

이진 삼차 재귀 신경망과 유전자 알고리즘을 이용한 문맥-자유 문법의 추론

정 순 호*

Inference of Context-Free Grammars using Binary Third-order Recurrent Neural Networks with Genetic Algorithm

Soon-Ho Jung*

요 약

이 논문은 이진 삼차 재귀 신경망(Binary Third-order Recurrent Neural Networks: BTRNN)에 유전자 알고리즘을 적용하여 문맥-자유 문법을 추론하는 방법을 제안한다. BTRNN은 각 입력심볼에 대응되는 재귀 신경망들의 다층적 구조이고 외부의 스택과 결합된다. BTRNN의 매개변수들은 모두 이진수로 표현되며 상태 전이와 동시에 스택의 한 동작이 실행된다. 염색체로 표현된 BTRNN들에 유전자 알고리즘을 적용하여 긍정과 부정의 입력 패턴들의 문맥-자유 문법을 추론하는 최적의 BTRNN를 얻는다. 이 방법은 기존의 신경망 이용방법보다 적은 학습량과 적은 학습회수로 작거나 같은 상태 수를 갖는 BTRNN을 추론한다. 또한 문법 표현의 염색체 이용방법보다 parsing과정에서 결정적인 상태전이와 스택동작이 실행되므로 입력 패턴에 대한 인식처리 시간복잡도가 우수하다. 문맥-자유 문법의 비단말 심볼의 개수 p , 단말 심볼의 개수 q , 그리고 길이가 k 인 문자열이 입력이 될 때, BTRNN의 최대 상태수가 m 이라고 하면, BTRNN의 인식처리 병렬처리 시간은 $O(k)$ 이고 순차처리 시간은 $O(km)$ 이다.

▶ Keyword : 이진삼차 재귀신경망, 문맥-자유 문법, 유전자 알고리즘

Abstract

We present the method to infer Context-Free Grammars by applying genetic algorithm to the Binary Third-order Recurrent Neural Networks(BTRNN). BTRNN is a multiple-layered

• 제1저자 : 정순호

• 투고일 : 2011. 12. 14, 심사일 : 2012. 01. 26, 게재확정일 : 2012. 02. 21.

* 부경대학교 컴퓨터공학과(Dept. of Computer Engineering, Pukyong National University)

※ 이 논문은 2003학년도 부경대학교 연구년 교수 지원사업에 의하여 연구되었음

architecture of recurrent neural networks, each of which is corresponding to an input symbol, and is combined with external stack. All parameters of BTRNN are represented as binary numbers and each state transition is performed with any stack operation simultaneously. We apply Genetic Algorithm to BTRNN chromosomes and obtain the optimal BTRNN inferring context-free grammar of positive and negative input patterns. This proposed method infers BTRNN, which includes the number of its states equal to or less than those of existing methods of Discrete Recurrent Neural Networks, with less examples and less learning trials. Also BTRNN is superior to the recent method of chromosomes representing grammars at recognition time complexity because of performing deterministic state transitions and stack operations at parsing process. If the number of non-terminals is p , the number of terminals q , the length of an input string k , and the max number of BTRNN states m , the parallel processing time is $O(k)$ and the sequential processing time is $O(km)$.

▶ Keyword : Binary Third-order Recurrent Neural Networks, Context-Free Grammars, Genetic Algorithm

1. 서론

음성인식, 영상인식, 로보틱스, 자연언어 처리 등의 분야에서 시간에 따라 변화하는 패턴들을 인식하고 생성하기 위하여 패턴들의 문법적 추론을 수행하기 위한 많은 연구가 이뤄져 왔다[1-2]. 문법적 추론 방법으로 문장구조를 분석하는 고전적 방법[3]으로부터 신경망의 적용방법[1-11], 유전자 알고리즘의 적용 방법[12-15] 과 유전자 프로그래밍의 접근 방법[16] 등이 제안되어 있다. 이렇듯이 최근에 진화적 신경망이 광범위하게 여러 분야에 사용되고 있고[11] 최적의 결과를 찾기 위해 현대적 기법으로 진화적 알고리즘이 적용되고 있다[13].

문법 추론에 적용된 많은 알고리즘들 중에서 수렴하는 (converging) 알고리즘, 즉 Delta rule 학습 알고리즘 등에 의해 효과적으로 학습될 수 있는 가장 복잡한 언어는 Chomsky의 언어 분류[17]에서 type 3인 정규언어이고 상위 부류의 언어들 중 type 2인 문맥-자유 언어를 위한 최신의 접근방법들이 제한적인 성공을 보여주는 상황이어서 문맥-자유 문법의 학습의 완성은 문법 추론에 대한 실질적인 도전이 된다[16].

문법 추론에 대한 최근의 진화적 접근 방법들을 그 구조와 결합 알고리즘 측면에서 살펴보면 아날로그 이차 재귀신경망 (analog second-order network)[8][18] 은 유한 오토마타의 성질을 전형적인 재귀신경망의 구조로서 매개변수들(가중치, 상태, 함수값)에 실수 범위의 값을 표시하고 gradient-descent algorithm을 사용하여 정규언어를 학습하고자 하였고 이산 이

차 재귀신경망(discrete second-order recurrent network)[8] 은 아날로그 이차 재귀 신경망의 학습 알고리즘에서 이산화 함수 (discretization function)[8]를 채택하여 상태의 값만을 두 개의 실수 값으로 이산화하여 학습함으로써 정규문법의 인식률을 높였고 외부에 실수 값으로 나타내는 스택을 결합하여 문맥-자유 문법을 인식하려는 시도를 하였다. 이진 이차 재귀 신경망(binary second-order recurrent network) [10] 은 이산 재귀 신경망의 구조에서 가중치를 포함한 모든 매개변수들의 값을 0과 1로 대체하여 표현하였고 각 상태들 사이의 연결 가중치들을 이진수로 표시하였다. 이 신경망의 학습 방법은 입력 문자열들을 구조적으로 분석하여 상태들의 연결 상태를 결정한다. 문법의 검색체 표현에 유전자 알고리즘을 이용한 접근방법[13][20]은 문법의 생성규칙들을 유전자로 나열하여 검색체로 표현하여 개체군을 형성하고 이 개체군에 유전자 알고리즘을 적용하여 PDA시뮬레이션을 통하여 적합도를 측정하여 유전자 알고리즘의 과정을 수행하여 최적의 문법을 얻는다. 유전자 프로그래밍(genetic programming)을 이용한 문맥-자유 문법의 유도 방법[16]은 문법의 생성규칙들을 구조화된 트리로 만들고 이 것을 리스트들로 개체군을 이루고 입력된 문자열들을 parsing하는 방법으로 CYK알고리즘을 적용하여 유전자 알고리즘과 같은 진화과정을 거치는 방법론을 제안하였으나 부분적인 실험에 그쳐있고 최적의 해를 보장하지 못하고 있다.

기존의 문맥-자유-문법 추론의 접근방법들은 크게 신경망을 이용한 접근법과 진화적 알고리즘을 이용한 접근법의 2가지 측면으로 나뉘 볼 수 있다. 이 두 측면의 문제점을 살펴 보면 먼저 신경망 접근방법[8]에서는 학습 알고리즘이 입력패

턴에 대한 인식을 높이기 위하여 내부 상태들 사이의 가중치들의 변화에 관심을 두고 내부 상태수의 최적화에 대한 고려가 없다. 최적화되지 않은 내부 상태들은 목적 문법에 속하지 않는 입력패턴에 대하여 인식을 승인하는 오류의 가능성이 높아진다. 진화적 알고리즘을 이용하는 접근 방법들 [13][20][16]에서는 문법의 생성규칙을 염색체내의 유전자로 표현하여 유전자 연산(교배, 돌연변이)을 하도록 하고 우수한 개체들의 선택을 위해 적합도를 산정할 때 입력패턴들의 parsing과정에서 푸쉬다운 오토마타(PDA)를 시뮬레이션하게 된다[13]. 이 PDA시뮬레이션은 parsing과정에서 생성규칙을 적용하면서 backtracking이 발생하게 된다. 또 다른 진화적 알고리즘인 유전자 프로그래밍의 접근방법은 정규문법에 대한 실험은 완성되었으나 문맥-자유 문법에 대하여 미완의 실험상태이고 최적의 해를 보장하지 못하고 있다[16].

이 논문에서는 새로운 접근방법으로 이진 삼차재귀 신경망(Binary Third-order Recurrent Neural Network: BTRNN)의 표현 구조를 소개하고 신경망의 일반적인 학습 알고리즘을 사용하지 않고 유전자 알고리즘을 적용하여 문맥-자유 문법을 추론할 수 있는 방법을 제안한다. BTRNN의 사용은 신경망의 동작에서 결정적(deterministic) 선택을 하게 하는 특성을 가지고 있으며 유전자 알고리즘에 의해서 입력패턴의 집합들을 만족하는 최적의 BTRNN을 생성한 결과를 보여 준다.

이 논문은 2장에서 관련 연구에 대해서 언급하고 3장에서는 제안된BTRNN의 모델을 정의하고, 4장에서는 BTRNN의 문법적 추론 방법을 설명하고, 5장은 제안된 모델과 알고리즘의 실험 내용과 그 결과를 보여주고 6장에서는 결론을 맺는다.

II. 관련 연구

문법적 추론에 관련하여 먼저 신경망의 종류로 BTRNN와 구조적으로 관련된 DRNN과 BSRNN을 소개하고 다음으로 문법을 염색체로 사용한 유전자 알고리즘을 소개한다.

1. 이진 재귀신경망(DRNN)

이산 재귀신경망(Discrete Recurrent Neural Networks)은 그림 1과 같이 여러 재귀 신경망들로 구성되고 각 재귀 신경망들 (net i, i=0, ..., m-1)은 서로 다른 입력 심볼에 의해 선택되어 동작된다. 신경망의 현 상태들, 과거 상태들, 입력 심볼값, 그리고 가중치들에 의하여 계산되어 결

정된다. 즉, $s_i^t = f(\sum_{j,k} w_{ij}^k s_j^{t-1} I_k)$ 이고 여기서 I_k 는 시간 t에 입력되는 k번째 심볼의 입력 값(해당 심볼이면 0, 아니면1)이고 w_{ij}^k 는 현재 시간 i번째 상태 s_i^k 와 바로 전 j번째 상태 s_j^{k-1} 를 연결하는 가중치이며 상태값을 얻기 위해 적용되는 함수 f는 다음과 같이 두 가지 값만을 갖도록 되어 있어 이산성(discreteness)의 성질을 갖는다. $f(x) = \begin{cases} 0.8 & \text{if } x > 0.5 \\ 0.2 & \text{otherwise.} \end{cases}$

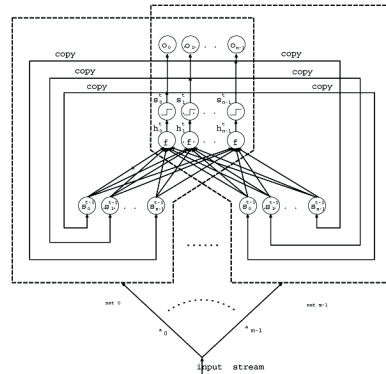


그림 1. 이산 재귀 신경망
Fig. 1. Discrete Recurrent Neural Network (DRNN)

초기 재귀 신경망은 초기 상태들로부터 시작하여 가중치들에 의해 다음 상태들이 결정되는 재귀적 연결구조를 갖추고 이것을 1차적 구조라 한다. 이 신경망은 초기 신경망에 입력 심볼들이 추가되고 이 심볼들에 따라 선택되는 같은 형식의 부신경망이 다층적으로 구성되기 때문에 이산 이차 재귀 신경망(Discrete Second-order Recurrent Neural Networks: DSRNN) 이라 부른다. 이 신경망은 정규 문법의 규칙을 추론하도록 제안되어 실험되고 외부 스택을 추가하여 문맥-자유 문법을 추론하려는 실험에 적용되었다[8]. 학습된 신경망은 내부 상태에 대한 최소화를 고려하지 않으므로 과잉된 상태가 존재하여 최적화된 신경망의 구성이 될 수 없는 결점을 가진다.

2. 이진 이차 재귀 신경망

이산 재귀 신경망의 매개변수들의 값들을 이진값으로 전환하여 구성된 신경망을 이진 신경망(binary neural network)이라 한다. 그림 2와 같이 이 신경망을 부신경망으로 하고 각 입력 심볼에 의해 이 부신경망이 선택되도록 하는 다층 부신경망의 구조가 이진 이차 재귀 신경망(BSRNN)이다[19]. 이것은 유한 오토마타와 동일하게 취급되며 이 신경

망으로 미니 어휘분석기를 구현하였다[19]. 이 신경망을 기반으로 각 상태 안에 스택 심볼들에 대한 스택 동작들을 내포하도록 하여 이진 삼차 재귀 신경망이 구성될 수 있다.

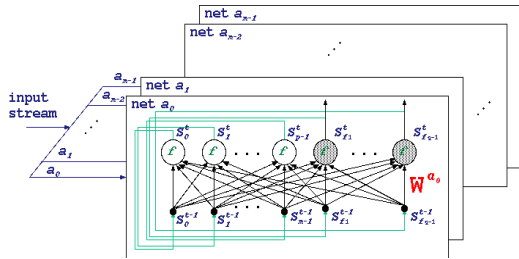


그림2 이진 이차 재귀 신경망
Fig. 2 Binary Second-order Recurrent Neural Network(BSRNN)

3. 문법 추론의 유전자 알고리즘

문맥-자유 문법의 추론에 유전자 알고리즘[20]은 문법의 생성규칙들을 염색체로 표현하여 개체들을 형성하고 모든 개체에 대하여 적합도를 평가하여 적합한 개체들을 선택-교배-돌연변이-대체의 과정을 종료조건을 만족할 때까지 새로운 세대를 생성하며 최고 적합도의 문법 염색체를 추론의 결과로서 제시한다. 적합도를 평가할 때 모든 입력 패턴들의 집합에 대하여PDA 시뮬레이터를 실행하게 되는데 문법 염색체 안의 생성규칙들을 탐색하게 되는데 이 과정에서backtracking을 하게 되므로 전체 수행시간의 지연을 초래할 수 있다는 단점이 있다.

III. 이진 삼차 재귀 신경망 (Binary Third-order Recurrent Neural Network: BTRNN)

이진 삼차 재귀 신경망(BTRNN)은 외부 스택과 결합하여 푸쉬다운 오토마타(PDA)와 같이 행동하도록 구성된다. 이진 이차 재귀 신경망(BSRNN), 스택 동작층, 그리고 연결선들로 이뤄지는 BTRNN의 정의는 정의 1과 같다.

<정의 1>

이진 삼차 재귀 신경망(BTRNN)은 9개의 항목으로 나타난다.

$$N = (\sum, Z, S, A, W, f, s_0, Z_0, s_{final}),$$

여기서

(1) $\sum = \{a_0, a_1, \dots, a_{m-1} = \$\}$ 은 입력심볼들의 유한 집합이며 $a_k \neq a_l (k \neq l)$ 이고 모든 k에 대하여 $a_k \neq \epsilon$ 이고 \$은 마지막 입력심볼을 나타낸다.

(2) $Z = \{b_0, b_1, \dots, b_{i-1} = Z_0\}$ 은 스택 심볼들의 유한 집합이고 Z_0 은 스택의 기초(bottom) 심볼이다.

(3) $S = \{s_0, s_1, \dots, s_{p-1}, s_{final}\}$ 은 p개의 미종료 상태들(non-final states) s_0, s_1, \dots, s_{p-1} 과 1개의 종료 상태(final state) s_{final} 을 가진 유한 집합이다. 이 상태들은 시각 t에서 현 상태들(current states)의 집합 S^t 과 다음 상태들(next states)의 집합 S^{t+1} 로 나타낼 수 있다.

ⓐ $S_{cur} = \{s_0^t, s_1^t, \dots, s_{p-1}^t, s_{final}^t\}$: 시각 t에서의 현재 상태들의 집합

ⓑ $S_{next} = \{s_0^{t+1}, s_1^{t+1}, \dots, s_{p-1}^{t+1}, s_{final}^{t+1}\}$: 시각 t에서의 다음 상태들의 집합

여기서 s_k^t 은 시각 t에서 k번째 상태를 나타내며 0 또는 1의 값을 나타낸다.

(4) 스택 동작의 순서열은

$$A = (A^0, A^1, \dots, A^{p-2}, A^{p-1}) = \left(\begin{matrix} (NOP_0, PS_{0b_0}, PP_{0b_0}, PS_{0b_1}, PP_{0b_1}, \dots, PS_{0b_{i-1}}, PP_{0b_{i-1}}), \\ (NOP_1, PS_{1b_0}, PP_{1b_0}, PS_{1b_1}, PP_{1b_1}, \dots, PS_{1b_{i-1}}, PP_{1b_{i-1}}), \\ \dots, \\ (NOP_{p-1}, PS_{p-1b_0}, PP_{p-1b_0}, PS_{p-1b_1}, PP_{p-1b_1}, \dots), \\ (PS_{p-1b_{i-1}}, PP_{p-1b_{i-1}}) \end{matrix} \right)$$

이다. $A^0, A^1, \dots, A^{p-2}, A^{p-1}$ 는 각각 $s_0^t, s_1^t, \dots, s_{p-1}^t$

에 해당되는 스택 동작들을 나타내는 순서열이고 상태

$$A^k = (A_0^k, A_1^k, A_2^k, \dots, A_{2l-1}^k) = (NOP_k, PS_{kb_0}, PP_{kb_0}, PS_{kb_1}, PP_{kb_1}, \dots, PS_{kb_{i-1}}, PP_{kb_{i-1}})$$

는 다음 상태 s_k^t 의 스택 동작들의 순서열을 나타낸다. NOP_k 는 No Operation (스택 동작 없음)을, PS_{kb_j} 는 push(b_j) (스택에 b_j 를 push)를, PP_{kb_j} 는 pop(b_j) (스택에서 반드시 b_j 를 pop)를 의미한다.

(5) 연결들(weights)의 유한 순서열은

$$W = (W^{a_0}, W^{a_1}, \dots, W^{a_{m-1}}) = \left(\begin{matrix} W_{00}^{a_0}, \dots, W_{0p-1}^{a_0}, W_{0final}^{a_0}, \dots, W_{p-10}^{a_0}, \dots, W_{p-1p-1}^{a_0}, W_{p-1final}^{a_0} \\ W_{00}^{a_1}, \dots, W_{0p-1}^{a_1}, W_{0final}^{a_1}, \dots, W_{p-10}^{a_1}, \dots, W_{p-1p-1}^{a_1}, W_{p-1final}^{a_1} \\ \vdots \\ W_{00}^{a_{m-1}}, \dots, W_{0p-1}^{a_{m-1}}, W_{0final}^{a_{m-1}}, \dots, W_{p-10}^{a_{m-1}}, \dots, W_{p-1p-1}^{a_{m-1}}, W_{p-1final}^{a_{m-1}} \end{matrix} \right)$$

이고 여기서

$$W_{ij}^x = \begin{cases} (w_{ijNOP_j}^x, w_{ijPS_{j_b}}^x, w_{ijPP_{j_b}}^x, \dots, w_{ijPS_{j_{b-1}}}^x, w_{ijPP_{j_{b-1}}}^x) & \text{if } x \neq \$ \\ (w_{ijNOP_j}^x, 0, 0, \dots, 0, 0) & \text{otherwise} \end{cases}$$

이다. 원소 w_{ijy}^x 는 입력 심볼이 x 일때 $s_i^t \in S_{cur}$ 과 $s_j^{t+1} \in S_{next}$ 사이의 연결로서 0 또는 1을 의미하고 스택의 동작으로 $ACT_j = \{NOP_j, PS_{j_b}, PP_{j_b}, \dots, PS_{j_{b-1}}, PP_{j_{b-1}}\}$ 에서 $y \in ACT_j$ 이다. 이것은 스택에 대하여 NOP 이거나 스택 심볼 b_k 을 $PUSH$ 하거나 POP 하는 것을 의미한다.

(6) f 는 뉴런 활성화 함수로 다음과 같은 OR 게이트이다.

$$f(x_0, x_1, \dots, x_{n-1}) = x_0 \wedge x_1 \wedge \dots \wedge x_{n-1}$$

(7) 각 입력심볼에 대응되는 재귀 부신경망들 $net\ a_0, net\ a_1, \dots, net\ a_{m-1}$, 이 있으며 이 신경망에 $x \in \Sigma$ 가 입력되면 부신경망 $net\ x$ 가 enable되고 이 $net\ x$ 에 포함되어 있는 공통적 성분들 S_{cur}, S_{next}, A, f 과 가중치 (weights) W^x 가 연결되어 동작한다.

(8) 시각 t 에서의 다음 상태들 $S_{next} = S^{t+1}$ 는 입력 심볼이 x 일 경우 현재 상태들 $S_{cur} = S^t$, 가중치 W^x , 함수 f 에 의해서 다음과 같이 결정된다.

$$\text{다음 상태: } s_j^{t+1} = f\left(\sum_i w_{ijy}^x s_i^t\right) \text{ for } y \in Z, x \in \Sigma,$$

스택 동작:

$$A_y^j = \begin{cases} y \in ACT_j & \text{if } w_{ijy}^x = 1 \text{ and } s_j^{t+1} = 1, \\ error & \text{otherwise.} \end{cases}$$

(9) 승인(acceptance) 또는 거부(rejection) :

입의 문자열 $\alpha = x_0 x_1 \dots x_{l-1}$ 의 식별 여부를 판단하기 위하여 이 신경망의 초기시각 $t=0$ 에서 초기상태 $(s_0^0, s_1^0, \dots, s_{p-1}^0, s_{final}^0) = (1, 0, \dots, 0, 0)$ 로 스택의 내용은 Z_0 를 초기값으로 시작하여 문자열의 심볼들을 하나씩 차례대로 입력하면서 (8)에서와 같이 다음 상태와 스택의 행동이 결정되어 문자열의 길이 l 만큼의 단계를 거치는 과정에서 이 BTRNN의 인식 결과를 얻는다. 마지막 입력 심볼 $\$$ 에서 상태가 $s_{final}^{t+1} = 1$ 이고 스택의 내용이 Z_0 이면 그 문자열을 승인하게 되고 마지막 입력심볼에서 상태가 $s_{final}^{t+1} = 0$ 이거나 스택 동작의 에러 또는 스택의 내용이 Z_0 가 아니면 거부된다.

이 정의에 따라 이진 삼차 재귀신경망의 구조는 아래 그림 3과 같다.

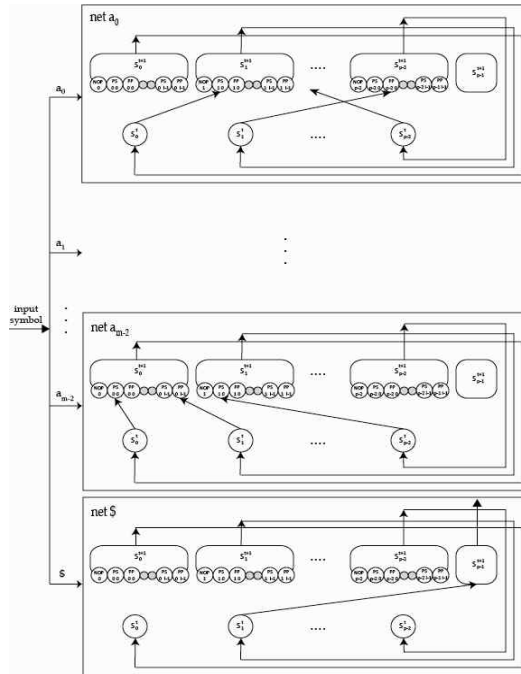


그림 3. 이진 삼차 재귀 신경망들
Fig. 3. Binary Third-order Recurrent Neural Networks (BTRNN)

입력심볼 $x \in \{a_0, a_1, \dots, a_{m-1}, \$\}$ 에 의해서 1차적으로 부신경망 $net\ a_0, net\ a_1, \dots, net\ a_{m-1}$ 중 하나가 선택되고 2차적으로 현 상태 S^t 에서 다음 상태 S^{t+1} 로 상태들이 전이되고 3차적으로 선택된 다음 상태 $s_j^{t+1} \in S^{t+1}$ 에서 스택에 대한 동작들

$$A^j = (NOP_j, PS_{j_b}, PP_{j_b}, PS_{j_{b_1}}, PP_{j_{b_1}}, \dots, PS_{j_{b_{l-1}}}, PP_{j_{b_{l-1}}})$$

중 $w_{ijb_k}^x = 1$ 로 연결된 하나의 동작이 실행된다. 이 과정에서 각 입력심볼에 의해서 동작하는 절차는 모두 3차례에 걸쳐 결정적인 선택(deterministic choice)을 하게 되므로 3차 신경망이라 일컫는다. 그림 4와 같이 BTRNN과 외부 stack이 결합되어 CFG 추론 시스템을 구성하게 되는데 BTRNN은 내부적으로 입력심볼 과 상태들을 결정하고 다음 상태와 같이 선택된 스택 동작 명령이 외부 스택에 연산을 지시하고 스택의 오류를 감지한다. 이 CFG 추론 시스템의 동작 흐름도를 나타내면 그림5와 같이 나타낼 수 있다.

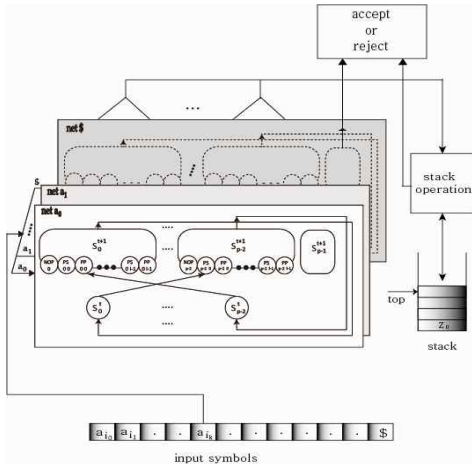


그림 4. BTRNN을 가진CFG추론 시스템 구성도
Fig. 4. Diagram of CFG Inference System with BTRNN

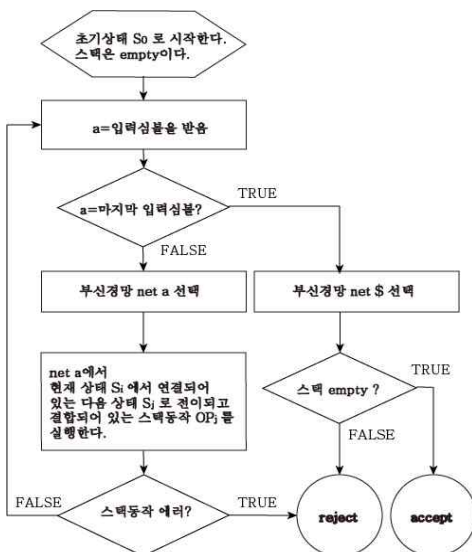


그림 5. BTRNN의 동작 흐름도
Fig. 5. Flowchart of BTRNN operations

IV. 문맥-자유 문법의 추론 (Inference of Context-Free Grammars)

이상과 같이 기술한BTRNN은 PDA의 기능을 수행할 수 있는 이진 값의(binary valued) 구성요소로 구성되어 있으므로 문맥-자유 언어(context-free grammars)의 긍정 또

는 부정의 예들을 학습하여 문법-자유 문법을 추론하도록 할 수 있다.

BTRNN의 학습 방법은 기존 신경망의 delta rule학습 알고리즘 대신에 유전자 알고리즘을 사용한다. 이 유전자 알고리즘을 적용하기 위하여 BTRNN의 개체 염색체화와 적자 생존의 우열을 가리기 위한 개체들의 적합도 함수(fitness function)를 결정하고 이를 이용한 BTRNN 염색체들의 유전자 알고리즘을 소개한다.

1. 염색체(Chromosomes)의 표현

BTRNN 염색체(chromosome)의 구성은 크게 입력심볼 별 부신경망(subnet)들의 정보로 구분되고 각 부신경망 $net a_i$ 은 현 상태 s_i^t 로부터 다음 상태 s_j^{t+1} 의 연결 유무와 스택 동작 A_j^i 을 나타내는 정보를 항목(entity) E_{ij} 로 표시하도록 구성한다. 염색체의 최대 개체수를 max_chroms, 입력 심볼들의 최대 종류수를 max_syms, 상태들의 최대 개수를 m_states라고 하면 각 염색체 표현 공간의 크기는 그림 6과 같이 max_syms×max_states개의 항목들이 있고 이 항목의 총 개수는 max_chroms × max_syms × m_states 개가 된다. 이러한 크기의 염색체를 그림6과 같이 2차원 배열로 표현할 수 있다. 각 항목 E_{ij} 의 내용은 (다음 상태, 스택 동작)으로 표시되고 이 항목은 입력심볼 i 가 선택되고 현 상태 j 일 때 전이될 다음 상태와 스택 동작의 정보는 다음과 같다.

$$E_{ij} = \begin{cases} (0,0) & \text{if 현상태 } i \text{ 와 다음상태 } j \text{ 사이 연결선 없음} \\ (k, NOP=1) & \text{if 다음상태 } k \text{ 로 전이, 스택동작이 없음} \\ (k, PUSH(b_k)=2) & \text{if 다음상태 } k \text{ 로 전이, 스택심볼 } b_k \text{ 를 push} \\ (k, POP(b_k)=3) & \text{if 다음상태 } k \text{ 로 전이, 스택심볼 } b_k \text{ 를 pop} \\ \vdots & \vdots \\ (k, PUSH(b_{-1})=2*t) & \text{if 다음상태 } k \text{ 로 전이, 스택심볼 } b_{-1} \text{ 를 push} \\ (k, POP(b_{-1})=2*t+1) & \text{if 다음상태 } k \text{ 로 전이, 스택심볼 } b_{-1} \text{ 를 pop} \end{cases}$$

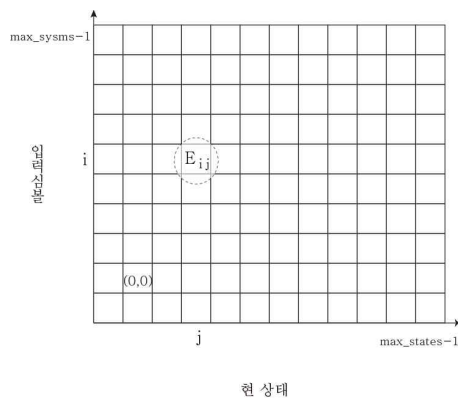


그림 6. BTRNN 염색체의 표현
Fig. 6. Representation of a BTRNN Chromosome

2. 적합도(Fitness) 함수

각BTRNN 개체에 입력 문자열들로 주어진 긍정과 부정의 예들(examples)을 인식하는 과정을 통하여 우수한 염색체 선발에 중요한 여러 요소들의 값들을 추출하여 각 개체의 적합도를 계산한다. 전체 BTRNN 염색체들 중 n 번째 염색체의 적합도 함수는 다음과 같이 계산된다.

$$fitness(n) = \alpha \times \text{긍정인식률} + \beta \times \text{부정인식률} + \gamma \times \text{상태수비율} + \delta \times \text{연결선수비율}$$

여기서 $\alpha + \beta + \gamma + \delta = 1$ 이고 $0 \leq fitness(n) \leq 1$ 이다.

긍정인식율은 전체 긍정 예들 중에서 인식 성공된 긍정 예들의 비율, 부정인식율은 전체 부정 예들 중에서 인식 실패한 부정 예들의 비율, 상태수비율은 사용 가능한 사용한 전체 상태개수 중에서 사용하고 있는 상태들의 개수가 c개수 중심의 최소 개수를 지향하는 비율로서 $(1 - \frac{|사용상태수 - d|}{\text{전체가능상태개수}})^2$, 연결선수비율은 전체 가능한 연결선의 개수 중에 사용하는 연결선의 개수가 d개 중심의 최소 개수를 지향하는 비율로서 $(1 - \frac{|사용연결선수 - d|}{\text{전체가능연결선수}})^2$ 를 사용한다. 여기서 c는 희망 최소 상태수이고 d는 희망 최소 연결선수 이다. 적합도의 4개의 계수들은 각 항의 중요성에 따라 가중치로 결정되되 $\alpha \geq \beta \geq \gamma \geq \delta$ 의미로 초기값을 설정하고 실험에서 경험적으로 조정하여 선택한다.

3. 유전자 알고리즘(Genetic Algorithm)

m개의 (m = 320개)의 BTRNN 염색체를 제1세대로 하여 개체 별 적합도 값을 계산하여 비교우위의 개체들 ($\frac{m}{2}$ 개)을 선정하여 다음 세대의 부모들로 결정하고 새로운 자식들은 부모 염색체들의 교배, 돌연변이를 통하여 생성한다. 이러한 진화과정은 모든 예들이 인식되고 최대 적합도 값이 인정 확률(p=0.95)이상일 경우에 승인(accept)되고 세대수가 제한된 회수(t=600)이상일 될 때까지 최대 적합도가 인정될 이하이거나 모든 예들이 인식되지 않으면 거절(reject)된다. 이러한 결과가 될 때까지 다음 세대를 생성하여 적합도 평가, 선택, 교배, 돌연변이 과정을 반복한다. 이를 수행하는 유전자 알고리즘은 그림 7와 같다.

```

BTRNN_Genetic_Algorithm()
{
    initializeChroms();
    epoch=0;
    evaluateChroms();
    while(!termination_condition(epoch)) {
        selectChroms ();
        crossoverChroms();
        mutateChroms();
        evaluateChroms();
        epoch++;
    }
    determine_accept_or_reject();
}
    
```

```

initializeChroms(){
    유전자 알고리즘에 필요한 전역변수들 초기화
    한다.
    염색체들의 유전자 값들을 초기화 한다.
}
evaluateChroms(){
    모든 염색체들의 적합도를 계산한다.
}
selectChroms(){
    모든 BTRNN 염색체에서 tournament 방식으로
    전체 BTRNN의 1/2를 선택한다.
}
    
```

```

crossoverChroms(){
    선택된 2개의 부모 염색체들부터 임의의 위치의
    개수만큼 임의의 위치들로 분할하여 분할한 구간
    을 서로 교배한 2개의 자식 염색체들을 생성한다.
}
mutateChroms(){
    임의의 염색체에 임의의 돌연변이 개수만큼을
    임의의 위치에 내용을 합당한 임의의 항목 값으
    로 변경하여 새로운 개체로 생성한다.
}

termination_condition(n){
    ① n이 허용된 세대수에 도달되면 true
    ② 최고 적합도가 허용치 이상일 때 true
    ③ 최고 적합도의 변화 없이 자연회수가 허
    용치 이상일 때 true,
    ④ 나머지 경우는 false
}
determine_accept_or_reject()
{
    ① 종료시 전역 변수의 정보에 따라 accept
    또는 reject
    ② 실행결과에 대한 정보 출력
}
    
```

그림 7. 유전자 알고리즘
Fig. 7. Genetic Algorithm

이 유전자 알고리즘의 동작은 모든 BTRNN 염색체들(m 개)의 내용을 허용 범위내의 임의의 값으로 초기화하고(initializeChroms) 세대수를 나타내는 변수 epoch를 0으로 초기화하고 모든 BTRNN 염색체들의 적합도를 계산한다(evaluateChroms). 다음의 3가지 기능들을 종료조건(termination_condition)이 될 때까지 반복 실행한다. 모든 BTRNN염색체들의 적합도 값들을 비교하여 순차적 토너먼트 방식으로 비교우위의 염색체들을 선택하여 m/2개의 다음 세대의 염색체들이 생존한다(selectChroms). 생존된 염색

체들 중 2개의 염색체들이 부모가 되어 그 염색체들의 정보표현으로 부터 임의의 지점들에서 임의의 회수만큼을 절단하고 교배하여 새로운 2개의 새로운 자식 염색체를 생성하고 나머지 m/2개의 개체수를 임의로 생성하여 보충한다(crossoverChroms). 임의의 염색체에 임의의 돌연변이 횟수만큼을 임의의 위치에 정보내용에 합당한 범위의 값으로 변경하여 개체의 정보표현을 수정한다(mutateChroms). termination_condition(n)은 현 세대 epoch에서 모든 예들에 대하여 인식이 성공하고 허용된 적합도 값에 해당하며 허용된 자연회수가 지나면 TRUE를 리턴하고 그 외의 경우 일 경우 FALSE를 리턴한다. 이와같이 선택, 교배, 돌연변이, 재생성의 반복적 실행이 끝나면 결과로 최적의 염색체를 가진 BTRNN에 대한 정보를 저장하거나 출력한다(determine_accept_or_reject).

V. 실험 및 결과

1. 추론 실험

문맥-자유 문법 추론에서 사용되는 시험적인 언어들을 취합하여 정리하면 다음의 7가지 언어들이 실험에 사용되었다[8][14-17][16].

- Balanced Parenthesis Problem (L1)
- $0^n 1^n$ Problem (L2)
- $0^n 1^{2n}$ Problem (L3)
- $(01)^n$ Problem (L4)
- 0의 개수는 짝수개이고 1의 개수는 홀수개인 문자열들 (L5)
- $a^m + b^m c^n$ Problem (L6)
- $a^n b^n c b^m a^m$ Problem (L7)

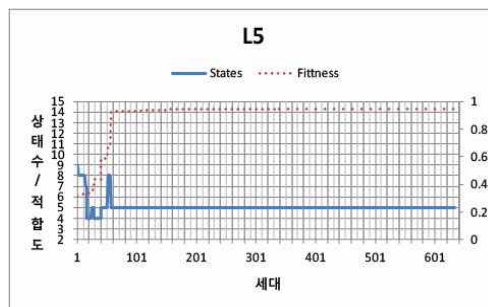
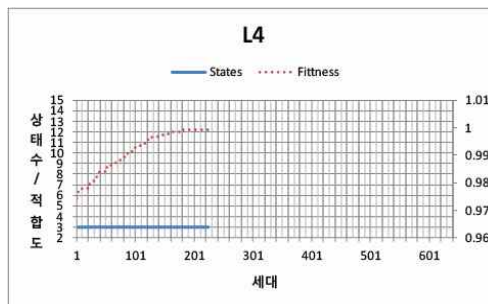
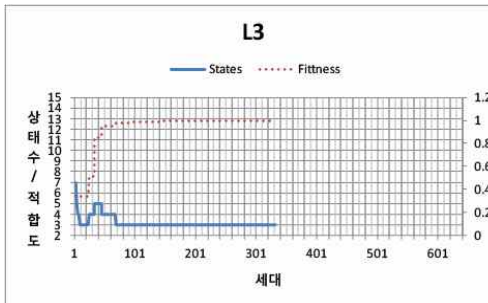
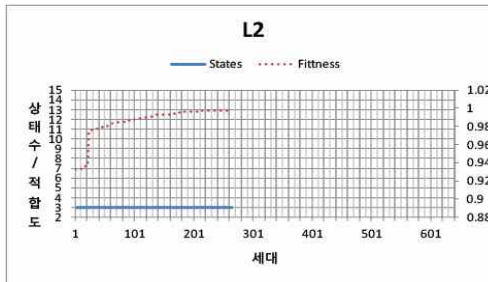
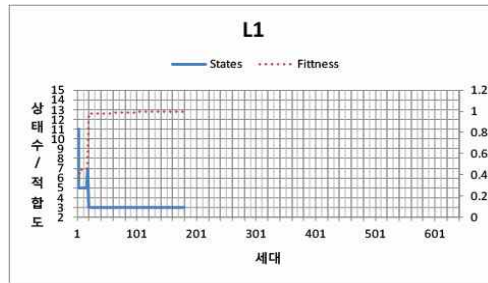
이 논문에서는 문맥-자유 문법의 추론의 실험과 검증을 위하여 위의 7가지 언어를 모두 사용한다. 실험에 사용된 매개변수 및 계수 값들은 40여 회의 실험을 통해 통계 및 경험적으로 결정된 것으로 모든BTRNN들이 적합도 함수(fitness)의 값이 0.95이상의 값을 얻을 수 있는 경우의 값이 선정되었고 유전자 알고리즘(GA)에 적용된 매개변수의 내용은 표1과 같다.

표1. 유전자 알고리즘의 매개변수
Table 1. GA Parameters

개체수	320	
입력 심볼의 수 (n)	2 또는 3	
신경망의 최대 상태수	16	
염색체의 유전자 수	32 또는 48	
교배률	0.0~1.0	
회당 최대 교배위치수	8	
돌연변이률	0.0~1.0	
회당 최대 돌연변이수	16	
긍적 예의 개수	10~59	
부정 예의 개수	22~52	
fitness함수의 계수	α	0.4
	β	0.3
	γ	0.2
	δ	0.1

실험에 사용된 총 개체(염색체)수는 320개를 사용했고 입력심볼은 마지막 심볼을 제외하고 L1~L5에서는 2 이고 L6~L7에서는 3개이다. BTRNN의 상태 수는 16개이므로 염색체내의 유전자 수는 입력 심볼의 개수에 따라 32 또는 48개가 된다. L1~L7의 문법 추론을 위하여 10~59개의 긍정의(positive) 예들과 22~52개의 부정의(negative) 예들을 사용했는데 인식률을 높이기 위해서 긍정 또는 부정의 예를 추가하여 실험하였고 대체로 긍정 보다는 부정의 예의 수가 더 많이 사용되었다. 두 염색체의 교배 시에 교배의 위치수가 (교배율x 회당 최대 교배 위치수) 0~8개 임의로 선정이 되고 두 교배 위치 사이의 유전자들의 교환이 일어난다. 교배로부터 생성된 자식 염색체들에게 돌연변이수 (돌연변이율x 회당최대 돌연변이수) 0~16개가 임의의 유전자 위치에 임의의 상태와 임의의 스택 동작으로 계산된 유전자 값으로 대체된다. 적합도 함수의 계수는 중요도에 따라 긍정예의 인식률(α) \geq 부정예의 인식률(β) \geq 상태수의 비율(γ) \geq 연결선수의 비율(δ)를 만족하는 $\alpha, \beta, \gamma, \delta$ 의 값으로 결정하여 실험하여 확인하였다.

L1~L7의 추론 실험의 결과를 살펴 보면 각 언어별로 성공적인 인식이 될 때까지 추론 세대수는 최대200번 정도의 차이가 발생한다. L1~L7에 대한 추론실험 중에서 평균적으로 세대변화에 따라 상태수와 적합도의 변화를 그래프로 나타내면 그림 8과 같다. L1, L2, L3, L4는 모두 3개의 상태수로 수렴하고 대체로 300세대(epoch)내외에서 최적의 BTRNN 학습 성공을 나타내고 있다. L5, L6, L7은 각각 상태수가 5, 4, 4 개에 수렴되고, 적합도는 400~700세대의 좀 더 긴 학습세대가 경과되었음을 볼 수 있으므로 추론의 난이도가 높다고 할 수 있다.



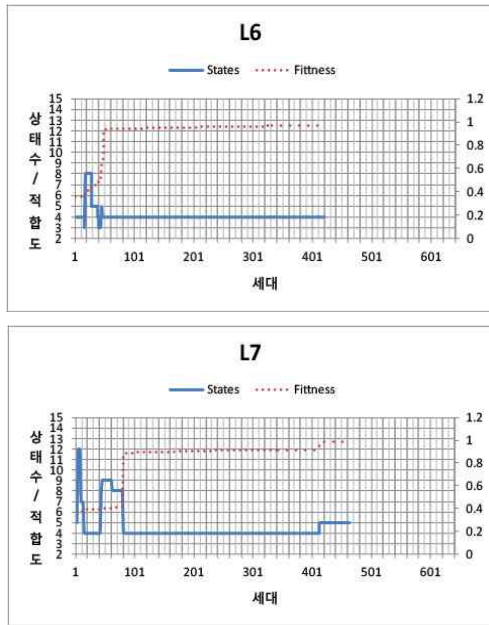


그림 8. 각 BTRNN의 상태수와 최적 적합도의 변화
Fig. 8. Variation of the number of states and the fitness value of each BTRNN

2. 결과

후론에 제시된 7가지 문맥-자유 언어들 L1~L7의 실험 결과로 생성된 염색체들의 내용을 BTRNN로 나타내면 그림 9와 같다. 이것은 3장에서 언급했던 BTRNN의 구조로 3~4개의 입력 심볼들과 이 심볼들에 대응되는 3~4개의 부신경망으로 구성된다. 각 부신경망에는 3~5개의 상태들이 포함되어 있고 마지막 상태를 제외한 각 출력 상태에는 5개의 스택 동작들, 즉 NOP, PUSH0, POP0, PUSH1, POP1중 한 개의 동작이 선택되도록 포함된다.

L1, L2, L3, L4 은 3개의 입력선과 3개의 상태들, L5는 3개의 입력선과 5개의 상태들로, L6는 4개의 입력선과 4개의 상태들, L7는 4개의 입력선과 5개의 상태들로 구성된다. 동작하는 순서는 입력 → 부신경망 선택 → 상태 및 스택동작 선택 → 다시 입력 → ... → 부신경망 선택 → ... → 마지막 상태 도달 의 과정을 밟는데 마지막 상태에서 스택이 empty이면 승인되고 마지막 상태에서 스택이 empty가 아니거나 마지막 상태에 도달하기 전에 상태의 전이가 없거나 스택의 동작에서 오류가 발생하면 거절된다.

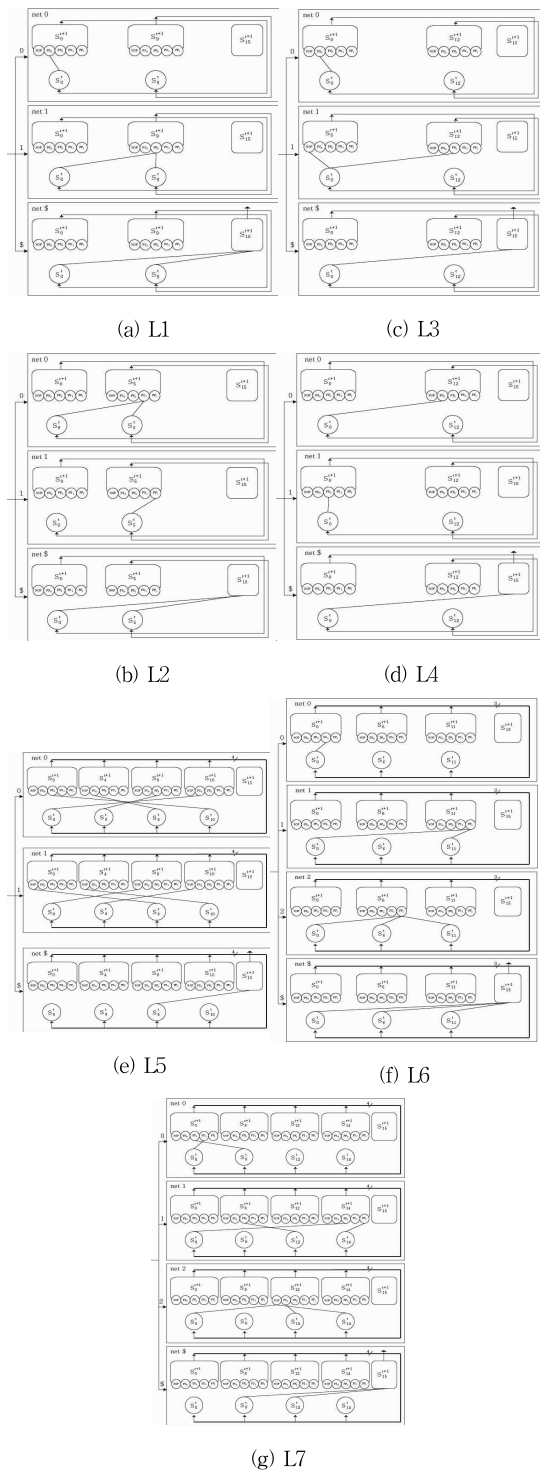


그림 9 BTRNN들
Fig. 9. BTRNNs

위의 BTRNN의 동작을 좀 더 이해하기 쉽도록 PDA전이도(transition diagram)으로 전환하여 표현하면 그림 10과 같이 대응되어 나타낼 수 있다. 여기서 PDA전이도는 FA 전이도의 입력심볼에 스택동작을 추가 결합하여 (입력심볼, 스택동작)을 PDA전이도의 입력 항으로 나타낸 것이다.

그림 10 (a)의 PDA상태전이 과정을 그림 9 (a)의 BTRNN과 비교하여 설명하면 BTRNN의 상태들 S0, S5, S15 은 PDA전이도의 상태들 S, A, F로 전환되고 BTRNN의 각 입력심볼에 따른 부신경망 내부에 각 상태에서 연결선으로 지정된 스택동작과 소속된 상태는 PDA상태전이도의 입력 항(입력심볼, 스택동작)에 따른 상태들의 전이로 나타난다. 즉 입력심볼 0 일때 S0에서 S5 로 전이되며 스택동작 PUSH1 을 하므로 PDA전이도에서 상태전이 S에서 A로전이 하며 입력항 (0, PUSH1)으로 표시한다.

처음 S상태에서 시작하여 입력심볼 0이 들어오면 스택심볼 1을 스택에 push하고 상태 A로 전이되고 마지막 입력심볼 \$가 들어오면 종료상태가 되며 곧 accept된다. 상태 A에서 입력심볼 0이면 스택심볼 1을 push하고 상태 A가 되며 입력심볼 1이면 스택에서 스택심볼 1을 pop해야 하고 상태 A가 된다. 이 때 지정된 스택심볼이 pop되지 않으면error가 되어 reject된다. 상태 A에서 마지막 입력심볼 \$이 들어오면 종료상태 F가 되고 곧 종료가 된다. 이 PDA 상태 전이도는 L1을 인식하는 최소의 상태 전이를 나타내고 나머지 L2~L7의 PDA 상태 전이 과정도 동일하게 설명할 수 있다. 추론된 BTRNN로부터 PDA 상태전이도를 표현할 수 있으며 서로가 동치임을 쉽게 증명할 수 있다.

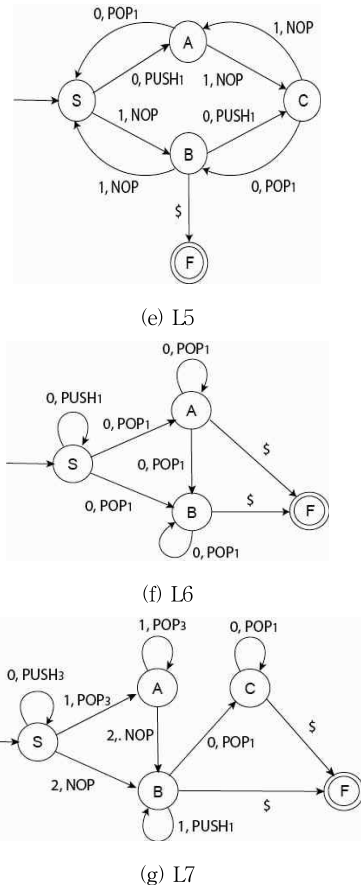
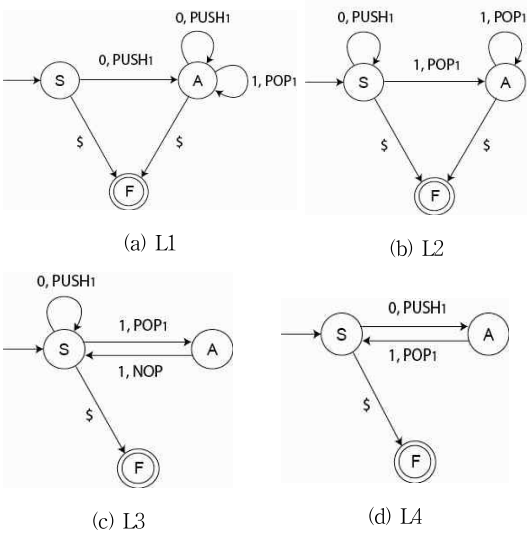


그림 10. L1-L7의 전환 표현된 PDA 상태전이도
Fig. 10. PDA state transition diagrams converted from BTRNNs



3. 성능 비교 분석

제안된 방법은 신경망의 구조와 유전자 알고리즘을 기반으로 동작하므로 이 두 가지 관점에서 기존의 방법들과 성능을 비교 평가한다. 먼저 신경망의 구조에 관련하여 DRNN을 이용한 방법과 비교 평가하고 다음으로 문법 검색체의 유전자 알고리즘의 방법과 비교 평가하면 다음과 같다.

3.1 DRNN의 방법과 비교

DRNN을 이용한 추론 방법[8]의 문헌에서 제공한 자료와 제안된 방법의 실험 결과의 자료를 (a)학습에 사용된 입력 패턴의 수, (b)추론(학습)의 경과된 세대수, (c)추론된 신경망 내부 상태수의 관점에서 그 내용을 비교하면 그림 11의 3가지 그래프로 나타내고 그 결과의 분석내용은 다음과 같다.

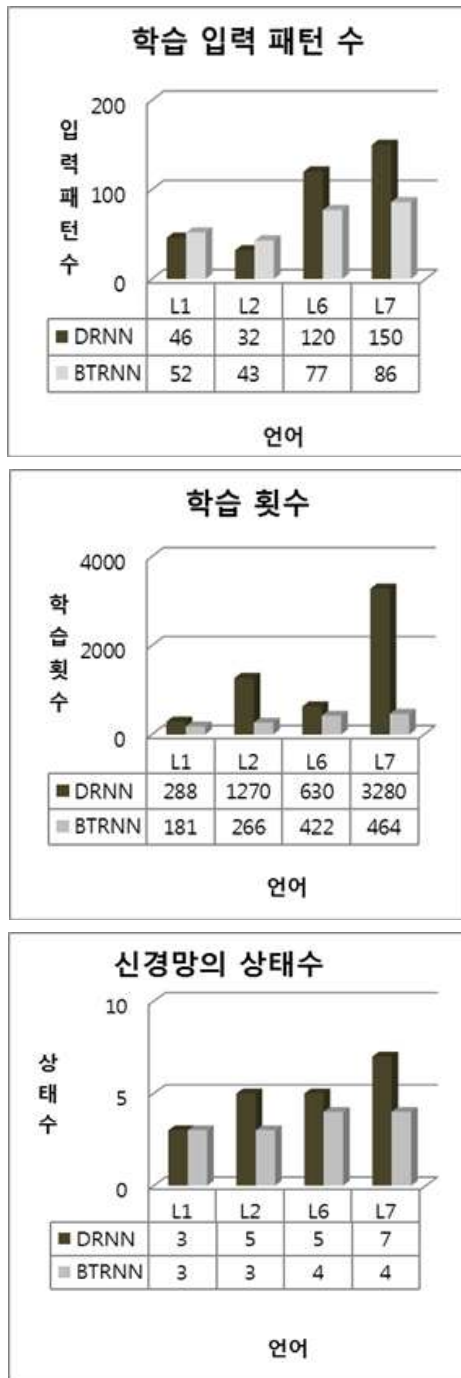


그림 11. DRNN의 추론방법과 비교
Fig.11. Comparison with the method of DRNN

(a) 추론에 사용된 입력 패턴 수
BTRNN은 모든 문법에 대하여 사용하는 입력 패턴의 수

는 52~86이고 DRNN의 경우는 46~150로 상대적으로 변동이 크다. 난이도가 높은 두 언어형태L6와 L7에 대해서 DRNN이 BTRNN보다 추론을 위해서 1.5배정도의 입력패턴 수가 필요하다. 난이도가 높은 언어일수록 BTRNN의 방법이 DRNN의 방법보다 상대적으로 적은 수의 입력 패턴들로부터 추론할 가능성이 높다고 볼 수 있다.

(b) 학습 회수

BTRNN의 추론과정에서 입력 패턴의 집합을 학습하는 횟수가 DRNN의 학습 횟수보다 모든 문법 종류에 대하여 작고 L7일 경우는 8배 정도 적게 걸리는 것을 확인할 수 있다. BTRNN이 DRNN보다 적은 학습횟수로 추론될 수 있음을 보여준다.

(c) 신경망의 상태 수

BTRNN의 상태 수는 3~4개이고 DRNN은 3~7개이다. BTRNN의 상태수가 모든 문법 종류에서 같거나 작다. 난이도가 높은 문법일수록 DRNN의 상태수가 커지는 결과를 보여주며 BTRNN은 최적화 결과로 최소의 상태수를 보여준다.

이와 같이 BTRNN의 방법이 DRNN의 방법보다 추론과정에서 적은 학습량과 적은 학습횟수로 학습이 가능함을 보여주며 모든 입력패턴에 대하여 같거나 적은 내부 상태 수를 갖고 더 간결하고 신속한 학습(추론)을 할 수 있음을 보여준다. 또한 제안된 방법은 제시된 모든 문법들의 긍정과 부정의 예들을 완전히 학습하여 최적의 BTRNN을 구성할 수 있고 시험의 예들을 완전하게 인식 가능하다. 이러한 BTRNN는 DSRNN와 구조적으로 이진화와 이산화의 차이점 이외에 연결성이 같다 볼 수 있으므로 DRNN보다 상태 수가 같거나 작은 BTRNN의 인식 처리 속도는 DRNN보다 나쁘지 않다.

3.2 문법 검색체의 추론방법과 비교

문법 검색체의 방법[13]과 제안한 방법은 공통적으로 유전자 알고리즘을 채택하고 있으나 각각 사용하는 검색체 표현의 차이로 인해 성능의 차이가 발생한다. 이 차이를 학습 과정과 시험 과정에서 평가해 보면 다음과 같다.

1) 학습(추론)과정

유전자의 연산들로 선택, 교배, 돌연변이, 재생성이 있는데 이 과정에서 교배, 돌연변이, 재생성 단계에서는 두 방법이 모두 검색체를 대상으로 기계적인 단순처리를 수행하므로 성능의 차이가 없으나 선택단계에서는 우수한 검색체를 선택하기 위해서 적합도를 계산하기 위한 과정에서 두 방법이 서로 다른 동작을 수행한다. 문법 검색체의 방법에서는 검색체안의 유전자인 생성규칙들을 이용하여 입력 문자열을 parsing을 하는데 이 과정에서 backtracking의 문제가 발생되고 이 것이 이 방법의 수행속도를 지연시키는 결과를 초

래하는 반면에 제안한 방법은 그림 5의 BTRNN의 동작 흐름 도에서 본 바와 같이 검색체 BTRNN를 결정적(deterministic)으로 전이하면서 적합도를 계산한다.

2) 시험(인식)과정

인식하고자 하는 문자열에 대한 승인과 거절의 판단을 하기 위해 추론된 검색체에 입력 문자열로서 적용한다. 문법 검색체의 방법은 위의 추론과정에서와 마찬가지로 parsing을 수행해야 한다. 반면에 제안된 방법은 입력 문자열의 길이 만큼 차례대로 BTRNN의 결정적인 절차를 밟으며 상태 전이와 스택 동작을 하나씩 수행하며 진행한다.

이 두 과정에서 두 방법의 이론적 차이점은 입력 문자열을 인식할 때 결정적인 처리와 비결정적인 처리가 된다. 이 두 방법에 대하여 언어 L1, L2, L4, L5를 사용하여 (a)추론 과정의 경과 시간, (b)경과 세대, (c)세대당 실행시간의 관점에서 제공된 자료와 실험 자료의 결과를 비교하여 나타내면 그림 12과 같다.

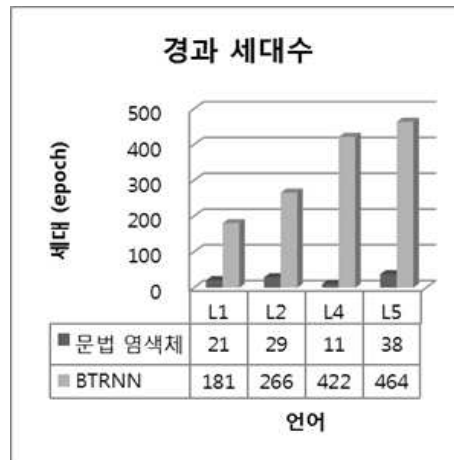
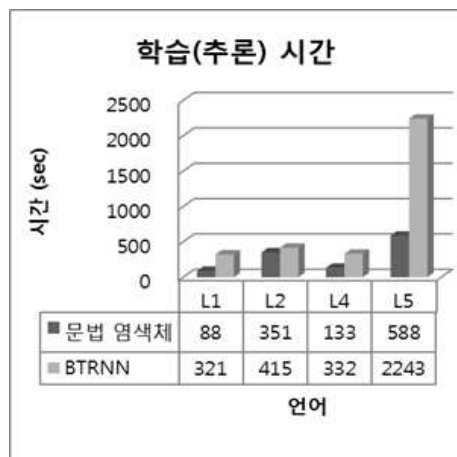
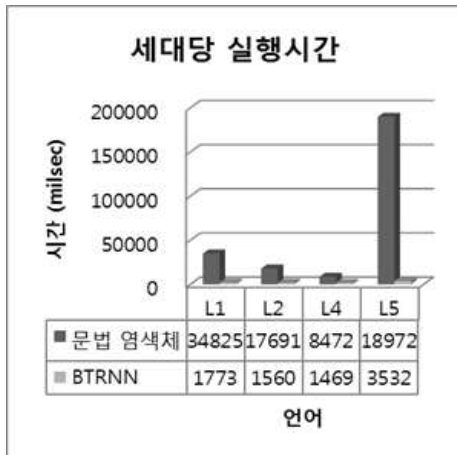


그림 12. 문법 검색체의 추론방법과의 비교
Fig. 12. Comparison with the method of Grammar Chromosome



(a) 학습(추론)시간

각 언어를 추론하는데 경과한 시간은 BTRNN이 모든 언어들에 대하여 추론 시간이 1.2~2 배 정도 더 경과되었다.

(b) 경과 세대수

BTRNN이 181~464의 세대수에 걸쳐 추론이 진행된 반면에 문법 검색체의 경우는 21~38의 세대수 동안에 학습이 진행되었다. BTRNN의 추론을 위한 세대수는 문법 검색체의 세대수 보다 9~40배 정도가 많은 세대가 필요하다.

(c) 세대당 실행시간 BTRNN의 세대당 실행시간이 2~6 배 적게 걸린다. 이것은 BTRNN이 입력 패킷을 받아 검색체를 해석하여 승인과 거절로 응답하는 시간이 문법표현 유전자 의 방법보다도 우수하다는 것을 의미한다.

두 방법은 추론(학습)과정에서 BTRNN이 비교적 더 많은 추론 시간이 걸렸고 입력 패킷들을 추론하는데 필요한 세대수도 또한 많다. 그러나 BTRNN이 세대당 소요된 실행시간이 3~5배 적게 걸리므로 입력 문자열에 해석에 대한 유효한 처리 속도를 보여준다. 이것은 BTRNN이 스택을 사용함에도 스택의 특성인 backtracking의 결점이 반영되지 않는 것은 스택을 단순한 저장확인 용도로만 이용하는 점과 이진 재귀 신경망의 특성으로 결정적 선택처리가 수행된다는 점 때문이다. 추론이 완료된 후의 BTRNN의 사용은 문법 검색체의 방법보다 인식기로서 우수한 속도를 보여준다. 이 결과에 대한 알고리즘의 시간 복잡도 분석을 나타내 보면 다음과 같다.

BTRNN의 인식과정의 시간복잡도는 문맥-자유 문법의 비단말 심볼의 개수 p, 단말 심볼의 개수 q, 스택 심볼의 개수 s, 그리고 길이가 k인 문자열이 입력이 된다고 할 때,

BTRNN상태의 최대 개수가 m 이라고 가정하면 병렬처리 시간은 $O(k)$ 이고 순차처리 시간은 $O(km)$ 이다. 반면에 문법 염색체 표현 방법에서는 비단말 노드를 left-hand side(lhs)에 가진 생성규칙이 단지 한 개씩만 존재하고 1차적 유도 적용으로 끝난다고 가정하면 파싱시 입력 문자열의 각 문자를 생성하기 위한 전체 생성규칙의 개수는 p 개가 되고 길이 k 인 문자열의 인식처리 시간은 $O(pk)$ 가 된다. 따라서 BTRNN의 인식처리 시간복잡도는 문법 표현의 방법보다 우수하다고 할 수 있다.

VI. 결론

이 논문은 BTRNN에 대하여 소개하고 문맥-자유 언어의 긍정과 부정의 예들을 BTRNN에 유전자 알고리즘으로 진화시켜 문법을 추론하는 방법론을 제안한다. 이 제안된 방법은 기존의 신경망의 접근방법보다 적은 학습량과 적은 학습시간으로 최적의 해를 학습할 수 있으며 최신의 문법표현 염색체들의 방법들보다는 입력패턴들에 대한 추론속도가 우수함을 보여준다.

BTRNN는 요소들이 이산적(discrete)이고 결정적(deterministic) 선택 구조를 갖추고 있으며 저장확인의 기능만을 가진 스택을 결합하여 PDA 상태 전이도와 동등한 성질을 가지게 된다. 이러한 특징으로 기존의 많은 CFG 추론 방법들에서 고려하는 파싱 시뮬레이션의 backtracking없이 해석이 가능하게 하며 전형적인 신경망을 이용한 추론 방법들에서 사용하는 아날로그식 요소들 때문에 발생하는 애매한 추론 없이 결정적인 신속한 판단을 제공한다. 또한 학습된 BTRNN의 구조는 문법-자유 언어의 인식 모델을 H/W 또는 S/W로 간편하게 만들 수 있게 한다는 장점이 있다.

향후 이 신경망의 추론 적용 연구는 규모가 더 크고 더 실용적인 문맥-자유 언어에 대하여 일반적이고 효율적인 접근 방법으로 확장되어야 하고 더 나아가 새로운 신경망 구조 및 추론 방법에 대한 연구에 도움을 줄 수 있을 것이다.

참고문헌

[1] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol.14, pp.179-211, Apr./Jun. 1990.

- [2] S. Porat and J. A. Feldman, "Learning automata from ordered examples," *Machine Learning*, vol.7, no. 2-3, pp.109-138, Sep. 1991.
- [3] K. Fu and T. L. Booth, "Grammatical inference: Introduction and survey," *IEEE Trans. Pattern Anal. & Machine Intell.*, vol.8, no.3, pp.343-375, May 1986.
- [4] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural Comp.*, vol.1, no.3, pp.327-381, Mar. 1989.
- [5] 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, pp.393-405, Mar. 1972.
- [6] 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, pp.54-78, 1993.
- [7] 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, pp.393-405, March 1992.
- [8] Z. Zeng, R. M. Goodman, and P. Smyth, "Discrete recurrent neural networks for grammatical inference," *IEEE Trans. Neural Networks*, vol.5, no.2, pp.320-330, June 1994.
- [9] 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, pp.511-513, May 1994.
- [10] 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, pp.1996-2007, Nov. 2000.
- [11] I. Tsoulos, D. Gavrilis, and E. Glavas, "Neural network construction and training using grammatical evolution," *Neurocomputing*, pp.1-9, Dec. 2008.

- [12] P. Wyard, "Context Free Grammar Induction Using Genetic Algorithm," 1994, Proceedings, Sixth Int'l Conference on Tools with Artificial Intelligence, pp.828-831, April 1994.
- [13] N. S. Choubey, and M. V. Kharat, "PDA Simulator for CFG Induction Using Genetic Algorithm," 12th Int'l Conference on Computer Modeling and Simulation, pp.92-97, March 2010.
- [14] H. M. Pandey, "Context Free Grammar Induction Library Using Genetic Algorithms," Int'l Conference on Computer & Communication Technology (ICCCCT), pp.752-758, Sept. 2010.
- [15] S. Lawrence, L. Giles, and S. Fong, "Natural Language Grammatical Inference with RNN," IEEE Trans. Knowledge and Data Engineering, vo.12, No.1, JAN/FEB, 2000.
- [16] E. Rodrigues and H. S. Lopes, "Genetic Programming for Induciton of Context-free Grammars," 7th International Conference on Intelligent Systems Design and Applications, pp.297-302, Oct. 2007.
- [17] J. E. Hopcroft and J. D. Ullman, "Introduction Automata Theory, Languages, and Computation," Addison-Wesley, pp.217-232, 1979.
- [18] U. Harigopal, and H. C. Chen, "Grammatical Inference using Higher Order Recurrent Neural Networks," Proceedings SSST '93., 25th Southeastern Symposium on System Theory, pp.338-342, March 1993.
- [19] S. Jung, "Generalized Binary Second-order Recurrent Neural Networks Equivalent to Regular Grammars," Journal Of Korea Intelligent Information System Society(KIIS), vol. 12, no. 1, pp.107-123, MAR, 2006.
- [20] N. S. Chouney and M. U. Kharat, "Stochastic Mutation Approach for Grammar Induction Using Genetic Algorithm," [International Conference on Electronic Computer Technology \(ICECT\)](#), pp.143-146, May 2010.

저자 소개



정 순 호

1980: 서울대학교 수학교육 학사

1982: 한국과학기술원 전산학과 석사

2000: 한국과학기술원 전산학과 박사

현 재: 부경대학교 컴퓨터공학과 교수

관심분야: 임베디드 지능시스템, 컴퓨

터 보안, 가상현실

Email: snow@pknu.ac.kr.