

論文 91-28B-10-5

## IDMLP를 이용한 한자인식에 관한 연구

## (A Study for the Chinese Character Recognition Using IDMLP)

呂 珍 璟\*, 李 宇 一\*, 鄭 鎬 宣\*

(Jin Kyung Ryeu, Wu Il Lee, and Ho Sun Chung)

## 要 約

인쇄체 한자로부터 특징들을 추출하고 이를 Input Driven Multi-Layer Perceptron (IDMLP) 신경회로망으로 학습하여 한자를 인식하고자 하였다. 또한 그 학습 결과의 회로 구현에 대해 설명하였다. 입력 노드의 수를 하드웨어 구현에 알맞게 하기 위해 한자로부터 특징점 정보와 stroke 정보들을 추출하여 이를 학습을 위한 입력으로 사용하였으며, 각 synapse의 weight는 정수의 형태가 되도록 학습하였다. 그러므로 이러한 결과를 CMOS 회로로 설계하는 것이 가능하다.

## Abstract

A learning method for the recognition of printed Chinese character by using the input driven multi-layer perceptron model was proposed and the circuit representing the learning result was designed. In learning, the extracted features from Chinese characters are used as inputs and the synapse's weight is integer value. So it is possible to implement the learning result with CMOS circuit.

## I. 서 론

한자 인식을 위해 여러가지 방향으로 많은 연구<sup>1)-5)</sup>가 진행되고 있다. 그 중에서도 error back propagation,<sup>6)</sup> Boltzmann machine,<sup>7)</sup> Neocognitron<sup>8)</sup> 등 신경회로망의 여러 모델을 사용하여, 학습을 통해 문자를 인식하고자 하는 연구가 활발히 진행되어 왔다. 실제로 수백자의 한자를 인식하는 소프트웨어 시스템도 개발되어 있다. 하지만 이러한 시스템은 소프트웨어의 한계를 벗어나지 못하므로, 인식 시스템을 one chip으로 제작하는 것이 필요하게 되었다. 그러나 이러한 chip 제작에 있어서는 여러가지 문제점이 있다. 그 하나는, 일정한 크기의 문자에 대해서 모든

화소의 정보를 입력으로 사용한다면 입력의 크기가 엄청나게 커져서 실제 회로의 입력으로는 부적합하다는 것이다. 둘째는, 이미 존재하는 많은 학습 알고리즘에 의해 결정된 synapse의 weight는 정수 값이 아니기 때문에 이를 정확하게 구현할 수 없다는 것이다.

그러므로 본 연구에서는 인쇄체 한자로부터 적절한 특징을 추출하여 이를 입력으로 사용하고, input driven multi-layer perceptron (IDMLP) 모델에 대해 새로운 학습 알고리즘을 사용하여 synapse의 weight가 정수값을 갖는 한자 인식회로를 설계하였다.

## II. 신경회로망 모델

본 연구에서 사용한 신경회로망의 형태를 간략하게 나타내면 그림1과 같다. 이 IDMLP 모델은, 기존의 multilayer perceptron (MLP) 모델<sup>9)</sup>과 유사하지만 입력 node가 모든 hidden node와 출력 node에

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

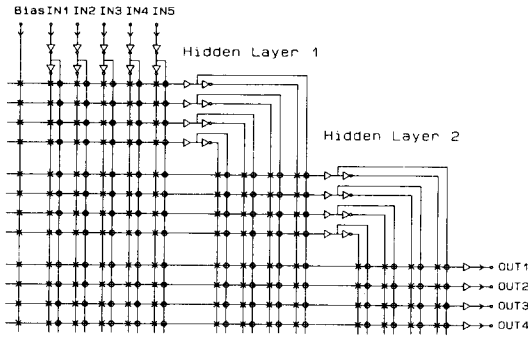


그림 1. IDMLP 모델  
Fig. 1. Input driven multi-layer perceptron (IDMLP) model.

연결되어 있는 것이 MLP 모델과 다른점이다.

이 모델의 특징은 다음과 같다. 첫째, hidden layer의 수가 미리 정해지지 않고, pattern의 구별이 어려워짐에 따라 그 수가 늘어나게 된다. 즉, hidden layer 하나로써 pattern을 구별하지 못하면, 원래의 입력과 첫번째 hidden layer의 출력을 입력으로 하는 두번째 hidden layer를 생성하게 된다. 만일 두번째 hidden layer에서도 구분이 불완전하면, 마찬가지로 세번째 hidden layer를 생성한다. 이렇게 하여 n개의 hidden layer를 사용하여 pattern이 제대로 구분되면, 이 n번째 hidden layer는 출력 layer의 역할을 하게 된다. 둘째, 각 hidden node의 수는 출력 node의 수에 따라 결정된다.

### III. 한자의 학습 데이터 추출과정

학습할 입력정보는 그림2와 같은 과정을 거쳐서 추출된다. 먼저 문자 데이터를 문서형태로 scanner를 통해 받아들여서 전처리 과정을 수행한다. 전처리 과정은 개별문자분리, 잡음제거, 선형화, 그리고 세선화 순으로 수행된다. 그 후 end point, three-forked point, 그리고 four-forked point 등 필요한 특징점들을 추출한다. 이러한 특징점들을 이용하여 stroke를 분리하고, 각 stroke이 가지는 방향을 조사한다. 또한 하나의 문자를 4영역으로 나누어서 각 영역에 포함된 end point의 수를 조사한다.

이러한 과정을 자세히 살펴보면 다음과 같다.

#### 1. 전처리 과정

입력된 문자영상에서 문자 하나의 영역을 결정하는 개별문자분리 과정을 가장 먼저 수행한다. 그 후

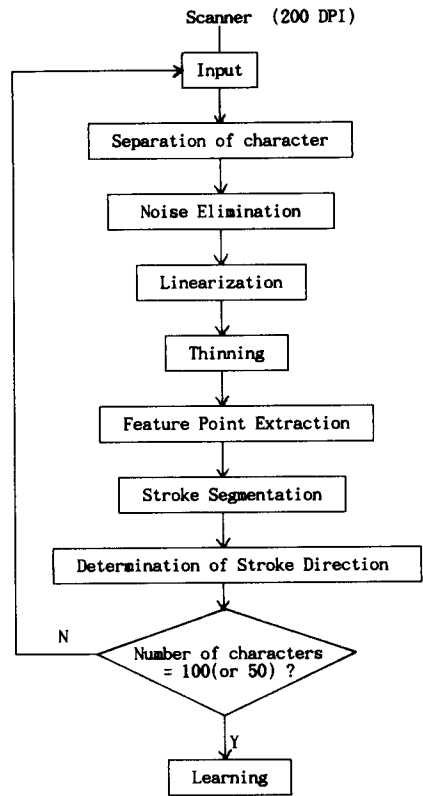


그림 2. 학습 데이터 추출과정  
Fig. 2. Procedure for the extraction of learning data.

인식에 나쁜 영향을 미칠 수 있는 잡음성분을 제거하고, 세선화에 영향을 미치는 요철부분에 대한 선형화 과정을 거친다. 그리고 특징점 추출을 용이하게 하기 위해 두꺼운 문자의 선폭을 하나의 화소폭으로 만들어 주는 세선화 과정을 수행한다.

#### 2. 특징점 추출 및 stroke 분리

Stroke을 분리하기 위해 end point, three-forked point, 그리고 four-forked point와 같은 특징점들을 이용한다. 두 end point 사이, end point와 three-forked point 또는 four-forked point 사이, 그리고 three-forked point 또는 four-forked point들 사이를 stroke으로 정한다. 이때 추출된 특징점 정보는 학습을 위한 입력 데이터로도 사용된다.

#### 3. Stroke의 방향 결정

분리된 각 stroke에 대해서 방향을 결정하여 이를 정보로 이용한다. 이때 방향은 그림3과 같이 4 방향

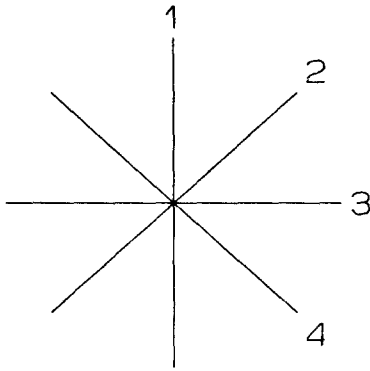


그림 3. Stroke의 4가지 방향  
Fig. 3. 4 directions of a storke.

으로 구분된다. 하나의 stroke을 구성하는 각 화소의 방향을 모두 조사하여 가장 주된 방향을 찾아 그 stroke의 방향으로 결정한다.

#### 4. 학습 데이터

학습에 사용된 입력정보는 위의 과정을 통해서 얻어진 다음과 같은 데이터이다.

- i) 각 방향별 storke의 수 (4×5-bit)
- ii) three-forked point의 수 (5-bit)
- iii) four-forked point의 수 (3-bit)
- iv) 각 영역별 end point의 수 (4×3-bit)

이 정보들은 2진수의 형태로 신경회로망의 입력 node에 입력된다. 이 때 총 입력 bit 수는 40-bit 이다.

본 연구에서의 학습은 super vised-learning이므로 각 입력에 대한 출력형태도 결정해야 한다. 출력으로는 워드 프로세서의 일종인 HWP에 로딩될 수 있는 16-bit 코드를 사용하였다.

### IV. 학습 알고리즘

먼저 하나의 hidden layer를 가정하고, 이 hidden layer의 출력이 원하는 출력과 같아지도록 hidden layer의 synapse weight를 조정한다. 이때 각 synapse의 weight는 1의 초기값을 가진다. Synapse의 weight 조정에 있어서는, 예를 들어 원하는 출력이 "1" 이고 실제 출력이 "0"일 때는 weight를 높이는 방향으로 변화시키고, 그 반대일 경우에는 weight를 낮추는 방향으로 변화시킨다. 이때 각 neuron을 거친 출력은 hard-limit 함수로써 결정된다. 이러한 과정을 일정한 횟수만큼 반복 수행한다. 구분할 pattern이 간단하다면 하나의 layer로도 학습이 정확하게 수행될

수 있으나, 한자와 같은 복잡하고 다양한 pattern을 구분하기 위해서는 여러개의 layer가 필요하게 된다. 그래서 하나의 layer로 구분이 제대로 되지 않으면, 두번째 layer를 생성하게 된다. 이 layer의 입력은, 원래의 입력과 첫번째 layer의 출력으로 구성된다. 또한 이 layer synapse의 초기 weight에 대해서, 초기의 입력이 들어가는 부분의 synapse들은 첫번째 layer에서의 학습결과 얻어진 weight값을 갖고, 첫번째 layer의 출력이 들어가는 부분은 1의 값을 가진다. 그 다음 layer에 대해서도 마찬가지이다.

이러한 학습 방법을 간단히 요약하면 다음과 같다.

1. 모든 weight를 1로 초기화 한다.
2. 하나의 입력 pattern을 회로망에 인가하고 출력을 계산한다.
3. a. 만일 출력이 원하는 출력과 일치하면, step 2로 간다;  
b. 출력이 틀리게 되고, 그 값이 0이면 weight를 증가시킨다;  
c. 출력이 틀리고, 그 값이 1이면 weight를 감소시킨다. 이 때 weight는 즉시 변화되는 것이 아니고 저장해 놓는다.
4. 마지막 pattern이 들어올 때까지 step 2로 간다.
5. a. 만일 error가 0이면, 학습을 종료하거나 step 3에서 저장된 값으로 weight를 변화시킨다.  
b. 일정 반복의 학습후에도 error가 0이 아니면, 새로운 layer를 생성하고 step 2로 간다.

학습을 수행할 때 외부에서 주는 파라미터는, 하나의 unit에서의 weight합의 최대치  $W_{max}$ 와 학습 횟수이다. 그 중  $W_{max}$ 는 학습 결과에서, 각 synapse의 weight 크기를 조정한다. 이  $W_{max}$ 가 너무 작으면 학습이 성공적으로 되지 않을 수가 있다. 또 이 값이 너무 크게되면 한 synapse가 가지는 weight가 너무 커서 실제 chip 제작상의 문제가 발생하게 된다. 그러므로 구분할 pattern의 종류와 특성에 따라 적절한  $W_{max}$  값을 선택해야 한다.

### V. 회로구성

하나의 synapse는 그림4와 같이 구성되어 있다. 입력이 0인지 1인지에 따라 ON되는 MOS가 결정된다. 입력이 0이면 MOS A와 D가, 입력이 1이면 MOS B와 C가 ON된다.

IDMLP 모델의 회로 구성을 자세히 살펴보기 위해 먼저 간단한 XOR와 NAND를 위한 회로에 대해

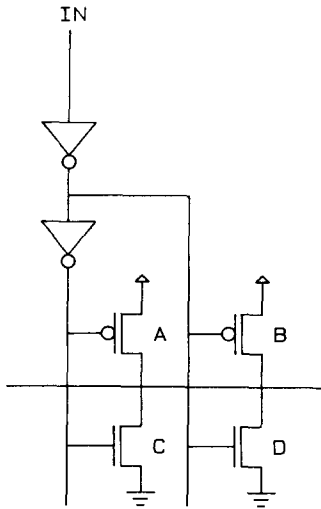


그림 4. 시냅스의 회로구현  
Fig. 4. The circuit of a synapse.

설명하고자 한다. 위에서 기술한 학습 알고리즘에 의해 표1의 입출력 데이터에 대해 학습을 수행한 결과 2개의 layer로써 학습이 성공적으로 이루어 졌다. 이 때 첫번째 layer 즉 hidden layer의 결과는 표 2 에, 두번째 layer 즉 출력 layer의 결과는 표3에 나타나 있다. 이러한 학습 결과 결정되어진 각 synapse의 weight 값을 CMOS 회로로 구현하면 그림6과 같다. 회로 왼쪽 부분의 PMOS는 bias 역할을 하는 것으로서, PMOS와 NMOS의 weight 합이 같을 때 출력이 1이 되도록 하기 위한 것이다.

회로의 동작은 다음과 같다. 입력 node에 들어오는 값에 따라 ON되는 MOS가 결정된다. ON된 MOS 중에서 PMOS의 weight 합이 NMOS의 그것보다 크거나 같으면 그 node의 출력은 "1"이 된다. 두번째 hidden layer에서는, 원래의 입력과 첫번째 hidden layer의 출력에 따라 ON되는 MOS가 결정되고, 출력 layer에서는, 입력과 두번째 hidden layer의 출력에 의해 최종 출력이 결정된다.

표 1. XOR와 NAND의 입출력 데이터

Table 1. Input and output data of XOR & NAND.

Input		Output	
0	0	0	1
0	1	1	1
1	0	1	1
1	1	0	0

표 2. XOR와 NAND를 위한 회로에서 첫번째 layer의 결과

Table 2. Result of the first layer in the circuit for XOR & NAND.

Input		h. value	
0	0	1	1
0	1	1	1
1	0	1	1
1	1	1	0

표 3. 두번째 layer의 결과

Table 3. Result of the second layer.

New Input				Output	
0	0	1	1	0	1
0	1	1	1	1	1
1	0	1	1	1	1
1	1	1	0	0	0

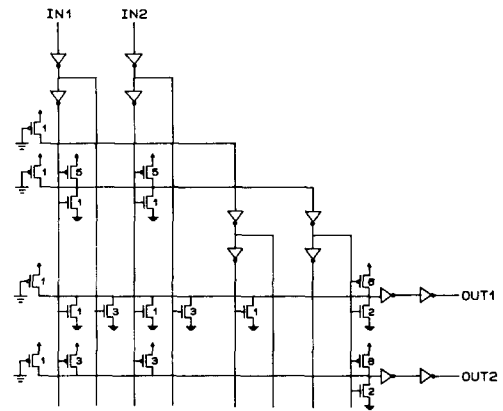


그림 5. XOR와 NAND의 회로구현 (OUT1=XOR, OUT2=NAND)

Fig. 5. The circuit for XOR & NAND.

VI. 결과 및 고찰

하나의 문자로부터 학습을 위한 입력 데이터를 추출하는 과정을 그림6에 나타내었다. 그림 (a)는 scanner로부터 받은 원래의 데이터에서 한 문자만 분리하여 이진 영상화한 것이고, (b)는 전처리 과정을 거친 것이다. 또한 (c)와 (d)는 각각 특징점 추출과 stroke 분리 과정을 거친 결과를 나타낸다. 그리고 (e)는 이러한 과정을 통해서 얻어진 40-bit의 데이터이다.

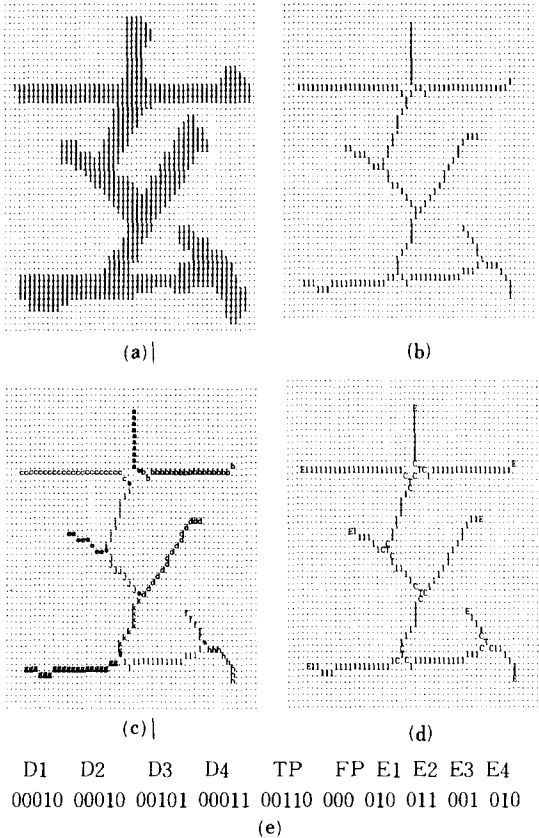


그림 6. 하나의 한자에 대한 학습 데이터 추출결과

- (a) 이진화된 원 영상
- (b) 전처리 과정을 거친 결과
- (c) 특징점 추출 결과
- (d) stroke 분리 결과
- (e) 추출된 데이터

Fig. 6. Result of extracting the learning data for a chinese character.

- (a) Original binary data,
- (b) After pre processing,
- (c) After extracting the feature points,
- (d) After segmenting the strokes,
- (e) Extracted data.

실제 학습에 사용한 한자의 입출력 데이터를 표 4에 나타내었다. 100자의 한자를,  $W_{max}$ 는 2000, 학습 반복횟수는 300번으로 하여 한꺼번에 학습한 결과 9개의 layer가 필요하였으며, 각 layer에서의 학습율과 최대 synapse의 weight 값은 표5와 같다.

이러한 결과를 살펴보면, synapse의 weight 값이 너무 크고, layer의 수도 많다는 것을 알 수 있다. 그러므로 이 결과를 실제 회로로 구현하는 것은 거의 불가능하다. 그래서 50개의 문자 단위로,  $W_{max}$ 를

표 4. 학습에 사용된 한자의 입출력

Table 4. Input and output of chinese characters for learning.

		INPUT	OUTPUT
1	天	000100000.0010000001000100010010010001	1111010101001000
2	地	0100100000001110000100010001010010001001	1111010000110010
3	玄	0001000010001010001100101001011010010100	1111000001101100
4	黄	0101100000100100000110010000010011001010	1111100101101100
5	宇	0010100000001110000000101010010011001001	1111000000001100
6	宙	0100000010010000000100000110001011000	1111001110110101
7	洪	0011000001011110001101010010100100000000	1111100100100101
8	荒	001100000000000000001111001100010019100	1111100100001000
9	日	00110000000011000000100000010010001011	1111000101111101
10	月	0011000000001100000011100001010001001	1111000001010101
.	.	.	.
91	道	0011100011010100001001111000100010010001	1110010101000011
92	垂	0100000001000000000100010001001011011	1110110000101001
93	拱	01011000010101100001010100101001010011	1110000111110010
94	平	001000000101100000000101001010010010001	1111011101010001
95	章	0011100001011010000101110000010011010001	1111001000100011
96	愛	0100000101011000001000111000011001010011	1110110110010011
97	育	010000000101001000010110001001001001001	1111000011000000
98	黎	0011001000010010001110101000011010010010	1110010111110011
99	首	0001100000010010001001000000010010000000	1110110001011111
100	臣	0011000000010000000000111100100001100100	1110100110101011

표 5. 100자의 한자에 대한 학습결과

Table 5. Learning result for 100 chinese characters.

layer	최대 synapse weight	학습율 (%)
0	950	77
1	431	84
2	871	92
3	1274	95
4	1281	97
5	1281	98
6	1281	99
7	1390	99
8	1414	100

100, 반복횟수를 200으로 했을 때와  $W_{max}$ 를 50, 반복횟수를 200으로 했을 때의 결과를 각각 표6과 7에 나타내었다.

이와 같은 실험 결과에서, 많은 수의 한자를 한꺼번에 학습하는 것은 회로설계에 부적당하고, 적절한 수의 한자에 대한 학습 결과를 하나의 회로로 구성하는 모듈화 방법을 사용하는 것이 바람직하다는 것을 알 수 있다.

표 6. 50자의 한자에 대한 학습결과(I)  
Table 6. Learning result for 50 chinese characters (I).

layer	최대 synapse weight	학습율 (%)
0	8	93
1	67	100

표 7. 50자의 한자에 대한 학습결과(II)  
Table 7. Learning result for 50 chinese characters (II).

layer	최대 synapse weight	학습율 (%)
0	4	88
1	26	98
2	26	100

Ⅶ. 결 론

한자로부터 특징들을 추출하여 이를 40-bit의 2진 정보로 만들고, 이를 IDMLP 모델을 사용하여 학습함으로써 한자에 대한 인식을 수행하였다. 특징을 추출하기 위해 선처리 과정으로서 개별문자분리, 잡음제거, 선형화 그리고 세션화 등을 수행하였고, 특징점 추출과 stroke 분리과정을 수행하였다. 학습에 사용된 입력 정보는 three-forked point의 수, four-forked point의 수, 분리된 각 stroke의 방향, 그리고 각 영역별 end point의 수 등이다. 또한 새로운 학습 알고리즘을 사용하여 synapse의 weight값이 정수가 되도록 학습을 수행하였다. 이렇게 정수의 weight를 가지는 synapse를 구성함으로써, 각 synapse를 그 weight에 따라 각각 다른 W/L을 가지는 MOS로 구현하여 학습 수행후의 결과를 실제 회로로 설계할 수 있다.

그러나 한꺼번에 많은 수의 문자를 구별하기 위해서는 hidden layer의 수가 너무 많고 weight도 너무 커지게 된다. 그러므로 앞으로는 인식하고자 하는 문자들을 몇개의 그룹으로 나누어 각 그룹에 대해 개별적으로 학습을 수행하여 그 결과를 모아서 하나의 인식 시스템을 만드는 방향으로 연구를 진행해야 할 것이다.

參 考 文 獻

- [1] C.H. Leung, Y. S. Cheung, and Y.L. Wong, "A Knowledge-Based Stroke-Matching Method for Chinese Character Recognition," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-17, no. 6, pp. 993-1003, December 1987.
- [2] G. Bao-Lan, Z. Cai-Lu, M. Ying-Li, and L. Su-Qin, "Printed Chinese Character Recognition Using Point Tracking Inclusive Matching Method with Backtracking Strategy," *Science in China*, vol. 32, no. 8, pp. 1011-1024, August 1989.
- [3] P.P. Wang and R.C. Shiau, "Machine Recognition of Printed Chinese Characters Via Transformation Algorithms," *Pattern Recognition*, vol. 5, pp. 303-321, 1973.
- [4] F.H. Cheng, W.H. Hsu, and C.A. Chen, "Fuzzy Approach To Solve the Recognition Problem of Handwritten Chinese Characters," *Pattern Recognition*, vol. 22, no. 2, pp. 133-141, 1989.
- [5] Y. Yong, "Handprinted Chinese Character Recognition Via Neural Networks," *Pattern Recognition Letters*, vol. 7, no. 1, pp. 19-25, 1988.
- [6] F.J. Pineda, "Generalization of Back-Propagation to Recurrent Neural Networks," *Physical Review Letters*, vol. 59, no. 19, pp. 2229-2232, 1987.
- [7] S.E. Fahlman and G.E. Hinton, "Connectionist Architectures for Artificial Intelligence," *IEEE Computer Mag.*, pp. 100-109, January 1987.
- [8] K. Fukushima, "Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-13, no. 5, pp. 826-824, 1983.
- [9] R.P. Lippmann, "An Introduction to Computing with Neural Nets, " *IEEE ASSP Ma.*, April 1987.

---

著 者 紹 介

---



呂 珍 環 (正會員)

1965年 10月 11日生. 1987年 慶  
북대학교 전자공학과 졸업. 1989  
年 동대학원 졸업. 1991年 동대  
학원 박사과정 수료. 주관심분야  
는 신경회로망, VLSI 설계 등임.

李 宇 一 (正會員) 第27卷 第6號 參照

현재 경북대학교 전자공학과  
교수

●  
鄭 鎬 宣 (正會員) 第28卷 B編 第9號 參照

현재 경북대학교 전자공학과  
교수