BP학습알고리즘을 이용한 다치신경회로망의 구성과 해석

Structure and Analysis of Multi-Valued Neural Networks Based on Back Propagation Learning Algorithm

박 미 경, 정 환 묵
대구효성가톨릭대학교 전자정보공학부

Park, Mee Kyeong and Chung, Hwan Mook
Faculty of Electronics & Information Engineering,
Catholic University of Taeguhyosung

요 약

최근 인공지능연구에서는 기호주의와 커넥션니즘이 독립적으로 연구되어 왔으나 차츰 융합의 필요성이 절실히 요구되고 있다. 본 연구에서는 먼저 기호주의의 일부분인 고전논리를 확장한 다치논리와 커넥션니즘의 기본부분인 신경회로망을 융합한 다치신경망을 구성하고, BP에 기반을 둔 학습 MVL네트워크를 이용하여 해석한다.

본 논문에서는 이러한 구성 및 해석 방법을 확장하여 비고전적인 다치신경회로망을 구성하는 방법을 제안한다.

1. 서론

인공지능,인지과학 등에 있어서 몇가지의 패러다임이 존재하지만 크게는 기호주의와 커넥션니즘(connectionism)의 두가지 흐름이라고할 수 있다. 기호주의는 기호의 형식적조작 즉, 계산에 의해 지능을 실현하려고하는 주의이다. 상식추론을 고전논리를 확장한것에 의해 실현하도록 하는 모든 논리주의가 그 표본이다. 기호주의에 대해서는 인공지능, 인지과학, 철학 분야등으로부터 여러가지 비판이 되고 있다. 이에 대하여 케넥션니즘은 주로 신경회로망을 모델로한 네트워크를 사용하여 각종 계산을 실행하는것에 의해 지능을 실현하도록 하는 주의이다. 이런 기호주의와 커넥션니즘의 통합 필요성은 많은 연구자에 의해 지지적되어 왔고, 또한 몇가지의 시도도 있었다.

최근 인공지능연구에서는 기호주의와 커넥션니즘이 독립적으로 연구되어 왔으나 차츰 융합의 필요성이 절실히 요구되고 있다. 본 연구에서는 먼저 기호주의의 일부분인 고전논리를 확장한 다치논리와 커넥션니즘의 기본부분인 신경회로망을 융합한 다치신경망을 구성하고, BP에 기반을 둔 학습 MVL네트워크를 이용하여 해석한다.

MVL네트워크는 이미지 처리와 음성인식 기법에서 나타나고 있다. 그러나, 이러한 네트워크들는 그들의 환경에서 점차적인 변화에 적응할 수 없다. 이런 기법들에 학습하는 능력을 추가함으로써 환경의 변화에 적응할 수 있고, 그들 능력에서 학습하는데 까지 그들 능력을 얻을 수 있으며, 다른 한편으로 그들의 능력으로 수정 및 확장될 수 있는 그런 시스템을 설계한다. MVL네트워크의 설계는 입력에서 출력으로의 변환이 입출력쌍의 집합으로 주어지고 특히 나타난 입력에 반응하도록 네트워크를 강요하도록 하기 위해 학습관계를 바탕으로 할 수 있다. 학습은 또한 입출력에 관한 정보가 미리 알려져 있지 않거나 불완전할 때 필요하다. 학습 능력에 관한 몇가지 MVL 네트워크는[9-10]에 제안 되어 있다. 전통적인 신경망과 같이 이러한 학습 MVL 네트워크들은 은닉층(hidden layer)과 유니트(unit)들의 수를 임의로 선택하도록 요구한다. 학습은 네트워크에 관한 사전지식없이 블랙박스(black box)안에서 실행하였다.

본 논문에서는 BP알고리즘을 기반으로 학습 MVL네트워크를 논의한다. 즉, 표준 MVL함수들을 기반으로 전방향 MVL네트워크를 구성하고, 방법은 신경망의 BP와 유사한 방법으로 학습을 수행 안내한다. MVL네트워크의 학습능력은 시뮬레이

선을 통하여 확실히 하도록 한다

2. 다치논리함수의 변화 및 성질

2.1 기본 정의

본 논문에서는 먼저, 법-M(Modulo-M)에 관한 기본 사항을 다음과 같이 정의한다.[5]

(1) $A \oplus B = \overline{A} \oplus \overline{B}$

(2) $A \ominus B = \overline{A} \oplus (-1)\overline{B}$

(3) $A + B = \max(A, B)$

(4) $A \cdot B = \min(A, B)$

(5) $A' = P \ominus A \quad (P = M \ominus 1)$

(6) $\overset{\alpha\beta}{X} = \begin{cases} P & (\alpha \le X \le \beta) \\ 0 & (\alpha) X \text{ or } X (\beta) \end{cases}$

2.2 다치 논리 함수의 구성과 기본정리

먼저 다치논리함수의 구조적인 개념 및 함수의 표준형과 최소화방법을 설명하기로한다.

임의의 양의 정수 m에 대하여

$Z_m = \{0, 1, \cdots, m-1\}$

을 법-m(Modulo-M)에 관한 정수환(the ring of integers modulo m)이라 하고,

$P = m - 1$

이라 놓자. 이때 함수

$f : Z_m^n \to Z_m$

$(x_1, \cdots, x-n) \longmapsto f(x_1, \cdots, x_n)$

를 n변수 m치 다치논리함수라 한다.

이를 진리표로 표시하면 <표1>과 같은 형태가 된다.

<표 1> n변수 m치 다치논리함수의 진리표

	x_1	x_2	...	x_n	f
$(p+1)^n$	0		...	0	k_1
	0		...	1	k_2
	0		...	2	k_3
			...		
	p	p	...	p	$k_{(p+1)^n}$

$(Z_m)^n$ 에서 하나의 교차점 $x(\alpha_1, \alpha_2, \cdots, \alpha_n)$이 Z_m의 k_α값에 사상되었다고 할 때 f_{k_α}는

$$f_{k_\alpha} = k_\alpha \left(\overset{\alpha_1\alpha_1}{X_1} \cdot \overset{\alpha_2\alpha_2}{X_2} \cdots \overset{\alpha_n\alpha_n}{X_n} \right)$$

$$= k_\alpha \cdot \prod_{i=1}^{n} \overset{\alpha_i\alpha_i}{X_i}$$

단, $1 \le k_\alpha \le p$ 로 표현할 수 있다.

몇 개의 정점이 임의의 값에 사상된 다치논리함수는

$$f = \Sigma f_1 + \Sigma f_2 + \cdots + \Sigma f_p = \sum_{i=1}^{p} \Sigma f_i$$

와 같이 다항함수로 표시된다.

[기본정리][7]

$A = A + AB$
$A = PA \quad (P = M \ominus 1)$
$A(B+C) = AB + AC$
$(A+B)' = A' \cdot B'$
$(A \cdot B)' = A' + B'$

[정의 1] 다치 논리 함수 $f(x)$에 대한 차분을 다음과 같이 정의한다.

$$df(x) = \frac{\Delta f(x)}{\Delta x} \cdot \Delta x = f(\overline{x}) \ominus f(x) \quad (\overline{x} = x \oplus 1)$$

[정의 2] 다치 논리 함수의 한 다치 변수 x_i에 대하여 $(x_i \oplus \alpha) \ominus x_i = \alpha$를 다치 변수 x_i의 증분이라 한다.

[정의 3] 다치 논리함수 f의 편차분 $d_{x_i(\alpha)}f$를 다음과 같이 정의한다.

$$d_{x_i(\alpha)}f = f(x_1, \cdots, x_i \oplus \alpha, \cdots, x_n) \ominus f(x_1, \cdots, x_i, \cdots, x_n)$$

[정의 4]

$$d_{x_i(\alpha,\beta)}f = d_{x_i(\beta \ominus \alpha)}f \mid_{x_i = \alpha}$$

[정의 5]

$$d_{x_i(\alpha,\beta)}f = f(x_1, \cdots, \beta, \cdots, x_n) \ominus f(x_1, \cdots, \alpha, \cdots, x_n)$$

$$= \begin{cases} \overset{00}{x_i}(\alpha \oplus \beta \ominus \alpha) f(x_1, \cdots, 0, \cdots, x_n) \\ \oplus \overset{11}{x_i}(\beta) f(x_1, \cdots, 1, \cdots, x_n) \\ \vdots \\ \oplus \overset{pp}{x_i}(\beta) f(x_1, \cdots, p, \cdots, x_n) \end{cases}$$

$$\ominus \begin{cases} \overset{00}{x_i}(\alpha) f(x_1, \cdots, 0, \cdots, x_n) \\ \oplus \overset{11}{x_i}(\alpha) f(x_1, \cdots, 1, \cdots, x_n) \\ \vdots \\ \oplus \overset{pp}{x_i}(\alpha) f(x_1, \cdots, p, \cdots, x_n) \end{cases}$$

$$= \overset{\beta\beta}{x_i}(\beta) f(x_1, \cdots, \beta, \cdots, x_n) \ominus \overset{\alpha\alpha}{x_i}(\alpha) f(X_1, \cdots, \alpha, \cdots, x_n)$$

$$= p \cdot f(x_1, \cdots, \beta, \cdots, x_n) \ominus p \cdot f(x_1, \cdots, \alpha, \cdots, x_n)$$

$$= f(\beta) - f(\alpha)$$

3. MVL 네트워크

MVL함수는 기본곱의 합(sum-of product)형태로 나타낼 수 있다는 것은 잘 알려져 있다.

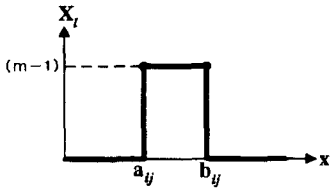
$$f(x_1, x_2, \cdots, x_n) = \sum f(e_1, e_2, \cdots, e_n) \cdot x_1^{e_1} \cdot x_2^{e_2} \cdot \cdots \cdot x_n^{e_n} \quad (1)$$

여기서, $x_i^{e_i} = \begin{cases} 1 & \text{if } x_i = e_i \\ 0 & \text{기타} \end{cases}$ 이다. 그리고 $x_1, x_2, \cdots, x_n$은 n개의 다치변수들이고, $e_i \in 0, 1, 2, \cdots, (m-1), i = 1, 2, \cdots, n$ 이다.

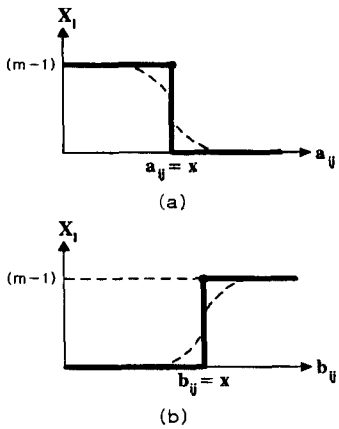
입력층에 있는 각 노드 1은 리터럴 함수이고, 그것의 노드 함수(node function)는 다음과 같다.

$$X_l = {}^{a_{ij}\ b_{ij}}_{x_l} = \begin{cases} m-1 & a_{ij} \le x \le b_{ij} \\ 0 & \text{기타} \end{cases} \quad (2)$$

여기서, $\{a_{ij}, b_{ij}\}$은 창 파라미터 집합(window parameter set)이고, $a_{ij} \le b_{ij}$ $(i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, m,\ l = 1, 2, \cdots, m)$ 이다. 노드 함수에 관한 리터럴 함수가 [그림 1]에 나타나 있다.



[그림 1] 리터럴 함수의 정의



[그림 2] a_{ij}(a) 와 b_{ij}(b)에 관한 리터럴함수의 변환 특성.

aij와 bij의 값이 변함에 따라서 리터럴 함수도 그에 따라서 변경되므로 리터럴 함수의 형태도 여러 가지로 나타난다. m치 n변수의 경우에 있어서 최대로 $m \times n$ 리터럴 노드(literal node)들은 각 입력벡터가 개별적으로 선택될 수 있도록 하기 위해 요구되었다. 최급하강에 의해 학습하1.도록 하기 위해서 함수의 미분이 필요하다. 그러나 [그림 2]에 나타나 있는것 처럼 리터럴 함수은 원도우 파라미터 aij와 bij에 대해 계단형(hard limiting) 의 비선형함수를 보이고 있다. 리터럴 함수의 미분은 두 점에서는 존재하지 않는다. 따라서 시그모이드 함수를 적용할 경우 결정영역이 통상 직선이 아닌 완만한 곡선으로 경계지워지므로 약간 복잡하지만 미분가능하며 은닉층을 학습할 수 있는 BP알고리즘을 수정할 수 있다. 그러므로, 신경망에 이용된 계단형의 비선형 함수는 시그모이드 함수(sigmoid function)라 불리는 소프트 리미팅 함수(soft limiting function) $f(h)$ 로 대치될 수 있다.

$$f(h) = \frac{1}{1 + \exp^{(-\lambda h)}} \quad (3)$$

여기서, $\lambda \to \infty$이다. 따라서 리터럴 함수은 다음과 같이 기술될 수 있다.

$$X_l = (m-1)\{1 - f(a_{ij} - x_i)\} = (m-1)\frac{1}{1 + \exp\{\lambda(a_{ij} - x_i)\}} \quad (4)$$

그리고

$$X_l = (m-1)f(b_{ij} - x_i) = (m-1)\frac{1}{1 + \exp\{-\lambda(b_{ij} - x_i)\}} \quad (5)$$

분명히, 이러한 함수들은 미분가능하며, 0 또는 (m-1) 양 끝점에 수렴한다. 이러한 함수들의 미분은 다음과 같이 그들 함수의 항(term)들로 표현될 수 있다.

$$\frac{\partial X_l}{\partial a_{ij}} = -(m-1)\lambda f(a_{ij} - x_i)\{1 - f(a_{ij} - x_i)\} \quad (6)$$

와

$$\frac{\partial X_l}{\partial b_{ij}} = (m-1)\lambda f(b_{ij} - x_i)\{1 - f(b_{ij} - x_i)\} \quad (7)$$

위의 논의에서 우리는 계단형의 비선형함수가 시그모이드 함수로 계산될 수 있도록 근사하게 된다. 리터럴 함수의 미분에 관한 형식(6)과 (7)은 매우 미숙하다. 우리는 단지 그것의 부호만을 사용하지 그것의 크기를 사용하지는 않는다. 따라서, 미분은 간단히 다음과 같이 정의한다.

$$\frac{\partial X_l}{\partial a_{ij}} = -1 \quad (8)$$

와

$$\frac{\partial X_l}{\partial b_{ij}} = 1 \quad (9)$$

중간층에 있는 노드는 MIN연산에 대응된다. 각 노드는 함수 개개의 영역을 선택하고, MIN항(term)내에 포함된 논리신호 1, 2, ¨,m-1에 의해 그것의 함수값 1, 2, ..., m-1로 정의한다. 그러므로, 그것은 최대 m^n MIN 노드에 대응된다. 함수는 다음과 같다.

$$M_k = M(k_a, x_1', x_2', \cdots, x_n')$$

여기서, $1 \le k_a \le (m-1)$이다.

최급하강에 의해 학습하도록 하기위해 MIN함수의 미분이 필요하다. 그러나, 우극한과 좌극한이 같지 않기 때문에 MIN함수의 미분은 그 점에서 존재하지 않는다. 이를 운영하는데 형식상 근거있는 방법은 단 하나의 점에서 모든 경사도들의 오목한(convex) 조합을 고려하고, 가장 최적의 알고리즘을 얻기 위해 이 집합에서 한 방향을 선택한다. 이 스킴(scheme)의 경험적 접근은 단 하나의 점에 대한 미분을 위하여 두 극한의 평균을 사용하도록 한다. 우리는 단순히 경험적 접근을 선택하

였다.

출력층의 노드는 곱의 항(product term)들간에 MAX 연산자를 나타낸다.

$$O = S(M_1, M_2, \cdots, M_m)$$

위에 기술된 학습 MVL 네트워크는 각 노드가 특별한 파라미터의 집합을 사용하여 그 노드로 들어오는 신호를 특별한 함수(노드 함수)를 이루는 다층 전방향 네트워크(multi-layer feed-forward network)이다. 노드 함수의 형은 층에서 층으로 변환시키며, 각 노드 함수는 네트워크 상에서 사전지식에 의해 정의 되어 질수 있다. 더욱이 변경 가능한 파라미터들은 단지 층 1 상에 나타낸다. 다른 가중치들은 유니트에서 고정된다. 이것은 최급하강이 3개의 층보다는 단지 하나의 층상에서 효과적으로 작업들을 처리한다는 것을 의미한다.

4. 학습 메카니즘(Learning Mechanism)

위에서 언급한 학습 MVL네트워크는 하나의 전방향네트워크이다. Θ는 주어진 다치함수에 의해 생성되는 다치입력 벡터들 $\vec{x}$와 대응하는 교사신호 벡터들 $\vec{T}$에서 추출한 쌍들의 집합이라 하자. 흔히 네트워크를 훈련하는 데 이용된 방법은 파라미터들(a_{ij}와 b_{ij})이 아래 형식의 비용함수를 최소화함으로써 수정된 BP알고리즘이라고 알려진 것이다.

$$E = \sum_{(\vec{x}, \vec{T}) \in \Theta} \left(\vec{O} - \vec{T} \right)^2 \qquad (10)$$

여기서, $\vec{O}$는 입력벡터 $\vec{x}$가 입력에 적용될 때 네트워크의 출력 벡터(출력층안의)이다. BP는 E를 최소화하기 위해 기울기를 따라가는 방법(gradient descent)를 사용한다. 사실 파라미터들(a_{ij}와 b_{ij})은 규칙에 따라 다음과 같이 바뀌어 진다.

$$\Delta a_{ij} = -\eta \frac{\Delta E}{\Delta a_{ij}} \text{ 와 } \Delta b_{ij} = -\eta \frac{\Delta E}{\Delta b_{ij}} \qquad (11)$$

여기서, a_{ij}와 b_{ij}는 리터럴 파리미터(literal parameter)들이고 $\eta(0<\eta<1)$은 학습율(learning rate)이다. 파라미터 Δa_{ij}와 Δb_{ij}에 대해서 [14]에 삽입되어 있는 Δa_{ij}와 Δb_{ij}에 관해서 미분해야 한다. 반복규칙(chain rule)을 사용하여 다음 수식을 얻었다.

$$\Delta a_{ij} = -\eta \frac{\partial E}{\partial a_{ij}}$$

$$= -\eta \sum \frac{\partial E}{\partial \vec{S}} \cdot \frac{\partial \vec{S}}{\partial a_{ij}}$$

$$= \eta \sum (\vec{T} - \vec{S}) \sum_k \frac{\partial \vec{S}}{\partial M_k} \cdot \frac{\partial M_k}{\partial a_{ij}}$$

$$= \eta \sum (\vec{T} - \vec{S}) \sum_k \frac{\partial \vec{S}}{\partial M_k} \cdot \frac{\partial M_k}{\partial a_{ij}} \cdot \frac{\partial X_l}{\partial a_{ij}}$$

$$= -\eta \sum (\vec{T} - \vec{S}) \sum_k \frac{\partial \vec{S}}{\partial M_k} \cdot \frac{\partial M_k}{\partial X_l}$$

.....(12)

와

$$\Delta b_{ij} = -\eta \frac{\partial E}{\partial b_{ij}}$$

$$= -\eta \sum \frac{\partial E}{\partial \vec{S}} \cdot \frac{\partial \vec{S}}{\partial b_{ij}}$$

$$= \eta \sum (\vec{T} - \vec{S}) \sum_k \frac{\partial \vec{S}}{\partial M_k} \cdot \frac{\partial M_k}{\partial b_{ij}}$$

$$= \eta \sum (\vec{T} - \vec{S}) \sum_k \frac{\partial \vec{S}}{\partial M_k} \cdot \frac{\partial M_k}{\partial X_l} \cdot \frac{\partial X_l'}{\partial b_{ij}}$$

$$= \eta \sum (\vec{T} - \vec{S}) \sum_k \frac{\partial \vec{S}}{\partial M_k} \cdot \frac{\partial M_k}{\partial X_l}$$

.....(13)

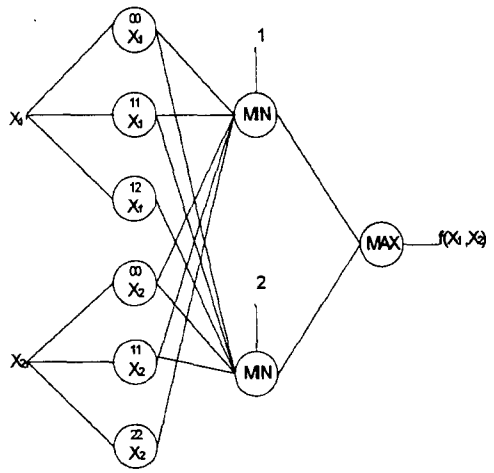
위의 학습방정식과 위에 정의된 미분을 사용하여 우리는 MVL네트워크에서 BP를 수행할 수 있다.

5. 응용 예

예를들어, <표 2>의 Karnaugh도를 만족하는 다치 논리함수를 식으로 표시하면 다음과 같다.

$$f(x_1, x_2) = 1\left(\begin{matrix} 11 & 00 \\ x_1 & x_2 \end{matrix} + \begin{matrix} 00 & 11 \\ x_1 & x_2 \end{matrix} + \begin{matrix} 11 & 22 \\ x_1 & x_2 \end{matrix} \right) + 2\left(\begin{matrix} 00 & 00 \\ x_1 & x_2 \end{matrix} + \begin{matrix} 12 & 11 \\ x_1 & x_2 \end{matrix} \right)$$

.....(14)



[그림 3] 3-치 논리함수에 대한 MVL구조를 바탕으로한 기본형의 실현

<표 2> 3변수 함수 $f(x_1, x_2)$의 예

x_2 \ x_1	0	1	2
0	2	1	0
1	1	2	2
2	0	1	0

6. 결론

본 논문에서는 다치신경망을 구성하고 BP학습 알고리즘을 이용하여 해석하였다. MVL 네트워크들 2.상의 사전 지식은 목표로부터 멀지 않은 파라미터 공간에서 한점으로부터 시작하도록 뉴럴 BP 학습과정을 수행하였다.

따라서 표준 MVL함수를 기반으로 전방향 MVL네트워크의 구성 및 해석 방법을 확장하여 비고전적인 다치 신경회로망을 구성하는 방법의 가능성을 제안하였다.

참고문헌

[1] A. Kandel and S. C. Lee, Fuzzy Switching and Automata : Theory and applications, crane and edward Arnold, 1979.

[2] A. Kandel and J. M. Francioni, "On the Properties and Applications of Fuzzy Valued Switching Functions", IEEE Trans. Computers, Vol. C-29, No. 11, pp. 986~993, Nov. 1980.

[3] A. Thayse and M. Davio, "Boolean Differential Calculus and Its Application to Switching Theory", IEEE Trans. Computers, Vol. C-22, No. 4, pp. 409~419, Apr. 1973.

[4] D. K. Pradhan, "A Theory of Switching Functions", IEEE Trans. Computers, Vol. C-27, pp. 239~248, Mar. 1978.

[5] H.M. Chung, "다치 논리함수의 구조해석과 전개", 정보과학회지, Vol. 13, No. 3, pp. 155~166, Aug. 1986.

[6] J.H. Turker, M.A. Tapia and A.Q. Bennett, "Boolean Differentiation and Integration Using Karnaugh Maps", In Proc. IEEE Southeast Conf., Williamsburg, VA, Apr. 1977.

[7] S. W. Lee and H. M. Chung, "기호 다치 논리함수와 그 변화 및 전개", 대한 전자공학회지, Vol.20, No.5, Sep. 1983.

[8] S. Y. H. Su and A. A. Sarris, "The Relationship Between Multi-valued Switching Algebra and Boolean Algebra under Different Definitions of Complements", IEEE Trans. Computers, Vol. C-21, No.5, pp. 479~485, May 1972.

[9] C. M. Allen and D. P. Givone, "A Minimization Technigue for Multiple Valued Logic Systems", IEEE Trans. on Computers. , Vol. C-17, pp. 182-184, 1968

[10] K. C. Smith, "A Multiple Valued Logic : A Tutorial and Application", Computer, Vol. 21, No. 4, pp.17-27, 1988