

다치오토마타 모델을 이용한 신경망 시스템 구현

Neural Network System Implementation Based on MVL-Automata Model

손창식, 박진희, 정환묵

대구가톨릭대학교 컴퓨터정보통신공학부

Chang-Sik Son, Jin-Hee Park, Hwan-Mook Chung

Dept. of Computer Information Communications Engineering, Catholic University of Daegu
(scs5152@hanmail.net, hmchung@amare.ac.kr)

ABSTRACT

기존의 유한오토마타는 입력값에 따른 상태 전이가 유한개의 문자열이 입력될 때는 정확하게 인식하나 무한개의 문자열이 입력될 때는 정확하게 인식하지 못한다는 문제점을 가지고 있다. 본 논문에서는 유한오토마타의 상태 전이를 다치오토마타 모델을 이용하여 무한개의 상태로 확장할 수 있는 가능성을 제시하고 이를 신경망 (Neural Network)으로 구현한 다치-신경망 시스템을 제안한다.

Keywords : MVL, Automata, MVL-Automata Model, Neural Network System

1. 서론

컴퓨터 기술의 급속한 발전을 통해 빠르고 정확한 정보처리에 따라 우리 인간의 생활은 날로 정보화되어 가고 있다. 최근 들어 멀티미디어 관련 기술이 발달함에 따라 영상, 사운드, 음성, 동영상 등의 여러 가지 미디어를 통해 인간과 보다 친숙한 방향으로 컴퓨터 기술이 발전되고 있다. 오토마타 이론은 수학적으로 추상화된 기계로 순차적이고 전기적인 회로의 특성을 모델링 하는데 사용되면서 그 응용의 범위가 점차 확대되어 컴퓨터 게임, 기계처리, 인공지능, 신경 시스템 동작, 그리고 로봇 이동 시스템 동작과 같은 행동 상황과 지능시스템을 모델링 하는데 사용되었다[1,2]. 지능 시스템의 여러 분야에서는 모델링의 필요성에 따라 오토마타에 확률, 퍼지 그리고 신경망 등의 개념을 도입한 다양한 형태의 오토마타 모델들이 사용되고 있다[2,3,4,5]. 본 논문에서는 지능 시스템 분야에서 사용될 수 있는 새로운 다치오토마타 모델을 제안하고, 다치오토마타 모델과 신경망을 병합하여 기존 오토마타에 상태변화를 확장 시킴으로써 무한개의 문자열이 입력으로 들어올 때 다음 상태가 어떻게 변하는가를 다치-

신경망을 통해 구현하고자 한다.

2. 오토마타

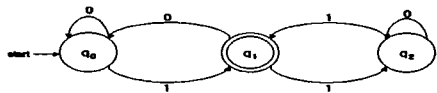
2.1 유한 오토마타

유한 오토마타는 유한한 상태들의 집합과 전이함수 (transition function)들의 집합으로 구성되어 있는데, 전이란 알파벳 Σ 로부터 선택된 입력 심볼 (input symbol)에 의해 생기는 상태에서 상태로의 변화이다. 입력 심볼에 따라 상태는 항상 변할 수 있으며, 원래의 상태로 다시 돌아가는 전이도 있을 수 있다. 보통 q_0 (q zero)으로 나타내는 상태를 시작 상태(start state)라고 하는데 이 시작 상태에서 오토마타가 시작하게 된다. 또한 어떤 상태들은 최종 상태 (final state) 또는 인식 상태 (accepting state)들로 지정되는데 그래프에서는 이중의 서클로 표시한다.

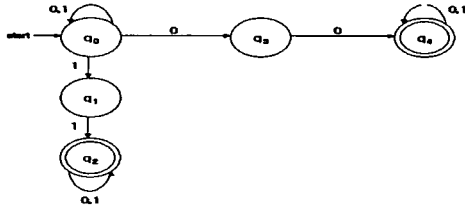
만약 입력 스트링 x 에 대해 시작 상태에서 최종 상태로 가는 경로가 존재한다면 FA는 스트링 x 를 인식한다.

유한 오토마타는 전이 방법에 따라 결정적 (deterministic) 유한 오토마타와 비결정적

(Nondeterministic) 유한 오토마타로 구분된다.



(a) 결정적 오토마타



(b) 비결정적 오토마타

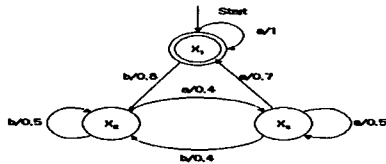
[그림2.1] 결정적 오토마타와 비결정적 오토마타

2.2 퍼지 오토마타

퍼지 오토마타는 다음과 같이 6가지 구성요소로 되어 있으며, 상태들 사이에 전이 정도를 가진다는 점에서 확률 오토마타와 유사하고, 퍼지에 min과 max 값을 취한다는 것이다[6].

FA = (Q, q₀, I, V, δ, f)

- Q : 상태들이 공집합이 아닌 유한집합.
- q₀ : q₀ ∈ Q인 초기 상태.
- I : 입력 알파벳으로서 공집합이 아닌 유한집합.
- V : [0,1]인 평가 공간.
- δ : Q×I → V로 정의되는 퍼지 전이함수.
- f : Q → V인 최종 상태 결정함수.

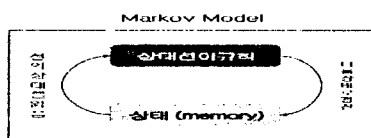


[그림2.2] 퍼지 오토마타의 전이 다이어그램

2.3 Markov 모델

볼츠만 머신은 마르코프 모델 (Markov Model)의 하나로 볼 수 있다.

Markov 모델에서는 상태가 시간과 함께 변화하지만 각 시점의 상태는 그 한 시점 이전의 상태만에 의존한다. 역으로 이와 같은 상태를 결정하는 기계라면 Markov 모델로 간주할 수 있다.



[그림2.3] Markov 모델

Markov 모델에서 과거에 관한 정보는 한 시점전의 상태에 집약되어 두고, 두 시점 이상 과거의 상태가 직접적으로 상태결정에 영향을 미칠 수 없고 상태를 결정하는 규칙 (상태전이규칙)은 확률적 (비결정적)이다.

Markov 모델의 상태전이규칙은 다음과 같이 정식화할 수 있다.

상태가 L개의 상태치 {α₁, α₂,..., α_L}을 취할 경우에 대하여 가정한다. 또한 현시점의 상태를 s(t), 다음 시점의 상태를 s(t+1)라 하면,

s(t+1)는 그 치로서 {α₁, α₂,..., α_L}중의 어떤 치를 갖게되므로, 그 치는 (식 2.1)과 같이 확률적으로 정한다.

$$p(s(t+1) = \alpha_i) = F_i(s(t)) \quad i=1,2,\dots,L \quad (2.1)$$

여기서, s(t+1)는 반드시 {α₁, α₂,..., α_L}의 값을 취하게 되므로 (식 2.2)와 같이 된다.

$$\sum_{i=1}^L p(s(t) = \alpha_i) = 1 \quad (2.2)$$

3. 다치오토마타 모델을 이용한 신경망 시스템

3.1 다치오토마타모델

다치오토마타 (MVL-Automata)는 다음과 같이 정의한다. 오토마타는 입력 (외부 환경) 스트링의 값, 상황 (상태) 및 출력 (행동)으로 나타낸다[3,5].

$$f = a_i \sum X_i X_j \quad (3.1)$$

(단, ij ∈ {1,2,...,n})

예를 들어, 다음과 같은 상태표를 가정한다.

현재 상태 (Present State) S(t)	입력 (Input) X(t)		
	a	b	c
a	α ₂	α ₂	α ₁
b	α ₀	α ₂	α ₀
c	α ₀	α ₂	α ₂

[그림3.1] 다치오토마타 상태표

다치오토마타 모델 (MVL-Automata Model)은 유한 상태 기계 (Finite-State Machine : FSM) 정의에 따라 5가지 요소로 구성될 수 있다.

$$MVLA = (I, O, S, f, g) \quad (3.2)$$

여기서, I, O 및 S는 다치 입력, 출력 및 상태들의 공집합이 아닌 유한집합이다. 여기서,

$f: I \times S \rightarrow S$ 은 상태 전이 함수이고, g 는 출력 함수이다. 즉 $g: I \times S \rightarrow O$ 이다. 카티시언 곱 (Cartesian Product) $I \times S$ 는 요소 $(X(t), S(t))$ 의 모든 쌍들을 포함한다. 상태 전이 함수 f 는 S 내에 다음 상태 $S(t+1)$ 상에 각각의 쌍 $(X(t), S(t))$ 을 사상하고, 그리고 출력 함수 g 는 식 (3.4)에 따라 O 내에 출력 $y(t)$ 상에 각각의 쌍 $(X(t), S(t))$ 을 사상한다.

$$S(t+1) = f(X(t), S(t)) \quad (3.3)$$

그리고

$$y(t) = g(X(t), S(t)) \quad (3.4)$$

여기서, $X(t)$ 는 입력, $S(t)$ 는 현재상태, 그리고 $S(t+1)$ 은 다음 상태이다. 다치오토마타는 여기서 단지 불연속 시간 (이산시간) $t = 0, 1, 2, \dots$ 으로 정의된다. 그리고 현재 상태 출력 함수는 식 (3.5)과 같고, 제안된 다치-신경망 시스템의 상태는 식 (3.3)에서 보여주는 것처럼 입력과 상태에 의존한다.

$$y(t) = g(S(t)) \quad (3.5)$$

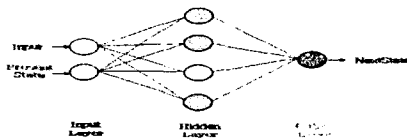
일반적으로, 상태 전이 함수 f , 그리고 출력함수 g 는 각각의 식 (3.3)과 (3.4)에 의해서 식 (3.6)과 (3.7) 같은 규칙들로 나타낼 수 있다.

$$\begin{aligned} & \text{If Input} = X(t) \text{ and PresentState} = S(t) \\ & \text{then NextState} = S(t+1) \end{aligned} \quad (3.6)$$

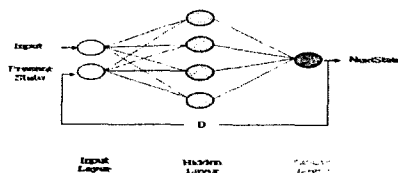
$$\begin{aligned} & \text{If Input} = X(t) \text{ and PresentState} = S(t) \\ & \text{then Output} = y(t) \end{aligned} \quad (3.7)$$

3.2 다치-신경망 시스템

다치-신경망 시스템의 전체 구조는 다음과 같이 구성된다.



[그림3.2] 훈련모드에서 다치-신경망 구조



[그림3.3] 재현모드에서 다치-신경망 구조

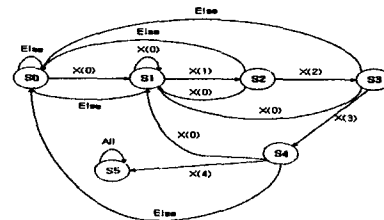
여기서, $X(t)$, $S(t)$, $S(t+1)$ 는 [그림3.2]에서와 같이, 입력, 상태, 다음상태를 나타내고, D (Unit Delay)는 시스템이 최종상태인지 아닌지를 결정하는 비교기 역할을 한다.

4. 적용 예

제안된 기법의 타당성을 보이기 위해 MVLA는 아래와 같은 상태 전이 다이어그램과 이들에 대응하는 상태 전이 테이블을 신경망에 적용하였다. 그리고 어떤 상태에서 또 다른 상태로 변하는 MVLA는 불연속인 경우에 도달하는 패턴 샘플 값들에 의존한다.

[표4.1] 훈련 데이터를 생성하기 위한 절차

1. 상태 다이어그램을 고안한다.
2. 상태 다이어그램으로부터 상태 전이 테이블을 생성한다.
3. 상태들에 수치적인 값을 할당한다.
4. 대응하는 상태와 함께 입력 범위내에 "Else" 입력을 생성한다.



[그림4.1] 상태 전이 다이어그램

[표4.2] 상태 전이 테이블

Input	Present State	Next State
X(0)	S0	S1
Else	S0	S0
X(0)	S1	S1
X(1)	S1	S2
Else	S1	S1
X(2)	S2	S3
Else	S2	S2
X(3)	S3	S4
Else	S3	S3
X(4)	S4	S5
Else	S4	S4
All	S5	S5

위 [표4.2]는 [그림4.1]에 상태 전이 다이어그램을 나타낸 것이다. 여기서, $S0$ 는 초기상태이고 $S5$ 은 최종상태를 의미한다.

상태 다이어그램을 고안하고, 상태 다이어그램으로부터 상태 전이 테이블을 생성한 다음, 아래와 같이 각 상태들에 수치적인 값을 할당한다. 아래 [표4.3]에 각 상태에 수치적인 값을 할당하는 이유는 신경망의 수렴성을 용이하게 하기 위한 것이다.

[표4.3] 수치적인 상태 할당

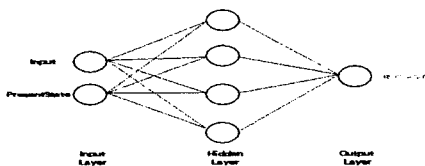
State	Numeric value
S0	4
S1	6
S2	2
S3	3
S4	5
S5	1

[표4.4] 훈련 데이터

X(t)	S(t)	S(t+1)	0	3	4
0	4	4	2	3	5
1	4	4	2	3	5
2	4	6	3	3	4
3	4	4	4	3	4
4	4	4	4	3	4
5	4	4	5	2	4
0	6	4	1	5	4
1	6	4	4	5	4
2	6	6	2	5	1
3	6	4	3	5	4
4	6	4	4	5	4
5	6	4	5	5	4
0	2	4	0	1	1
1	2	4	2	1	1
2	2	4	2	1	1
3	2	6	3	1	1
4	2	4	4	1	1
5	2	4	5	1	1

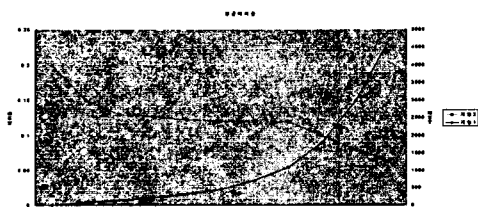
[표4.4]는 본 논문에서 제안하는 방법을 실험하기 위해서 [그림4.1]의 상태 전이 다이어그램을 보고 만든 데이터 값이다.

적용 예에 사용된 신경망의 구성은 다음과 같다.



[그림4.2] 적용 예에 사용된 신경망 구성도

[그림4.2]에 신경망의 구조는 입력층 1개 (2개 노드), 은닉층 1개 (4개 노드), 출력층 1개 (1개 노드)로 구성되어있다. 아래의 그림은 각각 사이클 회수에 따른 평균에러율과 입력층, 은닉층, 출력층에 대한 입력, 바이어스, 가중치 설정에 관한 것이다. (평균 에러율은 0.049993으로 설정하였다.)



[그림4.3] 사이클 회수에 따른 평균에러율

노드 0	입력층 : 입력	활성화 : 1.0000	바이어스 : 0.0000	에러 : 0.0000
노드 1	입력층 : 현재 상태	활성화 : 0.0000	바이어스 : 0.0000	에러 : 0.0000
노드 2	은닉층	활성화 : 0.0001	바이어스 : 0.2157	에러 : -0.0000
노드 3	은닉층	활성화 : 0.0000	바이어스 : 31.4192	에러 : -0.0000
노드 4	은닉층	활성화 : 0.0001	바이어스 : 14.6359	에러 : 0.0000
노드 5	은닉층	활성화 : 0.0001	바이어스 : -24.528932	에러 : -0.434369
노드 6	출력층 : 다음 상태	활성화 : 0.0000	바이어스 : 0.4452	에러 : -0.0000

[그림4.4] 입력층, 은닉층, 출력층에 대한 입력, 바이어스, 가중치 설정
위 실험은 [표4.5]와 같은 조건으로 수행되었다.

[표4.5] 훈련 매개변수

Fastest Error	0.07
Learning Rate	0.7
Momentum	0.5
Learning Cycle	4719
Input Number	6
State Number	6
Rule Number	39

[표4.6]은 실험 결과로 생성된 규칙이다.

[표4.6] 실험 결과 생성된 규칙

X(t)	S(t)	S(t+1)	0	3	4
0	4	4	4	1834	4
1	4	4	4	1862	5
2	4	4	4	1812	5
3	4	4	4	1120	5
4	4	4	4	0699	5
5	4	4	4	0461	4
0	6	4	4	1791	4
1	6	4	4	0394	4
2	6	4	4	8742	5
3	6	4	4	1020	5
4	6	4	4	0803	5
5	6	4	4	0484	5
0	2	4	0	1	1
1	2	4	2	1	1
2	2	4	2	1	1
3	2	6	3	1	1
4	2	4	4	1	1
5	2	4	5	1	1

위 [표4.6]에서 $X(t)$ 는 입력, $S(t)$ 는 현재 상태, $S(t+1)$ 는 다음 상태를 나타낸다.

예를 들어, 훈련 데이터를 학습시키고 난 뒤에 IF X(t) is 5 and S(t) is 6 then S(t+1) is 4,0484이다. 실험 결과로 나온 다음 상태값은 소수점 4째자리까지 나타낸 것이다.

5. 결론 및 향후 연구과제

본 논문은 다치이론을 적용한 다치오토마타 모델을 이용하여 무한개의 상태로 확장할 수 있는 가능성과 신경망으로 구현될 수 있음을 제안하였다. 여기서, 제안한 방법의 타당성을 증명하기 위하여 기존 오토마타의 입력값에 따라 상태변화를 다치오토마타 모델로 변환하여 각각의 6가지 입력과 상태 변화를 기존의 신경망 모델에 적용하여 실험하였다. 실험결과 초기에 훈련 데이터와 유사한 데이터가 생성됨을 볼 수 있었다.

따라서, 제안된 다치-신경망 시스템은 기존 오토마타 모델을 확장한 것이라 볼 수 있으며, 다치오토마타 모델로 기존 오토마타의 입력값에 따른 유한상태가 아닌 무한 상태까지도 적용 가능성을 제시하였다.

참고문헌

- [1] V. Drobot, "Formal Language and Automata Theory", Computer Science Press, 1989.
- [2] 조경은, 조형제, "방향코드와 유한오토마타를 이용한 사람 동작 프리미티브 패턴 분류기의 구현", 한국멀티미디어학회 99춘계학술발표논문집, pp. 428-432, 1999.
- [3] Doo-Ywan Kim, Chang-Sik Son, and Hwan-Mook Chung, "MVL-Automata for General Purpose Intelligent Model", The Second Korea-Japan Joint Symposium on Multiple-Valued Logic, Aug 27-29, 2001, pp. 41-44.
- [4] 임종석, 원성현, 정환목, "다변수-다치논리함수의 표현과 뉴런 모델", 한국정보처리학회 춘계학술발표 논문집, 제3권 제1호, pp. 564-567, 1996.
- [5] 정환목, "다치논리함수의 구조 해석과 전개", 한국정보과학회지, 제13권 제3호, pp. 153-166, 1986.
- [6] W. G. Wee, "On generalization of adaptive algorithm and application of the fuzzy sets concepts to pattern classification", Ph. D. dissertation, Purdue University, June 1967.