

린덴마이어-시스템의 진화를 통한 모듈형 신경망의 개발

이 지행, 조 성배
연세대학교 컴퓨터과학과

Development of Modular Neural Networks by Evolving Lindenmayer-System

Jee-Haeng Lee and Sung-Bae Cho
Computer Science Department, Yonsei University

요 약

모듈형 신경망은 인간의 정보처리 시스템이 고유한 목적이나 기능을 가진 모듈로 되어있다는 신경과학의 연구에 기반하여 제안된 모델이다. 하지만 모듈의 크기와 기능모듈간의 연결구조를 결정하는데 큰 어려움이 있다. 본 논문에서는 간단한 규칙으로 복잡한 구조를 생성해 낼 수 있는 린덴마이어-시스템을 이용하여 모듈형 신경망의 크기 및 연결구조를 만들어내는 과정에 대하여 고찰해본다 또한, 신경망의 생성규칙을 유전자형으로 표현하고 진화 알고리즘을 적용하여 주어진 문제를 해결할 수 있는 최적의 규칙을 찾아내는 방법을 제안한다. 본 논문에서 제안한 유전자형과 진화연산은 최적화된 문법규칙 및 신경망의 구조를 만들어 낼 수 있는 가능성을 보여준다.

1. 서론

생물의 두뇌구조는 오랜 시간의 진화과정을 거쳐 형성되어 생존과 번식에 필요한 기능을 수행하면서 특수화된 부시스템의 집합으로 생성되었음이 주지되고 있다 또한, 공학적으로도 대규모의 문제를 단순한 처리기로부터 학습시키는 것은 엄청난 시간을 소요한다는 사실을 알 수 있다 이런 이유로 기능단위인 모듈을 구성요소로 하고 그들의 연결을 통해 주어진 문제를 해결하는 신경망에 대한 연구가 많이 이루어졌으며, Murre[1] 등은 두뇌의 시피질내의 미니컬럼을 모형화한 모델을 제안하였다 하지만 모듈로 이루어진 신경망 시스템을 설계하는 경우에 주어진 기능을 수행해 낼 수 있는 구조를 결정하여야 하는데, 아직까지는 완전한 해답이 존재하지 않는다.

한편, 신경망 시스템 설계를 진화방식으로 해결하고자 하는 연구는 환경에 적절히 대응하는 지능시스템을 개발하는데 키나란 가능성을 보여주고 있다 진화적인 접근방식은 문제에 따라 손으로 신경망의 구조를 만들던 과거의 방법과는 달리 보다 일반적이고 효과적인 해결방법을 제시해주며, 이미 많은 연구결과가 발표된 바 있다[2,3]. 하지만, 신경망의 구조를 진화시키기 위해서는 효과적인 유전자형의 설계가 중요한 문제이다 신경망의 구조를 직접 유전자형에 표현하는 경우에는 작은 크기의 신경망에는 문제가 발생하지 않지만 표현해야하는 신경망의 크기가 커지면 커질수록 유전자형의 크기도 함께 커져야 하며, 진화알고리즘의 탐색 공간이 원저히 늘어나게 된다.

본 논문에서는 문제의 복잡성에 따라서 유전자 표현이 점차 커지고 탐색 시간 또한 증가하는 문제점을 해결하기 위하여 린덴마이어-시스템의 문법을 이용한 유전자 표현법을 개발하였다 문법을 이용하여 네트워크의 구조를 기술하게 되면 네트워크 크기의 증가가 유전자 표현에 따르는 크기의 증가와는 거의 무관하기 때문에 복잡한 문제일수록 이러한 표현방법을 사용하는 것이 바람직하다[4,5]. 본 논문에서 제시한 유전자형과 진화연산의 적용은 문법규칙 및 신경망의 구조를 효과적으로 만들어 낼 수 있는 가능성을 보여준다

2. 연구배경

2.1 모듈형 신경망

각 노드의 활성화 값은 식 (1)과 같이 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 것이다 입력값은 양의 값(흥분)이나 음의 값(억제)을 가질 수 있다 이러한 노드들로 구성된 모듈은 기본적으로 두뇌의 시피질내에 존재하는 미니컬럼을 모형화한 Murre[1]의 모델을 따른다.

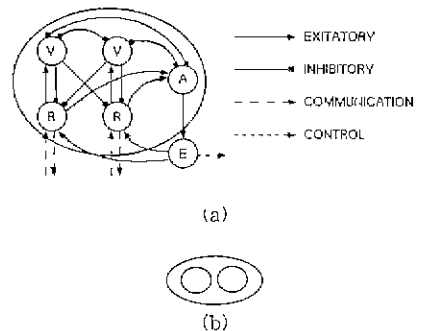


그림 1. (a) 모듈내부의 구조에 대한 대략적인 구조. (b) (a)의 모듈을 간략화한 표현. 두개의 원은 모듈내에 R-V 노드쌍이 두개 있음을 의미한다.

각 모듈의 기본적인 내부구조는 그림 1(a)와 같이 고정되어 있으며, 모든 모듈내의 연결강도들은 학습중에 변경되지 않는다. 모듈내에서 R-노드는 해당 모듈로 입력되는 특정한 패턴을 나타내고, V-노드는 모듈내의 다른 모든 노드들을 억제하며, A-노드는 모듈내의 경쟁정도에 따라 양의 함수를 활성화하여 E-노드로 하여금 경쟁의 정도를 조절하도록 한다. 모듈의 가장 중요한 특성은 입력되는 활성화 패턴을 자율적으로 서로 다른 부류로 분류함으로써, 주어진 입력패턴을 유일한 R-노드와 연관시키는 것이다.

한편 두 모듈간의 연결은 한 모듈에 있는 모든 R-노드와 다른 모듈에 있는 모든 R-노드가 연결되어 있음을 의미한다. 이러한 모듈사이의 연결강도는 식 (2), (3)과 같은 헤브의 규칙에 의해 변경된다.

$$\Delta w_{ij}(t+1) = \mu_i a_j (K - w_{ij}(t)) a_j - L w_{ij}(t) \sum_{j'} w_{ij'}(t) a_j, \quad (2)$$

$$\mu_i = d + w_{ij} a_E \quad (3)$$

단, a_i 와 a_j 는 각각 해당하는 R-노드의 활성화 정도를 나타내며, $w_{ij}(t)$ 는 R-노드 j 와 i 사이의 연결강도를 의미하고 $w_{ij}(t)$ 는 R-노드 i 와 이웃하는 R-노드 f 사이의 연결강도를 나타낸다. 그리고 $\Delta w_{ij}(t+1)$ 은 $t+1$ 시간에 j 와 i 를 연결하는 연결강도의 변화량이다 여기에서 L 과 K 는 양의 상수이고, d 는 작은 상수이며, a_E 는 E-노드의 활성화 값이다.

2.2 린덴마이어-시스템

린덴마이어-시스템(L-시스템)은 1968년 A. Lindenmayer가 제안한 것으로 다세포 생물의 성장과정을 모방하고 시뮬레이션하기 위한 rewriting system이다[5] 린덴마이어-시스템은 병렬 스트림 치환 속성을 지니는 일종의 문법으로서 이후의 연구에서는 주로 식물의 성장과정을 모델링하기 위한 도구로 많이 적용되어 왔다. 린덴마이어-시스템은 기본 L-시스템, Bracketed L-시스템, 문맥 의존적 L-시스템 등이 있으며, 모델링하는 대상에 따라 효과적으로 적용된다.

간단한 L-시스템만 가지고 식물의 성장과정을 모델링하기에는 표현력에 한계가 있기 때문에 이를 해결하기 위하여 Bracketed L-시스템이 제안되었다[5] 이 방법의 핵심은 '[' 와 ']' 를 사용하여 식물의 가치를 묘사하는 것이다. 즉, 괄호로 묶여진 부분을 식물의 가치로 해석하며, 재귀적으로 더 작은 가치가 그 묶음내에 존재할 수 있다

Bracketed L-시스템을 확장한 파라미터 L-시스템은 전개되기 전의 어떤 상태를 지정하고 그 상태를 다음 전개될 식에 반영하기 위한 방법으로 사용된다. 파라미터 L-시스템 문법의 정의는 다음과 같다.

$$G = \langle V, \sum, w, P \rangle$$

V: 알파벳

\sum : 형식 파라미터의 집합

$w \in (V \times R^*)^*$: 공리라고 불리는 파라미터 단어

$P \subset (V \times \Sigma^*) \times C(\Sigma) \times (V \times E(\Sigma)^*)$: 규칙의 유한 집합

$C(\Sigma)$: Σ 로 이루어진 논리식

$E(\Sigma)$: Σ 로 이루어진 연산식

문맥 의존적 파라미터 L-시스템은 파라미터 L-시스템에서 문맥을 고려하도록 확장한 것이다. 식 (4)에서처럼 주위의 문맥에 따라 다음 상태로 전개되며, 신호의 전달, 식물의 수분흡수, 환경과의 상호작용 등의 모델링에 응용되고 있다.

$$lc \langle pred \rangle rc : cond \rightarrow succ \quad (4)$$

단, '<'와 '>'는 규칙의 세가지 구성요소를 나누는 기호이며 lc 는 왼쪽 문맥, rc 는 오른쪽 문맥을 의미한다. $cond$ 는 이 규칙이 적용될 조건을 표시한다.

3. 생성규칙의 설계 및 진화

3.1 신경망의 생성을 위한 문법의 정의

문법을 이용하여 모듈형 신경망을 표현하기 위하여 약간 변형된 문맥 의존적 파라미터 L-시스템을 사용하였다. 심볼 하나는 하나의 모듈을 의미하며, 모듈의 크기와 연결의 위치를 표현할 수 있는 두 개의 파라미터를 가지고 있다. 그리고, 괄호로 둘러싸인 문장을 '기능그룹(Functional Group)'으로 정의하여 문법을 전개하고, 구조를 생성해낼 때 쉼표기호를 이용하여 다양한 연결을 만들어 낼 수 있도록 하였다.

기본적으로 이웃한 두 모듈이나 기능그룹은 연결된다. 만약 문장 사이에 쉼표가 있으면, 이 연결은 끊어지게 된다. 기능그룹으로 들어가는 연결은 기능그룹의 맨 처음 모듈로 해주며, 기능그룹 내에 쉼표가 있으면 쉼표 오른쪽의 모듈에도 연결시킨다. 반대로 기능그룹에서 밖으로 연결할 때는 기능그룹의 맨 마지막 모듈에서 연결이 되며, 기능그룹 내에 쉼표가 있으면 쉼표 왼쪽의 모듈에도 연결이 된다.

심볼의 왼쪽(오른쪽) 문맥은 심볼에 연결된 상태를 따라 결정된다. 이것은 식물 성장 모델링을 위한 L-시스템에서 괄호내의 그룹을 끝가지로 생각하고 문맥에서 제외시키는 것과는 다르다. 기본적으로 식이 전개되기 위한 파라미터 조건의 제약은 없는 것으로 한다. 다만 모듈의 크기가 음수가 되거나 너무 커지는 것을 막기 위하여, 그 범위를 만족하지 않을 경우 각각 최대값 및 최소값으로 대응시킨다.

이와같이 정의된 문법의 예는 다음과 같다:

$$V : \{A-C\} \cup \{', '\} \cup \{ '[', ']' \}$$

$$w : A(20,0)$$

$$p_1 : A(x,y) \rightarrow B(x,y), C(x/2,y)$$

$$p_2 : C(x_1,y_1) < B(x,y) \rightarrow C(x-1)$$

3.2 유전자형의 표현

규칙 하나의 유전자 코딩이 전체 문법 유전자형의 단위가 된다. 유전자의 전체 구조는 그림 2와 같다. 하나의 규칙 표현을 위한 유전자형은 그림 2(b)와 같이 현재의 심볼 이름과 좌우의 문맥 및 다음에 전개될 식의 심볼 정보 집합으로 되어있다. 전개될 식의 심볼은 그림 2(c)와 같이 자신의 심볼 이름, 파라미터의 정보, 심볼 자신 왼쪽의 괄호와 쉼표 정보를 저장하고 있다. 좌우의 문맥 및 전개될 식의 심볼이 문법에서 정의 되지 않은 알파벳일 경우 무의미한 것으로 간주한다.

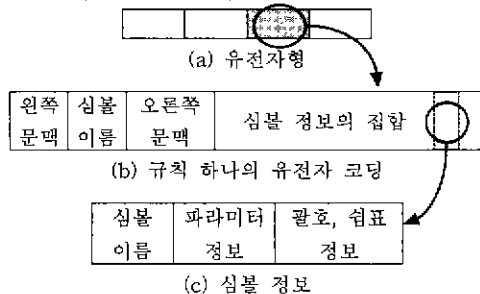


그림 2. 유전자형의 구조

본 논문에서는 간단하고 효율적인 유전자 코딩을 위하여 좌우의 문맥을 4개 이하로 제한하였으며, 새로 전개될 식의 심볼수를 10개 이하로 하였다. 파라미터는 현재 전개되어야 할 자신만의 정보를 이용하였다. 만약 생성된 규칙이 팔호의 좌우 개수가 맞지 않을 경우 문법을 만족하는 규칙으로 교체하는 보정과정을 거쳐 완성된 하나의 규칙을 만들어낸다 그림 3은 정의된 유전자형으로 문법규칙을 생성하고 전체 신경망의 구조를 만들어 내는 실제 예를 보여준다. 여기에서 '#'은 문법에서 정의되지 않은 알파벳을 의미하며, 심볼 정보의 집합은 그 심볼의 이름만을 나타내었고, 좌우 문맥의 파라미터는 사용하지 않으므로 표기를 생략하였다

####	A	####	#	#	B	B	#	#	C	#	#	D
####	B	#C##	D	B	#	#	#	#	#	#	#	#
B###	C	D###	#	#	#	#	#	#	D	#	#	#
####	D	####	#	#	#	D	#	#	#	#	#	#

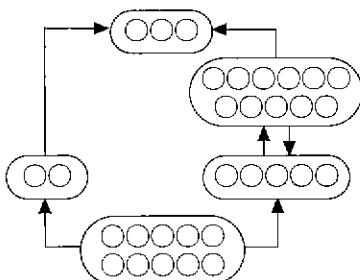
(a) 실제 유전자형의 정보

$A(x,y) \rightarrow B(x,y)[B(x/2,y)C(x-1,-1)]D(3,4)$
 $B(x,y) > C \rightarrow D(x/2,y/2),B(x,y)$
 $B < C(x,y) > D \rightarrow D(x+2,y-1)$
 $D \rightarrow D(x,y/2)$

(b) 유전자형으로부터 해석된 문법

$A(10,0)$
 $\rightarrow B(10,0)[B(5,0)C(9,-1)]D(3,4)$
 $\rightarrow B(10,0)[D(2,0),B(5,0)D(11,-2)]D(3,2)$
 $\rightarrow B(10,0)[D(2,0),B(5,0)D(11,-1)]D(3,1)$

(c) 문법을 통해 전개된 문장



(d) 생성된 신경망의 구조

그림 3. 실제 신경망의 생성과정

3.3 진화 알고리즘

규칙의 모임이 하나의 문법이 되는데, 규칙의 개수가 유동적이므로 전체 유전자형의 크기는 동일하지 않다 돌연변이 연산은 유전자형의 내용을 변화시키는 것 외에 규칙 개수의 증가와 감소를 유도한다. 그림 4(a)의 검은 부분은 돌연변이 연산이 적용된 부분을 의미하며, 규칙의 개수가 하나 늘어난 연산이 식용되었음을 알 수 있다. 그림 4(b)처럼 교차 연산은 유전자의 단위인 규칙 유전자형을 기준으로 이루어진다

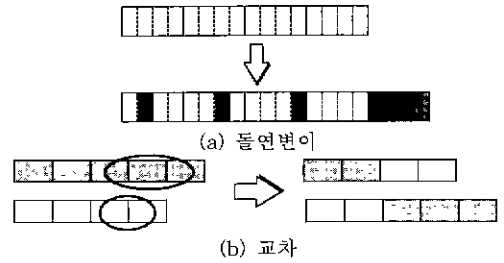


그림 4. 유전자 연산자

이런 유전자형과 진화연산을 바탕으로 주어진 문제를 해결하기 위한 진화 알고리즘을 수행한다. 유전자를 해석하여 L-시스템 문법을 만들어내고, 이를 이용하여 모듈형 신경망을 생성해낸다. 이 신경망을 문제에 적용하여 적합도를 계산하며, 이것을 바탕으로 초기에 임의로 생성된 개체들이 문제해결에 적합한 구조로 진화된다.

4. 결론

일반적으로 신경망의 구조 및 크기를 결정하는 데에는 많은 어려움이 따른다. 이의 해결을 위하여 본 논문에서는 린덴마이어-시스템의 진화적 접근에 대하여 고찰해 보았으며, 그 문법을 표현할 수 있는 유전자형을 설계하였다. 이러한 유전자형은 린덴마이어-시스템의 문법체계를 만족하는 임의의 규칙을 효과적으로 만들어 낼 수 있다. 현재 이 방법을 이용하여 실제 문제에 적용하여 모듈형 신경망을 학습시키는 연구가 진행 중에 있으며, DNA 코딩 방법[6]을 이용한 유전자형의 표현에 대해서도 가능성을 모색하고 있다.

참고문헌

- [1] J.M.J. Murre, R.H. Phaf and G. Wolters, "CALM: Categorizing and learning module," *Neural Networks*, vol. 5, pp. 55-82, 1992
- [2] X. Yao, "Evolutionary artificial neural networks," *Int. Journal of Neural Systems*, vol. 4, no. 3, pp. 203-222, 1993.
- [3] 조성배, "진화방식에 의해 개발된 모듈형 신경망의 기능분석," *한국정보과학회 논문지(B)*, 25권 3호, pp. 417-425, 1998년 3월.
- [4] E. Boers and 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.
- [5] P. Prusinkiewicz, M. Hammel, J. Hanan and R. Mech, "Visual models of plant development," *Handbook of Formal Languages*, Springer-Verlag, 1996
- [6] T. Yoshikawa, T. Furuhashi and Y. Uchikawa, "Emergence of effective fuzzy rules for controlling mobile robots using DNA coding method," *Proc. of 1996 IEEE Int. Conf on Evolutionary Computation*, pp. 581-586, 1996.