Fuzzy Logic, Neural Networks, and **Soft Computing**

LOTFI A. ZADEH

In retrospect, the year 1990 may well be viewed as the beginning of a new trend in the design of household appliances, consumer electronics, cameras, and other types of widely used consumer products. The trend in question relates to a marked increase in what might be called the Machine Intelligence Quotient (MIQ) of such products compared to what it was before 1990. Today, we have microwave ovens and washing machines that can figure out on their own what settings to use to perform their tasks optimally; cameras that come close to professional photographers in picture-taking ability; and many other products that manifest an impressive capability to reason, make intelligent decisions, and learn from experience.

There are many factors that underlie the marked increase in MIQ. It is the author's opinion that the most important factor is the use of what might be referred to as *soft computing*—and, in particular, fuzzy logic—to mimic the ability of the human mind to effectively employ modes of reasoning that are approximate rather than exact.

In traditional—hard—computing, the prime desiderata are precision, certainty, and rigor. By contrast, the point of departure in soft computing is the thesis that precision and certainty carry a cost and that computation, reasoning, and decision making should exploit—wherever possible—the tolerance for imprecision and uncertainty.

A case in point is the problem of parking an automobile. Most people are able to park an automobile quite easily because the final position of the vehicle and its orientation are not specified precisely. If they were, the difficulty of parking would grow geometrically with the increase in precision and eventually would become unmanageable for humans. What is important to observe is that the problem of parking is easy for humans when it is formulated imprecisely and difficult to solve by traditional methods because such methods do not exploit the tolerance for imprecision.

The exploitation of the tolerance for imprecision and uncertainty underlies the remarkable human ability to understand distorted speech, decipher sloppy handwriting, comprehend nuances of natural language, summarize text, recognize and classify images, drive a vehicle in dense traffic and, more generally, make rational decisions in an environment of uncertainty and imprecision. In effect, in raising the banner of "Exploit the tolerance for imprecision and uncertainty," soft computing uses the human mind as a role model and, at the same time, aims at a formalization of the cognitive processes humans employ so effectively in the performance of daily tasks.

As was observed earlier, the year 1990 may be viewed as a turning point in the evolution of the MIQ of consumer products. The basis for this observation are the following facts.

The industrial applications of fuzzy logic starting in the early 1980s—of which the prime examples are the F.L. Smidth cement kiln and the Sendai subway system designed by Hitachi—laid the groundwork for the use of fuzzy logic in the design and production of high-MIQ consumer products. The first such product—a fuzzy-logic-controlled shower head—was announced by Matsushita in 1987. This was followed by the first fuzzy-logic-based washing machine—also designed by Matsushita—in 1989.

In 1990, high-MIQ consumer products employing fuzzy logic began to grow in number and visibility. Somewhat later, neural network techniques combined with fuzzy logic began to be employed in a wide variety of consumer products, endowing such products with the capability to adapt and learn from experience. Such neurofuzzy products are likely to become ubiquitous in the years ahead. The same is likely to happen in the realms of robotics, industrial systems, and process control.

It is from this perspective that the year 1990 may be viewed as a turning point in the evolution of high-MIQ products and systems. Underlying this evolution was an acceleration in the employment of soft computing—and especially fuzzy logic—in the conception and design of intelligent systems that can exploit the tolerance for imprecision and uncertainty, learn from experience, and adapt to changes in the operating conditions.

At this juncture, the principal constituents of soft computing are fuzzy logic (FL), neural network theory (NN), and probabilistic reasoning (PR), with the latter subsuming belief networks, genetic algorithms, parts of learning theory, and chaotic systems. In the triumvirate of FL, NN, and PR, FL is primarily concerned with imprecision, NN with learning, and PR with uncertainty. What is important to note is that although there are substantial areas of overlap between FL, NN, and PR, in general FL, NN, and PR are complementary rather than competitive. For this reason, it is frequently advantageous to employ FL, NN, and PR in combination rather than exclusively. A case in point is the growing number of so-called neurofuzzy (NF) consumer products employing a combination of fuzzy logic and neural network techniques. Most NF products are fuzzy rule-based systems in which NN techniques are used for purposes of learning and/or adaptation.

The Meaning of Fuzzy Logic

When discussing fuzzy logic, there is a semantic issue which requires clarification. The term *fuzzy logic* is currently used in two different senses. In a narrow sense, fuzzy logic is a logical system that aims at a formalization of approximate reasoning. As such, it is rooted in multivalued logic, but its agenda is quite different from that of traditional multivalued logical systems, e.g., Lukasiewicz's logic. In this connection, what should be noted is that many of the concepts which account for the effectiveness of fuzzy logic as a logic of approximate reasoning are not a part of traditional multivalued logical systems. Among these are the concept of a linguistic variable, canonical form, fuzzy if-then rule, fuzzy quantifiers, and such modes of reasoning as interpolative reasoning, syllogistic reasoning, and dispositional reasoning.

In a broad sense, fuzzy logic is almost synonymous with fuzzy set theory. Fuzzy set theory, as its name suggests, is basically a theory of classes with unsharp boundaries. Fuzzy set theory is much broader than fuzzy logic in its narrow sense and contains the latter as one of its branches. Among the other branches of fuzzy set theory are, for example, fuzzy arithmetic, fuzzy mathematical programming, fuzzy topology, fuzzy graph theory, and fuzzy data analysis. What is important to recognize is that any crisp theory can be fuzzified by generalizing the concept of a set within that theory to the concept of a fuzzy set. Indeed, it is very likely that eventually most theories will be fuzzified in this way. The impetus for the transition from a crisp theory to a fuzzy one derives from the fact that both the generality of a theory and its applicability to real-world problems are substantially enhanced by replacing the concept of a set with that of a fuzzy set.

Today, the growing tendency is to use the term *fuzzy logic* in its broad sense. In part this reflects the fact that fuzzy set theory sounds less euphonious than fuzzy logic.

Linguistic Variables, Data Compression, and Granulation

A concept that plays a central role in the applications of fuzzy logic is that of a linguistic variable [29, 31]. The concept of a linguistic variable has become sufficiently well understood to make it unnecessary to dwell upon it here. There is, however, one basic aspect of the concept of a linguistic variable which is worthy of note since it is at the heart of its utility.

Specifically, consider a linguistic variable such as *Age* whose linguistic values are *young, middle-aged,* and *old,* with *young* defined by a membership function such as shown in Figure 1.

Clearly, a numerical value such as 25 is simpler than the function *young*. But *young* represents a choice of one out of three possible values whereas 25 is a choice of one out of, say, 100 values. The point of this simple example is that the use of linguistic values may be viewed as a form of data compression. It is suggestive to refer to this form of data compression as *granulation*.

The same effect can be achieved, of course, by conventional quantization. But in the case of quantization, the values are intervals whereas in the case of granulation the values are overlapping fuzzy sets. The advantages of granulation over quantization are a) it is more general; b) it mimics the way in which humans interpret linguistic values (i.e., as fuzzy sets rather than intervals); and c) the transition from one linguistic value to a contiguous linguistic value is gradual rather than abrupt, resulting in continuity and robustness.

Calculi of Fuzzy Rules and Fuzzy Graphs

The concept of a linguistic variable serves as a point of departure for other concepts in fuzzy logic whose use results in data compression. Among these are the concepts of a fuzzy if-then rule—or simply fuzzy rule—and fuzzy graph. There is a close relation between these concepts, and both may be interpreted as granular representations of functional dependencies and relations. Viewed from this perspective, fuzzy rules and fuzzy graphs bear the same relation to numerically-valued dependencies that linguistic variables bear to numerically-valued variables.

Like the concept of a linguistic variable, the concept of a fuzzy rule is sufficiently well understood to make it unnecessary to dwell upon it here. In what follows, we shall confine our attention to the less well-developed concept of a fuzzy graph.

The concept of a fuzzy graph was initially introduced in 1971 [32] and, in a more explicit form, in 1974 [28, 30]. In an implicit form, the concept of a fuzzy graph underlies the seminal work of Mamdani and Assilian

[14] on fuzzy control. In what follows, we shall assume for notational simplicity that mappings are from R to R.

As shown in Figure 2, a fuzzy graph f^*, of a functional dependence $f: X \rightarrow Y$, where X and Y are linguistic variables in U and V, respectively, serves to provide an approximate, compressed representation of f in the form

$$f^* = A_1 \times B_1 + A_2 \times B_2 + \ldots + A_n \times B_n$$

or more compactly,

$$f^* = \sum_{i=1}^{n} A_i \times B_i$$

where the A_i and B_i, $i = 1, \ldots, n$, are contiguous fuzzy subsets of U and V, respectively; $A_i \times B_i$ is the cartesian product of A_i and B_i; and $+$ is the operation of disjunction, which is usually taken to be the union. Expressed more explicitly in terms of membership functions of f^*, A_i and B_i, we have

$$\mu_{f^*}(u, v) = \vee_i(\mu_{A_i}(u) \wedge \mu_{B_i}(v)),$$

where $\wedge = min$, $\vee = max$, $u \in U$, and $v \in V$. In a more general setting, in place of $\wedge$ and $\vee$ we may employ t-norms and s-norms [35].

Alternatively, a fuzzy graph may be represented as a fuzzy relation f^*

f^*	A	B
	A_1	B_1
	A_2	B_2
	.	.
	.	.
	.	.
	A_n	B_n

or a collection of fuzzy if-then rules

f^* if X is A_1 then Y is B_1
if X is A_2 then Y is B_2
...
if X is A_n then Y is B_n

with the understanding that the fuzzy if-then rule

if X is A_i then Y is B_i, $i = 1, \ldots, n$

is interpreted as the joint constraint on X and Y defined by

(X, Y) is $A_i \times B_i$.

For example, with this understanding the fuzzy rule set

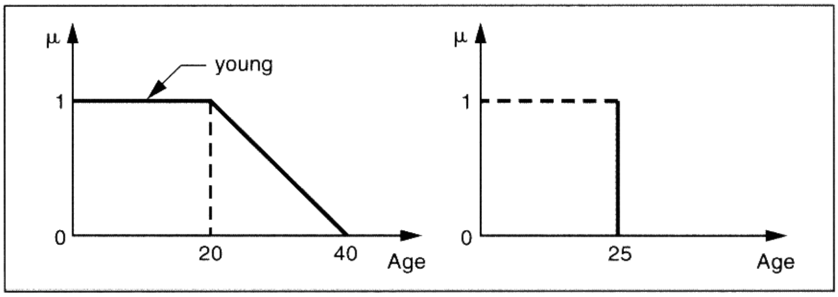


Figure 1. Linguistic and numerical values of *young*

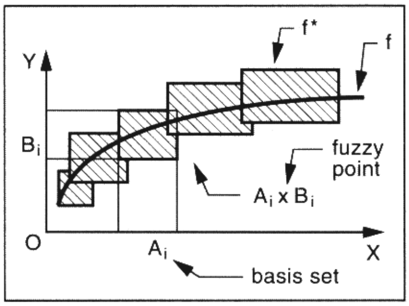


Figure 2. Representation of a function and its fuzzy graph

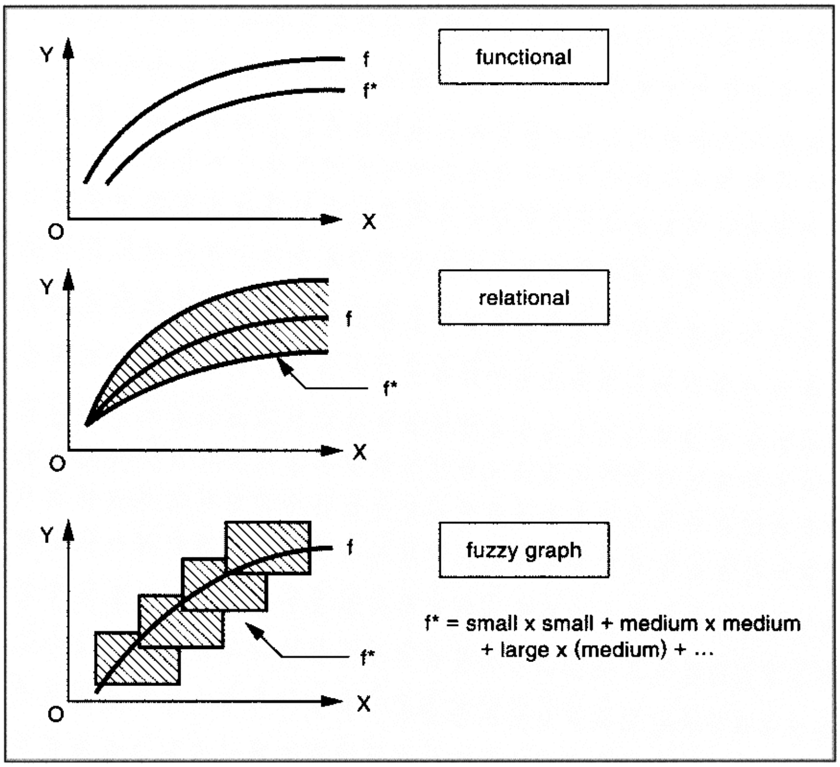


Figure 3. Types of approximation: functional, relational, and fuzzy graph

f^* *if* X is *small* then Y is *large*
if X is *medium* then Y is *medium*
...
if X is *large* then Y is *small*

may be represented equivalently as the fuzzy graph

$$f^* = small \times large + medium \times medium + \ldots + large \times small.$$

In effect, a fuzzy graph approximation to a given function combines a relational approximation with data compression (see Figure 3).

Central to the applications of the concept of a fuzzy graph is the fact that any type of function or relation can be represented by a fuzzy graph (see Figure 4).

Furthermore, fuzzy graph representations may be employed to approximate to probability distributions and membership functions (see Figure 5). Such representations play a particularly important role in qualitative decision analysis and fault diagnosis.

Operations on Fuzzy Graphs

A key issue in the calculus of fuzzy graphs relates to the development of computational methods for performing various basic operations on fuzzy graphs. The operations in question are generalizations of the corresponding operations on crisp (nonfuzzy) functions and relations. Some of the basic operations of this type are shown in Figure 6.

In dealing with this issue, it turns out that the necessary computations can be greatly simplified if an operation, $*$, is monotonically nondecreasing, i.e., if a, b, a', and b' are real numbers, then

$$a' \geq a,\ b' \geq b \rightarrow a' * b' \geq a * b$$
$$a' \leq a,\ b' \leq b \rightarrow a' * b' \leq a * b.$$

For such operations, it can readily be shown that $*$ distributes over $\vee$ (max) and $\wedge$ (min). Thus

$$a * (b \vee c) = a * b \vee a * c$$
$$a * (b \wedge c) = a * b \wedge a * c.$$

This implies that if

$$f^* = \sum_i A_i \times B_i$$

is a fuzzy graph and C is a fuzzy set then

$$C * \left(\sum_i A_i \times B_i\right) = \sum_i C * (A_i \times B_i).$$

As an illustration, consider the problem of finding the intersection of fuzzy graphs f^* and g^* (Figure 7), where

$$f^* = \sum_i A_i \times B_i$$

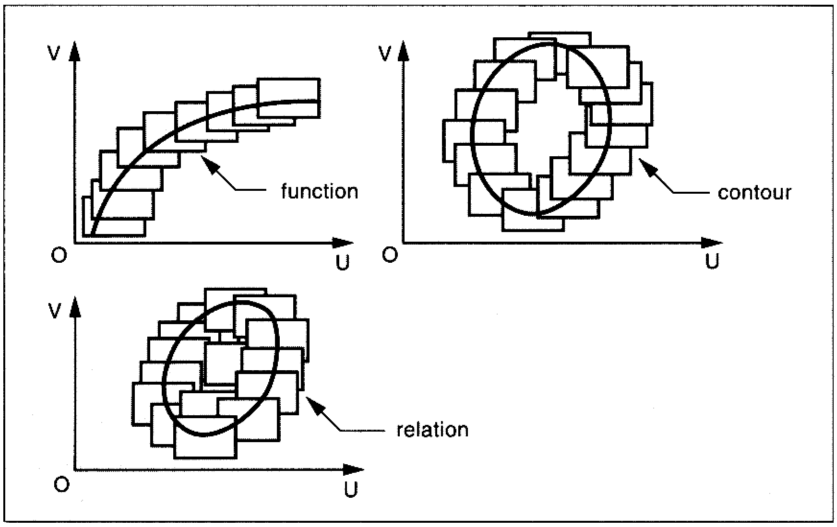


Figure 4. Fuzzy graph approximations to functions, contours, and relations

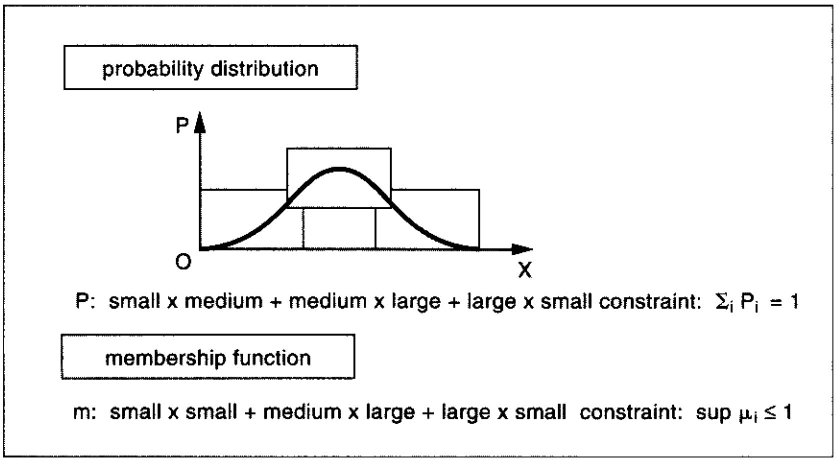


Figure 5. Fuzzy graph approximate representations of probability distributions and membership functions

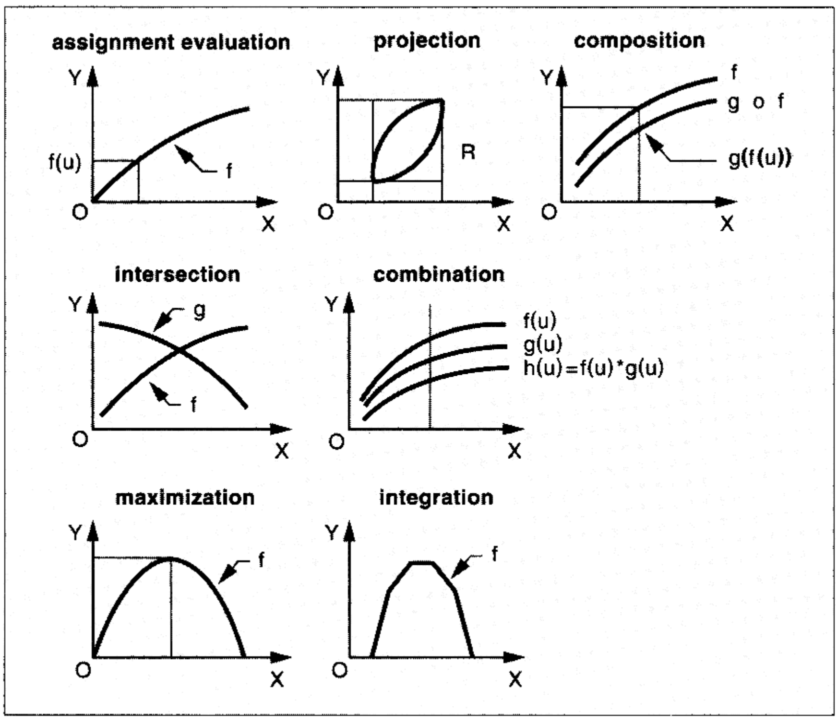


Figure 6. Basic operators on functions and relations

and

$$g^* = \sum_j C_j \times D_j.$$

In this case, we have

$$f^* \cap g^* = \sum_{i,j} (A_i \times B_i) \cap (C_j \times D_j)$$

which in the view of the distributivity of $\cap$ reduces to

$$f^* \cap g^* = \sum_{i,j} (A_i \cap C_j) \times (B_i \cap D_j).$$

This result has an immediate application to the interpolation of fuzzy if-then rules, a problem which plays a key role in fuzzy control.

More specifically, the problem of interpolation may be expressed as an inference query

$$\begin{array}{l} (X, Y) \text{ is } \Sigma_i A_i \times B_i \\ \underline{X \text{ is } A \qquad\qquad} \\ Y \text{ is } ?B \end{array}$$

On representing the major premise in the query as a fuzzy graph $f^* = \Sigma_i A_i \times B_i$ and the minor premise as a cylindrical extension, $\overline{A}$, of the fuzzy set A (see Figure 8), the computation of $?B$ reduces to that of finding the intersection of f^* and $\overline{A}$ and projecting the resulting fuzzy set on V, the domain of Y.

Thus,

$$B = projV(\overline{A} \cap (\sum_i A_i \times B_i)$$

which reduces to

$$B = projV(\sum_i (A \cap A_i) \times B_i)$$

or, more compactly,

$$B = \sum_i m_i \wedge B_i$$

where

$$m_i = sup(A \cap A_i)$$

represents the degree of match between A and A_i (see Figure 9).

It should be noted that in a different guise this technique of interpolation was employed in the seminal paper of Mamdani and Assilian [14] and is currently used in most rule-based control systems.

As a further example, consider the problem of combining f^* and g^* through the minimum operator. Thus, if

$$f^* = \sum_i A_i \times B_i$$

and

$$g^* = \sum_i A_i \times C_i$$

and *min* is the minimum operator, then

$$f^* \text{ min } g^* = \sum_i A_i \times (B_i \text{ min } C_i)$$

where $B_i \text{min} C_i$ is the minimum of B_i and C_i computed through the extension principle [29]. This result makes it possible to compute the intersection, $F \cap G$, of fuzzy sets F and G whose membership functions μ_F and μ_G are represented qualitatively in the form of fuzzy if-then rules (see Figure 10),

F: if X is A_i then μ_F is B_i
$i = 1, \ldots, n$
G: if X is A_i then μ_G is C_i
$i = 1, \ldots, n.$

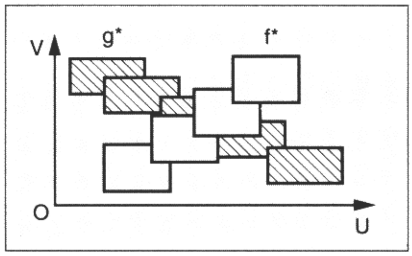
An important practical application of the fuzzy graph representation of the intersection $F \cap G$ relates to the case where F and G represent two conflicting goals and $F \cap G$ a maximizing decision [1].

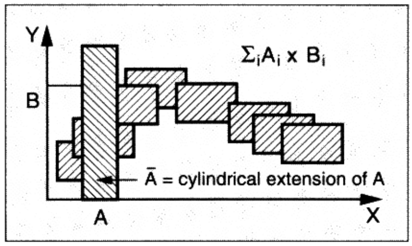
The concept of a fuzzy graph has an important connection with the representation of fuzzy relations. Thus, if R is a fuzzy relation with attributes which take linguistic values (see Figure 11) then R may be represented as a fuzzy graph

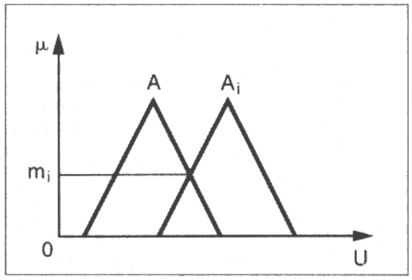
$$R = R_{11} \times R_{12} \times R_{13} + R_{21} \times R_{22} \times R_{23} + R_{31} \times R_{32} \times R_{33}.$$

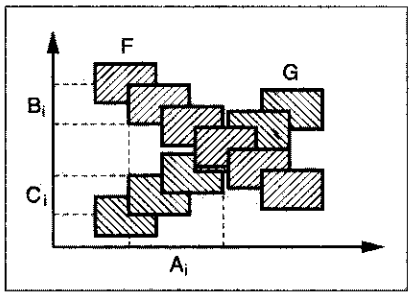
The representation of a fuzzy relation as a fuzzy graph may be applied to the representation of diagnostic tableaus (see Figure 12) in which the entries are linguistic values of tests and corresponding faults.

Thus, if the result of tests T_1, T_2, and T_3 are L, L, and M, respectively, then the degrees to which the faults F_1 and F_2 are present are L and Z,









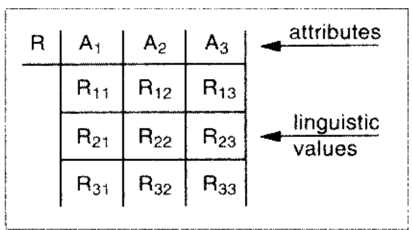


Figure 7. Intersection of fuzzy graphs f^* and g^*

Figure 8. Intersection of f^* and $\overline{A}$

Figure 9. The meaning of the degree of match between A and A_i

Figure 10. Intersection of fuzzy sets F and G whose membership functions are represented as fuzzy graphs

Figure 11. A relation with fuzzy-valued attributes

respectively. If the results of tests are not in the tableau, then the degrees of F_1 and F_2 may be computed through the use of interpolation applied to the fuzzy graph

$$DT = L \times L \times M \times L \times Z + M \times H \times L \times M \times Z + H \times L \times L \times H \times L.$$

Fuzzy Probabilities Expressed as Fuzzy Graphs

An important application area for the calculus of fuzzy graphs relates to computation with fuzzy probabilities in the context of qualitative decision analysis. In more specific terms, assume that X and Y are random variables whose probability distributions on finite sets are described in linguistic terms. For example,

$p(X)$: probability is low if X is small
probability is high if X is medium
probability is low if X is large
$q(Y|X)$: probability is high if X is small and Y is large

. . .

where $q(Y \mid X)$ is the conditional probability distribution of Y given X. The problem is to compute the probability distribution of Y in the form of a fuzzy graph.

Representing $p(X)$ and $q(Y \mid X)$ in the form of fuzzy graphs

$$p(X) = \sum_i A_i \times B_i$$

and

$$q(Y \mid X) = \sum_{i,j} A_i \times C_j \times E_{i,j}$$

where the A_i are fuzzy subsets of u, the B_i are fuzzy subsets of [0, 1], the C_j are fuzzy subsets of V, and the $E_{i,j}$ are fuzzy subsets of [0, 1], the problem is to find the fuzzy graph of the joint probability distribution

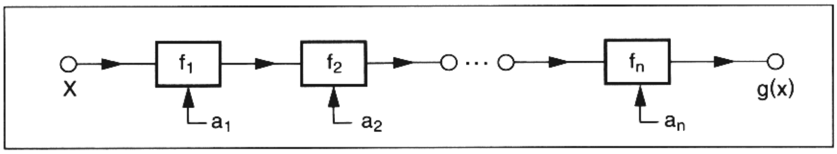
$$h(X, Y) = p(X)q(Y \mid X).$$

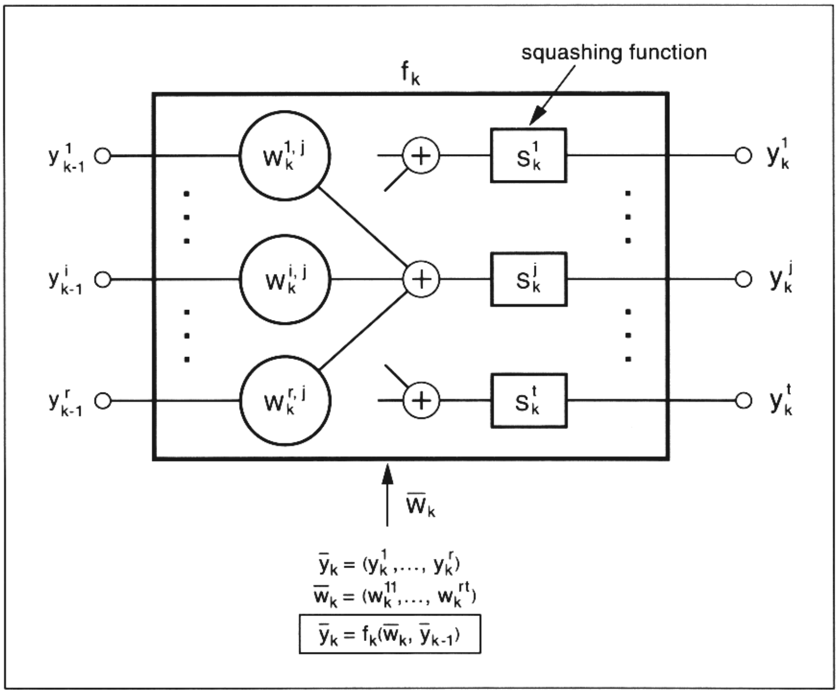
To this end, we use a general rule of combination of fuzzy graphs which may be stated as follows.

Assume that $f(X)$ and $g(Y)$ are functions which may be represented as fuzzy graphs

Figure 12. Diagnostic tableau. L = low; M = medium; H = high; Z = zero

DT	T_1	T_2	T_3	F_1	F_2
	L	L	M	L	Z
	M	H	L	M	Z
	H	L	L	H	L





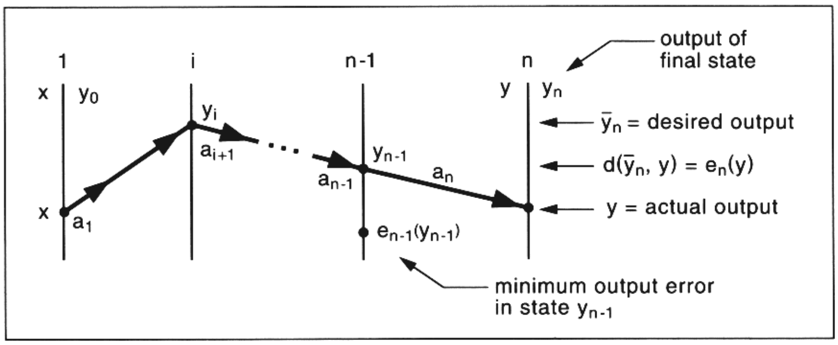


Figure 13. Multilayer feedforward architecture with adjustable parameters

Figure 14. Structure of a layer. $\overline{w}_k$ plays the role of the parameter vector a_k

Figure 15. Graphical representation of dynamic programming

$$f(X) = \sum_i (F_i \times F_i')$$

and

$$g(Y) = \sum_j (G_j \times G_j').$$

Then, if $f(X)$ and $g(Y)$ are combined through a binary operation $*$ (e.g., multiplication or addition), then the resulting function may be expressed as

$$h(X, Y) = f(X) * g(Y),$$

and the fuzzy graph representing $h(X, Y)$ is given by

$$h(X, Y) = \sum_{i,j} (F_i \times G_j) \times (F_i' * G_j').$$

This result provides the basis for computation with imprecisely known probabilities which are expressed as fuzzy graphs or equivalently as collections of fuzzy if-then rules involving linguistic probabilities. In the case of such probabilities, we have

$$p(X) = \Sigma_i(P_i \times P_i')$$
$$q(Y \mid X) = \Sigma_j(Q_j \times Q_j')$$
$$r(X, Y) = p(X)\, q(Y \mid X)$$

and

$$r(X, Y) = \sum_{i,j} (P_i \times Q_j) \times (P_i' * Q_j')$$

where $*$ represents the operation of multiplication of fuzzy probabilities P_i' and Q_j'.

Dynamic and Gradient Programming

A basic problem in the calculi of fuzzy rules and fuzzy graphs is that of the induction of rules from input/output data. For this purpose, it is expedient to employ the techniques of dynamic programming and gradient programming—which have been developed for multistage optimization—to identify parameters in multilayer structures employing feedforward architecture of the form shown in Figure 13.

In this architecture, $a_1, \ldots, a_n$ are vector parameters which play the role of weights in a neural network or parameters of membership functions in a fuzzy rule-based system; x is the input; y is the output, and $f_1, \ldots, f_n$ are functions defining the layers of the feedforward structure. In the case of a neural network, the structure of f_k, $k = 1, \ldots, n$, is shown in Figure 14, with $\overline{w}_k$ playing the role of the parameter vector a_k.

If $\bar{y}_n$ is the target output when the input is $x = x_0$, then $d(\bar{y}_n, y)$ is the distance between the output y and the target output $\bar{y}_n$ (see Figure 15). In effect, $d(\bar{y}_n, y)$ is the error, $e_n(y)$, at the output of the nth layer. Using dynamic programming and letting $e_i(y_i)$ denote the minimum output error achievable when the input to f_{i+1} is y_i and $a_{i+1}, \ldots, a_n$ are optimal, the recurrence equations for $e_i(y_i)$ may be written as

$$e_{n-1}(y_{n-2}) = \min_{a_n}(e_n(f_n(a_n, y_{n-1})))$$
$$e_{n-2}(y_{n-2}) = \min_{a_{n-1}}(e_{n-1}(f_{n-1}(a_{n-1}, y_{n-2})))$$
$$\cdots$$
$$e_1(y_1) = \min_{a_2}(e_2(f_2(a_2, y_1)))$$
$$e_0(y_0) = \min_{a_1}(e_1(f_1(a_1, y_0)))$$
$$y_0 = x_0$$

Thus, setting $y_0 = x_0$, we can successively compute optimal $a_1, \ldots, a_n$ and thereby determine the parameter vectors which minimize the output error. In these computations, x and y can be treated as vectors representing the input and output data sets.

To approximate to the solution of recurrence equations, we can employ gradient programming, which involves the use of chain differentiation. More specifically, in the case of the feedforward structure shown in Figure 13, we can write

$$\frac{\delta e_n(y_n)}{\delta a_n} = \frac{\delta e_n(y_n)}{\delta y_n} \cdot \frac{\delta y_n}{\delta a_n}$$
$$\frac{\delta e_n(y_n)}{\delta a_{n-i}} = \frac{\delta e_n(y_n)}{\delta y_{n-i}} \cdot \frac{\delta y_{n-i}}{\delta a_{n-i}}$$
$$\frac{\delta e_n(y_n)}{\delta y_{n-i}} = \frac{\delta y_n}{\delta y_{n-i+1}} \cdot \frac{\delta y_{n-i+1}}{\delta y_{n-i}}.$$

It is of interest to note that, on applying these equations to the structure shown in Figure 15, we arrive at the familiar recurrence equations associated with the backpropagation algorithm [3].

Alternatively, applying the equations in question to multilayer structures representing fuzzy rule-based systems, one obtains recursive algorithms for the computation of parameters of membership functions. More detailed expositions of methods of this type may be found in important contributions of Takagi, Sugeno, and Kang [19, 21], Jang [4, 5], Wang [24], Lee and Lin [13], among others.

Concluding Remarks

In this brief article we have attempted to summarize some of the basic ideas underlying soft computing and its relation to fuzzy logic, neural network theory, and probabilistic reasoning. The principal aim of soft computing is to achieve tractability, robustness, low solution cost, and high MIQ through the exploitation of the tolerance for imprecision and uncertainty. Insofar as fuzzy logic is concerned, its principal contribution to this aim centers on the concept of granulation, the concept of a linguistic variable, and the calculi of fuzzy rules and fuzzy graphs. Through these concepts and methods, fuzzy logic provides a model for modes of reasoning which are approximate rather than exact. The role model for fuzzy logic is the human mind. ■

References

1. Bellman, R.E and Zadeh, L.A. Decision-making in a fuzzy environment. *Manage. Sci. 17,* (1970), B-141–B-164.
2. Berenji, H.R. Fuzzy logic controllers. In *An Introduction to Fuzzy Logic Applications in Intelligent Systems.* Kluwer Academic Publishers, Boston, 1991, 69–96.
3. Hertz, J., Krogh, A. and Palmer, R. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Reading, Mass., 1991.
4. Jang, J.-S.R. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. Syst. Man Cybernet. 23,* 3 (May 1992).
5. Jang, J.-S.R. Self-learning fuzzy controller based on temporal back-propagation. *IEEE Trans. Neural Netw. 3,* 5 (Sept. 1992), 714–723.
6. Karr, C. Genetic algorithms for fuzzy controllers. *AI Exp. 6,* (1991), 26–33.
7. Kaufmann, A. and Gupta, M.M. *Fuzzy Mathematical Models in Engineering and Management Science.* North Holland, Amsterdam, 1988.
8. Kaufmann, A. and Gupta, M.M. *Introduction to Fuzzy Arithmetic.* Van Nostrand, New York, 1985.
9. Kosko, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence.* Prentice-Hall, Englewood Cliffs, N.J., 1991.
10. Langari, R. and Berenji, H.R. Fuzzy logic in control engineering. In *Handbook of Intelligent Control.* Van Nostrand, New York, 1992.

11. Lee, C.C. Fuzzy logic in control systems: Fuzzy logic controller. Part I and Part II. *IEEE Trans. Syst. Man Cybernet. 20,* (1990).
12. Lee, M.A. and Takagi, H. Integrating design stages of fuzzy systems using genetic algorithms. In *Proceedings of the Second International Conference on Fuzzy Systems (FUZZ-IEEE '93)* (Mar. 28-Apr. 1, 1993). IEEE, New York, 1993, pp. 612–617.
13. Lin, C.-T. and Lee, C.S.G. Neural-network-based fuzzy logic control and decision system. *IEEE Trans. Comput. 40,* 12 (Dec. 1991), 1320–1336.
14. Mamdani, E.H. and Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Stud. 7,* (1975).
15. Mamdani, E.H. and Gaines, B.R., Eds. *Fuzzy Reasoning and Its Applications.* Academic Press, London, 1981.
16. Negoita, C. *Expert Systems and Fuzzy Systems.* Benjamin Cummings, Menlo Park, Calif., 1985.
17. Pedrycz, W. *Fuzzy Control and Fuzzy Systems.* John Wiley, New York, 1989.
18. Sugeno, M. *Industrial Applications of Fuzzy Control.* Elsevier Science Publishers B.V., Amsterdam, 1985.
19. Sugeno, M. and Kang, G.T. Structure identification of fuzzy model. *Fuzzy Sets Syst. 28,* (1988), 15–33.
20. Takagi, H. and Hayashi, I. NN-driven fuzzy reasoning. *Int. J. Approx. Reason.* (1991), 191–212.
21. Takagi, T. and Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybernet.* (1985), 116–132.
22. Togai, M. and Watanabe, H. An inference engine for real-time approximate reasoning: Toward an expert system on a chip. *IEEE Exp. 1,* (1986), 55–62.
23. Turksen, I.B. Approximate reasoning for production planning. *Fuzzy Sets Syst. 26,* (1988), 23–37.
24. Wang, L.-X. Stable adaptive fuzzy control of nonlinear systems. *IEEE Trans. Fuzzy Syst. 1,* 1 (Feb. 1993).
25. Yager, R.R. and Zadeh, L.A., Eds. *An Introduction to Fuzzy Logic Applications in Intelligent Systems.* Kluwer Academic Publishers, Boston, 1991.
26. Yasunobu, S. and Myamoto, S. Automatic train operation by predictive fuzzy control. In *Industrial Applications of Fuzzy Control.* North Holland, Amsterdam, 1985.
27. Zadeh, L.A. The calculus of fuzzy if-then rules. *AI Exp. 7,* 3 (Mar. 1992), 22–27.
28. Zadeh, L.A. A fuzzy-algorithmic approach to the definition of complex or imprecise concepts. Electronics Res. Lab. Rep. ERL-M474, Univ. of California, Berkeley. 1974. Also in *Int. J. Man-Machine Stud. 8,* (1976), 249–291.
29. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning—I. *Inf. Sci. 8,* (1975), 199–249.
30. Zadeh, L.A. On the analysis of large scale systems. In *Systems Approaches and Environment Problems.* Vandenhoeck and Ruprecht, Gottingen, Germany, 1974, 23–37.
31. Zadeh, L.A. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybernet. SMC-3,* (1973), 28–44.
32. Zadeh, L.A. Toward a theory of fuzzy systems. In *Aspects of Network and System Theory.* Rinehart and Winston, New York, 1971, 469–490.
33. Zadeh, L.A. Thinking machines—a new field in electrical engineering. *Columbia Eng. 3,* (1950), 12–13, 30, 31.
34. Zadeh, L.A. and Yager, R.R., Eds. *Uncertainty in Knowledge Bases.* Springer-Verlag, Berlin, 1991.
35. Zimmerman, H.J. *Fuzzy Set Theory and Its Applications.* 2d ed. Kluwer-Nijhoff, 1990.

About the Author:
LOTFI A. ZADEH is Professor Emeritus and Director of the Berkeley Initiative in Soft Computing (BISC) in the Computer Science Division at the University of California at Berkeley. Current research interests include fuzzy logic, soft computing, and intelligent systems. **Author's Present Address:** Computer Science Division, University of California at Berkeley, Berkeley, CA 94720; email: zadeh@cs.berkeley.edu

Research supported by NASA Grant NCC 2-275, EPRI Agreement RP 8010-34, MICRO State Program No. 90-191 and the BISC (Berkeley Initiative in Soft Computing) program.