

## 잉여수계를 이용한 디지털 신경망회로의 실현

### (An Implementation of Digital Neural Network Using Systolic Array Processor)

尹賢植\*, 趙源敬\*\*

(Hyun Shik Yoon and Won Kyung Cho)

#### 要約

본 논문에서는 디지털 신경망 실현을 위한 어레이 프로세서를 설계하였다. 역전파 학습알고리즘은 행렬·벡터의 연산으로 나타낼 수 있으며, 이 연산과정은 순환, 반복적으로 이루어지므로 어레이 프로세서 구조로 설계하기에 적합하다. 신경망 연산에 필요한 승산과 시그모이드를 구하는 고속 연산회로를 효과적으로 실현하기 위하여 잉여수계를 이용한 연산회로를 설계하였다. 본 논문에서 제안한 방법은 현재 개발된 VLSI기술로 실현 가능하기 때문에 실제 신경망의 응용분야에 이용될 수 있을 것으로 기대된다.

#### Abstract

In this paper, we will present an array processor for implementation of digital neural networks. Back-propagation model can be formulated as a consecutive matrix-vector multiplication problem with some prespecified thresholding operation. This operation procedure is suited for the design of an array processor, because it can be recursively and repeatedly executed. Systolic array circuit architecture with Residue Number System is suggested to realize the efficient arithmetic circuit for matrix-vector multiplication and compute sigmoid function. The proposed design method would expect to adopt for the application field of neural networks, because it can be realized to currently developed VLSI technology.

#### 1. 서론

인간 두뇌의 처리과정은 다수의 신경세포로 부터

생성되는 자극을 이웃하는 신경세포에 전달하기 위한 시냅스 가중치(Synaptic Weights)의 상호 연결망으로 표현할 수 있다. 이와같은 신경망은 자극을 받아들이는 입력부분과 받아 들인 자극을 다른 신경세포에 전달 할것인지를 판별하는 활성화함수(Activation Function)부분으로 구성되며, 신경세포간의 가중치 강도는 자극의 세기에 따라 다르다. 신경세포를 전기적으로 모델링 했을 경우, 가중치를 조절하는 방법은 학습알고리즘에 따라 그 방법이 다른데, 본 논문에서는 역전파(Back-Propagation) 학습알고리즘을<sup>(1,4)</sup> 이용하였다. 근래에 와서 신경망은 VLSI 기술의 발전에 힘입어 하드웨어 구현이 가능하게 되었으며 주

\*正會員, 國立忠州産業大學 電子計算學科  
(Dept. of Computer Science, Chungju Nat'l Univ. of Polytech.)

\*\*正會員, 慶熙大學 電子工學科  
(Dept. of Elec. Eng., Kyunghee Univ.)

接受日字: 1992年 10月 5日

(※ 본 논문은 1992년도 경희대학의 교내연구비에 의하여 연구되었음)

로 패턴 및 음성인식, 제어이론, 영상처리등의 분야에 응용하고 있다. 그러나, 아날로그방법에 의한 대규모 신경망의 구현은 VLSI 설계시, 소비전력등 몇 가지 문제점들이 대두된다. 따라서, 현재의 반도체 기술로 실용성있는 신경망을 구현하기 위해서는 디지털 VLSI기술을 이용한 신경망의 설계에 대한 연구가 필요하다. 신경망 모델은 순환, 반복적으로 행렬·벡터 연산을 수행하기 때문에 어레이 프로세서구조<sup>[5]</sup>로 실현하기에 적합하다. 또한, 기존의 논문에서는 모든 연산과정이 실수연산으로 수행되며, 각 처리요소에 승산기가 필요하다.

본 논문에서는 행렬·벡터 연산과 활성화수를 잉여수계(Residue Number System)<sup>[8,9]</sup>를 이용한 어레이 프로세서로 구현하였다. 잉여수계는 모듈러스(Modulus)간에 캐리(Carry)정보가 필요없고 승산이 가산과 거의 동일한 복잡도와 속도로 실행될 수 있으므로 승산과 가산의 기능을 필요로 하는 어레이 프로세서의 처리요소(PE) 설계에 적합하다.

## II. 행렬과 벡터에 의한 BP 신경망 모델

BP 모델의 전 방향 전달과정을 행렬·벡터연산으로 표현하면 다음과 같다.

$$U^k = t[u_1^k \dots u_n^k] \quad (1)$$

$$A^k = t[a_1^k \dots a_n^k] \quad (2)$$

$$\delta^k = t[\delta_1^k \dots \delta_n^k] \quad (3)$$

$$T = t[T_1 \dots T_n] \quad (4)$$

$$W^{k+1} = \begin{bmatrix} W_{11}^{k+1} & \dots & W_{1n}^{k+1} \\ \vdots & \ddots & \vdots \\ W_{m1}^{k+1} & \dots & W_{mn}^{k+1} \end{bmatrix} \quad (5)$$

여기서  $t$ 는 전치기호이며,  $U^k$ 와  $A^k$ 는 각각 입력과 출력의 열벡터이고,  $\delta^k$ 는  $k$ 층의 입력층으로 역전파된 오차의 열벡터이며,  $T$ 는 출력층 목표값의 열벡터, 그리고  $W^{k+1}$ 은  $k$ 층과  $k-1$ 층사이의 가중치행렬이다. 전방향 전달의 경우, 입력과 가중치의 행렬연산은 식 (6)과 같이 기술할 수 있으며, 식 (6)에서 계산된 값은 시그모이드 함수의 입력으로 주어진다. 이 과정을 나타내면 식 (7)과 같다.

$$U^k = W^{k+1} \cdot A^{k-1} \quad (6)$$

$$A^k = F_s(U^k) \quad (7)$$

여기서  $F_s(U^k)$ 는 식 (8)로 정의된다.

$$F_s(U^k) = \left[ \frac{1}{1 + \exp(-u_1^k)} \dots \frac{1}{1 + \exp(-u_n^k)} \right] \quad (8)$$

역방향 전달의 경우, 목표값과 실제출력값 사이에는 오차가 발생하는데, 이 경우 오차에 관계된 식은 (9), (10)과 같다.

$$\sigma^k = W^{k+1} \cdot \delta^{k+1} \quad (9)$$

$$\delta^k = F_b(\sigma^k) \quad (10)$$

여기서  $F_b(\sigma^k)$ 는, 식 (11)로 정의된다.

$$F_b(\sigma^k) = t [a_1^k(1-a_1^k)\sigma_1^k, \dots, a_n^k(1-a_n^k)\sigma_n^k] \quad (11)$$

또한, 식 (12)는 최종층에서의 오차를 의미하고 식 (13)는 각층에서의 가중치 변경량을 의미한다.

$$\sigma^l = A^l - T \quad (12)$$

$$\Delta W^{k+1}(n+1) = \eta \delta^k \cdot A^{k+1} + \alpha \Delta W^{k+1} \quad (13)$$

( $\eta$ : 학습계수,  $\alpha$ : 관성계수)

## III. RNS를 이용한 BP모델의 어레이 프로세서 설계

### 1. RNS의 기본 이론

디지털 시스템의 산술연산회로는 일반적으로 웨이트드(Weighted) 수체계로서 2진(Binary)수체계를 사용하고 있다. 따라서, 캐리발생으로 인한 연산속도의 지연은 필연적이다. 특히, DSP분야, 영상처리 분야, 그리고 컴퓨터 그래픽 분야등과 같이 승산연산이 많고 실시간 처리가 요구되는 곳에서는 이와같은 연산속도의 지연 문제가 우선적으로 고려 되어야 한다. 잉여수(Residue Number)체계는 언웨이트드(Unweighted)수체계로서 서로소(Relative Prime Number)로 이루어진 모듈리(Moduli)를 이용하여, 각 모듈러스(Modulus)로 나눈 나머지만으로 수를 표현하며, 연산할때 캐리정보를 필요치 않으므로 연산의 고속화가 가능하고, 또한 연산모듈이 서로 독립적으로 동작될 수 있어서 어레이 프로세서(Array Processor)설계에 적합하다. 잉여수계에서 모듈리를  $P$ 라 하면,  $P = \{m_1, m_2, \dots, m_m\}$ 으로 표시하며, 이때 임의의 정수  $X$ 는 (14)식과 같이 표현할 수 있다.

$$X = q_i m_i + x_i \quad (i=1, 2, \dots, n) \quad (14)$$

$$(0 \leq x_i < m_i)$$

여기서  $q_i$ 은 모듈러스  $m_i$ 에 대한  $X$ 의 몫이고 그때의 나머지는  $x_i = X \bmod m_i$  또는  $x_i = |X|_{m_i}$ 로 표현할 수 있고 따라서 각 모듈러스에 대한  $X$ 의 잉여수 표현은 (15)식으로 나타낸다.

$$X \xrightarrow{\text{RNS}} \{x_1, x_2, \dots, x_n\} \quad (15)$$

이때 정수  $X$ 의 범위는,  $0 \leq X < M = \prod_{i=1}^n m_i$ 이다.

잉여수계에서, 정수  $x$ 와  $y$ 의 이항연산(Binary Operation)은 다음과 같이 정의된다.

$$|Z|_{m_i} = z_i = |x_i \cdot y_i|_{m_i} \quad (16)$$

여기서 연산자  $\cdot$ 는  $+$ ,  $-$ , 혹은  $*$  연산을 표시한다.

잉여수 체계는 정수만을 취급하고 수의 범위가 각 모듈러스간의 곱으로 제한되며 그 이상과 이하의 수에 대한 나머지 표현은 동일하게 반복된다.

### 2. 혼합기수 변환 (MRC: Mixed-Radix Conversion)

본 논문에서는 시그모이드 함수의 처리 구간을 결정하는데 혼합기수계<sup>[10]</sup>를 이용하였다. 혼합기수계는 웨이트드 수체계로서 수의 크기 비교가 쉽게 수행된다. 정수  $X$ 의 잉여수 표현을 혼합기수계로 변환하기 위해서 전개하면, (17)식과 같이 표현된다.

$$X = a_n \prod_{i=1}^{n-1} m_i + \dots + a_3 m_1 m_2 + a_2 m_1 + a_1 \quad (17)$$

혼합기수로 변환시 혼합기수의 계수를 구하는 방법은 다음과 같다.

i) 식(17)의 양변에 모듈로  $m_1$ 을 취하면 마지막 항을 제외 하고는 모두  $m_1$ 의 곱으로 되어 있으므로,  $|X|_{m_1} = a_1$  이고 즉, 그 첫 번째 잉여수(residue digit)가 혼합 기수계의 첫 번째 계수  $a_1$ 이다.

ii) 식(17)의 양변에서  $a_1$ 을 빼고  $| \cdot |_{m_1}$ 을 곱한 결과의 모듈로  $m_2$ 를 취하면, (18)식에 의해 계수  $a_2$ 를 구할 수 있다.

$$a_2 = \left\lfloor \frac{X - a_1}{m_1} \right\rfloor_{m_2} = \left\lfloor \left\lfloor \frac{X}{m_1} \right\rfloor \right\rfloor_{m_2} \quad (18)$$

iii) i), ii)의 과정을 반복함으로써 (19)식에 의해 계수  $a_n$ 을 구할 수 있다.

$$a_n = \left\lfloor \left\lfloor \frac{X}{m_1 m_2 \dots m_{i-1}} \right\rfloor \right\rfloor_{m_i} \quad (19)$$

(단,  $i = 3, \dots, n-1$ )

### 3. RNS에 의한 BP 신경망모델의 구성

디지털 신경망을 실현하기 위한 기본 연산회로를 잉여수계를 이용하여 설계한 기본구조는 그림 1과 같다. 그림 1의 회로<sup>[11]</sup>에서 입력데이터를 잉여수계로 변환하는 부분은 입력층에서만 필요하다. 그리고 모든 연산은 잉여수 연산으로 수행되고 각 연산 모듈은 1차원 어레이 프로세서로 설계 할 수 있다.

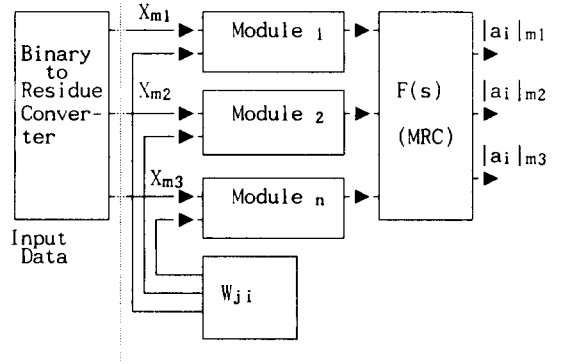


그림 1. 디지털 신경망의 연산구조  
Fig. 1. Block diagram of digital neural nets.

### 4. 3층 BP 신경망회로의 설계

BP 신경망은 입력부분, 가중치연산부분, 그리고 시그모이드 함수 연산부분으로 구성된다. 기본적으로 입력부는 들어오는 입력을 다음층에 전송하기 위해 입력값을 저장하는 버퍼 역할을 하며 가중치 연산부는 들어온 입력과 내부 가중치값에 대한 행렬과 벡터의 연산 결과를 시그모이드 함수 연산부로 전송한다. 시그모이드 함수 연산부는 가중치 연산부에서 전송한 데이터를 처리하는 부분으로 가중치 연산의 내용을 중속변수화하여 현재 계층의 출력을 결정하며 또한,

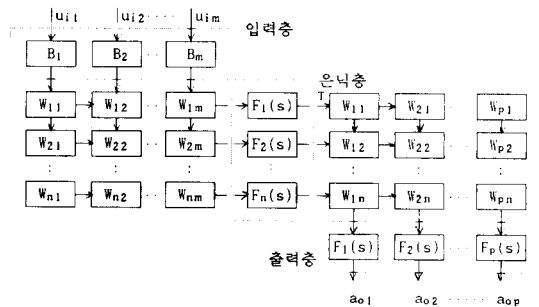


그림 2. 3층 신경망의 2차원 배열구조  
Fig. 2. 2-D array of 3 layers neural nets.

목표값과의 오차를 역전파하여 가중치를 갱신함으로써 수렴속도를 조절하기도 한다. 그림 2의 신경망구조는 그림 3과 같은 기본연산구조의 반복사용으로 실현되기 때문에 본 연구에서는 그림 3의 연산구조를 시스템릭어레이 설계알고리즘<sup>[7], [12]</sup>에 의해 1차원으로 사상(Mapping)하여 그림 4와 같이 설계하고, 이를 사용하여 디지털 신경망회로를 실현한다. 그리고 그 처리요소의 구조는 그림 5와 같다.

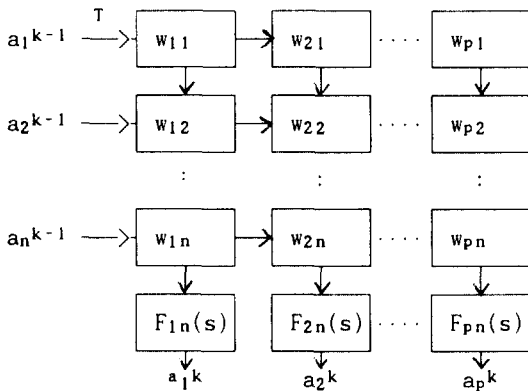


그림 3. 단일층의 프로세서 구조  
Fig. 3. The Structure of one layer neural nets.

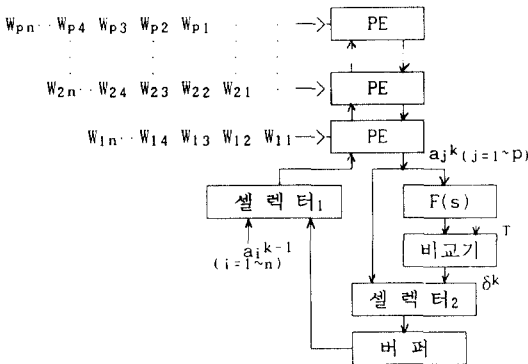


그림 4. BP 신경망의 1차원 시스템릭 어레이 구조  
Fig. 4 The structure of 1-D systolic array for BP model.

그림 4에서 셀렉터1은 초기치입력과 행렬·벡터의 연산결과에 대한 시그모이드함수의 출력값을 선택하는 역할을 하며 행렬·벡터의 연산결과를 따라 시계방향으로 이동한다. 또한 버퍼는 입력 데이터의 분할처리시 중간처리 결과와 최종처리 결과의 저

장에 이용한다.

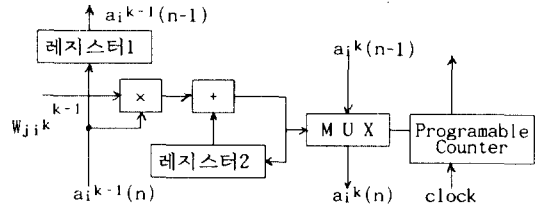


그림 5. PE의 구성  
Fig. 5. Processing element.

그림 5에서 레지스터1과 레지스터2는 각각 입력데이터의 저장과 입력과 가중치의 부분합을 저장하기 위한 용도이며, 프로그래머블 카운터 (Programable Counter)는 처리데이터의 수를 카운트하고 멀티플렉서(MUX)는 현상태의 출력과 이전상태의 출력을 선택한다.

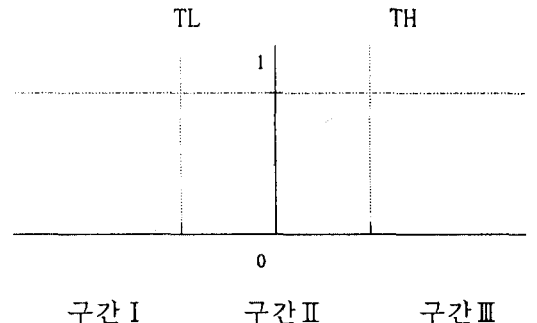
5. 시그모이드 함수의 처리과정

신경망에서 일반적으로 적용되는 활성화함수(Activation function)는 시그모이드(Sigmoid) 함수이며, 무한영역의 입력에 대해 일정범위로 출력을 결정하는 판별함수의 역할을 한다. 여기서, 시그모이드 함수는 0과 1사이의 값을 갖는다. 한편, 입력층을 제외한 k층 j번째 노드에 대한 행렬·벡터 연산은,

$$U_j^k = \sum_{i=1}^n W_{ji}^k a_i^{k-1} \tag{20}$$

이며, 이때 시그모이드의 출력  $a_j^k$ 는 다음과 같이 표시할 수 있다.

$$a_j^k = \frac{1}{1 + e^{-(w_j^k + \theta_j)}} \tag{21}$$



TL : 하위문턱값, TH : 상위문턱값, (TH-TL>0)  
그림 6. 시그모이드 함수의 분할  
Fig. 6. Segmentation of sigmoid function.

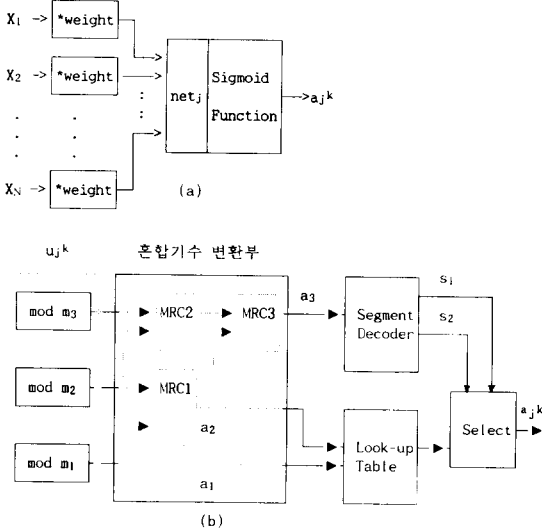


그림 7. (a) 단일 노드의 처리과정,  
 (b) 시그모이드함수 연산부  
 Fig. 7. (a)The processing of single node,  
 (b)Block of sigmoid function.

(21)식을 기존의 디지털 연산방식에 의하여 구하는 것은 대단히 어려운 문제이므로 본 논문에서는 RNS를 이용한 연산방식으로 연산부를 실현하고 시그모이드 함수를 처리하기 위해서 MRC(Mixed Radix Conversion)를 이용하였다. MRC 알고리즘을 이용하여 시그모이드함수를 구하는 과정은 그림 6과 같이 전체구간을 3개의 구간으로 분할하여 구간 I의 출력은 0으로, 구간 III의 출력은 1로 정의하고 구간 II의 영역은 연산표(Look-up table)를 이용하는 방법으로 구한다. 3개의 모듈러스를 사용하는 경우, 시그모이드 함수를 구하는 회로는 그림 7과 같이 구성된다.

6. 대규모 신경망을 위한 분할처리

위에서 설계한 어레이 프로세서의 PE 수는 고정되어 있으나 실제 응용문제에 적용할때에는 입력의 수가 가변적이고 경우에 따라서 많은 입력선을 필요로 한다. 이러한 환경에 대응하여 제안된 디지털 신경망은 프로그램 제어에 의해 동일한 하드웨어로서 다음과 같이 대규모 신경망을 분할처리 할 수 있다.

$$u_i^k = \sum_{j=1}^{B-1} a_i^{k-1} w_j^{k-1} + \sum_{j=2}^{2B-1} a_i^{k-1} w_j^{k+1} + \dots + \sum_{j=n-B}^n a_i^{k-1} w_j^{k+1} \tag{22}$$

위 과정은 PE수가 B개인 경우, n개의 입력 데이터에 대하여 분할처리 하는것을 의미 한다. 본 논문에서는 프로그램 제어에 의해 각층의 유니트수를 변경시키는 유연성을 지니게 하였고 이를 다층신경망에 적용할경우 대규모의 신경망의 구현도 가능하다.

IV. 모의실험 및 고찰

본 논문의 모의실험은 C언어로 작성하여 IBM PC 386에서 수행하였으며, 3층 역전파(Back Propagation)모델을 이용하여 시그모이드 함수 연산을 실수 연산을 이용하는 경우와 혼합기수 변환을 이용하는 경우에 대하여 비교분석하였다. 실험에 쓰인 입력은 0에서 1 사이의 값이며, 층(Layer)의 수는 은닉층을 포함하여 3개, 그리고 출력노드는 1개로 정하였다.

혼합기수 변환에 쓰인 모듈러스를 11, 13, 15으로 하면 연산표의 크기는 2145, 195, 165, 15의 4가지 경우로 구분된다. 또한 시그모이드 함수의 구간을 -1~1에서 -4~4 까지 변화시키면서 연산표의 크기에 따른 학습효율을 고찰하였다. 첫 단계로 시그모이드 함수 구간을 -1~1 사이로 정한 후 연산표의 크기를 4단계로 변화 할 경우는 그림 8 과 같다. 그림 8에 의하면

Comparing Iteration & target error of floating point vs MRC(range -1 ~ 1)

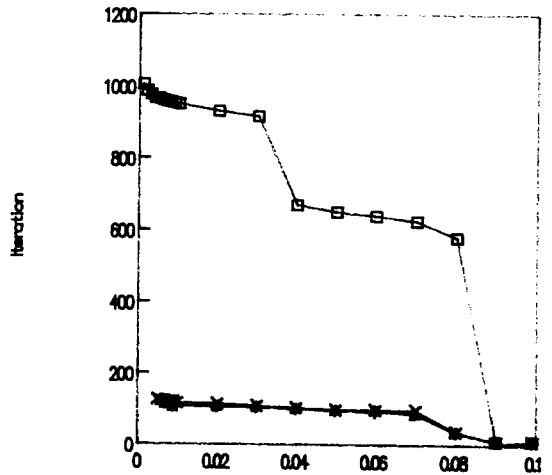


그림 8. 구간 -1~1에서 실수연산과 제안된 알고리즘의 목표값 도달에 대한 속도와 오차비교  
 Fig. 8. Comparing floating point processing with proposed algorithm in boundary -1 ~ 1 .

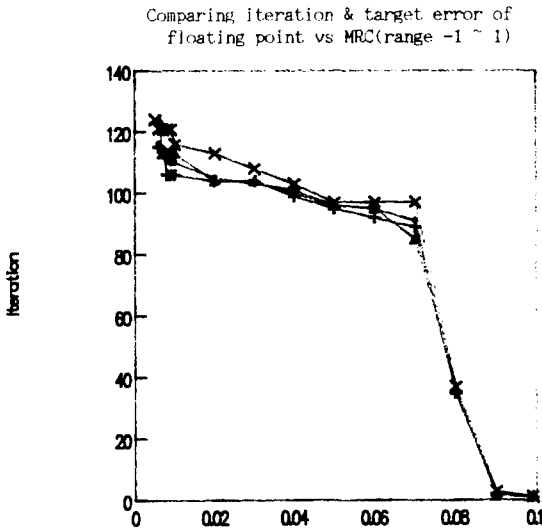


그림 9 구간 -1~1에서 연산표의 크기 선택에 따른 신경망의 수렴속도 비교

Fig. 9. Comparing iteration to approaching target value with 4 look-up table size in boundary -1~1.

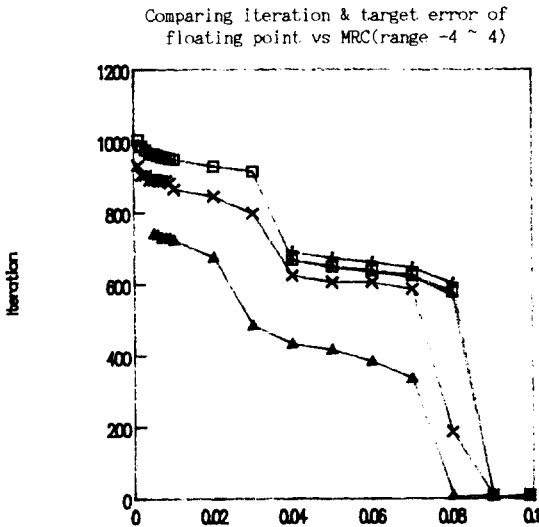


그림 10. 구간 -4 ~ 4 에서 실수연산과 제안된 알고리즘의 목표값 도달에 대한 속도와 오차 비교

Fig. 10. Comparing floating point processing with proposed algorithm in boundary -4~4.

제안된 알고리즘은 목표치와의 오차가 0.005이하인

경우에는 수렴을 하지 않으며 오차가 0.005 이상인 경우에는 시그모이드 함수를 실수 연산 처리하는 경우보다 학습속도가 8.8배 빠르게 수렴함을 보이며, 연산표의 크기가 15개와 2145개를 이용한 경우와 비슷한 결과를 얻었다. 그림 9는 연산표간의 수렴속도를 보인 것이며 여기서 목표값과의 오차가 적고 수렴속도가 빠른것은 연산표의 크기가 165개로 구성된 경우임을 알 수 있다. 또한, 그림 10 은 시그모이드 함수구간을 -4 ~ 4로 하였을 경우를 나타내며, 그림 8 보다는 심하지 않지만 목표 값과의 오차가 0.04인 곳에서 빠르게 수렴한다. 그림 11은 연산표간의 수렴속도를 나타내며 역시 목표값과의 오차가 적고 수렴속도가 빠른 경우는 연산표의 크기가 165개로 구성된 경우임을 나타낸다.

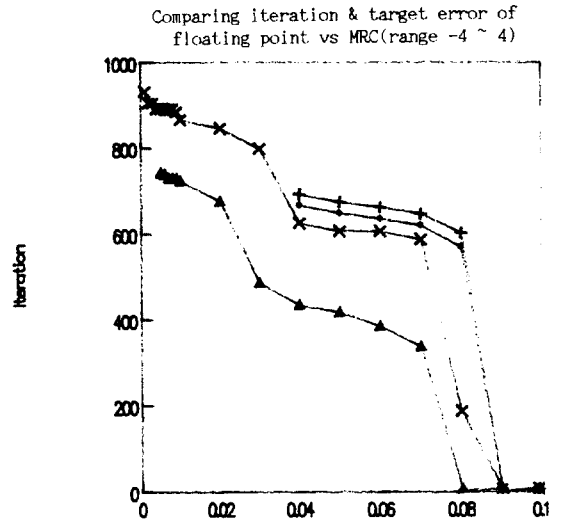


그림 11. 구간 -4 ~ 4 에서 연산표의 크기 선택에 따른 신경망의 수렴속도 비교

Fig. 11. Comparing iteration to approaching target value with 4 look-up table size in boundary -4 ~ 4.

### V. 결론

본 논문에서는 BP알고리즘에 필요한 행렬·벡터연산과 시그모이드 함수 연산을 고속으로 실행 할 수 있는 연산프로세서를 설계하기 위하여 RNS를 적용한 시스템적 어레이설계방법을 이용 하였다. 또한, 본 연구에서는 시그모이드 함수를 3개의 영역으로 분할하여 연산표의 크기를 줄이는 방법을 이용하여 실제적인 시그모이드 함수값을 고속으로 구할 수 있도록 하였다. 실제 설계된 프로세서는 1차원 시스템적

어레이구조로서, 이러한 구조를 재 배열하는 방법으로 입력 데이터의 수가 변하여도 쉽게 적용할 수 있게 설계하였다. 특히 고속처리를 필요로 하는 경우에는 동일구조의 프로세스를 파이프라인 방식으로 배열하여 3층구조의 BP 알고리즘을 고속으로 처리 할 수 있게 설계하였다.

모의 실험결과, 제안된 알고리즘이 동일한 조건하에서, 실수연산을 이용하는 경우보다 우수한것을 알 수 있었으나 목표값과의 오차가  $10^3$ 이하로 수렴하기 위해서는 연산표의 크기를 크게잡아야 하기때문에 신경망의 응용분야가 얼마만큼의 정확도를 요구하는가에 따라서 모듈의 선택이 결정되며, 그때마다 연산 표에 구성될 시그모이드 함수값도 입출력의 범위와 오차정도에 따라 달라지게 된다. 그림 10과 같이 시그모이드 함수구간을 -4 ~ 4로 정한경우 제안된 알고리즘과 실수연산 처리와의 비교에서 거의 같은 반복횟수를 보이고 있지만, 실제 하드웨어 수행상에서는 제안된 알고리즘이 연산속도면에서 빠르기 때문에 신경망의 학습속도는 실수연산을 이용한 경우보다 빠르다. 또한, 제안된 방식은 현재 실용화된 VLSI 기술로 쉽게 구현 할수 있기 때문에 고속 실시간 처리를 필요로 하는 인공지능 응용분야에 현실성있는 한 해결 방식이 될것으로 기대된다.

#### 參 考 文 獻

- [1] Yoh-Han Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley Publishing Company, Inc., 1989.
- [2] STEPHAN, I. GALLANT, "Perceptron-Based Learning Algorithms", *IEEE Transactions on Neural Networks* vol. 1, No.2, 1990.
- [3] D. E. Rumelhart, J. L. McClelland, et al,

*Parallel Distributed Processing*, the MIT Press, vol.1, 1986.

- [4] Takeshi Oohashi, "An Implementation of Multi Layered PDP Models on the Loosely Coupled Multi-processor", *日本電子工學會 論文誌 D-2* vol. j73-D-2, No.8, pp.1354-1359, Aug. 1990.
- [5] JENQ-NENG HWANG, "A Systolic Neural Network Architecture for Hidden Markov Models", *IEEE Transactions Acoustics Speech, and Signal Processing*, vol.37, No. 12, December 1989.
- [6] H.T.Kung, "Why Systolic Architectures?", *IEEE Computer*, pp.37-46, January 1982.
- [7] S.Y.Kung, *VLSI Array Processing*, Prentice Hall international edition, 1988.
- [8] Nicholas S.Szabo, M.S.& Richard I. Tanaka, Ph.D., *Residue Arithmetic And Its Applications to Computer Technology*, McGRAW-HILL Book Company, 1967.
- [9] 조 원경, "RNS를 이용한 연산 프로세서의 설계에 관한 연구", 한양대학교 박사학위 논문 1986.
- [10] 정 윤돈 외 3인, "디지털 신경망 모델의 시그모이드 함수 연산회로 설계에 관한 연구" 전자공학회 하계학술대회 논문집, pp.184-187, 1991.6.
- [11] 윤 현식 외 3인, "디지털 신경 회로망 실현을 위한 어레이 프로세서의 설계", 인공지능/신경망 및 퍼지시스템 워크샵, 1991, 11.
- [12] S. Y. Kung, J. N. Hwang, "A Unified Systolic Architecture for Artificial Neural Networks", *Journal of Parallel and Distribute Computing* vol.6, No.2, PP.358-387, Feb. 1989

#### 著 者 紹 介



尹 賢 植 (正會員)

1947年 7月 14日生 1972年 인하대학교 전자공학과(공학사) 1983年 경희대학교 전자공학과 졸업(공학석사) 1986年 3月 ~ 현재 경희대학교 전자공학과 박사과정 1981年 ~ 현재 국립충주산업대학 전자계산학과 부교수  
주관심 분야는 컴퓨터구조, 신경회로망, VLSI 설계등임

趙 源 敬 (正會員) 第 29卷 B編 第 2號 參照

현재 경희대학교 전자공학과 교수