MATHEMATICAL BIOSCIENCES 23, 151–177 (1975)

Fuzzy Neural Networks

SAMUEL C. LEE
Department of Electrical Engineering,
University of Houston, Houston, Texas

AND

EDWARD T. LEE
Department of Electrical Sciences, State University of New York
at Stony Brook, Stony Brook, New York

Communicated by Lawrence Stark

ABSTRACT

In this paper, the McCulloch-Pitts model of a neuron is extended to a more general model which allows the activity of a neuron to be a "fuzzy" rather than an "all-or-none" process. The generalized model is called a *fuzzy neuron*. Some basic properties of fuzzy neural networks as well as their applications to the synthesis of fuzzy automata are investigated. It is shown that any n-state minimal fuzzy automaton can be realized by a network of m fuzzy neurons, where $\lceil \log_2 n \rceil < m < 2n$. Examples are given to illustrate the procedure. As an example of application, a realization of fuzzy language recognizer using a fuzzy neural network is presented. The techniques described in this paper may be of use in the study of neural networks as well as in language, pattern recognition, and learning.

1. INTRODUCTION

In the nearly three decades since its publication, the pioneering work of McCulloch and Pitts [1], has had a profound influence on the development of the theory of neural nets, in addition to stimulating much of the early work in automata theory and regular events [2–12].

Although the McCulloch-Pitts model of a neuron has contributed a great deal to the understanding of the behavior of neural-like systems, it fails to reflect the fact that the behavior of even the simplest type of nerve cell exhibits not only randomness but, more importantly, a type of imprecision which is associated with the lack of sharp transition from the occurrence of an event to its nonoccurrence.

It is possible that a better model for the behavior of a nerve cell may be provided by what might be called a *fuzzy* neuron, which is a generalization of the McCulloch-Pitts model. The concept of a fuzzy neuron employs some of the concepts and techniques of the theory of fuzzy sets which was introduced by Zadeh [13, 14] and applied to the theory of automata by Wee and Fu [15], Tanaka et al. [16], Santos [17] and others. In effect, the introduction of fuzziness into the model of a neuron makes it better adapted to the study of the behavior of systems which are imprecisely defined by virtue of their high degree of complexity. Many of the biological systems, economic systems, urban systems, and, more generally, large-scale systems fall into this category.

In what follows, we shall present a preliminary account of a theory of fuzzy neural networks stressing its relations to automata and languages. Biogically oriented applications of this theory will be presented in subsequent papers.

2. FUZZY SETS

Essentially, a fuzzy set is a class with unsharp boundaries in which the transition from membership to nonmembership is gradual rather than abrupt. For example, the class of *tall man* is a fuzzy set, as are the classes labeled *middle-aged, small, red,* etc..

A fuzzy set A in a space X is defined by its *membership function* μ_A which associates with each point x in X its *grade of membership* $\mu_A(x)$ in A. $\mu_A(x)$ is usually assume to be a number in the interval [0, 1], with 0 and 1 representing nonmembership and full membership respectively.

If X is a finite set $X=\{x_1,\ldots,x_n\}$, and is convenient to represent a fuzzy set A in X as a linear combination

$$A=\mu_1 x_1+\cdots+\mu_n x_n,$$

where μ_i is the grade of membership of x_i in A. For example, if X is a group of four men named John, Tom, Dick, and Jim, and A is the fuzzy set of tall men in X, then we can write symbolically

$$X=\text{John}+\text{Tom}+\text{Dick}+\text{Jim},$$

and

$$A=0.8\,\text{John}+0.6\,\text{Tom}+0.9\,\text{Dick}+0.4\,\text{Jim}. \tag{1}$$

A basic attribute of a nonfuzzy set A is its cardinality, $|A|$, that is, the number of elements belonging to A. Since the notion of belonging loses its meaning when A is a fuzzy set, it is not meaningful to speak of the number of elements in a fuzzy set. However, the notion of cardinality may be extended to fuzzy sets [24] by defining $|A|$ in a more general sense as

$$|A| = \sum_i \mu_i,$$

with the summation ranging over all elements in X which have a positive grade of membership in A. For example, in the case of (1), we have

$$|A| = 0.8 + 0.6 + 0.9 + 0.4 = 2.7.$$

This notion of cardinality will be used in our definition of a fuzzy neuron.

3. FUZZY NEURONS

First, let us review the assumptions underlying the McCulloch and Pitts model of a cell (neuron) [1]. They are

(1) the activity of the cell is an "all-or-none" process;
(2) a certain fixed number of synapses must be excited within the period of latent addition in order to excite a cell at any time, and this number is independent of previous activity and position on the cell;
(3) the only significant delay within the cell is synaptic delay;
(4) the activity of any inhibitory synapse absolutely prevents excitation of the cell at that time;
(5) the structure of a neural network does not change with time.

Based on these five assumptions, McCulloch and Pitts [1] proposed the following model for modeling the logical aspect of a neuron.

DEFINITION 1.

A *McCulloch-Pitts neuron* is a multi-input and multi-output memory system:

(1) It has two types of inputs· excitatory and inhibitory. It is assumed that they can only take on values 0 and 1. The excitatory inputs and the inhibitory inputs of the neuron will be denoted by $e_j(k)$ and $i_j(k)$, which are symbollically represented by $\rightarrow$ and —• , respectively.
(2) The threshold of the neuron is a positive integer n.
(3) The firing rules of the neuron are:
  (a) all the inhibitory inputs to the neuron must be 0;
  (b) the sum of all e_j's to the neuron must be equal to or greater than the threshold n.

When both of these conditions are met at time k, the neuron will fire at time $k+1$.

(4) The outputs of the neuron are 1 when it is firing and are 0 when it is quiet.

A McCulloch-Pitts neuron is depicted in Fig. 1(a).

DEFINITION 2.

A *McCulloch-Pitts neural network* is a collection of interconnected McCulloch-Pitts neurons. Each neuron in the network receives one input from the input of the network which may be excitatory or inhibitory. In addition to this input from the network input, it may also receive other excitatory and inhibitory inputs which are the outputs of the neurons in the network including itself. An example of McCulloch-Pitts neural network is shown in Fig. 1(b). The analysis of McCulloch-Pitts neural networks may be found for example in Minsky [18].

Our model differs from the McCulloch-Pitts model mainly in the replacement of the first of the above assumptions with the less restrictive assumption.

(1′) The activity of the cell is a "fuzzy" process.[1]

This leads us to the following definition.

DEFINITION 3.

A *fuzzy neuron* is a McCulloch-Pitts neuron with the following modification:

$$(1)\quad 0 \leqslant e_j(k),\quad i_j(k) \leqslant 1, \tag{2}$$

$e_j(k)$ stands for the "degree" to which the jth excitatory input is excited at

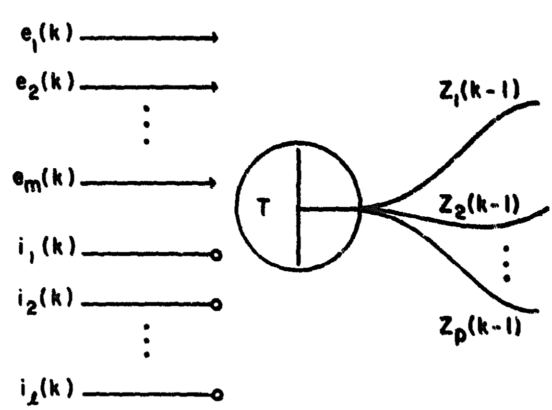

FIG. 1. A fuzzy neuron.

[1]Informally, a fuzzy process is a finite sequence of fuzzy operation. A formal definition of fuzzy process is given in Section 5.

time k, where the excitatory input is denoted by an arrow $\rightarrow$. Thus, the input is assumed to be a fuzzy set

$$E = e_1(k)E_1 + e_2(k)E_2 + \cdots + e_m(k)E_m,$$

where E_i is a label for the ith excitatory input. $i_j(k)$ stands for the jth inhibitory input at time k, where the inhibitory input is denoted by a circle $-\circ$.

(2) The threshold of the neuron is a positive *real* number.

(3) The outputs z_j's of the neuron are equal to some positive numbers μ_j's, $0 < \mu_j \leq 1$, if it is firing and are zero if it is quiet, i.e.,

$$z_j = \left\{ \begin{array}{ll} \mu_j & \text{if the neuron is firing} \\ 0 & \text{if the neuron is quiet} \end{array} \right\}$$

μ_j denotes the "degree" to which the jth output is fired. Thus, the output is assumed to be a fuzzy set Z

$$Z = \mu_1 Z_1 + \cdots + \mu_p Z_p,$$

where Z_j is a label for the jth output.

(4) The firing rules of the neurons are

(i) all the inhibitory inputs must be 0;

(ii) the sum of all e_i's must be equal to or greater than the threshold T of the neuron, i.e.,

$$\text{Neuron firing if } |E| \geq \text{threshold}, \tag{3}$$

when both of the above two firing rules are satisfied at time $t = k$, the neuron will fire at time $t = k + 1$, otherwise the fuzzy neuron remains in the quiet state.

A fuzzy neuron is depicted in Fig. 2(a).

DEFINITION 4.

A fuzzy network is defined exactly the same way as a McCulloch-Pitts neural network except its components are fuzzy neurons. A fuzzy neural network is shown in Fig. 2(b).

Two remarks about the fuzzy neuron and the fuzzy neural network should be made here.

(1) The inhibition caused by a nonzero inhibitory input of a fuzzy neuron is assumed to be absolute, which means that if any one of the inhibitory inputs is nonzero, the neuron will cease to fire regardless of the status of its excitatory inputs. This is the same assumption made in the McCulloch-Pitts neuron.

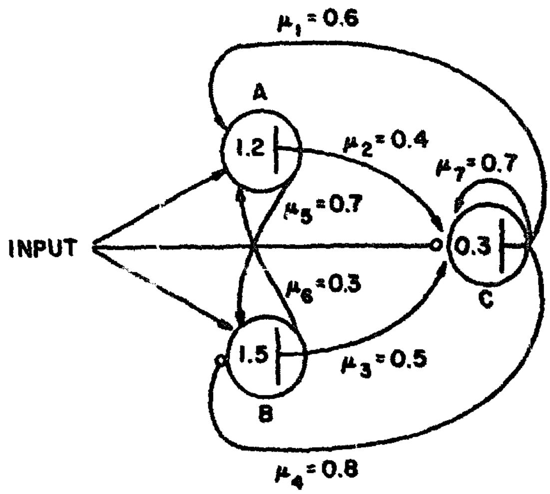

FIG. 2(a). A neural network.

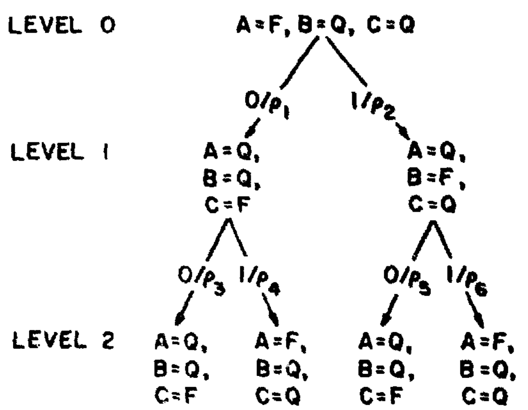

FIG. 2(b). The state transition tree of the network.

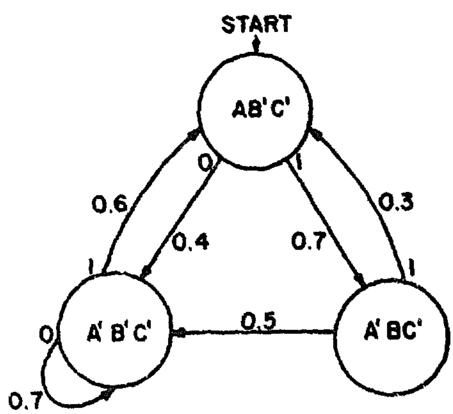

FIG. 2(c). The state diagram of the network.

(2) A fuzzy neuron is neither nondeterministic nor probabilistic. It is a fuzzy system [13, 14, 23], so is a fuzzy neural network.

In what follows, we shall present a general synthesis procedure for reasider the analysis of networks composed of fuzzy neurons, namely, the fuzzy neural networks.

4. ANALYSIS OF FUZZY NEURAL NETWORKS

It has been shown [18] that any (nonfuzzy) neural network is a finite state automaton. The state of a neural network is determined by the Cartesian product of the states (firing or quiet) of the neurons. The derivation of a state diagram from a fuzzy neural network is best illustrated by the use of an example.

Consider the network of Fig. 2(b). Assume that the initial state is the one where neuron A is firing (F) and neurons B and C are quiet (Q). In order to obtain the state transition diagram, we must first find the *fuzzy state transition tree* of the network which is described as follows:

(1) Apply the input symbol 0 to the initial state ($A=F, B=Q, C=Q$) and find the next states of the neurons A, B, and C by Definition 3. For example,

$$\left.\begin{matrix} A=F \\ B=Q \\ C=Q \end{matrix}\right\} \xrightarrow{\text{input}=0} \left\{\begin{matrix} A=Q \\ B=Q \\ C=F \end{matrix}\right.$$

The next states of the neurons A, B, and C of the initial state with an input 1 applied are obtained similarly. Display them as shown in Fig. 2(c).

(2) Repeat (1). A branch b of level k will terminate if the k-level leaf associated with the branch b has appeared in the first k levels.

(3) The process ends when all the branches are terminated.

(4) The final step is to determine the grade of membership ρ of the state transition from the μ_i's, which is defined as follows:

DEFINITION 5.

Define the σ_i of a firing neuron as

$$\sigma_i = \max[\mu_1, \mu_2, \ldots, \mu_s], \tag{4}$$

and the σ_i of a quiet neuron to be zero, where μ_i are the inputs to the neuron at the time $t=k$, excluding the input of the network at time $t=k$.

We postulate that the membership ρ of the state transition from the state at time k to the state at time $k+1$ is given by

$$\rho=\min[\sigma_1,\ldots,\sigma_{m_f}], \tag{5}$$

where m_f is the number of firing neurons at time $t=k+1$. The ρ so defined implies that each firing neuron has at least one input whose value is greater than or equal to ρ.

The complete fuzzy state transition tree of the neural network of Fig. 2(b) is shown in Fig. 2(c) and the grades of membership ρ_i of the state transitions are tabulated in Table 1. For simplicity, we shall abbreviate a firing neuron and a quiet neuron by the capital letter with a prime. For instance, the initial state is abbreviated by $AB'C'$. Using the states, the state transition diagram of the network is given in Fig. 2(d).

It should be remarked here that the state transition tree of a neural network containing n neurons, can have *at most* 2^n levels, which, in this example is $2^3=8$.

5. SYNTHESIS OF FUZZY AUTOMATA BY FUZZY NEURONS

A fuzzy automaton is based on a fuzzy process which is formally defined as follows:

DEFINITION 6.

A (finite) process in discrete time with a discrete state space $K=(q_1,q_2,\ldots,q_n)$ is called a (finite) *fuzzy process* if it satisfies the following conditions:

(1) The matrix F which describes the state transition has the following form

$$\mathbf{F}=\begin{array}{c|cccc|} & q_1 & q_2 & \cdots & q_n \\ \hline q_1 & \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ q_2 & \rho_{21} & \rho_{22} & \cdots & \rho_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ q_n & \rho_{n1} & \rho_{n2} & \cdots & \rho_{nn} \end{array},$$

where $0\leqslant\rho_{ij}\leqslant 1$, denotes the grade of membership of state transition from state q_i to state q_j. This matrix will be called *fuzzy state transition matrix* of the fuzzy process.

TABLE 1
The Values of ρ_i of the State Transition of the Neural Network of Fig. 2(b)

ρ_1	0.4	ρ_4	0.6
ρ_2	0.7	ρ_5	0.5
ρ_3	0.7	ρ_6	0.3

(2) Let M be a fuzzy set defined on K, and let

$$w_M^{(0)} = \left[\eta_{q_1}^{(0)}, \eta_{q_2}^{(0)}, \ldots, \eta_{q_n}^{(0)}\right]$$

be a row-vector, called *initial state designator of* A, where $\eta_{q_1}^{(0)}$ is the grade of membership of q_i with respect to M at time $t=0$. Then the state designator of M at $t=k$,

$$w_M^{(k)} = \left[\eta_{q_1}^{(k)}, \eta_{q_2}^{(k)}, \ldots, \eta_{q_n}^{(k)}\right]$$

is obtained by

$$\begin{aligned} w_M^{(k)} &= w_M^{(0)} \circ \underbrace{(\mathbf{F} \circ \mathbf{F} \circ \cdots \circ \mathbf{F})}_{(k-1)\text{ operations}} \\ &= w_M^{(0)} \circ \mathbf{F}^k = w_M^{(0)} \circ \left[\rho_{ij}^{(k)}\right], \end{aligned} \tag{6}$$

where

$$\rho_{ij}^{(k)} = \max_{\substack{\text{overall parallel paths} \\ \text{from } q_i \text{ to } q_j \text{ with } (k-1) \\ \text{numbers of transitions}}} \left\{ \min_{\substack{\text{overall series paths} \\ \text{from } q_i \text{ to } q_j \text{ with } (k-1) \\ \text{numbers of transitions}}} \left\{\rho_{i l_1}, \rho_{l_1 l_2}, \ldots, \rho l_{(k-1)} j\right\} \right\}. \tag{7}$$

COMMENTS

(1) The fuzzy state transition matrix $\mathbf{F}$ of a fuzzy process resembles somewhat to stochastic matrix of a Markov chain for $0 \leqslant \rho_{ij} \leqslant 1$; however, in general $\sum_{k=1}^{n} \rho_{ik} = 1$, for all i.

(2) The reason we use the max min (7) rule in defining $\rho_{ij}^{(k)}$ is that the state transition from q_i to q_j in a fuzzy process may be considered as water (gas, electricity, traffic flow, etc.) flow through a water supply system of which the water pipes are series-parallel interconnected.

(3) When ρ_{ij} takes only two values 0 and 1, the process becomes a

nondeterministic process. In addition, if only any one element of each row of matrix F is 1 and the remaining elements of each row are equal to 0, then the process is a *deterministic process*.

A Fuzzy Finite Automaton (FFA)' and the grade of acceptance are defined in Definition A.1 and Definition A.2.

Let Σ^* denote the set of all input sequences (tapes) including the empty word ϵ over a finite set of alphabets Σ, then the domain of F is ex empty word and I_n is $n\times n$ identity matrix:

$$F(\sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_k}) = F(\sigma_{i_1})\circ F(\sigma_{i_2})\circ\cdots\circ F(\sigma_{i_k}),$$

$$= [\rho_{ij}^{(k)}],$$

$$k \geqslant 2 \quad \text{and} \quad \sigma_{i_j}\in\Sigma, \qquad j=1,2,\ldots,k, \tag{8}$$

The symbol $\circ$ denotes the max min, operation, and $\rho_{ij}^{(k)}$ is defined as in (7).

In the previous section, it has been shown that a fuzzy neural network is a fuzzy finite state automaton (FFSA) or automaton (FFA). Then it is natural to ask the question: For a given fuzzy automaton transition diagram, can we always find a fuzzy neural network realizing it? The answer is yes. We shall begin this discussion by introducing the following definitions.

DEFINITION 7.

Let M be a FFA and q be a state of M. State q is said to be *homogeneous* if all the state transition lines of the state diagram of M incident to q are driven by the same input; otherwise we call it a *nonhomogeneous state*. If the input space is (0, 1), the homogeneous state driven by 1 and 0, are, called *1-state* and *0-state*, respectively.

For example, in Fig. 2(d), the states $AB'C'$ and $A'BC'$ are 1-states, and the states $A'B'C$ is a 0-states.

DEFINITION 8.

A FFA M is *homogeneous* if all its states are homogeneous, otherwise it is *nonhomogeneous*.

For example, the automaton of Fig. 2(d) is a homogeneous automaton.

PART A. SYNTHESIS OF STATE TRANSITION DIAGRAM

Part A-I. Synthesis of Homogeneous FFA

The synthesis procedure is illustrated by the following example.

Example 1. Consider the automaton M_1, described by the state diagram of Fig. 3(a).

(1) For a homogeneous automaton M with n states $A, B, C, \ldots$, use n neurons. Each of the neurons is labeled by $A, B, C, \ldots$.

(2) Classify the 1-states and 0-states of the automaton A: 1-state and B, C, D: 0-states.

(3) Define the identity mapping ϕ from the set of n states onto the set of n neurons as $\phi(X) = X$, $X = A, B, C, \ldots$.

(4) Start with the input node, and draw a line to $\phi(X)$ with an excitatory input if X is a 1-state and with an inhibitory input if X is a 0-state. We shall call the neuron $\phi(X)$ a 1-neuron and 0-neuron if X is a 1-state and 0-state, respectively.

(5) Draw additional lines with excitatory inputs from the output of $\phi(X)$ to the input of $\phi(Y)$ for all state transitions from state X to state Y.

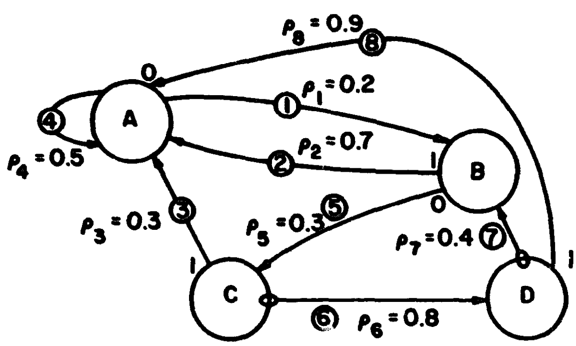

FIG. 3(a). The state diagram of automaton M_1.

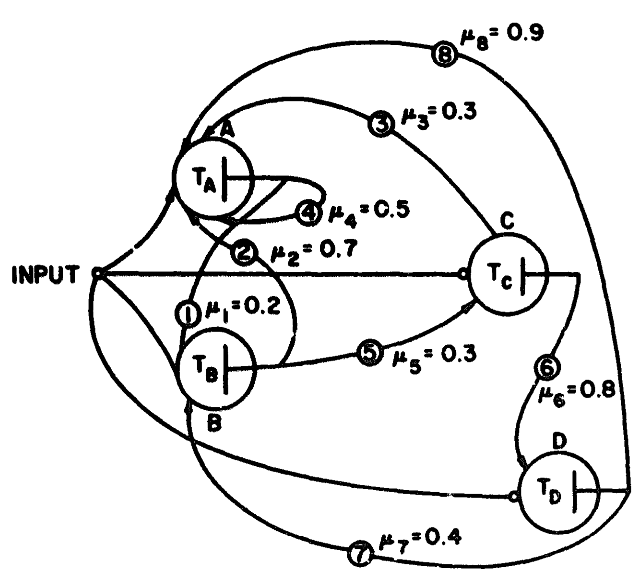

FIG. 3(b). Its fuzzy neural network realization.

(6) Define a one-to-one mapping ϕ from the n states of the automaton into the states of the n neurons as follows:

the n states of the automaton		the states of the n neurons
A	$\xrightarrow{\phi}$	$AB'C'\cdots$
B	$\xrightarrow{\phi}$	$A'BC'\cdots$
C	$\xrightarrow{\phi}$	$A'B'C\cdots$
	....	

(7) Determine the threshold values of the neurons. The way of determining the threshold values of the neurons is to construct the state transition table of the neural network of Fig. 4.3(b) according to the state transition diagram of M_1.

If state X to state Y is by an input a (either 0 or 1), then the state of the n neurons $\phi(X)$ to $\phi(Y)$ is by the same input a. This is shown in Table 2.

We find that the ranges of the threshold values of the fuzzy neurons are

$$0.9 < T_A \leqslant 1.3,$$

$$0 < T_B \leqslant 0.2,$$

$$0 < T_C \leqslant 0.3,$$

$$0 < T_D \leqslant 0.8.$$

The fuzzy neural network realization of automaton M_1 is shown in Fig. 3(b).

The way of determining the threshold values of the neurons described in step 7 of the above procedure is rather tedious. We shall prove that it is unnecessary.

THEOREM 1.

In the realization of a homogeneous automaton, the range of the threshold values of a 1*-neuron is*

$$\mu_{\max} < T_{1\text{-neuron}} \leqslant 1 + \mu_{\min}, \tag{9}$$

when there is only one 1*-neuron, in the neural network and*

$$1 < T_{1\text{-neuron}} \leqslant 1 + \mu_{\min}, \tag{10}$$

when there are more than one 1*-neuron in the neural network and the range of*

TABLE 2
The State Transition Table and the Requirement on the Threshold Values of the Neurons of the Neural Network of Fig. 3(b).

	INPUT 0		INPUT 1	
Present State	Next State	Requirement on the Value T of the Threshold	Next State	Requirement on the Value T of the Threshold
A	A'	$T_A > 0.5$	A	$T_A < 1.5$
B'	B	$T_B < 0.2$	B'	no information[a]
C'	C'	$T_C > 0$	C'	no information
D'	D'	$T_D > 0$	D'	no information
A'	A'	$T_A > 0.7$	A	$T_A < 1.7$
B	B'	$T_B > 0$	B'	no information
C'	C	$T_C < 0.3$	C'	no information
D'	D'	$T_D > 0$	D'	no information
A'	A'	$T_A > 0.3$	A	$T_A < 1.3$
B'	B'	$T_B > 0$	B'	no information
C	C'	$T_C > 0$	C'	no information
D'	D	$T_D < 0.8$	D'	no information
A'	A'	$T_A > 0.9$	A	$T_A < 1.9$
B'	B	$T_B < 0.4$	B'	no information
C'	C'	$T_C > 0$	C'	no information
D	D'	$T_D > 0$	D'	no information

[a]No information about the requirement of the value of the neuron is provided when it has an input with input value 1.

the threshold values a 0-neuron is

$$0 < T_{0\text{-neuron}} \leqslant \mu_{\min}, \tag{11}$$

where

$$\mu_{\min} = \min_{\substack{\text{all excitatory}\\ \text{inputs } \mu_i \text{ to}\\ \text{the neuron}}} [\mu_1, \mu_2, \ldots, \mu_v]. \tag{12}$$

$$\mu_{\max} = \max_{\substack{\text{all excitatory inputs}\\ \mu_i \text{ to the neuron except}\\ \text{to the neural network}}} [\mu_1, \mu_2, \ldots, \mu_r]. \tag{13}$$

Proof. We shall prove that the values of $T_{1\text{-neuron}}$ and $T_{0\text{-neuron}}$ cannot be outside the ranges described above. First consider the 0-neuron case shown in Fig. 4(a). When the input is 1, i.e., the inhibitory input having input value of 1, the neuron never fires. In this case, no information is provided in determining the threshold value of the neuron. Consider the case where the input of the neural network is 0. There is at most one input to the 0-neuron with input value other than 0, because at each moment one and only one neuron of the neural network can be in the firing state. The threshold value of the 0-neuron cannot be 0, because otherwise it fires all the time. This implies that the automaton always remains in the same state, since at each moment one and only one neuron can be in the firing state. An automaton corresponding to a neural network containing such a neuron is a trivial automaton (one state automaton) from that moment on. So we eliminate the value 0 as the threshold. The threshold value of the 0-neuron cannot be greater than μ_{min} either, since otherwise there will exist some excitatory input to the 0-neuron for which the 0-neuron will not fire. This will make the fuzzy neural network unable to realize the given automaton. When the threshold value is in the range $0 < T_{0\text{-neuron}} \leqslant \mu_{min}$ the neuron can provide both quiet and firing states in accordance with the transition diagram of the automaton.

Now consider the 1-neuron as shown in Fig. 4(b) and Fig. 4(c). The reason for that the threshold value cannot be greater than $1+\mu_{min}$ is similar to the case where the 0-neuron with threshold value cannot be greater than μ_{min}, just discussed.

For any 1-neuron, when the input is 0, its next state must be quiet, because the next state of the automaton cannot be in the state corresponding to this 1-neuron. For this reason, the threshold cannot be less than μ_{max}. When there is *only one* 1-neuron in the network and its threshold value is in the range $\mu_{max} \leqslant T_{1\text{-neuron}} \leqslant 1+\mu_{min}$, it can provide both quiet and firing states depending upon the state transition of the automaton. However, when there are *more than one* 1-neuron, then the range of the threshold values of 1-neuron is $1 < T_{1\text{-neuron}} \leqslant 1+\mu_{min}$.

Therefore, step 7 of the synthesis procedure can be replaced by step 7′.

(7′) Determine the threshold values of the neurons. The range of the threshold values of a 1-neuron is $\mu_{max} < T_{1\text{-neuron}} \leqslant 1+\mu_{min}$ when there is only one 1-neuron in the neural network and $1 < T_{1\text{-neuron}} \leqslant 1+\mu_{min}$ when there are more than one 1-neuron in the neural network and the range of the threshold values a 0-neuron is $0 < T_{0\text{-neuron}} \leqslant \mu_{min}$.

For example, in the neural network of Fig. 3(b), the A neuron is a 1-neuron with threshold T_A; and the B, C, and D neurons are 0-neurons with thresholds T_B, T_C, and T_D, respectively, which agrees with the theorem.

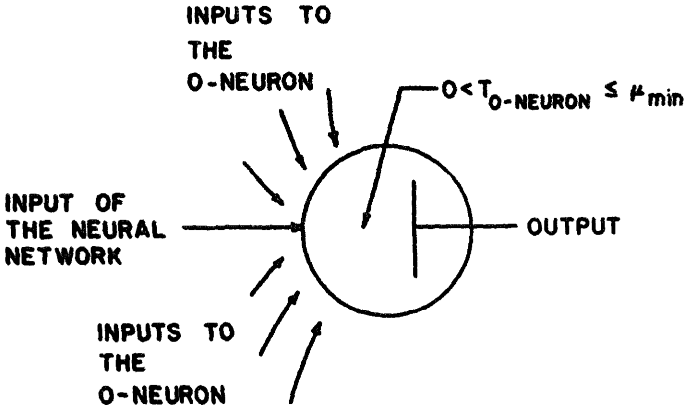

FIG. 4(a). The range of the threshold value of a 0-neuron.

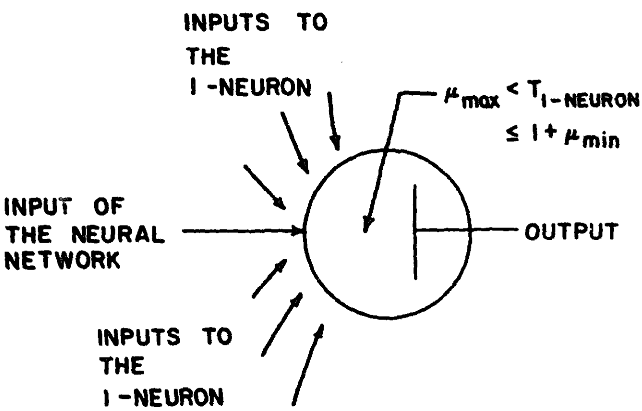

FIG. 4(b). The range of the threshold value of a 1-neuron when there is only One 1-neuron in the neural network.

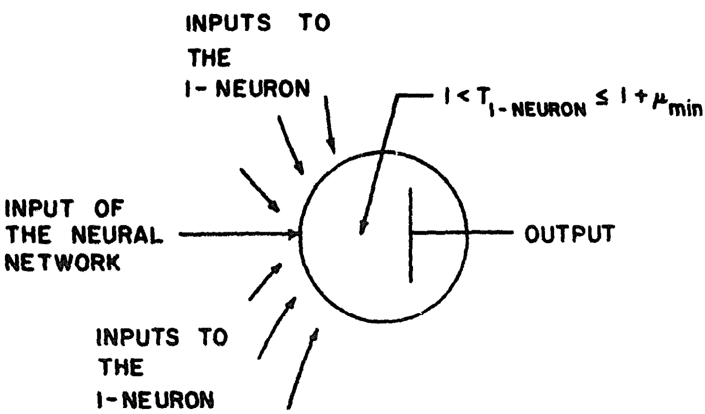

FIG. 4(c). The range of the threshold value of a 1-neuron when there are more than one 1-neuron in the neural network.

Part A-II. Synthesis of Nonhomogeneous FFA

The synthesis procedure for nonhomogeneous FFA consists of two steps:

(1) Construct a homogeneous automaton which is equivalent to the given automaton. The way of constructing such an equivalent automaton is to split each nonhomogeneous state into two, a 1-state and a 0-state, which results in a homogeneous automaton.

(2) Synthesize the equivalent homogeneous automaton using the method described in Part A-I.

To show the first step is always feasible, we describe the procedure for obtaining the homogeneous states from a nonhomogeneous state as follows.

(1) Split the nonhomogeneous state into two states. One is a 1-state, and the other is a 0-state.

(2) Consider the following three possible transition lines incident to the nonhomogeneous state:

(2-1) The transition lines coming to the state from other states. Draw all the transition lines driven by input 1 incident to the 0-state with the values of ρ_i's unchanged.

(2-2) The transition lines going out of the state. For each transition line going out of the state driven by an input which is either 0 or 1, if the state to which the transition line is incident, is a homogeneous state, draw two lines driven by the same input to that state with the same membership ρ_i. If the state to which the transition line is incident, is a nonhomogeneous state, draw two lines driven by the same input with the same membership $\rho_p$ut is 1 or 0.

(2-3) The self-looped transition lines of the state. If a self-looped transition line is driven by 1, draw a self-looped transition line driven by 1 at the 1-state and a transition line from the 0-state driven by 1 to the 1-state. If a self-looped transition line is driven by 0, draw a self-looped transition line driven by 0 at the 0-state and a transition line from the 1-state driven by 0 to the 0-state. Again, the membership ρ_i of all the state transitions are kept the same.

This procedure is illustrated by the following example.

Example 2. An automaton M_2 is described by the state transition diagram of Fig. 5(a), in which states A, B, and D are nonhomogeneous states and state C is a homogeneous 0-state. States A, B, and D are split into two states A_1 and A_0, B_1 and B_0, D_1 and D_0, respectively. The subscripts 1 and 0 denote the 1-state and 0-state. The state transition diagram of the homogeneous automaton M_2' in Figure 5(b) is constructed by the procedure just described.

DEFINITION 9.

Two states of a fuzzy automation are equivalent if they are (1) equivalent in the ordinary sense, i.e., for all input sequences applied to the two states outputs are identical, and (2) the grade of acceptance to all input sequences starting from the two states are identical.

DEFINITION 10.

Two fuzzy automata M_a and M_b are equivalent if for every state of M_a, there is at least one equivalent state in M_b and vice versa.

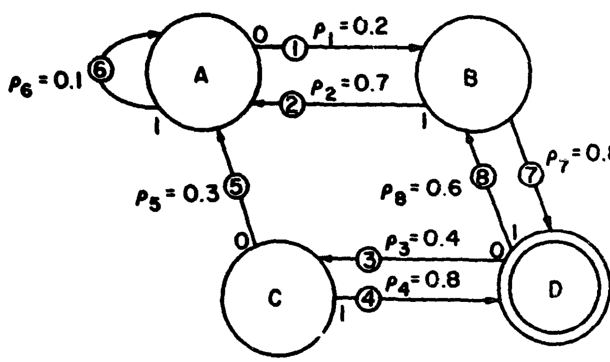

FIG. 5(a). A nonhomogeneous automaton M_2.

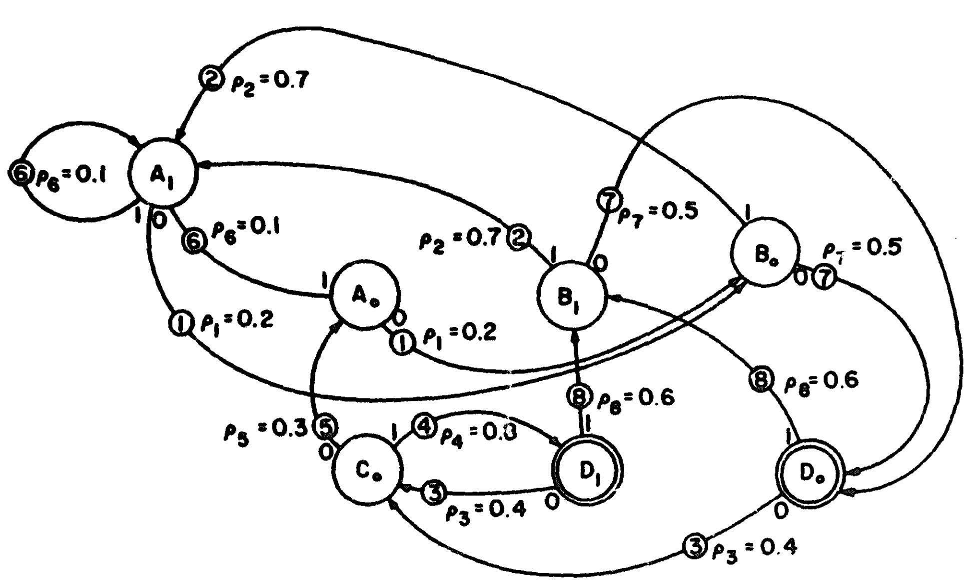

FIG. 5(b). The equivalent homogeneous automaton M_2'.

We can easily show the following theorem.

THEOREM 2.

Let M be a nonhomogeneous FFA.

(a) *For any nonhomogeneous state A of M, A_1 and A_0 are always obtainable.*

(b) *States A_1, A_0, and A are equivalent.*

(c) *If B is a homogeneous state of M, then B remains homogeneous under the nonhomogeneous state-to-homogeneous states transformation described above.*

Example 2 (*continued*). The equivalence relation among the states of the two automata: Automaton M_2 can be best seen from the state transition tables of the two automata which are shown in Table 3. In this table, besides the next states, the memberships of the state transition are also shown. For example, the entities $A_1/0.1$ at the first row and the first column of the left table means that the membership of the state transition from state A_1 to state A_1 by the input 1 is 0.1. It is seen from Table 3 that states A_1, A_0, and A, B_1, B_0, and B, and D_1, D_0, and D are equivalent states. Since for every state of automaton M_2, there is at least one equivalent state in the homogeneous automaton M_2' and vice versa, hence the two automata are equivalent.

COROLLARY 1.

For any nonhomogeneous automaton, there exists a homogeneous automaton which is equivalent to it.

Example 2 (*continued*). Realize the nonhomogeneous automaton M_2 of Fig. 5(a) using a fuzzy neural network.

TABLE 3
The state transition tables of M_2 and its homogeneous automaton equivalent M_2'

x_k / q_k	1	0	Equivalent class	x_k / q_k	1	0	Equivalent class
A_1	$A_1/0.1$	$B_0/0.2$	I	A	$A/0.1$	$B/0.2$	I
A_0	$A_1/0.1$	$B_0/0.2$					
B_1	$A_1/0.7$	$D_0/0.5$	II	B	$A/0.7$	$D/0.5$	II
B_0	$A_1/0.7$	$D_0/0.5$					
C_0	$D_1/0.8$	$A_0/0.3$	III	C	$D/0.8$	$A/0.3$	III
D_1	$B_1/0.6$	$C_0/0.4$	IV	D	$B/0.6$	$C/0.4$	IV
D_0	$B_1/0.6$	$C_0/0.4$					

The equivalent homogeneous automaton of M_2 was obtained above. Applying the method described in Part A-I, the realization is shown in Fig. 5(b), in which the output of the neural network is taken from the outputs of neurons D_1 and D_0 feeding to the input of an OR neural network. The ranges of the threshold values of the fuzzy neurons are obtained by Theorem 1 which are shown in Table 4.

The following theorem concerns with the lower bound of the number of neurons required for synthesizing an n-state *minimal* automaton, that is, in this minimal automaton no two states are equivalent.

THEOREM 3.

For an n-state minimal automaton, there must be at least $\lceil \log_2 n \rceil$[2] *number of neurons to realize it.*

Proof. Since a neural network containing m neurons can have at most 2^m state, 2^m must be greater than or equal to n. Since m must be an integer $m \geqslant \lceil \log_2 n \rceil$.

COROLLARY 2.

Let m denote the number of neurons required to realize an n-state minimal machine by the method described in this section. Then such a realization with $\lceil \log_2 n \rceil \leqslant m \leqslant 2n$ *can always be obtained.*

TABLE 4
The Ranges of Threshold Values of the Neurons of the Realization of M_2

Threshold	Lower Bound The threshold is greater than	Upper Bound The threshold is less than or equal to
T_{A_1}	1	1.1
T_{A_0}	0	0.3
T_{B_1}	1	1.6
T_{B_0}	0	0.2
T_{D_1}	1	1.8
T_{D_0}	0	0.5
T_{C_0}	0	0.4

[2]The symbol $\lceil \log_2 n \rceil$ denotes the least integer which is greater than or equal to $\log_2 n$.

PART B. REALIZATION OF THE GRADE OF ACCEPTANCE OF THE FFA

Part A has presented the synthesis of homogeneous and nonhomogeneous state transition diagram using fuzzy neurons. In this part, we shall present the realization of the degree of acceptance of the FFSA obtained in Part A. The grade of acceptance is defined in (7). The problem is to evaluate the grade of acceptance. The method for evaluating the grade of acceptance is best illustrated by the use of an example.

Example 3. Consider the FFA M_2 of Example 2.

$$M_2=(K,\Sigma,\delta,q_0,F),$$

where

$$K=(A,B,C,D),$$

$$\Sigma=(0,1),$$

$$q_0=\{(A,0.6),(B,0.7),(C,0.8),(D,0.9)\},$$

$$F=\{D\},$$

and δ is the fuzzy mapping from $K\times\Sigma$ to K as represented by $M(1)$ and $M(0)$ where

$$M(1)=\begin{array}{c} \\ A\\ B\\ C\\ D\end{array}\begin{array}{c}\begin{array}{cccc}A & B & C & D\end{array}\\ \begin{bmatrix}0.1 & 0 & 0 & 0\\ 0.7 & 0 & 0 & 0\\ 0 & 0 & 0 & 0.8\\ 0 & 0.6 & 0 & 0\end{bmatrix}\end{array},$$

and

$$M(0)=\begin{array}{c} \\ A\\ B\\ C\\ D\end{array}\begin{array}{c}\begin{array}{cccc}A & B & C & D\end{array}\\ \begin{bmatrix}0 & 0.2 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0.3 & 0 & 0 & 0\\ 0 & 0 & 0.4 & 0\end{bmatrix}\end{array}.$$

We have shown that M_2' of Fig. 5(b) is equivalent to M_2.

$$M_2'=(K',\Sigma,\delta',q_0',F'),$$

where

$$\Sigma=(0,1)$$

$$K'=(A_1,A_0,B_1,B_0,C_0,D_1,D_0),$$

$$q'_0=\{(A_1,0.6),(A_0,0.6),(B_1,0.7),(B_0,0.7),$$
$$(C_0,0.8),(D_1,0.9),(D_0,0.9)\},$$

$$F'=(D_1,D_0),$$

and δ' is the fuzzy mapping from $K'\times\Sigma$ to K' as represented by $M'(1)$ and $M'(0)$, where

$$M'(1)=\begin{array}{c} \\ A_1\\ A_0\\ B_1\\ B_0\\ C_0\\ D_1\\ D_0 \end{array}\begin{array}{c} \begin{array}{ccccccc} A_1 & A_0 & B_1 & B_0 & C_0 & D_1 & D_0 \end{array}\\ \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0.1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0.7 & 0 & 0 & 0 & 0 & 0 & 0\\ 0.7 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0.8 & 0\\ 0 & 0 & 0.6 & 0 & 0 & 0 & 0\\ 0 & 0 & 0.6 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

$$M'(0)=\begin{array}{c} \\ A_1\\ A_0\\ B_1\\ B_0\\ C_0\\ D_1\\ D_0 \end{array}\begin{array}{c} \begin{array}{ccccccc} A_1 & A_0 & B_1 & B_0 & C_0 & D_1 & D_0 \end{array}\\ \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0.2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5\\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5\\ 0 & 0.3 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0.4 & 0 & 0\\ 0 & 0 & 0 & 0 & 0.4 & 0 & 0 \end{bmatrix} \end{array}.$$

The procedure for evaluating the grade of acceptance is as follows: Suppose $x=100$. The grade of acceptance for $q'_0=(A_1,0.6),(A_0,0.6),(B_1,0.7),(B_0,0.7),(C_0,0.8),(D_1,0.9),(D_0,0.9)$, $x=100$, and $F'=(D_1,D_0)$, $\mu_1=(0.6, 0.6, 0.7, 0.7, 0.8, 0.9, 0.9)$ so

$$\mu_4=\mu_1M'(100) \qquad \text{[by (A.3)]},$$

$$\mu_4=[[[\mu_1M'(1)]M'(0)]M'(0)] \qquad \text{[by (A.4)]}.$$

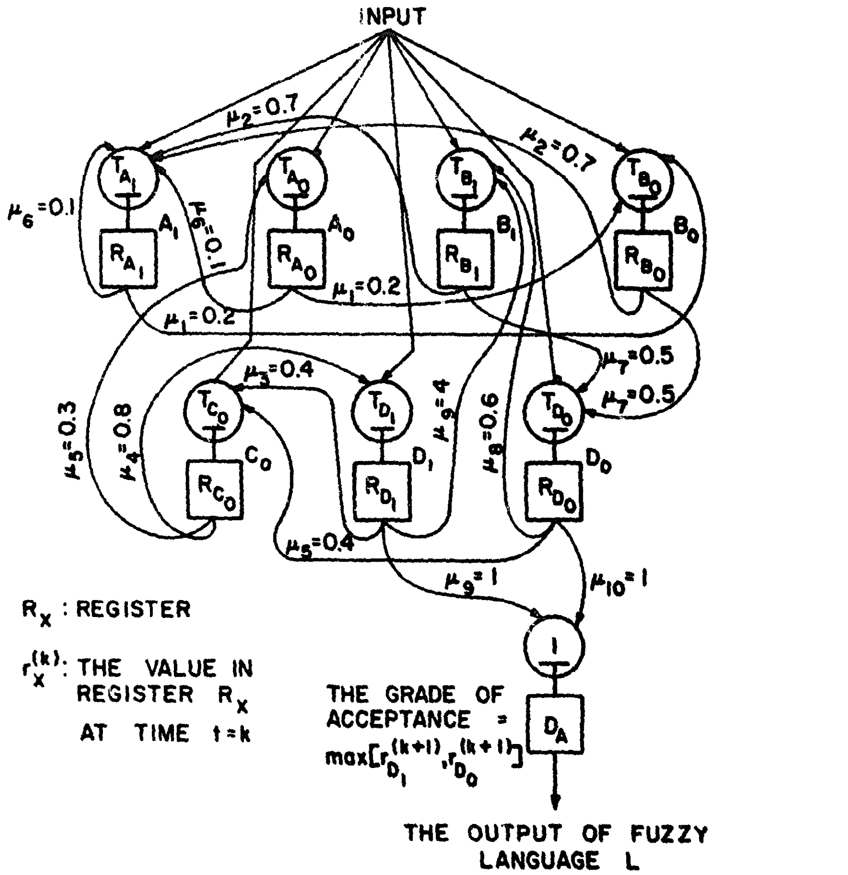

FIG. 6. The complete realization of M_2.

(1) At time $t=1$, set the initial values in the registers $R_{A_1},\ldots,R_{D_0}$ (see Fig. 6) to be the values of the initial designator, namely

$$\mu_1=\left[r_{A_1}^{(1)},r_{A_0}^{(1)},r_{B_1}^{(1)},r_{B_0}^{(1)},r_{C_0}^{(1)},r_{D_1}^{(1)},r_{D_0}^{(1)}\right]$$

$$=[0.6,0.6,0.7,0.7,0.8,0.9,0.9]$$

where $r_{A_1}^{(1)}$ is the value in R_{A_1} at time $t=1$, $r_{A_0}^{(1)},\ldots,r_{D_0}^{(1)}$ are similarly defined, the superscript denotes the time. Note that the row vector $[r_{A_1}^{(k)},r_{A_0}^{(k)},r_{B_1}^{(k)},r_{B_0}^{(k)},r_{C_0}^{(k)},r_{D_1}^{(k)},r_{D_0}^{(k)}]$ denotes the grade of membership of the states at time k which is the same as Q_k in (A.5).

(2) At time $t=2$, if the input is 1, set all the excitatory inputs to the 0-neurons to be zero. If the input is 0, set all the excitatory inputs to the 1-neurons to be zero. Define

$$(\nu_x)_{\min}=\min[\mu_x,r_x^{(1)}] \tag{13}$$

TABLE 5
The Sequences of the Values in the Registers

$$\pi'_k = [r_{A_1}^{(k)}, r_{A_0}^{(k)}, r_{B_1}^{(k)}, r_{B_0}^{(k)}, r_{C_0}^{(k)}, r_{D_1}^{(k)}, r_{D_0}^{(k)}]$$

where $r_{A_1}^{(k)}$ is the value in R_{A_1} at time k.

Time	R_{A_1}	R_{A_0}	R_{B_1}	R_{B_0}	R_{C_0}	R_{D_1}	R_{D_0}	π'_k
$t=0$	0.6	0.4	0.2	0.7	0.8	0.3	0.9	π'_0
$t=1$	0.7	0	0.8	0	0	0.8	0	π'_1
$t=2$	0	0	0	0.2	0.4	0	0.5	π'_2
$t=3$	0	0.3	0	0	0.4	0	0.2	π'_3

where μ_x is coming from the register R_x with value $r_x^{(1)}$. Replace the $r_x^{(1)}$ by the maximum of the $(\nu_x)_{\min}$'s which are incident to the neuron x. Denote this value by $r_x^{(2)}$.

(3) Repeat step 2 $(K-1)$ times, where K is the length of the input sequence the μ_K is thus obtained. For our example μ_2, μ_3, μ_4 are tabulated in Table 5.

It is of interest to note that if the input at time $t=k$ is 1, the values $r_{0\text{-neuron}}^{(k+1)}$ in all the registers of the 0-neurons are zero. If the input at time $t=k$ is 0, the values $r_{1\text{-neuron}}^{(k+1)}$ in all the registers of the 1-neurons are zero.

After μ_k is obtained, the grade of acceptance can be obtained by using a neuron *OR* network as shown in Fig. 6. It should be remarked that one additional unit delay is introduced by the *OR* neural network. The complete realization of M_2 is shown in Fig. 6.

6. FUZZY LANGUAGE RECOGNIZERS

The neural realization of FFA presented in the previous section can be used as fuzzy language recognizers. This is demonstrated by the following example.

Example 4. Consider the FFA M_2. Let L be a fuzzy language generated by the FFA M_2. Suppose there are two sequences of input symbols $x = 100$ and $y = 1001$. What is the grade of acceptance of x and y in L?

It is seen from Table 5 that the grade of acceptance of x is 0.2 by (A.5).

$$F' = (0,0,0,0,0,1,1),$$

$$Q_4 = (0,0.3,0,0,0.4,0,0,2) \qquad \text{from Table 5,}$$

$$\begin{aligned} \mu_M(x) &= \mu_M(100), \\ &= \max_{q_j} \mu_{F' \cap Q_4}(q_j), \\ &= 0.2. \end{aligned}$$

If one additional input symbol 1 is applied, then

	R_{A_1}	R_{A_0}	R_{B_1}	R_{B_0}	R_{C_0}	R_{D_1}	R_{D_0}
$t=5$	0.1	0	0.2	0	0	0.4	0

Thus the grade of acceptance of $y = 1001$ is

$$\begin{aligned} \mu_M(y) &= \mu_M(1001) \\ &= \max_{q_j} \mu_{F' \cap Q_5}(q_j) \\ &= 0.4 \end{aligned}$$

It should be clear that the fuzzy neural network (Sec. 5) corresponding to the FFA M is the fuzzy language recognizer for the fuzzy language accepted by this FFA M.

7. CONCLUSION

The McCulloch-Pitts cell has been generalized to the fuzzy neuron, which is based on the assumption that the activity of the cell is a "fuzzy" process rather than an "all-or-none" process. It has been shown that any fuzzy automaton can be realized by a fuzzy neural network. The lower and upper bound on the number of neurons to realize an n-state minimal fuzzy automaton are given. The realizations of fuzzy languages using fuzzy neural networks have been presented. The method may be applied to solving problems in many areas involving information processing, pattern recognition, and decision making.

APPENDIX

A fuzzy automaton may be viewed as a generalization of the concept of a nondeterministic automaton. Thus, if on a nondeterministic automaton A

the set of possible next states, given the present state and the present input, is a subset Γ of the state-space K, then in its fuzzy counterpart A the set of next states is a fuzzy subset Γ of K. By this process, which will be referred to as fuzzification, it is a simple matter to define such notion as fuzzy finite-state automaton, a fuzzy push-down automaton, a fuzzy Turing machine, a fuzzy algorithm, a fuzzy Markoff algorithm, etc. [21,22].

FUZZY AUTOMATA

To illustrate this process in more concrete terms, we define a fuzzy finite automaton (FFA) as follows:

DEFINITION A.1.

A fuzzy finite automaton (FFA) over an alphabet Σ is a quintuple (K,Σ,δ,q_0,F) where K is a finite nonempty set of states, Σ is a finite input alphabet, $q_0 \in K$ is the initial state, and $F \subset K$ is a set of final states. The symbol δ denotes a fuzzy mapping form $K \times \Sigma$ to K. This means each pair (q_i,a) in $K \times \Sigma$ defines a fuzzy "next state" in K which is characterized by a conditioned membership function $\mu(q_j|q_i,a)$ having as its arguments $q_j \in K$, $q_i \in K$ and $a \in \Sigma$.

If the state of FFA at time n is a fuzzy set Q_n in K defined by a membership function $\mu_n(q_i)$, and an input a is applied, then the state of FFA at time $n+1$ is a fuzzy set Q_{n+1} in K whose membership function is given by

$$\mu_{n+1}(q_j) = \bigvee_{q_i} (\mu_n(q_i) \wedge \mu(q_j|q_i,a)), \tag{A.1}$$

where $\bigvee_{q_i}$ denotes the supremum over $q_i \in K$. Note that when Q_n is a singleton $\{q_i\}$ the membership function for next state reduces to $\mu(q_j|q_i,a)$.

To simplify the notation, it is convenient to represent (3.1) in matrix form. Specifically, let $M(a)$ denote a matrix whose ijth element is $\mu(q_j|q_i,a)$; let μ_n and μ_{n+1} denote, respectively, row vectors with components $\mu_n(q_1),\ldots,$ $\mu_n(q_{|k|})$ and $\mu_{n+1}(q_1),\ldots,\mu_{n+1}(q_{|k|})$, where $|k|$ is the number of states. Then, (3.1) may be written as

$$\mu_{n+1} = \mu_n M(a), \tag{A.2}$$

in which the matrix product is understood in the normal sense, except that the sum $(+)$ is replaced by $\vee$ and the produce $(\cdot)$ by $\wedge$.

More generally, if $x = a_1 a_2 \ldots a_m \in \Sigma^*$ is an input string of length m, thus

$$\mu_{n+m} = \mu_n M(x), \tag{A.3}$$

where $M(x)$ is given by the matrix product

$$M(x) = M(a_1)M(a_2)\cdots M(a_m), \tag{A.4}$$

which should be interpreted in the same sense as (A.2).

Equation (A.3) defines the membership function μ_{n+m}, of the fuzzy state at time $n+m$ into which the fuzzy state at time n (characterized by μ_n) is taken by the input string $x = a_1a_2\ldots a_m$. It should be noted that (A.3) and (A.4) are analogous to the equations defining stochastic automata, except that they involve membership functions rather than probability distributions and the operations involved in them are $\vee$ and $\wedge$ rather $+$ and $\cdot$.

THE GRADE OF ACCEPTANCE

In what follow, fuzzy automata will be used primarily to characterize fuzzy languages in terms of their acceptance by such automata. For this purpose, we define the *grade of acceptance* of a string $x = a_1a_2\ldots a_m \in \Sigma^*$ by a fuzzy automaton. In the case of an FFA, the definition is stated as follows.

DEFINITION A.2.

Let Q_m denote the fuzzy state into which the initial state q_0 (at time $t=1$) is taken by an input string $x = a_1a_2\ldots a_m$, $a_1, a_2, \ldots, a_m \in \Sigma$. Let F be a designated set of final states, which may be a fuzzy subset of K. Then, $\mu_M(x)$, the *grade of acceptance* of x by the FFA, is given by the maximal grade in $F \cap Q_m$, the intersection of F and Q_m. More specifically,

$$\mu_M(x) = \max_{q_j} \mu_{F\cap Q_m}(q_j), \tag{A.5}$$

or

$$\mu_M(x) = \bigvee_{q_j} (\mu_F(q_j) \wedge \mu_{Q_m}(q_j)), \tag{A.6}$$

where the supremum is taken over all states in K. Note that by (A.3)

$$\mu_{Q_m} = \mu_1 M(x), \tag{A.7}$$

where μ_1 has only one nonzero component (equal to one) in the position corresponding to q_0.

The authors are indebted to Professor L. A. Zadeh of University of California, Berkeley, for his invaluable comments, suggestions, and encouragement throughout this research.

REFERENCES

1 W. S. McCulloch and W. Pitts, A logical calculus of the ideas imminent in nervous activity, *Bull. Math. Biophys.* 5, 1943.

2 John von Neumann, Probabilistic logics and the synthesis of reliable organisms from unreliable components, *Automata Studies*, Princeton Univ. Press, Princeton, N.J. (1956), pp. 43–98.

3 E. F. Moore, Gedanken-experiments on sequential machines, *Automata Studies*, Princeton Univ. Press, Princeton, N.J. (1956), pp. 129–153.

4 S. C. Kleene, Representation of events in nerve nets and finite automata, *Automata Studies* (Annals of Mathematics Studies, No. 34), Princeton Univ. Press, Princeton, N.J. (1956).

5 M. O. Rabin and D. Scott, Finite automata and their decision problems, *IBM J. Res. Develop.* 3, 114–125 (1959).

6 R. McNaughton and H. Yamada, Regular expressions and state graphs for automata, *Trans. IRE Prof. Group Electron. Comp.* EC-9, no. 1, 39–47 (March 1960).

7 R. McNaughton, The theory of automata, a survey, *Advances in Computers*, Vol. 2, Academic, New York (1961), pp. 379–421.

8 A. Gill, *Introduction to the Theory of Finite-State Machines*, McGraw-Hill, New York (1962).

9 M. A. Harrison, *Introduction to Switching and Automata Theory*, McGraw-Hill, New York (1965).

10 J. Harmanis and R. E. Stearns, *Algebraic Structure Theory of Sequential Machines*, Prentice-Hall, Englewood Cliffs, N.J. (1966).

11 S. Ginsburg, *An Introduction to Mathematical Machine Theory*, Addison-Wesley, Reading, Mass. (1962).

12 A. Ginzburg, *Algebraic Theory of Automata*, Academic, New York (1968).

13 L. A. Zadeh, Fuzzy sets, *Inform. Control* 8, 338–353 (June 1965).

14 L. A. Zadeh, Biological application of the theory of fuzzy sets and systems, *Proc. Symp. Biocybern. Cent. Nerv. Sys.* 1968.

15 W. G. Wee, and K. S. Fu, A formulation of fuzzy automata and its application as a model of learining systems, *IEEE Trans. Sys. Sci. Cybern.* SSC5, No. 3 (July 1969).

16 M. Mizumoto, J. Toyoda, and K. Tanaka, Some considerations on fuzzy automata, *J. Comp. Sys. Sci.* 3, 409–422 (1969).

17 E. S. Santos, Maximin automata, *Inform. Control* 13, 363–377 (1968).

18 M. L. Minsky, *Computation—Finite and Infinite Machines*, Prentice Hall, Englewood Cliffs, N.J. (1967).

19 E. T. Lee and L. A. Zadeh, Note on fuzzy languages, *Inform. Sci.* 1, 421–434 (1969).

20 R. E. Bellman, R. Kalaba, and L. A. Zadeh, Abstraction and pattern classification, *J. Math. Anal. Appl.* 13, 1–7 (1966).

21 L. A. Zadeh, Fuzzy algorithms, *Inform. Control* 12, 99–102 (February 1968)

22 E. S. Santos, Fuzzy algorithms, *Inform. Control* 17, 326–339 (November 1970).

23 R. E. Bellman and L. A. Zadeh, Decision-making in a fuzzy environment, *Management Sci.* 17, No. 4, B141–B164 (December 1970).

24 L. A. Zadeh, Fuzzy languages and their relation to human and machine intelligence, Memorandum No. ERL-M302, Department of Electrical Engineering and Computer Science, University of California, Berkeley, August 1971.