論文94-31B-4-18

비선형 시냅스를 갖는 확장 가능한 Analog Neuro-chip의 설계

(Design of Expandable Neuro-Chip with Nonlinear Synapses)

朴 晶 培*, 崔 倫 競*, 李 壽 永*

(Jeong Bae Park, Yoon Kyung Choi, Soo Young Lee)

要 約

대규모 신경회로망 시스템을 구현하기 적합한 비선형 시냅스를 갖는 확장 가능한 Analog 신경회로망 칩을 설계하였다. 집적도를 높이기 위해 시냅스를 단 한개의 MOSFET을 사용하여 구현 하였으며, 여러 개의 칩을 연결하여 뉴런이 갖는 시냅스 수를 확장할 수 있도록 되어있다. 이 회로의 시냅스는 비선형적인 동작을 하게되므로 Error-Back-Propagation 방법을 수정하여 학습에 이용하였으며, 수정된 Error-Back-Propagation 방법은 MOSFET 고유의 전류-전압 특성을 수식에 이용하도록 되어있다.

Abstrist

An analog neural network circuit for high density integration is introduced. It's prototype chip is designed in 3 by 3 mm2 die. It uses only one MOSFET to implement a synapse. The number of synapses per neuron can be expanded by cascading several chips. The influence of nonlinearity in synapses is analyzed. A formalization of the back propagation which can be applied to this circuit is shown. Some simulation results are shown and disscussed.

1. 서론

신경회로망의 연구는 인간 특유의 사고능력을 뇌세포를 모델로 한 계산소자를 이용하여 컴퓨터에 구현하고자 하는 것이다. 인간은 일상생활에서 글을 읽거나, 걸어 다니거나, 이야기를 주고 받는 등의 일을 쉽게 행하고 있으나, 이를 컴퓨터에 구현하기 위해서는 엄청난 양의 정보를 빠른 시간 안에 처리할 수 있는 능력이 요구된다. 그러나 기존의 Serial Machine (von Neumann형 컴퓨터)은 수치계산과 Data 저장능력은 우수하나 Algorithm에 따라 한번에 하나씩 처리하므로 인간 특유의 직관적인 사고는 구조적으로 곤란한 면이 있다. 반면에 신경회로망 컴퓨터는 Neuro Chip을 이용할 경우 정보처리가 동시에 이루어지므로 빠른 계산속도를 기대할 수 있으며 회로 자체가 학습하므로 Algorithm을 찾기 힘든 문제에도 적용할 수 있는 장점이 있다.

신경회로망은 현재 음성인식, 문자인식, Robot 제어 등의 여러분야에 응용되고 있다. 음성인식 분야에서는 사람이 키보드를 치는 대신에 말로하여 컴퓨터

* 正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electroical Eng., KAIST)

接受日字 : 1993年 3月 8日

에 입력시키는 음성인식 컴퓨터를 목표로 연구중이며, Control 분야에서는 종래의 제어 방법으로는 다루기 곤란한 비선형적인 시스템을 신경회로망을 이용해 Identification하여 제어하는 방법 등을 활발히 연구중이다. 그밖의 분야에서는 신경회로망을 이용한 필기체 문자 인식 시스템 등을 연구하고 있고, 작곡이나 주가 예측 등에도 신경회로망이 이용되고 있다. 그러나 이와같은 응용분야에서 실시간 처리와 같은 요구에 부응하기 위해서는 기존의 컴퓨터에서 Simulation에 의해 인공 신경회로망을 구현하는 방법으로는 빠른 계산속도를 얻기 어려우므로 병렬처리 효과를 얻을 수 있는 하드웨어의 필요성이 절실해진다.

신경회로망 칩을 설계하는 데 가장 중요한 것은 얼마나 많은 시냅스를 구현하여 얼마나 빠르게 동작하는가 하는 문제일 것이다. 제한된 칩 면적 안에 많은 시냅스를 구현하기 위해서는 회로가 간단해야 할 것이고, 한 칩에 구현할 수 있는 시냅스보다 큰 시스템을 꾸미기 위해서는 칩 단위로 뉴런을 확장할 수 있어야 한다. 본 논문에서는 이 두가지 문제에 촛점을 맞추어 회로를 설계하였다. 현재 상당수의 칩들이 칩 단위로 뉴런을 확장시킬 수 없도록 설계되어 있고, 계산회로가 복잡하여 많은 계산시간과 넓은 칩 면적을 요구한다. 이 논문에서 제안한 One MOSFET Synapse 회로(이하 OMS회로라 함)는 이 두가지 관점에서 볼 때 매우 효과적인 해결책이라고 생각한다. 특별한 곱셈회로나 덧셈회로 없이 각각 하나의 시냅스 역할을 하는 MOSFET들의 연결로 뉴런을 구성하여 출력을 낼 수 있도록 되어 있고, 시냅스의 비선형 특성을 고려하여 학습시에 MOSFET의 전류-전압 특성을 Back Error Propargation에 적용하여 학습시킨다.

Ⅱ. One MOSFET Synapse 회로의 기본 원리

기존의 Analog 방식의 칩에서는 Summation하는 회로를 주로 전류 형태로 구현한다. 전류에 의한 덧셈 회로는 시냅스들의 출력값을 전류 형태로 표현하여 이 전류들을 한 Node에 보내 전류의 합이 얻는다.[1] [2] 이 회로는 구성이 간단하고 정확한 면이 있으나 신호의 흐름과 Weight의 저장은 전압 형태로 이루어지는 것이 편리하므로 전압에 비례하는 전류를 발생시키는 회로 등이 필요하고 여러 단계를 거쳐야 연산이 이루어진다. OMS회로는 신호의 Summation이 전류에 의해 행해지지 않고 전압형태로 이루어진다. 따라서 전압을 전류로, 전류를 전압으로 바꾸는 작업이 불필요하고 칩 외부에서 Summation이 이루어지므로 자유롭게 뉴런을 확장할 수 있다.

OMS회로의 기본 원리는 그림 1의 회로로 설명할 수 있다. 그림 1에서 G_k는 Conductance 값을 나타내며, G_1, G_2, G_3, ... G_n이 모두 일정할 경우 $\hat{y}$ 는 x_1, x_2, x_3, ... x_n의 산술 평균이 될 것이다. 그러나, G_1, G_2, G_3, ... G_n이 각각 다를 경우 $\hat{y}$ 는 x_1, x_2, x_3, ... x_n 중의 가장 큰 값과 가장 작은 값 사이의 어떤 값으로 결정되는데 큰 Conductance로 연결된 x일수록 $\hat{y}$ 에 영향을 많이 미치게 될 것이다. 이는 신경회로망의 뉴런과 시냅스의 역할과 유사하며 이를 수식으로 나타내면 $\hat{y}$ 값은 KCL(Kirchhoff's Current Law)에 따라 다음과 같이 구해진다.

$$\sum_{i}^{n} G_i(x_i - \hat{y}) = 0 \tag{1}$$

$$\hat{y}\sum_{i}^{n} G_i = \sum_{i}^{n} G_i x_i \tag{2}$$

$$\hat{y} = \frac{\sum_{i}^{n} G_i x_i}{\sum_{i}^{n} G_i} \tag{3}$$

여기서 $w_i = G_i / \sum_{i}^{n} G_i$ 로 생각하면 식(3)의 $\hat{y}$는 뉴런의 Net Input ($\sum_{i} w_i x_i$)과 같음을 알 수 있다. 이 회로의 시냅스는 Normalize Term($\sum_{i}^{n} G_i$)이 더 첨가되어 있으며, 이는 이론적으로 무한히 많은 시냅스를 연결하여도 Net Input값이 항상 일정한 동작범위를 갖게한다.

OMS회로는 그림 1의 회로를 VLSI기술을 이용해 구현하기 위하여 각 저항들을 Ohmic Region에서 동작하는 하나의 P-MOS Transister로 꾸몄다. 각

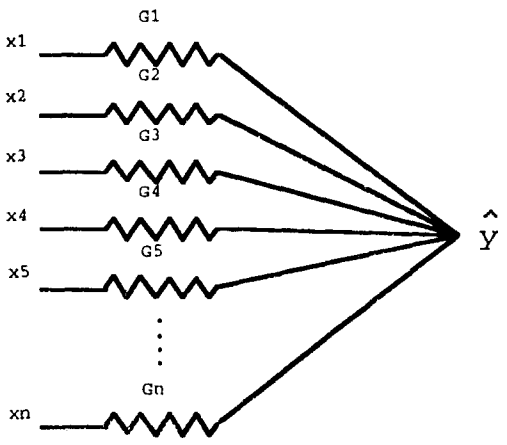


그림 1. 저항에 의한 전압 덧셈 회로

저항값은 MOSFET의 Gate 전압을 이용하여 변화시킬 수 있게 하였다. 즉 하나의 MOSFET가 하나의 시냅스 역할을 하며 Weight는 Analog 전압의 형태로 Gate에 연결한다. ETANN [2], ECAM [3] 과 비교해 볼 때 ETANN의 회로는 시냅스 당 6개의 MOSFET가 필요하고 ECAM은 80개(8개의 패턴을 수록 할 경우)가 필요한데 반해 OMS회로는 단 1개의 MOSFET로 시냅스가 구현되므로 훨씬 경제적이다. OMS회로의 시냅스와 뉴런이 그림 2에 나타나 있다. 출력측의 Sigmoid회로는 Buffer 역할도 동시에 하여 이와 연결된 시냅스들을 구동할 수 있는 능력이 있도록 설계하였다.

MOSFET Device는 Drain, Gate, Source 전압에따라 Ohmic Region, Saturation Region, Subthreshold Region, Cut Off Region으로 동작범위가 구분 된다. 이중에서 Ohmic Region은 전압-전류 특성이 저항에 가까운 영역으로 V_{ds}가 커질수록 많은 전류가 흐른다. 이동작 영역에서 Drain과 Source를 흐르는 전류는

$$I = K\frac{W}{L}\left(V_{gs} - V_t - V_{ds}/2\right)V_{ds} \qquad (4)$$

식(4)로 근사화 할 수 있다. 이 식을 다시 고쳐쓰면,

$$I = K\frac{W}{L}\left[\frac{1}{2}\left(V_g - V_t - V_s\right)^2 + \frac{1}{2}\left(V_g - V_t - V_s\right)^2\right] \qquad (5)$$

식(5)와 같이 나타내어진다. OMS회로에서 Gate에는 Weight가 걸리고, Drain 및 Source에는 입력과 Net Input이 연결된다. 따라서 V_g V_s V_d를 w_{ij}, x_i, $\hat{y}_j$로 치환 하고 KCL을 적용하면

$$\sum_i\left[\frac{1}{2}\left(w_{ij} - V_t - x_i\right)^2 + \frac{1}{2}\left(w_{ij} - V_t - \hat{y}_j\right)^2\right] = 0 \qquad (6)$$

여기서 V_t를 상수라고 가정하고 w_{ij}-V_t를 편의상 w_{ij}'으로 표기하자.

$$\sum_i\left(w_{ij}' - x_i\right)^2 = \sum_i\left(w_{ij}' - \hat{y}_j\right)^2 \qquad (7)$$

식(2.7)를 정리하면,

$$\hat{y}_j = \frac{\sum_i w_{ij}' - x_i}{\sum_i w_{ij}'} - \frac{\sum_i x_i^2 - n\hat{y}_i^2}{2\sum_i w_{ij}'} \quad where\ i = 1,2,\cdots,n \qquad (8)$$

식(8)의 두번째 항은 서로 비슷한 크기를 갖는 두 항의 차가 $2\Sigma w_{ij}$로 나누어지므로 첫번째 항에 비해 상대적으로 그 효과가 작을 것을 예측할 수 있다. 이 식을 식 (3)과 비교해 볼 때 이 회로는 대체로 그림 1의 회로와 비슷하게 동작하며 약간의 이차함수적인 요소를 포함하고 있음을 알 수가 있다.

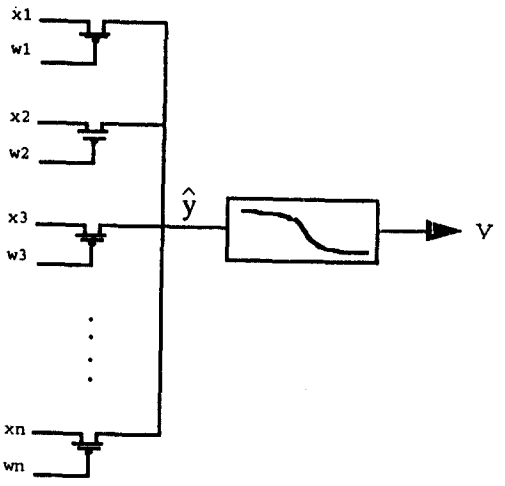


그림 2. OMS회로의 스냅스와 뉴런

보다 정밀한 해석을 위해 SPICE Simulator를 이용해 시냅스의 특성을 알아보고자 한다. Simulation에 사용한 회로는 5개의 MOSFET 시냅스가 연결된 구조이며 한개 시냅스의 x와 w를 변화 시키고 나머지 4개 시냅스의 w와 x는 고정 시킨 상태에서 w와 x의 변화에 대한 $\hat{y}$의 변화를 관찰하였다(그림 3, 그림 4.a ~ 그림 4.e 참조). 그림에서 가로축은 x값을 나타내고 세로축은 $\hat{y}$를 나타낸다. 선형 시냅스를 갖는 경우에서 한 개를 제외한 나머지 시냅스의 Weight와 입력을 고정시켜 놓으면

$$\hat{y}_j = w_k x_k + K \qquad (9)$$

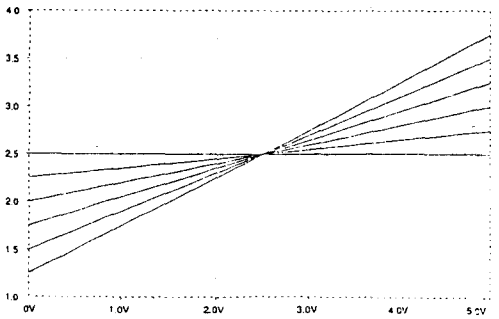
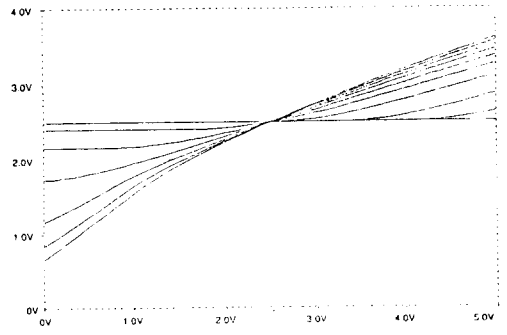


그림 3. 선형 시냅스를 갖는 경우

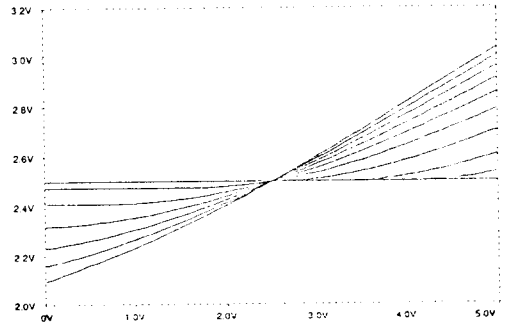
와 같이 되어 w_k에따라 여러가지 기울기를 갖는 직선(그림 3)이 될것이다. OMS회로는 이와는 달리 식(8)에서 예측한 것처럼 2차 함수적인 Curve를 그릴 것이다. 그러나 OMS회로는 다른 시냅스들이 어떤 상태인가에 따라 다른 모양의 Curve를 그리게 되므로 아래와 같은 몇가지 조건에 대해 실험해 보았다.

* 중간 전압에서 동작할 경우
  (a) 다른 시냅스들이 적은 Conductance로 연결된 경우(그림 4.a)
  (b) 다른 시냅스들이 큰 Conductance로 연결된 경우(그림 4.b)
  (c) 10개의 시냅스가 연결된 경우(그림 4.c)
* 가장 높은 전압에서 동작할 경우(그림 4.d)
* 가장 낮은 전압에서 동작할 경우(그림 4.e)

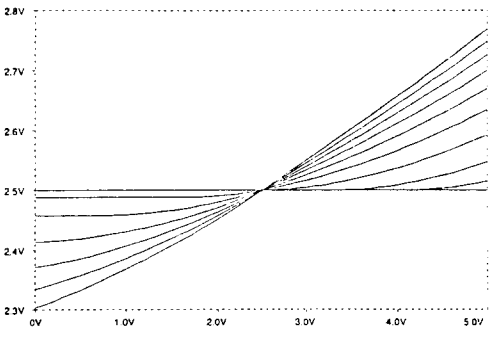
그림 4의 (a)~(e)를 비교해 보면 식(8)에서 보았듯이 다른 시냅스들이 큰 값을 갖거나 많은 시냅스들이 연결된 경우에는 $\hat{y}$ 의 변화의 폭이 좁아 짐을 알 수 있다. 따라서 많은 시냅스가 연결된 경우 원하는 y값에 도달 하도록 학습 시키기 위해서는 어느 한두개의 시냅스의 역할로는 불가능 하며 많은 시냅스들이 함께 작용 해야 한다. 이는 학습되는 정보가 보다



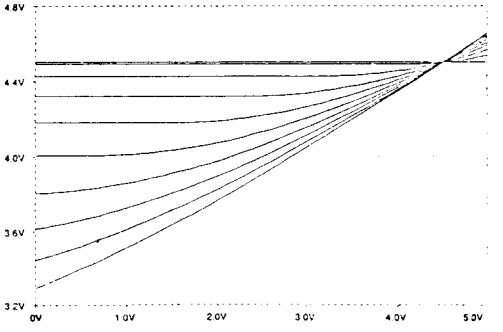
(a)



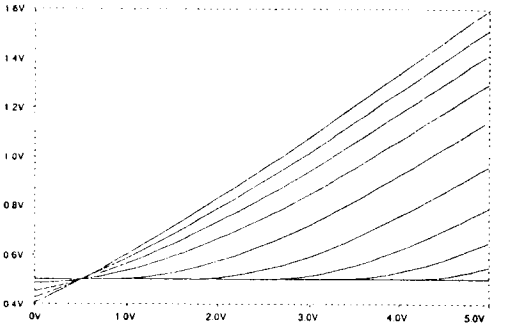
(b)



(c)



(d)



(e)

그림 4. (a) 중간 전압에서 다른 시냅스들이 적은 값을 갖는 경우
(b) 중간 전압에서 다른 시냅스들이 큰값을 갖는 경우
(c) 중간 전압에서 많은 시냅스들이 연결된 경우
(d) 높은 전압에서 동작하는 경우
(e) 낮은 전압에서 동작하는 경우

더 많은 시냅스에 분산되어 저장되도록 한다. $\hat{y}$의 값을 일반적인 수식으로 표현하면,

$$\sum_i g(x_i, w_{ij}, \hat{y}_j) = 0 \tag{10}$$

식(10)과 같이 쓸 수 있다. 여기서 g는 Drain, Gate, Source 전압이 각각 x_i, w_{ij}, y_j 일때 시냅스로 꾸며진 P-MOS의 전류를 나타내는 함수이다. 그림 4에서 시냅스의 특성은 비선형성이 많이 나타나나, $\hat{y}$의 값은 g(x, w, y)함수의 단조 증가 성질에 의해 언제나 x와 w에 대하여 단조 증가(Monotonic Increasing)하는 것을 볼 수 있다. 하드웨어에서는 $\hat{y}$의 값이 순간적으로 결정되나, 이를 Simulation에 의해 구하기 위해서는 Newton-Raphson Method를 통해 $\hat{y}$ 값을 바꾸어 가면서 식(10)을 만족하는 $\hat{y}$를 찾아가야 한다. $\hat{y}$이 x_i중의 가장 작은 값이면 식(10)의 좌변은 항상 음의 값이고, $\hat{y}$이 x_i중의 가장 큰 값이면 식(10)의 좌변은 항상 양의 값을 갖을 것이다. 그런데, $\hat{y}$은 이 구간에서 단조 증가하므로 $\hat{y}$의 값은 언제나 x_i의 최대 성분과 최소 성분 사이에서 Unique하게 결정됨을 알 수 있다.

Ⅲ. MOSFET의 특성을 이용한 학습방법

신경회로망 학습에서 주류를 이루고 있는 것은 Gradient Descent에 의한 학습방법이다. 이는 $\partial E/\partial w_{ij}$를 구하여 이와 반대 방향으로 일정한 비율만큼 시냅스를 변화시켜 주고, 이를 반복함으로써 Error를 점점 감소시켜가는 방법이다. 즉,

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{11}$$

식(11)에 의한 시냅스 변화를 반복하여 Global Minimum에 도착하게 한다. 여기서 η 는 학습률(Learning Rate)이라 하며 학습시 Weight를 바꾸어 주는 정도를 조절하는 계수이다. 이러한 학습방법은 시냅스가 선형적인 경우에만 적용이 되는 것이 아니라 시냅스가 비선형적인 경우에도 적용할 수 있다. 다만 시냅스가 비선형적인 경우에는 Local Minima 문제가 더욱 심해지는 경우가 있고, 학습시키기 위해 필요한 계산 양이 많게 되나 시냅스의 비선형이 크지 않은 경우 학습이 가능한 것으로 알려져 있다.[4] 그러나 OMS회로의 경우에는 비선형성은 그리 심하지 않으나 MOSFET들의 Loading Effect에 의해 출력이 계산되므로 시냅스가 비선형적으로 동작하는것 이외에 각각의 시냅스들은 서로 독립적이지 않고 서로 영향을 준다.

그러므로 단순히 비선형 시냅스를 갖는 신경회로망보다 좀 더 복잡한 경우라 할 수 있으며 이를 학습시키기 위해서는 기존의 학습방법을 이 회로에 적용할 수 있게 바꾸어야 한다. 따라서, 이 회로에서는 학습을 시키기 위해 필요한 $\triangle w_{ij}$를 구하는데 있어 다른 시냅스들의 조건을 고려해 주어야 한다. 그러나 이 회로 또한 Error가 줄어드는 방향으로 w_{ij}(게이트 전압)들을 바꾸어주어 가면 학습이 될 것이며, MOSFET 의 전압-전류 특성 곡선을 이용하여 $\partial E/\partial w_{ij}$을 구할 수가 있었다.

수식에 사용된 기호들의 이해를 돕기 위해 그림5와 같은 2 Layer 구조의 Perceptron의 예를 들어 설명하고자 한다.

여기에서 w_{ij}는 첫번째 Layer의 시냅스들을 나타내며, w_{jk}는 두번째 Layer의 시냅스들을 나타낸다. p번째 입력 패턴이 입력되었을때 얻고자 하는 출력 패턴을 Y_k^p라고 하면 이 신경회로망의 Error는 식(12)과 같이 정의할 수 있다.

$$E^p = \sum_k \left(Y_k^p - y_k^p\right)^2 \tag{12}$$

Batch Type으로 학습을 시킬 경우에는 모든 패턴에 대한 $\partial E/\partial w_{ij}$를 합하여 한꺼번에 Weight를 바꾸어 주고, Pattern Type으로 학습을 시킬 경우에는 각 패턴에 대한 $\partial E/\partial w_{ij}$를 Summation 하지 않고 한번에 한 패턴 씩만 가하여 그때마다 Weight를 Update 하게 된다. 아래 전개되는 내용은 Pattern Type으로 학습을 시키는 경우에 대한 수식이며 따라서, 첨자 p는 생략한다.

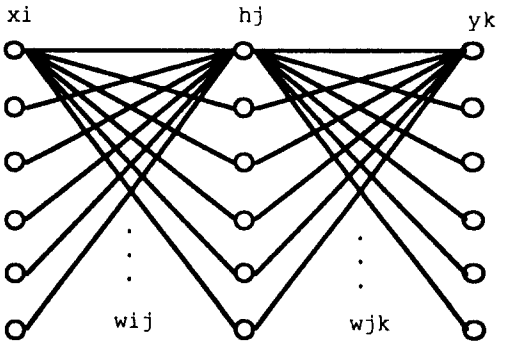


그림 5. 2-Layer Perceptron의 구조

Error를 w_{ij}로 미분하면,

$$\frac{\partial E}{\partial w_{jk}} = 2(Y_k - y_k)\frac{\partial f(\hat{y}_k)}{\partial \hat{y}_k}\frac{\partial \hat{y}_k}{\partial w_{jk}} \tag{13}$$

여기서 f(•)은 Sigmoid함수를 나타내고, $\partial \hat{y}_k / \partial w_{jk}$ 은 다음과 같이 식(10)을 미분하여 구할 수 있다.

$$\sum_j g\left(h_j, w_{jk}, \hat{y}_k\right) = 0 \quad \text{where } \hat{j} = 1,2,\cdots,j,\cdots \tag{14}$$

$$\sum_j \left[\frac{\partial g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial w_{jk}} + \frac{\partial g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial \hat{y}_k}\frac{\partial \hat{y}_k}{\partial w_{jk}}\right] = 0 \tag{15}$$

$$\frac{\partial \hat{y}_k}{\partial w_{jk}} = -\frac{\dfrac{\partial g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial w_{jk}}}{\displaystyle\sum_j \frac{\partial g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial \hat{y}_k}} \tag{16}$$

따라서,

$$\Delta w_{jk} = -2\eta(Y_k - y_k)\frac{\partial f(\hat{y}_k)}{\partial \hat{y}_k}\frac{\dfrac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial w_{jk}}}{\displaystyle\sum_j \frac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial \hat{y}_k}} \tag{17}$$

$\triangle w_{jk}$가 위와 같이 구해지므로 이를 이용하여 두번째 Layer의 시냅스를 학습시킬 수 있다.

Error의 첫번째 Layer의 시냅스에 대한 미분($\partial E/\partial w_{ij}$)을 구하기 위해서는 $\partial E/\partial h_j$를 먼저 구해야 한다. 식(12)를 h_j에대해 미분하면,

$$\frac{\partial E}{\partial h_j} = \sum_k 2(Y_k - y_k)\frac{\partial f(\hat{y}_k)}{\partial \hat{y}_k}\frac{\partial \hat{y}_k}{\partial h_j} \tag{18}$$

$\partial \hat{y}_k / \partial h_j$는 식(3.4)를 미분 하여 얻어진다.

$$\sum_j \left[\frac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial h_j} + \frac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial \hat{y}_k}\frac{\partial \hat{y}_k}{\partial h_j}\right] = 0 \tag{19}$$

$$\frac{\partial \hat{y}_k}{\partial h_j} = -\frac{\dfrac{\partial g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial h_j}}{\displaystyle\sum_j \frac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial \hat{y}_k}} \tag{20}$$

그러므로,

$$\frac{\partial E}{\partial h_j} = -\sum_k e(Y_k - y_k)\frac{\partial (\hat{y}_k)}{\partial \hat{y}_k}\frac{\dfrac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial h_j}}{\displaystyle\sum_j \frac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial \hat{y}_k}} \tag{21}$$

식(21)은 은닉층 값의 변화에 대한 Error의 변화를 알수 있게 해주며, $\partial E/\partial h_j$의 값은 출력측의 Error가 역전파된 것으로 생각할 수 있다. 즉, 이 값을 두번째 Layer학습 시의 $\partial E/\partial y_k$와 같이 생각하면 식(14)~식(17)과 같은 방법을 사용하여,

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial h_j}\frac{\partial h_j}{\partial w_{ij}} \tag{22}$$

$$\frac{\partial h_j}{\partial w_{ij}} = -\frac{\partial f\left(\hat{h}_j\right)}{\partial \hat{h}_j}\frac{\dfrac{g\left(x_i, w_{ij}, \hat{h}_j\right)}{\partial w_{ik}}}{\displaystyle\sum_i \frac{g\left(x_i, w_{ij}, \hat{h}_j\right)}{\partial \hat{h}_j}} \tag{23}$$

식(23)과 식(21)을 식(22)에 대입하면,

$$\Delta w_{jk} = \sum_k 2\eta(Y_k - y_k)\frac{\partial f(\hat{y}_k)}{\partial \hat{y}_k}\frac{\dfrac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial w_{jk}}}{\displaystyle\sum_j \frac{g\left(h_j, w_{jk}, \hat{y}_k\right)}{\partial \hat{y}_k}}\frac{\partial f\left(\hat{h}_j\right)}{\partial \hat{h}_j}\frac{\dfrac{g\left(x_i, w_{ij}, \hat{h}_j\right)}{\partial w_{ij}}}{\displaystyle\sum_i \frac{g\left(x_i, w_{ij}, \hat{h}_j\right)}{\partial \hat{h}_j}} \tag{24}$$

식(24)와 같이 $\triangle w_{ij}$를 구할 수 있고 이를 이용하여 첫번째 Layer의 시냅스를 학습시킬 수 있다.

Ⅳ. 시뮬레이션에 의한 학습 결과

위와 같은 학습 Algorithm을 적용하여 회로를 Simulation에 의해 학습시키기 위해 다음과 같은 두 가지의 Look up table을 준비하였다.

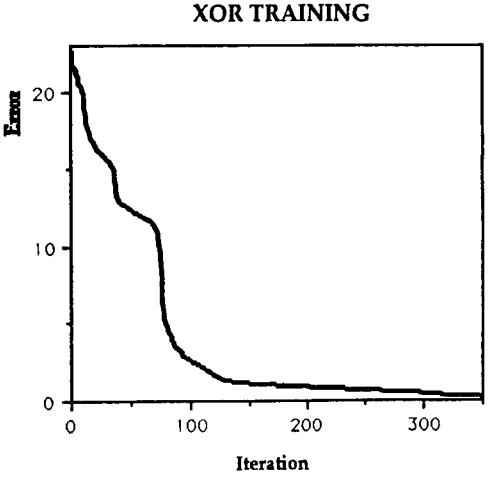


그림 6. XOR Pattern의 학습 곡선

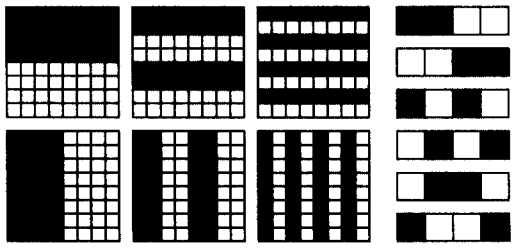


그림 7. 학습에 사용한 Othogonal Pattern
(a) Input Pattern (b) Output Pattern

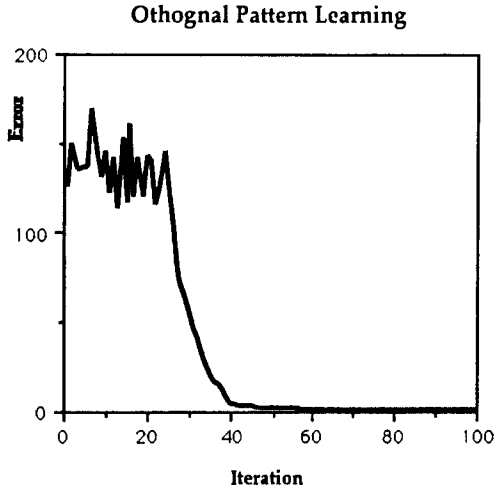


그림 8. Othogonal Pattern의 학습곡선

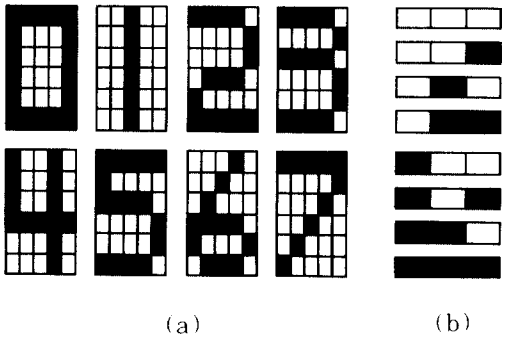


그림 9. 학습에 사용한 숫자모양 패턴
(a) Input Pattern
(b) Output Pattern

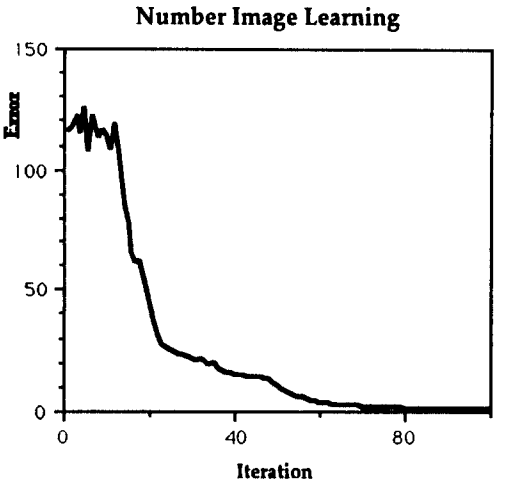


그림 10. 숫자 학습곡선

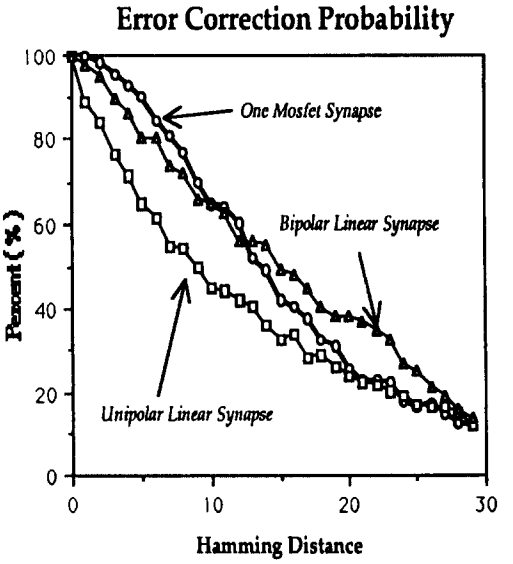


그림 11. Hamming Distance 증가에 따른 인식 률

$f(x)$: Sigmoid 함수를 나타내는 Table이며 V장에서 설명할 시그모이드 회로의 SPICE 시뮬레이션 결과를 이용하여 만들어 졌다.

$g(x, w, y)$: 시냅스로 사용한 W/L=3μm/75μm의 P-MOS에서 Drain, Gate, Source 전압이 각각 x, w, y 일때의 전류값을 역시 SPICE 시뮬레이션을 이용해 얻었다.

이 학습 방법을 이용해 XOR 문제를 학습시킨 결과 그림 6과 같은 모양으로 Error가 감소하며 학습

이 되었다. XOR 문제는 Linearly Separable하지 않은 대표적인 경우로 신경회로망의 성능 Test시 가장 많이 사용되는 학습 패턴이다. 이 회로의 경우 학습률(Learning Rate)를 높여주면 처음에는 Error가 진동하는 듯한 Curve를 그리지만 보다 빠른 속도로 학습을 시킬 수 있었다. 그림 7~그림 10은 학습에 사용한 Othogonal 패턴과 숫자모양의 패턴에 대해 각각의 학습 곡선을 보여준다.

이 회로의 성능을 일반적인 선형 퍼셉트론과 비교하기 위해 그림 9와 같은 숫자 모양의 패턴을 Unipolar 시냅스를 갖는 선형 퍼셉트론과 Bipolar 시냅스를 갖는 퍼셉트론에 각각 학습시켰다. Original Input에 Error 를 첨가해가며 Error의 Hamming Distance 값마다 각각의 패턴에 대해 125번씩의 테스트를 하여 올바른 출력을 내는 경우를 확률로 나타내어 보았다. (그림 11)

Simulation에서는 화소(畵素)의 갯수의 두배 만큼의 Input 뉴런을 두어 각각 정상 Image와 반전 Image를 가하였고, 각 Layer마다 Adaptive Threshold를 두었다. 즉, 숫자 모양 패턴의 경우 60개의 Input 뉴런이 사용되었다. 위의 결과로 미루어 볼때 OMS회로의 비선형 성에 의한 성능의 감소는 나타나지 않았고 OMS회로 자체가 Unipolar한 시냅스를 갖고 있으나 Bipolar Perceptron에 뒤지지 않는 성능을 보였다.

V. 칩의 제작

OMS회로를 테스트하기 위해 0.5μm의 CMOS 공정으로 3 by 3mm^2 크기의 칩을 설계하였다. 이 칩은 10개의 Input, 10개의 시냅스를 갖는 뉴런이 5개 구현되어 있으며 각 뉴런들은 Asynchronous하게 동작한다. 그림 12에서 중앙부분에 규칙적으로 배열된 회로들이 Basic Cell로 Analog Storage와 시냅스로 구성되는데 대부분의 면적을 Analog Storage가 차지한다. 이 칩은 테스트 목적이므로 아날로그 Storage를 일단 설계하기 쉬운 Static RAM (Random Access Memory)과 D/A(Digital to Analog) Converter를 이용하여 구현하였다. 그러나 상용화 목적으로 칩을 제작한다면 EEPROM이나 Capacitor 등을 이용하면 Analog Storage가 차지하는 면적을 1/10이상 줄일 수 있을 것이다. 그림 12에서 Basic Cell Array의 왼쪽에 세로로 배열된 작은 회로들은 Weight의 Address를 Decoding 해주는 회로이고, Basic Cell Array의 오른쪽에 세로로 배열된 회로들은 Sigmoid 회로이다.

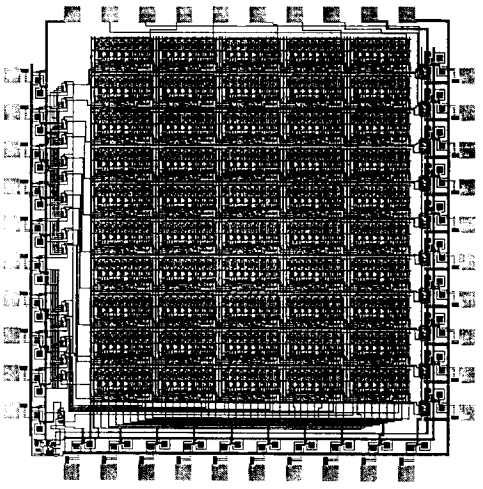


그림 12. 전체 칩의 Layout

(a) Sigmoid 회로

Sigmoid 회로는 뉴런의 Net Input 값을 Saturation시켜서 내보내는 역할을 하며, 다음 층의 시냅스를 구동하기 위한 Buffer 역할도 동시에 할 수 있도록 설계되었다(그림 13). 기본적으로 이 회로는 CMOS Inverter를 저항을 이용해 입출력 특성곡선의 기울기를 완만하게 바꾸어 준 것이며, 왼쪽의 두 MOSFET가 Inverter 형태로 구성되어 있고, 나머지 두 MOSFET가 저항 역할을 한다. 이상적인 버퍼는 부하가 많이 걸렸을 때와 적게 걸릴 때 출력값의 변화가 없어야 한다. 그림 14는 이 시그모이드 회로

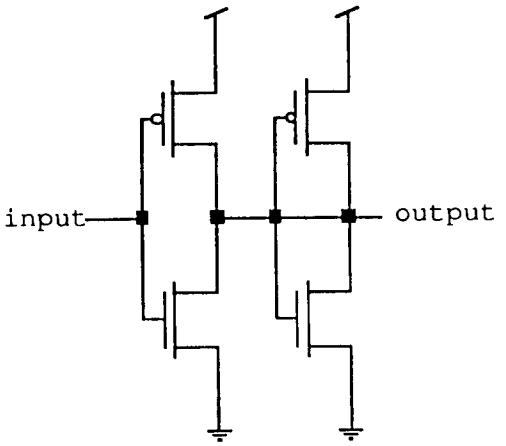


그림 13. Sigmoid 회로

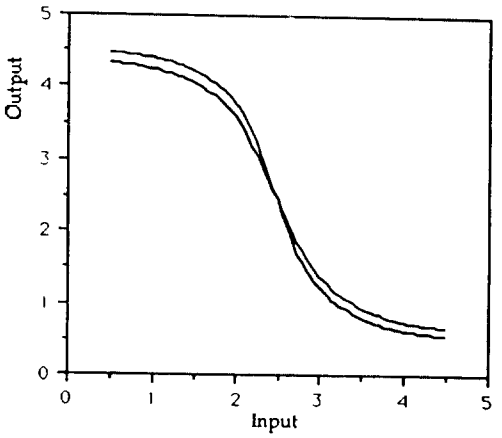


그림 14. Sigmoid 회로의 출력

에 연결된 10개의 시냅스들이 가장 전류를 적게 흘릴 때와 가장 많이 흘릴 때 두가지에 대해서 전압 출력 특성을 나타낸다. 그림에서 보듯이 이 회로는 전류의 변화에 의한 전압 변동이 나타나나 뉴런의 출력값을 크게 바꿀 수 없을 정도로 작다. 이 칩에서 시그모이드 회로는 뉴런의 확장성을 구현하기 위해 칩의 Input 핀에 배치하였다. 칩을 다층구조로 연결하면 앞 층의 모든 출력은 시그모이드를 통해 다음 단에 들어가게 된다.

(b) 계산속도

OMS회로의 장점중의 하나는 회로가 간단하므로 계산 시간이 적게 든다는 점이다. 다른 회로의 경우 계산과정이 여러 단계에 걸쳐서 이루어지므로 그만큼 계산시간이 많이 걸린다. 그림 15는 테스트 칩과 같이 10개의 시냅스를 갖는 뉴런이 동작 하는데 걸리는 시간을 Simulation한 결과이다. 그림에서 입력이 가해진 후 뉴런의 출력은 약 150ns 후에 정상 상태에

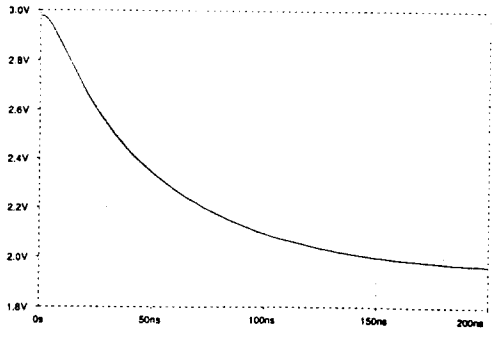


그림 15. OMS회로의 Transition Analysis

도달했으며 1초에 약 6.6×10^6번 동작할 수 있다. 이 칩은 테스트 목적이으로 불과 50개의 시냅스만을 갖고 있으나, 동작 속도가 빠르므로 약 330M(interconnection/sec)의 속도로 계산할 수 있다.

(c) 뉴런의 확장

한 칩에서 구현할 수 있는 Network의 크기보다 큰 Network이 요구되는 Application에서는 Board 상에 여러개의 칩을 꽂아 뉴런을 확장할 수 있어야 한다. OMS회로에서는 전압 형식으로 덧셈이 이루어지는데 이는 시냅스들의 단순한 연결로(wiring) 구현이 되어 있다. 따라서 이 부분을 칩 외부로 내보내어 다른 칩과 연결하게 되면 다른 칩의 시냅스들도 같은 노드에 모두 연결되어 있는 형태가 되므로 연결된 모든 시냅스들의 출력이 더해지게 되는 것이다. 또 Layer의 수를 늘리기 위해서는 칩의 출력을 다른 칩의 입력으로 사용 하면 된다.

Sigmoid 회로는 칩의 입력 부분에 연결되어 있고, 칩의 입력이 Sigmoid회로를 거쳐서 시냅스로 가는 경우와, Sigmoid 회로를 거치지 않고 직접 시냅스로 가는 경우를 외부에서 선택할 수 있게 되어 있다. 그러므로 칩의 입력을 Input Neuron으로 사용하기 위해서는 Sigmoid 회로를 bypass 시키고, Hidden Neuron으로 사용하기 위해서는 Sigmoid 회로를 통과시킨다. 따라서 칩과 칩 간에 전달되는 신호는 뉴런의 값이 아니라 Net Input이다. 이와같이 Sigmoid 회로를 입력핀 쪽에 배치한 이유는 칩 외부로 나오는 신호가 비선형 함수를 거친 후의 결과이면 다른 칩의 시냅스 값을 Linear하게 합할 수 없기 때문이다.

5개의 시냅스를 갖는 5개의 뉴런이 구현된 칩의 경우를 예를들면 이 칩을 이용하여 10개의 Input Neuron과 5개의 Hidden Neuron, 5개의 Output Neuron을 갖는 2 Layer 구조의 신경회로망으로 꾸미기 위해서는 그림 16과 같이 연결하면 된다. 즉 칩을 병렬로 연결하면 뉴런을 확장할 수 있고, 직렬로 연결하면 Layer를 확장 시킬 수 있다.

이 칩들을 이용하여 큰 Network을 구성 한다면 칩은 Array형태로 배열되어질 것이다. 신경회로망 컴퓨터를 구현하기 위해서는 Host Computer에서 학습시킨 Weight를 Down Load 받아서 저장해 두고, 입력을 받아서 계산을 한 후 다시 출력을 Host Computer로 보낸다. 학습 방법은 Host Computer에서 Simulation에 의해 칩의 출력을 계산하여 완전히 학습된 Weight를 Down Load 하는 방법과 칩의 출력을 학습에 이용하는 방법(Chip in Loop

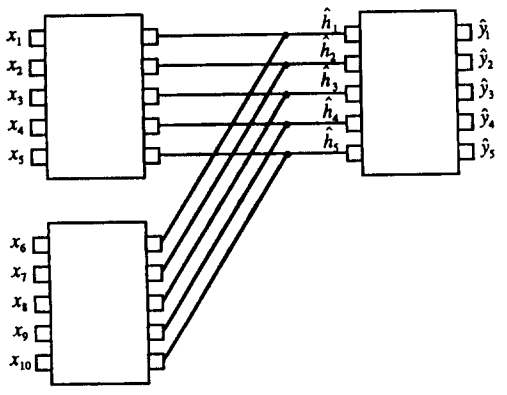


그림 16. 칩단위 뉴런 확장

Learning)이 있는데, 이 회로의 경우 시냅스들의 비선형성으로 인해 Simulation의 계산량이 많으므로 신호의 순방향 전파를 Chip 을 이용하여 수행하고 이 때의 칩의 출력을 학습에 이용하는 Chip in Loop Learning을 하는 것이 바람직하다.

V. 결론

대규모의 신경회로망 시스템을 구현하기 위해서 대규모 집적이 편리하면서 칩단위로 뉴런을 확장할 수 있는 OMS회로를 고안하였다. 기존의 신경회로망 하드웨어가 특정한 수식을 만족하는 뉴런을 구현하기 위해 설계된 것과는 달리 OMS회로는 하드웨어를 설계한 후 하드웨어 고유의 특성에 맞게 출력값을 정의하는 수식과 학습방법을 고쳐 주었다. 이회로의 시냅스는 하드웨어 고유의 비선형적인 동작을 하지만 원하는 출력을 낼 수 있게끔 학습이 가능했다. 이 회로는 단 하나의 MOSFET로써 시냅스가 구현되었으며 기존의 회로에 비해 매우 간단하게 설계되어 있으므로 대규모로 집적시키기에 매우 편리하다. 또한 회로가 간단하므로 연산하는데 드는 시간이 상대적으로 적게 걸린다. 그리고, 칩 단위로 뉴런을 확장시킬 수 있게 되어 있으므로 한 칩에 구현할 수 있는 Network Size 보다 큰 시스템을 구현할 수가 있다.

이러한 회로 구현 방법을 테스트하기 위해 OMS회로의 칩을 설계하였다. 이 칩은 10개의 시냅스를 갖는 5개의 뉴런이 구현되어 있으며 칩단위 뉴런확장을 고려해 Sigmoid 회로를 칩의 Input 쪽에 배치하고 이를 Bypass할 수 있도록 하였다. Simulation결과 뉴런이 한번 동작하는데 걸리는 시간은 150ns 정도로 50개의 시냅스만으로 약 330M CPS의 계산 속도를 낼 수 있는 것으로 나타났다.

하드웨어 고유의 특성에 맞도록 출력값의 계산방법과 학습방법(Error Back Propagation)을 변형하여 학습시킨 결과 XOR 문제는 물론 Othogonal 패턴과 숫자 모양 패턴 등을 학습 시킬 수 있었으며, 기존의 회로에 비해 성능이 떨어지지 않음을 알 수가 있었다. 하지만 회로의 출력이 간단한 수식으로 정의 되는 것이 아니라 비선형 함수의 해를 수치해석적으로 구해야 하므로 학습시키는데 필요한 계산량이 많아지는 단점이 있었다. 만약에 Chip in Loop Learning을 할 수 있도록 시스템이 꾸며진다면 학습시키는데 필요한 계산량은 줄어들 수 있을 것이다. 또한 비선형성에 의한 Local Minimum의 문제가 재기될 수 있으며 이에 대한 Analysis가 좀 더 이루어 져야 겠다.

參考文獻

[1] Carver Mead, "Analog VLSI and Neural Systems", Addison-Wealsey, 1989.

[2] Intel Corporation, "Electically Trainable Analog Neural Networks", May 1990.

[3] R. M. Goodman and T. D. Chiueh, "VLSI Implementaion of Neural Associative Memory and Its Application to Vecter Quantiztion", Int'l. Conf. Neural Networks, Paris, 1990, pp. 635-638.

著 者 紹 介

朴 晶 培(正會員)

1968年 3月 1日生. 1990年 2月 한양대학교 전자통신공학과 졸업(학사). 1993年 2月 한국과학기술원 전기및 전자공학과 졸업(석사). 주관심 분야는 신경회로망의 VLSI구현 등임.

李 壽 永(正會員)

1952年 1月 15日生. 1975年 서울대학교 전자공학과 학사과정 졸업. 1977年 한국과학기술원 전기및전자공학과 석사과정졸업. 1984年 미국 Polytechnic Institute of New York 박사학위취득. 1977年 ~ 1980年 대한엔지니어링 주식회사 근무. 1982年 ~ 1985年 미국 General Physics Corp 근무. 1986年 ~ 현재 한국과학기술원 전기및전자공학과 근무. 부교수. 주관심 분야는 신경회로망, 수치해석 등임.

崔 倫 競(正會員)

1967年 10月 6日生. 1990年 2月 고려대학교 전자전산공학과 졸업(학사). 1992年 2月 한국과학기술원 전기및 전자공학과 졸업(석사). 1992年 ~ 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중. 주관심 분야는 신경회로망의 VLSI 구현 등임.