

# 새로운 Preceding Layer Driven MLP 신경회로망의 학습 모델과 그 응용

## (Learning Model and Application of New Preceding Layer Driven MLP Neural Network)

韓 孝 鎮\*, 金 董 勳\*, 鄭 鎬 宣\*\*

(Hyo Jin Han, Dong Hoon Kim, and Ho Sun Chung)

### 要 約

본 논문에서는 PLD(preceding layer driven) MLP (multi layer perceptron) 신경 회로망의 모델과 학습 알고리즘 그리고 칩 구현을 위한 방법을 제안한다. 제안한 학습 알고리즘은 기존의 방법과 다른 층을 단위로 학습을 하며 변화하는 가중치의 값이 정수이고 활성 함수로 계단 함수를 사용하였다. PLDMLP를 학습하고 학습한 결과를 칩으로 구현하기 위해 은닉층이 확장될 수 있다. 학습은 변화하는 가중치 값이 정수값을 가지기 때문에 각 층을 단위로 쉽게 할 수 있다. PLDMLP의 신경회로망 학습은 교차 학습을 갖는다.

### Abstract

In this paper, the novel PLD (Preceding Layer Driven) MLP (Multi Layer Perceptron) neural network model and its learning algorithm is described. This learning algorithm is different from the conventional. This integer weights and hard limit function are used for synaptic weight values and activation function, respectively. The entire learning process is performed by layer-by-layer method. the number of layers can be varied with difficulty of training data. Since the synaptic weight values are integers, the synapse circuit can be easily implemented with CMOS. PLD MLP neural network was applied to English Characters, arbitrary waveform generation and spiral problem.

### I. 서 론

최근 활발히 연구되고 있는 신경 회로망은 응용 분야가 넓고, 기존의 이론으로 구현하기 힘든 여러가지 기능을 갖는 문제의 해결을 수학적인 모델이나 구체적인 알고리즘 없이도 그 기능을 실현할 수 있

다. 이러한 것들은 인간의 뇌가 대량의 정보 처리를 효율적으로 처리할 수 있고 학습의 기능을 가진다는 것을 근거로 하여 제안되었다. 그런데 이러한 신경회로망을 실제 문제에 적용하고 충분히 이용하기 위해서는 칩으로 구현하는 것이 반드시 필요하다. 칩 구현에서는 컴퓨터 프로그램에 의한 소프트웨어로 시뮬레이션할 때와는 달리 많은 제약이 따른다. 칩 구현은 현재의 VLSI 기술에 의존해야 하는데, 이때 신경회로망에서 가장 중요한 가중치의 값과 뉴런의 활성화 함수가 소프트웨어 시뮬레이션에서처럼 자유롭지 못하고, 소

\*準會員, \*\*正會員, 慶北大學校 電子工學科  
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)  
接受日次: 1990年 12月 26日

프트웨어에서는 부동 소수점을 사용한 실수연산을 처리할 수 있고 연결과 노드의 수를 많이할 수 있지만, 이것을 VLSI로 구현하기에는 여러가지 문제점이 있다.

연산 회로를 이용하여 부동 소수점(floating point)의 값을 계산하도록 하면 소프트웨어 시뮬레이션된 가중치 값을 그대로 이용할 수도 있지만 이것은 소프트웨어 시뮬레이션의 계산기능을 칩으로 하려면 회로가 상당히 복잡하게 되며 계산 속도가 상당히 늦어지게 된다. 시냅스의 가중치가 정수값을 가질 경우 각각의 가중치는 CMOS의 컨덕턴스 크기를 조절함으로써 CMOS 회로로 구현하기가 용이하다.

본 연구에서는 정수 가중치 값(integer weight value)과 계단 제한 활성화(hard limit activation) 함수를 가지는 preceding layer driven MLP(multi layer perceptron) 신경회로망 알고리즘과 VLSI 구현에 대해서 연구하였다. 가중치 값들은 가능한 작은 정수 값으로 제한하고 계단 활성화 함수와 은닉층의 값을 예측하는 새로운 학습 알고리즘으로 회로망을 학습하였다. 제안된 Preceding layer driven 다층 퍼셉트론 신경 회로망은 CMOS 기술로써 칩 구현이 용이하도록 가중치의 값을 정수로 하고 뉴런의 활성화 함수는 계단함수를 사용하며 단층 퍼셉트론의 학습에 기초하여 은닉층의 수가 고정되어 있지 않고 학습의 난이도에 따라 은닉층이 증가하는 새로운 학습 알고리즘으로 학습되었다. 선형 입력에 의해 두 영역으로 분할하는 나선형(spiral) 문제, 사인(sine) 파형, 영문자 인식과 원하는 임의의 파형을 발생하는 신경회로망을 설계하였다.

## II. PLD다층구조 퍼셉트론

### 1. 회로망의 구성

일반적인 다층 구조 신경회로망(multilayer neural network)과는 다른 Lang-Witbrock의 신경 회로망을 그림1에서 보여준다. 이 모델은 두개의 나선형 문제를 해결하기 위해 제안되어 졌다. 학습은 back propagation 알고리즘이 사용되고, 입력은 아날로그 값이며 출력은 이진수의 디지털 값을 갖는다. 그들의 회로망은 두개의 노드를 갖는 입력층과 다섯개의 노드를 갖는 두개의 은닉층, 그리고 한개의 노드를 갖는 출력층으로 구성되어 있다.

그림2는 Lang-Witbrock 신경회로망 모델의 개념을 이용한 보다 확장된 Preceding layer driven MLP 모델을 보여주며, 이 그림은 명확함을 위해 단순화 하였다. 이 구조의 회로망에서 Lang-Witbrock 모델과 다른점은 입출력이 아날로그 값이 아니

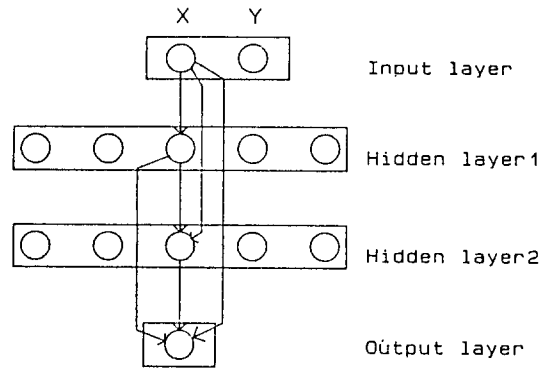


그림 1. 두개의 나선형 문제를 해결하기 위해 사용한 Lang-Witbrock의 회로망

Fig. 1. Lang and witbrock's network for learning the two-spirals problem.

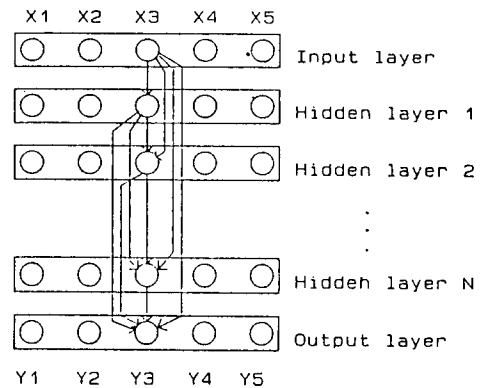
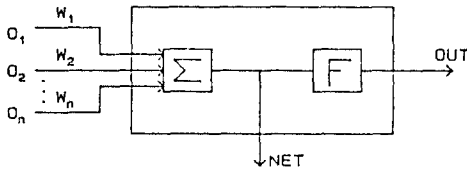


그림 2. 제안된 Preceding Layer Driven MLP회로망  
Fig. 2. Proposed preceding layer driven MLP network.

고 이진수 디지털 값을 갖으며 은닉층의 수가 정해져 있지 않고 은닉층의 노드수는 출력층의 노드수와 같다는 것이다. 은닉층의 출력을 원하는 출력으로 보고 항상 학습은 single layer에 기초하여 학습을 하고 은닉층의 수는 학습의 난이도에 따라 증가하게 되어 로컬 미니멈에 빠지는 경우가 거의 없고 학습 속도도 매우 빠르다. 이 회로망은 입력(input), 은닉(hidden) 그리고 출력(output) 층의 세 부분으로 구성되어지고, 은닉층의 수는 완전한 학습을 하기 위해 증가되어 질 수 있다. 새로운 알고리즘에 의하여 은닉층의 출력이 원하는 출력과 같다고 보고 학습을 하여 첫번

층에서 학습이 완전히 되지 않으면 다음 층으로 가서 전단의 입력과 출력을 입력으로 받아 학습이 완전히 끝날때까지 계속 학습을 하게 됨으로 은닉층의 수가 제한되어 있지 않고 가중치의 값을 정수로 하고 뉴런의 활성화 함수를 계단함수를 사용하는 새로운 학습 알고리즘에 따라 늘어나게 된다. 즉 은닉층의 층수와 입력 노드수는 학습 알고리즘에 따라 학습의 정도에 따라 증가하게 된다.

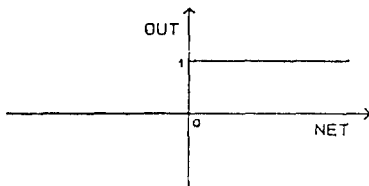
Preceding Layer Driven MLP 모델에서 은닉과 출력층 노드의 기본적 블록(block) 구성을 그림 3에서 보여준다. 들어오는 입력을 입력 노드에서는 더하지 않고, 은닉 또는 출력층 노드에서 더한다. NET는 원 입력과 앞단의 출력을 각각의 연결 가중치와 곱해서 합한다. OUT은 계단특성을 갖는 활성화 함수의 입력으로 NET의 값을 취한다. 시그모이드(Sigmoid) 활성화 함수를 변화하는 가중치가 정수인 경우는 적합하지 못하다.



$$NET = O_1W_1 + O_2W_2 + \dots + O_nW_n = \sum_{i=1}^n O_iW_i$$

$$OUT = F(NET)$$

그림 3. 활성화 함수를 갖는 인공 뉴런  
Fig. 3. Artificial neuron with activation function.



$$OUT = F(NET)$$

$$= 1/2 (|NET|/NET + 1),$$

if  $NET > 0$ ,  $OUT$  is 1,  
else,  $OUT$  is 0.

그림 4. 계단 특성 함수  
Fig. 4. A step activation function.

그림4는 학습을 위한 활성화 함수로 적당하다. 그것을 수식으로 표현하면 다음 식과 같다.

$$OUT = 1/2 (|NET|/NET + 1)$$

OUT의 결과값은 '0' 또는 '1'의 값을 갖고, NET이 '0'보다 크면 OUT은 '1'이고 그 외는 '0'이다.

2. 신경 회로망의 학습 알고리즘

학습은 우선 한 개의 층을 갖고서 학습한다. 학습하기 전에 입력 벡터(vector)와 그것의 각각에 해당하고 출력으로 원하는 목표(target) 벡터를 서로 해당하는 것과 각각 쌍을 짓는다. 이 쌍을 학습쌍(training pair)이라 부른다. 서로 다른 층간의 노드 연결은 '1'과 '0'의 두 입력 모두에 영향을 받는다. 이것은 실질적으로 하나의 시냅스에 '1'과 '0'에 대한 두개의 가중치 값이 있다는 것을 의미한다. 그리고 모든 가중치는 '1'의 값으로 초기화를 한다. 이것은 모든 가중치가 동일한 값을 의미한다. PLD-MLP의 회로망을 학습하는 것은 다음 과정을 따른다.

- step 1. Training set으로 부터 다음 training pair를 선택한다. 이것을 단지 한 층을 갖는 즉 입력과 출력 노드만 있는 회로망의 입력에 적용한다.
- step 2. 각 노드에서 입력의 가중치 합을 구하고 활성화 함수에 의해 그 회로망의 출력값을 계산한다.

$$NET_j = \sum X_i W_{ij} \tag{1}$$

$NET_j$  = 노드 j의 net 출력 값

$X_i$  = i번째 입력

$W_{ij}$  = 노드 i에서 j로 연결될 시냅스의 가중치

$OUT_j = 1$   $NET_j$ 가 0보다 크거나 같을 때

$OUT_j = 0$  그 외의 경우

$OUT_j$  = 노드 j의 출력값

- step 3. 회로의 출력과 training pair중 그 출력에 해당하는 target 벡터간의 오차(error)를 계산한다.

$$error_j = target_j - OUT_j \tag{2}$$

- step 4. 오차를 최소화하는 방향으로 회로망의 가중치를 조정한다.

$$W_{ij}(t+1) = W_{ij}(t) + error_j \tag{3}$$

$W_{ij}(t)$  = 시간 t에서 노드 i에서 j로 연결되는 시냅스의 가중치

step5. 위 step1.에서 step 4.까지의 과정을 모든 입력 집합의 오차가 '0'이거나 학습의 횟수가 학습이전에 받아들인 제한 값과 같을 때까지 반복한다.

step6. step 5과정 후 전자에 의해 그 과정이 끝나면 학습은 완전히 끝난다. 그러나, 후자의 경우는 하나의 층을 더 증가시켜 step 1에서 step 5를 계속해서 반복한다.

위의 step 1과 2에서 요구하는 연산은 학습이 끝난 후 궁극적으로 사용되어질 방법과 같다. 즉, 입력 벡터(input vector)가 회로에 적용되고 결과 출력은 차례로 계산되어진다. step 3에서는 그림3에 있는 OUT이라 명명된 회로망 출력은 각각 오차를 찾기 위해 목표 벡터의 일치하는 성분과 뺄셈을 한다. 이 오차는 step 4에서 회로망의 가중치를 조정하기 위해 사용되어진다. 가중치의 극성과 크기는 학습 알고리즘에 의해 결정된다. 네 과정의 제한된 반복후 실제 출력 값과 목표 값과의 차가 '0'이면 학습이 완전히 끝이 난다. 오차의 값이 '0'가 아니면 회로망은 한층 더 증가되어진다. 회로망은 오차의 값이 '0'가 되기 위해 다시 학습된다. 오차가 '0'라는 것은 학습이 된것을 의미한다. 이 시점에서 회로망은 인식을 위해 사용되어지고 가중치는 변화되지 않는다. step 1과 step 2는 forward path를 구성하고, 신호는 회로망의 입력층에서 부터 은닉층과 출력층의 조정된 가중치를 통해 진행한다. 목표값(value)은 각 층의 뉴런 각각에 유효하기 때문에 연관된 가중치를 조정하는 것은 향상된 규칙을 사용함으로써 쉽게 이루어진다. 내부층 즉, 은닉층의 가중치의 가중치는 비교를 위한 목표값을 가지기 때문에 쉽게 조절이 된다. 임의의 층에서 뉴런(neuron)의 출력  $\delta$ 는 에러를 구하기 위해 목표값으로 부터 뉴런의 출력 OUT을 빼 주며,

$$\delta = target - OUT.$$

결과  $\delta$ 는 가중치에 더한다. 같은 과정이 은닉층의 뉴런으로부터 출력층의 뉴런으로 진행하여, 각각의 가중치에 수행되어진다.

다음의 식들은 위의 계산을 한다

$$\Delta_w = \delta \times INPUT$$

$$w(n+1) = w(n) + \Delta_w$$

여기에서,

$w(n)$  + step n에서 은닉층이 한 뉴런으로 부터 출력층의 다른 뉴런으로의 가중치값

$w(n+1)$  = step n+1에서의 가중치값

$\delta$  = 출력층에서 한 뉴런에 대한 오차값

INPUT = 은닉층에서 한 뉴런에 대한 입력값

### 3. 신경 회로망의 학습 예제들

이차원 평면에 있는 두개의 나선형(spiral)에 관한 문제를 그림3,5에서 보여준다. 이 문제에서 회로망은 단위 평면에 있는 두개가 서로 꼬인 나선형상의 점들을 구분해야 한다. 원이 표시된 곳은 '1'에 해당하고 표시되지 않은 곳은 '1'에 해당한다. 이 문제는 선형으로 분리할 수가 없다. 단위 평면에 있는 모든 원이 표시된 곳과 표시되지 않는 곳을 두 영역으로 서로 구분하기 위해서 하나의 직선으로 나누는 방법은 없다. 그림에서 보여주는 회로망은 표시 원은 x와 y의 좌표에 표시된다. 입력 유니트의 활동범위는 이전값에 제한되어 '0' 또는 '1'의 값을 갖는다.

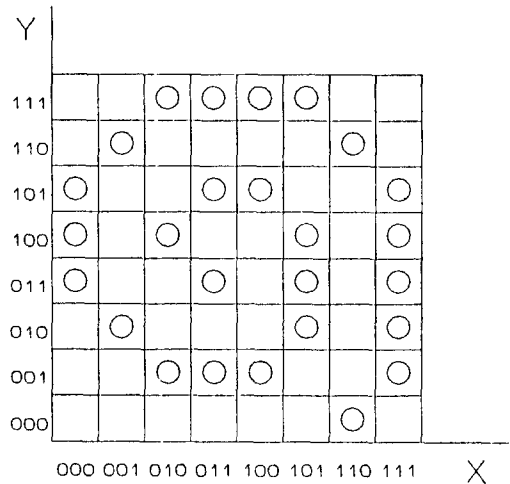


그림 5. 두개의 나선형 문제를 해결하기 위한 학습 데이터

Fig. 5. Training points for the two-spirals problem.

그림6은 학습과정 후에 각 층의 노드 출력이며 그림5의 문제의 해결 과정을 보여준다. 첫번째 층의 노드 출력을 그림으로 나타낸 것을 보면 거의 구별을 못하다가 학습 알고리즘에 따라 층이 증가될수록 원하는 값을 얻는 것을 볼 수 있다. 우측 하단의 마지막 층에서의 출력을 나타낸 그림을 보면 원하는 결과를 얻은 것을 볼 수 있다. 나선형 문제는 아홉개의 층으로 학습된다. 즉 여덟개의 은닉층과 하나의 출력층을 갖고 각각의 층은 여섯개의 유니트가

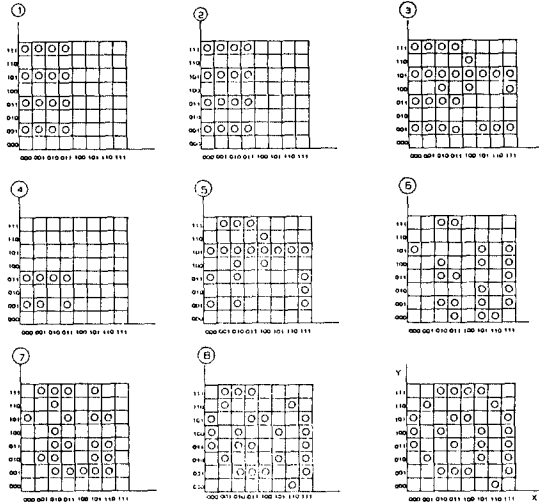


그림 6. 두개의 나선형 회로망에서 unit들의 응답 결과  
 Fig. 6. Response function plots for the units in the two-spirals network.

있다. 각각의 층에서 유닛들은 그 아래의 모든 층의 유닛으로부터 연결되어 있다. 층을 건너 뚫은 연결은 더 낮은 층으로부터 더 직접적인 정보를 준다. 은닉층의 추가는 인의의 신경 회로망이 입력 영역을 임의의 영역으로 효과적인 분할을 하도록 하는 특징을 갖고 있다.

그림7은 사인 파형을 나타내며 이 파형을 지나는

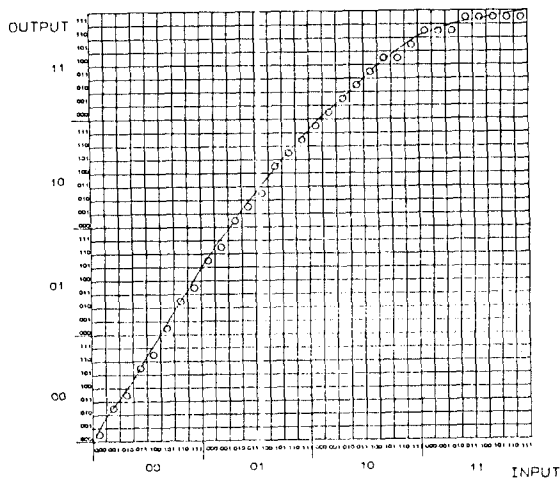


그림 7. 사인 파형의 학습 데이터  
 Fig. 7. Training points for the sine wave from problem.

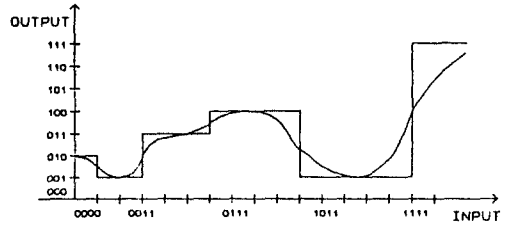


그림 8. 임의 파형의 학습 함수  
 Fig. 8. The arbitrary wave form function for learning.

표 1. 사인 파형 학습 데이터  
 Table 1. Training data for the sine wave form problem.

DEGREE (°)	INPUT	OUTPUT
0	0 0 0 0 0	0 0 0 0 0
3	0 0 0 0 1	0 0 0 1 0
6	0 0 0 1 0	0 0 0 1 1
9	0 0 0 1 1	0 0 1 0 1
12	0 0 1 0 0	0 0 1 1 0
15	0 0 1 0 1	0 1 0 0 0
18	0 0 1 1 0	0 1 0 1 0
21	0 0 1 1 1	0 1 0 1 1
24	0 1 0 0 0	0 1 1 0 1
27	0 1 0 0 1	0 1 1 1 0
30	0 1 0 1 0	1 0 0 0 0
33	0 1 0 1 1	1 0 0 0 1
36	0 1 1 0 0	1 0 0 1 0
39	0 1 1 0 1	1 0 1 0 0
42	0 1 1 1 0	1 0 1 0 1
45	0 1 1 1 1	1 0 1 1 0
48	1 0 0 0 0	1 0 1 1 1
51	1 0 0 0 1	1 1 0 0 0
54	1 0 0 1 0	1 1 0 0 1
57	1 0 0 1 1	1 1 0 1 0
60	1 0 1 0 0	1 1 0 1 1
63	1 0 1 0 1	1 1 1 0 0
66	1 0 1 1 0	1 1 1 0 0
69	1 0 1 1 1	1 1 1 0 1
72	1 1 0 0 0	1 1 1 1 0
75	1 1 0 0 1	1 1 1 1 0
78	1 1 0 1 0	1 1 1 1 0
81	1 1 0 1 1	1 1 1 1 1
84	1 1 1 0 0	1 1 1 1 1
87	1 1 1 0 1	1 1 1 1 1
90	1 1 1 1 0	1 1 1 1 1

작은 영역은 입력에 대한 출력을 나타낸다. 5비트로 90도를 30등분하여 입력의 값은 3도의 간격(step)을 가진다. 출력은 최저 값 '0'에서 최고 값 '1' 사이를 32등분하여 적당한 값에 연결된다. 표1에서 이 관계를 나타내어 준다. 사인 파형은 여덟개의 층으로 해결된다.

선형적인 입력에 임의의 디지털 파형의 출력을 갖는 그래프를 그림8에서 보여준다. 이것은 4비트 입력 데이터에 3비트 출력 데이터를 갖고 입력에 대한 출력을 예상할 수 없는 그래프를 보여준다. 즉 수식으로 그래프를 표현할 수 없다. 회측이 입력이고 종축이 출력이다. 입력과 출력의 비트를 작게한 것은 학습 결과를 제시하기 위한 것이다.

표2는 학습 후 가중치의 결과 값이다. 원하는 결과를 얻기 위해 네개의 층이연결이 필요하다는 것을 알 수 있다. 결과의 가중치값은 정수이며 절대값이 17이하로 작음을 알 수 있다. 그리고, 층이 증가할

때마다 가중치의 노드수가 출력 비트수와 같은 3비트씩 증가함을 알 수 있다.

4. 신경회로망의 VLSI 구현

제안한 알고리즘을 갖고 학습한 후의 결과를 회로로 구현하는데 쓰이는 Preceding Layer Driven MLP 모델로 그림9에서 보여준다.

이 회로는 입력, 출력 그리고 은닉층 유니트로 구성되며, 임의 층의 출력은 다음 층의 출력으로 들어간다. 회로에서 x표는 NMOS, O표는 PMOS를 나타낸다. 유니트(unit)인 뉴런은 인버터(inverter) 두개로 이루어진 버퍼(buffer)로 구성된다. 활성화(activation) 함수를 칩으로 구현할 경우 두개의 인버터를 이용하면 입력에 의해 'ON'되는 PMOS와 NMOS 값의 합이 클 때 출력이 '1'인지 '0'인지 구별하기 어려워진다. 그래서 학습된 가중치의 합이 적어야 한다. 차동 증폭기를 이용하면 두개의 인버터를 이용한 것보다 더 정확한 활성화 함수를 만들 수 있지

표 2. 16개의 입력 데이터와 4개 입력, 3개 출력노드를 갖는 임의 파형의 학습결과

Table 2. The results of learning of an arbitrary wave form with 16inputs and 4 input and 3 output nodes.

layer number	input data	The weight values after good learning												
0	1	- 2	0	- 1	0									
	0	1	0	0	1									
1	1	- 3	1	- 1	0	- 1	- 2	0						
	0	3	1	- 1	0	1	- 3	- 1						
2	1	- 4	2	- 3	0	- 1	2	0	1	- 4	4			
	0	3	3	2	- 1	0	- 3	- 1	- 2	3	- 5			
3	1	0	8	- 6	- 1	0	2	- 2	1	- 4	6	- 9	- 10	- 2
	0	- 2	- 10	4	- 1	- 2	- 4	0	- 3	2	- 8	7	- 8	0

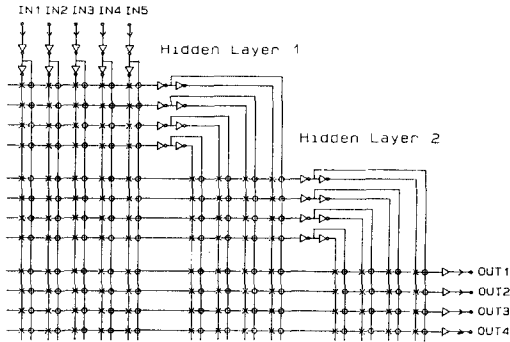


그림 9. Preceding Layer Driven MLP 신경 회로망의 칩 구현 모델

Fig. 9. Preceding layer driven MLP neural network for hardware implementation.

만 회로가 복잡해지고 설계가 어려워지며, 수행 속도가 감소하는 등의 문제점들이 있다. 물론 연산회로를 이용하여 부동 소숫점(floating point)의 값을 계산하도록 하면 소프트웨어 시뮬레이션된 가중치 값을 그대로 이용할 수도 있지만 이것은 소프트웨어 시뮬레이션의 계산기능을 칩으로 하려면 회로가 상당히 복잡하게 되며 계산 속도가 상당히 늦어지게 된다. 시냅스의 가중치가 정수값을 가질 경우 각각의 가중치는 CMOS의 컨덕턴스 크기를 조절함으로써 CMOS 회로로 구현 하기가 용이하다.

표3에서 간단한 XOR(exclusive or)를 예로 들어 회로의 구성을 설명한다. XOR를 학습한 후의 가중치 결과를 표4에서 보여준다. 학습 속도는 기존의 학습 알고리즘의 학습보다 10배정도 빠르다. 두개의 층으로 원하는 결과를 찾을 수 있다는 것을 보여주며, 각 층에서 위의 줄은 '1'의 입력 아래의 줄은 '0'의 입력이 들어올 때 출력에 영향을 주는 가중치이다.

학습된 결과값을 칩으로 구현하는 방법은 다음과 같다. 그림10에서 A의 PMOS는 IN의 값이 '0'이고 가중치 결과 값이 양수이고, B의 NMOS는 IN의 값이 '1'이고 가중치 결과 값이 양수이고, D의

표 3. XOR 학습을 위한 X, Y입력과 출력 데이터  
Table 3. Data of X and Y inputs and outputs for XOR learning.

$Y \oplus X$	0	1
0	0	1
1	1	0

표 4. XOR 학습 후 가중치의 학습 결과값  
Table 4. Results of weight after XOR learning.

layer number	input data	The weight values after good learning			
0	1	0	0		
	0	1	1		
1	1	1	1	2	
	0	-1	-1	-2	

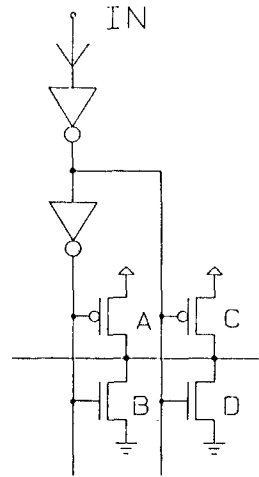


그림10. 시냅스 회로  
Fig. 10. Circuit of synapse.

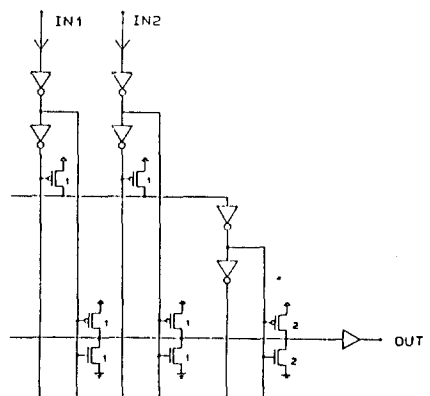


그림11. XOR 학습 결과의 VLSI 구현  
Fig. 11. VLSI implementation of results of XOR learning.

NMOS의 경우는 IN의 값이 '0'이고 가중치 결과 값이 음수이다.

그림11은 XOR를 회로로 구현한 것을 보여준다. 가중치의 결과가 양인것은 PMOS, 음인것은 NMOS 그리고 '0'은 아무런 표시를 하지 않는다. PMOS는 선의 위 부분에 NMOS는 아랫부분에 배치를 한다. 각 층 윗줄에 있는 가중치는 PMOS로 나타내며 인버터 하나를 통과한 선에 연결하고 음의 가중치는 NMOS로 나타내며 인버터 두개를 통과한 선에 연결한다. 아랫줄에 있는 것은 위치를 반대로 하면 된다.

그림12는 표2를 회로로 간략히 나타낸 것이다. 숫자의 값은 가중치이고 MOS의 W/L의 비를 나타낸다. 가중치의 최대값은 17이고 NMOS가 16이다.

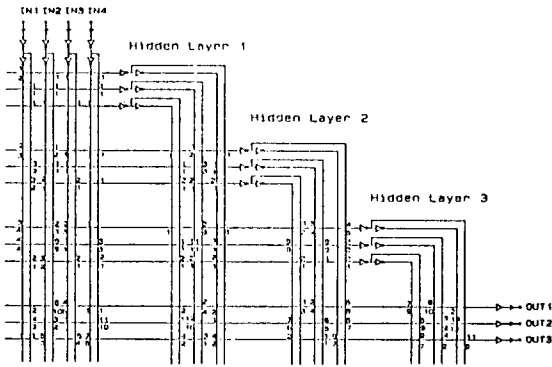


그림12. 임의 파형 함수의 PLD MLP 회로망  
Fig. 12. PLD MLP's network for the arbitrary wave form function.

5. 영문자 인식을 위한 학습 적용

영문자 인식을 위해서는 왜곡된 입력이 들어와도 많은 입력 대신 중요한 문자의 특징을 알아내기 위한 전처리 단계가 필요하다. 전처리 단계에서는 스캐너와 같은 입력 장치로 받아들인 데이터 화일에서 문자 각각에 대한 특징들을 알아내야 한다. 그런 후 받아들인 데이터의 특징과 데이터 베이스에서 각 문자에 대한 특징을 비교하여 문자인식을 한다. 본 연구에서는 문자 간의 특징 비교를 PLD MLP 학습 모델로 학습하여 영문자 인식에 적용하였다.

한 문자가 가지는 특징점은 문자 인식 알고리즘에 따라서 여러 형태로 분류할 수 있다. 학습에는 시간이 많이 걸리는 세선화 과정 대신 Tracing 과정을 사용하여 구한 특징점 데이터를 사용하였다. 한 문자가 가지는 가지점과 4방향의 stroke 갯수를 영문자 분류의 특징점 데이터로 이용했다.

문자를 인식하는데 필요한 특징점은 스캐너의 출력 영상에서 개별 문자로 분리한 후 Tracing 과정을 수행한 문자 데이터에서 추출한다. 가지점에 의한 분류는 가지점은 한 stroke를 tracing하면서 가지점의 후보점이 가지점으로 인정되는 점이다. 4 방향의 stroke 갯수는 한 stroke를 tracing 하면서 가장 많이 나타나는 방향을 그 stroke의 방향으로 한다. 표 3와 3.6은 학습에 필요한 영문 대문자 26자와 소문자 26자에 대한 입·출력 데이터이다. 입력인 가지점의 갯수를 3비트, 4방향 stroke 각각의 갯수를 3비트씩 나타내었고 출력은 각 문자에 해당하는 값을 6비트로 나타내었다.

영문 대문자 26자와 소문자 26자 총 52자에 대한 학습한 결과를 표7에서 보여준다. 원하는 결과를 얻기 위해 4의 층까지 연결이 필요하다는 것을 알 수 있다. 결과의 가중치 값은 정수이며 절대값이 63 이하

표 5. 영문 대문자의 입·출력 데이터  
Table 5. Input and output data for English capital characters.

Char.	BP No.	INPUT				OUTPUT
		STROKE DIR No.				
		1	2	3	4	
A	010	001	010	000	000	000001
B	100	011	010	001	000	000010
C	000	000	000	100	000	000011
D	001	010	100	000	000	000100
E	100	010	011	001	000	000101
F	101	011	000	000	001	000110
G	001	010	000	010	000	000111
H	011	010	100	000	000	001000
I	001	010	000	000	000	001001
J	001	010	011	000	000	001010
K	010	011	010	000	001	001011
L	000	001	000	001	000	001100
M	011	011	010	000	001	001101
N	010	011	001	000	000	001110
O	000	011	000	001	001	001111
P	011	111	000	001	000	010000
Q	101	010	011	100	000	010001
R	100	011	000	001	000	010010
S	000	011	000	001	000	010011
T	001	110	000	000	001	010100
U	001	011	010	000	001	010101
V	001	010	000	001	000	010110
W	011	010	000	010	010	010111
X	010	001	010	000	010	011000
Y	010	011	000	001	010	011001
Z	010	100	000	000	001	011010



표 6. 영문 소문자의 입·출력 데이터  
Table 6. Input and output data for English small characters.

Char.	INPUT					OUTPUT
	BP No.	STROKE DIR No.				
		1	2	3	4	
						0
a	011	011	010	000	000	100001
b	010	010	000	001	000	100010
c	000	100	000	000	000	100011
d	010	110	000	000	000	100100
e	001	101	000	000	000	100101
f	011	101	000	001	000	100110
g	001	000	000	101	000	100111
h	010	111	000	000	000	101000
i	000	010	000	000	001	101001
j	000	001	001	000	000	101010
k	100	110	000	000	000	101011
l	000	010	000	000	000	101100
m	101	101	000	010	000	111101
n	001	011	001	000	000	111110
o	000	000	000	001	001	111111
p	011	111	000	000	000	110000
q	010	011	000	001	000	110001
r	011	101	000	000	000	110010
s	000	000	000	101	000	110011
t	001	011	000	000	000	110100
u	000	001	000	000	001	110101
v	010	110	000	000	011	110110
w	101	010	000	000	110	110111
x	100	000	010	000	011	111000
y	001	111	000	000	000	111001
z	010	011	010	000	000	111010

로 나타났다. 그리고, 층이 증가할 때마다 가중치의 노드 수가 출력 비트수와 같은 6비트씩 증가함을 알 수 있다. 각각의 층에서 200번씩 학습을 반복하여 학습 시간은 선 워크스테이션상에서 8분 23초였다.

III. 결과 및 고찰

앞에서 나선형 문제, 디지털 사인파 그리고 임의의 파형 생성문제와 회로 구현을 위한 XOR를 학습시켜 보았다. 나선형은 아홉개층, 디지털 사인파는 여덟개의 층으로 학습이 이루어졌다. 이것은 가중치값이 부동 소숫점인 일반 학습에 의한 것보다 층 수가 많으나 가중치 값은 침화하기에 좋은 작은 정수로 이루어지며, 학습 시간은 많이 단축된다. XOR 문제의 경우를 예를 들어 비교해 보면 본 논문에서 제안된 새로운 학습 알고리즘으로 학습한 결과 학습이 완전히 될 때까지 걸리는 시간이 66[msec]로 1초가 안걸려 기존의 학습 알고리즘의 학습보다 10배이상 빠르다. 학습 횟수를 더 많이 함에 따라 층수의 갯수가 줄어들었다. 그러나 어떤 한계치에 도달하면 학습 횟수를 증가시켜도 소용없다. 그러나 어떤 한계치에 도달하면 학습 횟수를 증가시켜도 소용없다. 그리고 나선형 문제에서 출력 데이터 노드에 임의의 값을 추가함으로써 학습이 더 잘되는 것을 볼 수 있었다. 하나의 층에서 학습이 끝난 경우도 있고 몇개의 층에서 학습된 경우도 있다. 각각의 층에서 결과를 보면 첫번째 은닉층의 출력값은 어느정도 원하는 출력과 비슷한 출력값을 가진다.

표 7. 학습 결과  
Table 7. Result of learning.

layer number	input data	The weight values after good learning															
0	1	- 2	3	4	- 3	- 2	- 6	-20	- 5	- 5	-10	-12	- 6	1	- 2	-10	
		- 2	2	2	4	9	11	1	- 8	4	13	- 5	5	1	19	9	
		0	- 2	- 3	2	2	0	7	3	10	-19	6	- 3	0	5	- 1	
		- 8	- 6	- 1	- 3	- 1	- 3	- 5	1	- 4	- 6	3	- 2	3	- 5	0	
		1	4	- 2	- 7	- 2	- 3	-17	- 9	- 2	- 1	- 1	- 5	6	- 7	2	
			- 5	- 6	- 2	- 1	2	- 3	- 7	3	- 5	5	5	- 4	8	- 2	- 3
		0	- 1	- 6	- 7	0	- 1	3	17	2	2	7	9	3	- 4	- 1	7
	6		2	2	0	- 5	- 7	3	12	0	- 9	9	- 1	3	-15	- 5	
	1		3	4	- 1	- 1	1	- 6	- 2	- 9	20	- 5	4	1	- 4	2	
	6		4	- 1	1	- 1	1	3	- 3	2	4	- 5	0	- 5	3	- 2	
- 3	- 6		0	5	0	1	15	7	0	- 1	- 1	3	- 8	5	- 4		
		4	5	1	0	- 3	2	6	- 4	4	- 6	- 6	3	- 9	1	2	

표 7. 학습 결과  
Table 7. Result of learning.

1	1	9 10 5 1 -10 -13 . . . -5 16 6 1 4 -16
		25 31 6 19 11 22 . . . 7 15 7 39 16 17
		-9 12 -14 -7 -13 -5 . . . -2 16 3 -2 -11 24
		15 -16 2 0 1 -2 . . . -3 -4 -8 -6 3 -19
		12 11 0 -3 -10 -6 . . . -2 7 5 3 3 3
	-15 -9 -7 -16 9 -7 . . . -4 -28 1 -7 -18 -10	
	0	-13 -14 -9 -5 6 9 . . . 1 -20 -10 -5 -8 12
		-17 -23 2 -11 -3 -14 . . . 1 -7 1 -31 -8 -9
		-10 -13 14 6 12 4 . . . 1 -17 -4 1 10 -25
		13 14 -4 -2 -3 0 . . . 1 2 5 4 -5 17
-13 -12 -1 2 9 5 . . . 1 -8 -6 -4 -4 -4		
13 7 5 14 -11 5 . . . 2 26 -3 5 16 8		
2	1	11 11 1 24 0 -9 . . . 2 -41 -10 -3 24 2
		30 29 3 24 10 21 . . . -2 24 9 14 0 25
		-3 0 -13 -13 -9 6 . . . 12 -20 -3 -4 -9 -45
		-22 -13 15 -15 -1 -12 . . . -51 22 7 14 6 -10
		-4 -13 9 -10 -8 2 . . . 10 9 -5 -20 0 -26
	2 0 -16 -1 5 1 . . . -15 -21 -19 -21 5 21	
	0	-13 -13 -3 -26 -2 7 . . . -4 39 8 1 -26 -4
		-24 -23 3 -18 -4 -15 . . . 8 -18 -3 -8 6 -19
		4 1 -4 14 10 -5 . . . -11 21 4 5 10 46
		20 11 -7 13 -1 10 . . . 49 -24 -9 -16 -8 8
1 10 -12 7 5 -5 . . . -13 -12 2 17 -3 23		
3 5 21 8 0 4 . . . 20 26 24 26 0 -16		
3	1	8 11 11 26 -3 -11 . . . 16 6 4 -15 -15 19
		-16 32 2 14 7 9 . . . 2 14 -19 -11 23 7
		-7 3 -4 13 -20 4 . . . 4 -14 31 -37 17 12
		10 -35 14 0 -14 -5 . . . 14 -1 5 16 -30 -1
		1 7 -3 -17 -9 -15 . . . -17 21 -28 11 48 -16
	-15 19 -4 -8 9 -7 . . . -0 -15 -7 -24 16 46	
	0	-11 -14 -14 -29 0 8 . . . -19 -9 -7 12 12 -22
		-13 -29 1 -11 -4 -6 . . . 1 -11 22 14 -20 -4
		10 0 7 -10 23 -1 . . . -1 17 -28 40 -14 -9
		8 33 -16 -2 12 3 . . . 12 -1 -7 -18 28 -1
-2 -8 2 16 8 14 . . . 16 -22 27 -12 -49 15		
17 -17 6 10 -7 9 . . . 2 17 9 26 -14 -44		
4	1	5 11 5 13 -9 -14 . . . 52 16 5 1 0 -16
		-4 4 -18 -5 9 0 . . . -2 34 -14 -3 -22 0
		2 1 -1 1 -2 -1 . . . -2 1 18 -2 -2 5
		5 -14 18 -6 -20 -7 . . . 4 -4 -7 46 10 -9
		2 0 0 0 0 2 . . . 0 -1 0 -1 1 0
	-2 0 -1 2 -1 0 . . . 1 3 0 0 0 11	
	0	-8 -14 -8 -16 6 11 . . . -55 -19 -8 -4 -3 13
		3 -5 17 4 -10 -1 . . . 1 -24 10 4 20 0
		-1 1 1 -1 0 1 . . . 1 0 -15 1 2 -3
		-2 -10 -15 4 15 5 . . . -5 2 5 -40 -9 10
-1 1 0 0 2 0 . . . 1 2 -1 1 -2 0		
0 2 2 0 2 1 . . . -1 -1 0 0 2 -9		

## IV. 결 론

## 參 考 文 獻

신경 회로망을 기존의 학습 알고리즘으로 시뮬레이션한 결과의 가중치 값이 부동소숫점을 갖기 때문에 현재의 VLSI 기술로 칩 구현 하는데는 어려움이 많다. 본 연구에는 신경회로망의 칩 구현을 목적으로 PLD(preceding layer driven) MLP (multi layer perceptron) 모델로 새로운 학습 알고리즘을 제안하였다. 그래서 회로망 가중치의 절대치가 작은 정수의 값이 되도록 하였고 계단함수를 활성화 함수로 사용하였다. 학습할 training pair가 많으면 은닉층의 갯수가 상당히 늘어나며 가중치의 값이 커지는 문제점이 있다. PLD MLP 신경회로망은 기존의 산술로 회로 즉, 전가산기 또는 반가산기와 같은 덧셈기 및 병렬 입력의 데이터를 계산하는 카운터에 적용할 수 있다. 그리고 문자인식이나 음성인식과 같은 인식에 이용될 수 있다.

- [1] P.D. Wasserman, "Neural Computing Theory and Practice," ANZA Research, Inc. 1989.
- [2] D.S. Touretzky and D.A. Pomerleau, "What's Hidden in the Hidden Layers?" BYTE, pp. 227-233, Aug. 1989.
- [3] 김태경, "신경회로망의 VLSI 구현" 경북 대학교 석사 논문 1988.
- [4] A.N. Michel and D.L. Gray, "Analysis and Synthesis of Neural Networks with Lower Block Traingular Interconnecting Structure," IEEE pp. 2490-2494, 1990.
- [5] 고휘진, "IDMLP 신경회로망을 이용한 영문자 인식 및 트레이닝칩 설계" 경북대학교 석사 논문예정 1990.

## 著 者 紹 介

## 韓 孝 鎮 (準會員)

1989년 2월 경북대학교 전자공학과 졸업. 1991년 2월 경북대학교 전자공학과 석사학위 취득. 1991년 3월~현재 삼성전관 근무중. 주관심분야는 신경회로망의 VLSI구현, CAD tool개발 등임.

## 金 董 勳 (準會員)

1990년 2월 경북대학교 전자공학과 졸업. 1990년 3월~현재 경북대학교 전자공학과 석사과정. 주관심분야는 신경회로망 및 퍼지이론의 응용 및 VLSI 구현, X WINDOW SYSTEM S/W 및 CAD tool 개발 등임.

## 鄭 鎬 宣 (正會員) 第28卷 B編 第9號 參照

현재 경북대학교 전자공학과 교수