

어레이 프로세서를 이용한 홉필드 모델의 구현에 관한 연구

홍 봉 화*, 이 지 영**

A Study on the Implementation of Hopfield Model using Array Processor

Bong-Wha Hong* , Jie-Young Lee**

요 약

본 논문은 홉필드 모델의 실수연산을 고속으로 수행할 수 있는 디지털 신경회로망의 구현에 관한 연구이다. 홉필드 모델(1)-(8)의 연산과정은 행렬-벡터의 연산으로 기술 할 수 있으며, 이 연산과정은 순환, 반복적으로 이루어지므로 어레이프로세서 구조로 설계하기에 적합하다. 또한, Look-up-Table(연산표)에 의하여 비선형 함수를 출력함으로써, 고속의 실수 연산을 수행할 수 있도록 설계하였다.

본 논문에서 제안한 방법은 현재 개발된 VLSI기술로 실현 가능하기 때문에 실제 신경회로망의 응용 분야에 이용될 수 있을 것으로 기대된다.

Abstract

This paper concerns the implementation of a digital neural network which performs the high speed operation of Hopfield model's arithmetic operation.

It is also designed to use a look-up table and produce floating point arithmetic of nonlinear function with high speed operation.

The arithmetic processing of Hopfield is able to describe the matrix-vector operation, which is adaptable to design the array processor because of its recursive and iterative operation.

The proposed method is expected to be applied to the field of real neural networks because of the realization of the current VLSI techniques.

* 세명대학교 컴퓨터과학과 교수

** 세명대학교 컴퓨터과학과 교수

논문접수: 1999.10.20. 심사완료: 1999.12.1.

I. 서론

인공 지능의 한 분야로서, 인간의 뇌 기능을 가진 로봇이나 컴퓨터에 대한 연구가 진행되어 왔으며, 이와 같은 연구분야 중에서 인간의 뇌를 단순화한 신경 모델로 모델링한 신경 회로망 연구분야가 대두되었다. 이러한 신경 회로망은 신경단위인 뉴런의 특성을 갖는 기본소자로 구성하여 아날로그 형태로 처리하거나 디지털 형태로 처리하고자 하였으며, 패턴 인식이나 영상처리, 적응 제어 등 여러 분야에 연구의 관점을 갖고 진행되고 있다.[1]-[3][5]

또한, 간단한 특성의 기본소자를 대규모로 상호 연결 시킴으로 인하여 높은 상호 연결성에 따른 병렬처리 특성과 고도의 연산처리 능력을 갖게 되며, 실용화를 위한 가능성을 제시하고 있다.

최근들어 신경회로망은 VLSI 기술의 발전에 힘입어 하드웨어 구현이 가능하게 되었으며 주로 패턴 및 음성인식, 제어이론, 영상처리 등의 분야에 응용하고 있다. 그러나, 아날로그 방법에 의한 대규모 신경회로망의 구현은 VLSI 설계시 잡음 및 소비전력 등의 문제점이 야기된다. 따라서, 현재의 반도체 기술로 실용성 있는 신경회로망을 구현하기 위해서는 디지털 VLSI 기술을 이용한 신경회로망의 설계에 대한 연구가 요구된다. 신경회로망 모델은 순환, 반복적으로 행렬·벡터 연산을 수행하기 때문에 어레이 프로세서구조로 실현하기에 적합하다.[4][6][7][8]

본 논문에서는 신경 회로망의 입력과 내부가중치 연산 과정을 행렬·벡터로 기술했고, 연산표를 이용하여, 비선형 함수(시그모이드)를 고속으로 출력 할 수 있도록 하였으며, 어레이 프로세서 기법을 이용한 디지털 신경회로망 설계방법에 대하여 고찰하였다.

II. 홉필드 모델

그림 1의 홉필드 신경회로망은 1982년 미국 캘리포니아 공과대학 물리학자인 존 홉필드(John Hopfield)에 의하여 제안되었으며, 연상기억장치 또는 콘텐츠 행렬과 같은 시냅틱 가중치의 프로그래밍에 의해 여러 가지 복합적인 문제를 풀기 위해 이용된다.

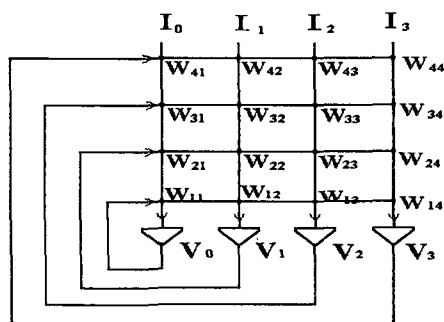


그림 1. 기본적인 홉필드 모델
Fig. 1 The Basic Hopfield model

기본적인 Hopfield 모델은 확률적인 알고리즘에 따라 2가지상태를 갖는 신경회로망이다. 즉 i번째 노드의 두 상태는 V_{i0} , V_{i1} 출력을 갖는다.

각 노드의 입력은 외부입력 I_i 와 다른 노드로부터의 입력, 두 가지 형태로 나타낼 수 있다. Hopfield 모델의 에너지 함수는 다음과 같이 나타낼 수 있다.

$$E = - \sum_{i,j} T_{i,j} V_i V_j - \sum_i I_i V_i \quad (1)$$

식 (1)에서 T_{ij} 는 j노드로부터 i노드로의 내부 연결강도를 의미한다. 주어진 Hopfield 모델의 에너지함수는 서로 다른 두 수준사이의 특성을 갖으며($V_i = 1$ 일때 ON, $V_i = 0$ 일때 OFF) 그 차이는 식 (2)와 같다.

$$E_{i-on} - E_{i-off} = \Delta E_i = - \sum_{j=1}^N T_{ij} V_j - I_i \quad (2)$$

만약 에너지 ΔE_i 가 음이면 유니트는 에너지를 최소화 하기 위하여 ON되어야하며, 반면에 양이면 OFF되어 야 한다. 불연속 모델에 있어 각 노드 i는 노드 j로부터

$T_{ij}V_{ij}$ 의 입력을 받고 bias 전류입력 I_i 를 받는다. i 번째 노드의 불연속적인 상태는 식 (3), (4)와 같이 공식화 할 수 있다.

$$U_i(k) = \sum_{j=1}^N T_{ij}V_j(k) + I_i \quad (3)$$

$$V_i(k+1) = \text{step}[U_i(k)] \quad (4)$$

식(4)에서 $\text{step}(U_i(k))$ 는 단일 step 함수이며, $U_i(k) \geq 0$ 이면 1이고, $U_i(k) < 0$ 이면 0이다. 그리고, N 은 노드의 수이고 T_{ij} 는 내부연결 행렬의 요소로써 각 노드의 연결 가중치를 나타낸다. 입력 U_i 는 증폭기의 임계치에 의해 억제되어지고 출력 V_i 는 그들의 상태를 수정하기 위해 다른 증폭의 입력으로 되돌려진다. 식 (4)를 하드웨어로 구현하기는 쉬우나 최적 해를 구하는데 있어 국소해(local minimum)에 빠질 우려가 있으므로, 본 논문에서는 스텝함수 대신에 실제적인 신경세포의 입출력 관계를 잘 표현하는 시그모이드 함수를 도입하여 전체적인 최소해(global minimum)로 수렴할 수 있도록 하였다. 불연속적인 홉필드 모델을 연속적인 상태 모델로 변형하면 다음과 같다.

$$U_i(k) - U_i(k-1) = \sum_{j=1}^N T_{ij}V_j(k) + I_i \quad (5)$$

$$V_i(k+1) = F[U_i(k)] \quad (6)$$

식 (6)에서 $F(u_i(k))$ 는 시그모이드 함수로써 식 (7)과 같이 표현할 수 있다.

$$F[U_i(k)] = \frac{1}{1 + \exp(-U_i(k) + \theta_j)} \quad (7)$$

식 (5)에서 우변은 흥분원으로 고려될 수 있고 좌변은 주어진 상태의 수정량을 의미하며, 에너지함수로 표현하면 다음과 같이 정의할 수 있다.

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N T_{ij}V_iV_j - \sum_{i=1}^N I_iV_i \quad (8)$$

Ⅲ. 행렬·벡터에 의한 Hopfield 모델의 표현

Hopfield 모델은 연속적인 행렬·벡터연산으로 공식화 할 수 있다. 앞에서 언급한 식 (5), (6)을 행렬·벡터로 나타내면 다음과 같다.

$$U(k) = T V(k) + I + U(k-1) \quad (9)$$

$$V(k+1) = F[U(k)] \quad (10)$$

식 (10)에서 $F[U(k)]$ 함수는 벡터 $U(k)$ 의 각 요소에 대한 비 선형 임계함수 값을 의미하며, 식 (9)와 (10)을 행렬·벡터로 표현하면 식 (11)과 같다.

$$\begin{aligned} U &= [U_1 \ U_2 \ \dots \ U_N]^T \\ V &= [V_1 \ V_2 \ \dots \ V_N]^T \\ I &= [I_1 \ I_2 \ \dots \ I_N]^T \\ T &= \begin{bmatrix} T_{11} & T_{12} & \dots & T_{1N} \\ T_{21} & T_{22} & \dots & T_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ T_{M1} & \dots & T_{MN} \end{bmatrix} \end{aligned} \quad (11)$$

Ⅳ. 어레이 프로세서를 이용한 Hopfield 모델의 설계

Hopfield 신경회로망의 연산과정은 가중치 연산부분과 비선형(시그모이드)함수 연산부분으로 구성될 수 있다.

1 가중치 연산부

가중치 연산부는 들어온 입력과 내부 가중치값에 대한 행렬과 벡터의 연산 결과를 비 선형 함수 연산부로 전송한다.

그림 1의 Hopfield 모델의 행렬·벡터의 연산에 관한 데이터 흐름도는 그림 2와 같다.

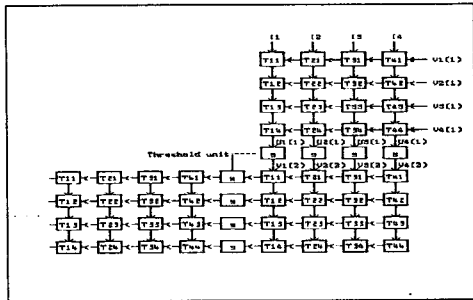


그림 2. Hopfield 모델의 데이터 흐름도
Fig. 2 The data flow of Hopfield model

그림 2의 기본연산 구조를 2차원 어레이 구조로 나타내면 그림 3과 같다.

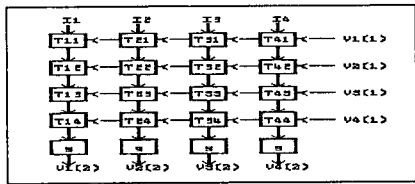


그림 3. Hopfield 모델의 기본연산블록
Fig. 3 The basic arithmetic unit of Hopfield model

그림 2의 프로세서를 직접 하드웨어로 실현하는 경우 속도는 빠르지만 입력선의 수가 증가하면 PE의 수가 크게 증가되므로 하나의 칩에 실현 할 수 없고 비경제적이다. 그러므로 본 연구에서는 그림 2의 프로세서가 그림 3과 같은 기본연산 블록으로 구성되는 점을 이용하여 기본연산블록(그림 3)만을 실현하여 이를 반복적으로 이용한다. 그림 3의 기본연산 블록의 PE는 2차원 배열구조이다. 이를 어레이구조의 설계기법을 이용하여 그림 4와 같이 1차원으로 사상(mapping)하여 1차원 어레이구조로 설계하였다.

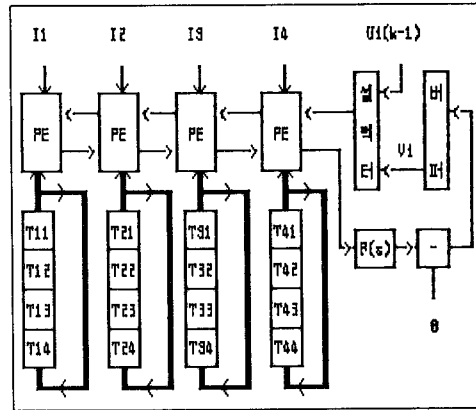


그림 4. Hopfield 모델의 1차원 어레이 구조
Fig. 4 The structure of one dimensional array for the Hopfield model

그림 4의 각 PE(Processing Element)의 내부 구조는 그림 5와 같다.

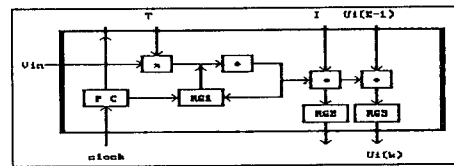


그림 5. Hopfield 모델의 PE구조
Fig. 5 The structure of processing element for the Hopfield model

또한, 그림 4에서 가중치의 순환과정을 나타내면 그림 6과 같이 나타낼 수 있다.

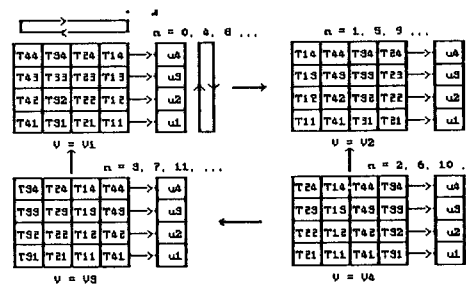


그림 6. 가중치 처리과정
Fig. 6 The processing of weight value

그림 6은 4개의 노드를 갖는 홉필드 모델에 있어, 4개의 쉬프트 레지스터에 저장된 16개의 가중치가 clock(n)

에 따라 순환 되는 과정을 나타내며, $V=V_i(i=1,2,3,4)$ 는 스텝에 따른 연산 결과를 나타낸다. 그림 6에서 K번째 순환일 경우, PE의 동작은 다음과 같이 기술할 수 있다.

- 1) 각 PE의 출력(V_1, V_2, \dots, V_n)은 순환되고, N 클럭 사이클 동안 i번째 PE를 통해 통과할 것이다.
- 2) V_j 가 i번째 PE를 통과할때 T_{ij} 가 곱해지고, 그 결과는 I_i 와 $U_i(K-1)$ 의 값에 더해진다.
3. N 클럭 사이클 후에 $U_i(K)$ 에 대한 계산은 완전하게 되고, 이것은 임계치 동작을 위해 준비한다.
4. 임계치 동작 후에 각 PE의 노드출력은 $V_i(K+1)$ 로 보내진다. 이 과정은 수렴에 도달할 때까지 반복적으로 수행된다.

2. 비선형 연산부

비선형 함수 연산부는 가중치 연산부에서 전송한 데이터를 처리하는 부분으로 가중치 연산의 내용을 종속변수로 하여 현재 계층의 출력을 결정하며 출력을 갱신하여 수렴속도를 조절하기도 한다. 그러나, 기존의 디지털 연산방식에 의하여 구하는 문제는 대단히 어려운 문제이다. 그러므로, 그림 7과 같이 비 선형 함수의 전체 구간을 분할하여 구할 수 있다.

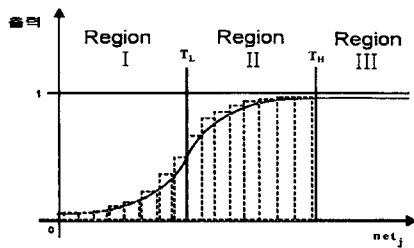
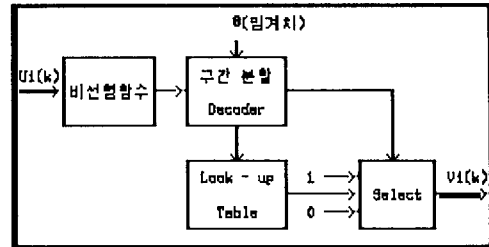


그림 7. 비선형 함수의 처리구간 설정
Fig. 7 The setup of the processing boundary for nonlinear function

그림 7에서 구간 I은 비선형 함수의 입력 값이 임계값(Threshold)보다 작으며, 이 경우 비선형 함수의 출력은 "0"이 된다. 구간 II의 값은 상위 임계치 하위 임계치 사이의 값으로 사전에 저장된 연산표를 통해 출력되며, 구간 III은 비선형 함수의 입력 값이 임계치 보다 큰 경우 "1"를 출력한다. 위의 과정을 간략하게 나타내면 그림 8과 같다.



TL : 하위 임계값 TH : 상위 임계값
그림 8. 비선형 함수의 처리과정
Fig. 8 The processing of nonlinear function

그림 8에서 구간분할 디코더는 비선형 함수의 구간을 정하기 위한 부분이며, 비선형 함수값이 구간 I일 경우 "0", 구간 III일 경우 "1"을 출력하며, 구간 II일 경우는 연산표의 값을 출력한다.

IV. 모의 실험 및 고찰

모의 실험에 이용한 홉필드 모델은 뉴런수가 4개, 8개, 16개인 모델을 이용하였다. 모의 실험에 이용한 가중치 값은 -1 과 1사이의 값을 갖도록 이산적으로 발생하여 실험하였으며, 시그모이드의 구간 II를 16개로 표분화하여 실험하였다.

그림 9는 입력 패턴이 4×4 , 8×8 , 16×16 인 경우, 홉필드 모델의 2차원 어레이 구조와 1차원 어레이 구조의 PE수의 비교이다.

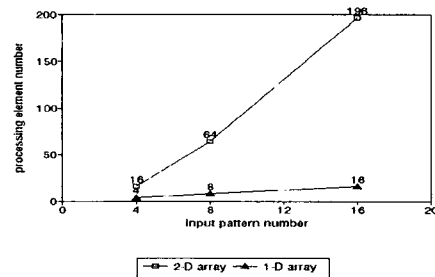


그림 9. 2 차원 어레이 구조와 1차원 어레이 구조의 PE수 비교

Fig. 9 Comparison of PE number of the two dimensional array and one dimensional array

그림 9에서와 같이 2차원 어레이 구조의 PE수는 1차원 어레이 구조의 배가됨을 알 수 있다.

〈표 1〉 에너지 상태와 수렴과정의 비교
(Table. 1) The Comparison of energy state and the processing of convergence

| 상태 | 에너지 | 기존의 홉필드 모델 | 본 논문에서 설계한 홉필드모델 |
|----|------|----------------|------------------|
| 0 | 0 | local minimum | global minimum |
| 1 | -122 | local minimum | global minimum |
| 2 | 108 | local minimum | global minimum |
| 3 | -136 | global minimum | global minimum |
| 4 | 48 | local minimum | global minimum |
| 5 | 34 | local minimum | global minimum |
| 6 | 226 | local minimum | global minimum |
| 7 | 90 | local minimum | global minimum |
| 8 | 0 | local minimum | global minimum |
| 9 | -74 | local minimum | global minimum |
| 10 | 215 | global minimum | global minimum |
| 11 | 19 | local minimum | global minimum |
| 12 | -25 | local minimum | global minimum |
| 13 | 9 | local minimum | global minimum |
| 14 | 260 | local minimum | global minimum |
| 15 | 172 | local minimum | global minimum |

또한, 본 논문에서 설계한 홉필드 모델을 평가하기 위하여 가중치 발생 랜덤 변수 값을 7로 고정하여 실험한 결과 같은 에너지 상태에 도달하는데 있어 본 논문에서 설계한 신경회로망이 표 1에서 알 수 있듯이 안정한 수렴을 나타내었다. 홉필드 모델에 있어 노드의 수를 n 이라 하면 상호 연결의 총수(NT)는 $2n^2$ 이 된다.

본 논문에서 설계한 홉필드 모델의 경우 1회의 연산 시간은 (12)와 같이 나타낼 수 있다.

$$T_w = \max(a + 3b + c) * n \quad (12)$$

윗 식에서 a, b, c,는 승산, 가산, 비선형 함수에 대한 PE의 처리속도이다. 실행 속도는 전체 접속의 수를 실행 시간으로 나눈 값이 되므로 식 (13)과 같이 기술 할 수 있다.

$$S = \frac{N_T}{T_w}$$

S: 실행속도, N_T : 접속의 총수, T_w : 실행시간
(13)

홉필드 모델을 1차원 어레이로 구성했을 때 4개의 입

력과 16개 가중치와의 행렬·벡터 연산과정에 있어 각각의 PE에서는 4번의 승산과 6번의 가산이 이루어지며, 4개의 PE에서 연산된 결과가 출력되기까지는 12스텝이 소요되며, N개의 뉴런을 갖는 경우는 3N스텝이 소요됨을 고찰하였다.

V. 결론

본 논문에서는 홉필드 모델에 필요한 행렬·벡터연산과 비선형 함수 연산을 수행할 수 있는 연산기를 어레이 프로세서 설계 기법을 이용하여 설계하였다. 실제 설계된 연산기는 1차원 어레이구조로서 이러한 구조를 재배열하는 방법으로 입력 데이터선의 수가 변하여도 쉽게 적용할 수 있게 설계하였다. 특히, 고속처리를 필요로 하는 경우에는 동일구조의 프로세서를 파이프라인 방식으로 배열하여 홉필드 모델을 고속으로 처리 할 수 있게 설계하였다.

본 논문에서 설계한 신경회로망의 경우, 4개의 입력과 12개의 가중치와의 연산에서 결과가 출력되기까지는 총 12스텝이 소요되며, N개의 뉴런 수를 가질 경우는 3N스텝이 소요된다. 또한, 본 논문에서 설계한 연산기를 이용하여 패턴 연산에 적용한 결과 30%미만인 패턴의 경우, 거의 98%의 연상효과를 보였다. 제안된 방식은 현재 실용화된 VLSI 기술로 쉽게 실현 할수 있기 때문에 고속 실시간 처리를 필요로 하는 인공신경회로망 응용분야에 현실성 있는 한 해결 방식이 될 것으로 기대된다.

참고 문헌

- [1] " DARPA Neural Network study " : AFCE A International press 1987.
- [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 vol.1", the MIT Press(1986).
- [4] D. A. Arpin and Y. Kim. Parsor : A Parallel Processor for sparse matrix solution by SOR iteration. In Proc. of Int' l cof. on Paralleled processing, 1986.
- [5] IEEE "International Joint conference on neural networks ", Washington D.C June 18-22, vol.2, 1989.
- [6] Takeshi OOHASHI : " An Implementation of Multi Layered PDP Models on The Loosely Coupled Multiprocessor", 日本電子工學會 論文誌 D-2 vol.j 73 - D-2, No. 8, pp. 1354-1359 1990, 8.
- [7] JENQ-NENG : " A Systolic Neural Network Architecture for Hidden Markov Models", IEEE Transaction Acoustics Speech, and signal processing, vol.37, No.12,1989.
- [8] S. Y. Kung : "VLSI Array Processing", Prentice-Hall International edition.1988.

저 자 소개

홍 봉 화

한국 OA학회 논문지 제 4권 2호 참조
현재 세명대학교 컴퓨터 과학과 교수

이 지 영

한국 OA학회 논문지 제3권 2호 참조
현재 세명대학교 컴퓨터 과학과 교수