about the state of the machine, the quality and type of the original document to make decisions regarding operating parameters of the photocopier. Some of the fuzzy input parameters used by the rules are temperature, humidity, toner density, image density and image background. The output of the system controls the parameters such as exposure lamp, drum voltage and toner density among others. Interestingly, the parameters of the fuzzy system were designed automatically by neural networks, i.e. the position and width of the fuzzy membership functions were tuned by a gradient method. It is important to note that the neural network was used during the development stage and not during the operation. The same approach has been used by Matsushita to design vacuum cleaners, rice cookers and washing machines.

7.4 A Rolling Mill with Fuzzy Recognition

Hitachi have manufactured and run a rolling mill system since 1991 [25] whose aim is to produce constant thickness metal rolls. The surface of the plate which is being driven through 20 rolls is scanned. The scanned pattern is matched against standard template patterns by a neural network. The standard templates are used as the antecedent of the fuzzy rules, for example:

IF pattern is *template_1* THEN action is *output_1*
IF pattern is *template_2* THEN action is *output_2*
IF pattern is *template_3* THEN action is *output_3*

The level of matching identified by the neural network is the strength of each fuzzy rule. The aggregated final output of the fuzzy system determines the output to the rolls. This approach is called neural network-driven fuzzy reasoning.

7.5 Other Interesting Consumer Products

Mitsubishi Electric Corp introduced fuzzy inference into their videoconferencing system in 1991 [26]. The aim was to improve the data compression coding method based on the extent of change in successive frames. The fuzzy rules improved the motion tracking ability of their product by 30—50%. A smart TV produced by Mitsubishi [26] continuously adjusts the controls on the TV set to produce optimum picture quality according to the brightness of the room and the viewer's distance from the set. Canon has used fuzzy logic in their camera to improve the auto-focus functions [26]. Sanyo have done similar work on their camcorders [26]. Siemens have done extensive work on the application of fuzzy reasoning to various aspects of ATM networks such as call admission control and usage parameter control [27].

There are many more of such applications as documented in many journals and books [24]. The wealth of sources of applications show that the soft computing technology is reliable, cost effective and applicable to real-world problems. These are the factors that make soft computing an attractive technology from an industrial point of view. However, there are many active research areas within the soft computing framework that are and will be producing new directions for exploiting this technology in other challenging areas of applications.

7.6 Human/Computer Interaction — A New and Challenging Area of Application

This area of research can benefit from soft computing in many different ways because of the inherent uncertainty and vagueness in natural language, image recognition, hand-writing recognition, speech recognition and gesture understanding. The uncertainty is either due to poor sensor technology and data, or lack of processing algorithms and background information. There are already products available that use component technologies of soft computing to perform many of the above mentioned tasks e.g. neural networks have been used extensively for image and speech recognition, fuzzy logic has been used in areas such as face recognition [28], hand-writing recognition and speech recognition. However, as discussed in section 2, the merger of these technique would improve the overall characteristics of the resulting system and therefore it is anticipated that this will be an active area of research in the next few years. In the next section we will briefly mention future trends in soft computing research.

8. Future Research Directions

The majority (70%) of the publications in this area are concerned with the fusion of fuzzy systems and neural networks (FS-NN). About 25% of the publications are in the area of combining neural networks and GAs (NN-GA), and the remainder are in the area of merging fuzzy systems and GAs (GA-FS). The most promising areas in FS-NN are in the automatic design of fuzzy systems using neural networks and in neural networks whose structure is based on fuzzy rules (generally similar to that shown in Fig 10) which has produced results significantly superior compared to conventional neural networks. Within NN-GA, GAs have been used for optimisation of synaptic weights in ANNs and have produced better results when combined with back-propagation (BP) compared to BP on its own. In the GA-FS area the performance of static GA has been improved by incorporating a set of fuzzy rules to dynamically change the parameters of the GA in order to improve its overall performance. On the other hand GAs have been used to optimise the selection of best fuzzy rules as well as optimising the rules themselves. In general, improvements are being made in the areas of soft computing where individual components seem to have deficiencies. In summary the future research directions are as follows:

- a better understanding of the trade-offs between training time and size of neural networks is necessary;
- neural network implementations of fuzzy systems must be able to learn on-line in order to respond to changes in their environment;
- we must be able to extract the knowledge learnt by neural networks;
- GAs must be able to handle qualitative fitness functions as well as quantitative ones;
- generic soft computing platforms are essential for further research in these areas.

These are just some of the challenges faced by the soft computing research community.

9. Conclusions

This paper has given a definition for soft computing and described its relevance to intelligent systems. The principal aim of soft computing is to achieve robustness, low solution cost and high machine IQ, through the exploitation of tolerance for imprecision and uncertainty. The individual components of soft computing each exhibit certain characteristics beneficial to the aim of increasing MIQ. Fuzzy logic provides a model for approximate reasoning, as well as a representation for smooth transition from a symbolic paradigm to a numeric one. Neural networks operate on numeric data and provide low-level, fast-processing units that can adapt and learn. GAs are used for optimisation to evolve better performance. A number of successful applications, particularly in the area of consumer products, have shown that synergism of these techniques can provide a route to building intelligent systems [29].

Acknowledgement

The authors wish to acknowledge the advice and encouragement provided by Professor Lotfi Zadeh in the preparation of this paper.

References

1. Zadeh L A: 'The roles of fuzzy logic and soft computing in the conception, design and deployment of intelligent systems', BT Technol J, 14, No 4, pp 32—36 (October 1996).
2. Uhr L and Honavar V: 'Introduction', in: 'Artificial Intelligence and Neural Networks — Steps Toward Principled Integration', Academic Press (1994).

3. Zadeh L A: 'Fuzzy Logic, Neural Networks and Soft Computing', Comm of ACM, 37, No 3, pp 77—84 (March 1994).

4. Mamdani E H: 'Towards Soft Computing', Proc BCS Expert Systems Conference, Cambridge (December 1995).

5. Kosko B: 'Neural Networks and Fuzzy Systems — A Dynamical Systems Approach to Machine Intelligence', Prentice Hall (1992).

6. Horstkotte E: http://www.quadralay.com/www/ Fuzzy/Fuzzy.html

7. Munakata T and Jani Y: 'Fuzzy Systems: An Overview', Comm of ACM, 37, No 3, pp 69—76 (March 1994).

8. Zadeh L A: 'Soft Computing and Fuzzy Logic', IEEE Software, 11, No 6, pp 48—58 (1994).

9. Barto A, Sutton R and Anderson C: 'Neuro-like Adaptive Elements that can Solve Difficult Control Problems', IEEE Tran on Systems, Man and Cybernetics, No 13 (1983).

10. Dayhoff J: 'Neural Network Architectures', Van Nostrand Reinhold (1990).

11. Goldberg D E: 'Genetic and Evolutionary Algorithms Come of Age', Comm of ACM, 37, No 3, pp 113—119 (March 1994).

12. Fukura T: 'Fuzzy-neuro-GA Based Intelligent Robotics', in: 'Computational Intelligence Imitating Life', IEEE Press, pp 352—363 (1994).

13. Fox J: 'Towards a reconciliation of fuzzy logic and standard logic', Int J of Man Machine Studies, 15, pp 213—220 (1981).

14. Bezdek J C: 'What is Computational Intelligence', in: 'Computational Intelligence Imitating Life', IEEE Press, pp 1—12 (1994).

15. Carpenter G, Grossberg S, Markuzon N, Reynold H J and Rosen D B: 'Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analogue Multi-dimensional Maps', IEEE Tran on Neural Networks, 3, No 5, pp 698—713 (1992).

16. Tsao E C K, Bezdek J C and Pal N R: 'Image Segmentation Using Fuzzy Clustering Networks', in North American Fuzzy Information Processing, pp 98—107 (1992).

17. Wang L X: 'Training Fuzzy Logic Systems Using Nearest Neighbourhood Clustering', Manuscript (1992).

18. Asakawa K and Takagi H: 'Neural Networks in Japan', Comm of ACM (March 1994).

19. Medsker A R: 'Hybrid Intelligent Systems', Kluwer Academic Publishers (1995).

20. Berenji H R: 'Fuzzy Systems That Can Learn', in: 'Computational Intelligence Imitating Life', IEEE Press, pp 23—30 (1994).

21. NEFCON: http://sol.ibr.cs.tu-bs.de/ibr/projects/ nefcon/

22. Hitachi News Release: 'Neuro and fuzzy logic automatic washing machine and fuzzy logic dryer', No. 91-024 (February 1991).

23. Sanyo News Release: 'Electric fan series in 1991', (March 1991).

24. Takagi H: 'Co-operative system of neural networks and fuzzy logic and its application to consumer products', in: 'Industrial Applications of Fuzzy Control and Intelligent Systems', IEEE Press (1994).

25. Nakajima M, Okada T, Hattori S and Morroka Y: 'Application of pattern recognition and control techniques to shape control of the rolling mill', Hitachi Review, 75, No 2 (1993).

26. Takagi H: 'Survey of fuzzy logic applications in image processing equipment', in: 'Industrial Applications of Fuzzy Control and Intelligent Systems', IEEE Press (1995).

27. Hellendoorn H, Metternich W, Nissel M, Seising R and Thomas C: 'Traffic management for broadband networks with fuzzy logic — call admission control and usage parameter control', Proceedings of EUFIT'96 (September 1996).

28. Baldwin J and Martin T: 'Basic concepts of fuzzy logic data browser with applications', in Nwana H and Azarmi N (Eds): 'Software Agents and Soft Computing: Towards Enhancing Machine Intelligence', Springer Verlag, Berlin (December 1996).

29. Zadeh L A: 'Foreword', in Medsker L R: 'Hybrid Intelligent Systems', Kluwer Academic Publishers (1995).