8

FUZZY ASSOCIATIVE MEMORIES

FUZZY SYSTEMS AS BETWEEN-CUBE MAPPINGS

Chapter 7 introduced multivalued or fuzzy sets as points in the unit hypercube $I^n = [0, 1]^n$. Within the cube we were interested in the distance between points. This led to measures of the size and fuzziness of a fuzzy set and, more fundamentally, to a measure of how much one fuzzy set is a subset of another fuzzy set. This *within-cube* theory directly extends to the continuous case where the space X is a subset of R^n or, in general, where X is a subset of products of real or complex spaces.

The next step considers mappings *between* fuzzy cubes. This level of abstraction provides a surprising and fruitful alternative to the propositional and predicate-calculus reasoning techniques used in artificial-intelligence (AI) expert systems. It allows us to reason with sets instead of propositions.

The fuzzy-set framework is numerical and multidimensional. The AI framework is symbolic and one-dimensional, with usually only bivalent expert "rules" or propositions allowed. Both frameworks can encode structured knowledge in linguistic form. But the fuzzy approach translates the structured knowledge into a flexible *numerical* framework and processes it in a manner that resembles neural-network processing. The numerical framework also allows us to adaptively infer and modify fuzzy systems, perhaps with neural or statistical techniques, directly from problem-domain sample data.

Between-cube theory is fuzzy-systems theory. A fuzzy set defines a point in a cube. A fuzzy system defines a mapping between cubes. A fuzzy system S maps fuzzy sets to fuzzy sets. Thus a **fuzzy system** S is a transformation $S\colon I^n \to I^p$. The n-dimensional unit hypercube I^n houses all the fuzzy subsets of the domain space, or input *universe of discourse*, $X = \{x_1, \ldots, x_n\}$. I^p houses all the fuzzy subsets of the range space, or output universe of discourse, $Y = \{y_1, \ldots, y_p\}$. X and Y can also denote subsets of R^n and R^p. Then the fuzzy power sets $F(2^X)$ and $F(2^Y)$ replace I^n and I^p.

In general a fuzzy system S maps families of fuzzy sets to families of fuzzy sets, thus $S\colon I^{n_1} \times \cdots \times I^{n_r} \to I^{p_1} \times \cdots \times I^{p_s}$. Here too we can extend the definition of a fuzzy system to allow arbitrary products of arbitrary mathematical spaces to serve as the domain or range spaces of the fuzzy sets.

(A technical comment is in order for sake of historical clarification. A tenet, perhaps the defining tenet, of the classical theory [Dubois, 1980] of fuzzy sets as functions concerns the fuzzy extension of any mathematical function. This tenet holds that any function $f\colon X \to Y$ that maps points in X to points in Y extends to map the fuzzy subsets of X to the fuzzy subsets of Y. The so-called *extension principle* defines the set-function $f\colon F(2^X) \to F(2^Y)$, where $F(2^X)$ denotes the fuzzy power set of X, the set of all fuzzy subsets of X. The formal definition of the extension principle is complicated. The key idea is a supremum of pairwise minima. Unfortunately, the extension principle achieves generality at the price of triviality. In general [Kosko, 1986a, 1987] the extension principle extends functions to fuzzy sets by stripping the fuzzy sets of their fuzziness, mapping the fuzzy sets into bit vectors of nearly all 1s. This shortcoming, combined with the tendency of the extension-principle framework to push fuzzy theory into largely inaccessible regions of abstract mathematics, led in part to the development of the alternative sets-as-points geometric framework of fuzzy theory.)

We shall focus on fuzzy systems $S\colon I^n \to I^p$ that map *balls* of fuzzy sets in I^n to balls of fuzzy sets in I^p. These continuous fuzzy systems behave as associative memories. They map close inputs to close outputs. We shall refer to them as **fuzzy associative memories**, or FAMs.

The simplest FAM encodes the **FAM rule** or association (A_i, B_i), which associates the p-dimensional fuzzy set B_i with the n-dimensional fuzzy set A_i. These minimal FAMs essentially map one ball in I^n to one ball in I^p. They are comparable to simple neural networks. But we need not adaptively train the minimal FAMs. As discussed below, we can directly encode structured knowledge of the form "If traffic is heavy in this direction, then keep the stop light green longer" in a Hebbian-style FAM correlation matrix. In practice we sidestep this large numerical matrix with a virtual representation scheme. In place of the matrix the user encodes the fuzzy-set association (HEAVY, LONGER) as a single linguistic entry in a FAM-bank linguistic matrix.

In general a FAM system $F: I^n \rightarrow I^p$ encodes and processes in parallel a FAM bank of m FAM rules $(A_1, B_1), \ldots, (A_m, B_m)$. Each input A to the FAM system activates each stored FAM rule to different degree. The minimal FAM that stores (A_i, B_i) maps input A to B'_i, a partially activated version of B_i. The more A resembles A_i, the more B'_i resembles B_i. The corresponding output fuzzy set B combines these partially activated fuzzy sets $B'_1, \ldots, B'_m$. B equals a weighted average of the partially activated sets:

$$B = w_1 B'_1 + \cdots + w_m B'_m$$

where w_i reflects the credibility, frequency, or strength of the fuzzy association (A_i, B_i). In practice we usually "defuzzify" the output waveform B to a single numerical value y_j in Y by computing the fuzzy centroid of B with respect to the output universe of discourse Y.

More general still, a FAM system encodes a bank of compound FAM rules that associate multiple output or consequent fuzzy sets $B_i^1, \ldots, B_i^s$ with multiple input or antecedent fuzzy sets $A_i^1, \ldots, A_i^r$. We can treat compound FAM rules as compound linguistic conditionals. This allows us to naturally, and in many cases easily, obtain structural knowledge. We combine antecedent and consequent sets with logical conjunction, disjunction, or negation. For instance, we would interpret the compound association $(A^1, A^2; B)$ linguistically as the compound conditional "IF X^1 is A^1 AND X^2 is A^2, THEN Y is B" if the comma in the fuzzy association $(A^1, A^2; B)$ denotes conjunction instead of, say, disjunction.

We specify in advance the numerical universes of discourse for fuzzy variables X^1, X^2, and Y. For each universe of discourse or fuzzy variable X, we specify an appropriate *library* of fuzzy-set values, $A_1^r, \ldots, A_k^r$. Contiguous fuzzy sets in a library overlap. In principle a neural network can estimate these libraries of fuzzy sets. In practice this is usually unnecessary. The library sets represent a weighted, though overlapping, quantization of the input space X. They represent the fuzzy-set *values* assumed by a fuzzy *variable*. A different library of fuzzy sets similarly quantizes the output space Y. Once we define the library of fuzzy sets, we construct the FAM by choosing appropriate combinations of input and output fuzzy sets. Adaptive techniques can make, assist, or modify these choices.

An adaptive FAM (AFAM) is a *time-varying* FAM system. System parameters gradually change as the FAM system samples and processes data. Below we discuss how neural network algorithms can adaptively infer FAM rules from training data. In principle, learning can modify other FAM system components, such as the libraries of fuzzy sets or the FAM-rule weights w_i.

Below we propose and illustrate an unsupervised adaptive clustering scheme, based on competitive learning, to "blindly" generate and refine the bank of FAM rules. In some cases we can use supervised learning techniques if we have additional information to accurately generate error estimates.

FUZZY AND NEURAL FUNCTION ESTIMATORS

Neural and fuzzy systems estimate sampled functions and behave as associative memories. They share a key advantage over traditional statistical-estimation and adaptive-control approaches to function estimation. They are *model-free* estimators. Neural and fuzzy systems estimate a function without requiring a mathematical description of how the output functionally depends on the input. They "learn from example." More precisely, they learn from samples.

Both approaches are numerical, can be partially described with theorems, and admit an algorithmic characterization that favors silicon and optical implementation. These properties distinguish neural and fuzzy approaches from the symbolic processing approaches of artificial intelligence.

Neural and fuzzy systems differ in how they estimate sampled functions. They differ in the kind of samples used, how they represent and store those samples, and how they associatively "inference" or map inputs to outputs.

These differences appear during system construction. The neural approach requires the specification of a nonlinear dynamical system, usually feedforward, the acquisition of a sufficiently representative set of numerical training samples, and the encoding of those training samples in the dynamical system by repeated learning cycles. The fuzzy system requires only that we partially fill in a linguistic "rule matrix." This task is markedly simpler than designing and training a neural network. Once we construct the systems, we can present the same numerical inputs to either system. The outputs will reside in the same numerical space of alternatives. So both systems define a surface or manifold in the input-output product space $X \times Y$. We present examples of these surfaces in Chapters 9, 10, and 11.

Which system, neural or fuzzy, is more appropriate for a particular problem depends on the nature of the problem and the availability of numerical and structured data. To date engineers have applied fuzzy techniques largely to control problems. These problems often permit comparison with standard control-theoretic and expert-system approaches. Neural networks so far seem best applied to ill-defined two-class pattern-recognition problems (defective or nondefective, bomb or not, etc.).

Fuzzy systems estimate functions with *fuzzy-set* samples (A_i, B_i). Neural systems use *numerical-point* samples (x_i, y_i). Both kinds of samples reside in the input-output product space $X \times Y$.

Figure 8.1 illustrates the geometry of fuzzy-set and numerical-point samples taken from the function $f: X \rightarrow Y$.

Engineers sometimes call the fuzzy-set association (A_i, B_i) a "rule." This is misleading, since reasoning with sets is not the same as reasoning with propositions. Reasoning with sets is harder. Sets are multidimensional, and matrices, not propositional conditionals, house associations. We must take care how we define each term and operation. We shall refer to the antecedent term A_i in the fuzzy association (A_i, B_i) as the **input associant** and the consequent term B_i as the **output associant**.

The fuzzy-set sample (A_i, B_i) encodes *structure*. It represents a mapping, a minimal *fuzzy association* of part of the output space with part of the input space.

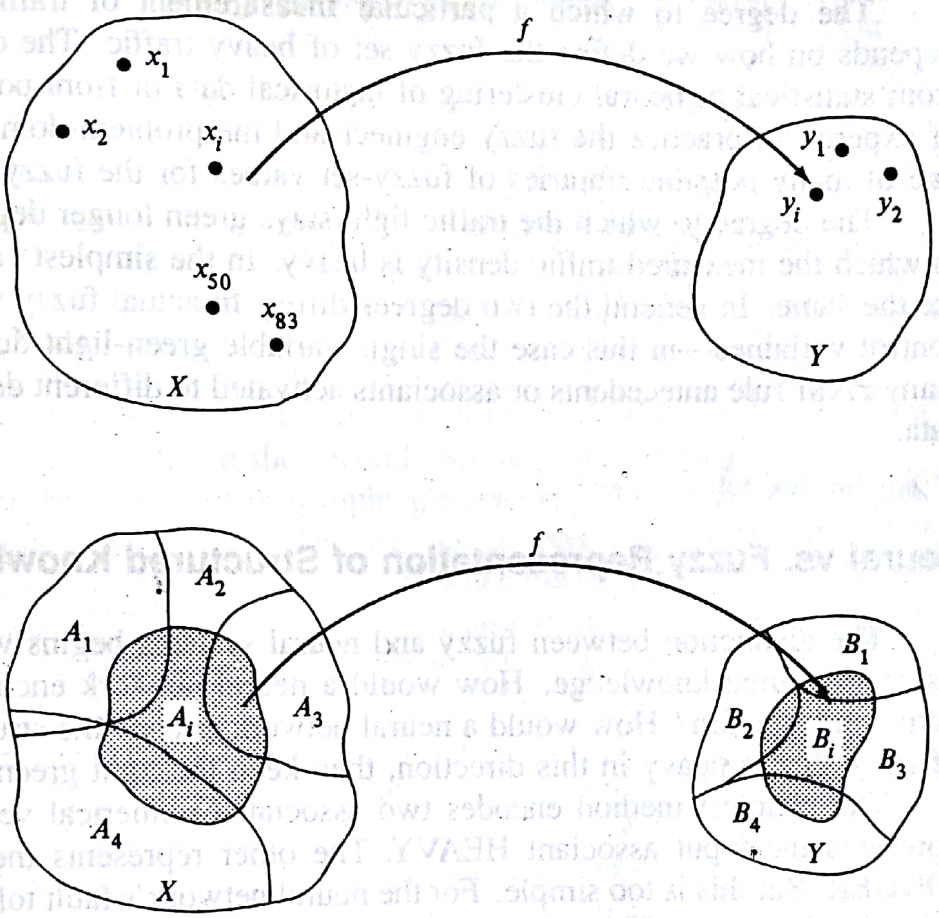

FIGURE 8.1 Function f maps domain X to range Y. In the first illustration we use several numerical-point samples (x_i, y_i) to estimate f: $X \longrightarrow Y$. In the second case we use only a few fuzzy subsets A_i of X and B_i of Y. The fuzzy association (A_i, B_i) represents system structure, as an adaptive clustering algorithm might infer or as an expert might articulate. In practice there are usually fewer different output associants or "rule" consequents B_i than input associants or antecedents A_i.

In practice this resembles a meta-rule—IF A_i, THEN B_i—the type of structured linguistic rule an expert might articulate to build an expert-system "knowledge base." The association might also represent the result of an adaptive clustering algorithm.

Consider a fuzzy association for the intelligent control of a traffic light: "If the traffic is heavy in this direction, then keep the light green longer." The fuzzy association is (HEAVY, LONGER). The input fuzzy variable *traffic density* assumes the fuzzy-set value HEAVY. The output fuzzy variable *green light duration* assumes the fuzzy-set value LONGER. Another fuzzy association might be (LIGHT, SHORTER). The fuzzy system encodes each linguistic association or "rule" in a numerical *fuzzy associative memory* (FAM) mapping. The FAM then numerically processes numerical input data. A measured description of traffic density (e.g., 150 cars per unit road surface area) then corresponds to a unique numerical output (e.g., 3 seconds), the "recalled" output.

The degree to which a particular measurement of traffic density is heavy depends on how we define the fuzzy set of heavy traffic. The definition may arise from statistical or neural clustering of historical data or from pooling the responses of experts. In practice the fuzzy engineer and the problem-domain expert agree on one of many possible libraries of fuzzy-set values for the fuzzy variables.

The degree to which the traffic light stays green longer depends on the degree to which the measured traffic density is heavy. In the simplest case the two degrees are the same. In general the two degrees differ. In actual fuzzy systems the output-control variables—in this case the single variable green-light duration—depend on many FAM-rule antecedents or associants activated to different degrees by incoming data.

Neural vs. Fuzzy Representation of Structured Knowledge

The distinction between fuzzy and neural systems begins with how they represent structured knowledge. How would a neural network encode the same associative information? How would a neural network encode the structured knowledge "If the traffic is heavy in this direction, then keep the light green longer"?

The simplest method encodes two associated numerical vectors. One vector represents the input associant HEAVY. The other represents the output associant LONGER. But this is too simple. For the neural network's fault tolerance now works to its disadvantage. The network tends to reconstruct partial inputs to complete sample inputs. It erases the desired partial degrees of activation. If an input is close to A_i, the output will tend to be B_i. If the output is distant from A_i, the output will tend to be some other sampled output vector or a spurious output altogether.

A better neural approach encodes a mapping from the heavy-traffic subspace to the longer-time subspace. Then the neural network needs a representative sample set to capture this structure. Statistical networks, such as adaptive vector quantizers, may need thousands of statistically representative samples. Feedforward multilayer neural networks trained with the backpropagation algorithm in Chapter 5 may need hundreds of representative numerical input-output pairs and may need to recycle these samples tens of thousands of times in the learning process.

The neural approach suffers a deeper problem than just the computational burden of training. *What* does it encode? How do we know the network encodes the original structure? What does it recall? There is no natural inferential audit trail. System nonlinearities wash it away. Unlike an expert system, we do not know which inferential paths the network uses to reach a given output or even which inferential paths exist. There is only a large system of synchronous or asynchronous nonlinear functions. Unlike, say, the adaptive Kalman filter, we cannot appeal to a postulated mathematical model of how the output state depends on the input state. Model-free estimation is, after all, the central computational advantage of neural networks. The cost is system inscrutability.

We are left with an unstructured computational black box. We do not know what the neural network encoded during training or what it will encode or forget in further training. (For competitive adaptive vector quantizers we do know that synaptic vectors asymptotically estimate sample-space centroids and perhaps higher-order moments.) We can characterize the neural network's behavior only by exhaustively passing all inputs through the black box and recording the recalled outputs. The characterization may use a summary scalar like mean-squared error.

This black-box characterization of the network's behavior involves a computational *dilemma*. On the one hand, for most problems the number of input-output cases we need to check is computationally prohibitive. On the other, when the number of input-output cases is tractable, we may as well store these pairs and appeal to them directly, and without error, as a look-up table. In the first case the neural network is unreliable. In the second case it is unnecessary.

A further problem is sample generation. Where did the original numerical point samples come from? Did we ask an expert to give numbers? How reliable are such numerical vectors, especially when the expert feels most comfortable giving the original linguistic data? This procedure seems at most as reliable as the expert-system method of asking an expert to give condition-action rules with numerical uncertainty weights.

Statistical neural estimators require a "statistically representative" sample set. We may need to randomly "create" these samples from an initial small sample set by bootstrap techniques or by random-number generation of points clustered near the original samples. Both sample-augmentation procedures assume that the initial sample set sufficiently represents the underlying probability distribution. The problem of where the original sample set comes from remains. The fuzziness of the notion "statistically representative" compounds the problem. In general we do not know in advance how well a given sample set reflects an unknown underlying distribution of points. Indeed when the network adapts on-line, we know only past samples. The remainder of the sample set resides in the unsampled future.

In contrast, fuzzy systems directly encode the linguistic sample (HEAVY, LONGER) in a dedicated numerical matrix, perhaps of infinite dimensions. The default encoding technique is the fuzzy Hebb procedure discussed below. For practical problems we need not store this large, perhaps infinite, numerical matrix. Instead we use a virtual representation scheme. Numerical point inputs permit this simplification. Mathematically we implicitly pass large unit bit vectors, or delta pulses in the continuous case, through the FAM-rule matrix. In general we describe inputs by an uncertainty distribution, probabilistic or fuzzy. Then we must use the entire matrix or reduce the input to a scalar by averaging.

For instance, if the *heavy traffic* input is 150 cars, we can omit the FAM matrix. Below we refer to these systems as binary input-output FAMs, or BIOFAMs. But if the input is a Gaussian curve with mean 150, then in principle we must process the vector input with a FAM matrix. (In practice we might use only the mean.) The dimensions of the *linguistic* FAM-bank matrix are usually small. The dimensions reflect the quantization levels of the input and output spaces, the number of fuzzy-set values assumed by the fuzzy variables.

The fuzzy approach combines the purely numerical approaches of neural networks and mathematical modeling with the symbolic, structure-rich approaches of artificial intelligence. We acquire knowledge symbolically—or numerically if we use adaptive techniques—but represent it numerically. We also process data numerically. Adaptive FAM rules correspond to common-sense, often nonarticulated, behavioral rules that improve with experience.

This approach does not abandon neural-network techniques. Instead, it limits them to *unstructured* parameter and state estimation, pattern recognition, and cluster formation. The system *architecture* remains fuzzy.

FAMs as Mappings

Fuzzy associative memories (FAMs) are transformations. *FAMs map fuzzy sets to fuzzy sets.* They map unit cubes to unit cubes, as in Figure 8.1. In the simplest case the FAM system consists of a single association, such as (HEAVY, LONGER). In general the FAM system consists of a bank of different FAM associations. Each association corresponds to a different numerical FAM matrix, or a different entry in a linguistic FAM-bank matrix. We do not combine these matrices as we combine or superimpose neural-network associative-memory (outer-product) matrices. (An exception is the *fuzzy cognitive map* [Kosko, 1988; Taber, 1987, 1991].) We store the matrices separately and access them in parallel. This avoids crosstalk. Since we use a virtual (BIOFAM) representation scheme, the computational burden of the parallel access is light.

We begin with single-association FAMs. For concreteness let the fuzzy-set pair (A, B) encode the traffic-control association (HEAVY, LIGHT). We quantize the domain of traffic density to the n numerical variables $x_1, x_2, \ldots, x_n$. We quantize the range of green-light duration to the p variables $y_1, y_2, \ldots, y_p$. The elements x_i and y_j belong respectively to the ground sets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_p\}$. x_1 might represent zero traffic density. y_p might represent 10 seconds.

The fuzzy sets A and B are multivalued or fuzzy subsets of X and Y. So A defines a point in the n-dimensional unit hypercube $I^n = [0, 1]^n$, and B defines a point in the p-dimensional fuzzy cube I^p. Equivalently, A and B define the membership functions m_A and m_B that map the elements x_i of X and y_j of Y to degrees of membership in [0, 1]. The membership values, or *fit* (fuzzy unit) values, indicate how much x_i belongs to or fits in subset A, and how much y_j belongs to B. We describe this with the abstract functions $m_A\colon X \longrightarrow [0, 1]$ and $m_B\colon Y \longrightarrow [0, 1]$. We shall freely view sets both as functions and as points in fuzzy power sets.

The geometric *sets-as-points* interpretation of finite fuzzy sets A and B as points in unit cubes allows a natural vector representation. We represent A and B by the numerical *fit vectors* $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_p)$, where $a_i = m_A(x_i)$, and $b_j = m_B(y_j)$. We can interpret the identifications A = HEAVY and B = LONGER to suit the problem at hand. Intuitively the a_i values should increase

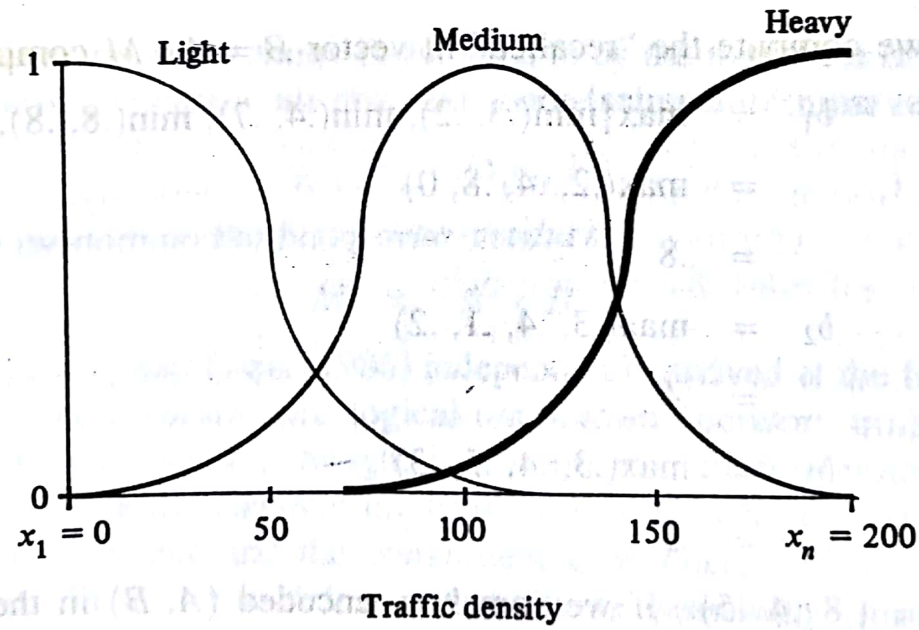

FIGURE 8.2 Three possible fuzzy subsets of traffic-density space X. Each fuzzy sample corresponds to such a subset. We draw the fuzzy sets as continuous membership functions. In practice membership values are sampled or quantized. So the sets are points in some unit hypercube I^n.

as the index i increases, perhaps approximating a sigmoid membership function. Figure 8.2 illustrates three possible fuzzy subsets of the universe of discourse X.

FUZZY HEBB FAMS

Most fuzzy systems found in applications are fuzzy Hebb FAMs [Kosko, 1986b]. They are fuzzy systems S: $I^n \to I^p$ constructed in a simple neurallike manner. As discussed in Chapter 4, in neural-network theory we interpret the classical Hebbian hypothesis of correlation synaptic learning [Hebb, 1949] as unsupervised learning with the signal product $S_i S_j$:

$$\dot{m}_{ij} = -m_{ij} + S_i(x_i)S_j(y_j) \tag{8-3}$$

For a given pair of bipolar row vectors (X, Y), the neural interpretation gives the *outer-product* correlation matrix

$$M = X^T Y \tag{8-4}$$

We define pointwise the **fuzzy Hebb matrix** by the minimum of the "signals" a_i and b_j, an encoding scheme we shall call **correlation-minimum encoding:**

$$m_{ij} \quad = \quad \min(a_i,\, b_j) \tag{8-5}$$

given in matrix notation as the *fuzzy outer-product*

$$M \quad = \quad A^T \circ B \tag{8-6}$$

Mamdani [1977] and Togai [1986] independently arrived at the fuzzy Hebbian prescription (8-5) as a multivalued logical-implication operator: truth $(a_i \rightarrow b_i) = \min(a_i, b_j)$. The min operator, though, is a symmetric truth operator. So it does not properly generalize the classical implication $P \rightarrow Q$, which is false if and only if the antecedent P is true and the consequent Q is false, $t(P) = 1$ and $t(Q) = 0$. In contrast, a like desire to define a "conditional-possibility" matrix pointwise with continuous implication values led Zadeh [1983] to choose the Lukasiewicz implication operator: $m_{ij} = \text{truth } (a_i \rightarrow b_j) = \min(1,\, 1 - a_i + b_j)$. Unfortunately the Lukasiewicz operator usually equals or approximates unity, for $\min(1,\, 1-a_i+b_j) < 1$ iff $a_i > b_j$. Most entries of the resulting matrix M are unity or near unity. This ignores the information in the association (A, B). So $A' \circ M$ tends to equal the largest fit value a'_k for any system input A'.

We construct an *autoassociative* fuzzy Hebb FAM matrix by encoding the redundant pair (A, A) in (8-6) as the fuzzy autocorrelation matrix:

$$M = A^T \circ A \tag{8-7}$$

In the previous example the matrix M was such that the input $A = (.3\ \ .4\ \ .8\ \ 1)$ recalled fit vector $B = (.8\ \ .4\ \ .5)$ upon max-min composition: $A \circ M = B$. Will A still recall B if we replace the original matrix M with the fuzzy Hebb matrix found with (8-6)? Substituting A and B in (8-6) gives

$$M \;=\; A^T \circ B \;=\; \begin{pmatrix} .3 \\ .4 \\ .8 \\ 1 \end{pmatrix} \circ (.8\ \ .4\ \ .5) \;=\; \begin{pmatrix} .3 & .3 & .3 \\ .4 & .4 & .4 \\ .8 & .4 & .5 \\ .8 & .4 & .5 \end{pmatrix}$$

This fuzzy Hebb matrix M illustrates two key properties. First, the ith row of M equals the pairwise minimum of a_i and the output associant B. Symmetrically, the jth column of M equals the pairwise minimum of b_j and the input associant A:

$$M \quad = \quad \begin{bmatrix} a_1 \wedge B \\ \hline \vdots \\ \hline a_n \wedge B \end{bmatrix} \tag{8-8}$$

$$= \quad [b_1 \wedge A^T | \ldots | b_m \wedge A^T] \tag{8-9}$$

where the cap operator denotes pairwise minimum: $a_i \wedge b_j = \min(a_i,\, b_j)$. The term $a_i \wedge B$ indicates componentwise minimum:

$$a_i \wedge B \quad = \quad (a_i \wedge b_1,\, \ldots,\, a_i \wedge b_n) \tag{8-10}$$

Hence if some $a_k = 1$, then the kth row of M equals B. If some $b_l = 1$, the lth column of M equals A. More generally, if some a_k is at least as large as every b_j, then the kth row of the fuzzy Hebb matrix M equals B.

Second, the third and fourth rows of M equal the fit vector B. Yet no column equals A. This allows perfect recall in the forward direction, $A \circ M = B$, but not in the backward direction, $B \circ M^T \neq A$:

$$A \circ M = (.8 \ .4 \ .5) = B$$
$$B \circ M^T = (.3 \ .4 \ .8 \ .8) = A' \subset A$$

A' is a proper subset of A: $A' \neq A$ and $S(A', A) = 1$, where S measures the degree of subsethood of A' in A, as discussed in Chapter 7. In other words, $a'_i \leq a_i$ for each i and $a'_k < a_k$ for at least one k. The bidirectional FAM theorems below show that this holds in general: If $B' = A \circ M$ differs from B, then B' is a proper subset of B. Hence fuzzy subsets map to fuzzy subsets.

The Bidirectional FAM Theorem for Correlation-Minimum Encoding

Analysis of FAM recall uses the traditional [Klir, 1988] fuzzy-set notions of the *height* and the *normality* of fuzzy sets. The **height** $H(A)$ of fuzzy set A is the maximum fit value of A:

$$H(A) = \max_{1 \leq i \leq n} a_i$$

A fuzzy set is **normal** if $H(A) = 1$, if at least one fit value a_k is maximal: $a_k = 1$. In practice fuzzy sets are usually normal. We can extend a nonnormal fuzzy set to a normal fuzzy set by adding a dummy dimension with corresponding fit value $a_{n+1} = 1$.

Recall accuracy in fuzzy Hebb FAMs constructed with correlation-minimum encoding depends on the heights $H(A)$ and $H(B)$. Normal fuzzy sets exhibit perfect recall. Indeed (A, B) is a bidirectional fixed point—$A \circ M = B$, and $B \circ M^T = A$—if and only if $H(A) = H(B)$, which always holds if A and B are normal. This is a corollary of the bidirectional FAM theorem [Kosko, 1986a] for correlation-minimum encoding. Below we present a similar theorem for correlation-product encoding.

Correlation-minimum bidirectional FAM theorem. If $M = A^T \circ B$, then

(i)	$A \circ M = B$	iff $H(A) \geq H(B)$
(ii)	$B \circ M^T = A$	iff $H(B) \geq H(A)$
(iii)	$A' \circ M \subset B$	for any A'
(iv)	$B' \circ M^T \subset A$	for any B'

Proof. Observe that the height $H(A)$ equals the *fuzzy norm* of A:

$$A \circ A^T = \max_i a_i \wedge a_i = \max_i a_i = H(A)$$

Then

$$\begin{aligned} A \circ M &= A \circ (A^T \circ B) \\ &= (A \circ A^T) \circ B \\ &= H(A) \circ B \\ &= H(A) \wedge B \end{aligned}$$

So $H(A) \wedge B = B$ iff $H(A) \geq H(B)$, establishing (i). Now suppose A' is an arbitrary fit vector in I^n. Then

$$\begin{aligned} A' \circ M &= (A' \circ A^T) \circ B \\ &= (A' \circ A^T) \wedge B \end{aligned}$$

which establishes (iii) since $A' \circ A^T \leq H(A)$. A similar argument using $M^T = B^T \circ A$ establishes (ii) and (iv). **Q.E.D.**

The equality $A \circ A^T = H(A)$ implies an immediate corollary of the bidirectional FAM theorem. Supersets $A' \supset A$ behave the same as the encoded input associant A: $A' \circ M = B$ if $A \circ M = B$. Fuzzy Hebb FAMs ignore the information in the difference $A' - A$, when $A \subset A'$.

Correlation-Product Encoding

Correlation-product encoding provides an alternative fuzzy Hebbian encoding scheme. The standard mathematical outer product of the fit vectors A and B forms the FAM matrix M. Then

$$m_{ij} = a_i b_j \tag{8-11}$$

and in matrix notation,

$$M = A^T B \tag{8-12}$$

So the ith row of M equals the fit-scaled fuzzy set $a_i B$, and the jth column of M equals $b_j A^T$:

$$M = \begin{bmatrix} a_1 B \\ \hline \vdots \\ \hline a_n B \end{bmatrix} \tag{8-13}$$

$$= [b_1 A^T | \ldots | b_m A^T] \tag{8-14}$$

If $A = (.3\ \ .4\ \ .8\ \ 1)$ and $B = (.8\ \ .4\ \ .5)$ as above, we encode the FAM rule (A, B) with correlation product in the following matrix M:

$$M = \begin{pmatrix} .24 & .12 & .15 \\ .32 & .16 & .2 \\ .64 & .32 & .4 \\ .8 & .4 & .5 \end{pmatrix}$$

Note that if $A' = (0\ \ 0\ \ 0\ \ 1)$, then $A' \circ M = B$. The FAM system recalls output associant B to maximal degree. If $A' = (1\ \ 0\ \ 0\ \ 0)$, then $A' \circ M = (.24\ \ .12\ \ .15)$. The FAM system recalls output B only to degree .3.

Correlation-minimum encoding produces a matrix of clipped B sets, while correlation-product encoding produces a matrix of scaled B sets. In membership-function plots, the scaled fuzzy sets $a_i B$ all have the same shape as B. The clipped fuzzy sets $a_i \wedge B$ are flat at or above the a_i value. In this sense correlation-product encoding preserves more information than correlation-minimum encoding, an important point in fuzzy applications when we add output fuzzy sets together as in Equation (8-17) below. In the fuzzy-applications literature this often leads to the selection of correlation-product encoding.

Unfortunately, the fuzzy literature invariably confuses the correlation-product *encoding* scheme with the max-product composition method of recall or *inference*, as mentioned above. This widespread confusion warrants formal clarification.

In practice, and in the fuzzy applications developed in the next chapters, the input fuzzy set A' is a binary vector with one 1 and all other elements 0—a row of the n-by-n identity matrix (or a delta pulse in the continuous case). A' represents the occurrence of the crisp measurement datum x_i, such as a traffic density value of 30. When applied to the encoded FAM rule (A, B), the measurement value x_i activates A to degree a_i. This is part of the max-min composition recall process, for $A' \circ M = (A' \circ A^T) \circ B = a_i \wedge B$ or $a_i B$ depending on whether we encoded (A, B) in M with correlation-minimum or correlation-product encoding. We activate or "fire" the output associant B of the "rule" to degree a_i.

Since the values a'_i are binary, $a'_i m_{ij} = a'_i \wedge m_{ij}$. So the max-min and max-product composition operators coincide. We avoid this confusion by referring to both the recall process and the correlation encoding scheme as **correlation-minimum inference** when we combine correlation-minimum encoding with max-min composition, and as **correlation-product inference** when we combine correlation-product encoding with max-min composition.

We now prove the correlation-product version of the bidirectional FAM theorem.

Correlation-product bidirectional FAM theorem. If $M = A^T B$ and A and B are nonnull fit vectors, then

$$\begin{aligned} &\text{(i)} \quad A \circ M = B && \text{iff } H(A) = 1 \\ &\text{(ii)} \quad B \circ M^T = A && \text{iff } H(B) = 1 \\ &\text{(iii)} \quad A' \circ M \subset B && \text{for any } A' \\ &\text{(iv)} \quad B' \circ M^T \subset A && \text{for any } B' \end{aligned}$$

Proof.

$$\begin{aligned} A \circ M &= A \circ (A^T B) \\ &= (A \circ A^T) B \\ &= H(A) B \end{aligned}$$

Since B is not the empty set, $H(A)B = B$ iff $H(A) = 1$, establishing (i). ($A \circ M = B$ holds trivially if B is the empty set.) For an arbitrary fit vector A' in I^n:

$$\begin{aligned} A' \circ M &= (A' \circ A^T) B \\ &\subset H(A) B \\ &\subset B \end{aligned}$$

since $A' \circ A \le H(A)$, establishing (iii). (ii) and (iv) follow similarly using $M^T = B^T A$. Q.E.D.

Superimposing FAM Rules

Now suppose we have m FAM rules or associations $(A_1, B_1), \ldots, (A_m, B_m)$. The fuzzy Hebb encoding scheme (8-6) leads to m FAM matrices $M_1, \ldots, M_m$ to encode the associations. The natural neural-network temptation is to add, or in this case maximum, the m matrices pointwise to distributively encode the associations in a single matrix M:

$$M = \max_{1 \le k \le m} M_k \tag{8-15}$$

This superimposition scheme fails for fuzzy Hebbian encoding. The superimposed result tends to be the matrix $A^T \circ B$, where A and B denote the pointwise maximum of the respective m fit vectors A_k and B_k. We can see this from the pointwise inequality

$$\max_{1 \le k \le m} \min(a_i^k, b_j^k) \le \min(\max_{1 \le k \le m} a_i^k, \max_{1 \le k \le m} b_j^k) \tag{8-16}$$

Inequality (8-16) tends to hold with equality as m increases, since all maximum terms approach unity [Kosko, 1986a]. We lose the information in the m associations (A_k, B_k).

The fuzzy approach to the superimposition problem *additively superimposes the m recalled vectors B'_k* instead of the fuzzy Hebb matrices M_k. B'_k and M_k correspond to

$$\begin{aligned} A \circ M_k &= A \circ (A_k^T \circ B_k) \\ &= B'_k \end{aligned}$$

for any fit-vector input A applied in parallel to the bank of FAM rules (A_k, B_k). This requires separately storing the m associations (A_k, B_k), as if each association in the FAM bank represents a separate feedforward neural network.

Separate storage of FAM associations consumes space but provides an "audit trail" of the FAM inference procedure and avoids crosstalk. The user can directly determine which FAM rules contributed how much membership activation to a "concluded" output. Separate storage also provides knowledge-base modularity. The user can add or delete FAM-structured knowledge without disturbing stored knowledge. Both of these benefits are advantages over a pure neural-network architecture for encoding the same associations (A_k, B_k). Of course we can use neural networks exogenously to estimate, or even individually house, the associations (A_k, B_k).

Separate storage of FAM rules brings out another distinction between FAM systems and neural networks. A fit-vector input A activates all the FAM rules (A_k, B_k) in parallel but to different degrees. If A only partially "satisfies" the antecedent associant A_k, the consequent associant B_k only partially activates. If A does not satisfy A_k at all, B_k does not activate at all. B'_k equals the null vector.

Neural networks behave differently. They try to reconstruct the entire association (A_k, B_k) when stimulated with A. If A and A_k mismatch severely, a neural network will tend to emit a nonnull output B'_k, perhaps the result of the network dynamical system falling into a "spurious" attractor in the state space. We may desire this for metrical classification problems, but not for inferential problems and, arguably, for associative-memory problems. When we ask an expert a question outside his field of knowledge, it may be more prudent if he gives no response than if he gives an educated guess.

Recalled Outputs and "Defuzzification"

The recalled fit-vector output B equals a weighted sum of the individual recalled vectors B'_k:

$$B = \sum_{k=1}^{m} w_k B'_k \tag{8-17}$$

where the nonnegative weight w_k summarizes the credibility or strength of the kth FAM rule (A_k, B_k). The credibility weights w_k are immediate candidates for adaptive modification. In practice we choose $w_1 = \ldots = w_m = 1$ as a default.

In principle, though not in practice, the recalled fit-vector output equals a normalized sum of the B'_k fit vectors. This keeps the components of B unit-interval valued. We do not use normalization in practice because we invariably "defuzzify" the output distribution B to produce a single numerical output, a single value in the output universe of discourse $Y = \{y_1, \ldots, y_p\}$. The information in the output waveform B resides largely in the relative values of the membership degrees.

The simplest defuzzification scheme chooses that element $y_{\max}$ that has maximal membership in the output fuzzy set B:

$$m_B(y_{\max}) = \max_{1 \le j \le k} m_B(y_j) \tag{8-18}$$

The popular probabilistic methods of maximum-likelihood and maximum-a-posteriori parameter estimation motivate this **maximum-membership defuzzification scheme.**

The maximum-membership defuzzification scheme has two fundamental problems. First, the mode of the B distribution is not unique. This problem affects correlation-minimum encoding, as the representation (8-8) shows, more than it affects correlation-product encoding. Since the minimum operator clips off the top of the B_k fit vectors, the additively combined output fit vector B tends to be flat over many regions of universe of discourse Y. For continuous membership functions this leads to infinitely many modes. Even for quantized fuzzy sets, there may be many modes.

In practice we can average multiple modes. For large FAM banks of "independent" FAM rules, some form of the central limit theorem (whose proof ultimately depends on Fourier transformability, not probability) tends to hold. The waveform B tends to resemble a Gaussian membership function. So a unique mode tends to emerge. It tends to emerge with fewer samples if we use correlation-product encoding.

Second, the maximum-membership scheme ignores the information in much of the waveform B. Again correlation-minimum encoding compounds the problem. In practice B is often highly asymmetric, even if it is unimodal. Infinitely many output distributions can share the same mode.

The natural alternative is the **fuzzy centroid defuzzification** scheme. We directly compute the real-valued output as a (normalized) convex combination of fit values, the *fuzzy centroid* $\bar{B}$ of fit-vector B with respect to output space Y:

$$\bar{B} = \frac{\sum_{j=1}^{p} y_j m_B(y_j)}{\sum_{j=1}^{p} m_B(y_j)} \tag{8-19}$$

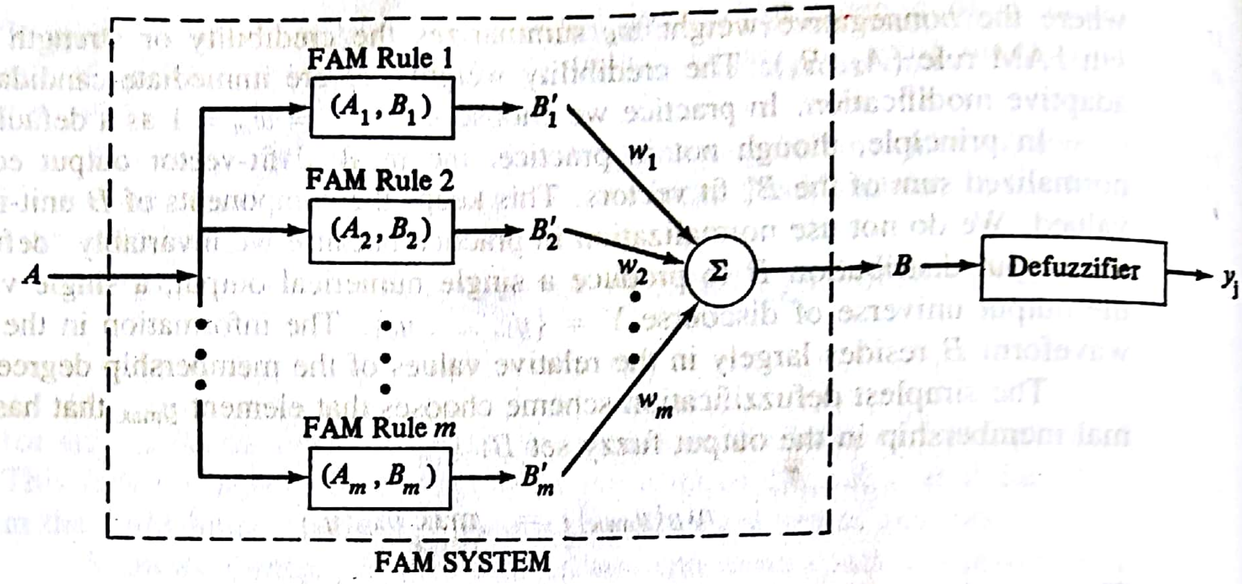

FIGURE 8.3 FAM system architecture. The FAM system F maps fuzzy sets in the unit cube I^n to fuzzy sets in the unit cube I^p. Binary input sets model exact input data. In general only an uncertainty estimate of the system state confronts the FAM system. So A is a proper fuzzy set. The user can defuzzify output fuzzy set B to yield exact output data, reducing the FAM system to a mapping between Boolean cubes.

The fuzzy centroid is unique and uses all the information in the output distribution B. For symmetric unimodal distributions the mode and fuzzy centroid coincide. In many cases we must replace the discrete sums in (8-19) with integrals over continuously infinite spaces. We show in Chapter 11, though, that for libraries of trapezoidal fuzzy-set values we can replace such a ratio of integrals with a ratio of simple discrete sums.

Computing the centroid (8-19) is the only step in the FAM inference procedure that requires division. All other operations are inner products, pairwise minima, and additions. This promises realization in a fuzzy optical processor. Already some form of this FAM-inference scheme has led to digital [Togai, 1986] and analog [Yamakawa, 1987, 1988] VLSI circuitry.

FAM System Architecture

Figure 8.3 schematizes the architecture of the nonlinear FAM system F. Note that F maps fuzzy sets to fuzzy sets: $F(A) = B$. So F defines a fuzzy-system transformation $F: I^n \rightarrow I^p$. In practice A equals a bit vector with one unity value, $a_i = 1$, and all other fit values zero, $a_j = 0$, or a delta pulse.

We defuzzify the output fuzzy set B with the centroid technique to produce an exact element y_j in the output universe of discourse Y. In effect defuzzification produces an output binary vector O, again with one element 1 and the rest 0s. At this level the FAM system F maps sets to sets, reducing the fuzzy system F to a mapping between Boolean cubes, F: $\{0, 1\}^n \rightarrow \{0, 1\}^p$. In many applications we model X and Y as continuous universes of discourse. So n and p are quite large. We shall call such systems binary input-output FAMs.

Binary Input-Output FAMs: Inverted-Pendulum Example

Binary input-output FAMs (BIOFAMs) are the most popular fuzzy systems for applications. BIOFAMs map system state-variable data to control, classification, or other output data. In the case of traffic control, a BIOFAM maps traffic densities to green (and red) light durations.

BIOFAMs easily extend to multiple FAM-rule antecedents, to mappings from product cubes to product cubes. There has been little theoretical justification for this extension, aside from Mamdani's [1977] original suggestion to multiply relational matrices. In the next section we present a general method for dealing with multiantecedent FAM rules. First, though, we present the BIOFAM algorithm by illustrating it, and the FAM construction procedure, on a standard control problem.

Consider an inverted pendulum. We wish to adjust a motor to balance an inverted pendulum in two dimensions. The inverted pendulum is a classical control problem and admits a math-model control solution. This provides a formal benchmark for BIOFAM pendulum controllers.

There are two state fuzzy variables and one control fuzzy variable. The first state fuzzy variable is the *angle* θ that the pendulum shaft makes with the vertical. Zero angle corresponds to the vertical position. Positive angles are to the right of the vertical, negative angles to the left.

The second state fuzzy variable is the *angular velocity* $\Delta\theta$. In practice we approximate the instantaneous angular velocity $\Delta\theta$ as the difference between the present angle measurement θ_t and the previous angle measurement θ_{t-1}:

$$\Delta\theta_t = \theta_t - \theta_{t-1}$$

The control fuzzy variable is the *motor current* or angular velocity v_t. The velocity can be positive or negative. We expect that if the pendulum falls to the right, the motor velocity should be negative to compensate. If the pendulum falls to the left, the motor velocity should be positive. If the pendulum successfully balances at the vertical, the motor velocity should be zero.

The real line R is the universe of discourse of the three fuzzy variables. In practice we restrict each universe of discourse to a comparatively small interval, such as $[-90, 90]$ for the pendulum angle, centered about zero.

We can quantize each universe of discourse into five overlapping fuzzy-set values. We know that the fuzzy variables can be positive, zero, or negative. We can quantize the magnitudes of the fuzzy variables finely or coarsely. Suppose we quantize the magnitudes as small, medium, and large. This leads to seven fuzzy-set values:

NL: Negative Large
NM: Negative Medium
NS: Negative Small
ZE: Zero
PS: Positive Small
PM: Positive Medium
PL: Positive Large

For example, θ is a fuzzy *variable* that takes NL as a fuzzy-set *value*. Different fuzzy quantizations of the angle universe of discourse allow the fuzzy variable θ to assume different fuzzy-set values. The expressive power of the FAM approach stems from these fuzzy-set quantizations. In one stroke we reduce system dimensions, and we describe a nonlinear numerical process with linguistic commonsense terms.

We are not concerned with the exact shape of the fuzzy sets defined on each of the three universes of discourse. In practice the quantizing fuzzy sets are usually symmetric triangles or trapezoids centered about representive values. (We can think of such sets as *fuzzy numbers*.) The set ZE may define a Gaussian curve for the pendulum angle θ, a triangle for the angular velocity $\Delta\theta$, and a trapezoid for the motor current v. But all the ZE fuzzy sets center about the numerical value zero, which will have maximum membership in the set of zero values.

How much should contiguous fuzzy sets overlap? This design issue depends on the problem at hand. Too much overlap blurs the distinction between the fuzzy-set values. Too little overlap tends to resemble bivalent control, producing excessive overshoot and undershoot. In Chapter 11 we determine experimentally the following default heuristic for ideal overlap: *Contiguous fuzzy sets in a library should overlap approximately 25 percent.*

Inverted-pendulum FAM rules are triples, such as (NM, ZE; PM). They describe how to modify the control variable for observed values of the pendulum state variables. A FAM rule associates a motor-velocity fuzzy-set value with a pendulum-angle fuzzy-set value and an angular-velocity fuzzy-set value. So we can interpret the triple (NM, ZE; PM) as the set-level implication

IF the pendulum angle θ is negative but medium
AND the angular velocity $\Delta\theta$ is about zero,
THEN the motor velocity should be positive but medium

These commonsensical FAM rules are comparatively easy to articulate in natural language. Consider a terser linguistic version of the same two-antecedent FAM rule:

$$\begin{array}{ll} \text{IF} & \theta = \text{NM AND } \Delta\theta = \text{ZE} \\ \text{THEN} & v = \text{PM} \end{array}$$

Even this mild level of formalism may inhibit the knowledge-acquisition process.

On the other hand, the still terser FAM triple (NM, ZE; PM) allows knowledge to be acquired simply by filling in a few entries in a linguistic FAM-bank matrix. In practice this often allows us to develop a working system in minutes.

We specify the pendulum FAM system when we choose a *FAM bank* of two-antecedent FAM rules. Perhaps the first FAM rule to choose is the *steady-state FAM rule*:(ZE, ZE; ZE). The steady-state FAM rule describes what to do in equilibrium. For the inverted pendulum we should do nothing.

Many control problems require nulling a scalar error measure. We can control many multivariable problems by nulling the norms of the system error vector and error-velocity vectors, or, better, by directly nulling the individual scalar variables. (Chapter 11 shows how error nulling can control a real-time target tracking system.) Adaptive error-nulling extends the FAM methodology to nonlinear estimation, control, and decision problems of high dimension.

The pendulum FAM bank is a 7-by-7 matrix with linguistic fuzzy-set entries. We index the columns by the seven fuzzy sets that quantize the angle θ universe of discourse. We index the rows by the seven fuzzy sets that quantize the angular velocity $\Delta\theta$ universe of discourse.

Each matrix entry can equal one of seven motor-current fuzzy-set values or equal no fuzzy set at all. Since a FAM rule is a mapping or function, there is exactly one output motor-current value for every pair of angle and angular-velocity values. So the 49 entries in the FAM bank matrix represent a subset of the 343 (7^3) possible two-antecedent FAM rules. In practice most of the entries are blank. In the adaptive FAM case discussed below, we adaptively generate the entries from process sample data.

Common sense and engineering judgment dictate the entries in the pendulum FAM-bank matrix. Suppose the pendulum does not move. So $\Delta\theta$ = ZE. If the pendulum tilts to the right of vertical, the motor velocity should be negative to compensate. The farther the pendulum tilts to the right, the larger the negative motor velocity should be. The motor velocity should be positive if the pendulum tilts to the left. So the fourth row of the FAM bank matrix, which corresponds to $\Delta\theta$ = ZE, should equal the ordinal inverse of the θ row values. This assignment includes the steady-state FAM rule (ZE, ZE; ZE).

Now suppose the angle θ is zero but the pendulum moves. If the angular velocity is negative, the pendulum will overshoot to the left. So the motor velocity should be positive to compensate. If the angular velocity is positive, the motor velocity should be negative. The greater the angular velocity is in magnitude, the greater the motor velocity should be in magnitude. So the fourth column of the FAM-bank matrix, which corresponds to θ = ZE, should equal the ordinal inverse of the $\Delta\theta$ column values. This assignment also includes the steady-state FAM rule.

Positive θ values with negative $\Delta\theta$ values should produce negative motor-current values, since the pendulum heads toward the vertical. So (PS, NS; NS) is a candidate FAM rule. Symmetrically, negative θ values with positive $\Delta\theta$ values should produce positive motor-current values. So (NS, PS; PS) is another candidate FAM rule.

This gives 15 FAM rules altogether. In practice these rules can successfully balance an inverted pendulum. Different, and smaller, subsets of FAM rules can also balance the pendulum. The software problems at the end of the chapter explore these cases.

We can represent the bank of 15 FAM rules as the 7-by-7 linguistic matrix

$\Delta\theta$ \ θ	NL	NM	NS	ZE	PS	PM	PL
NL				PL			
NM				PM			
NS				PS	NS		
ZE	PL	PM	PS	ZE	NS	NM	NL
PS			PS	NS			
PM				NM			
PL				NL			

The BIOFAM system F admits a geometric interpretation. The set of all possible input-outpairs $(\theta, \Delta\theta; F(\theta, \Delta\theta))$ defines a *FAM surface* in the input-output product space, in this case in R^3. We plot examples of these control surfaces in Chapters 9, 10 and 11.

The BIOFAM *inference procedure* activates in parallel the antecedents of all 15 FAM rules. The binary or pulse nature of inputs picks off single fit values from the quantizing fuzzy-set values of the fuzzy variables. We can use either the correlation-minimum or correlation-product inferencing technique. For simplicity we shall illustrate the procedure with correlation-minimum inferencing.

Suppose the current pendulum angle θ equals 15 degrees and the angular velocity $\Delta\theta$ equals -10. This amounts to passing two bit vectors of one 1 and all else 0 through the BIOFAM system. What is the corresponding motor-current value $v = F(15, -10)$?

Consider first how the input data pair $(15, -10)$ activates the steady-state FAM rule (ZE, ZE; ZE). Suppose we define the antecedent and consequent fuzzy sets for ZE with the triangular fuzzy-set membership functions in Figure 8.4. Then the angle datum 15 defines a zero angle value to degree .2: $m^{\theta}_{\text{ZE}}(15) = .2$. The angular-velocity datum -10 defines a zero angular-velocity value to degree .5 : $m^{\Delta\theta}_{\text{ZE}}(-10) = .5$.

We combine the antecedent fit values with minimum or maximum depending on whether we combine the antecedent fuzzy sets with the conjunctive AND or the disjunctive OR. Intuitively, it should be at least as difficult to satisfy both antecedent conditions as to satisfy either one separately.

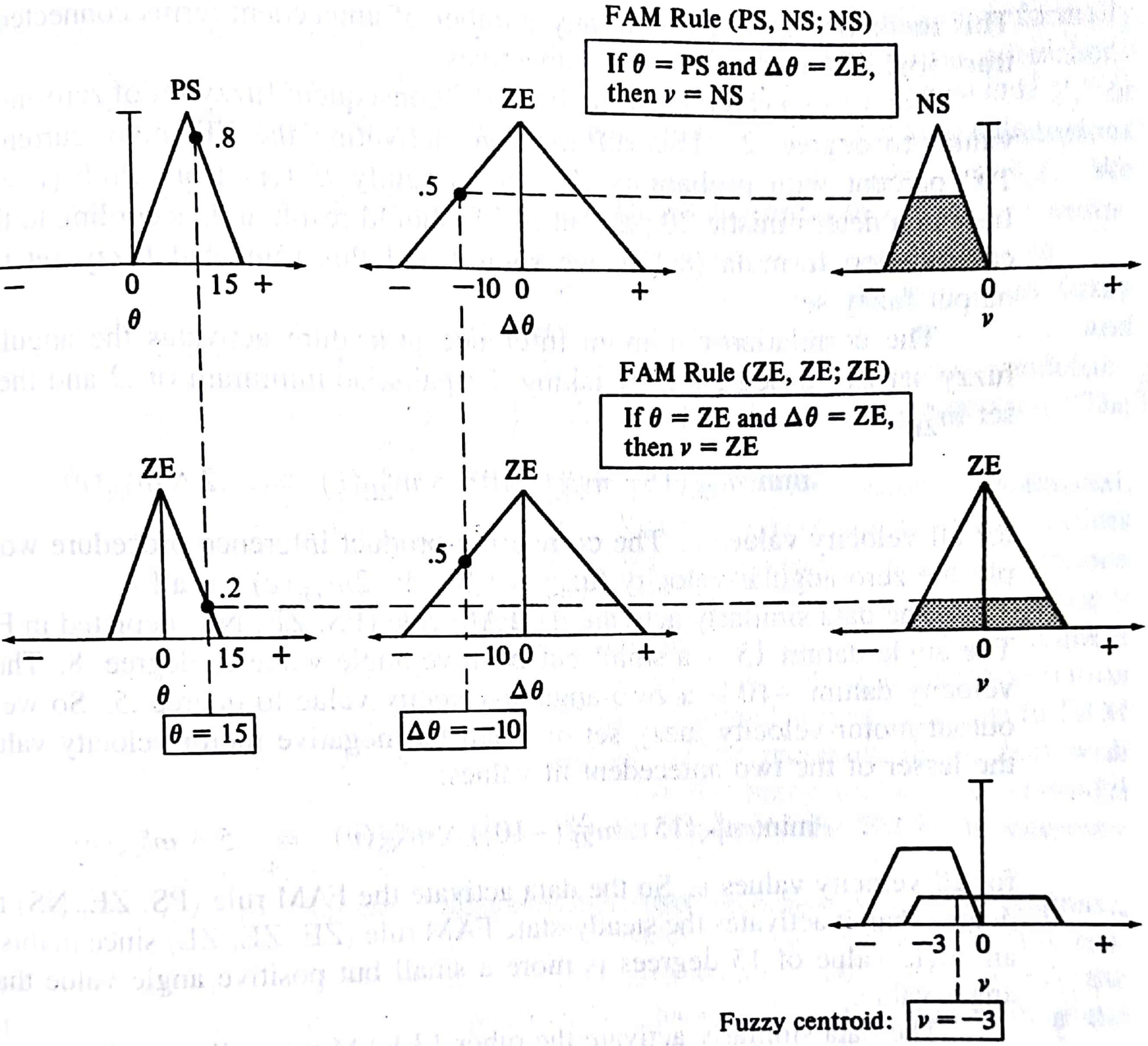

FIGURE 8.4 FAM correlation-minimum inference procedure. The FAM system consists of the two two-antecedent FAM rules (PS, ZE; NS) and (ZE, ZE; ZE). The input angle datum equals 15 and is more a small but positive angle value than a zero angle value. The input angular-velocity datum equals −10, and is a zero angular-velocity value only to degree .5. The system combines antecedent fit values with minimum, since the conjunction AND combines the antecedent terms. The combined fit value then scales the consequent fuzzy set with pairwise minimum. The system adds the minimum-scaled output fuzzy sets and computes the fuzzy centroid of this output waveform. This yields the system output-current value −3.

The FAM-rule notation (ZE, ZE; ZE) implicitly assumes that we combine antecedent fuzzy sets conjunctively with AND. So the data satisfy the compound antecedent of the FAM rule $(ZE, ZE; ZE)$ to degree

$$\min(m^{\theta}_{\text{ZE}}(15), m^{\Delta\theta}_{\text{ZE}}(-10)) = \min(.2, .5) = .2$$

This methodology extends to any number of antecedent terms connected with arbitrary logical (set-theoretical) connectives.

The system should now activate the consequent fuzzy set of zero-motor-current values to degree .2. This differs from activating the ZE motor-current fuzzy set 100 percent with probability .2, and certainly differs from Prob $\{v = 0\} = .2$. Instead a deterministic 20 percent of ZE should result and, according to the additive combination formula (8-17), we should add this truncated fuzzy set to the final output fuzzy set.

The correlation-minimum inference procedure activates the angular-velocity fuzzy set ZE to degree .2 by taking the pairwise minimum of .2 and the ZE fuzzy set m^v_{ZE}:

$$\min(m^{\theta}_{\text{ZE}}(15), m^{\Delta\theta}_{\text{ZE}}(-10)) \wedge m^v_{\text{ZE}}(v) = .2 \wedge m^v_{\text{ZE}}(v)$$

for all velocity values v. The correlation-product inference procedure would multiply the zero-angular-velocity fuzzy set by .2: $.2m^v_{\text{ZE}}(v)$ for all v.

The data similarly activate the FAM rule (PS, ZE; NS) depicted in Figure 8.4. The angle datum 15 is a small but positive angle value to degree .8. The angular-velocity datum -10 is a zero-angular-velocity value to degree .5. So we scale the output motor-velocity fuzzy set of small but negative motor-velocity values by .5, the lesser of the two antecedent fit values:

$$\min(m^{\theta}_{\text{PS}}(15), m^{\Delta\theta}_{\text{ZE}}(-10)) \wedge m^v_{\text{NS}}(v) = .5 \wedge m^v_{NS}(v)$$

for all velocity values v. So the data activate the FAM rule (PS, ZE; NS) to greater degree than it activates the steady-state FAM rule (ZE, ZE; ZE) since in this example an angle value of 15 degrees is more a small but positive angle value than a zero angle value.

The data similarly activate the other 13 FAM rules. We combine the resulting minimum-scaled consequent fuzzy sets according to (8-17) by summing pointwise. We can then compute the fuzzy centroid with Equation (8-19), with perhaps integrals replacing the discrete sums, to determine the specific output motor velocity v. In Chapter 11 we show that, for symmetric fuzzy-set values of fuzzy variables, we can always compute the centroid exactly with simple discrete sums even if the fuzzy sets are continuous. In many real-time applications we must repeat this entire FAM inference procedure hundreds, perhaps thousands, of times per second. This may require fuzzy VLSI or optical processors.

Figure 8.4 illustrates the equal-weight additive combination procedure for just the FAM rules (ZE, ZE; ZE) and (PS, ZE; NS). In this case the fuzzy-centroidal motor-velocity value equals -3.