정보학연구 제4권 제3호 2001년 9월(pp.155-163)
The Journal of Information Technology, Vol.4, No.3

시그모이드 함수의 디지털 구현에 관한 연구

이호선* · 홍봉화**

요 약

디지털 신경회로망의 구현에 있어 시그모이드 함수의 구현은 매우 복잡하고 구현하기 어렵다. 따라서, 본 논문에서는 디지털 신경회로망 구현에 문제가 되는 시그모이드 함수처리를 위한 설계 방법을 제안 하였다. 제안된 방법은 잉여수계를 이용하여 MAC(Multiplier and Accumulator) 연산 시, 캐리 전파 없 이 고속의 연산 을 수행할 수 있고 시그모이드 함수처리를 고속으로 수행할 수 있다.
모의실험결과, 각각의 신경 프로세스에 있어서 4.6nsec 이상의 속도를 보임으로써 고속디지털 신경회 로망 구현에 적용될 수 있을 것으로 기대된다.

Ⅰ. 서론

신경망 이론은 1940년대에 이미 제안되었지만 반도체 기술이 발전한 근래에 이르러 많은 발전을 하게 되었다. 현재까지 연구되고 있는 신경망의 응용분야는 인식(패턴 인식, 음성인식), 제어이론, 영상처리 분야 등이 있다.[1]-[5]

최근 영상신호처리 및 패턴인식 분야에서 대량의 데이터를 실시간으로 처리 하여야하는 필요성이 증가하고 있다. 컴퓨터와 사용자간의 인터페이스 문제에 있어서 실시간 처리는 중요한 해결 수단이다. 이와 같은 응용분야에 신경망을 이용하기 위해서는 대량의 데이터를 실시간으로 처리할 수 있는 고속의 MAC연산기와 판별(시그모이드)함수를 고속으로 처리할 수 있는 연산기가 요구된다.

본 논문에서는 신경망을 디지털 회로로 구현 시 문제가 되는 시그모이드 함수를 잉여수계(Residue Number System)를 이용하여 구현 하고자 한다.

뉴런프로세서 구현 시 문제가 되는 시그모이드(Sigmoid)함수 처리는 잉여수계를 이용한 ROM 테이블 (ROM Look-Up Table) 방식을 사용함으로써 효율적으로 수행할 수 있다.(10)(11)

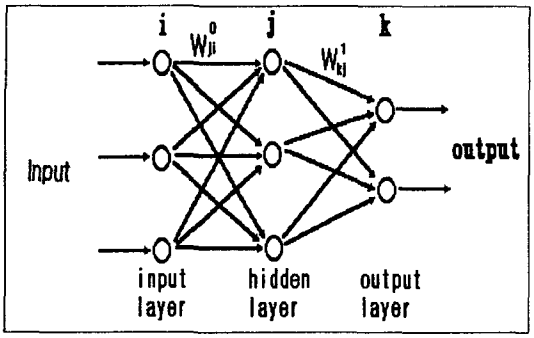
신경망의 기능을 디지털 방식을 이용하여 구현하는 것은 아날로그 방식 및 광을 이용한 신경회로망이 아직 연구단계에 있는 것에 비하여 현재의 발전된 디지털 VLSI 설계 기술을 이용하여 실제로 이용 가능한 수준의 신경 칩을 제작할 수 있기 때문에 중요하다.

Ⅱ. 역전파 알고리즘

역 전파 알고리즘은 각층간의 오차를 역전파로 학습하는 모델이며 그림 1과 같은 구조를 갖

* 주) 두일 전자통신
** 세명대학교 컴퓨터수리정보학과 교수

는다.



(그림 1) 역 전파 신경회로망

Fig. 1. Back propagation neural network

그림 1에서 j층의 한 노드에 대한 입력(net_j)과 활성함수를 통과한 후의 출력은 다음식과 같다.

$$net_j = \sum W_{ji} O_i = U_j \quad (1)$$

$$O_j = F(U_j) = Oj = \frac{1}{1+\exp(-(netj+\theta j)/\theta o)} \quad (2)$$

(θ_j : 문턱 값, θ_o : 기울기)

시그모이드 활성함수는 출력 값을 0과 1사이의 값으로 제한하며, 다음 단 노드의 입력 값이 된다. 단일노드의 연산은 층의 순서에 의해 식 (1)과 (2)의 연산을 반복 수행하며, 출력 층 k층에서의 출력 O_k를 출력한다. 여기서 목표 값을 T_k라 하고, 실제 출력 값을 O_k라 하면, 오차 값 E는 식 (3)의 최소자승오차(*LMS: least mean* square) 식으로써 구해질 수 있다.[1]-[5]

$$E_p = \frac{1}{2}\Sigma(T_k - O_k)^2 \quad (3)$$

실제출력과 훈련(training)패턴간의 학습과정은 가중치의 변경에 의해 처리된다. 가중치 변경은 식 (3)에 의해 얻어진 오차 E에 의해서 조절된다. 그림 1에서 j층의 어느 한 노드의 p번째 패턴의 오차 값 δ_{pj}는 식 (4)와 같다.

$$\delta_{pj} = O_{pj}(1 - O_{pj})\sigma_{pj} \quad (4)$$

여기서 상위 층인 k층의 오차 값에 의하여 σ_{pj}는 식 (5)처럼 표현된다.

$$\sigma_{pj} = \sum \delta_{pk} W_{kj} \quad (5)$$

출력 층 k에서 p번째 패턴 오차 σ_{pk}는 식(6)에 의해서 구할 수 있다.

$$\sigma_{pk} = T_{pk} - O_{pk} \quad (6)$$

i 층과 j층 사이의 t+1단계 가중치 변화량 $\Delta W_{ji}(t+1)$은 식 (7)로 구할 수 있다.

$$\Delta W_{ji}(t+1) = \eta(\delta_j O_i) + \alpha \Delta W_{ji}(t) = \eta(\delta_j O_i) + \alpha \Delta W_{ji}(t) \quad (7)$$

여기에서 η와 α는 각각 학습계수와 관성계수로서 1회의 가중치 갱신에 따른 비례 값이다. 각 층에서 계산된 오차는 역방향으로 가중치를 순차적으로 갱신한다.

Ⅲ. 잉여수계를 이용한 시그모이드 함수의 설계

3.1 잉여수계의 특징

일반적인 2진 정수계에 의한 연산기는 캐리정보로 인하여 가산 및 승산기 설계 시 문제가 된다. 특히, 대량의 입력 데이터를 처리하는 영상신호처리, 패턴인식 분야의 경우, 연산기의 크기 및 처리속도 향상에 많은 어려움이 있다. 잉여수계는 가중치가 없는 수체계이며, 각 모듈 간에 독립성을 가지므로 캐 리 정보가 필요 없고 승산과 가산이 거의 동일한 시간에 이루어질 수 있다는 큰 장점을 갖기 때문에 대량의 데이터를 처리하는 고속의 병렬처리 연산에 유용하다.[6][7][8]

3.2. 잉여수계의 기본연산

정수의 잉여수계 표현 방법은 다음과 같다. 서로소인 모듈리 P를 선택하고 P = { m_1, m_2, m_3 }일 때 정수 X의 표시 가능한 범위는 다음과 같다.

$0 \le X \le M(M = \prod_{i=1}^{N}$) 이다.

잉여수계 정수 X는 n 테이블로 표시할 때

$X \xrightarrow{RNS} (X_1, X_2, X_3, \ldots, X_n)$로 된다.

(단, $X_i = X \bmod m_i = |X|_{m_i}$)

예를 들면, P={2, 3, 5}일 경우, 정수의 잉여수 표현 예를 표1에 나타내었다. 정수 X = 25는 $|X|_{m_1} = 1$, $|X|_{m_2} = 1$, $|X|_{m_3} = 0$ 즉 (1, 1, 0)로 표현된다.

기본 연산은 다음과 같다.

Z = X ○ Y ('○' 연산자: * + -)가 성립한다.

즉 $|Z|_{m_i} = |X|_{m_i} \circ |Y|_{m_i} = |X \circ Y|_{m_i}$

예를 들면 X = 12, Y = 5의 가산의 경우

P = (2, 3, 5)일 때

	mod 2	mod 3	mod 5	
	0	0	2	; x = 12
+	1	2	0	; y = 5
	1	2	2	; Z = 17

연산 결과는 표 1을 참조하면 17임을 알 수 있다.

〈표 1〉 모듈리{2, 3, 5}의 경우 잉여 수 표현 예
Table 1. Example of residue number presentation of moduli {2,3,5}

정수 X	잉여수 표현 m1 m2 m3	정수 X	잉여수 표현 m1 m2 m3
0	0 0 0	16	0 1 1
1	1 1 1	17	1 2 2
2	0 2 2	18	0 0 3
3	1 0 3	19	1 1 4
4	0 1 4	20	0 2 0
5	1 2 0	21	1 0 1
6	0 0 1	22	0 1 2
7	1 1 2	23	1 2 3
8	0 2 3	24	0 0 4
9	1 0 4	25	1 1 0
10	0 1 0	26	0 2 1
11	1 2 1	27	1 0 2
12	0 0 2	28	0 1 3
13	1 1 3	29	1 2 4
14	0 2 4	30	0 0 0
15	1 0 0		

3.3. 혼합 기수변환(MRC: Mixed Radix Conversion)

임의의 정수 X를 혼합기수[6][7][8] 형식으로 표현하면 식 (8)과 같다.

$$X=a_N\prod_{i=1}^{N-1}R_i+\cdots+a_3R_1R_2+a_2R_1+a_1 \quad (8)$$

R_i : radides

a_i : mixed-radix digit 또는 mixed-radix 계수

$0 \le a_i < R_i$

$X = \langle a_n, a_{n-1}, \ldots, a_1 \rangle$

혼합기수로 변환시 혼합기수의 계수를 구하는 과정은 아래와 같이 구할 수 있다.

i) 식 (8)의 양변에 모듈로 m_1을 취하면 마지막 항을 제외하고는 모두 m_1의 곱으로 되어 있으므로,

$$|X|_{m_1} = a_1$$

즉, a_1은 첫 번째 residue digit인 r_1이 된다.

ii) 식 (8)의 양변에 a_1을 빼주고 모듈로 m_2를 취하면 다음과 같다.

$$|X-a_1|_{m2}=\left|a_N\prod_{i=1}^{N-1}m_i+\cdots+a_3m_1m_2+a_2m_1\right|_{m2}$$

$$=|a_2m_1|_{m2}$$

$$a_2=\left|\frac{x-a_1}{m_1}\right|_{m2}$$

$$=\left|\left[\frac{X}{m_1}\right]\right|_{m2}$$

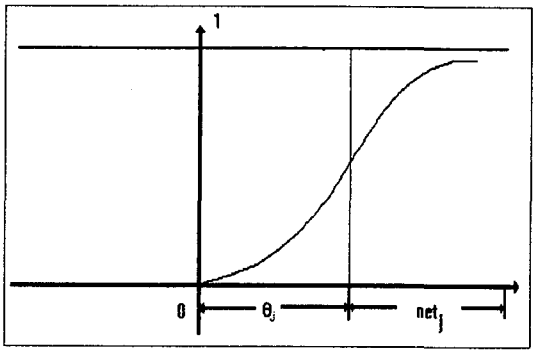
$$a_i=\left|\left[\frac{x}{m_1m_2\cdots m_{i-1}}\right]\right|_{m_i}$$

iii) i), ii)의 방식으로 나머지 혼합기수 계수를 구하게 되며, $i > 1$ 조건에서 계수 a_i는 아래식과 같이 표현된다.

Ⅳ. 잉여수계를 이용한 시그모이드 함수의 구현

잉여수계 연산은 입력된 수를 선택된 모듈리에 따라 분할하고 각 모듈리 별로 연산하므로, 연산기의 크기가 감소되며 모듈리 간에 캐리정보가 필요 없으므로 고속의 연산을 수행할 수 있다.

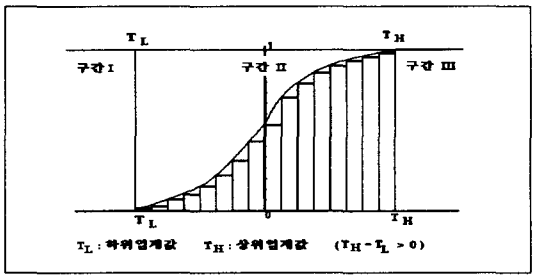
시그모이드 함수처리는 Ⅱ절의 식(2)와 같이 표현되며 이 부분은 신경회로망의 디지털회로 구현 시, 문제가 되는 부분이다. 본 논문에서는 이 부분을 잉여수계의 MRC(Mixed Radix Conversion)을 사용하여 구간을 분할하여 처리하고자 한 다. 식(2) 에서 θ_j,θ_0에 따라 시그모이드 함수는 그림 2와 같은 특성을 갖는다.



(그림 2) θj 와 θ0에 의한 시그모이드 함수의 이동과 기울기

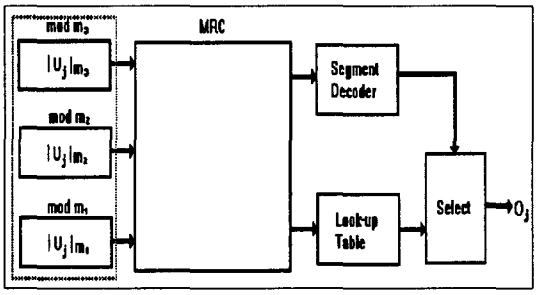
Fig. 2. Shifting and Slope of sigmoid function by θj and θ0

그림 2를 그림 3과 같이 3구간으로 분할하여 구간 분할시 MRC를 사용하며, 시그모이드 함수 처리부분을 그림 4와 같이 구성하였다.



(그림 3) 시그모이드 함수의 구간설정

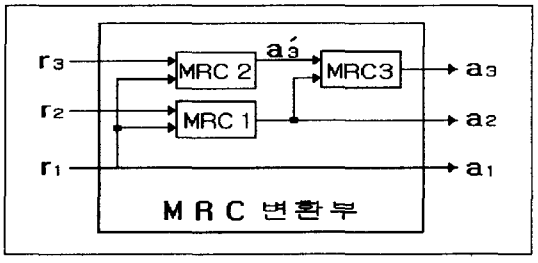
Fig. 3. Decision of boundary in sigmoid function



(그림 4) 시그모이드 함수의 연산처리과정

Fig. 4. Processing of sigmoid function

그림 4에서 MRC 처리부분을 나타내면 그림 5와 같으며, 내부연산과정은 식(10)과 같다.



(그림 5) MRC 처리부분

Fig. 5. Processing unit of Mixed radix conversion(MRC)

$$x - a_1|_{m_2} - a_{n-1}\prod_{i=1}^{n-1} m_i + \cdots + a_3 m_1 m_2 + a_2 m_1 \Big|_{m_3}$$

$$- |a_3 m_1 m_2|_{m_4}$$

$$a_3 = \left|\frac{x - a_1}{m_1}\right|_{m_2}$$

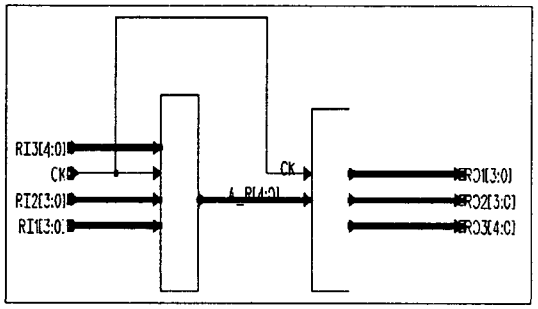
$$- \left|\left|\frac{x}{m_1}\right|_{m_2}\right| \qquad (10)$$

그림 5에서 잉여수계 입력(r_1,r_2,r_3)를 받아서 MRC(혼합기수 변환)를 거쳐 가중 치 체계의 수로 변형된 출력(a_1,a_2,a_3)은 연산 표(Look up table)에 저장된 잉여 수를 찾아 출력한다. 가중치 체계로 변환된 출력 a_3는 연산표의 입력 값으로 구간 설정에 쓰이며, a_1, a_2를 이용하여 그림 3에 있는 구간 II에 해당되는 값을 사전에 저장된 연산표의 값으로 출력한다.

V. 모의실험 및 고찰

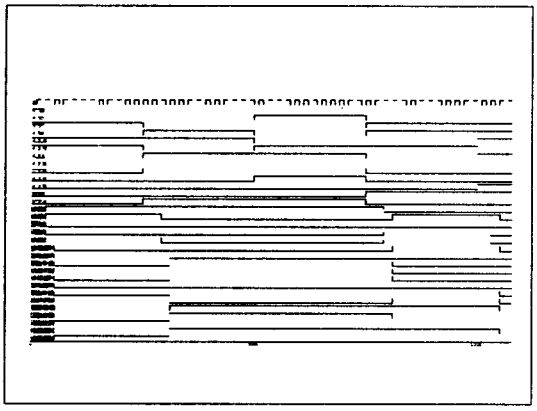
본 논문에서 설계한 고속 디지털 뉴런프로세서는 C 언어 및 COMPASS tool과 스파이스 3(SPICE3)를 이용하여 논리적인 검증을 수행하였다. 모의실험의 실행조건으로 입력 값과 연결강도를 각각 0 ~ 10, -32 ~ 32의 정수 값으로 하였으며, 그림 3의 시그모이드 함수처리를 위하여 구간 II를 9,11,15등분하고 잉여수의 모듈리는 11,13,17로 하여 실험하였다.

그림 6은 모듈리를 11, 13, 17로 할 경우 모듈리 17를 이용하여 시그모이드 함수 전체 구간을 17개로 분할하고 구간 I 를 3개 (a_3 = 0, 1, 2), 구간 III을 3개(a_3 = 14,15,16)로 0과 1을 출력하고, 구간 II를 11개로 정하여 시그모이드 함수 값을 출력하는 시그모이드 함수처리부의 VHDL 합성 결과를 타낸다.



(그림 6) VHDL로 합성한 시그모이드 함수 처리부의 논리회로

Fig. 6. Logic circuit of sigmoid function for synthesis of VHDL



(그림 7) VHDL로 합성한 시그모이드 함수처리부의 입·출력 신호

Fig. 7. Input-output signal of sigmoid function unit for synthesis of VHDL

그림 7은 시그모이드 처리부의 입·출력 신호를 나타나며, 바렐 쉬프터를 이용하여 MAC연산기를 구성할 경우 최대 100Mhz로 동작하는 연산기를 구현 할 수 있다.

실험결과, 설계된 시그모이드 함수 처리부는 약 4.6ns의 연산 속도와 1926여 개의 게이트수를 보였다. 표 2는 실수 연산기를 이용한 뉴런 프로세서와 잉여수(모듈리 11, 13, 17)를 이용하여 뉴런프로세서를 구현할 경우, 하드웨어의 크기 비교를 나타낸다.

〈표 2〉 시그모이드 함수의 크기 비교

Table 2. The comparison of size of sigmoid function

연산기의 종류	표현수의 범위	총 게이트 수
실수 연산기를 이용한 시그모이드 함수	210= 1024	2626
잉여수계를 이용한 시그모이드 함수	11×13×17=2431	1151

표 2에서 알 수 있듯이 수의 범위를 1024로 할 경우, 실수 연산기를 이용한 뉴런프로세서는 총 2626여개의 게이트로 구성되며, 잉여 수(모듈리 11, 13,17)를 이용한 뉴런프로세서는 1151여개의 게이트로 구성된다. 따라서 잉여수계를 이용한 방법이 일반적인 실수 연산기를 이용하여 구현한 뉴런프로세서에 비하여 하드웨어 크기가 1/2이상 감소됨을 고찰하였다. 또한, 승산기, 가산기 및 시그모이드 함수 처리부의 속도를 각각 t_m, t_a, t_s라 할 경우 뉴런프로세서의 속도(t_N)는 $t_N = \max(t_m, t_a, t_s)$가 된다. 따라서, 일반적인 실수 연산기를 이용한 뉴런 프로세서의 속도는 약 50Mhz(승산기가 50Mhz 동작할 경우)로 동작하며 잉여수계를 이용하여 구현한 뉴런프로세서는 약 217 Mhz로 동작한다. 따라서, 본문에서 구현한 뉴런프로세서가 약 4배정도 빠름을 고찰

하였다.

Ⅵ. 결론

영상신호처리 및 패턴인식 분야에서 대량의 데이터를 실시간으로 처리하여야 하는 필요성이 증가하고 있다. 컴퓨터와 사용자간의 인터페이스 문제에 있어서 실시간 처리는 중요한 해결 수단이다. 이와 같은 응용 분야에 신경회로망을 이용하기 위해서는 대량의 데이터를 실시간으로 처리할 수 있는 고속의 MAC 연산기와 시그모이드 함수처리를 위한 연산기가 요구된다.

따라서, 본 논문에서는 신경회로망 구현 시, 문제가 되는 시그모이드 함수의 처리에 있어 혼합 계수변환(MRC: Mixed radix conversion)를 이용하여 활성영역을 그림 2와 같이 세 구간으로 분할하고 제 Ⅱ구간을 최대 모듈리를 이용하여 등분한 표본 값을 연산 표에 저장하여두고 이용함으로 시그모이드 함수를 고속으로 처리할 수 있는 연산기를 구현하였다.

잉여수 연산은 정수연산을 수행하며, 수를 각각의 모듈리로 분리하여 연산함으로 모듈리간에 캐리정보가 필요치 않으므로 연산기 크기가 감소하며, 고속의 연산기 설계가 가능하다. 일반적인 실수 연산기를 이용한 프로세서와 본 논문에서 설계한 프로세서와 비교할 때, MRC를 이용한 방법이 1/2 이상 연산기 크기가 축소하며, 연산속도가 4배 이상 향상됨을 고찰하였다.

앞으로의 연구방향은 연구결과를 실용화하기 위하여, 본 논문에서 구현한 시그모이드 처리기를 이용한 고속의 디지털 신경회로망 구현에 관한 연구가 계속되어야 할 것이다.

참고문헌

[1] D.E Rumelhart, J.L. McClelland, et al., "Parallel Distributed processing" Vol. 1, the MIT Press 1986.

[2] "DARPA Neural Network study", AFCEA International press,1987.

[3] You-Han Pao,"Adaptive Pattern Recognition and Neural Networks",Addison Wesley Publishing Company Inc.1989.

[4] G. A. Carpenter,"Neural Network Models for Pattern Recognition and Associative Memory",Neural Networks ,Vol.2 ,pp. 243-257, 1989.

[5] K.Fukushima "A Neural Network for Visual Pattern Recognition", IEEE Computers, Vol. 21, no. 3 ,pp. 65-75, March 1988.

[6] Nicholas S. Szabo, M.S. & Richard I. Tanaka,Ph.D,"Residue Arithmetic And Its Applications to computer Technology", McGRAWHill Book Company, 1987.

[7] K.H.O, Keefe,"A Note on Fast Base Extension for Residue Number Systems with Three Moduli ", IEEE Transaction on Computers, Vol. C-24, pp. 1132-1133, November 1975.

[8] D. K. Banerji and J. A. Brzozowski "On Translation Algorithm in Residue Number Systems ",IEEE Transaction on Computers, Vol. C-21, pp.1281 -1285, December 1972.

[9] L.E.Arlas and Y.Suzuki, "Digital Systems for Artificial Neural Networks", IEEE

Circuits and Device Magazine,pp. 20-24, July 1990.

[10] 정 윤돈 "디지털 신경회로망의 시그모이드 함수 연산회로 설계에 관한 연구", 경희대학교 대학원 전자공학과 석사학위 논문, 1992.

[11] 조 원경 외 1 " 잉여수계를 이용한 디지털 신경회로의 실현", 전자 공학회 논문집 제30권,B편,2, 1993.

On the Digital Implementation of the Sigmoid function

Ho-Sun, Lee*, Bong-Wha, Hong**

Abstract

In this paper, we implemented sigmoid active function which make it difficult to design of the digital neuron networks. Therefore, we designed of the high speed processing of the sigmoid function in order to digital neural networks. we designed of the MAC(Multiplier and Accumulator) operation unit used residue number system without carry propagation for the high speed operation.

we designed of MAC operation unit and sigmoid processing unit are proved that it could run of the high speed. On the simulation, the faster than 4.6ns on the each order, we expected that it adapted to the implementation of the high speed digital neural network.

* Dewell electric communication Inc.
** Dept. of computer aided mathematical information science, Semyung university