

# L-시스템을 이용한 모듈형 신경망의 구조진화

## Evolution of Modular Neural Networks by L-System

이승익<sup>o</sup>    조성배  
연세대학교 컴퓨터과학과

Seung-Ik Lee    Sung-Bae Cho  
Dept. of Computer Science, Yonsei University

### 요약

신경망은 입력력 관계가 명시적으로 표현되기 어려운 경우에 수집된 데이터를 이용하여 원래의 함수를 근사할 수 있는 특성이 있다. 최근에는 신경망의 모델링 성능을 향상시키기 위하여 여러개의 모듈을 기반으로 신경망을 구성하는 모듈형 신경망이 활발히 연구되고 있다. 본 논문에서는 린덴마이어 시스템(L-시스템)의 문법적 적용을 통하여 이러한 모듈형 신경망의 구조를 결정하는 방법을 제시하고자 한다. L-시스템은 본래 식물의 성장과정을 기술하기 위하여 제안된 방법인데, 본 논문에서는 신경망의 모듈형 구조가 L-시스템의 문법을 통하여 적절히 결정됨을 보인다.

### 1. 서론

최적의 신경망을 진화방식으로 구축하고자 하는 시도는 환경에 따라서 구조와 학습규칙을 변경할 수 있어야 하는 적응시스템을 개발하는데 커다란 가능성을 보여주고 있다. 이미 진화적 접근방식으로 신경망을 설계하는 방법에 대해서 수많은 연구결과가 발표되고 있다 [1, 2, 3]. 진화방식의 장점중 하나는 동적인 환경에의 적응성에 있으며, 그러한 적응과정은 연결강도와 구조 및 학습규칙의 진화를 통해서 이루어진다 [2].

신경망의 최적구조를 설계하는 것은 각 점이 구조를 표현하는 공간상에서의 탐색문제로 정형화할 수 있으며, 가능한 모든 구조의 성능수준이 이 공간상에서 일종의 함수를 형성한다. 그러한 함수는 실제 전통적인 방법에 비하여 진화적 알고리즘에 의해 보다 효과적으로 해결될 수 있는 여러가지 특성을 갖고 있다 [4].

하지만 이제까지 대부분의 진화형 신경망은 구조적인 제한이 있는 경우가 많다. 어떤 경우에는 각 노드들 사이의 완전연결을 가정하기도 하고, 또 어떤 경우에는 제층적인 다층의 구조를 가정하여 한 층의 모든 노드들이 이웃하는 층의 모든 노드들과 연결되도록 하기도 한다. 그러나 신경생리학의 많은 연구로부터 인간의 정보처리 시스템은 일종의 모듈로 구성되어 있으며, 각 모듈은 고유한 목적이나 기능을 갖고 있음이 보고되어 있다 [5].

모듈로 이루어진 신경망 시스템을 설계하는 경우에 주어진 구조를 기능과 연결시키는 문제를 해결하여야 하는데 아직까지는 완전한 해답이 존재하지 않는다. 반면에 생물의 두뇌구조는 오랜시간의 진화과정을 거쳐 형성되어 생존과 번식에 필요한 기능을 수행하면서 특수화된 부시스템의 집합으로 생성되었음이 주지되고 있으며, 공학적으로도 대규모의 문제를 단순한 처리기로부터 학습시키는 것은 엄정

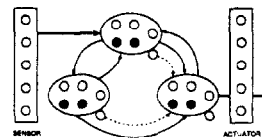


그림 1: 모듈형 신경망의 전체적인 구조에. 여기에서는 세 개의 모듈이 상호연결되어 센서로부터의 입력을 처리하고 있다.

난 시간을 소요한다는 사실을 알 수 있다.

본 논문에서는 Murre [6]가 제안한 모듈을 기본 구성요소로 하고, 문법을 사용한 진화방식으로 모듈형 신경망의 구조를 결정할 수 있도록 하기 위한 예비과정으로서 기본 문법을 정의하고 이를 이용하여 적절한 네트워크의 구조가 결정될 수 있음을 보인다. 이 논문에서 정의된 문법규칙을 진화방식에 적용하면 주어진 문제에 적합한 네트워크의 구조를 찾을 수 있을 것이라 기대된다.

### 2. 모듈형 신경망

제안하는 시스템의 전체적인 구성은 그림 1과 같다. 각 노드의 활성화 값은 식 1과 같은 식으로 계산된다:

$$e_i = \sum_j w_{ij} a_j(t), \quad (1)$$

단,  $w_{ij}$  는  $j$  노드에서  $i$  노드로의 연결강도를 나타낸다.  $i$  노드로의 실제 입력값  $e_i$ 는 이 노드의 입력부에 연결된 모든 노드의 개별적인 활성화 값을 가중치에 따라 합한 것이

다. 입력값은 양의 값(홍분)이나 음의 값(억제)을 가질 수 있다. 이러한 노드들로 구성된 모듈은 기본적으로 두뇌의 신피질내에 존재하는 미니컬럼을 모형화한 Murre [6]의 모델을 따르는데, 이 모델에 내재된 제약조건은 다음과 같다.

1. 데일의 법칙(각 뉴런은 양이나 음중 하나의 값만을 출력한다).
2. 국소적인 학습을 허용하여 올바른 응답에 대한 지식을 요구하지 않는다.
3. 새로운 입력과 학습된 입력을 구별하고 그에 따라 다르게 행동한다.

Murre는 이러한 원칙에 입각하여 입력된 활성화 패턴을 자동적으로 분류할 수 있는 모듈을 제안하고 그 동적 작동원리의 타당성을 규명하였다 [6]. 보다 자세한 사항은 참고문헌 [6]을 참조하기 바란다.

### 3. L-시스템

L-시스템은 1968년 Aristid Lindenmayer가 제안한 것으로서 다세포 생물의 성장과정을 모방하고 시뮬레이션하기 위한 형식이다[7]. 특히 L-시스템은 parallel string rewriting 속성을 지니는 일종의 문법이기 때문에 이후의 연구들은 주로 식물의 성장과정을 모델링하기 위한 도구로서 많이 적용하여 왔다.

#### 3.1 간단한 L-시스템

간단한 L-시스템은 다음과 같은 세가지 구성요소로 이루어져 있다.

1. **Alphabet** 심볼들의 유한 집합  $V$ . 주로  $a, b, c$  같은 문자들이 쓰이지만 다른 문자도 무방하다.  
예)  $V = \{a, b, c\}$
2. **Axiom** 초기인자(Initiative)라고도 불린다. 집합  $V$ 에서 정의된 심볼들의 연속된 문자열의 집합을  $V^*$ 라고 한다면 axiom은  $V^*$  집합의 한 원소이다.  
예)  $V = \{a, b, c\}$  라고 가정하면  $abc, aaa, ab$  등은 모두  $V^*$ 의 원소이고 axiom이 될 수 있다.
3. **Production** 하나의 심볼  $a \in V$ 를 하나의 스트링  $\omega \in V^*$ 에 매핑시키는 것.  
예) production  $p: a \rightarrow \omega$

이밖에도 문자열의 길이  $|\omega|$ 는 문자열에 들어있는 심볼의 개수라고 할 수 있으며, 특히 하나의 심볼을 길이가 0인 문자열에 매핑할 수 있으며 하나의 심볼에서 자기 자신만을 가지고 있는 문자열로의 매핑( $a \rightarrow a$ )도 가능하다. 특정 심볼에 대하여 어떤 production도 주어지지 않으면 자기 자신으로의 매핑을 가정한다.

#### 3.2 Bracketed L-시스템

간단한 L-시스템만 가지고 식물의 성장과정을 모델링하기에는 표현력에 한계가 있기 때문에 이를 해결하기 위하여 연구자들은 bracketed L-시스템을 고안하였다[7]. 이 방

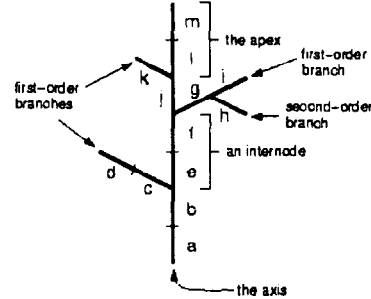


그림 2: 나무의 가지 구조[7].

법의 핵심은 '[' , ']' 기호를 사용하여 식물의 가지를 묘사하는 것이다. 그림 2와 같은 나무를 bracketed L-시스템을 사용하여 표현하면 식 2와 같다.

$$\omega = ab[cd]ef[g[h]i]j[k]lm \quad (2)$$

그림 2의 각 가지는 '[' , ']'로 묶여 있다. 그러므로 식 2에서 가지는  $[cd]$ ,  $[g[h]i]$  그리고  $[k]$  이렇게 세 종류이다.

#### 3.3 문맥에 의존적 파라미터 L-시스템

기본 L-시스템을 연장한 파라미터 L-시스템은 다음과 같이 정의된다.

##### Parametric L - System

- ▶ ordered quadruple  $G = \langle V, \sum, \omega, P \rangle$
- ▶  $V$  is the alphabet of the system
- ▶  $\sum$  is the set of formal parameters
- ▶  $\omega \in (V \times R^*)^+$  is a nonempty parametric word called axiom
- ▶  $P \subset (V \times \sum^*) \times C(\sum) \times (V \times E(\sum))^*$  is a finite set of productions
- ▶  $C(\sum)$  is logical expression with parameters from  $\sum$
- ▶  $E(\sum)$  is arithmetic expression with parameters from  $\sum$

여기서 production이 식 4와 같은 형태를 지니고 있으면 문맥의존적 파라미터 L-시스템이라고 불린다.

$$lc < pred > rc : cond \rightarrow succ \quad (4)$$

'<', '>'는 production의 세가지 구성요소를 나누는 기호인데, '<' 왼쪽의  $lc$ 를 왼쪽 문맥이라고 하고 '>' 오른쪽의

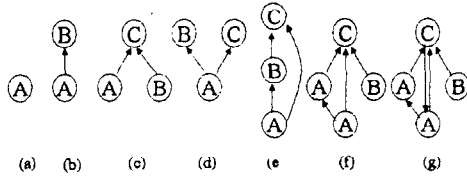


그림 3: 예제 네트워크.

rc를 오른쪽 문맥이라고 한다. cond는 이 규칙이 적용될 조건을 표시한다. 예를 들어서 식 5와 같은 production 규칙은 식 6과 같은 경우에 적용가능하다[7].

$$A(x) < B(y) > C[D(z)]F : x + y + z > 10 \quad (5)$$

$$\rightarrow U((x+y)/2)[V((y+z)/2)]$$

$$\dots A(4)B(5)C[D(6)E]F \dots \quad (6)$$

L-시스템의 규칙을 적용하기 위하여 문맥을 고려할 때 한 가지 주의하여야 할 사항은 스트림에서 '[', ']'은 production의 문맥으로 고려되지 않는다는 점이다.

#### 4. 문법을 이용한 네트워크의 표현

##### 4.1 네트워크 표현을 위한 문법 정의

문법을 이용하여 모듈형 신경망을 표현하기 위하여 우리는 문제에 적합하도록 약간 변형된 문맥 의존적 파라미터 L-시스템을 사용하였다. 그림 3을 참고로 우리가 정의한 L-시스템을 설명한다.

- ① 기본 문법은 문맥 의존적 파라미터 L-시스템을 사용한다.
- ② L-시스템에서 하나의 심볼은 하나의 네트워크 모듈을 지칭한다. 그림 3에서 (a)는 A로 표현된다.
- ③ Bracket으로 둘러싸인 스트림은 “기능그룹(Functional Group)”이라 부른다. 그림 3에서 (a)는 하나의 모듈을 가진 기능그룹 [A]로 표현가능하고 (b)는 두개의 모듈을 가진 기능그룹 [AB]로 표현 가능하다.
- ④ 두 모듈간 또는 기능그룹간 또는 모듈과 기능그룹간의 연결성
  1. 이웃한 두 모듈(심볼)간 또는 두 기능그룹간 또는 모듈과 기능그룹간은 연결된다. 그림 3에서 (a)는 A 또는 [A]로 나타낼 수 있고 (b)는 AB 또는 A[B] 또는 [A]B로 나타낼 수 있다.
  2. 이웃한 두 모듈간 또는 두 기능그룹간 또는 모듈과 기능그룹간의 비연결성은 ','로 표시한다. 그림 3의 (b)는 AB로 표현되고 A 모듈과 B 모듈은 연결되어 있다. 그러나 만약 이웃한 두 심볼 A와 B를 서로

연결하고 싶지 않다면 A, B로 표현할 수 있다[8].

3. 기능그룹에서 모듈로 연결한 경우 기능그룹안의 끝 모듈에서 연결할 모듈로 연결한다. 단, 기능그룹내에 비 연결성이 존재할 경우 서로 연결되지 않은 기능그룹내의 각 구성요소에서 각각 연결을 하되 각 구성요소의 끝 모듈에서 연결한다. 그러므로 그림 3의 (c)는 [A, B]C로 표현될 수 있다.

4. 모듈에서 기능그룹으로의 연결은 기능그룹을 시작하는 모듈로 연결된다. 단, 기능그룹내에 비 연결성이 존재할 경우 서로 연결되지 않은 기능그룹내의 각 구성 요소에 연결을 하되 각 구성요소를 시작하는 모듈로 연결한다. 그러므로 그림 3의 (d)는 A[B, C]로 표현될 수 있다.

- ⑤ 스트림의 각 알파벳은 두 개의 파라미터를 갖는다.
- ⑥ 심볼의 첫 번째 파라미터는 그 심볼이 나타내는 모듈의 크기들 의미한다. 예를 들어서 A(5) 하면 노드의 수가 5인 하나의 모듈을 지칭한다.
- ⑦ 심볼의 두 번째 파라미터 n(정수)은 “지나침거리”를 의미한다[8]. n이 0이거나 양수일 경우 스트림에서 현재 심볼(또는 기능그룹)의 오른쪽으로 n번 이동한 후 그 다음번 심볼과도 연결한다. 음수일 경우 현재 심볼의 왼쪽으로 n번 이동한 심볼과도 연결한다. 그러므로 그림 3의 (e)는 A(#, 1)BC로 표현 가능하며 (f)는 A(#, 2)A(#, 1), BC로 표현할 수 있고 (g)는 A(#, 2)A(#, 1), BC(#, -3)로 표현된다.
- ⑧ 심볼 A의 왼쪽(오른쪽) 문맥은 A로(로부터) 연결된 모듈이나 기능그룹을 의미한다. 이것은 L-시스템에서 심볼을 기준으로 매칭시키는 것과는 다르다. 가령  $A < C \rightarrow D$ 라는 production을 A(#, 1)BC에 적용할 때 L-시스템인 경우 C의 왼쪽 문맥은 B가 되어 production이 적용되지 않으나 우리가 정의한 바에 의하면 C는 A에서 입력을 받고 있으므로(그림 4) A는 C의 왼쪽 문맥이 된다. 따라서 production이 스트림에 적용되어 결과적으로 A(#, 1)BD가 된다.
- ⑨ '\*'는 문맥에서 wild card 역할을 한다. 심볼 A의 왼쪽(오른쪽) 문맥으로 쓰이면 A가 모듈이나 기능그룹으로부터(에게) 입력(출력)을 받고(보내고) 있으면 문맥에 맞다.  $* < C \rightarrow D$ 와 같은 production에서 그림 4와 같은 A(#, 1)BC 경우에 적용하였을 때 C는 B나 A로부터 입력을 받고 있으므로 production이 적용되어 A(#, 1)BD가 된다.

##### 4.2 네트워크의 구조 결정

모듈형 네트워크의 구조를 적절히 표현하기 위한 문법을 정의하였으므로 여기서는 이를 이용하여 네트워크의 구조를 유도해내는 과정을 보이고자 한다. 문법을 이용한 복잡한 네트워크의 구조도 몇가지의 간단한 production 규

<sup>1</sup>여기서 #는 don't care를 의미한다. 단순히 다음번 1이 두 번째 파라미터라는 것을 보여주기 위하여 넣은 것이다.

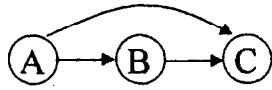


그림 4:  $A(\#,1)BC$ .

칙으로 표현 가능하다. 예를 들어서 다음과 같이 심볼과 axiom, 그리고 production이 주어졌다고 가정하면.

심볼 :  $\{A-Z\} \cup \{,\} \cup \{*\}$

Axiom :  $A(20,0)$

production:

$$A(x,y) \rightarrow B(x,y)C(x/2,y)D(x/4,y)$$

$$C(x,0) > * \rightarrow D(x,1)E(2,0)$$

$$B(x_1,y_1) < D(x,y) > D(x_2,y_2) \rightarrow [B(x,y), B(x,y)]$$

$$E(x_1,y_1) < D(x,y) : y == 0 \rightarrow D(x,-1)$$

(7)

이와 같이 주어졌을 때 전개과정은 식 8과 같다.

$$A(20,0) \rightarrow B(20,0)C(10,0)D(5,0)$$

$$\rightarrow B(20,0)D(10,1)E(2,0)D(5,0)$$

$$\rightarrow B(20,0)[B(10,1), B(10,1)]E(2,0)D(5,-1)$$

(8)

식 8과 같은 유도과정에서 형성되는 네트워크의 구조는 그림 5와 같다.

## 5. 결론

본 논문에서는 모듈형 신경망을 진화적으로 개발하는 방법론의 하나로서 문법을 이용한 방식을 소개하였다. 복잡한 구조의 네트워크라고 하여도 간단한 몇가지 production 규칙에 의하여 유도될 수 있음을 볼 수 있었다. 따라서 이러한 방법을 진화방식을 이용한 네트워크의 구조결정문제에 적용하면 보다 짧은 유전자코딩으로도 보다 크고 복잡한 네트워크를 구성할 수 있을 것이다.

이러한 예비 연구결과를 바탕으로 이후에 진화적으로 production 규칙을 결정하여 주어진 문제에 가장 적합한 네트워크의 구조를 찾는 연구를 계획 중이다.

### 참고 문헌

- [1] D.T. Cliff, I. Harvey and P. Husbands, "Incremental evolution of neural network architectures for adaptive behavior," *Technical Report CSRP 256*, University of Sussex School of Cognitive and Computing Science, 1992.
- [2] X. Yao, "Evolutionary artificial neural networks," *Int. Journal of Neural Systems*, vol. 4, no. 3, pp. 203-222, 1993.

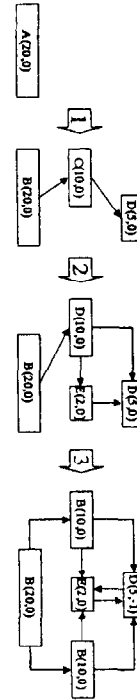


그림 5: 네트워크 형성과정.

- [3] S. Nolfi, O. Miglino and D. Parisi, "Phenotypic plasticity in evolving neural networks: Evolving the control system for an autonomous agent," *Technical Report PCIA-94-04*, Institute of Psychology, C.N.R., Rome, 1994.
- [4] G.F. Miller, P.M. Todd and S.U. Hedge, "Designing neural networks using genetic algorithms," in *Proc. Third Int. Conf. on Genetic Algorithms and Their Applications*, pp. 379-384, Morgan Kaufmann, San Mateo, CA, 1989.
- [5] M.A. Arbib, *The Handbook of Brain Theory and Neural Networks*, MIT Press, 1995.
- [6] J.M.J. Murre, R.H. Phaf and G. Wolters, "CALM: Categorizing and learning module," *Neural Networks*, vol. 5, pp. 55-82, 1992.
- [7] P. Prusinkiewicz, M. Hammel, J. Hanan, R. Mech, "Visual Models of Plant Development," *Handbook of Formal Languages*, Springer-Verlag, 1996.
- [8] E. Boers, H. Kuiper, "Biological metaphors and the design of modular artificial neural networks," *Master's Thesis*, Dept. of Computer Science and Experimental and Theoretical Psychology, Leiden Univ., Netherlands, 1992.