In: Kosko, B., NEURAL NETWORKS AND FUZZY SYSTEMS, Prentice-Hall, 1990

N91-21779

CHAPTER 17

FUZZY ASSOCIATIVE MEMORIES

Fuzzy Systems as Between-Cube Mappings

In Chapter 16, we introduced continuous 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 is to consider 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 fuzzy systems to be adaptively inferred and modified, perhaps with neural or statistical techniques, directly from problem domain sample data.

Between-cube theory is fuzzy systems theory. A fuzzy set is a point in a cube. A fuzzy system is a mapping between cubes. A fuzzy system S maps fuzzy sets to fuzzy sets. Thus a **fuzzy system** S is a transformation $S : I^n \rightarrow I^p$. 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 be 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 $S : I^{n_1} \times \ldots \times I^{n_r} \rightarrow I^{p_1} \times \ldots \times I^{p_s}$. 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 $f : X \rightarrow Y$ that maps points in X to points in Y can be extended to map the fuzzy subsets of X to the fuzzy subsets of Y. The so-called *extension principle* is used to define the set-function $f : F(2^X) \rightarrow F(2^Y)$, where $F(2^X)$ is 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. One can show [Kosko, 1986a-87] that in general 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 $S : I^n \rightarrow I^p$ 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 the minimal FAMs need not be adaptively trained. As discussed below, structured knowledge of the form "If traffic is heavy in this direction, then keep the stop light green longer" can be directly encoded in a Hebbian-style FAM matrix. In practice we can eliminate even this matrix. In its place the user encodes the fuzzy-set association (HEAVY, LONGER) as a single linguistic entry in a FAM bank 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 $B_1', \ldots, B_m'$. In the simplest case B is a weighted average of the partially activated sets:

$$B \;=\; w_1 B_1' \;+\; \ldots \;+\; w_m \, B_m' \quad ,$$

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. Structured knowledge can then be naturally, and in many cases easily, obtained. 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 $(A^1, A^2;\; B)$ stood for conjunction instead of, say, disjunction.

We specify in advance the numerical universes of discourse X^1, X^2, and Y. For each universe of discourse 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. A different library of fuzzy sets similarly quantizes the output space Y. Once the library of fuzzy sets is defined, we construct the FAM by choosing appropriate combinations of input and output fuzzy sets. We can use adaptive techniques to 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, for "blindly" generating and refining the bank of FAM rules. In some cases we can use supervised learning techniques, though we need 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 a linguistic "rule matrix" be partially filled in. 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 be in the same numerical space of alternatives. So both systems correspond to a surface or manifold in the input-output product space $X \times Y$. We present examples of these surfaces in Chapters 18 and 19.

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 fuzzy techniques have been most successfully applied 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.). The application of both approaches to new problem areas is just beginning, amid varying amounts of enthusiasm and scepticism.

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 are from the input-output product space $X \times Y$. Figure 17.1 illustrates the geometry of fuzzy-set and numerical-point samples taken from the function $f\colon X \rightarrow Y$.

The fuzzy-set association $(A_i,\ B_i)$ is sometimes called 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 associations are housed in matrices, not conditionals. 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.**

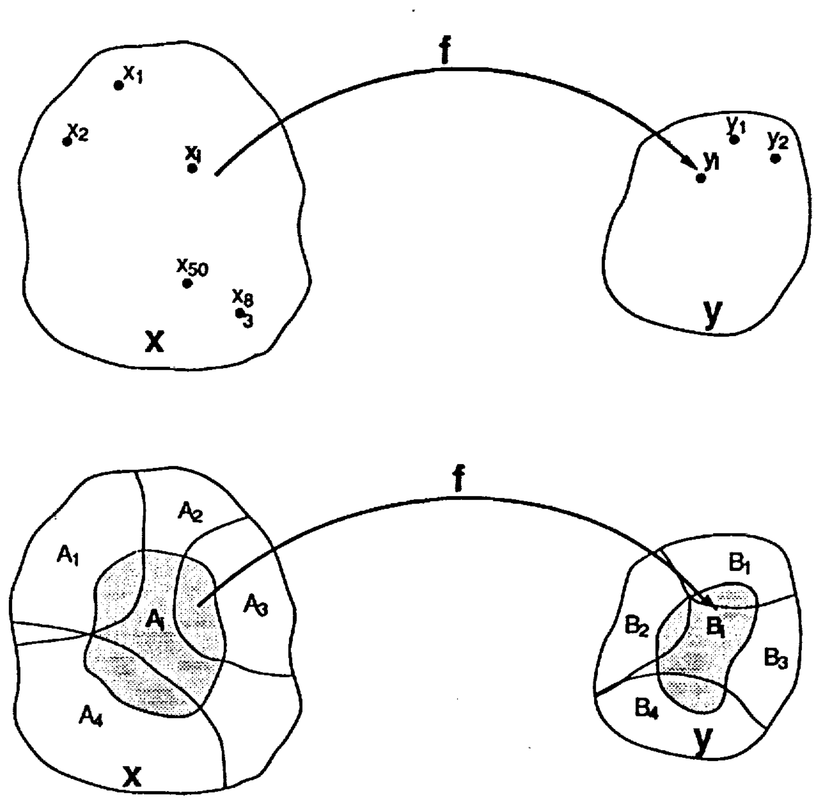

FIGURE 17.1 Function f maps domain X to range Y. In the first illustration we use several numerical point samples $(x_i,\ y_i)$ to estimate f: $X \longrightarrow Y$. In the second case we use only a few fuzzy subsets A_i of X and B_i of Y. The fuzzy association $(A_i,\ B_i)$ represents system structure, 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.

The fuzzy-set sample $(A_i,\ B_i)$ encodes *structure*. It represents a mapping itself, a minimal *fuzzy association* of part of the output space with part of the input space. 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 be the result of an adaptive clustering algorithm.

Consider a fuzzy association that might be used in 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). 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 be obtained 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 definitions for the variables in question.

The degree to which the traffic light is kept green longer depends on the degree to which the measurement is heavy. In the simplest case the two degrees are the same. In general they 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 that are activated to different degrees by incoming data.

Neural vs. Fuzzy Representation of Structured Knowledge

The functional distinction between how fuzzy and neural systems differ 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 is to encode two associated numerical vectors. One vector represents the input associant HEAVY. The other vector 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 is to encode 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 multi-layer neural networks trained with the backpropagation algorithm 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 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 sample-space centroids are asymptotically estimated.) 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 be in terms of 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? Was an expert asked 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 is adapting on-line, we know only past samples. The remainder of the sample set is in the unsampled future.

In contrast, fuzzy systems directly encode the linguistic sample (HEAVY, LONGER) in a dedicated numerical matrix. The default encoding technique is the fuzzy Hebb procedure discussed below. For practical problems, as mentioned above, the numerical matrix need not be stored. Indeed it need not even be formed. Certain numerical inputs permit this simplification, as we shall see below. In general we describe inputs by an uncertainty distribution, probabilistic or fuzzy. Then we must use the entire matrix.

For instance, if a *heavy traffic* input is simply the number 150, we can omit the FAM matrix. 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.) This difference is explained below. The dimensions of the linguistic FAM bank matrix are usually small. The dimensions reflect the quantization levels of the input and output spaces.

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 non-articulated, behavioral rules that improve with experience.

We can acquire structured expertise in the fuzzy terminology of the knowledge source, the "expert." This requires little or no force-fitting. Such is the expressive power of fuzziness. Yet in the numerical domain we can prove theorems and design hardware.

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, though perhaps adaptively so. In the same spirit, no one believes that the brain is a single unstructured neural network.

FAMS as Mappings

Fuzzy associative memories (FAMs) are transformations. *FAMs map fuzzy sets to fuzzy sets.* They map unit cubes to unit cubes. This is evident in Figure 17.1. In the simplest case the FAM consists of a single association, such as (HEAVY, LONGER). In general the FAM consists of a bank of different FAM associations. Each association is represented by a different numerical FAM matrix, or a different entry in a FAM-bank matrix. These matrices are not combined as with neural network associative memory (outer-product) matrices. (An exception is the *fuzzy cognitive map* [Kosko, 1988; Taber, 1987, 1990].) The matrices are stored separately but accessed in parallel.

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 fuzzy subsets of X and Y. So A is point in the n-dimensional unit hypercube $I^n = [0, 1]^n$, and B is a point in the p-dimensional fuzzy cube I^p. Equivalently, we can think of A and B as membership functions m_A and m_B mapping 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 $m_A : X \longrightarrow [0, 1]$ and $m_B : Y \longrightarrow [0, 1]$. We shall freely view sets both as functions and as points.

The geometric *sets-as-points* interpretation of 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 as the index i increases, perhaps approximating a sigmoid membership function. Figure 17.2 illustrates three possible fuzzy subsets of the universe of discourse X.

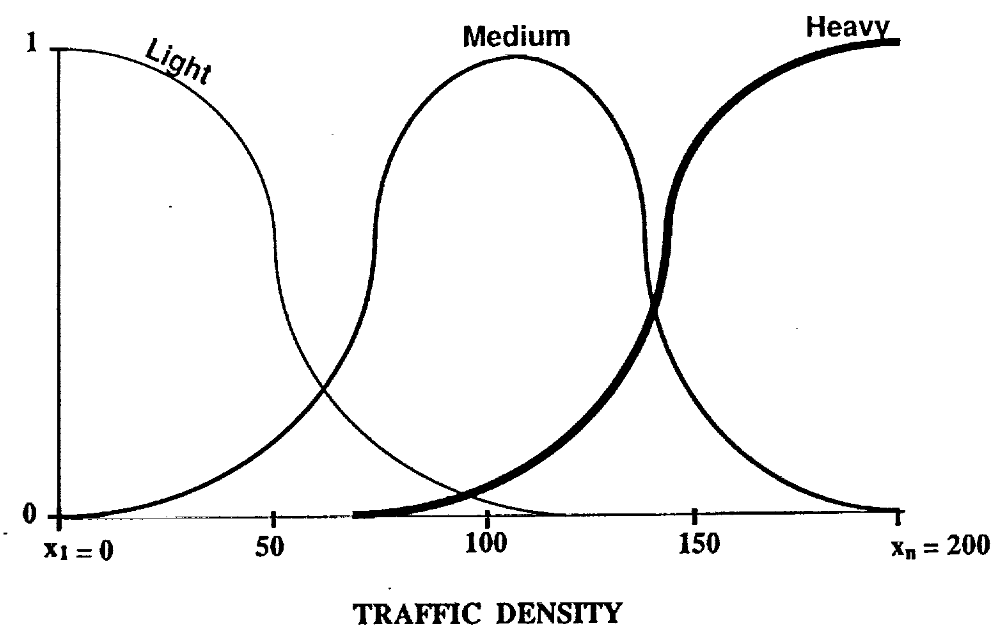

FIGURE 17.2 Three possible fuzzy subsets of traffic density space X. Each fuzzy sample corresponds to such a subset. We draw the fuzzy sets as continuous membership functions. In practice membership values are quantized. So the sets are points in the unit hypercube I^n. Each fuzzy sample corresponds to such a subset.

Fuzzy Vector-Matrix Multiplication: Max-Min Composition

Fuzzy vector-matrix multiplication is similar to classical vector-matrix multiplication. We replace pairwise multiplications with pairwise minima. We replace column (row) sums with column (row) maxima. We denote this **fuzzy vector-matrix composition** relation, or the **max-min composition** relation [Klir, 1988], by the composition operator "o". For row fit vectors A and B and fuzzy n-by-p matrix M (a point in $I^{n \times p}$):

$$A \circ M = B , \tag{1}$$

where we compute the "recalled" component b_j by taking the fuzzy inner product of fit vector A with the jth column of M:

$$b_j = \max_{1 \leq i \leq n} \min(a_i, m_{ij}) \quad . \tag{2}$$

Suppose we compose the fit vector $A = (.3\ .4\ .8\ 1)$ with the fuzzy matrix M given by

$$M = \begin{pmatrix} .2 & .8 & .7 \\ .7 & .6 & .6 \\ .8 & .1 & .5 \\ 0 & .2 & .3 \end{pmatrix} \quad .$$

Then we compute the "recalled" fit vector $B = A \circ M$ component-wise as

$$\begin{aligned} b_1 &= \max\{\min(.3,\ .2),\ \min(.4,\ .7),\ \min(.8,\ .8),\ \min(1,\ 0)\} \\ &= \max(.2,\ .4,\ .8,\ 0) \\ &= .8 \ , \\ b_2 &= \max(.3,\ .4,\ .1,\ .2) \\ &= .4 \ , \\ b_3 &= \max(.3,\ .4,\ .5,\ .3) \\ &= .5 \ . \end{aligned}$$

So $B = (.8\ .4\ .5)$. If we somehow encoded $(A,\ B)$ in the FAM matrix M, we would say that the FAM system exhibits *perfect recall* in the forward direction.

The neural interpretation of max-min composition is that each neuron in field F_Y (or field F_B) generates its signal/activation value by fuzzy linear composition. Passing information back through M^T allows us to interpret the fuzzy system as a bidirectional associative memory (BAM). The Bidirectional FAM Theorems below characterize successful BAM recall for fuzzy correlation or Hebbian learning.

For completeness we also mention the **max-product composition** operator, which replaces minimum with product in (2):

$$b_j \quad = \quad \max_{1 \leq i \leq n} a_i \, m_{ij} \quad .$$

In the fuzzy literature this composition operator is often confused with the fuzzy correlation encoding scheme discussed below. Max-product composition is a method for "multiplying" fuzzy matrices or vectors. Fuzzy correlation, which also uses pairwise products of fit values, is a method for constructing fuzzy matrices. In practice, and in the following discussion, we use only max-min composition.

FUZZY HEBB FAMs

Most fuzzy systems found in applications are fuzzy Hebb FAMs [Kosko, 1986b]. They are fuzzy systems $S : I^n \rightarrow I^p$ constructed in a simple neural-like 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 $S_i \, S_j$:

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

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

$$M \quad = \quad X^T \, Y \quad . \tag{4}$$

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

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

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

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

Mamdani [1977] and Togai [1986] independently arrived at the fuzzy Hebbian prescription (5) as a multi-valued logical-implication operator: $\text{truth}(a_i \rightarrow b_i) = \min(a_i, b_j)$. 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: $m_{ij} = \text{truth}(a_i \rightarrow b_j) = \min(1, 1 - a_i + b_j)$. The problem with the Lukasiewicz operator is that it usually 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 (6), as the fuzzy auto-correlation matrix:

$$M \quad = A^T \circ A \quad . \tag{7}$$

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 B still be recalled if we replace the original matrix M with the fuzzy Hebb matrix found with (6)? Substituting A and B in (6) gives

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

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

$$M = \left[\begin{array}{c} a_1 \wedge B \\ \hline \vdots \\ \hline a_n \wedge B \end{array}\right] \tag{8}$$

$$= [b_1 \wedge A^T \mid \ldots \mid b_m \wedge A^T] \quad , \tag{9}$$

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

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

Hence if some $a_k = 1$, then the kth row of M is B. If some $b_l = 1$, the lth column of M is 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 is B.

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

$$A \circ M = (.8 \ .4 \ .5) = B \quad ,$$

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

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 16. 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 is a general property: If $B' = A \circ M$ differs from B, then B' is a proper subset of B. Hence fuzzy subsets truly 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:

$$H(A) \quad = \quad \max_{1 \le i \le n} a_i \quad .$$

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 the content 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

$$\text{(i)} \quad A \circ M \; = \; B \quad \text{iff} \quad H(A) \ge H(B) \quad ,$$

$$\text{(ii)} \quad B \circ M^T \; = \; A \quad \text{iff} \quad H(B) \ge H(A) \quad ,$$

$$\text{(iii)} \quad A' \circ M \; \subset \; B \qquad \text{for any } A' \; .$$

$$\text{(iv)} \quad B' \circ M^T \; \subset \; A \qquad \text{for any } B' \; .$$

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

$$A \circ A^T \;=\; \max_i\, a_i \wedge a_i \;=\; \max_i\, a_i \;=\; H(A) \quad .$$

Then

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

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

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

which establishes (iii). 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

An alternative fuzzy Hebbian encoding scheme is **correlation-product encoding.** The standard mathematical outer product of the fit vectors A and B forms the FAM matrix M. This is given pointwise as

$$m_{ij} \quad = \quad a_i \; b_j \quad , \tag{11}$$

and in matrix notation as

$$M \quad = \quad A^T \; B \quad . \tag{12}$$

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

$$M \quad = \quad \left[\begin{array}{c} a_1 \; B \\ \hline \vdots \\ \hline a_n \; B \end{array} \right] \tag{13}$$

$$= \quad [b_1 \; A^T \mid \ldots \mid b_m \; A^T] \quad , \tag{14}$$

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:

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

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

Correlation-minimum encoding produces a matrix of clipped B sets. 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 largely flat. In this sense correlation-product encoding preserves more information than correlation-minimum encoding, an important point in fuzzy applications when output fuzzy sets are added together as in equation (17) below. In the fuzzy-applications literature this often leads to the selection of correlation-product encoding.

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

In practice, and in the fuzzy control applications developed in Chapters 18 and 19, 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. 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 correlation-minimum or correlation-product encoding is used. 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 correlation-minimum encoding is combined with max-min composition, and as **correlation-product inference** when correlation-product encoding is combined 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 non-null fit vectors, then

(i) $A \circ M = B$ iff $H(A) = 1$,

(ii) $B \circ M^T = A$ iff $H(B) = 1$,

(iii) $A' \circ M \subset B$ for any A' .

(iv) $B' \circ M^T \subset A$ for any B' .

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 $A' \circ A \leq H(A)$, establishing (iii). (ii) and (iv) are proved 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 (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:

$$M = \max_{1 \leq k \leq m} M_k \ . \tag{15}$$

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

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

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

The fuzzy approach to the superimposition problem is to *additively superimpose the m recalled vectors B'_k* instead of the fuzzy Hebb matrices M_k. B'_k and M_k are given by

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

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 were a separate feedforward neural network.

Separate storage of FAM associations is costly but provides an "audit trail" of the FAM inference procedure. 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 is only partially activated. If A does not satisfy A_k at all, B_k does not activate at all. B'_k is 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 non-null output B'_k, perhaps the result of the network dynamical system falling into a "spurious" attractor in the state space. This may be desirable for metrical classification problems. It is undesirable for inferential problems and, arguably, for associative memory problems. When we ask an expert a question outside his field of knowledge, in many cases it is more prudent for him to give no response than to give an educated, though wild, guess.

Recalled Outputs and "Defuzzification"

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

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

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 is 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 is to choose that element $y_{\max}$ that has maximal membership in the output fuzzy set B:

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

The popular probabilistic methods of maximum-likelihood and maximum-a-posteriori parameter estimation motivate this **maximum-membership defuzzification** scheme. The maximum-membership scheme (18) is also computationally light.

There are two fundamental problems with the maximum-membership defuzzification scheme. First, the mode of the B distribution is not unique. This is especially troublesome with correlation-minimum encoding, as the representation (8) shows, and somewhat less troublesome with 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 apply. 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:

$$\bar{B} \quad = \quad \frac{\sum_{j=1}^{p} y_j \; m_B(y_j)}{\sum_{j=1}^{p} m_B(y_j)} \quad . \tag{19}$$

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 (19) with integrals over continuously infinite spaces. We show in Chapter 19, though, that for libraries of trapezoidal fuzzy sets we can replace such a ratio of integrals with a ratio of simple discrete sums.

Note that computing the centroid (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, 1987-88] VLSI circuitry.

FAM System Architecture

Figure 17.3 schematizes the architecture of the nonlinear FAM system F. Note that F maps fuzzy sets to fuzzy sets: $F(A) = B$. So F is in fact a fuzzy-system transformation $F: I^n \rightarrow I^p$. In practice A is a bit vector with one unity value, $a_i = 1$, and all other fit values zero, $a_j = 0$.

The output fuzzy set B is usually defuzzified 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**.

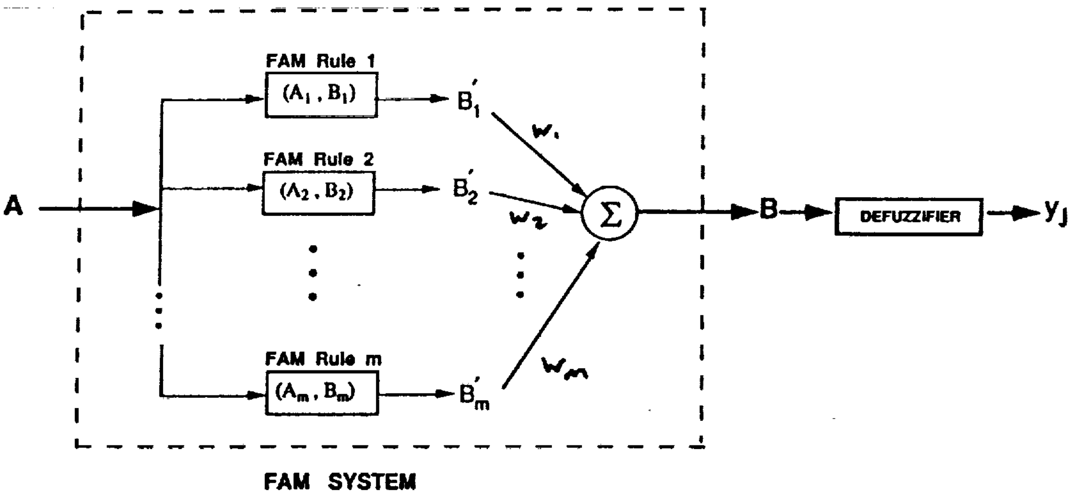

FIGURE 17.3 FAM system architecture. The FAM system F maps fuzzy sets in the unit cube I^n to fuzzy sets in the unit cube I^p. Binary input fuzzy sets are often used in practice to model exact input data. In general only an uncertainty estimate of the system state is available. So A is a proper fuzzy set. The user can defuzzify output fuzzy set B to yield exact output data, reducing the FAM system to a mapping between Boolean cubes.

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 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. The extension to multi-antecedent FAM rules is easier applied than formally explained. In the next section we present a general explanation for dealing with multi-antecedent FAM rules. First, though, we present the BIOFAM algorithm by illustrating it, and the FAM construction procedure, on an archetypical control problem.

Consider an inverted pendulum. In particular, consider how to adjust a motor to balance an inverted pendulum in two dimensions. The inverted pendulum is a classical control problem. It admits a math-model control solution. This provides a formal benchmark for BIOFAM pendulum controllers.

There are two state variables and one control variable. The first state 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 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}:

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

The control variable is the motor current or *angular velocity* v_t. The velocity can also 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 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 sets. We know that the system variables can be positive, zero, or negative. We can quantize the magnitudes of the system variables finely or coarsely. Suppose we quantize the magnitudes as small, medium, and large. This leads to seven linguistic 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 common-sense 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 be a Gaussian curve for the pendulum angle θ, a triangle for the angular velocity $\Delta\theta$, and a trapezoid for the velocity v. But all the ZE fuzzy sets will be centered 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 overshoot and undershoot. In Chapter 19 we determine experimentally the following default heuristic for ideal overlap: *Contiguous fuzzy sets in a library should overlap approximately* 25%.

FAM rules are triples, such as $(NM, Z;\ PM)$. 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 $(NM, Z;\ PM)$ 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 three-antecedent FAM rule:

$$\text{IF} \quad \theta = NM \quad \text{AND} \quad \Delta\theta = ZE ,$$

$$\text{THEN} \quad v = PM .$$

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 a working system to be developed in hours, if not 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.

This is typical of many control problems that require nulling a scalar error measure. We can control 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 19 shows how error nulling can control a realtime target tracking system.) Error nulling tractably 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 is one of seven motor-velocity fuzzy-set values. Since a FAM rule is a mapping or function, there is exactly one output velocity value for every pair of angle and angular-velocity values. So the 49 entries in the FAM bank matrix represent the 49 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.

Commonsense dictates the entries in the pendulum FAM bank matrix. Suppose the pendulum is not changing. So $\Delta\theta = ZE$. If the pendulum is to the right of vertical, the motor velocity should be negative to compensate. The farther the pendulum is to the right, the larger the negative motor velocity should be. The motor velocity should be positive if the pendulum is to the left. So the fourth row of the FAM bank matrix, which corresponds to $\Delta\theta = ZE$, should be 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 is moving. 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 $\theta = ZE$, should be 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 velocity values, since the pendulum is heading 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 velocity values. So $(NS, PS;\ PS)$ is another candidate FAM rule.

This gives 15 FAM rules altogether. In practice these rules are more than sufficient to successfully balance an inverted pendulum. Different, and smaller, subsets of FAM rules may also successfully balance the pendulum.

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

θ

$\Delta\theta$ \ 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 also 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 18 and 19.

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

Consider first how the input data pair (15, -10) activates 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 17.4. Then the angle datum 15 is a zero angle value to degree .2 : $m^{\theta}_{ZE}(15) = .2$. The angular velocity datum -10 is a zero angular velocity value to degree .5 : $m_{ZE}^{\Delta\theta}(-10) = .5$.

We combine the antecedent fit values with minimum or maximum according as the antecedent fuzzy sets are combined 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.

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

$$\begin{aligned} \min(m_{ZE}^{\theta}(15),\ m_{ZE}^{\Delta\theta}(-10)) &= \min(.2,\ .5) \\ &= .2 \quad . \end{aligned}$$

Clearly 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 velocity values to degree .2. This is not the same as activating the ZE motor velocity fuzzy set 100% with probability .2, and certainly not the same as $\text{Prob}\{v = 0\} = .2$. Instead a deterministic 20% of ZE should result and, according to the additive combination formula (17), should be added 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_{ZE}^{v}:

$$\min(m_{ZE}^{\theta}(15),\ m_{ZE}^{\Delta\theta}(-10)) \wedge m_{ZE}^{v}(v) \quad = \quad .2 \wedge m_{ZE}^{v}(v)$$

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

The data similarly activate the FAM rule $(PS, ZE;\ NS)$ depicted in Figure 17.4. 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 the output motor velocity fuzzy set of small but negative motor velocity values is scaled by .5, the lesser of the two antecedent fit values:

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

for all velocity values v. So the data activate the FAM rule $(PS, ZE;\ NS)$ to greater degree than 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 (17) by summing pointwise. We can then compute the fuzzy centroid with equation (19), with perhaps integrals replacing the discrete sums, to determine the specific output motor velocity v. In Chapter 19 we show that, for symmetric fuzzy sets of quantization, the centroid can always be computed exactly with simple discrete sums even if the fuzzy sets are continuous. In many realtime applications we must repeat this entire FAM inference procedure hundreds, perhaps thousands, of times per second. This requires fuzzy VLSI or optical processors.

Figure 17.4 illustrates this equal-weight additive combination procedure for just the FAM rules $(ZE, ZE;\ ZE)$ and $(PS, ZE;\ NS)$. The fuzzy-centroidal motor velocity value in this case is -3.

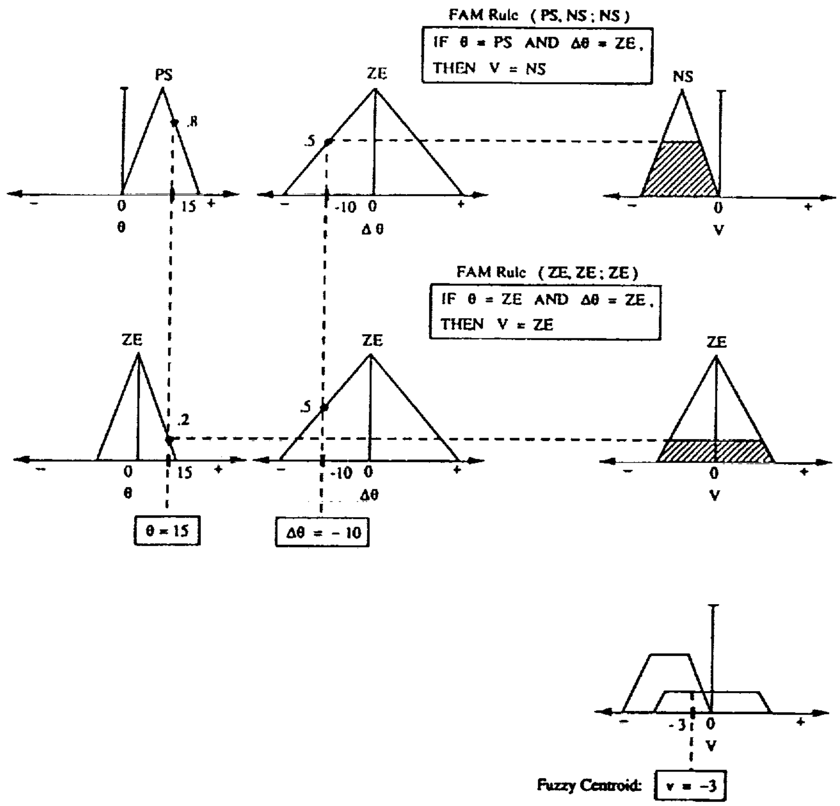

FIGURE 17.4 FAM correlation-minimum inference procedure. The FAM system consists of the two two-antecedent FAM rules $(PS, ZE;\ NS)$ and $(ZE, ZE;\ ZE)$. The input angle datum is 15, and is more a small but positive angle value than a zero angle value. The input angular velocity datum is -10, and is only a zero angular velocity value to degree .5. Antecedent fit values are combined with minimum since the antecedent terms are combined conjunctively with AND. The combined fit value then scales the consequent fuzzy set with pairwise minimum. The minimum-scaled output fuzzy sets are added pointwise. The fuzzy centroid of this output waveform is computed and yields the system output velocity value -3.

Multi-Antecedent FAM Rules: Decompositional Inference

The BIOFAM inference procedure treats antecedent fuzzy sets as if they were propositions with fuzzy truth values. This is because fuzzy logic corresponds to 1-dimensional fuzzy set theory and because we use binary or exact inputs. We now formally develop the connection between BIOFAMs and the FAM theory presented earlier.

Consider the compound FAM rule "IF X is A AND Y is B , THEN C is Z," or $(A, B;\ C)$ for short. Let the universes of discourse X, Y, and Z have dimensions n, p, and q: $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_p\}$, and $Z = \{z_1, \ldots, z_q\}$. We can directly extend this framework to multiple antecedent and consequent terms.

In our notation X, Y, and Z are both universes of discourse and fuzzy variables. The fuzzy *variable* X can assume the fuzzy set *values* $A_1, A_2, \ldots$, and similarly for the fuzzy variables Y and Z. When controlling an inverted pendulum, the identification "X is A" might represent the natural-language description "The pendulum angle is positive but small."

What is the matrix representation of the FAM rule $(A, B;\ C)$? The question is nontrivial since A, B, and C are fuzzy subsets of different universes of discourse, points in different unit cubes. Their dimensions and interpretations differ. Mamdani [1977] and others have suggested representing such rules as fuzzy multidimensional relations or arrays. Then the FAM rule $(A, B;\ C)$ would be a fuzzy subset of the product space $X \times Y \times Z$. This representation is not used in practice since only exact inputs are presented to FAM systems and the BIOFAM procedure applies. If we presented the system with a genuine fuzzy set input, we would no doubt preprocess the fuzzy set with a centroidal or maximum-fit-value technique so we could still apply the BIOFAM inference procedure.

We present an alternative representation that decomposes, then recomposes, the FAM rule $(A, B;\ C)$ in accord with the FAM inference procedure. This representation allows neural networks to adaptively estimate, store, and modify the decomposed FAM rules. The representation requires far less storage than the multidimensional-array representation.

Let the fuzzy Hebb matrices M_{AC} and M_{BC} store the simple FAM associations (A, C) and (B, C):

$$M_{AC} = A^T \circ C \ , \tag{20}$$

$$M_{BC} = B^T \circ C \ . \tag{21}$$

The fuzzy Hebb matrices M_{AC} and M_{BC} *split* the compound FAM rule $(A, B;\ C)$. We can construct the splitting matrices with correlation-product encoding.

Let $I_X^i = (0 \ldots 0\ 1\ 0 \ldots 0)$ be an n-dimensional bit vector with ith element 1 and all other elements 0. I_X^i is the ith row of the n-by-n identity matrix. Similarly, I_Y^j and I_Z^k are the respective jth and kth rows of the p-by-p and q-by-q identity matrices. The bit vector I_X^i represents the occurrence of the exact input x_i.

We will call the proposed FAM representation scheme **FAM decompositional inference**, in the spirit of the max-min compositional inference scheme discussed above. FAM decompositional inference ***decomposes*** the compound FAM rule $(A, B;\ C)$ into the component rules (A, C) and (B, C). The simpler component rules are processed in parallel. New fuzzy set inputs A' and B' pass through the FAM matrices M_{AC} and M_{BC}. Max-min composition then gives the recalled fuzzy sets $C_{A'}$ and $C_{B'}$:

$$C_{A'} = A' \circ M_{AC} \quad , \tag{22}$$

$$C_{B'} = B' \circ M_{BC} \quad . \tag{23}$$

The trick is to *recompose* the fuzzy sets $C_{A'}$ and $C_{B'}$ with intersection or union according as the antecedent terms "X is A" and "Y is B" are combined with AND or OR. The negated antecedent term "X is NOT A" requires forming the set complement $C_{A'}^c$ for input fuzzy set A'.

Suppose we present the new inputs A' and B' to the single-FAM-rule system F that stores the FAM rule $(A, B;\ C)$. Then the recalled output fuzzy set C' equals the intersection of $C_{A'}$ and $C_{B'}$:

$$\begin{aligned} F(A',\ B') &= [A' \circ M_{AC}] \cap [B' \circ M_{BC}] \\ &= C_{A'} \cap C_{B'} \\ &= C' \quad . \end{aligned} \tag{24}$$

We can then defuzzify C', if we wish, to yield the exact output I_Z^k.

The logical connectives apply to the antecedent terms of different dimension and meaning. Decompositional inference applies the set-theoretic analogues of the logical connectives to subsets of Z. Of course all subsets C' of Z have the same dimension and meaning.

We now prove that decompositional inference generalizes BIOFAM inference. This generalization is not simply formal. It opens an immediate path to adaptation with arbitrary neural network techniques.

Suppose we present the exact inputs x_i and y_j to the single-FAM-rule system F that stores $(A, B;\ C)$. So we present the unit bit vectors I_X^i and I_Y^j to F as nonfuzzy set inputs. Then

$$\begin{aligned} F(x_i, y_j) \;=\; F(I_X^i,\ I_Y^j) &= [I_X^i \circ M_{AC}] \;\cap\; [I_Y^j \circ M_{BC}] \\ &= a_i \wedge C \;\cap\; b_j \wedge C \qquad (25) \\ &= \min(a_i,\ b_j) \wedge C \quad . \qquad (26) \end{aligned}$$

(25) follows from (8). Representing C with its membership function m_C, (26) is equivalent to the BIOFAM prescription

$$\min(a_i,\ b_j) \wedge m_C(z) \qquad (27)$$

for all z in Z.

If we encode the simple FAM rules (A, C) and (B, C) with correlation-product encoding, decompositional inference gives the BIOFAM version of correlation-product inference:

$$\begin{aligned} F(I_X^i, I_Y^j) &= [I_X^i \circ A^T C] \;\cap\; [I_Y^j \circ B^T C] \\ &= a_i\, C \;\cap\; b_j\, C \qquad (28) \\ &= \min(a_i,\ b_j)\, C \qquad (29) \\ &= \min(a_i,\ b_j)\, m_C(z) \qquad (30) \end{aligned}$$

for all z in Z. (13) implies (28). $\min(a_i\ c_k,\ b_j\ c_k) = \min(a_i,\ b_j)\ c_k$ implies (29).

Decompositional inference allows arbitrary fuzzy sets, waveforms, or distributions A' and B' to be applied to a FAM system. The FAM system can house an arbitrary FAM bank of compound FAM rules. If we use the FAM system to control a process, the input fuzzy sets A' and B' can be the output of an independent state-*estimation* system, such as a Kalman filter. A' and B' might then represent probability distributions on the exact input spaces X and Y. The filter-controller cascade is a common engineering architecture.

We can split compound consequents as desired. We can split the compound FAM rule "IF X is A AND Y is B, THEN Z is C OR W is D," or $(A, B;\ C, D)$, into the FAM rules $(A, B;\ C)$ and $(A, B;\ D)$. We can use the same split if the consequent logical connective is AND.

We can give a propositional-calculus justification for the decompositional inference technique. Let A, B, and C be bivalent propositions with truth values $t(A)$, $t(B)$, and $t(C)$ in $\{0, 1\}$. Then we can construct truth tables to prove the two consequent-splitting tautologies that we use in decompositional inference:

$$[A \longrightarrow (B \text{ OR } C)] \longrightarrow [(A \longrightarrow B) \text{ OR } (A \longrightarrow C)] \ , \tag{31}$$

$$[A \longrightarrow (B \text{ AND } C)] \longrightarrow [(A \longrightarrow B) \text{ AND } (A \longrightarrow C)] \ , \tag{32}$$

where the arrow represents logical implication.

In bivalent logic, the implication $A \rightarrow B$ is false iff the antecedent A is true and the consequent B is false. Equivalently, $t(A \rightarrow B) = 1$ iff $t(A) = 1$ and $t(B) = 0$. This allows a "brief" truth table to be constructed to check for validity. We chose truth values for the terms in the consequent of the overall implication (31) or (32) to make the consequent false. Given those restrictions, if we cannot find truth values to make the antecedent true, the statement is a tautology. In (31), if $t((A \rightarrow B) \text{ OR } (A \rightarrow C)) = 0$, then $t(A) = 1$ and $t(B) = t(C) = 0$, since a disjunction is false iff both disjuncts are false. This forces the antecedent $A \rightarrow (B \text{ OR } C)$ to be false. So (31) is a **tautology**: It is true in all cases.

We can also justify splitting the compound FAM rule "IF X is A OR Y is B, THEN Z is C" into the disjunction (union) of the two simple FAM rules "IF X is A, THEN Z is C " and "IF Y is B , THEN Z is C " with a propositional tautology:

$$[(A \text{ OR } B) \longrightarrow C] \longrightarrow [(A \longrightarrow C) \text{ OR } (B \longrightarrow C)] \ . \tag{33}$$

Now consider splitting the original compound FAM rule "IF X is A AND Y is B , THEN Z is C " into the conjunction (intersection) of the two simple FAM rules "IF X is A , THEN Z is C " and "IF Y is B , THEN Z is C ." A problem arises when we examine the truth table of the corresponding proposition

$$[(A \text{ AND } B) \longrightarrow C] \longrightarrow [(A \longrightarrow C) \text{ AND } (B \longrightarrow C)] \ . \tag{34}$$

The problem is that (34) is not always true, and hence not a tautology. The implication is false if A is true and B and C are false, or if A and C are false and B is true. But the implication (34) is valid if *both* antecedent terms A and B are true. So if $t(A) = t(B) = 1$, the compound conditional $(A \text{ AND } B) \rightarrow C$ implies both $A \rightarrow C$ and $B \rightarrow C$.

The simultaneous occurrence of the data values x_i and y_j satisfies this condition. Recall that logic is 1-dimensional set theory. The condition $t(A) = t(B) = 1$ is given by the 1 in I_X^i and the 1 in I_X^j. We can interpret the unit bit vectors I_X^i and I_Y^j as the (true) bivalent propositions "X is x_i" and "Y is y_j." Propositional logic applies coordinate-wise. A similar argument holds for the converse of (33).

For general fuzzy set inputs A' and B' the argument still holds in the sense of continuous-valued logic. But the truth values of the logical implications may be less than unity while greater than zero. If A' is a null vector and B' is not, or vice versa, the implication (34) is false coordinate-wise, at least if one coordinate of the non-null vector is unity. But in this case the decompositional inference scheme yields an output null vector C'. In effect the FAM system indicates the propositional falsehood.

Adaptive Decompositional Inference

The decompositional inference scheme allows the *splitting matrices* M_{AC} and M_{BC} to be arbitrary. Indeed it allows them to be eliminated altogether.

Let $N_X: \quad I^n \rightarrow I^q$ be an arbitrary *neural network* system that maps fuzzy subsets A' of X to fuzzy subsets C' of Z. $N_Y: \quad I^p \rightarrow I^q$ can be a different neural network. In general N_X and N_Y are time-varying.

The adaptive decompositional inference (ADI) scheme allows compound FAM rules to be adaptively split, stored, and modified by arbitrary neural networks. The compound FAM rule "IF X is A AND Y is B, THEN Z is C," or $(A, B;\ C)$, can be split by N_X and N_Y. N_X can house the simple FAM association (A, C). N_Y can house (B, C). Then for arbitrary fuzzy set inputs A' and B', ADI proceeds as before for an adaptive FAM system $F: \quad I^n \times I^p \rightarrow I^q$ that houses the FAM rule $(A, B;\ C)$ or a bank of such FAM rules:

$$\begin{aligned} F(A', B') &= N_X(A') \cap N_Y(B') \\ &= C_{A'} \cap C_{B'} \\ &= C' \ . \end{aligned} \tag{35}$$

Any neural network technique can be used. A reasonable candidate for many unstructured problems is the backpropagation algorithm applied to several small feedforward multilayer networks. The primary concerns are space and training time. Several small neural networks can often be trained in parallel faster, and more accurately, than a single large neural network.

The ADI approach illustrates one way neural algorithms can be embedded in a FAM architecture. Below we discuss another way that uses unsupervised clustering algorithms.

ADAPTIVE FAMs: PRODUCT-SPACE CLUSTERING IN FAM CELLS

An **adaptive FAM (AFAM)** is a time-varying mapping between fuzzy cubes. In principle the adaptive decompositional inference technique generates AFAMs. But we shall reserve the label AFAM for systems that generate FAM rules from training data but that do not require splitting and recombining FAM data.

We propose a geometric AFAM procedure. The procedure adaptively clusters training samples in the FAM system *input-output product space.* FAM mappings are balls or clusters in the input-output product space. These clusters are simply the fuzzy Hebb matrices discussed above. The procedure "blindly" generates weighted FAM rules from training data. Further training modifies the weighted set of FAM rules. We call this unsupervised procedure **product-space clustering**.

Consider first a discrete 1-dimensional FAM system $S: \; I^n \rightarrow I^p$. Then a FAM rule has the form "IF X is A_i , THEN Y is B_i " or $(A_i, \; B_i)$. The input-output product space is $I^n \times I^p$.

What does the FAM rule $(A_i, \; B_i)$ look like in the product space $I^n \times I^p$? It looks like a cluster of points centered at the numerical point $(A_i, \; B_i)$. The FAM system maps points A near A_i to points B near B_i. The closer A is to A_i, the closer the point $(A, \; B)$ is to the point $(A_i, \; B_i)$ in the product space $I^n \times I^p$. In this sense FAMs map balls in I^n to balls in I^p. The notation is ambiguous since $(A_i, \; B_i)$ stands for both the FAM rule mapping, or fuzzy subset of $I^n \times I^p$, and the numerical fit-vector point in $I^n \times I^p$.

Adaptive clustering algorithms can estimate the unknown FAM rule $(A_i, \; B_i)$ from training samples of the form $(A, \; B)$. In general there are m unknown FAM rules $(A_1, \; B_1), \ldots,$ $(A_m, \; B_m)$. The number m of FAM rules is also unknown. The user may select m arbitrarily in many applications.

Competitive *adaptive vector quantization* (AVQ) algorithms can adaptively estimate both the unknown FAM rules (A_i, B_i) and the unknown number m of FAM rules from FAM system input-output data. The AVQ algorithms do not require fuzzy-set data. Scalar BIOFAM data suffices, as we illustrate below for adaptive estimation of inverted-pendulum control FAM rules.

Suppose the r fuzzy sets $A_1, \; \ldots, \; A_r$ quantize the input universe of discourse X. The s fuzzy sets $B_1, \; \ldots, \; B_s$ quantize the output universe of discourse Y. In general r and s are unrelated to each other and to the number m of FAM rules $(A_i, \; B_i)$. The user must specify r and s and the shape of the fuzzy sets A_i and B_i. In practice this is not difficult.

Quantizing fuzzy sets are usually trapezoidal, and r and s are less than 10.

The quantizing collections $\{A_i\}$ and $\{B_j\}$ define rs FAM cells F_{ij} in the input-output product space $I^n \times I^p$. The FAM cells F_{ij} overlap since contiguous quantizing fuzzy sets A_i and A_{i+1}, and B_j and B_{j+1}, overlap. So the FAM cell collection $\{F_{ij}\}$ does not partition the product space $I^n \times I^p$. The union of all FAM cells also does not equal $I^n \times I^p$ since the patches F_{ij} are fuzzy subsets of $I^n \times I^p$. The union provides only a fuzzy "cover" for $I^n \times I^p$.

The *fuzzy Cartesian product* $A_i \times B_i$ defines the FAM cell F_{ij}. $A_i \times B_i$ is just the fuzzy outer product $A_i^T \circ B_i$ in (6) or the correlation product $A_i^T\ B_i$ in (12). So a FAM cell F_{ij} is simply the fuzzy correlation-minimum or correlation-product matrix M_{ij} : $F_{ij} = M_{ij}$.

Adaptive FAM Rule Generation

Let $\mathbf{m}_1, \ldots, \mathbf{m}_k$ be k quantization vectors in the input-output product space $I^n \times I^p$ or, equivalently, in I^{n+p}. $\mathbf{m}_j$ is the jth column of the synaptic connection matrix $\mathbf{M}$. $\mathbf{M}$ has $n + p$ rows and k columns.

Suppose, for instance, $\mathbf{m}_j$ changes in time according to the differential competitive learning (DCL) AVQ algorithm discussed in Chapters 6 and 9. The competitive system samples concatenated fuzzy set samples of the form $[A|B]$. The augmented fuzzy set $[A|B]$ is a point in the unit hypercube I^{n+p}.

The synaptic vectors $\mathbf{m}_j$ converge to FAM matrix centroids in $I^n \times I^p$. More generally they estimate the density or distribution of the FAM rules in $I^n \times I^p$. The quantizing synaptic vectors naturally weight the estimated FAM rule. The more synaptic vectors clustered about a centroidal FAM rule, the greater its weight w_i in (17).

Suppose there are 15 FAM-rule centroids in $I^n \times I^p$ and $k > 15$. Suppose k_i synaptic vectors $\mathbf{m}_j$ cluster around the ith centroid. So $k_1 + \ldots + k_{15} = k$. Suppose the *cluster counts* k_i are ordered as

$$k_1 \geq k_2 \geq \ldots k_{15} \quad . \tag{36}$$

The first centroidal FAM rule is as at least as frequent as the second centroidal FAM rule, and so on. This gives the adaptive FAM-rule weighting scheme

$$w_i = \frac{k_i}{k} \quad . \tag{37}$$

The FAM rule weights w_i evolve in time as new augmented fuzzy sets $[A|B]$ are sampled. In practice we may want only the 15 most-frequent FAM rules or only the FAM rules with at least some minimum frequency w_{min}. Then (37) provides a quantitative solution.

Geometrically we count the number k_{ij} of quantizing vectors in each FAM cell F_{ij}. We can define FAM-cell boundaries in advance. High-count FAM cells outrank low-count FAM cells. Most FAM cells contain zero or few synaptic vectors.

Product-space clustering extends to compound FAM rules and product spaces. The FAM rule "IF X is A AND Y is B, THEN Z is C", or $(A,\ B;\ C)$, is a point in $I^n \times I^p \times I^q$. The t fuzzy sets $C_1, \ldots, C_t$ quantize the new output space Z. There are rst FAM cells F_{ijk}. (36) and (37) extend similarly. X, Y, and Z can be continuous. The adaptive clustering procedure extends to any number of FAM-rule antecedent terms.

Adaptive BIOFAM Clustering

BIOFAM data clusters more efficiently than fuzzy-set FAM data. Paired numbers are easier to process and obtain than paired fit vectors. This allows system input-output data to directly generate FAM systems.

In control applications, human or automatic controllers generate streams of "well-controlled" system input-output data. Adaptive BIOFAM clustering converts this data to weighted FAM rules. The adaptive system transduces behavioral data to behavioral rules. The fuzzy system learns causal patterns. It learns which control inputs cause which control outputs. The system approximates these causal patterns when it acts as the controller.

Adaptive BIOFAMs cluster in the input-output product space $X \times Y$. The product space $X \times Y$ is vastly smaller than the power-set product space $I^n \times I^p$ used above. The adaptive synaptic vectors $\mathbf{m}_j$ are now 2-dimensional instead of $n + p$-dimensional. On the other hand, competitive BIOFAM clustering requires many more input-output data pairs $(x_i, y_i) \in R^2$ than augmented fuzzy-set samples $[A|B] \in I^{n+p}$.

Again our notation is ambiguous. We now use x_i as the numerical sample from X at sample time i. Earlier x_i denoted the ith ordered element in the finite nonfuzzy set $X = \{x_1, \ldots, x_n\}$. One advantage is X can be continuous, say R^n.

BIOFAM clustering counts synaptic quantization vectors in FAM cells. The system samples the nonfuzzy input-output stream $(x_1, y_1), (x_2, y_2), \ldots$ Unsupervised competitive learning distributes the k synaptic quantization vectors $\mathbf{m}_1, \ldots, \mathbf{m}_k$ in $X \times Y$. Learning distributes them to different FAM cells F_{ij}. The FAM cells F_{ij} overlap but are nonfuzzy subcubes of $X \times Y$. The BIOFAM FAM cells F_{ij} cover $X \times Y$.

F_{ij} contains k_{ij} quantization vectors at each sample time. The cell counts k_{ij} define a frequency *histogram* since all k_{ij} sum to k. So $w_{ij} = \frac{k_{ij}}{k}$ weights the FAM rule "IF X is A_i, THEN Y is B_j."

Suppose the pairwise-overlapping fuzzy sets NL, NM, NS, ZE, PS, PM, PL quantize the input space X. Suppose seven similar fuzzy sets quantize the output space Y. We can define the fuzzy sets arbitrarily. In practice they are normal and trapezoidal. (The boundary fuzzy sets NL and PL are ramp functions.) X and Y may each be the real line. A typical FAM rule is "IF X is NL, THEN Y is PS."

Input datum x_i is nonfuzzy. When $X = x_i$ holds, the relations $X = NL, \ldots, X = PL$ hold to different degrees. Most hold to degree zero. $X = NM$ holds to degree $m_{NM}(x_i)$. Input datum x_i partially activates the FAM rule "IF X is NM, THEN Y is ZE" or, equivalently, $(NM; ZE)$. Since the FAM rules have single antecedents, x_i activates the consequent fuzzy set ZE to degree $m_{NM}(x_i)$ as well. Multi-antecedent FAM rules activate output consequent sets according to a logic-based function of antecedent term membership values, as discussed above on BIOFAM inference.

Suppose Figure 17.5 represents the input-output data stream $(x_1, y_1), (x_2, y_2), \ldots$ in the planar product space $X \times Y$:

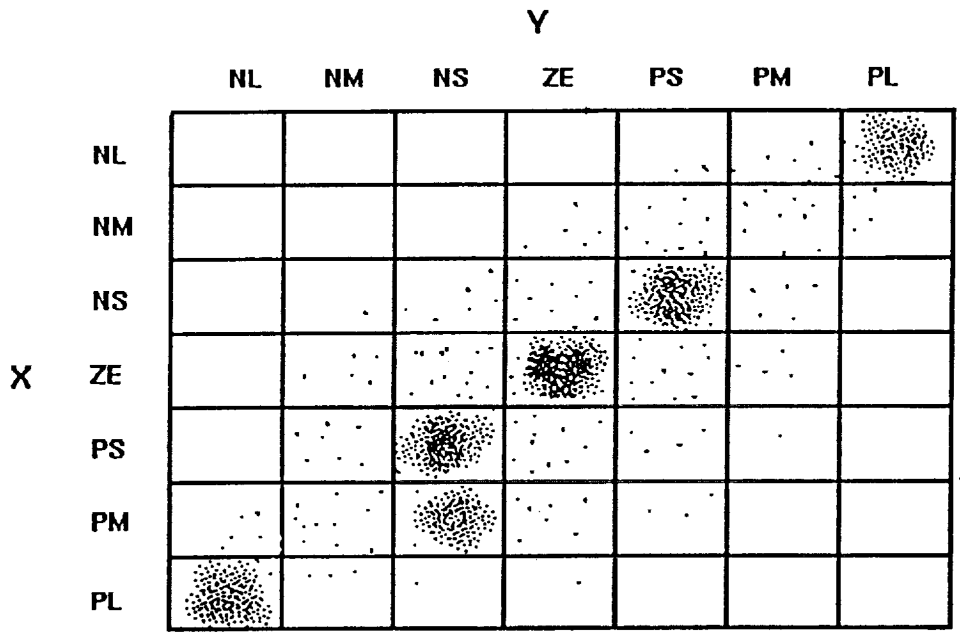

FIGURE 17.5 Distribution of input-output data (x_i, y_i) in the input-output product space $X \times Y$. Data clusters reflect FAM rules, such as the steady-state FAM rule "IF X is ZE, THEN Y is ZE".

Suppose the sample data in Figure 17.5 trains a DCL system. Suppose such competitive learning distributes ten 2-dimensional synaptic vectors $\mathbf{m}_1, \ldots, \mathbf{m}_{10}$ as in Figure 17.6:

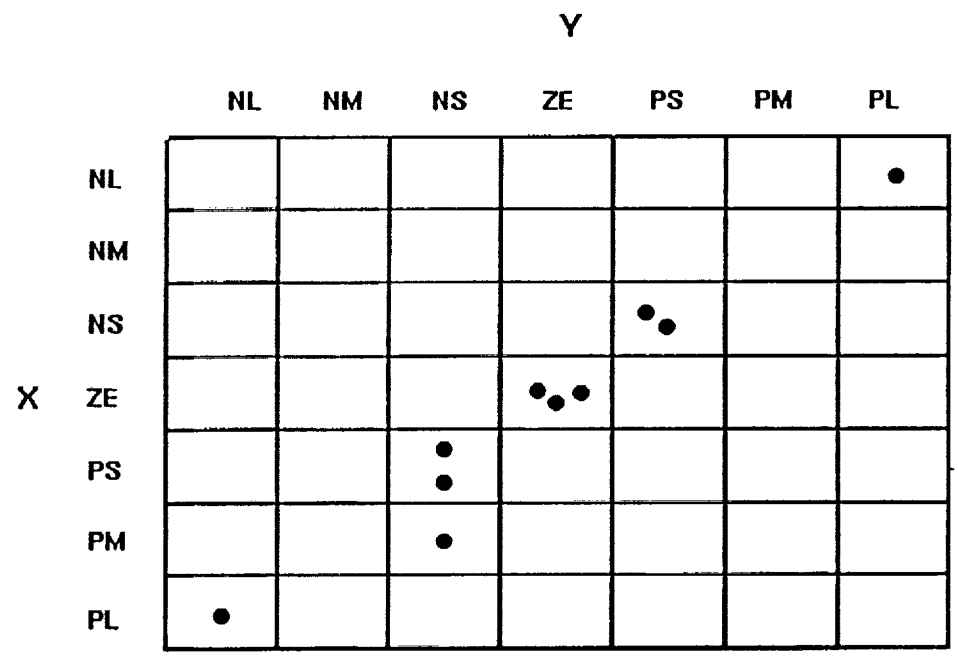

FIGURE 17.6 Distribution of ten 2-dimensional synaptic quantization vectors $\mathbf{m}_1, \ldots, \mathbf{m}_{10}$ in the input-output product space $X \times Y$. As the FAM system samples nonfuzzy data (x_i, y_i), competitive learning distributes the synaptic vectors in $X \times Y$. The synaptic vectors estimate the frequency distribution of the sampled input-output data, and thus estimate FAM rules.

FAM cells do not overlap in Figures 17.5 and 17.6 for convenience's sake. The corresponding quantizing fuzzy sets touch but do not overlap.

Figure 17.5 reveals six sample-data clusters. The six quantization-vector clusters in Figure 17.6 estimate the six sample-data clusters. The single synaptic vector in FAM cell $(PM;\ NS)$ indicates a smaller cluster. Since $k = 10$, the number of quantization vectors in each FAM cell measures the percentage or frequency weight w_{ij} of each possible FAM rule.

In general the additive combination rule (17) does not require normalizing the quantization-vector count k_{ij}. $w_{ij} = k_{ij}$ is acceptable. This holds for both maximum-membership defuzzification (18) and fuzzy centroid defuzzification (19). These defuzzification schemes prohibit only negative weight values.

The ten quantization vectors in Figure 17.6 estimate at most six FAM rules. From most to least frequent or "important", the FAM rules are $(ZE;\ ZE)$, $(PS;\ NS)$, $(NS;\ PS)$, $(PM;\ NS)$, $(PL;\ NL)$, and $(NL;\ PL)$. These FAM rules suggest that fuzzy variable X is an error variable or an error velocity variable since the steady-state FAM rule $(ZE;\ ZE)$ is most important. If we sample a system only in steady-state equilibrium, we will estimate only the steady-state FAM rule. We can accurately estimate the FAM system's global behavior only if we representatively sample the system's input-output behavior.

The "corner" FAM rules $(PL;\ NL)$ and $(NL;\ PL)$ may be more important than their frequencies suggest. The boundary sets Negative Large (NL) and Positive Large (PL) are usually defined as ramp functions, as negatively and positively sloped lines. NL and PL alone cover the important end-point regions of the universe of discourse X. They give $m_{NL}(x) = m_{PL}(x) = 1$ only if x is at or near the end-point of X, since NL and PL are ramp functions not trapezoids. NL and PL cover these end-point regions "briefly". Their corresponding FAM cells tend to be smaller than the other FAM cells. The end-point regions must be covered in most control problems, especially error nulling problems like stabilizing an inverted pendulum. The user can weight these FAM-cell counts more highly, for instance $w_{ij} = c\ k_{ij}$ for scaling constant $c > 0$. Or the user can simply include these end-point FAM rules in every operative FAM bank.

Most FAM cells do not generate FAM rules. More accurately, we estimate every possible FAM rule but usually with zero or near-zero frequency weight w_{ij}. For large numbers of multiple FAM-rule antecedents, system input-output data streams through comparatively few FAM cells. Structured trajectories in $X \times Y$ are few.

A FAM-rule's mapping structure also limits the number of estimated FAM rules. A FAM rule maps fuzzy sets in I^n or $F(2^X)$ to fuzzy sets in I^p or $F(2^Y)$. A fuzzy associative memory maps every domain fuzzy set A to a unique range fuzzy set B. Fuzzy set A cannot map to multiple fuzzy sets B, B', B'', and so on. We write the FAM rule as $(A;\ B)$ not

(A; B or B' or B'' or ...). So we estimate *at most* one rule per FAM-cell row in Figure 17.6.

If two FAM cells in a row are equally and highly frequent, we can pick arbitrarily either FAM rule to include in the FAM bank. This occurs infrequently but can occur. In principle we could estimate the FAM rule as a compound FAM rule with a disjunctive consequent. The simplest strategy picks only the highest frequency FAM cell per row.

The user can estimate FAM rules without counting the quantization vectors in each FAM cell. There may be too many FAM cells to search at each estimation iteration. The user never need examine FAM cells. Instead the user checks the synaptic vector components m_{ij}. The user defines in advance fuzzy-set intervals, such as $[l_{NL}, u_{NL}]$ for NL. If $l_{NL} \leq m_{ij} \leq u_{NL}$, then the FAM-antecedent reads "IF X is NL."

Suppose the input and output spaces X and Y are the same, the real interval $[-35, 35]$. Suppose we partition X and Y into the same seven disjoint fuzzy sets:

$$\begin{aligned}
\text{NL} &= [-35, -25] \\
\text{NM} &= [-25, -15] \\
\text{NS} &= [-15, -5] \\
\text{ZE} &= [-5, 5] \\
\text{PS} &= [5, 15] \\
\text{PM} &= [15, 25] \\
\text{PL} &= [25, 35] \ .
\end{aligned}$$

Then the observed synaptic vector $\mathbf{m}_j = [9, -10]$ increases the count of FAM cell $PS \times NS$ and increases the weight of FAM rule "IF X is PS, THEN Y is NS."

This amounts to nearest-neighbor classification of synaptic quantization vectors. We assign quantization vector $\mathbf{m}_k$ to FAM cell F_{ij} iff $\mathbf{m}_k$ is closer to the centroid of F_{ij} than to all other FAM-cell centroids. We break ties arbitrarily. Centroid classification allows the FAM cells to overlap.

Adaptive BIOFAM Example: Inverted Pendulum

We used DCL to train an AFAM to control the inverted pendulum discussed above. We used the accompanying C-software to generate 1,000 pendulum trajectory data. These product-space training vectors $(\theta,\ \Delta\theta,\ v)$ were points in R^3. Pendulum angle θ data ranged between -90 and 90. Pendulum angular veclocity $\Delta\theta$ data ranged from -150 to 150.

We defined FAM cells by uniformly partitioning the effective product space. Fuzzy variables could assume only the five fuzzy set values NM, NS, ZE, PS, and PM. So there were 125 possible FAM rules. For instance, the steady-state FAM rule took the form $(ZE,\ ZE;\ ZE)$ or, more completely, "IF $\theta = ZE$ AND $\Delta\theta = ZE$, THEN $v = ZE$."

A BIOFAM controlled the inverted pendulum. The BIOFAM restored the pendulum to equilibrium as we knocked it over to the right and to the left. (Function keys F9 and F10 knock the pendulum over to the left and to the right. Input-output sample data reads automatically to a training data file.) Eleven FAM rules described the BIOFAM controller. Figure 17.1 displays this FAM bank. Observe that the zero (ZE) row and column are ordinal inverses of the respective row and column indices.

$\Delta\theta$ \ θ	NM	NS	Z	PS	PM
NM			PM		
NS			PS	Z	
Z	PM	PS	Z	NS	NM
PS		Z	NS		
PM			NM		

FIGURE 17.7 Inverted-pendulum FAM bank used in simulation. This

BIOFAM generated 1,000 sample vectors of the form $(\theta,\ \Delta\theta,\ v)$.

We trained 125 3-dimensional synaptic quantization vectors with differential competitive learning, as discussed in Chapters 4,6, and 9. In principle the 125 synaptic vectors could describe a uniform distribution of product-space trajectory data. Then the 125 FAM cells would each contain one synaptic vector. Alternatively, if we used a vertically stabilized pendulum to generate the 1,000 training vectors, all 125 synaptic vectors would concentrate in the $(ZE,\ ZE;\ ZE)$ FAM cell. This would still be true if we only mildly perturbed the pendulum from vertical equilibrium.

DCL distributed the 125 synaptic vectors to 13 FAM cells. So we estimated 13 FAM rules. Some FAM cells contained more synaptic vectors than others. Figure 17.8 displays the synaptic-vector histogram after the DCL samples the 1,000 samples. Actually Figure 17.8 displays a truncated histogram. The horizontal axis should list all 125 FAM cells, all 125 FAM-rule weights w_k in (17). The missing 112 entries have zero synaptic-vector frequency.

Figure 17.8 gives a snapshot of the adaptive process. In practice, and in principle, successive data gradually modify the histogram. "Good" training samples should include a significant number of equilibrium samples. In Figure 17.8 the steady-state FAM cell $(ZE,\ ZE;\ ZE)$ is clearly the most frequent.

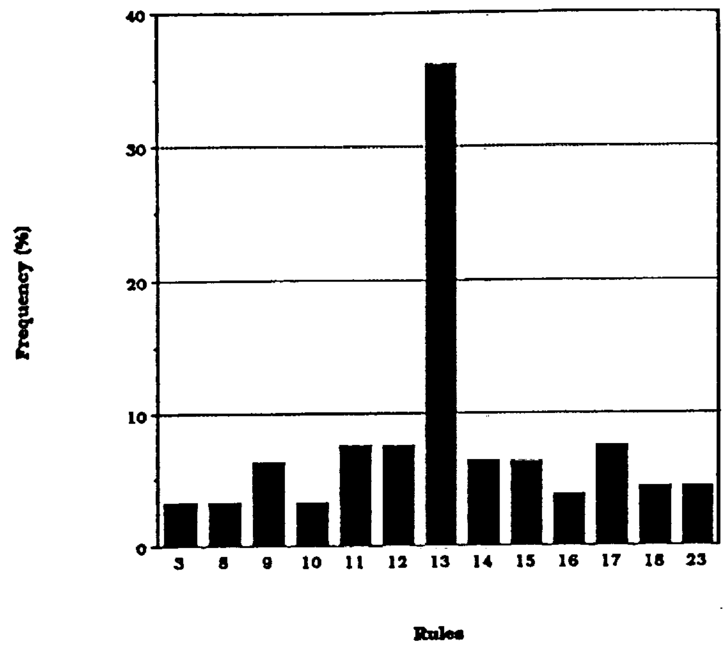

FIGURE 17.8 Synaptic-vector histogram. Differential competitive learning allocated 125 3-dimensional synaptic vectors to the 125 FAM cells. Here the adaptive system has sampled 1,000 representative pendulum-control data. DCL allocates the synaptic vectors to only 13 FAM cells. The steady-state FAM cell (ZE, ZE; ZE) is most frequent.

Figure 17.9 displays the DCL-estimated FAM bank. The product-space clustering method rapidly recovered the 11 original FAM rules. It also estimated the two additional FAM rules (PS, NM; ZE) and (NS, PM; ZE), which did not affect the BIOFAM system's performance. The estimated FAM bank defined a BIOFAM, with all 13 FAM-rule weights set w_k equal to unity, that controlled the pendulum as well as the original BIOFAM did.

$\Delta\theta$ \ θ	NM	NS	Z	PS	PM
NM			PM	Z	
NS			PS	Z	
Z	PM	PS	Z	NS	NM
PS		Z	NS		
PM		Z	NM		

FIGURE 17.9 DCL-estimated FAM bank. Product-space clustering recovered the original 11 FAM rules and estimated two new FAM rules. The new and original BIOFAM systems controlled the inverted pendulum equally well.

In nonrealtime applications we can in principle omit the adaptive step altogether. We can directly compute the FAM-cell histogram if we exhaustively count all sampled data. Then the (growing) number of synaptic vectors equals the number of training samples. This procedure equally weights all samples, and so tends not to "track" an evolving process. Competitive learning weights more recent samples more heavily. Competitive learning's metrical-classification step also helps filter noise from the stream of sample data.

REFERENCES

Dubois, D., Prade, H., *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.

Hebb, D. *The Organization of Behavior*, Wiley, 1949.

Klir, G.J., Foger, T.A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, 1988.

Kosko, B., "Fuzzy Knowledge Combination," *International Journal of Intelligent Systems*, vol. 1, 293 - 320, 1986.

Kosko, B., "Fuzzy Associative Memories," *Fuzzy Expert Systems*, A. Kandel (ed.), Addison-Wesley, in press, December 1986.

Kosko, B., "Fuzzy Entropy and Conditioning," *Information Sciences*, vol. 40, 165 - 174, 1986.

Kosko, B., *Foundations of Fuzzy Estimation Theory*, Ph.D. dissertation, Department of Electrical Engineering, University of California at Irvine, June 1987; Order Number 8801936, University of Microfilms International, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

Kosko, B., "Hidden Patterns in Combined and Adaptive Knowledge Networks," *International Journal of Approximate Reasoning*, vol. 2, no. 4, 377 - 393, October 1988.

Mamdani, E.H., "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," *IEEE Transactions on Computers*, vol. C-26, no. 12, 1182 - 1191, December 1977.

Taber, W.R., Siegel, M.A., "Estimation of Expert Weights Using Fuzzy Cognitive Maps," *Proceedings of the IEEE 1st International Conference on Neural Networks (ICNN-87)*, vol. II, 319 - 325, June 1987.

Taber, W.R., "Knowledge Processing with Fuzzy Cognitive Maps," *Expert Systems with Applications*, in press, 1990.

Togai, M., Watanabe, H., "Expert System on a Chip: An Engine for Realtime Approximate Reasoning, *IEEE Expert*, vol. 1, no. 3, 1986.

Yamakawa, T., "A Simple Fuzzy Computer Hardware System Employing MIN & MAX Operations," *Proceedings of the Second International Fuzzy Systems Association (IFSA)*, Tokyo, 827 - 830, July 1987.

Yamakawa, T., "Fuzzy Microprocessors–Rule Chip and Defuzzification Chip," *Proceedings of the International Workshop on Fuzzy Systems Applications*, Iizuka-88, Kyushu Institute of Technology, 51 - 52, August 1988.

Zadeh, L.A., "A Computational Approach to Fuzzy Quantifiers in Natural Languages," *Computers and Mathematics*, vol. 9, no. 1, 149 - 184, 1983.

PROBLEMS

1. Use correlation-minimum encoding to construct the FAM matrix M from the fit-vector pair $(A,\ B)$ if $A\ =\ (.6\ 1\ .2\ .9)$ and $B\ =\ (.8\ .3\ 1)$. Is (A, B) a bidirectional fixed point? Pass $A'\ =\ (.2\ .9\ .3\ .2)$ through M and $B'\ =\ (.9\ .5\ 1)$ through M^T. Do the recalled fuzzy sets differ from B and A?

2. Repeat Problem 1 using correlation-product encoding.

3. Compute the fuzzy entropy $E(M)$ of M in Problems 1 and 2.

4. If $M = A^T \circ B$ in Problem 1, find a different FAM matrix M' with greater fuzzy entropy, $E(M') > E(M)$, but that still gives perfect recall: $A \circ M' = B$. Find the *maximum entropy fuzzy associative memory* (MEFAM) matrix $\mathbf{M}^*$ such that $A \circ \mathbf{M}^* = B$.

5. Prove: If $M = A^T \circ B$ or $M = A^T B, A \circ M = B$, and $A \subset A'$, then $A' \circ M = B$.

6. Prove: $\max_{1 \le k \le m} \min(a_k, b_k) \le \min(\max_{1 \le k \le m} a_k, \max_{1 \le k \le m} b_k)$.

7. Use truth tables to prove the two-valued propositional tautologies:
   (a) [A $\longrightarrow$ (B OR C)] $\longrightarrow$ [(A $\longrightarrow$ B) OR (A $\longrightarrow$ C)] ,
   (b) [A $\longrightarrow$ (B AND C)] $\longrightarrow$ [(A $\longrightarrow$ B) AND (A $\longrightarrow$ C)] ,
   (c) [(A OR B)$\longrightarrow$ C] $\longrightarrow$ [(A $\longrightarrow$ C) OR (B $\longrightarrow$ C)] ,
   (d) [(A $\longrightarrow$ C) AND (B $\longrightarrow$ C)] $\longrightarrow$ [(A AND B) $\longrightarrow$ C] .

   Is the converse of (c) a tautology? Explain whether this affects BIOFAM inference.

8. BIOFAM inference. Suppose the input spaces X and Y are both $[-10, 10]$, and the output space Z is $[-100, 100]$. Define five trapezoidal fuzzy sets-NL, NS, ZE, PS, PL— on X, Y, and Z. Suppose the underlying (unknown) system transfer function is $z = x^2 - y^2$. State at least five FAM rules that accurately describe the system's

behavior. Use $z = x^2 - y^2$ to generate streams of sample data. Use BIOFAM inference and fuzzy-centroid defuzzification to map input pairs (x, y) to output data z. Plot the BIOFAM outputs and the desired outputs z. What is the arithmetic average of the squared errors $(F(x, y) - x^2 + y^2)^2$? Divide the product space $X \times Y \times Z$ into 125 overlapping FAM cells. Estimate FAM rules from clustered system data (x, y, z). Use these FAM rules to control the system. Evaluate the performance.

Software Problems

The following problems use the accompanying FAM software for controlling an inverted pendulum.

1. Explain why the pendulum stabilizes in the diagonal position if the pendulum bob mass increases to maximum and the motor current decreases slightly. The pendulum stabilizes in the vertical position if you remove which FAM rules?

2. Oscillation results if you remove which FAM rules? The pendulum sticks in a horizontal equilibrium if you remove which FAM rules?