

論文 95-32B-11-18

# E-MIND II를 이용한 고립 단어 인식 시스템의 설계 (Isolated Word Recognition with the E-MIND II Neurocomputer)

金俊佑\* , 丁弘\* , 金明源\*\*

(Joon-Woo Kim, Hong Jeong, and Myeong-Won Kim)

## 요 약

본 논문에서는 256개의 DNP(Digital Neural Processor) II가 이차원의 원통형 웨이브 프론트 어레이 프로세싱 구조로 연결된 신경망 컴퓨터인 E-MIND II를 사용하여 고립 단어 인식 시스템을 구현하였다. E-MIND II를 구성하고 있는 단위 프로세서인 DNP II는 학습을 포함한 인공 신경망의 기능을 40MHz 이상의 속도로 수행할 수 있으며, 하나의 DNP II가 수천 개 이상의 단위 세포의 역할을 담당할 수 있다. 이러한 DNP II를 이차원 원통형 격자 구조로 연결함으로써 E-MIND II는 2 Gcps이상의 연산 속도를 가지게 된다. E-MIND II는 빠른 속도와 확장성 및 학습 능력을 가지므로 음성 인식 등과 같이 실시간 처리가 필요한 영역에 적합하다. 본 논문은 시간 지연 신경망(TDNN)을 E-MIND II에 맵핑하는 알고리즘을 제안하고 이 것을 이용하여 화자 독립 고립 단어 인식을 위한 인식 및 학습 알고리즘을 보이겠다. 하드웨어 시뮬레이션을 통하여 로봇트 제어 명령 30 단어를 대해 97%의 인식률을 얻었다.

## Abstract

This paper introduces an isolated word recognition system realized on a neurocomputer called E-MIND II, which is a 2-D torus wavefront array processor consisting of 256 DNP IIs. The DNP II is an all digital VLSI unit processor for the EMIND II featuring the emulation capability of more than thousands of neurons, the 40 MHz clock speed, and the on-chip learning. Built by these PEs in 2-D toroidal mesh architecture, the E-MIND II can be accelerated over 2 Gcps computation speed. In this light, the advantages of the E-MIND II in its capability of computing speed, scalability, computer interface, and learning are especially suitable for real time application such as speech recognition. We show how to map a TDNN structure on this array and how to code the learning and recognition algorithms for a user independent isolated word recognition. Through hardware simulation, we show that recognition rate of this system is about 97% for 30 command words for a robot control.

## I. 서 론

\* 正會員, 浦港工科大學校 電子電氣工學科

(Dept. of Elec. Eng., Pohang Univ. of Science and Techindogy)

\*\* 正會員, 崇實大學校 人工知能工學科

(School of Computing, Soongsil Univ.)

\* 이 논문은 한국 전자 통신 연구소의 출연금에 의해 수행된 결과입니다.

接受日字: 1995年3月15日, 수정완료일: 1995年11月3日

80년대 후반에 다시 주목을 받게 된 신경망은 음성 인식 연구에 새로운 방법론을 제공해 주었다. 신경망은 인간의 두뇌의 생물학적 신경 계통에 근거한 간단하고 많은 처리 요소들을 병렬로 연결하여 이루어진 것으로, 학습을 통하여 입력 패턴에 내재하는 정보를 처리하는데 용이하여 이를 음성 인식에 이용하려는 연구가 여러 분야에서 다양하게 시도되어 왔다.

특히 음성 인식이 있어서 신경망의 사용은 여러 가지 장점이 있는데 신경망은 인간의 신경 계통을 모방한 것이므로 인간이 음성을 인식하는 방법과 근접하며, 하드웨어로 구현된다면 병렬 처리가 가능하여 많은 계산을 빨리 할 수 있다. 또한, 작은 에러나 음성 신호의 잡음 등에 의한 영향에 비교적 잘 견디어 낼 수 있으며, 신경망의 구조를 변화시켜 원하는 특성을 잘 감지할 수 있도록 구현하기가 용이하다.

하지만 범용의 컴퓨터에서 신경망을 소프트웨어적으로 시뮬레이션할 경우 막대한 양의 시간이 소모되는 단점이 있다. 일반적으로 어려운 문제일 수록 더욱 큰 신경망을 필요로 하므로 이러한 신경망을 학습시키는데 수주 또는 수개월이 소요된다. 이때, 신경망 컴퓨터를 사용하면 신경망의 특성 중 하나인 병렬성을 잘 살릴 수 있으므로 이러한 문제점을 해결할 수 있게 된다. 이에 많은 연구소 또는 기업에서 여러 종류의 신경망 컴퓨터 들을 연구, 개발 및 판매하고 있다<sup>[11]</sup>.

이와 같은 신경망 컴퓨터는 특히 실시간 처리가 요구되는 문제 - 패턴 인식, 음성 인식 등 - 에 적합하며, 또한 실시간성이 요구되지 않은 문제라 하더라도 일반적으로 신경망의 학습에는 많은 시간이 소요되므로 이들의 학습 시간을 단축시키기 위하여 필수적이다. 국내에서도 최근 신경망 컴퓨터에 관한 관심이 고조되기 시작하여 몇몇 연구소 및 대학에서 자체적으로 신경망을 제작하고 있다. 특히 한국 전자 통신 연구소(Electronic Telecommunication Research Institute)에서는 8비트 정도(resolution)를 갖는 단위 프로세서(Processing Element) 1개를 포함하는 DNP(Digital Neural Processor) 칩을 개발하고 DNP칩 36개를 상호 연결하여 약 25MCPS의 연산 속도를 낼 수 있는 신경망 컴퓨터(EMIND: ETRI-Machine Imitating NeuroDynamics)를 제작하였다. 그리고 E-MIND 상에서 필기체 숫자 인식 시스템을 구현하여 98.6%의 인식률로 초당 20 - 30자를 인식할 수 있는 성능을 얻었다<sup>[12]</sup>.

현재 한국 전자 통신 연구소의 기초 기술 연구부에서는 DNP 칩의 상위 버전인 DNP II칩을 개발하고 이 칩을 사용하여 E-MIND II를 제작 중에 있다. 이 신경망 컴퓨터는 (1) 256개의 PE들이 2-D toroidal mesh wavefront array 구조로 연결되어 구성되어 있고 각 신경 단위를 시분할하여 대규모의 신경망을 구현할 수 있으며 (2) 각 PE들의 내부 메모리를 LUT

로 사용하여 임의의 비선형 변환이 가능하고, (3) 성능이 2GCPS 이상이므로 실시간 음성 인식이 가능하며, (4) 디지털 VLSI 기술을 사용하였으므로 범용 컴퓨터와의 접속이 용이하다는 장점이 있다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 신경망 칩인 DNP II와 이를 이용하여 만들어진 신경망 컴퓨터 E-MIND II의 구조를 간략히 설명 하고 3장에서는 TDNN을 E-MIND II에 맵핑하는 알고리즘을 제안한 뒤 4장에서 이를 이용하여 음성 인식을 구현한다. 마지막으로 5장에서는 결론을 맺는다.

## II. 신경망 컴퓨터 E-MIND II의 모델링

디지털 신경망 컴퓨터 E-MIND II는 DNP II칩이 사방으로 연결되어 신경망의 연산을 병렬적으로 수행할 수 있다. 이 장에서는 DNP II의 기능과 구조 및 E-MIND II의 구조에 대하여 알아본다.

### 1. 단위 프로세서 DNP II

ETRI에서 92년에 개발한 DNP I의 상위 버전인 DNP II 칩은 DNP I이 한 개의 PE만을 갖고 있는 것 과는 달리 네 개의 독립된 동일한 구조의 PE와 그것들을 제어하는 중앙 제어 블록(CCB: Central Control Block)으로 구성되어 있으며 독립적인 버스를 가지는 하버드 구조(Harvard architecture)를 채용하여 두개의 메모리에 동시에 접근할 수 있으므로 명령어와 operand를 가져오는 데 걸리는 시간을 단축할 수 있다.

DNP II 내부의 프로세서는 40MHz로 동작하고 최대 50MHz까지 동작 속도를 높일 수 있으며 16 16 병렬 파이프라인 곱셈기와 128위드의 입력 메모리(XM), 256위드의 프로그램 메모리, 그리고 512위드의 입력 강도 메모리(WM)를 갖고 주위에 있는 네 개의 다른 프로세서들과 병렬 비동기 통신을 할 수 있는 기능을 가진다.

### 2. 정수연산의 구현

DNP II의 내부 연산의 모든 단계에서 값들은 -32768에서 32767 사이의 범위 안에 있어야 한다. 덧셈이나 뺄셈을 수행한 후에 overflow가 발생하면 DNP II는 그 결과를 최대값(32767) 혹은 최소값(-32768)으로 대체한다. 또한 DNP II는 곱셈을 수행한 후에 그 결과로 얻어지는 32비트를 상위 30비트에

서 하위 15비트까지 잘라 해로 삼는다. 이때 그 결과가 양수의 경우는 32768로 나눈 후 몫의 정수 부분만을 취한 것과 같지만 음수의 경우에는 2의 보수로 표시되므로 32768로 나눈 후 몫의 정수 부분에서 다시 1을 뺀 값이 된다. 그리고 최소값과 최소값이 곱해지는 경우에는 곱셈에서도 overflow가 발생한다. 이 경우에는 최대값(32767)으로 결과를 대체한다.

이 것을 아래의 식으로 나타낼 수 있다.

$$A \otimes B = \begin{cases} \left[ \frac{A \times B}{32768} \right], & \text{if } A \times B \geq 0 \\ \left[ \frac{A \times B}{32768} \right] - 1, & \text{if } A \times B < 0 \\ 32767, & \text{if } A = B = 32767, 1 \rightarrow AC \end{cases}$$

이 식은 DNP II 내부에서 사용되는 곱셈 연산을 나타내고 있으며, E-MIND II를 시뮬레이션할 때 시간 지연 신경망의 전방향 연산이나 역방향 연산의 모든 단계에서 일반의 곱셈 대신 사용하게 된다. 또한 덧셈은 아래와 같이 요약할 수 있다.

$$A \oplus B = \begin{cases} 32767, & \text{if } A + B \geq 32767, \\ -32768, & \text{if } A + B \leq -32768, \\ A + B, & \text{otherwise} \end{cases}$$

한편, DNP II는 지수함수의 연산이 허용되지 않으므로 연결 강도 메모리에 참조표로 저장해야 하며, 연결 강도 메모리(WM)는 현재 512 워드로 설계되어 있으므로 무한대의 레벨로 저장할 수 없고 512개로 양자화되어야 한다. 참조표를 만드는 데 사용된 시그모이드 함수는 아래와 같다.

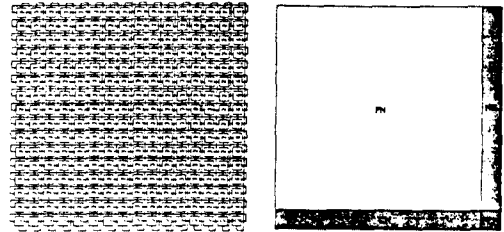
$$f(x) = \frac{2}{1 + e^{-x}} - 1$$

### 3. 신경망 컴퓨터 E-MIND II

신경망 어레이 프로세서의 종류에는 여러 가지가 있으나 그 중에서 대형 시스템을 만들기 가장 적합한 것은 WAP구조이다<sup>13)</sup>. EMIND-II은 WAP구조를 채택하여, 그림 1(a)과 같이 256개의 PE들을 이차원 격자 형태로 배열한 후 각 행과 열의 프로세서들을 링의 형태로 연결하여 이차원 원통형 격자(2-D toroidal mesh) 구조를 구현하였다.

그림 1(b)에 나타난 것과 같이 E-MIND II는 크게 보아 PN(Program Node), CN(Communication

Node), NF(Nonlinear Function node), 그리고 FCN (Final Communication Node)의 네 가지 기능 노드군으로 나뉜다.



(a) (b)

그림 1. E-MIND II의 네가지 기능 노드군  
Fig. 1. Four functional nodes groups of E-MIND II.

일반적으로 CN들은 신경망에 입력 패턴을 제공하고, PN들은 신경망의 가중 합을 계산하며, NF들은 그 결과를 비선형 변환하는 역할을 한다. 한편, DNP II는 비동기 방식의 프로세서이므로 핸드 웨이킹을 통하여 데이터를 주고 받기 때문에 SIMD구조에 비해 전송 시간이 길어지는 단점이 있다. 따라서 이 전송 시간을 줄이기 위해 데이터가 이차원으로 배열된 프로세서 위를 물결처럼(wavefront) 지나가는 방식을 사용한 것이다. 이러한 시스템의 장점은, 일반의 대형 시스템에서는 클럭의 분배가 매우 중요한 문제인데 반하여, 데이터에 의존해서 동작(data-driven)하기 때문에 클럭 분배에 대한 부담이 줄어든다는 것이며, 따라서 동기 시스템보다 고속으로 동작시킬 수 있다. 또한, 동작 중에 하나의 프로세서가 오 동작을 하여도 신경망의 특징인 오류 복구성(fault tolerance)을 가지므로 훈련을 반복하면 주위의 프로세서들이 이를 보상할 수 있다.

그림 2에 E-MIND II 전체 시스템의 기능 블록도를 나타내었다.

### III. E-MIND II를 이용한 시간 지연 신경망의 구현

이장에서는 II장에서 설명한 모델을 바탕으로 E-MIND II에 시간 지연 신경망을 구현 하는 방법에 대해 기술한다. 먼저 시간 지연 신경망에 대해 설명하고, 학습과 인식 단계를 E-MIND II에 구현하는 방법을

제안한다.

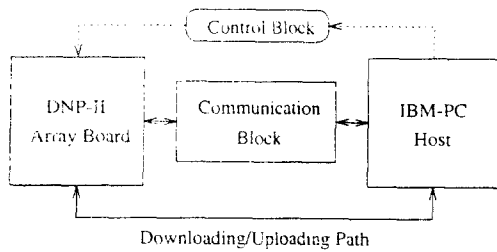


그림 2. E-MIND II 전체 시스템의 기능 블럭도  
Fig. 2. Functional block diagram of overall E-MIND II system.

1. 시간 지연 신경망(Time-Delay Neural Network)의 구조

음성 신호는 시간적인 정보를 가지고 있으므로 효과적인 음성 인식을 위해서는 신경망의 구조가 시간 굴곡(time-warping) 현상과 시간 지연(time-delay) 현상 등의 음성 신호의 특성을 잘 감지 할 수 있는 형태이어야 할 것이다. 시간 지연 신경 회로망은 이를 위해서 제안된 구조이다<sup>[4]</sup>.

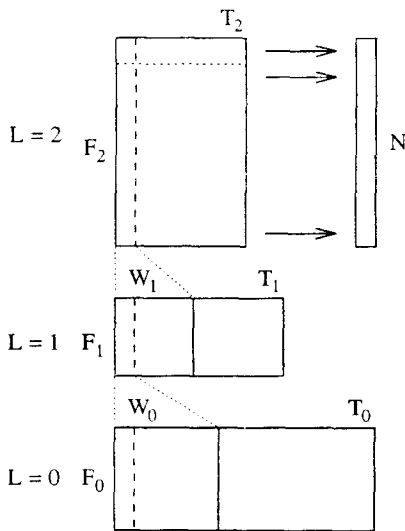


그림 3. 일반적인 3층 구조 시간 지연 신경망  
Fig. 3. A 3-layer TDNN.

시간 지연 신경망에서 하나의 처리 요소는 현재의 음성 벡터  $f(t)$ 와 함께 시간적으로 지연된  $f(t-1)$ ,  $f(t-2)$ , ...,  $f(t-k)$ 를 동시에 입력으로 받아들이고 이들에 각각 연결 강도를 가중합한 것을 활성화 함수

를 통하여 출력한다. 따라서 시간 축에서 인접한 입력들의 관계를 잘 반영할 수 있으므로 음성 신호의 시간적 굴곡을 잘 감지할 수 있게 된다. 사용되는 활성화 함수는 일반적으로 시그모이드 함수가 쓰이나 학습 수렴성의 향상을 위하여 다양한 형태가 제안되고 있다<sup>[5,6]</sup>.

그림 3은 두개의 은닉층을 가지는 시간 지연 신경망을 나타낸 것이다. 각 파라미터는 다음과 같이 정의된다.  $L$ 은 각 계층의 번호를 나타내며 아래에서부터 차례로  $L=0$ ,  $L=1$ ,  $L=2$ 와  $L=3$ 이다. 여기에서  $L=0$ 는 입력층을,  $L=3$ 은 출력층을 나타낸다. 따라서 은닉층은  $L=1$ 과  $L=2$ 의 2층이다. 이 구조를 3층 구조 시간 지연 신경망이라고 한다.  $F_n$ 는 입력 패턴의 주파수 बैं크 수를 나타내며  $F_1$ 과  $F_2$ 는 각 계층의 출력 뉴런의 갯수를 나타낸다.  $W_n$ ,  $w_n$ 은 각 층의 입력 윈도우 크기를 나타내며  $T_n$ ,  $t_n$ 는 시간적으로 나열된 입력 패턴의 수를 나타낸다. 마지막으로  $N$ 은 인식 단어 수가 된다.

2. 시간 지연 신경망의 학습

학습의 단계에서 시간 지연 신경망은 일반적인 역전파(back-propagation) 알고리즘<sup>[7]</sup>을 사용하지만 그 구조가 일반적인 다층 인식자와 달라 원래의 것에서 조금 바뀌어진다<sup>[8]</sup>.

일반적인 BP에서 목적 함수(objective function)는 실제 출력 값  $O_j$ 와 목적 값  $D_j$ 의 차의 제곱의 합으로 정의된다.

$$E = \frac{1}{2} \sum_j (O_j - D_j)^2$$

여기에서 시간 지연 신경망의 출력값은 두번째 은닉층에서 시간적으로 정렬된 노드들의 출력값들  $(y_{\mu}^{(2)}, y_{\mu}^{(2)}, \dots, y_{\mu}^{(2)})$ 의 제곱의 합인

$$\frac{\partial E}{\partial y_{\mu}^{(2)}} = 2y_{\mu}^{(2)} \left( \sum_j (y_{\mu}^{(2)} - D_j) \right)$$

이 된다. 따라서 시간 지연 신경망에서는 이 값에 비선형 함수의 미분값을 곱한 후 그 결과를 두번째 은닉층의  $y_{\mu}^{(2)}$ 로부터 역전파하여 학습이 진행된다. 그리고 각 연결 강도(weight)에 대하여 역전파가 진행된 후, 음성 신호의 특성이 시간적인 위치에 무관하게 감지할 수 있도록 구조적으로 같은 위치에 있는 연결 강도들

을 평균하여 복사하는 과정을 거친다.

3. 전방향 연산의 맵핑

시간 지연 신경망은 기본적으로 2층의 은닉층을 갖는 MLP으로 이해할 수 있다. 한편 E-MIND II는 일반적으로 한번의 phase(한 종류의 계산의 물결)로 두 계층 간의 연산을 수행할 수 있으므로 모두 세번의 phase면 입력층 부터 출력층 까지의 모든 연산이 끝나게 된다. 그러나 두번째 은닉층과 출력층 사이의 연산은 단순한 제곱함이므로 한번의 phase를 절약할 수 있어, 단지 하나의 은닉층을 갖는 단순한 MLP과 마찬가지로 2 phase로 모든 전방향 연산을 마치게 된다.

E-MIND II를 구동하기 위해서는 먼저 각 노드에 필요한 정보를 적재(downloading)해야 한다. 필요한 정보에는 우선 각 노드마다 자신을 구동하기 위한 프로그램과 레지스터들을 위해 사용될 상수값들이 있다. 또한 비선형 변환을 위한 512단계의 참조표와 입력 패턴, 초기 연결 강도 값들, 임계치(bias)들, 그리고 학습을 위한 목적 출력 값(desired output value)과 학습률 및 관성률(momentum)이 필요하다.

그림 4는 전방향 연산시 데이터가 처리되는 과정을 보여주고 있다.

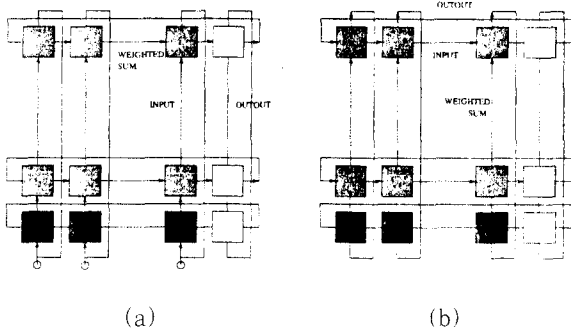


그림 4. 전방향 연산시 데이터의 흐름: (a) 첫번째 은닉층의 연산, (b) 두번째 은닉층과 출력층의 연산

Fig. 4. Data flow in the forward processing: (a) the first hidden layer, (b) the second and the output layer.

그림 4(a)는 첫번째 은닉층의 연산 과정을 보인 것이다. 일단 프로그램이 시작되면 CN은 XM에 저장된 입력 패턴의 값들을 상층의 노드로 전달한다. 이것을  $[\frac{F_0}{15}]$   $T_1$ 번 반복하면 모든 입력 패턴값의 전달이 끝나게 된다.

각 PN은 하측에서 올라온 입력값을 저장하고 다시 상측으로 전파(broadcast)한다. 그리고 좌측에서 전달된 부분합에 XM에 들어 있는 입력값을 WM에 저장된 연결 강도와 곱한 것을 더하여 우측의 PN으로 전달한다. 이것을  $[\frac{F_0}{15}]$   $w_1$ 번 반복하면 첫번째 은닉층의 한열의 연산이 끝난다. 그리고 위의 동작을 반복한다. 첫번째 phase에서는 이것을  $[\frac{F_0}{15}]$   $T_1$ 번 반복하게 된다.

NF 노드까지 전파된 부분합들은  $[\frac{F_0}{15}]$   $w_1$ 번 더해진 후 XM에 저장된 임계치(bias)를 뺀 후에 WM에 저장된 참조표에 따라 비선형화되고 다음 층의 입력으로 쓰기 위해 다시 XM에 저장된다. 이 것이  $[\frac{F_0}{15}]$   $T_1$ 번 반복되면 첫번째 은닉층의 연산이 끝나게 된다. 이것을 식으로 나타내면 다음과 같다.

$$y_{jk}^{(1)} = f \left( \sum_{i=0}^{t+w-1} \sum_{l=0}^{F_0} W_{j,i}^{(0)} \cdot F_{i \cdot F_0 + k - l} y_{ik}^{(0)} - \phi_j^{(0)} \right)$$

이 식에서  $y_{jk}^{(1)}$ 는 첫번째 은닉층의 j번째 뉴런의 t번째 열의 값을 나타내는 것이고  $y_{ik}^{(0)}$ 는 (i,k)좌표의 입력 패턴값이며  $W_{j,i}^{(0)}$ 와  $\phi_j^{(0)}$ 는 각각 입력층의  $i \times F_0 + k - l$ 번째 뉴런과 은닉층의 j번째 뉴런과의 연결 강도와, 첫번째 은닉층의 j번째 뉴런의 임계치를 나타낸다.

그림 4(b)는 두번째 은닉층의 연산 과정을 보인 것이다. 이 단계에서 입력값들은 NF에서 제공된다. 링을 통하여 최우측열의 NF로부터 최좌측의 PN열로 입력값을 전송하고 이 값들이 우측으로 전파된다.

각 PN은 좌측에서 전파된  $[\frac{F_0}{15}]$   $T_1$ 개의 입력값을 저장하고 다시 우측으로 전파(broadcast)한다. 그 다음 하측에서 전달된 부분합에 XM에 들어 있는 입력값을 WM의 연결 강도와 곱한 것을 더하여 상층의 PN으로 전달한다. 이것을  $[\frac{F_0}{15}]$   $w_1$ 번 반복하면 두번째 은닉층의 한열의 연산이 끝난다. 두번째 phase에서는 이것을  $[\frac{F_0}{15}]$   $T_2$ 번 반복하게 된다. 링을 통해 CN까지 전파된 부분합들은  $[\frac{F_0}{15}]$   $w_1$ 번 더해진 후 XM에 저장된 임계치(bias)를 뺀 후에 WM에 저장된 참조표에 따라 비선형화되고 출력층의 연산을 위해 다시 XM에 저장된다. 이 것이  $[\frac{F_0}{15}]$   $T_2$ 번 반복되면 두번째 은닉층의 연산이 끝나게 된다. 이것을 식으로 나타내면 다

음과 같다.

$$y_{jt}^{(2)} = f \left[ \sum_{k=1}^{t-1} W_{jk}^{(1)} y_{ik}^{(1)} + \sum_{k=t}^{F-1} W_{jk}^{(1)} y_{(j,k-t)}^{(1)} - \phi_j^{(1)} \right]$$

이 식에서  $y_{jt}^{(2)}$ 는 두번째 은닉층의 j번째 뉴런의 t번째 열의 값을 나타내는 것이고  $y_{jk}^{(1)}$ 는 첫번째 은닉층의 i번째 뉴런의 k번째 열의 값을 나타내는 것이며  $\sum_{k=t}^{F-1} W_{jk}^{(1)} y_{(j,k-t)}^{(1)}$ 와  $\phi_j^{(1)}$ 은 각각 첫번째 은닉층의  $i \times F_1 + k - t$ 번째 뉴런과 두번째 은닉층의 j번째 뉴런과의 연결 강도와, 두번째 은닉층의 j번째 뉴런의 임계치를 나타낸다.

마지막으로 출력층의 값들은 두번째 은닉층의 뉴런들의 값들을 시간에 따라 제공합한 것인데 이미 이 값들은 두번째 phase의 결과로서 CN의 XM에 저장되어 있으므로 이 값들을 제공함하여 XM에 저장한다.

$$O_i = \sum_{j=0}^{T_2-1} (y_{jt}^{(2)})^2, \quad \text{for } j = 0, 1, \dots, N-1$$

이것으로 전방향 연산이 모두 끝난다. 이 출력값들 중에서 최고치를 찾아 인식된 단어로 한다.

4. 역방향 연산의 경우

그림 5는 역방향 연산시 데이터가 처리되는 과정을 보여주고 있다.

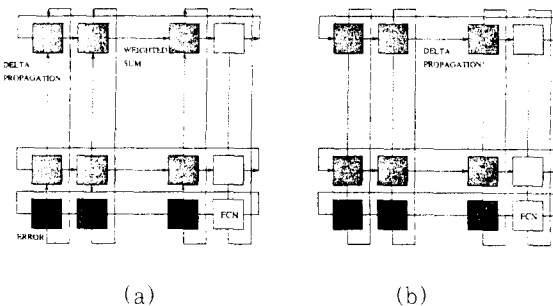


그림 5. 역방향 연산시 데이터의 흐름: (a) 두번째 은닉층의 델타 연산, (b) 첫번째 은닉층의 델타 연산

Fig. 5. Data flow in the backward processing: (a) calculation of delta for the second hidden layer, (b) calculation of delta for the first hidden layer.

두번째 은닉층의 델타를 구하기 위해서는 데이터를 전파할 필요가 없다. 각 CN은 전방향 연산에서 구한 출

력값에 목적 출력값을 뺀 오차값을, 저장하고 있던 출력값과 곱한 후 시그모이드 함수의 미분치와 곱하여 두번째 은닉층의 델타를 구한다. 이 것으로 임계치를 갱신하고 상층의 PN으로 델타값을 전파한다.

$$\delta_{jt}^{(2)} = (O_j - D_j) y_{jt}^{(2)} \frac{\partial f}{\partial y_{jt}^{(2)}}$$

여기에서  $\delta_{jt}^{(2)}$ 은 두번째 은닉층의 j번째 뉴런의 t번째 열의 델타를 나타낸다. 결과적으로 각 CN은  $\lfloor \frac{N}{15} \rfloor T_2$ 개의  $\delta_{jt}^{(2)}$ 을 전파하게 된다.

그림 5(a)는 두번째 은닉층과 첫번째 은닉층 사이의 연결 강도를 갱신하는 과정을 보여주고 있다. 각 PN들은 델타값과 학습률 및 관성률을 사용하여 연결 강도를 조정한다.

$$\Delta W_{jk}^{(1)} = \eta \frac{1}{T_2} \sum_{t=0}^{T_2-1} \delta_{jt}^{(2)} y_{ik}^{(1)} + \alpha \Delta W_{jk}^{(1)}$$

$$\Delta \phi_j^{(1)} = \eta \frac{1}{T_2} \sum_{t=0}^{T_2-1} \delta_{jt}^{(2)} + \alpha \Delta \phi_j^{(1)}$$

여기에서  $\eta$ 는 학습률,  $\alpha$ 는 관성률,  $\Delta W_{jk}^{(1)}$ 는 지난 번의 연결 강도의 변화량, 그리고  $\Delta \phi_j^{(1)}$ 는 지난번의 임계치의 변화량을 나타낸다. 그 후, 가지고 있는 연결 강도를 곱하고 좌측 노드에서 들어온 부분합과 더하여 우측으로 전파한다.

그림 5(b)는 첫번째 은닉층과 입력층 사이의 연결 강도를 갱신하는 과정을 보여주고 있다. NF는 좌측에서 들어오는 가중합을 더해서 첫번째 은닉층을 위한 델타값들을 구한 뒤 저장하고 있던 출력값 및 시그모이드 함수의 미분치와 곱하고 링을 통해 우측으로 전파한 후 임계치를 갱신한다. 각 PN은 위에서와 마찬가지로 저장하고 있던 입력값, 학습률 및 관성률을 사용하여 연결 강도를 조정한다.

$$\delta_{jt}^{(1)} = f' (y_{jt}^{(1)}) \sum_{k=0}^{N-1} W_{jk}^{(0)} \delta_{kt}^{(1)}$$

$$\Delta W_{jk}^{(0)} = \eta \frac{1}{T_1} \sum_{t=0}^{T_1-1} \delta_{jt}^{(1)} y_{kt}^{(0)} + \alpha \Delta W_{jk}^{(0)}$$

$$\Delta \phi_j^{(0)} = \eta \frac{1}{T_1} \sum_{t=0}^{T_1-1} \delta_{jt}^{(1)} + \alpha \Delta \phi_j^{(0)}$$

5. 메모리의 사용량

III 3절과 III 4절에서 기술된 알고리즘을 이용할 경우 메모리의 사용 정도는 다음과 같다.

표 1. 전체 메모리 사용량  
Table 1. Total Usage of Memory.

기능 블록	메모리	사용량
CN	XM	$\left[ \frac{F_0}{15} \right] T_1 - \left[ \frac{N}{15} \right] T_2 + \left[ \frac{N}{15} \right]$
	WM	LUT
PN	XM	$\left[ \frac{F_1}{15} \right] T_1 - \left[ \frac{F_1}{15} \right] T_1$
	WM	$2 \left[ \frac{F_1}{15} \right] \left[ \frac{F_1}{15} \right] W_0 - 2 \left[ \frac{N}{15} \right] \left[ \frac{F_1}{15} \right] W_1$
NF	XM	$2 \left[ \frac{F_1}{15} \right] - 2 \left[ \frac{F_1}{15} \right] T_1$
	WM	LUT

6. 성능 평가

L개의 은닉층을 가지는 시간 지연 신경망을 E-MIND II에 구현했을 경우 전방향 연산시 걸리는 시간은 다음과 같다.

$$T_s \approx \sum_{i=1}^L \left\{ t_1 T_1 - \left[ \frac{F_0}{15} \right] \left[ \frac{F_1}{15} \right] W_1 T_{i+1} + t_2 + t_3 \right\} - t_1 T_1 \left[ \frac{N}{15} \right]$$

여기에서  $t_1$ 은 한개의 데이터를 전파하는 데 걸리는 시간,  $t_2$ 는 한 개의 부분합을 처리하는 데 걸리는 시간, 그리고  $t_3$ 는 출력층을 계산하는 데 걸리는 시간이다.  $t_1 = 2, t_2 = 5,$  그리고  $t_3 = 4$ 로 하면 V장에서와 같이 독립 변수가  $(F_0, W_0, T_0, F_1, W_1, N) = (15, 8, 20, 15, 7, 30)$ 일 경우 성능은 2.3Gcps정도가 된다.

IV. 실험 및 결과 고찰

이 장에서는 III장에서 제안한 맵핑 알고리즘을 이용하여 음성 인식을 구현하는 방법을 보인다.

음성 인식 시스템은 기본적으로 특징 추출, 인식, 그리고 후처리의 세단계로 분류할 수 있다. 본 논문에서 구현하고 있는 시스템은 그림 6과 같다.

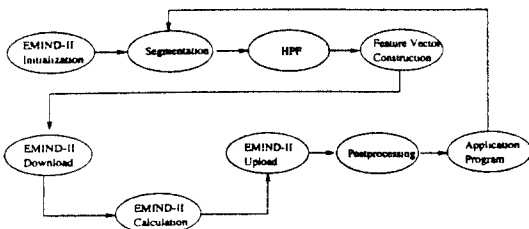


그림 6. 전체 시스템의 기능도  
Fig. 6. Diagram of the overall system.

초기화 프로그램은 시스템이 구동될 때 컴퓨터의 내

부 상태를 초기화 하고 E-MIND II를 동작시키기 위하여 필요한 각 DNP II 칩의 프로그램, 초기 연결 강도 등을 적재(download)한다. 마이크를 통해 들어온 음성 신호는 먼저 세그멘테이션 과정을 거치고 양자화(quantization)된 후, 음질을 높이기 위해 고대역 필터를 거친다. 특징점 추출(feature construction) 과정에서는 음성 인식기의 입력이 될 15차의 필터뱅크 계수들을 생성한다. 이 데이터로 부터 입력 피쳐 벡터를 구성하고 E-MIND II로 download하게 된다. 처리된 결과를 다시 upload하고 후처리를 거쳐 응용 프로그램을 동작시킨다. 다시 처음으로 돌아가 음성 신호를 받아 처리하는 작업을 반복한다.

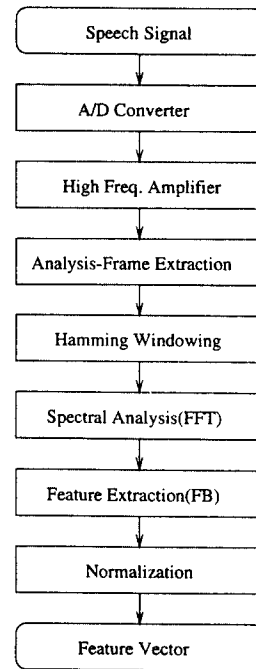


그림 7. 전처리 과정  
Fig. 7. Preprocessing.

1. 음성 신호의 특징 추출

이번 실험에 사용된 음성 데이터 베이스는 본 실험실에서 구축된 것으로 8KHz로 음성으로 로보트를 구동하기 위해 열명의 화자가 발음한 30단어로 이루어져 있다.

이 음성 데이터 베이스를 그림 7과 같은 전처리 과정을 거쳐의 음성 스펙트럼<sup>19)</sup> 패턴을 만들어 신경망의 입력 패턴으로 한다. 이때  $F_0 = 15, T_0 = 20$ 이 된다.

## 2. E-MIND II를 이용한 음성 인식기

앞절에서 기술한 것과 같은 전처리 과정을 통하여 3000개의 입력 패턴을 구하고 이 중 70%를 훈련용 패턴으로, 그리고 나머지 30%를 테스트용 패턴으로 하였다. 시간 지연 신경망의 독립 변수가  $(F_0, W_0, T_0, F_1, W_1, N) = (15.8, 20.15, 7.30)$ 와 같을 때 2100개의 훈련 패턴에 대하여 97.10%, 300개의 테스트 패턴에 대해서는 96.56%의 인식률을 얻었다. 이것은 DNP II 내부의 연산을 쓰지 않고, 32 비트의 실수 연산 실수 연산 시뮬레이션에서 얻은 98.14%, 97.14%의 인식률과 근소한 차이이며 E-MIND II가 2Gcps 이상의 계산 속도를 가지는 것을 고려하면 실시간 음성 인식에 충분히 사용될 수 있음을 알 수 있다.

## V. 결 론

이상에서 신경망 컴퓨터인 E-MIND II에 시간 지연 신경망을 구현하기 위한 알고리즘을 기술하고, 이것을 사용하여 한국어 고립 단어 인식 시스템을 구현하였다.

맵핑된 신경망은 on-line 학습이 가능하도록 구현되었으며 셉스트럼을 이용해 만들어진 단어 패턴을 입력으로 사용하여 음성을 인식하도록 훈련되었다. 이 시스템은 시뮬레이션을 통하여 다화자의 테스트 패턴에 대하여 96.56%의 인식률을 보였다. 추후 하드웨어가 완성되어 충분하고도 다양한 훈련이 가능해지면 대용량 인식 단어에 대하여 빠른 학습 능력을 이용하여 높은 인식률을 보일 것으로 기대된다. 한편, 인식 과정만 수행할 경우 대부분의 연산 시간은 음성으로 부터 입력 패턴을 만들어 내는 전처리 과정에 소요된다. 따라서 인식 속도를 올리기 위해서는 추후 전처리 과정에서 소모되는 시간을 줄이는 방법이 고안되어야 할 것으로 생각된다.

## 참 고 문 헌

[1] Dan Hammerstrom, "Neural Networks at Work," *IEEE Spectrum*, pp. 26-32, June 1993.  
 [2] Myung Won Kim, Youngjik Lee, Hoon Bock Lee, Junk Sik Lee, Jong-Moon Kim, Jea-Hoon Kim, Sang-Hoon Oh,

Chae Deok Lim, and Hyun Kyung Song, "EMIND: an Implementation of a Digital Neurocomputer and Its Application to Handwritten Digit Recognition," *IJCNN*, vol. 3, pp. 258-263, Beijing, China, Nov. 1992.

- [3] S. Y. Kung, *VLSI Array Processors*. Prentice-Hall Int., 1988.  
 [4] K. J. Lang and A. Waibel, "A Time-Delay Neural Network Architecture for Isolated Word Recognition," *Neural Networks*, vol. 3, pp. 23-43, 1990.  
 [5] J. B. Hampshire and Barak Pearlmuter, "Equivalence Proofs for Multi-Layer Perceptron Classifiers and the Bayesian Discriminant Function," *Proc. of 1990 Connectionist Models Summer School*, pp. 159-172, 1991.  
 [6] J. I. Takami and S. Sagayama, "A Pairwise Discriminant Approach to Robust Phoneme Recognition by Time-Delay Neural Networks," *Proc. of ICASSP*, vol. 1, pp. 81-84, 1991.  
 [7] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagation errors," *Parallel Distributed Processing*, vol. 1, D. E. Rumelhart and J. L. McClelland, eds., MIT Press, Cambridge, MA, 1986.  
 [8] K. J. Lang and G. E. Hinton, "The development of the time-delay neural network architecture for speech recognition," Technical Report CMU-CS-88-152, Carnegie-Mellon University, PA, 1988.  
 [9] John R. Deller, Jr., John G. Proakis, and John H. L. Hansen, *Discrete-Time Processing of Speech Signals*, New York: Macmillan Publishing Co., 1993.



## — 저 자 소 개 —



金 俊 佑(正會員)

1991년 포항공과대학에서 전자공학 학사학위를 취득했으며, 1995년 동 대학 전자공학과 석사학위를 취득하였다. 주관심분야는 음성 신호처리, 신경망 그리고 병렬프로세스이다.



丁 弘(正會員)

1977년 서울대학교에서 전자공학 학사학위를 취득하였으며, 1979년 KAIST에서 전자공학 석사학위를 취득하였다. 1984년, 1986년, 1988년 각각 미국 MIT에서 전자전산 S.M., E.E., 그리고 박사학위를 취득하였다. 1972년~1982년 경북대학교에서 재직했으며, 1988년~현재 포항공과대학에 교수로 재직중이다. 그는 Sigma-Xi의 회원이며, 주관심분야는 음성인식, 컴퓨터비전, 그리고 신경회로망이다.

金 明 源(正會員)

서울대학교에서 응용수학 학사학위를 취득했으며, 미국 massachusetts대학 컴퓨터학과에서 석사와 박사학위를 취득하였다. 1985년~1987년 AT&Bell 연구실에서 근무하였으며, 1987년~1994년 ETRI에서 신경망구룹을 이끌었다. 1994년~현재 그는 숭실대학교 계산학과에서 교수로 재직 중이다. 그의 관심분야는 신경망, 패턴인식, 지식표현과 신경망의 병렬연산이다.