

정규문법과 동등한 일반화된 이진 이차 재귀 신경망*

정순호

부경대학교 전자컴퓨터통신공학부
(snow@pknu.ac.kr)

이 논문은 정규문법과 동등한 의미를 가지는 일반적인 이진 이차 재귀 신경망(Generalized Binary Second-order Recurrent Neural Networks: GBSRNN)의 구조 및 학습 방법을 제안하며 이를 이용하여 정규언어를 인식하는 어휘분석기 구현을 소개한다. GBSRNN는 성분들의 이진값 표현으로 정규문법과 동치인 모든 표현에 대하여 하드웨어로 표현할 수 있는 방법을 제공하며 정규 문법과의 구조적 관련성을 보여준다. 정규문법에서 심볼들의 개수 m , 비단말 심볼의 개수 p , 단말 심볼의 개수 q , k 인 문자열이 입력된다고 할 때, GBSRNN의 크기는 $O(m(p+q)^2)$ 이고 병렬처리 시간은 $O(k)$ 이며 순차처리 시간은 $O(k(p+q)^2)$ 이다.

논문접수일 : 2005년 10월

게재확정일 : 2006년 2월

교신저자 : 정순호

1. 서론

최근 음성인식, 음성생성, 영상인식, 로봇틱스, 그리고 자연언어처리 등의 분야에서 시간에 따라 변화하는 패턴들(time-varying patterns)을 인식하기 위하여 패턴들의 문법 추론에 재귀신경망회로(Recurrent Neural Network)를 이용하는 연구가 진행되고 있다[3][4][7-9][11-12][14-16]. 신경망의 문법 추론과 관련하여 문법과 신경망 사이의 연구들은 두 가지로 나누어 볼 수 있는데 하나는 문법에 동등한 신경망의 표현에 관한 연구이고 다른 하나는 입력 패턴들을 학습하여 문법을 추론하는 신경망의 구조에 관한 연구이다. 전자의 경우에 정규언어 인식과정을 유한 오토마타와 유사하게 그래프를 이용하여 수학적으로 표현하는 뉴로마타(neuromata)[22]가 소개되었다. 후자인 경우

는 재귀 신경망에 문자열들을 학습하여 정규 문법을 추론하기 위하여 1차 재귀 신경망(First-Order Recurrent Neural Network : FRNN) [20]이 처음 소개되었고 이 것을 보완하여 이중 입력을 가진 2차 재귀 신경망(Second-Order Recurrent Neural Network : SRNN)이 제안되었다[16]. 이후에 정규 문법의 추론상태를 정해진 수의 뉴론들에 학습하기 위하여 SRNN에 이산성을 부여한 이산 재귀 신경망(Discrete Recurrent Neural Network : DRNN)이 제안되었다[15]. 이 신경망은 제한된 수의 뉴론들에 많은 입력 패턴들을 학습하는 경우에 정규 문법의 정확한 추론을 보장하지 못하는 결점을 가지고 있다. 이 점을 극복하기 위하여 먼저 입력 패턴들로부터 신경망의 뉴론수를 결정하여 정규문법에 대응되는 이진 2차 재귀 신경망(Binary Second-order Recurrent Neural Network :

* 이 논문은 2000년도 부경대학교 발전기금의 지원에 의하여 연구되었음.

BSRNN)을 구성하는 방법이 제안되었다[21].

현재까지 신경망의 문법적 추론 연구에서는 패턴들의 인식을 위한 일반적 신경망의 구조와 학습 알고리즘의 찾기 위하여 제한적이고 소규모적인 패턴들을 대상으로 실험하는 방법론들이 제안되어 왔다. 이러한 신경망들은 시행착오적 실험들을 통하여 제안된 모델들이어서 문법과의 이론적 접근성이 부족하다. 본 연구에서는 정규문법 추론을 위한 신경망 구조의 설계를 위한 근본적인 연구로서 정규문법과 동치적인 신경망 모델을 제안하고 이 모델의 성분들이 이진성을 갖게 하므로 쉽게 논리 회로 설계가 가능하도록 하며 실제적인 예로서 미니 파스칼의 어휘 분석기를 구성한다.

앞으로 2장에서는 제안되는 신경망의 구조 및 특성을 이해 하기 위하여 필요한 이론들을 나타내고 3장에서는 제안하고자 신경망 모델을 정의하고 그 규모 및 성능 등에 대하여 평가하며 4장에서는 실제적인 구현으로 미니 파스칼에 대한 어휘 분석기를 이 신경망으로 나타낸다. 그리고 5장에서는 이 연구에 대한 결론과 향후 과제에 대하여 기술한다.

2. 관련 연구

이 장에서는 유한 오토마타, 이산 재귀신경망(Discrete Recurrent Neural Networks), 그리고 이진 이차 재귀 신경망(Binary Second-order Recurrent Neural Networks)을 간단히 살펴본다.

2.1 정규문법(Regular Grammar)과 유한 오토마타(Finite Automata)

유한 오토마타는 정규문법 그리고 정규표현과

동등한 관계이고 정규문법과 정규표현이 수학적이라면 유한 오토마타는 절차적이라 할 수 있다[1]. 따라서 정규문법이 나타내는 모든 문자열들은 오토마타에 의해 인식될 수 있으며 정규표현에 의해 나타내어질 수 있으며 역으로도 가능하다. 정규문법과 유한 오토마타의 정의는 각각 다음과 같다.

<정의 1-1>

유한 오토마타(Finite Automata)는 5개의 항목으로 구성된다.

$$G = (N, \Sigma, P, S),$$

여기서 N 은 미종료 심볼들(non-terminal symbols)의 유한 집합, Σ 는 입력 심볼들의 유한 집합, P 는 $A \rightarrow xB$ 또는 $A \rightarrow x$ 형태의 생성규칙(production)의 유한 집합이다. 이 때 $A, B \in N$ 이고 $x \in \Sigma$ 이다. S 는 문장 생성의 시작 심볼(start symbol)로서 $S \in N$ 이다.

<정의 1-2>

유한 오토마타(Finite Automata)는 5개의 항목으로 구성된다.

$$M = (Q, \Sigma, \delta, q_0, F),$$

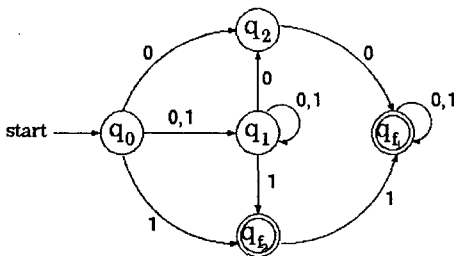
여기서 Q 는 상태들의 유한 집합, Σ 는 입력 심볼들의 유한 집합, δ 는 $Q \times \Sigma$ 에서 2^Q 로 대응하는 매핑 함수, 그리고 $q_0 \in Q$ 는 초기상태를 나타내고 $F \subset Q$ 는 최종상태들의 집합이다. 상태들의 모든 개수가 최종상태가 r 개라 할 때, $F = \{q_{f_0}, q_{f_1}, \dots, q_{f_{r-1}}\}$ 가 된다.

이 유한 오토마타는 비결정적(nondeterministic)이며 결정적(deterministic)인 경우를 포함한다. 유한 오토마타의 $M = (\{q_0, q_1, q_2, q_f, q_{f_1}, q_{f_2}\}, \{0, 1\}, \delta, q_0, \{q_f, q_{f_1}, q_{f_2}\})$,

$$\delta: \delta(q_0, 0) = \{q_1, q_2\}, \delta(q_0, 1) = \{q_f, q_{f_1}\}, \delta(q_1, 0) = \{q_1, q_2\}, \delta(q_1, 1) = \{q_f, q_{f_1}\},$$

$$\delta(q_2, 0) = \{q_f, q_{f_2}\}, \delta(q_2, 1) = \{q_f, q_{f_2}\}$$

의 상태 전이도는 [그림 1]과 같다.



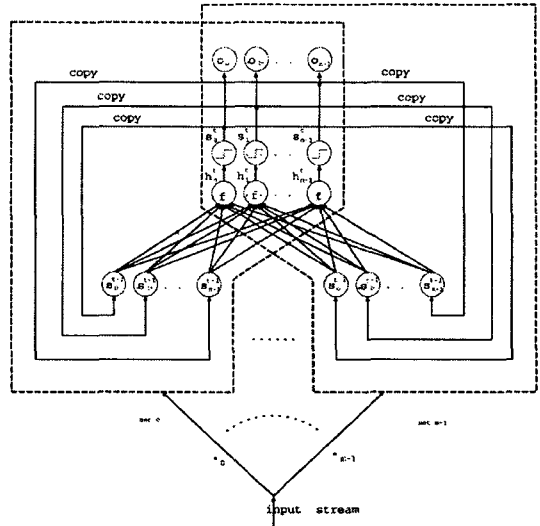
[그림 1] FA의 상태전이도

2.2 이산 재귀 신경망(Discrete Recurrent Neural Networks)

이산 재귀신경망은 [그림 2]와 같이 여러 재귀 신경망들로 구성되는데 각 재귀 신경망들(net i , $i=0, \dots, m-1$)은 서로 다른 입력 심볼에 의해 선택되어 동작된다. 신경망의 현재 상태들을 과거(前)의 상태들, 입력 심볼값, 그리고 가중치들에 의하여 다음과 같이 계산되어 결정된다; $s'_i = f(\sum_{j,k} w_{ij}^k s_j^{t-1} I_k)$.

여기서 I_k 는 시간 t 에 입력되는 k 번째 심볼의 입력 값(해당 심볼이면 0, 아니면 1)이고 w_{ij}^k 는 현재 시간 i 번째 상태 s'_i 와 바로 전 j 번째 상태 s_j^{t-1} 를 연결하는 가중치이며 상태값을 얻기 위해 적용되는 함수 f 는 다음과 같이 두 가지 값만을 갖도록 되어 있어 이산성(discreteness)의 성질을 갖는다.

$$f(x) = \begin{cases} 0.8 & \text{if } x > 0.5, \\ 0.2 & \text{otherwise} \end{cases}$$



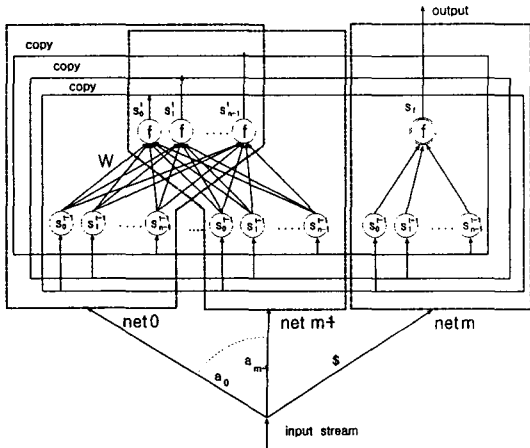
[그림 2] 이산재귀신경망(DRNN)

이렇게 동작하는 신경망은 여러 입력 스트림들을 학습하여 정규 문법의 규칙을 추론하도록 실험되었다[15].

2.3 이진 이차 재귀 신경망(Binary Second-order Recurrent Neural Networks)

이 모델은 수정된 유한 오토마타(Modified Finite Automata)-입력 문자열에 마지막 심볼로 '\$'가 반드시 존재하고 입력 심볼로 이 마지막 심볼이 나타날 때만 최종 상태에 도달할 수 있는 유한 오토마타 -에 동등한 이차 재귀 신경망 모델이다. 이 신경망은 [그림 3]과 같이 각 입력에 대응하는 net들로 구성된다. 마지막 심볼 '\$'을 제외한 모든 입력 심볼에 대응하는 net들은 공통적으로 전 상태들($s_0^{t-1}, s_1^{t-1}, \dots, s_{n-1}^{t-1}$)과 현재 상태들($s_0^t, s_1^t, \dots, s_{n-1}^t$)을 공유하고 net '\$'은 독자적으로 최종 상태 s_f 를 가지며 다른 net와 같이 전 상태들을 가진다. 모든 net는 각자의 가중치

$W^x = \{w_{00}^x, w_{10}^x, \dots, w_{n-10}^x, \dots, w_{n-1n-2}^x, w_{0n-1}^x, \dots, w_{n-1n-1}^x\}$ 를 가지고 현재 상태를 계산하는 함수 f 는 하드 리미터(hard limiter)이다.



[그림 3] 이진이차재귀신경망(BSRNN)

3. 일반화된 이진 이차 재귀 신경망 (Generalized Binary Second-order Recurrent Neural Networks: GBSRNN)

결정적인 경우를 포함해서 비결정적 유한 오토마타(FA)가 인식하는 입력 스트링들을 동등하게 분류할 수 있는 신경망을 표현하기 위해서 먼저 모델의 구조를 정의한다. 그리고 유한 오토마타로부터 동치적인 신경망을 이 정의를 이용해서 신경망으로 변환하는 알고리즘을 보여주고 이 유한 오토마타와 변환된 신경망사이의 동치적 관계를 증명한다.

3.1 GBSRNN의 구조

일반화된 BSRNN은 앞서 소개된 이진 이차 재

귀신경망(BSRNN)을 일반화하여 표현한 신경망으로 입력 심볼에 마지막 심볼을 고려하지 않는 일반적인 유한 오토마타에 동치적인 신경망 모델이다. 이 신경망은 각 입력 심볼에 대응되는 재귀 신경망들로 구성되며 이 들 각 재귀 신경망은 모두 공통적으로 같은 전 상태들과 현 상태들을 공유한다. 그리고 각 재귀 신경망들의 성분들- 즉 입력, 상태들, 연결강도, 출력, 그리고 함수 등-의 모든 값들이 이진수 (0 또는 1)로 표현된다. 이 일반화된 이진 이차 재귀 신경망(GBSRNN)의 정의는 다음과 같다.

<정의 2>

일반화된 이진 이차 재귀 신경망(GBSRNN)은 5개의 항목을 갖는다.

$$N = (I, S, S_{final}, f, W),$$

여기서

(1) $I = \{a_0, a_1, \dots, a_{m-1}\}$ 은 심볼들의 유한 집합이고 $a_i \neq a_j (i \neq j)$ 이고 $a_i \neq \epsilon$ 이다.

(2) $S = \{s_0, s_1, \dots, s_{p-1}, s_{f_0}, s_{f_1}, \dots, s_{f_{q-1}}\}$ 은 p개의 미종료 상태들(non-final states) s_0, s_1, \dots, s_{p-1} 과 q개의 종료 상태들(final states) $s_{f_0}, s_{f_1}, \dots, s_{f_{q-1}}$ 을 가진 유한 집합이다. 이 상태들은 시각 t에서 전 상태들(previous states)의 집합 S^{t-1} 과 현재 상태들(current states)의 집합 S^t 로 나타낼 수 있고 이 신경망의 최종 결과는 현재 시각 t의 종료상태들의 집합 $S_{final} \subset S^t$ 으로 나타낸다.

③ $S_{new} = S^t = \{s'_0, s'_1, \dots, s'_{p-1}, s'_{f_0}, s'_{f_1}, \dots, s'_{f_{q-1}}\}$

: 시각 t 에서의 현재 상태들의 집합

ⓑ $S_{old} = S^{t-1} = \{s_0^{t-1}, s_1^{t-1}, \dots, s_{p-1}^{t-1}, s_{f_0}^{t-1}, s_{f_1}^{t-1}, \dots, s_{f_{q-1}}^{t-1}\}$:

시각 t 에서의 전 상태들의 집합

ⓒ $S_{final} = \{s'_{f_0}, s'_{f_1}, \dots, s'_{f_{q-1}}\}$: 시각 t 에서의

종료상태들의 집합

여기서 s'_i 은 시각 t 에서 i 번째 상태를 나타내며 0 또는 1의 값을 나타낸다.

(3) f 는 뉴런 활성화 함수로 다음과 같은 OR 게이트이다.

$$f(x_0, x_1, \dots, x_{k-1}) = x_0 \vee x_1 \vee \dots \vee x_{k-1}$$

(4) 연결 강도들(weights)의 유한 집합은

$$\begin{aligned} W &= (W^{a_0}, W^{a_1}, \dots, W^{a_{m-1}}) \\ &= (W_{00}^{a_0}, W_{01}^{a_0}, \dots, W_{p+q-1, p+q-1}^{a_0}, \\ &\quad W_{00}^{a_1}, W_{01}^{a_1}, \dots, W_{p+q-1, p+q-1}^{a_1}, \\ &\quad \dots, \\ &\quad W_{00}^{a_{m-1}}, W_{01}^{a_{m-1}}, \dots, W_{p+q-1, p+q-1}^{a_{m-1}}) \end{aligned}$$

이다.

여기서 $W^x = (W_{00}^x, W_{01}^x, \dots, W_{p+q-1, p+q-1}^x)$ 는 심볼 x 가 입력되었을 때 현재 상태들 S^t 을 결정하기 위해 관련되는 가중치들의 집합이다. 원소 w_{ij}^x 는 입력 심볼이 x 일때 $s_i^{t-1} \in S_{old}$ 과 $s'_j \in S_{new}$ 사이의 연결 강도이고 그 값은 0 또는 1이다.

(5) 각 입력심볼에 대응되는 재귀 부신경망들(recurrent subnets) $net a_0, net a_1, \dots, net a_{m-1}$ 이 있으며 이 신경망에 $x \in I$ 가 입력되면 부신경망 $net x$ 가 enable되고 이 $net x$ 에 포함되어 있는 공통적 성분들 $S_{new}, S_{old}, S_{final}, f$ 과 가중치(weights) W^x 가 연결되어 동작한다.

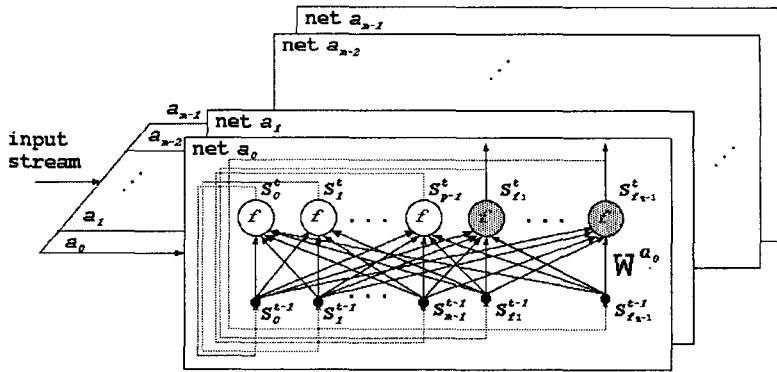
(6) 시각 t 에서의 현재 상태들 S^t 는 입력 심볼 x 을 받을 경우 전 상태들 S^{t-1} , 가중치 W^x , 함수 f 에 의해서 다음과 같이 결정된다.

$$s'_j = f(\sum_i w_{ij}^x s_i^{t-1})$$

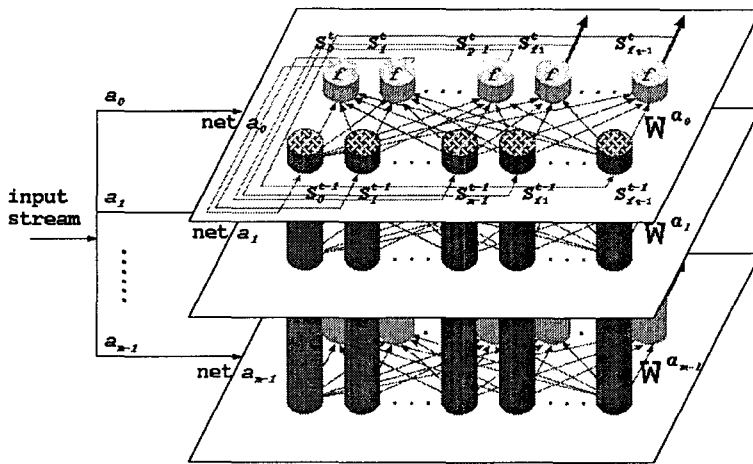
(7) 승인(acceptance) 또는 거부(rejection) : 임의의 문자열 $\alpha = x_0 x_1 \dots x_{l-1}$ 의 식별 여부를 판단하기 위하여 이 신경망의 초기 시각($t=0$)의 상태는 $(s_0^0, s_1^0, \dots, s_{p-1}^0, s_{f_0}^0, s_{f_1}^0, \dots, s_{f_{q-1}}^0) = (1, 0, \dots, 0)$ 으로 시작하여 문자열의 심볼들을 하나씩 차례대로 입력하면서 (6)의 동작을 문자열의 길이 l 만큼의 단계를 거쳐 이 GBSRNN의 결과를 얻는다. 결과로 $s'_{f_i} = 1$ 인 $s'_{f_i} \in S_{final}$ 가 존재하면 그 문자열을 승인하고 모든 $s'_{f_i} \in S_{final}$ 가 $s'_{f_i} = 0$ 이면 거부된다.

위 정의에 따라 구성된 일반화된 이진 이차 재귀 신경망(GBSRNN)의 구조는 [그림 4-a]과 같다. 각 입력 심볼에 의해 선택되어 동작되는 재귀 부신경망(recurrent subnet)이 입력 심볼의 개수만큼 m 개층으로 구성된다. 이 모든 부신경망들은 [그림 4-b]와 같이 입체적으로 살펴보면 전 상태들, 현재 상태들을 공유하고 있다.

GBSRNN의 크기는 내부에 사용된 주요 성분들의 개수를 확인하여 나타내볼 수 있다. 입력으로 사용되는 심볼들의 개수 m 에 따라 재귀 부신경망들이 있고 각 재귀 부신경망은 $2(p+q)$ 개의 상태들, $p+q$ 개의 함수들(OR gates), 그리고 $(p+q)^2$ 개의 가중치를 포함하게 된다. 따라서 GBSRNN의 크기는 $O(m(p+q)^2)$ 이다. GBSRNN의 동작은 입력되는 문자열의 길이 l 만큼 재귀적으로 이뤄지므로 GBSRNN의 병렬처리 실행시간



[그림 4-a] GBSRNN의 구조



[그림 4-b] GBSRNN의 입체적 구조

은 $O(l)$ 이고 순차처리 실행시간은 $O(l(p+q)^2)$ 이다.

3.2 FA로부터 GBSRNN구성

정의 1로 규정된 FA $M=(Q, \Sigma, \delta, q_0, F)$ 로부터 대응되는 GBSRNN $N=(I, S, S_{final}, f, W)$ 를 구성하는 방법은 다음과 같다.

<변환 알고리즘>

입력: FA $M=(Q, \Sigma, \delta, q_0, F)$

출력: BSRNN $N=(I, S, S_{final}, f, W)$

1. 입력 심볼들의 전환:

FA의 입력 심볼들의 집합 Σ 은 GBSRNN의 입력 심볼들의 집합 I 이 된다.

$$I \Leftrightarrow \Sigma$$

2. 상태들의 전환:

FA의 상태의 집합 Q 로부터 GBSRNN의 S

를 만든다. 그리고 S 로부터 S_{new} 과 S_{old} 를 만든다.

$$\forall i: q_i \in Q \Leftrightarrow s_i \in S$$

$$\text{i.e., } Q = \{q_0, q_1, \dots, q_{n-1}\} \Leftrightarrow \begin{cases} S = \{s_0, s_1, \dots, s_{n-1}\}, \\ S_{new} = \{s'_0, s'_1, \dots, s'_{n-1}\}, \\ S_{old} = \{s_0^{-1}, s_1^{-1}, \dots, s_{n-1}^{-1}\} \end{cases}$$

3. 최종 상태의 전환:

FA의 최종상태의 집합 F 으로부터 S_{final} 을 만든다.

$$\forall i: q_i \in F \Leftrightarrow s_i \in S_{final}$$

$$\text{i.e., } F = \{q_{f_0}, q_{f_1}, \dots, q_{f_{q-1}}\} \Leftrightarrow S_{final} = \{s_{f_0}, s_{f_1}, \dots, s_{f_{q-1}}\}$$

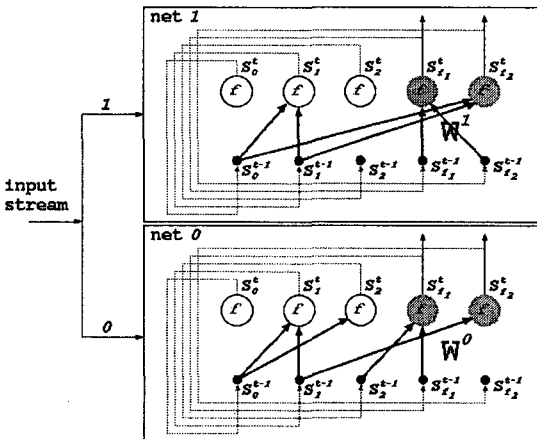
4. 학습 과정:

FA의 모든 입력 심볼에 대하여 매핑 함수의 관계로부터 연결 가중치들의 값을 정한다.

$$q_i \in \delta(q_j, a) \wedge a \in \Sigma \Leftrightarrow w_{ij}^a = 1$$

$$q_i \notin \delta(q_j, a) \wedge a \in \Sigma \Leftrightarrow w_{ij}^a = 0$$

이 변환 알고리즘을 [그림 1]의 FA M에 적용하면 변환된 GBSRNN N를 얻을 수 있으며 변환된 GBSRNN $N = (I, S, S_{final}, f, W)$ 의 구조는 [그림 5]



[그림 5] [그림 1]의 GBSRNN

와 같다. 여기서 $I = \{0,1\}$, $S = \{s_0, s_1, s_2, s_{f_1}, s_{f_2}\}$, $S_{final} = \{s_{f_1}, s_{f_2}\}$, 그리고 $W = W^0 \cup W^1$, $W^0 = (0,1,1,0,0,0,1,0,0,1,0,0,0,1,0,0,0,0,0,0)$, $W^1 = (0,1,0,0,1,0,1,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0)$ 이다.

3.3 GBSRNN과 FA의 동치성

위의 변환 알고리즘을 이용하여 유한 오토마타 (FA) M에 대응하는 일반화된 이진 이차 재귀 신경망(GBSRNN) N을 구성할 수 있다. 입력으로 사용된 FA M이 인식하는 언어 L(M)와 출력으로 구성된 GBSRNN N이 식별하는 언어 L(N)의 동치관계는 다음과 같이 증명된다.

정리 2:

유한 오토마타(FA) M이 주어졌을 때 변환 알고리즘에 의해 일반화된 이진 이차 재귀 신경망 (GBSRNN) N이 구성된다. 이 때, FA N과 GBSRNN M은 서로 동치이다. 즉, $L(M)=L(N)$ 이다. 증명) 변환 알고리즘의 입력으로 유한오토마타(FA) $M = (\{q_0, q_1, \dots, q_{p-1}, q_{f_0}, \dots, q_{f_{q-1}}\}, \{c_0, \dots, c_{m-1}\}, \delta, q_0, \{q_{f_0}, \dots, q_{f_{q-1}}\})$ 가 주어지면 출력된 GBSRNN은

$N = (\{c_0, \dots, c_{m-1}\}, \{s_0, s_1, \dots, s_{p-1}, s_{f_0}, \dots, s_{f_{q-1}}\}, \{s_{f_0}, \dots, s_{f_{q-1}}\}, f, W)$ 이 된다.

$L(M) = L(N)$ 임을 보이기 위해 $L(M) \subset L(N)$ 과 $L(M) \supset L(N)$ 임을 증명한다.

i) $L(M) \subset L(N)$:

$x \in L(M)$ 에 대하여, $x = a_0 a_1 \dots a_{L-1} (a_i \in \Sigma, L > 0)$ 라고 표현한다. FA M은 x 를 인식하므로 어떤 $q_{k_0} = q_0, q_{k_1}, \dots, q_{k_{L-1}}, q_{k_L} \in Q$ 과 $q_{k_L} = q_{f_i} \in F$ 에 대

하여 다음 전이 과정(transition process)을 거친다.

$$q_0 \xrightarrow{a_0} q_{k_1} \xrightarrow{a_1} q_{k_2} \cdots q_{k_{l-1}} \xrightarrow{a_{l-1}} q_{k_l}$$

이 것은 $q_{k_1} \in \delta(q_0, a_0), q_{k_2} \in \delta(q_{k_1}, a_1), \dots, q_{k_l} \in \delta(q_{k_{l-1}}, a_{l-1})$ 을 의미하고 변환 알고리즘에 의하여 GBSRNN의 가중치들 $w_{k_1 k_0}^{a_0} = 1, w_{k_2 k_1}^{a_1} = 1, \dots, w_{k_{l-1} k_{l-2}}^{a_{l-2}} = 1, w_{k_l k_{l-1}}^{a_{l-1}} = w_{f_l k_{l-1}}^{a_{l-1}} = 1$ 로 변환된다.

GBSRNN N는 초기상태 $(s_0^0, s_1^0, \dots, s_{p-1}^0, s_{f_0}^0, s_{f_1}^0, \dots, s_{f_{q-1}}^0) = (1, 0, \dots, 0)$ 로부터 $s_{k_0}^0 = 1 (k_0 = 0)$ 이고 가중치 $w_{k_0 k_0}^{a_0} = 1$ 가 존재하므로 $s_{k_0}^0 w_{k_0 k_0}^{a_0} = 1$ 이다. 따라서 시각 $t=1$ 일 때 $s_{k_1}^1 = f(\sum_j w_{k_1 j}^{a_0} s_j^0) = w_{k_1 0}^{a_0} s_0^0 \vee \dots \vee w_{k_1 p-1}^{a_0} s_{p-1}^0 \vee w_{k_1 f_0}^{a_0} s_{f_0}^0 \vee \dots \vee w_{k_1 f_{q-1}}^{a_0} s_{f_{q-1}}^0 = 1$ 가 된다.

이어서 시각 $t=2$ 일 때는 $w_{k_0 k_0}^{a_0} = 1$ 이고 $s_{k_1}^1 w_{k_2 k_1}^{a_1} = 1$ 이므로 $s_{k_2}^2 = f(\sum_j w_{k_2 j}^{a_1} s_j^1) = \dots \vee w_{k_2 k_1}^{a_1} s_{k_1}^1 \vee \dots = 1$ 이다. 일반적으로 시각 t 일 때는 $s_{k_{t-1}}^{t-1} = 1$ 이고 $w_{k_t k_{t-1}}^{a_{t-1}} = 1$ 이므로 $s_{k_t}^t = f(\sum_j w_{k_t j}^{a_{t-1}} s_j^{t-1}) = \dots \vee w_{k_t k_{t-1}}^{a_{t-1}} s_{k_{t-1}}^{t-1} \vee \dots = 1$ 가 된다.

이러한 과정으로 마지막 단계까지 진행된다면 즉, 시각 $t=L$ 일 때는 $s_{k_{L-1}}^{L-1} = 1$ 이고 $w_{k_L k_{L-1}}^{a_{L-1}} = 1$ 이므로 $s_{k_L}^L = f(\sum_j w_{k_L j}^{a_{L-1}} s_j^{L-1}) = \dots \vee w_{k_L k_{L-1}}^{a_{L-1}} s_{k_{L-1}}^{L-1} \vee \dots = 1$ 가 된다.

이것은 변환 알고리즘의 3에 의하여 $s_{k_L} = s_{f_L} \in F$ 이며 $s_{k_L} = 1$ 이므로 x 는 GBSRNN N에 의해서 인식된다.

이와 같이 모든 $x \in L(M)$ 는 $x \in L(N)$ 이다. 따라서 $L(M) \subset L(N)$ 이다.

ii) $L(N) \subset L(M)$:

$x \in L(N)$ 에 대하여 $x = a_0 a_1 \cdots a_{L-1} (a_i \in I, L > 0)$ 라고 하자. GBSRNN N는 다음과 같이 x 를 입력심볼들로 사용하여 초기 상태에서부터 최종 상태에 도달한다.

$$(s_0^0, s_1^0, \dots, s_{p-1}^0, s_{f_0}^0, s_{f_1}^0, \dots, s_{f_{q-1}}^0) = (1, 0, \dots, 0)$$

$$\xrightarrow{a_0 a_1 \cdots a_{L-1}} (s_0^L, s_1^L, \dots, s_{p-1}^L, s_{f_0}^L, s_{f_1}^L, \dots, s_{f_{q-1}}^L) = (0, 0, \dots, 0, x, \dots, 1, \dots, x)$$

여기서 $s_{f_i}^L \in F$ 가 1이라고 가정하고 x 는 1 또는 0을 의미한다.

최종상태 $s_{f_L}^L$ 로부터 초기상태 s_0^0 까지 역방향으로 1이 되는 적어도 한 개 이상의 경로가 존재하는데 그 중에 한 경로를 다음과 같이 나열할 수 있다; $s_{k_L}^L = s_{f_L}^L \xleftarrow{a_{L-1}} s_{k_{L-1}}^{L-1} \xleftarrow{a_{L-2}} \dots s_{k_1}^1 \xleftarrow{a_0} s_{k_0}^0 = s_0^0$.

다음 상태가 전 상태의 값1을 전달 받기 위해서는 인접된 상태들 사이에 입력심볼에 종속된 가중치들은 $w_{k_L k_{L-1}}^{a_{L-1}} = w_{f_L k_{L-1}}^{a_{L-1}} = 1, w_{k_{L-2} k_{L-2}}^{a_{L-2}} = 1, \dots, w_{k_2 k_1}^{a_1} = 1, w_{k_0 k_0}^{a_0} = 1$ 이 되어야 한다.

변환 알고리즘에 의해서 대응되는 FA M의 전이 과정은 $q_0 \xrightarrow{a_0} q_{k_1} \xrightarrow{a_1} q_{k_2} \cdots q_{k_{L-1}} \xrightarrow{a_{L-1}} q_{k_L} = q_{f_L}$ 이다. 따라서 $x = a_0 a_1 \cdots a_{L-1}$ 은 FA M에 의해서 인식된다. 즉, $x \in L(M)$ 이다. 이상의 결과로 $L(N) \subset L(M)$ 이다.

i)과 ii)에 의해서 $L(N) = L(M)$ 이다.

정리 2에 의하여 주어진 유한오토마타(FA) M과 그에 대응되는 일반화된 이진이차재귀 신경망(GBSRNN) N은 동치적이다. 이러한 결과는 정규

문법, 정규표현, 유한오토마타, 및 일반화 이진 이차 재귀 신경망 사이의 동등한 의미 관계를 다음과 같이 정리할 수 있다.

정리 3:

L이 어떤 알파벳 V로부터 이뤄지는 언어라 할 때, 다음은 모두 동등한 의미를 나타낸다.

1. L은 어떤 정규문법에 의해 생성된다.
2. L은 유한 오토마타에 의해 인식된다.
3. L은 정규표현에 의해 표현된다.
4. L은 일반화된 이진이차재귀신경망에 의해 식별된다.

3.4 GBSRNN의 논리회로 표현

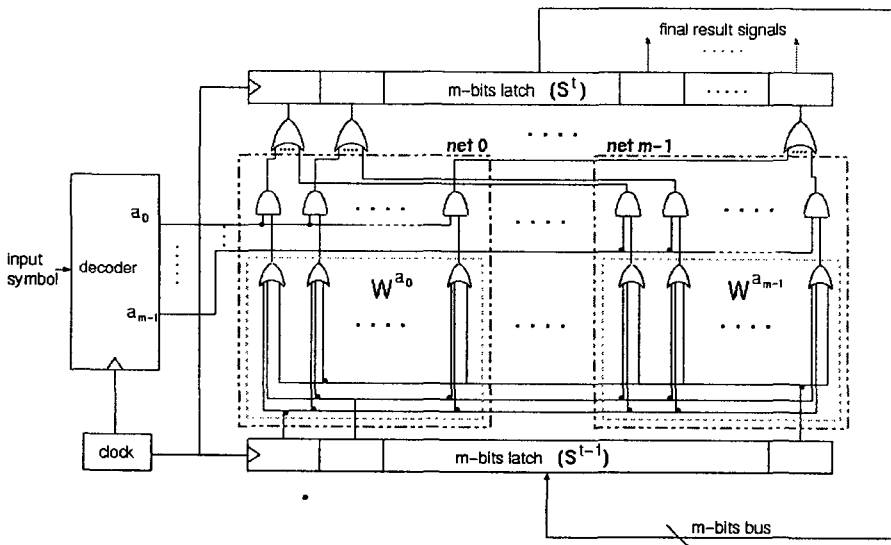
GBSRNN의 성분들의 값은 모두 이진수로 표현되므로 GBSRNN을 논리회로로 나타내면 <표 1>과 같이 상태들, 가중치들, 함수(OR gate)들, 입력 심볼 분류, 부신경망 enable들을 램치(latch), 선의

<표 1> GBSRNN의 논리 회로 성분들

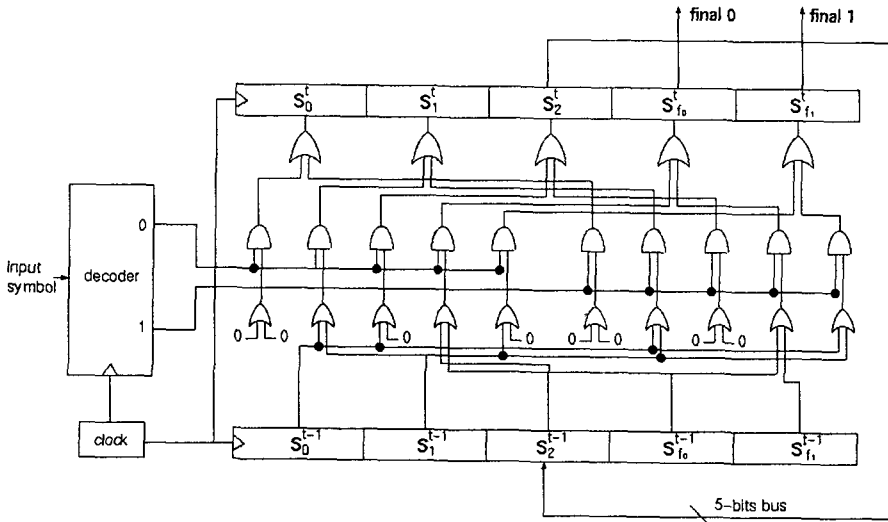
	GBSRNN의 성분	논리회로
1	상태들	램치(latch)
2	가중치들	선의 연결 여부
3	함수	OR gate
4	입력 심볼 분류	디코더(decoder)
5	부신경망 enable	AND gate

연결 여부, OR 게이트, 디코더, AND 게이트등의 논리회로로 전환하여 간단하게 설계할 수 있다.

GBSRNN의 기본적인 논리회로의 표현은 <표 1>에서 언급한 성분들을 이용하여 구성하면 [그림 6]과 같이 나타낸다. 상위 램치, 하위 램치, 그리고 입력심볼들을 분류하는 디코더는 클럭에 의해서 동기되고 있다. 상위 램치의 값들이 하위 램치에 전달되고 하위 램치의 값들이 모든 부신경망의 가중치들로서 사용되어 OR게이트에 입력되고 출력된 값들은 입력 심볼 분류 디코더로 부터의



[그림 6] GBSRNN의 논리 회로



[그림 7] [그림 5]의 GBSRNN 논리회로

출력이 한 개의 부신경망을 선택한다. 이 부신경망의 출력값들이 최종OR게이트의 입력값들로 사용되고 이 결과 값들이 상위 래치에 전달된다.

적용 예로서 [그림 5]의 GBSRNN의 논리회로는 [그림 7]과 같다.

4. GBSRNN 어휘분석기의 구현

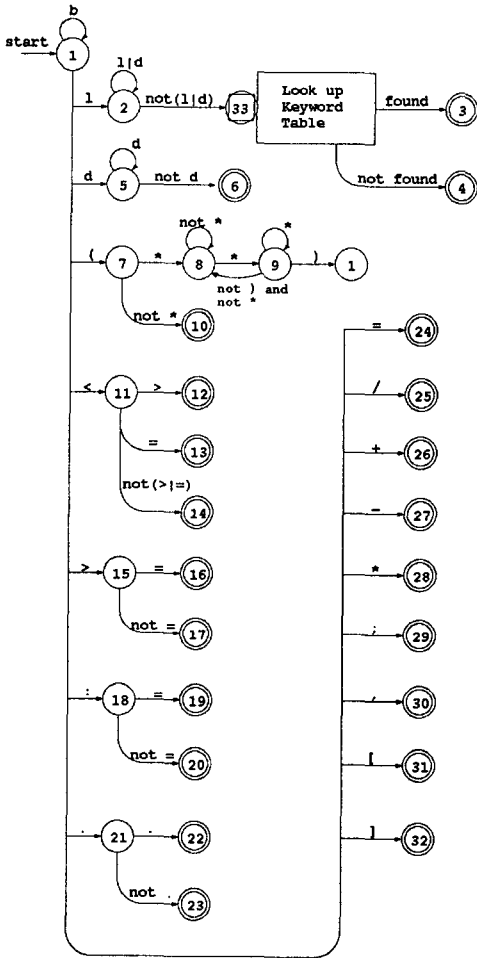
이 장에서는 정규표현, 정규문법과 동등한 유한 오토마타의 표현을 변환 알고리즘에 따라 동등한 GBSRNN를 구현한다. 먼저 미니 파스칼 언어의 어휘 분석기의 유한 오토마타 표현을 살펴보고 다음으로 이에 대응되는 GBSRNN을 구성 과정을 보인다.

4.1 미니 파스칼의 어휘분석기의 FA

미니 파스칼로 작성된 프로그램의 토큰을 생성

하기 위한 어휘분석과정의 FA는 [그림 8]과 같다 [23].

이 FA는 33개의 상태를 가지고 있고 상태 전이에 사용되는 심볼들은 b(빈칸), l(영문자), d(숫자), (, *,), <, >, =, :, ,, /, +, -, *, ;, comma, [,], 그리고 그 밖의 심볼들로 포함하여 19개의 문자들이다. 이 FA의 동작은 상태1로부터 시작하여 입력 심볼에 따라 다음 상태로 전이되고 최종 상태에 도달되면 이제 까지 입력된 심볼들이 토큰으로 결정된다. 그리고 FA는 다시 상태 1로부터 시작된다. 최종 상태에서 초기 상태로 돌아갈 때, 사용된 입력심볼을 다음 초기 상태에서 다시 사용하도록 하는 (현재 심볼을 가리키는 포인터를 전진하지 않는다.) 경우가 있는데 상태33, 6, 10, 14, 17, 20, 23가 이에 해당한다. 상태33은 '키워드 찾기' 처리 과정이 결부되어 있어 특별한 행동 처리를 요구한다. 따라서 이 FA는 몇 가지 상태들에 대하여 일반적인 상태전이 과정 외에 특별한 행동을 처리하는 단계를 필요로 한다.



[그림 8] 어휘분석 상태전이도

4.2 어휘분석 GBSRNN의 구성

앞서의 어휘분석기 FA에 대하여 변환 알고리즘을 적용하여 GBSRNN을 구성해 보면 33개의 상태(뉴런)들이 형성되고 19개의 입력심볼 부류를 사용하며 연결 가중치들은 FA의 상태들의 전이 내용을 반영하여 1과 0 (on과 off)로 정해진다. 입력 심볼들의 19개 부류에 따라 이에 대응되는 19개의 재귀 부신경망(recurrent subnet)들이 형성

되고 각 부신경망은 지정된 심볼이 입력으로 나타날 때 동작이 허용되는 내부 성분들 - 과거 상태들, 현재 상태들, 과거와 현재 상태들 사이를 연결하는 연결선들, 현재 상태 값을 출력하기 위한 함수 f , 그리고 현재 상태를 전의 상태로 참조하기 위한 feedback라인-로 구성된다. 상태들, 함수, 피이드 백의 구성은 앞서 언급한 바와 같이 간단하게 구성될 수 있다. 연결선의 구성은 FA의 심볼들과 상태들의 전이 관계를 나타내는 전이 함수의 내용을 입력심볼 별로 정리하여 학습과정으로서 가중치들을 결정하게 되므로 이뤄진다. 이 학습에 의해 결정된 연결선은 강도가 1인 경우는 전상태와 현재상태를 연결하는 것을 의미하며 이를 두 상태 사이에 화살표로 나타내고 0인 경우는 전상태와 현재상태 사이에 연결이 끊긴 것을 의미하며 화살표 표시를 하지 않는다. 즉, 각 입력 심볼들에 대한 부신경망 안의 현재상태(s'_j)와 전상태(s'_{j-1}) 사이의 연결(w'_{ij}) 강도들이 FA의 매핑 함수로부터 얻어져서 표시될 수 있다. <표 2>는 연결 상태를 전상태와 현재상태의 연결관계 ($i \rightarrow j$)로 나타낸 것이며 이 것에 의하여 동등한 GBSRNN의 연결 가중치를 결정하는 학습 내용이 된다.

이 표의 내용에 따라 GBSRNN을 구성하면 [그림 9]와 같이 각 입력 심볼에 대응하는 부신경망들 19개가 구성되고 각 부신경망 안에는 9~10개의 연결선을 가지게 됨을 볼 수 있다. 입력 심볼이 들어오면 그 부류에 따라서 해당되는 부신경망이 선택되어 활성화된다. 이 활성화 된 부신경망에는 33개 상태들에 대한 현재 상태(s'_j)와 전상태(s'_{j-1})의 값들을 갖도록 유지한다. 이 상태들의 초기값은 (1,0,0,...,0)로부터 시작해서 순차적으로 입력되는 심볼에 따라서 활성화된 상태들을 변화시켜

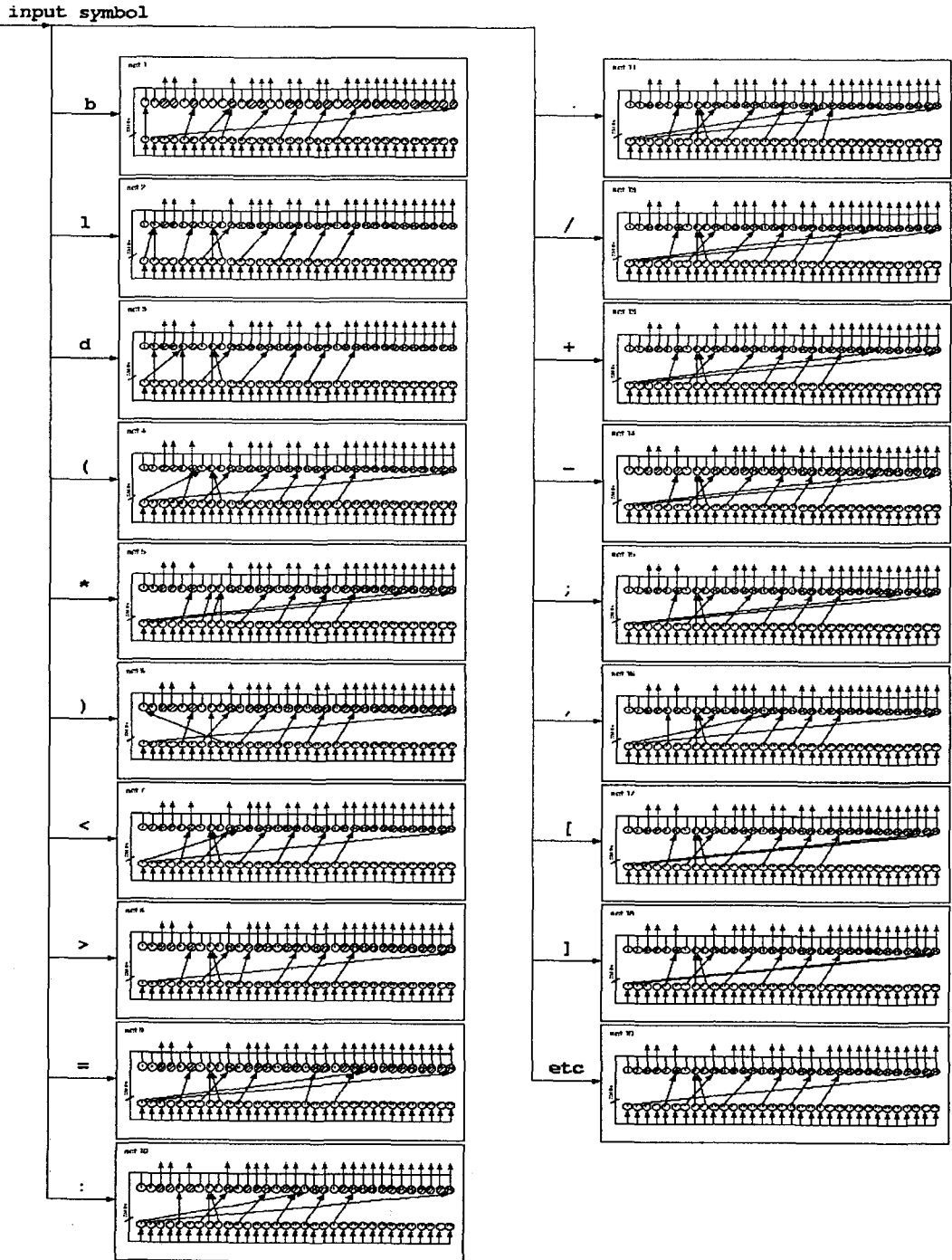
<표 2> 어휘분석 GBSRNN의 학습 상태

Net 번호	입력심볼 x	$w_{ij}^x = 1$ 인 연결 상태들 : ($i \rightarrow j$)
1	b (blank)	1→1, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
2	l (letters)	1→2, 2→2, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
3	d (digits)	1→5, 2→2, 5→5, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
4	(1→7, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
5	*	1→28, 2→33, 5→6, 7→8, 8→9, 9→9, 11→14, 15→17, 18→20, 21→23
6)	2→33, 5→6, 7→10, 8→8, 9→1, 11→14, 15→17, 18→20, 21→23
7	<	1→11, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
8	>	2→33, 5→6, 7→10, 8→8, 9→8, 11→12, 15→17, 18→20, 21→23
9	=	1→24, 2→33, 5→6, 7→10, 8→8, 9→8, 11→13, 18→19, 21→23
10	:	1→18, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
11	.	1→21, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→22
12	/	1→25, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 18→20, 15→17, 21→23
13	+	1→26, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
14	-	1→27, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
15	;	1→29, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
16	,	1→30, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
17	[1→31, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
18]	1→32, 2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23
19	그 밖의 문자들 (other chars)	2→33, 5→6, 7→10, 8→8, 9→8, 11→14, 15→17, 18→20, 21→23

간다. 이렇게 진행되다가 지정된 최종상태가 활성화($s'_j = 1$)되면 최종결과 단자(final result signal)가 1이 되고 이에 관련된 특별한 행동처리를 지시한다. 어떤 최종상태가 활성화된다는 것은 어휘분석기에서 한 개의 토큰이 생성되는 것을 나타내므로 이때 토큰을 처리하기 위한 행동 프로시저(procedure)가 필요하고 다음 상태들의 초기화와 입력 심볼의 재사용을 결정해야 된다. 이 것은 GBSRNN의 상태전이 타이밍을 늦추거나 다음 입력심볼의 출현을 지연하여 현재 사용된 심볼을 재사용하도록 할 수도 있어야 한다. 이러한 최종상태들에 대한 행동처리 프로시저는 다음과 같이 구성된다.

4.2.1 최종상태의 토큰 처리

최종상태에 대한 일반 토큰처리는 입력부에서 입력 심볼에 대한 버퍼에 저장하다가 최종상태에서 도달했을 때 이 저장된 내용을 사용할 수 있도록 한다. 그러나 특별한 처리를 요구하는 경우인 상태 33인 경우에 이 상태가 활성화되면 모든 입력 심볼의 입력 동기(sync)가 정지되고 버퍼에 저장된 단어를 키워드 테이블에서 찾아보는 처리를 요구한다. 그 결과에 따라서 성공과 실패를 상태 3이나 상태 4를 활성화 시켜 알려주게 된다. 그러나 이 상태에 대한 구현은 필요하지 않다.



[그림 9] 어휘분석 GBSRNN

4.2.2 상태들의 초기화 및 입력심볼의 재사용 결정

모든 최종상태는 예외없이 필요한 동작을 완료한 후에 다음 상태들의 값을 초기화(1,0,0, ..., 0) 한다. 한편, 현재에 입력된 심볼을 재사용하는 것의 여부는 이 심볼이 현재의 토큰에 포함되는 경우에는 새로운 입력 심볼을 요구하고 포함되지 않는 경우에는 재사용을 요구한다. 재사용을 요구하는 상태의 번호는 33(3, 4포함), 6, 10, 14, 17, 23에 해당하고 나머지 12, 13, 16, 19, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32는 새로운 입력 심볼을 요구한다.

4.3 GBSRNN 어휘 분석기의 실험

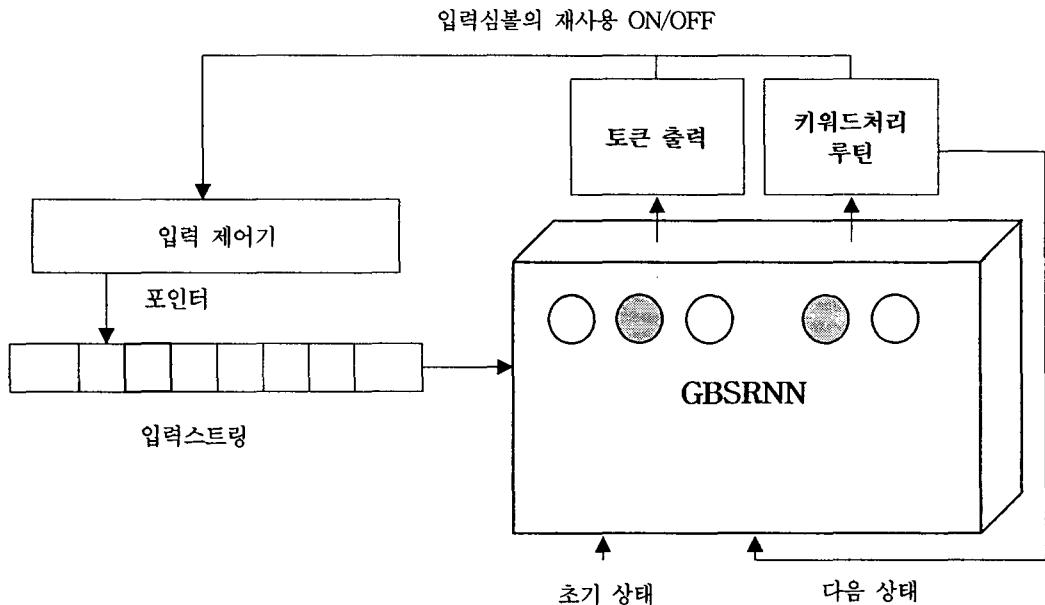
미니 파스칼의 어휘분석 오토마타를 학습하여 생성된 GBSRNN는 [그림 10]과 같이 몇 가지 특별한 신호단자들에 연결된 처리모듈과 협동하여 동작한다. GBSRNN는 처음 초기 상태로부터 시

작하고 입력스트링의 포인터가 지시하는 심볼들을 입력으로 받아들여 GBSRNN의 내부 동작이 이뤄진다. 이 동작의 상태가 출력 신호 단자들을 통해서 나타나고 그 결과에 따라 키워드 처리 루틴의 실행, 입력 심볼의 재사용을 위한 입력 제어, 그리고 토큰이 결정될 때 마다 토큰의 내용을 알려주는 토큰 출력과 결합되어 동작한다.

이 어휘분석GBSRNN의 시뮬레이션 실험은 C++로 구현하여 실행되었고 미니 파스칼로 표현할 수 있는 여러 샘플 프로그램들을 대상으로 실험하여 각 프로그램의 토큰들이 프로그램으로 구현하여 실행한 결과와 같음을 확인할 수 있다.

5. 결론

이 논문은 가장 낮은 수준의 문법인 정규문법에 대응되는 GBSRNN을 정의하고 정규표현 또는



[그림 10] GBSRNN을 포함한 어휘분석기

유한 오토마타를 동등하게 인식할 수 있음을 보인다. 이 GBSRNN 은 모든 성분들의 이진화 표현으로 논리 회로 설계가 가능하며 유한 오토마타의 예제로 미니 파스칼의 어휘분석기를 이 신경망으로 구현함으로써 실제적인 하드웨어의 구성이 가능함을 보인다.

제안된 GBSRNN의 특징은 첫째로 정규문법, 정규문법, 유한 오토마타와 동등한 의미의 또 다른 논리적 표현 방법을 제안하며 둘째로 이 재귀 신경망은 일반적인 정규 언어 학습을 위한 유연한 문법적 추론 신경망의 설계에 참조가 될 수 있는 기초적 구조를 제시하고 셋째로 이 재귀 신경망을 직접인 논리회로로 간단히 설계할 수 있으므로 정규문법으로 나타낼 수 있는 동등한 표현들을 하드웨어로 설계할 수 있는 방법을 제공한다.

향후 이 신경망은 정규문법과의 동치성을 가지게 되므로 정규문법 추론을 위한 신경망의 구조 및 학습 알고리즘 연구에 도움을 줄 수 있을 뿐만 아니라 한 차원 높은 문법인 문맥 자유 문법(context-free grammar)의 문장들의 처리를 위한 신경망 구조의 연구에 도움을 줄 수 있을 것이다.

참고문헌

- [1] A.V. Aho and J.D. Ullman, The theory of parsing, translation, and compiling, Prentice-Hall, 1972.
- [2] A.D. Back and A.C. Tsoi, "FIR and IIR synapses. A new neural network architecture for time series modeling," *Neural Comp.*, vol.3(1991), 375-385.
- [3] A. Cleeremans, D. Servan-Schreiber, and J.L. McClelland, "Finite state automata and simple recurrent networks," *Neural Comp.*, vol.1, no.3(1989), 327-381.
- [4] J.L. Elman, "Finding structure in time," *Cognitive Science*, vol.14(1990), 179-211.
- [5] P. Frasconi, M. Gori, and G. Soda, "Local feedback multilayered networks," *Neural Comp.*, vol.4(1992), 120-130.
- [6] K. Fu and T.L. Booth, "Grammatical inference: Introduction and survey," *IEEE Trans. Pattern Anal. & Machine Intell.*, vol.8, no.3(May 1986), 343-375.
- [7] C.L. Giles, C.B. Miller, D. Chen, H.H. Chen, G.Z. Sun, and Y.C. Lee, "Second-order recurrent neural networks," *Neural Comp.*, vol.4, no.3(1972), 393-405.
- [8] C.L. Giles, and C.B. Miller, "The effect of higher order in recurrent neural networks: Experiments," *Artificial Neural Networks for Speech and Vision*, Chapman & Hall (1993), 54-78.
- [9] C.L. Giles, C.B. Miller, D. Chen, H.H. Chen, G.Z. Sun, and Y.C. Lee, "Learning and Extracting finite state automata with second-order recurrent neural networks," *Comm. ACM*, vol.4, no.3(1992), 393-405.
- [10] S. Mitra and S.K. Pal, "Fuzzy multi-layer perceptron, inference and rule generating," *IEEE Trans. Neural Networks*, vol.6, no.1(1995), 51-63.
- [11] C.W. Omlin and C.L. Giles, "Extraction of rules from discrete-time recurrent neural networks," *Neural Networks*, vol.9, no.1(1996), 41-52.
- [12] S. Porat and J.A. Feldman, "Learning automata from ordered examples," *Machine Learning*, vol.7(1991), 109-138.
- [13] S. Crespi-Reghezzi, M.A. Melkanoff, and L. Lichten, "The use of grammatical inference for designing programming languages,"

- Comm. ACM, vol.16, no.2(1997), 83-90.
- [14] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comp.* vol.1, no.2(1989), 270-280.
- [15] Z. Zeng, R.M. Goodman, and P. Smyth, "Discrete recurrent neural networks for grammatical inference," *IEEE Trans. Neural Networks*, vol.5, no.2(1994), 320-330.
- [16] Z. Zeng, R.M. Goodman, and P. Smyth, "Learning finite state machine with self-clustering recurrent networks," *Neural Comp.*, vol.5, no.6(1993), 976-990.
- [17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol.5, no.2(1994), 157-166.
- [18] S. Hui and S.H. Zak, "The Widrow-Hoff algorithm for McCulloch-Pitts type neurons," *IEEE Trans. Neural Networks*, vol.5, no.6(1994), 924-929.
- [19] R.J. Mammone, "Artificial Neural Networks for Speech and Vision," Chapman & Hall, 1994.
- [20] M.W. Goudreau, C.L. Giles, S.T. Chakradhar, and D. Chen, "First-order versus second-order single-layer recurrent neural networks," *IEEE Trans. Neural Networks*, vol.5, no.3(1994), 511-513.
- [21] S. Jung and H. Yoon, "Binary Second-Order Recurrent Neural Networks for Inferring Regular Grammars," *IEICE Trans. on Info. and Sys.*, vol.E83-D, no.11(2000), 1996-2007.
- [22] Jiří Šíma and Jiří Wiedermann, "Theory of Neuraomata," *Jour. Of ACM*, vol.45, no.1(Jan 1998), 155-178.
- [23] 오세만, 컴파일러 입문, 정익사, 2002.

Abstract

Generalized Binary Second-order Recurrent Neural Networks Equivalent to Regular Grammars

Jung Soon-Ho*

We propose the Generalized Binary Second-order Recurrent Neural Networks(GBSRNN) being equivalent to regular grammars and show the implementation of lexical analyzer recognizing the regular languages by using it. All the equivalent representations of regular grammars can be implemented in circuits by using GBSRNN, since it has binary-valued components and shows the structural relationship of a regular grammar. For a regular grammar with the number of symbols m , the number of terminals p , the number of nonterminals q , and the length of input string k , the size of the corresponding GBSRNN is $O(m(p+q)^2)$ and its parallel processing time is $O(k)$ and its sequential processing time, $O(k(p+q)^2)$.

Key words : 일반화된 이진 이차 재귀 신경망(Generalized Binary Second-order Recurrent Neural Networks: GBSRNN), 정규문법, 유한 오토마타, 어휘 분석기

* Div. of Electronic, Computer & Telecommunication Engineering, Pukyong National University