

유전자 알고리즘을 이용한 선형 신경 회로망 합성 방법

A Genetic Algorithm Approach to Linear Threshold Neural Network Synthesis

박주현, 이정훈

Computational Vision and Fuzzy Systems Laboratory

한양대학교 전자공학과

425-791 경기도 안산시 사1동 1271번지

{jhpark, frhee}@fuzzy.hanyang.ac.kr

Abstract

신경회로망은 높은 정확도의 학습 결과를 제시하는 장점을 가지고 있어서 패턴 인식을 포함한 여러 분야에서 널리 사용되어지고 있다. 그러나 신경회로망의 설계에 있어 최적의 뉴런과 층의 개수, 그리고 그 연결 등의 기하학적 해답을 제시 하기가 어렵고 성능이 우수하다고 알려진 역전파 학습 알고리즘도 오차가 없는 완벽한 학습 결과를 제시하지 못하며, 상당히 많은 학습 시간이 걸린다는 단점들을 가지고 있다. 이러한 단점들을 극복하기 위해 선형 신경회로망을 합성하는 새로운 방법을 제안하는데, 이진 함수 최소화 (binary function minimization) 과정을 거친 minimal-sum-of-product (MSP)를 통해서 이진 클래스 패턴 (binary class pattern)을 표현 함으로써 오차가 없는 학습 결과를 얻을 수 있으며, 학습에 필요한 패턴과 학습에 걸리는 시간도 대폭 줄일 수 있다. 본 논문에서는 유전자 알고리즘을 이용하여 선형 신경회로망을 합성하는 방법을 제안하며, 여러 가지 예제를 통해 제안한 방법의 우수성을 보인다.

I. 서론

최근에 들어 유전자 알고리즘 (Genetic Algorithm)이 광범위한 최적화 문제의 해결에 강력한 도구로 널리 사용되어지고 있다[1]. 본 논문에서는 패턴공간을 선형적으로 구분할 수 있는 linear threshold unit (LTU)을 정의하는 변수 (i.e., weight, threshold)들을 찾는데 유전자 알고리즘을 이용하는데, 신경회로망의 입력 데이터를 본 알고리즘의 target 패턴으로 놓고 유전자 알고리즘의 탐색과정을 행한다. 그러나 이 방법의 문제점은 target 패턴이 커질 경우 유전자 알고리즘이 해답을 찾는데 걸리는 시간이 너무 커지는 단점을 가지므로, 이를 보완 하기 위해 minimal-sum-of-product (MSP)가 이진함수의 최소화된 표현이라는 특성을 이용하여 유전자 알고리즘의 target 패턴의 수를 큰 폭으로 줄일 수 있다.

일반적인 유전자 알고리즘을 이용하여 다층의 선형 신경회로망을 합성하기 위하여, 이진 함수 최소화[2]와 threshold logic[3]등의 이론을 사용하며, 제안되는 이론은 아래의 세 과정을 연속적으로 수행한다.

- 1) 주어진 이진 클래스 패턴을 표현하는 minimal-

sum-of-product (MSP) 함수를 찾는다.

- 2) 각 linear threshold unit (LTU)에 해당하는 target 패턴을 대표하는 새로운 줄어든 target 패턴을 찾는다. 새롭게 만들어진 target 패턴을 유전자 알고리즘에 적용하기 위해 선형 부등식의 집합으로 만든다.
- 3) 선형 부등식의 집합에 일반적인 유전자 알고리즘을 적용하여 선형 신경회로망의 변수들을 찾는다.

II. 본론

2.1 Minimal-Sum-of-Product (MSP)와 Linear Threshold Unit (LTU)의 선형 분할성

Minimal-sum-of-product는 이진 함수의 최소화된 표현으로서 이를 구하는 방법으로는 K-map 방법이 주로 사용되어지는데, 계산량이 너무 많고 또 복잡하며 10개 이내의 변수를 가지는 문제가 아니면 사실상 계산이 거의 불가능한 단점을 가지고 있어서 K-map 방법의 이러한 단점들을 모두 해결하는 방법인 tabular method[2]를 이용해 MSP를 구한다.

임의의 MSP 함수 $f(x_1, x_2, \dots, x_n)$ 에 대해, f 의 변수 x_i 가 uncomplemented(complemented)의 형태로 나타날 때, x_i 를 f 를 대해 positive(negative)하다고 하며, 또 f 에서 x_i 가 모두 positive이거나 negative한 형태로 나타날 때, x_i 는 f 에 대해 unate하다고 한다. 그리고

Acknowledgement

이 논문은 1995년도 한국학술진흥재단의 대학부설연구소 연구과제 연구비에 의하여 연구되었음

만약 함수 f 의 모든 변수가 unate하면, 그 함수 f 를 unate하다고 한다. 주어진 이진함수를 MSP의 형태로 표현하였을 때, 어떠한 입력 변수도 positive 형태와 negative 형태가 동시에 존재해서는 한 개의 linear hyperplane으로 분할할 수 없다는 unate 특성은 LTU의 중요한 특성이다. 그러나 이 성질은 필요 조건일 뿐 unate 성질을 만족한다고 해서 패턴 공간을 항상 선형적으로 분할 할 수 있는 것은 아니다[3].

Linear threshold unit (LTU)는 threshold neural network (TNN)을 구성하는 가장 기본단위로서, 각 뉴런의 입력의 가중합이 임계치 보다 클 때만 1이 출력되고 그 외에는 0이 출력되는 구조이다. 식 (1) 은 TNN의 기본 단위인 LTU를 나타내는 식이며, 식 (2) 의 벡터 V 는 LTU를 정의하는 weight-threshold vector를, 그리고 그림 1 은 가장 기본적인 LTU의 구조를 보여준다.

$$y = 1 \text{ if and only if } \sum_{i=1}^N w_i x_i - T \geq 0 \quad (1)$$

$$y = 0 \text{ if and only if } \sum_{i=1}^N w_i x_i - T < 0$$

$$V = \{w_1, w_2, \dots, w_N; T\} \quad (2)$$

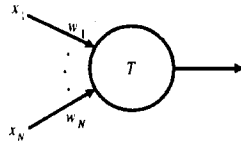


그림 1. 기본적인 Linear Threshold Unit (LTU)의 구조.

2.2 Linear Threshold Neural Network (TNN)의 합성

LTU를 구성하는데 필요한 변수들을 구하기 위해서 유전자 알고리즘이 이용되는데, 아래에서는 유전자 알고리즘을 이용하여 MSP함수로부터 LTU가 합성되는 과정을 간단한 예제를 들어보인다. 먼저 다음과 같은 이진 함수가 있다고 가정하자.

$$f(x_1, x_2, x_3, x_4) = \sum(2, 3, 6, 7, 10, 12, 14, 15)$$

이진 함수 최소화 방법인 tabular method[2]를 이용하면 주어진 함수의 MSP형태는 $f = x_1x_3 + x_2x_3 + x_1x_4' + x_2x_4'$ 이며, f 는 unate하지 않다. 즉, 선형적으로 분할 할 수 없다. 따라서 최소한 두개의 threshold element를 cascade 해서 TNN을 합성할 필요가 있다. f 의 MSP에서, 앞의 세 term과 마지막의 한 term이 각각 서로 unate하므로, 두개의 unate한 그룹을 각각 함수 g 와 h 로 나타낸다. 두 함수 g 와 h 는 각각 positive function $g' = x_1x_3 + x_2x_3 + x_1x_4$, $h' = x_2x_4$ 로 변환된다.

Minimal true vertex들과 maximal false vertex들은 MSP함수가 표현하는 모든 vertex들을 대표하며, minimal true, maximal false vertex들을 선형적으로 분할

하는 linear hyperplane은 MSP함수를 선형적으로 분할하는 linear hyperplane이 된다. 이러한 특성을 이용하면 유전자 알고리즘의 target 패턴들의 수를 대폭 줄일 수 있다. 위에서 구해진 positive 함수에서, g' 의 minimal true vertex들은,

$$(1010) \quad (0110) \quad (0011) \text{이며,}$$

그리고, maximal false vertex들은,

$$(0100) \quad (1101) \text{ 이 된다.}$$

아래의 식 (3) 은 minimal true, maximal false vertex들을 선형적으로 분할하는 부등식이다.

$$\left. \begin{matrix} w_1 + w_3 \\ w_2 + w_3 \\ w_3 + w_4 \end{matrix} \right\} > \begin{cases} w_2 \\ w_1 + w_2 + w_4 \end{cases} \quad (3)$$

위의 식 (3) 에서 함수 g' 의 weight constraint들을 의미하고, 유전자 알고리즘의 target 패턴들로 사용되는 아래와 같은 6개의 부등식을 얻을 수 있다.

$$\begin{matrix} w_1 + w_3 > w_2 & w_3 > w_2 + w_4 \\ w_3 > 0 & w_3 > w_1 + w_4 \\ w_3 + w_4 > w_2 & w_3 > w_1 + w_2 \end{matrix}$$

첫번째 element의 weight-threshold vector와 두 번째 element의 weight-threshold vector는 유전자 알고리즘을 이용한 탐색과정을 통하여 구해지며 각각 다음과 같다.

$$V_g = \{1, 1, 7, 1; 7.5\}, V_h = \{2, 2, 0, 2; 5\}$$

원래 함수 g 와 h 의 weight-threshold vector는 처음에 positive로 변환시켰던 negative 변수의 weight를 원래대로 negative로 바꿔준 후, 변환된 weight만큼 threshold에서 빼주면 된다. 최종적인 weight-threshold vector는 다음과 같다.

$$V_g = \{-1, 1, 7, -1; 5.5\}, V_h = \{2, 2, 0, -2; 3.0\}$$

그림 2 는 최종 결과로 구해진 두개의 element를 cascade 한 network이다.

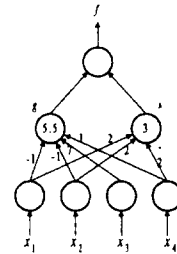


그림 2. 함수 f 를 합성한 선형 신경회로망의 구조.

2.3 TNN합성을 위한 유전자 알고리즘

유전자 알고리즘은 적자생존과 유전학의 구조에 근거하여 확률적인 탐색을 하는 탐색방법 이다. 유전자 알고리즘은 individual (각각의 individual의 집합을 population이라 한다)이라 불리 우는 initial

solution으로 시작하는데, 이러한 individual들은 crossover, mutation 등의 유전자 알고리즘 연산을 통해 보다 우수한 새로운 individual을 생성하는 반복적인 진화과정을 거치게 되고, 유전자 알고리즘의 마지막 부분에서는 individual의 우수한 정도를 나타내는 fitness에 의거하여 최상의 individual을 탐색결과로 제공한다[1][7]. 기본적인 유전자 알고리즘은 다음과 같이 설명된다.

```

Create initial population of fixed size;
Do{
    Choose parent1, and parent2 from population;
    Offspring = crossover (parent1, parent2);
    Mutation (offspring), THEN
    Replace (population, offspring);
}
until (stopping condition);
Report the best answer;

```

본 논문에서 유전자 알고리즘은 minimal true vertex들과 maximal false vertex들을 선형적으로 구분하기 위해 각 LTU의 weight-threshold vector를 구하는데 목적이 있다. 각 individual을 n bit인 $M(N+1)$ 개의 block을 갖는 binary bit string으로 표현함으로써 하나의 individual로 정수값을 갖는 임계치의 집합을 나타낼 수 있다. 여기서 M 은 LTU의 개수를, N 은 weight의 개수를 의미하며, 그림 3에 이 내용이 나타나 있다.

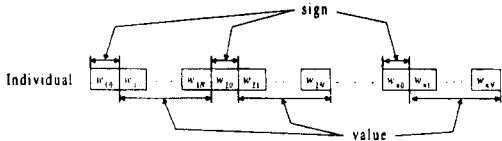


그림 3. Individual의 구조.

Evaluation measure는 각 individual의 우열의 정도를 나타내는 fitness를 결정하는데, minimal true, maximal false vertex들을 선형적으로 분할하는 부등식을 target pattern으로 선택하면, evaluation measure과정에서 각 individual은 target패턴들의 전부 혹은 일부를 만족시키게 되며, individual이 만족시키는 target 패턴의 숫자가 각 individual의 fitness로 할당된다.

2.4 실험 결과

제안된 알고리즘의 성능을 보이기 위하여 몇 가지 실험 결과와 기존의 알고리즘을 사용한 실험 결과를 비교해본다.

A. 양자화된 원형 영역의 근사

먼저, 제안된 알고리즘을 비교하기 위하여 이전에 발표된 논문[5][6]에서 사용한 예제를 사용한다. 이 실험의 목적은 2차원 공간에서 원형 영역을 그 배경으로부터 선형적으로 구분 해내는 것이다. 그림 4에서 보는 것과 같이 지름이 4인 원이 가운데(4, 4) 위치해 있는 영역의 각 좌표는 3bit으로 양자화 되는데, (3개의 변수들로 8개의 레벨을 표현할 수 있다.) 이는 모든 영역을 64개의 동일한 정 사각형 영역으로 분할한다. 이렇게 분할된 영역에서, 52개의 정사각형은 원의 바깥을 나타내고(0의 출력 값으로 표현됨), 그리고 12개의 정사각형은 원은 내부를 나타내게 된다(1을 출력함).

아래의 함수는 원형 영역을 3bit으로 양자화한 결과를 나타내는 MSP함수 이다.

$$f = (x_1'x_2x_4'x_5x_6 + x_1'x_2x_3x_4'x_5) + (x_1'x_2x_3x_4x_5' + x_1'x_2x_4x_5'x_6) + (x_1x_2'x_4'x_5x_6 + x_1x_2'x_3'x_4'x_5) + (x_1x_2'x_3'x_4x_5' + x_1x_2'x_4x_5'x_6)$$

구해진 MSP함수에는 4개의 unate한 그룹이 존재한다 (위의 함수에서 ()로 묶여진 부분이 unate한 하나의 그룹을 나타낸다). 각각의 unate한 그룹은 minimal true와 maximal false vertex들을 찾기 위해 모두 양의 함수로 변환되며, 구해진 minimal true와 maximal false vertex들로부터 부등식의 집합 (target 패턴)이 구해진다. 다음 단계에서는 유전자 알고리즘을 이용하여 LTU의 weight-threshold vector를 구하는데, 유전자 알고리즘의 변수들의 조건은 initial population은 30, crossover rate는 0.6 (60%), crossover point는 6개, mutation rate는 0.001 (0.1%)로 두었으며, 하나의 individual에서 각 weight가 차지 하는 bit수를 2bit으로 설정하였으므로, 하나의 individual을 위해 모두 12bit의 bit-string이 필요 하게 된다.

이 실험에서 유전자 알고리즘으로 LTU의 weight-threshold vector(인식을 100%)를 찾아내는데 2번의 generation과 178번의 evaluation의 과정이 걸리는데 (실제 계산시간은 수초 이내), 실험 결과로 합성된 TNN은 4개의 은닉층을 가진다. 합성된 Linear TNN의 각 LTU의 가중치와 임계치들의 값은 그림 5와 표 1에 나타나 있다.

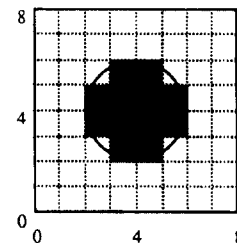


그림 4. 6bit으로 양자화된 원형 영역.

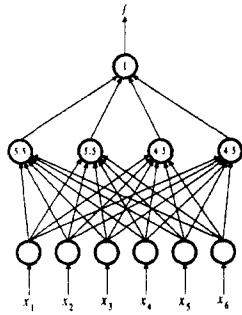


그림 4. 양자화된 원형 영역을 분할 하기위해 제안한 방법을 이용하여 합성한 신경회로망.

표 1. 양자화된 원형 영역을 분할하는 신경 회로망의 변수.

Region	Weight						T
1	-3	3	1	-3	2	1	5.5
2	-3	3	1	3	-2	-1	5.5
3	3	-3	-1	-3	2	1	4.5
4	3	-3	-1	3	-2	-1	4.5

기존에 제안된 방법과 결과를 비교해 보면, Expand-and-Truncate Learning (ETL)방법(6)은 5개의 은닉층을 가지며, Boolean-like training algorithm (BLTA)방법(5)은 16개의 은닉층을 가진다.

B. Iris Data 의 인식

두 번째 예제는 패턴 인식 분야에서 널리 사용되어지고 있는 Iris data의 패턴 인식에 관한 문제이다. Iris data는 3개의 class와 4개의 feature를 가지며, 각 class는 50개의 패턴을 포함하고 있다. 이번 실험에서는 unequal partition이 적용되는데, 이는 partition된 하나의 영역에 다른 class의 패턴이 존재 하지 않도록 partition을 하는 방법이다. Unequal partition으로 Iris data를 나누는 데에는 6bit 가 필요 하며, 각 class마다 1,2,2,1 bit가 각각 사용되었다.

그림 6 과 표 2 는 제시된 방법으로 합성한 선형 신경회로망의 구조와 회로망의 가중치와 임계치를 보여준다.

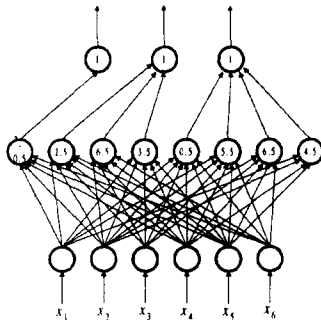


그림 5. 제안한 알고리즘을 이용하여 합성한 Iris data 인식을 위해 합성된 신경 회로망.

표 2. Iris data인식을 위해 합성된 신경 회로망의 변수.

Class	Weight						Threshold
1	-1	-3	0	-1	-3	-3	-0.5
2	-1	2	-1	-3	-3	3	2.5
	0	2	-3	2	1	2	6.5
	-2	2	1	-3	1	0	3.5
3	-3	-1	-3	-1	3	-2	0.5
	-2	3	-3	1	-3	2	5.5
	0	2	1	2	3	-2	6.5
	1	-1	0	3	1	0	4.5

III. 결론

본 논문에서는 MSP 함수를 이용 하여 선형 신경 회로망을 합성하는 새로운 방법에 대해 제안 했으며, 제안된 방법이 다른 기존에 제안된 방법에 비해 더 작은 회로망을 구현 할 수 있다는 실험 결과를 보였다. 현재 multiple prime implicant method를 이용 하여 네트워크를 더 간략화 하는 연구가 진행 중이며, 영상 등의 데이터를 변환 하기 위한 몇 가지의 quantization 방법도 연구 중이다. 위의 새로운 시도를 통해 multiple output문제에 대해 하나의 minterm을 두개 이상의 출력에 공유 되게 할 수 있기 때문에, 더욱 간단한 형태의 선형 신경회로망을 구성할 수 있을 것으로 예상된다. 그리고 제안된 알고리즘이 제공하는 네트워크 변수들은 VLSI 하드웨어 구현에서도 유리할 것으로 보인다.

IV. 참고문헌

- [1] David E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, Reading, MA, 1988.
- [2] Edward J. McCluskey, *Logic Design Principles: with Emphasis on Testable Semicustom Circuits*, Prentice-Hall, NJ, 1986.
- [3] Zvi Kohavi, *Switching and Finite Automata Theory*, McGraw-Hill, 1970.
- [4] Julius T. Tou and Rafael C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, MA, 1974.
- [5] Donald L. Gray and Anthony N. Michel, "A training algorithm for Binary Feedforward Neural Networks," *IEEE Trans. Neural Networks*, vol. 3, no. 2, March 1992.
- [6] Jung H. Kim and Sung-Kwon Park, "The geometrical learning of binary neural networks," *IEEE Trans. Neural Networks*, vol. 6, no. 1, January 1995.
- [7] Thang Nguyen Bui and Byung Ro Moon, "Genetic algorithm and graph partitioning," *IEEE Trans. Computers*, vol. 45, no. 7, July 1996.