Theory Research on a New Type Fuzzy Automaton

QingE Wu[1], Tuo Wang[1], YongXuan Huang[1], and JiSheng Li[1]

School of Electronic and Information Engineering Xi'an Jiao Tong University, Xi'an, Shaanxi, 710049, P.R. China
wqe969699@163.com

Abstract. For better solving some complicated problems in fuzzy automata hierarchy, simultaneously, in order to accomplish better task for fuzzy signal processing, this paper presents a kind of new automaton–fuzzy infinite-state automaton. The basic extracted frame of fuzzy infinite-state automaton is introduced by using neural networks. To the extracted fuzzy infinite-state automaton, this paper describes that it is equivalent to fuzzy finite-state automaton, and its convergence and stability on its hierarchy system will be discussed. Finally, the simulation is carried on and the simulation results show that the states of fuzzy infinite-state automaton converge to some stable states with extraction frame and training for weights what this paper provides at last. Finally, some problems of fuzzy infinite-state automaton and neural networks to be solved and development trends are discussed. These researches will not only extend further automata hierarchy, but also increase a new tool for application of fuzzy signal processing. It is an important base in the application of fuzzy automata theory.

1 Introduction

In previous work, we classify the fuzzy automata according to recognizing the type of the language. Accordingly, fuzzy automata have as well partition according to recognizing the feature of the language, and then the automaton is classified into deterministic automaton and non-deterministic automaton or fuzzy automaton (FA). The FA is classified into fuzzy finite-state automaton (FFA) and fuzzy infinite-state automaton (FIA). Non-deterministic automata and fuzzy automata can be transformed into deterministic automata.

Previously, fuzzy knowledge equivalence representations between neural networks, fuzzy systems and models of automata are discussed [1]. From a control point of view, fuzzy finite-state automata with recurrent neural networks [2-4] for often imitating fuzzy dynamical systems are very useful. Previously, works have been shown how FFA can be mapped into recurrent neural networks with second-order weights using a crisp representation of FFA states [5].

Until present, the automata of fuzzy or defuzzization what we study are all finite-state automata [6]. However, FIA is not introduced. In this paper, we will discuss extraction of the FIA by using recurrent neural networks. The equivalence

L. Wang et al. (Eds.): FSKD 2006, LNAI 4223, pp. 1–10, 2006.

of FIA and FFA is described. Since the FFA is equally powerful as the deterministic finite-state automaton (DFA), as well as FIA is equivalent to FFA, the FIA is equivalent to the DFA at last. The convergence and stability of FIA is also discussed. Some new definitions and theorems are given. Finally, the simulation results show that the states of FIA converge surely some stable points. Through these studies in this paper, it not only strengthens the relationship between the fuzzy systems and hierarchy of fuzzy automata, but also these problems to be solved will be directly impulse the development of theories and applications of fuzzy automata, and it will show the further and more spacious prospect in wide applications. Thus, there will be a theoretic base for extraction and application of any automata.

2 Preliminary FIA

According to [5], the definition of FFA is introduced as follows:

Definition 2.1. A fuzzy automaton (FA) is named for a fuzzy finite-state automaton (FFA) M if it consists of a six-tuple $M = (Q, \Sigma, F, Q_0, G, V)$. Each factor of the six-tuple denotes respectively as follows:

Where Q is a finite set of states; Σ is a finite set of input alphabet; $Q_0 \subseteq Q$ is a fuzzy set of initial states; $G \subseteq Q$ is a fuzzy set of final states; $V \subseteq [0,1]$ is a membership degree set of transition relation; and $F \in V : Q \times \Sigma \times Q \to V$ is a fuzzy relation between Q, Σ and Q, i.e., $F(q_i, \sigma, q_j) \in V$, where $q_i, q_j \in Q, \sigma \in \Sigma$. Then, the fuzzy automaton (FA) is called FFA.

Now, introduce how a FFA accepts the fuzzy language.

For $\sigma \in \Sigma$, denote $F_\sigma \in V$ by $F_\sigma(q_i, q_j) = F(q_i, \sigma, q_j)$. The degree $\big(L(FFA)\big)(\omega)$ that a FFA M accepts a word $\sigma_1 \cdots \sigma_n \in \Sigma^*$ is defined by:

$$\big(L(FFA)\big)(\sigma_1 \cdots \sigma_n) = P(q_0) \circ F_{\sigma_1} \circ \cdots \circ F_{\sigma_n} \circ G(q_n)$$

Where $P(q_0)$ and $G(q_n)$ are the membership degree in the initial state q_0 and the final state q respectively, and $\circ$ denotes the max-min composition of fuzzy relation, i.e.,

$$\big(L(FFA)\big)(\sigma_1 \cdots \sigma_n) = \bigvee_{q_0, q_1, \cdots, q_n \in Q} P(q_0) \bigwedge F_{\sigma_1}(q_0, q_1) \bigwedge \cdots \bigwedge F_{\sigma_n}(q_{n-1}, q_n) \bigwedge G(q_n)$$

$L(FFA)$ in Σ^* is called the fuzzy language accepted by FFA. $L(FFA)$ denotes the fuzzy language as follows:

$$L(FFA) = \Big\{(\omega, \mu) | \omega \in \Sigma, \mu = \bigvee_i \mu_i, F_\omega(q_0, q_i) = \mu_i \in V, \exists q_i \in G\Big\}. \quad (1)$$

Where $\mu = \bigvee_i \mu_i$ signifies μ is obtained by 'or' operator of μ_i.

Definition 2.2. A fuzzy automaton is called a fuzzy infinite-state automaton (FIA) M if it also consists of a six-tuple $M = (Q, \Sigma, \delta, Q_0, G, V)$. Each factor of the six-tuple also denotes respectively as follows:

Where Q is an infinite set of states, Σ is a set of input symbols, $Q_0 \subseteq Q$ is a fuzzy set of initial states, $G \subseteq Q$ is a fuzzy set of final states, $V \subseteq [0, 1]$ is a membership degree set of transition function, and V is an infinite set, simultaneously, for any $\mu \in V$, $\delta : Q \times \Sigma \xrightarrow{\mu} Q$ is a transition function, i.e., $\delta(q_i, a, \mu) = \{q_j\}$, where $q_i, q_j \in Q, a \in \Sigma^*, \mu \in V$. Finally, the $G_\mu(q)$ denotes the fuzzy membership degree $\mu \in V$ at final state $q \in G$. Then, the FA is called the FIA.

Similarly, introduce how a FIA accepts the fuzzy language.

The degree $\big(L(FIA)\big)(\sigma_1 \cdots \sigma_n)$ that a FIA M accepts a word $\sigma_1 \cdots \sigma_n \in \Sigma^*$ is defined by $\big(L(FIA)\big)(\sigma_1 \cdots \sigma_n) = G_\mu\big(\delta(q_0, \sigma_1 \cdots \sigma_n, \mu)\big)$, where $q_0 \in Q_0, \mu \in V$, and where $\delta(q_0, \sigma_1 \cdots \sigma_n, \mu) = \delta\big(\delta(q_0, \sigma_1, \mu_1), \sigma_2 \cdots \sigma_n, \mu_2 \cdots \mu_n\big) = \cdots = \delta(q_{n-1}, \sigma_n, \mu_n) = \{q_n\}$, and $q_i \in Q$, $\mu_i \in V$, $i = 1, \cdots, n$.

$L(FIA)$ is called a fuzzy language accepted by FIA. The $L(FIA)$ is represented by the following set and where $\mu = \bigvee_{ij} \mu_{ij}$ signifies μ is obtained by 'or' operator of μ_{ij}.

$$L(FIA) = \Big\{(\omega, \mu) | \omega \in \Sigma^*, \mu = \bigvee_{ij} \mu_{ij}, \delta(q_i, \omega, \mu_{ij}) = \{q_j\}, \exists q_j \in G, \forall \mu_{ij} \in V, \forall q_i \in Q_0\Big\} \quad (2)$$

A fuzzy language L is acceptable by a FIA iff, $L = L(FIA)$, for some FIA.

3 Recurrent Neural Network Architecture for FIA

Based on a previous result that we encode FFA into recurrent neural networks [6], here we use the discrete recurrent neural network structure for mapping FIA into recurrent networks. The network architecture for extracting FIA is shown in Fig. 1.

3.1 Basic Structure of Recurrent Networks for FIA

The networks for FIA consist of two parts that are the trained networks and the extraction networks of FIA respectively. In training layer of networks, the recurrent neural networks are formed of N recurrent hidden neurons, and N output neurons, labeled $Y_j(t), j = 0, 1, \cdots, N-1$; M input neurons, labeled $x_l(t), l = 0, 1, \cdots, M-1$ with some weights ω_{jl}, associated to the links of these neurons. On extraction layer of networks for FIA, let neurons of extraction layer be a number of neurons and label L that are infinite, the L competitive neurons connect with the N output neurons by $N * L$ weights labeled $w_{ij}, i = 0, 1, \cdots, L-1, j = 0, 1, \cdots, N-1$.

The hidden unit activation function is the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$. The output in training layer is discrete value that is determined by discretization function $D(x)$, and $D(x)$ is given by the following (3) or (4).

(I) When the membership degree is any variable value in interval [0,1], i.e., there is the infinite number of membership degrees: Then, we divide interval

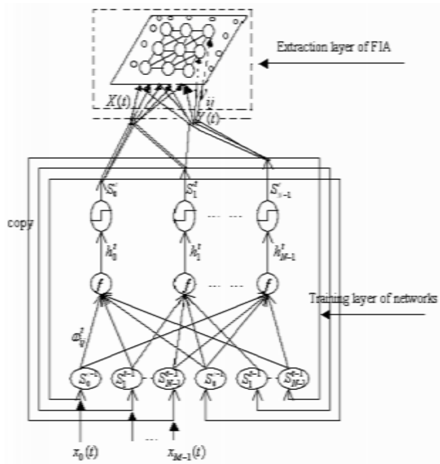

Fig. 1. Recurrent network architecture for FIA

[0,1]. Since the number of recurrent neuron is N, let us split the N intervals [0,1] into $n(n>1)$ coordinate subinterval, and the interval end-point value θ_s is obtained, where $s=0,1,\cdots,n$. Then, set

$$D(x)=\frac{\theta_i+\theta_{i+1}}{2} \quad if \;\; \theta_i<x\leqslant\theta_{i+1}, i=0,1,\cdots,n-1. \tag{3}$$

Where $\theta_0=0,\theta_n=1,\theta_i=\frac{i}{n}$.

(II) Based on statistic knowledge, when membership degree values are close to the corresponding finite real value $\{\theta_0,\theta_1,\cdots,\theta_{m-1}\}$, we set

$$D(x)=\theta_i, \;\; if \;\; |x-\theta_i|<\varepsilon, i=0,1,\cdots,m-1; x\in V. \tag{4}$$

Where ε is decided according to our demand, i.e., the final value x of neuron is close to θ_i after the whole string has been processed by the network.

For any values $\frac{\theta_i+\theta_{i+1}}{2}\neq 0$ and $\frac{\theta_i+\theta_{i+1}}{2}\neq 1$ are chosen instead of 0 and 1 here in order to give some power of influence to each of the current hidden unit values at the next time step, since a unit with value 0 would eliminate any influence of that unit.

We use h_i^t to denote the analog value of hidden unit i at the time step t, and S_i^t to denote the discretized value of hidden unit i at the time step t. ω_{ij}^t is the weight from unit j of layer 1 to unit i of layer 2 in training layer [6].

The dynamic process of network in training layer is described as follows:

$$h_i^t=f\Big(\sum_j \omega_{ij}^{t-1}S_j^{t-1}\Big), \forall i,t, \quad f(x)=\frac{1}{1+e^{-x}}, \quad S_i^t=D(h_i^t),$$

where $D(x)$ is obtained by the above equality (3) and (4).

3.2 Fuzzy States Representation for FIA

The current fuzzy state of FIA is a union of states $\{q_i\}$ with different fuzzy membership degrees. Consider the state q_j of FIA and the fuzzy state transition $\delta\Big(q_j, a_k, \{\theta_{ijk}\}\Big) = \{q_{i1}, \cdots, q_{ir}, \cdots\}$, where $a_k \in \Sigma$ is an input symbol and $\theta_{ijk} \in V$ is a fuzzy membership degree. We assign the corresponding recurrent state neuron S_j for state set $q_j = \{q_{j1}, \cdots, q_{jn} | n = 1, 2, \cdots\}$ of FIA, and the corresponding neurons $S_{i1}, \cdots, S_{ir}$ for states set $q_{i1}, \cdots, q_{ir}$ of FIA. The activation of recurrent state neuron S_i represents certainty θ_{ijk} with some state transition $\delta(q_j, a_k, \theta_{ijk}) = q_i$, i.e., $S_i^{t+1} \approx \theta_{ijk}$. If no state can reach q_i at time $t + 1$, then let $S_i^{t+1} \approx 0$.

4 Extraction of FIA

To training of networks and extraction algorithm for FFA, see [6,7] for them in detail. It is similar to Kohonen's self-organizing feature map (SOFM)[7]. The extraction for FIA is similar to FFA, but there are some differences between the two. Here, a part of algorithm way that is different from the one of [6,7] is only given as follows:

(1) Input a sample signal to networks and train the networks, we obtain an output vector $Y(t)$, where $Y(t) = (S_0^t, S_1^t, \cdots, S_{N-1}^t)^T$, S_j^t is an input signal in extraction layer for $\forall t$ and $j \in \{0, 1, \cdots, N-1\}$. Let $w_{ij}(t)$ be the weights of connection from training layer, unit j to extracting layer, unit i in the case of binary inputs, and Let $W(t) = (w_{ij})$ be weights matrix. Regard the output vector $Y(t)$ as input vector $X(t)$ on extraction layer of FIA; the input $X(t)$ is obtained from $Y(t)$ with $X(t) = (X_{k_i}^i(t))_{L\times 1}$, where several pieces S_j^t in $Y(t)$ unite and achieve the vectors $X_{k_i}^i(t) = (S_0^t, S_1^t, \cdots, S_{k_i}^t)^T$, $0 \leqslant i \leqslant L-1$, $L = \{1, 2, \cdots\}$, $0 \leqslant j, k_i \leqslant N-1$.

(2) At first, in the region D of the large range, regulate the weights matrix $W(t)$. Parameter D will be obtained by trial and error, If the FIA has been extracted, next task is to check whether it recognizes all the training examples or not. If the answer is affirmative, we have found an optimal D^* that we are looking for. In the case of negative answer, the value of D is decreased for one unit. Otherwise, the procedure ends. Once the networks of extracting layer are trained, a unit will represent a state of the FIA. Regulate the weights of the connecting neurons from training layer to extraction layer with $w_{ij}(t+1) = w_{ij}(t) + \alpha(t)\big(S_j^t - w_{ij}(t)\big)$, where $j \in \{0, 1, \cdots, k_i\}$, $0 \leqslant i \leqslant L-1$, $0 \leqslant k_i \leqslant N-1$; $0 \leqslant \alpha(t) \leqslant 1$ is a kind of variable velocity of study, i.e., the more the difference of the fuzzy membership degree S_j^t and weight $w_{ij}(t)$ is at the moment t, the bigger the value $\alpha(t)$ is. Assume $B = \max_{i,j}\{|S_j^t - w_{ij}(t)|\}$, and here set,

$$\alpha(t) = \begin{cases} \frac{|S_j^t - w_{ij}(t)|}{B} & B \neq 0 \\ 0 & B = 0 \end{cases}.$$

(3) There is an index I set that is the number of neuron to participate in competition at a time in extraction layer, i.e., $I \subset \{0, 1, \cdots, L-1\}$. If $\forall s \in I$,

$\exists i \in I$, there is $||W^i_{k_i}(t) - X^i_{k_i}(t)||_2 = \min\limits_{s\in I}\{||W^s_{k_s}(t) - X^s_{k_s}(t)||_2\}$, where $||\bullet||_2$ is an Euclidean 2-norm, $0 \leqslant k_i, k_s \leqslant N-1$, $W^i_{k_i}(t) = (w_{i0}, w_{i1}, \cdots, w_{ik_i})^T$, $0 \leqslant i \leqslant L-1$. We obtain the winner unit C_i, then the state of FIA that is extracted is q_i/x_i at the moment, where x_i is a fuzzy membership degree corresponding to the state q_i and is obtained by labeled $x_i = S^t_p$, with $|w_{ip}(t) - S^t_p| = \min\limits_j\{|w_{ij}(t) - S^t_j|\}$ for any $j \in \{0, 1, \cdots, k_i\}$, $\exists p \in \{0, 1, \cdots, k_i\}$, $0 \leqslant k_i \leqslant N-1$.

(4) Regulate again the weights to connect the winner node C_i and the weights to connect the interior node in geometry neighborhood of C_i with $w_{ij}(t+1) = w_{ij}(t) + \alpha(t)\big(S^t_j - w_{ij}(t)\big)$. The $w_{ij}(t+1)$ has a larger or a smaller regulating until the $w_{ij}(t+1)$ approaches to the S^{t+1}_j in range of error. Therefore there is $||W^i_{k_i}(t+1) - X^i_{k_i}(t+1)||_2 \leqslant ||W^i_{k_i}(t) - X^i_{k_i}(t)||_2$. Assume $B_i = \max\limits_j\big\{|S^t_j - w_{ij}(t)|\big\}$, and set,

$$\alpha(t) = \begin{cases} \frac{|S^t_j - w_{ij}(t)|}{B_i} & B_i \neq 0 \\ 0 & B_i = 0 \end{cases}.$$

The procedures of extracting FIA are shown as follows:

① At time $t = 0$, initialize S^0_0 to be 0.8 and all other S^0_j to be 0.2, $j \neq 0$ in order to give some power of influence for each of the current hidden unit values at the next time step. The network weights ω^0_{jl} are initialized randomly with a uniform distribution from -1 to 1 and ω^t_{jl} are given by trial and error later at time $t \neq 0$. Initialize $w_{ij}(0)$ randomly and let its value be in [0,1]. According to [6,7], by competition, the input $X(0)$ activates a unit (j_0, h_0) at extraction layer, which is taken as the initial state of the FIA, labeled $q_{(j_0,h_0)}/x_0$ that is determined by the vector $X^i_{k_i}(0) = (S^0_0, S^0_1, \cdots, S^0_{k_i})^T$, $0 \leqslant i \leqslant L-1$, $0 \leqslant k_i \leqslant N-1$, where $x_0 = S^0_p$ is a fuzzy membership degree.

② Starting out from the current activity unit (j, h) associated to state $q_{(j,h)}$ of FIA at time t. Introduce a previously unprocessed symbol $\xi_l \in \Sigma$ into the networks of training layer, and then an input vector $X(t)$ is obtained from producing an output vector $Y(t)$ and it activates a winner unit (m, n) that is taken as the corresponding state of the FIA. Now, a new state $q_{(m,n)}$ in FIA is or isn't created, but the associated transition, $\delta(q_{(j,h)}, \xi_l, \mu_{jm,hn}) = q_{(m,n)}$ is created. Calculate the membership degrees $\mu_{jm,hn}$ of state transitions by the above (3).

③ The following ξ_{l+1} is introduced into the networks at time $t+1$. Accordingly, it also obtains an active unit (m', n'). Thus, the transition has been created in the FIA from the activated unit (m, n) to the activated unit (m', n').

④ Repeat ②③ until all the symbols are processed.

5 Equivalence of FIA

The equivalence of FIA and FFA is discussed as follows:

Theorem 5.1. The FIA is equivalent to FFA.

Proof. Assume FIA $M_I = (Q, \Sigma, \delta, Q_0, G, V)$ accepts language $L(M_I)$, accordingly, a FFA $M_F = (Q_F, \Sigma, \delta_F, Q_{0F}, G_F, V_F)$ is made.

Since the V is a membership degree set of any transition and states of FIA, and the membership degree is from 0 to 1, choose $V = [0, 1]$ for general instance.

When the membership degree is any variable value in interval [0,1], i.e., there is the infinite number of membership degrees; let us divide the interval [0,1] into coordinate $n(n > 1)$ subinterval, and the interval end-point value θ_s is obtained, where $s = 0, 1, \cdots, n$. Then: $\mu_i = \frac{\theta_i+\theta_{i+1}}{2}$ if $\theta_i < x \leqslant \theta_{i+1}, i = 0, 1, \cdots, n-1$. Where $\theta_0 = 0, \theta_n = 1, \theta_i = \frac{i}{n}, x \in V$. We set $V_F = \{\mu_i | i = 0, 1, \cdots, n-1\}$.

$$q_i = \Big\{q_x \Big| \exists q \in Q, \exists i, \delta(q_x, \sigma, x) = q, \sigma \in \Sigma^*, \forall q_x \in Q, \forall x \in V, \theta_i < x \leqslant \theta_{i+1}\Big\}, i = 0, 1, \cdots, n-1. \tag{5}$$

At the same time, we set $Q_F = \bigcup_{i=0}^{n-1}\Big\{q_i\Big\}$ if there is a transition $\delta(q_j, \sigma, x_j) = \{q_i\}$, where $\forall \sigma \in \Sigma^*, \forall x_j \in V$.

It is obvious that the bigger n is, the more accurate FIA is equal to FFA.

Based on statistic knowledge, when membership degree values are close to the corresponding finite real value $\{\theta_0, \theta_1, \cdots, \theta_{m-1}\}$, i.e., $|x - \theta_i| < \varepsilon, x \in V$, let $\mu_i = \theta_i, i = 0, 1, \cdots, m-1$. At time, we set: $V_F = \{\mu_i | i = 0, 1, \cdots, m-1\}$.

$$q_i = \Big\{q_x \Big| \exists q \in Q, \exists i, \delta(q_x, \sigma, x) = q, \sigma \in \Sigma^*, \forall q_x \in Q, \forall x \in V, |x - \theta_i| < \varepsilon\Big\}, i = 0, 1, \cdots, m-1. \tag{6}$$

At the same time, we set $Q_F = \bigcup_{i=0}^{m-1}\Big\{q_i\Big\}$ if there is a transition $\delta(q_j, \sigma, x_j) = \{q_i\}$, where $\forall \sigma \in \Sigma^*, \forall x_j \in V$.

Assume $l = m$ or $l = n$, then the element of the Q_F is the $[q_0, q_1, \cdots, q_{l-1}]$; $Q_{0F} = [Q_0]$; $\bigcup_{i=0}^{l-1} q_i = Q$; $G_F \subseteq Q_F$ and each state of the G_F is one state subset of the final states of the M_I, i.e., the state of G_F is the following set:

$$q_G = \Big\{q_x \Big| \exists q \in Q, \exists i, \delta(q, \sigma, x) = q_x, \sigma \in \Sigma^*, \forall q_x \in G, \forall x \in V, \theta_i < x \leqslant \theta_{i+1} or |x - \theta_i| < \varepsilon\Big\}.\ i = 0, 1, \cdots, l-1.$$

The δ_F is defined by $\delta_F\Big([q_0, q_1, \cdots, q_{l-1}], a, \mu_i\Big) = [p_0, p_1, \cdots, p_k]$ iff $\delta\Big(\{q_0, q_1, \cdots, q_{l-1}\}, a, x\Big) = \{p_0, p_1, \cdots, p_k\}$ is satisfied, where $\mu_i \in V_F, x \in V$. It manifests that δ_F is obtained by solving δ, i.e., $\bigcup_{i=0}^{l-1} \delta(q_i, a, x) = \{p_0, p_1, \cdots, p_k\}$, $p_i \subseteq Q$ $(i, k \in \{0, 1, \cdots, l-1\})$, the subset $\{p_0, p_1, \cdots, p_k\}$ implies $[p_0, p_1, \cdots, p_k]$, i.e., $\delta_F\Big([q_0, q_1, \cdots, q_{l-1}], a, \mu_i\Big) = [p_0, p_1, \cdots, p_k]$. It is obvious the $[q_0, \cdots, q_{l-1}]$ is the state of the FFA M_F.

Now, we prove the equality $L(FIA) = L(FFA)$.

With regard to the length of the string ω is proved as follows:

$$\delta_F(q_{0F}, \omega, \mu_i) = [q_0, q_1, \cdots, q_{l-1}] \Longleftrightarrow \delta(q_0, \omega, x) = \{q_0, q_1, \cdots, q_{l-1}\} \qquad (*)$$

where $\mu_i \in V_F, x \in V$.

If $|\omega| = 0$, i.e., $\omega = \varepsilon$, there is $\delta_F(q_{0F}, \varepsilon, 1) = q_{0F}$, $\delta(q_0, \varepsilon, 1) = \{q_0\}$, $\forall q_0 \in Q_0$, $q_{0F} \in Q_{0F}$. Since $Q_{0F} = [Q_0]$, the conclusion is affirmed.

If $|\omega| \leqslant k$, assume the above $(*)$ is true.

Then, if $|\omega| = k+1$, i.e., $\omega = \omega_1 a$, $\omega_1 \in \Sigma^*$, $a \in \Sigma$, immediately, there is $\delta_F(q_{0F}, \omega_1 a, \mu_i) = \delta_F\big(\delta_F(q_{0F}, \omega_1, \mu_{i1}), a, \mu_{i2}\big)$ and $\delta_F(q_{0F}, \omega_1, \mu_{i1}) = [p_0, p_1, \cdots, p_i] \Leftrightarrow \delta(q_0, \omega_1, x_1) = \{p_0, p_1, \cdots, p_i\}$ is obtained by induction assumption.

Again, by the definition of the δ_F, $\delta_F\big([p_0, p_1, \cdots, p_i], a, \mu_{i2}\big) = [r_0, r_1, \cdots, r_j]$ is obtained and $\delta\big(\{p_0, p_1, \cdots, p_i\}, a, x_2\big) = \{r_0, r_1, \cdots, r_j\}$ is also satisfied. So, there is $\delta_F(q_{0F}, \omega_1 a, \mu_i) = [r_0, r_1, \cdots, r_j] \Longleftrightarrow \delta(q_0, \omega_1 a, x) = \{r_0, r_1, \cdots, r_j\}$ where $\mu_{i1}, \mu_{i2}, \mu_i \in V_F; x_1, x_2, x \in V$, $0 \leqslant i, j \leqslant l-1$.

Finally, there must be $\delta_F(q_{0F}, \omega, \mu_i) \in G_F$ only if there is $\delta(q_0, \omega, x) \in G$. Thus, it proves that the equality $L(FIA) = L(FFA)$ holds.

Theorem 5.2. [8] The FFA is equally powerful as some L-nested system of DFA.

6 Stability and Convergence of FIA

Now, we discuss the stability of FIA. Let us divide the stability of FIA into two parts, which are the stability of the trained networks layer and the stability of the extraction layer of FIA respectively. For the stability of the trained networks, see [9][10]. Therefore, we now discuss only the stability of the extraction layer for FIA.

Definition 6.1. For the extraction FIA that has been obtained, assume the input vector to be $X(t)$ and the corresponding weights vector to be $w(t)$ in extraction layer. We call the fuzzy automaton to be stable, if there are always $||w(t) - X(t)|| < \varepsilon$ while $t > t_0$, for any $\varepsilon > 0$, $\exists t_0 > 0$.

Theorem 6.1. The extracted FIA is stable by the above extraction algorithm.

Proof. According to the above algorithm (4) for extraction of FIA and the definition 6.1, the conclusion is true.

Definition 6.2. Let V be a membership degree set of FIA. For $\mu_i \in V$, the neighborhood NV_i of μ_i is defined by: $NV_i = \big\{\mu_j | \mu_j \in V, |\mu_i - \mu_j| < \varepsilon\big\}$. There exists some μ_{ij} neighborhood NV_{ij}, for any $\mu_{lj} \in NV_{ij}$, if $\delta(q_i, \omega, \mu_{ij}) = q_j$ and $\delta(q_l, \omega, \mu_{lj}) = q_j$, then let q_l be the same as q_i, and μ_{lj} be the same as μ_{ij}, where ε is an error bound by requiring, $q_i, q_l, q_j \in Q$, $\omega \in \Sigma^*$.

Theorem 6.2. The states of FIA converge to some stable states.

Proof. According to the characteristics of the membership degree set V and the dividing algorithm of V in the above section 3.1, we can always obtain the finite membership degree values $\mu_i, i = 0, 1, \cdots, l-1$ for FIA. Thus, by the definition 6.2, the states of FIA can converge to some stable states.

(I) When the membership degree is any variable value in $V \subseteq [0,1]$, let us divide the interval [0,1] into coordinate $n(n > 1)$ subinterval, and the interval end-point value θ_s is obtained, where $s = 0, 1, \cdots, n$.

Set $\mu_i = \frac{\theta_i + \theta_{i+1}}{2}$ if $\theta_i < x \leqslant \theta_{i+1}, i = 0, 1, \cdots, n-1$. ①
Where $\theta_0 = 0, \theta_n = 1, \theta_i = \frac{i}{n}, x \in V$. Let $\varepsilon_1 = \frac{1}{2n}$.

(II) When the membership degree values in V are close to the corresponding finite real value $\{\theta_0, \theta_1, \cdots, \theta_{m-1}\} \subseteq [0,1]$, i.e., $|x - \theta_i| < \varepsilon_2$ for any $x \in V$, $i = 0, 1, \cdots, m-1$.

We set $\mu_i = \theta_i, i = 0, 1, \cdots, m-1$. ②

Assume $l = m$ or $l = n$, and let $\varepsilon = \varepsilon_1$ or $\varepsilon = \varepsilon_2$.

Thus, there are the corresponding l states q_i, $i = 0, 1, \cdots, l-1$.

By the definition 6.2, then, for $\forall \mu_i$, there exists its neighborhood NV_i, $i = 0, 1, \cdots, l-1$. Again, by $\theta_i < x \leqslant \theta_{i+1}$ in the above ① or $|x - \theta_i| < \varepsilon_2$, for any $x \in V$, there is always $x \in NV_i$, $i = 0, 1, \cdots, l-1$. Regulate the weights of the connecting neurons to make the state q_x of FIA satisfy $\delta(q_i, \omega, \mu_{ij}) = q_j$ and $\delta(q_x, \omega, \mu_{xj}) = q_j$, then q_x converge to q_i, $i = 0, 1, \cdots, l-1$.

So, the states of FIA converge to some stable states q_i, $i = 0, 1, \cdots, l-1$.

7 Simulation Results

In order to simplify in simulation, here we discuss the input $X(t)$ is a two-dimensional vector. The weight vector $w(t)$ is an eight-dimensional regulated vector in the networks of extraction layer. The activating function f is a gauss function. The simulation time T is 100 seconds.

When we calculate in the experiment, in order to make networks more quickly reflect the state distribution law of FIA on the whole, in general, the study speed α and the region D are chosen to be relatively bigger value at the beginning of training networks. Generally, the training time in the back period is 10-100 times that of training time of the fore period. The simulation results are shown in Fig.2 and Fig.3. From the Fig.2 known, the simulation results indicate that extraction

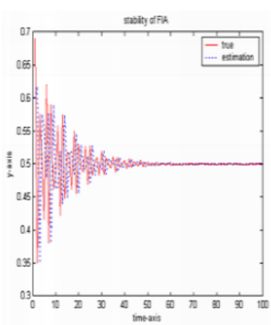

Fig. 2. Stability of FIA

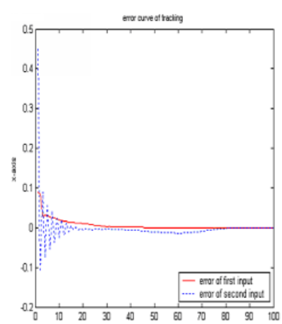

Fig. 3. Error curve of tracking

of FIA that has obtained is surely stable and convergent. By the Fig.3, the error curve for the difference of weights and inputs value in extraction layer reduces gradually and trends towards stability and convergence.

8 Conclusions

In this paper, these problems with respect to the definition, extraction, equivalence, convergence and stability of FIA are solved. The simulation results show that such extraction algorithm for FIA is surely stable and convergent. In conclusion, we have presented the basic ideas and algorithms for implementing stable recurrent networks and learning FIA in this paper. The network has similar capabilities for learning FIA as the analog FFA. These equivalent theorems imply that any two of FFA, FIA and L-nested systems of DFA are equally powerful. Then, the FIA is equivalent to the DFA at last.

Now, some questions require to be solved in the future: In order to learn better FIA, it is difficult how the states of FIA are minimized appropriate degree, i.e., how a new appropriate FFA will be obtained, and let it be equal to the FIA. It is difficult how the number of neuron and the layer of network are selected and designed for extracting the more stable FIA. These problems to be solved are worth being studied.

References

1. Blanco,A, Delgado,M, Pegalajar,M.C.: Identification of fuzzy dynamic systems using max-min recurrent neural networks. Fuzzy Set and Systems. 1 (2000) 63-70
2. Kosmatopoulos, E.B., Christodoulou, M.A.: Neural networks for identification of fuzzy dynamical systems: An Application to identification of vehicle highway systems. In Proc. 4^{th} IEEE Mediterranean Symp. New Directions in control and Automation. 1 (1996) 23-38
3. Kosmatopoulos, E.B., Polycarpou, M.M., Christodoulou, M.A., etc.: High-order neural networks for identification of dynamical systems. IEEE Trans. Neural networks. 6 (1995) 422-431
4. Kosmatopoulos, E.B., Christodoulou, M.A.: Recurrent neural networks for approximation of fuzzy dynamical systems. Int.J. Intell. Control Syst. 1 (1996) 223-233
5. Omlin,C.W, Thornber,K.K, Giles,C.L.: Fuzzy finite state automata can be deterministically encoded into recurrent neural networks. IEEE Trans. Fuzzy Syst. 6 (1998) 76-89
6. Giles, C.L., Miller, C.B., Chen, D., etc.: Learning and extracting finite state automata with second-order recurrent neural networks. Neural Computation. 4 (1992) 393-405
7. Lalande,A, Jaulent,M.: A fuzzy automaton to detect and quantify fuzzy artery lesions from arteriograms. In: Proceedings of the Sixth International Conference IPMU'96. In Canada. 3 (1996) 1481-1487
8. Radim, Bělohlávek.: Determinism and fuzzy automata. Information Sciences. 143 (2002) 205-209
9. Zhang.Naiyao, Yan.Pingfan.: Neural Networks and Fuzzy Control. Tsing-Hua University Press (1998) (Chinese)
10. Yuan.Jiangwen, Shankou. Heng, Gu.Longsi.: Neural network and fuzzy signal processing. Science Press (2003) (Chinese)