

# 학습기능을 내장한 신경회로망 모듈 칩 설계

## A Modular Design of a FNNs with Learning

최명렬 · 조화현

한양대학교 전자·전기·제어계측공학과 ASIC Lab.

ASIC Lab. Dept. of EECS, Hanyang University.

Tel: 0345-400-4036, Fax: 0345-501-8114

### ABSTRACT

본 논문에서는 간단한 비선형 시냅스 회로를 이용하여 온 칩 학습기능을 포함한 모듈 칩을 구현하였다. 학습 회로는 MEBP(modified error back-propagation) 학습 규칙을 적용하여 구현하였으며, 제안된 회로는 표준 CMOS 공정으로 구현되었고, MOSIS AMI 1.5 $\mu$ m 공정 HSPICE 파라미터를 이용하여 그 동작을 검증하였다. 구현된 모듈 칩은 온 칩 학습기능을 가진 확장 가능한 신경회로망 칩으로 대규모의 FNNs(feedforward neural networks) 구현에 매우 적합하리라 예상된다.

## I. 서론

많은 연구자들은 많은 종류의 인공신경망 모델을 가지고 신경망의 소프트웨어 또는 하드웨어 형태의 구현에 대해 연구하고 있다. 소프트웨어 구현은 대개 신경망의 해석과 구조에 기반을 둔 알고리즘을 채용하여 기존의 디지털 컴퓨터 상에서 수행된다. 이 소프트웨어 의한 방법은 현재 구현된 신경회로망 모델 및 구조 결정과정에서 유효하게 사용될 수 있다. 그러나 신경회로망의 크기가 증가하게 되면 소프트웨어에 의한 구현은 속도가 느리게 되어 큰 규모의 신경회로망이 필요한 시스템 개발에는 적합하지 않다. 또한 학습과정에 필요한 자료의 양이 방대하고 여러번의 iteration이 소요되므로 학습이 빈번하게 필요한 응용회로에 적합하지 않다. 학습회로를 포함한 인공 신경 회로망의 하드웨어 구현은 일반적으로 선형 곱셈회로를 가지는 선형 시냅스 회로를 사용하여 구현한다 [1,2]. 본 논문에서는 비선형 시냅스 회로와 비선형 곱셈기를 이용하여 단일/다중 학습이 가능한 모듈 칩을 구현하였으며, 학습 회로는 MEBP(modified error back-propagation) 학습

규칙을 적용하여 구현하였다. 디자인한 모듈 칩은 임의의 뉴런 수를 증가시키기 위하여 수평방향과 수직 방향으로 추가적인 연결이 가능하므로 큰 규모의 학습 기능을 가진 FFNNs 구현에 적합하다.

## II. 본론

### 2.1 MEBP 학습 규칙의 구현

EBP(error back-propagation) 학습 규칙은 다층 FNNs을 위하여 가장 널리 사용되는 학습 규칙이다. EBP 규칙은 시냅스 가중치와 그들 사이의 비선형 곱셈을 가진 뉴런의 출력 사이의 선형 곱셈을 교체 하므로써 변형된다. 비선형 곱셈은 비선형 함수의 미분으로써 간단히 계산되기 때문에 시그모이드 형태의 비선형 함수가 비선형 곱셈기로써 동작한다.

MEBP 학습 규칙은 EBP 규칙의 안정성과 수렴도를 잃지 않으면서 EBP 규칙의 시그모이드 도함수 항을 제외한 것이다[3,4,5]. MEBP 학습 규칙에서 뉴런  $k$ 가 출력 층에 있으면 식 (1)로써 표현되며, 은닉 층에 뉴런  $j$ 가 있는 경우

식 (2)로 표현된다.

$$\bar{w}_{kj} = \eta(\tau_k - \bar{y}_k)y_j - \alpha_{ji}\bar{w}_{ji} \quad (1)$$

$$\dot{w}_{kj} = \eta \sum_m \bar{w}_{mj}(\tau_m - \bar{y}_m)y_i - \alpha_{ji}w_{ji} \quad (2)$$

여기서  $\bar{y}_k$ 는 출력 층 뉴런  $k$ 의 출력이고,  $y_j$ 는 은닉 층 뉴런  $j$ 에서의 출력이고,  $\eta$ 와  $\alpha$ 는 상수를 나타낸다. 비선형 MEBP 규칙은 EBP 규칙의 안정과 수렴도를 잃지 않으려면 선형 MEBP 규칙에 비감소 미분 단조 시그모이드형 함수(nondecreasing differentiable monotone sigmoid-type function)를 이용해야만 한다. 일반적으로 입력 과 출력 층 뉴런  $j$ 는 식(2)와 식(3)으로 주어진다.

$$u_j = \sum_i g_i(w_{ji} \cdot o_{pi}) \quad (2)$$

$$y_i = S_j(u_j) \quad (3)$$

여기서  $u_j$ 는 뉴런  $j$ 의 입력이다.  $y_j$ 는 뉴런  $j$ 의 출력이고,  $S_j(\cdot)$ 는 시그모이드 함수이며,  $g(\cdot)$ 는 비감소 단조 미분 함수이다. 그리고 비선형 MEBP 규칙은 식 (4)처럼 주어진다.

$$\bar{w}_{kj} = \eta \sum_p f(\tau_{pk} - \bar{y}_{pk}) \cdot y_{pj} - \alpha_{ji}\bar{w}_{ji} \quad (4)$$

$$\dot{w}_{ji} = \eta \sum_k g(w_{kj} \cdot f(\sum_p (\tau_{pk} - \bar{y}_{pk}) \cdot y_{pj})) - \alpha_{ji}w_{ji}$$

여기서  $f(\cdot)$ 와  $g(\cdot)$ 는 비감소 미분 단조 시그모이드형 함수이다.

## 2.2 모듈 칩의 구현

하나의 모듈 칩은 출력 층의 학습 회로와 은닉 층의 학습회로가 동일하므로 어떤 층에서도 사용 가능하도록 설계하였다.  $n \times m$  모듈 칩의 블록 다이어그램을 그림 1에 나타내었으며, 그림 1의 오른쪽 아래 부분의 박스는  $n \times m$  모듈 칩의 표현이다. 여기서  $n$ 는 입력의 수이고  $m$ 는 출력 수를 나타낸다.  $\bar{e}$ 는 다음 상위 층으로부터  $n$ 차원 에러 벡터를 나타낸다.  $e$ 는 이전의

하위 층에 대한  $n$ 차원 BPE(back-propagated error)를 나타낸다.  $n$ -D는  $n$ -차원 벡터 곱셈기를 나타내고, 1-DI는 capacitor를 포함한 1-차원 곱셈기를 나타내며 MUX는 2x1 아날로그 멀티플렉서를 나타낸다. 그림 2는 큰 규모의 FNNs의 확장에 사용하기 위한  $m$ 차원 nMOS 아날로그 덧셈기를 나타내었다. MUX는 입력  $y_i$  또는 0을 select 신호 의해서  $z_i$ 를 출력하게 된다.

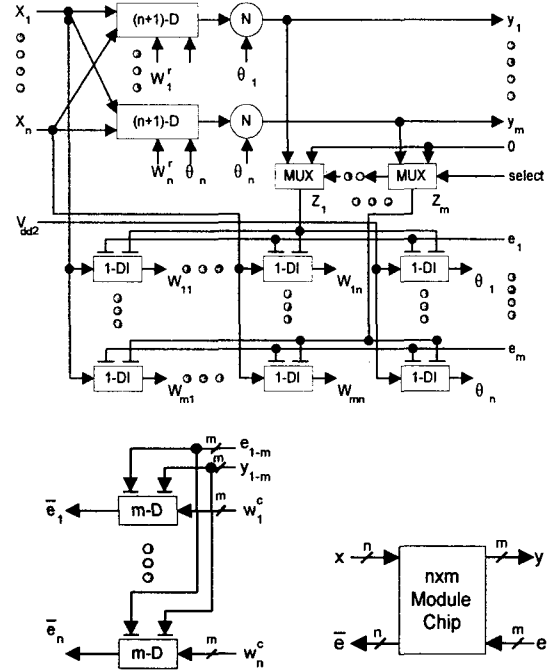


그림 1.  $n \times m$  모듈 칩의 블록 다이어그램

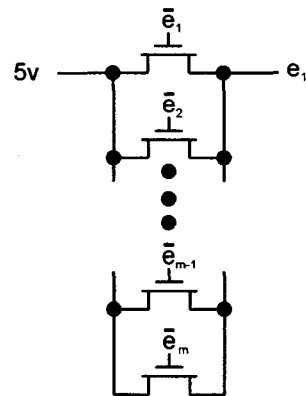


그림 2.  $m$  차원 아날로그 비선형 덧셈기

모듈 칩은 피드포워드(feedforward) 회로와 학습회로 구성된다. 피드포워드 회로는 출력  $m$

( $y_i$ )를 발생시킨다. 출력  $y_i$ 는 다음 상위층의 입력으로 적용되거나 가중치( $w_{ij}$ ) 계산에 사용된다.  $n \times m$  모듈 칩의 회로 구현은 식 (5)로 표시된다.

$$y_i = s_n(k_{n+1}(\sum_j w_{ij}x_j + \theta_i)) \quad (5)$$

$$RC\dot{\omega}_{kj} = k_1(e_k - z_k) \cdot x_j - \omega_{kj}$$

$$\bar{e}_i = k_m \sum_m w_{mi}(e_i - z_i)$$

여기서 모듈 칩이 출력 층에 사용시  $z_i = y_i$ 이고, 은닉 층인 경우  $z_i = 0$ 을 출력한다.

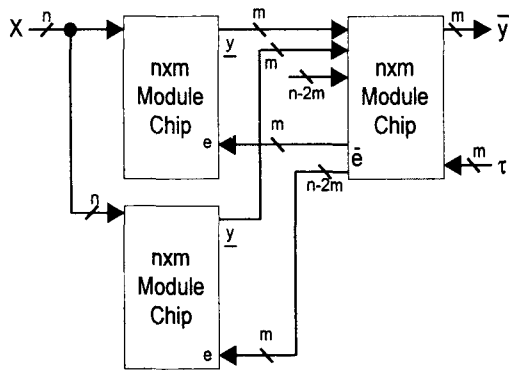


그림 3. 학습기능을 내장한  $n \times 2m \times m$  FNNs

그림 3은 수평과 수직 방향으로 확장하기 위하여 모듈 칩을 어떻게 이용하는가에 대한 간단한 그림을 나타내었다. 입력 층은 단지 전압 노드  $x$ 로 구성 되어있다. 중간에 위치한 두 개의 모듈 칩은 은닉 층을 구성하며, 은닉 층의 출력은 출력 층의 입력으로 작용하게 된다. 출력 층에서 목표 벡터는 에러 벡터( $e$ )를 통하여 공급되게 된다. 그러면 BPE(back-propagation error) 벡터를 생성하고 은닉 층에 존재하는 모듈 칩의 에러 벡터 노드에 적용된다. 만약 더 이상의 하위 층이 없다면, 에러 벡터는 더 이상 전달되지 않는다. 표 2는 HSPICE 시뮬레이션으로  $4 \times 4 \times 2$ 의 결과를 나타내었다. 이 표에서 제안된 모듈 칩의 출력( $\bar{y}_1, \bar{y}_2$ )은 목표치( $\tau_1, \tau_2$ )에 수렴하며, 안정된 가중치 값을 가지는 것으로 나타났다.

표 1.  $4 \times 4 \times 2$  시뮬레이션 결과

|               |                  |      |      |      |      |      |
|---------------|------------------|------|------|------|------|------|
| Input         | $x_1$            | 0    | 0    | 5    | 5    | 5    |
|               | $x_2$            | 0    | 5    | 0    | 5    | 5    |
|               | $x_3$            | 0    | 5    | 5    | 5    | 0    |
|               | $x_4$            | 0    | 5    | 0    | 0    | 0    |
| Target        | $\tau_1$         | 0    | 0    | 0    | 0    | 0    |
|               | $\tau_2$         | 5    | 5    | 5    | 5    | 5    |
| Actual output | $\bar{y}_1$      | 0    | 0    | 0    | 0    | 0    |
|               | $\bar{y}_2$      | 4.7  | 4.8  | 4.8  | 4.8  | 4.8  |
| Weight        | $w_{1-11}$       | 5    | 5    | 0    | 0    | 0    |
|               | $w_{1-12}$       | 5    | 0    | 5    | 0    | 0    |
|               | $w_{1-13}$       | 5    | 0    | 0    | 0    | 5    |
|               | $w_{1-14}$       | 5    | 0    | 5    | 5    | 5    |
|               | $w_{1-21}$       | 5    | 5    | 0    | 0    | 0    |
|               | $w_{1-22}$       | 5    | 0    | 5    | 0    | 0    |
|               | $w_{1-23}$       | 5    | 0    | 0    | 0    | 5    |
|               | $w_{1-24}$       | 5    | 0    | 5    | 5    | 5    |
|               | $w_{2-11}$       | 5    | 5    | 0    | 0    | 0    |
|               | $w_{2-12}$       | 5    | 0    | 5    | 0    | 0    |
|               | $w_{2-13}$       | 5    | 0    | 0    | 0    | 5    |
|               | $w_{2-14}$       | 5    | 0    | 5    | 5    | 5    |
|               | $w_{2-21}$       | 5    | 5    | 0    | 0    | 0    |
|               | $w_{2-22}$       | 5    | 0    | 5    | 0    | 0    |
|               | $w_{2-23}$       | 5    | 0    | 0    | 0    | 5    |
|               | $w_{2-24}$       | 5    | 0    | 5    | 5    | 5    |
|               | $\bar{w}_{1-11}$ | 5    | 5    | 5    | 5    | 5    |
|               | $\bar{w}_{1-12}$ | 5    | 5    | 5    | 5    | 5    |
|               | $\bar{w}_{1-13}$ | 5    | 5    | 5    | 5    | 5    |
|               | $\bar{w}_{1-14}$ | 5    | 5    | 5    | 5    | 5    |
|               | $\bar{w}_{2-11}$ | 1.47 | 1.13 | 1.13 | 1.13 | 1.13 |
|               | $\bar{w}_{2-12}$ | 1.47 | 1.13 | 1.13 | 1.13 | 1.13 |
|               | $\bar{w}_{2-13}$ | 1.47 | 1.13 | 1.13 | 1.13 | 1.13 |
|               | $\bar{w}_{2-14}$ | 1.47 | 1.13 | 1.13 | 1.13 | 1.13 |

### III. 결론

본 논문에서는 온 칩 학습기능을 수행 할 수 있으며, 비선형 시냅스 회로와 MEBP(modified error back-propagation) 학습 규칙을 적용하여 온 칩 모듈 회로를 구현하였다. 학습 규칙은 단일/다중 패턴에 대하여 학습을 수행하기 위하여 피드포워드(feedforward) 회로와 학습 회로 사이에 스위치 제어 회로를 추가하여 구현하였다. 디자인한 모듈 칩은 임의의 뉴런 수를 증가시키기 위하여 수평방향과 수직 방향으로 추가적인 연결이 가능하므로 대규모의 학습 기능을 가진 FFNs 구현에 적합하리라 예상된다.

---

본 연구는 과학기술부의 뇌정보처리연구  
사업의 지원을 받아 수행되었습니다.

### IV. 참고 문헌

- [1] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group Eds., Parallel Distributed Processing Explorations in the Microstructure of Cognition , vol. 1, Foundations, Cambridge, MA:MIT Press, 1986.
- [2] F. Kub et al., Programmable Analog Vector-Matrix Multipliers , IEEE J. Solid-State Circuits, SC-25, pp. 207-214, Feb. 1990.
- [3] M. R. Choi, Analog Vector Multipliers for Feedforward Neural Networks , Journal of Engineering & Technology, Vol. 2. No. 1, Hanyang University, Korea, pp. 13-21, Sept. 1993.
- [4] M. R. Choi and F. Salam, Implementation of Feedforward Artificial Neural Networks With Learning Using Standard CMOS Technology , ISCAS 91, Singapore, pp. 1509-1512, May 1991.
- [5] M. R. Choi, Implementation of Feedforward Neural Networks with Simple Nonlinear Circuits , Journal of Hanyang