

단조 결정 함수를 갖는 축약 분산 기억 장치

권희용[†]·장정우^{††}·임성준^{††}·조동섭^{†††}·황희웅^{††††}

요 약

최근 축약 분산 기억 장치(SDM)가 적응적 문제 해결 능력과 하드웨어화의 용이성으로 인해 현실성이 있는 신경망의 한 모델로 제안되었다. 그러나 다층 인식자의 개별 뉴런이 선형 또는 비선형 결정 함수로 해 공간을 이분하고 그들이 다양하게 결합함으로써 일반적인 문제 해결 능력을 갖는데 비해, 축약 분산 기억 장치의 뉴런은 해 공간에서 자신을 중심으로 한 일정 반경 영역을 안과 밖으로 이분하고 이들을 단순하게 합하므로써, 해 공간이 실수 공간과 같이 크기 관계를 갖는 경우 비효율적인 모델로 된다. 본 논문에서는 이러한 축약 분산 기억 장치의 특성과 그 원인을 규명하고, 문제의 해 공간이 단조 증가 또는 감소 결정 함수로 양분되는 경우, 기존의 축약 분산 기억 장치에 크기 비교 과정을 도입함으로써, 주어진 문제를 효율적으로 해결할 수 있는 수정된 축약 분산 기억 장치 모델을 제안한다. 아울러 제안된 모델을 ATM망에서의 호 수락 제어 과정에 적용한 예를 보인다.

Sparse Distributed Memory with Monotonic Decision Function

Hee-Yong Kwon[†] · Jung-Woo Jang^{††} · Sung-Joon Lim^{††} ·
Dong-Sub Cho^{†††} · Hee-Yeung Hwang^{††††}

ABSTRACT

Sparse Distributed Memory (SDM) has been proposed as a practical neural network model due to its adaptability in problem solving and simplicity in hardware implementation. SDM, however, becomes an inefficient model in case there are inequality relations among the objects in the solution space like real number, because the neurons of a SDM bisect a solution space into the inside and the outside of a circle with a specific radius, and are unioned simply. On the other hand, the neurons of a multilayer perceptron (MLP) bisect a solution space using a linear/nonlinear decision function, and are variously combined with each other to solve general problems. Thus it can solve a general problem. In this paper, we study the characteristics and the cause of the inefficiency of a SDM, and propose a modified SDM which can solve it effectively when a solution space is divided into two regions by a monotonic decision function. To solve it, we introduce a magnitude comparing step into the conventional SDM algorithm. In addition, we show the experimental result by applying the proposed model to an ATM call admission control.

키워드 : 신경망(Neural Networks), 다층인식자(Multilayer Perceptron), 축약 분산 기억 장치(Sparse Distributed Memory), 호 수락 제어(Call Admission Control)

1. 서 론

기존의 신경망 연구는 대부분 오류 역전파 학습에 의한 다층 인식자(Multi-Layer Perceptron) 신경망에 집중되고 있다. 다층 인식자는 뛰어난 문제 해결 능력을 기반으로 많은 적용 예를 보이고 있다. 다층 구조로 되어 있는 다층 인식자는 각 계층의 가중치를 모두 학습하여 출력을 결정할 수 있다는 장점이 있다[1, 2]. 그러나 하드웨어 구현이 어렵다는 점과 새로운 자료의 학습시 기존의 모든 자료가 함께

학습이 되어야 하기 때문에 환경이 빠르게 변화하는 응용 영역에서는 적용 속도가 느리다는 문제점이 있었다[3, 4]. 이러한 문제를 해결할 수 있는 신경망 모델 중 하나가 축약 분산 기억 장치(Sparse Distributed Memory) 신경망이다. 축약 분산 기억 장치는 Kanerva가 처음으로 제안한 신경망 모델로써 다차원 공간상의 연상 기억 장치의 일종으로써 RAM과 유사하지만 적은 수의 주소로써 방대한 주소 공간을 표현할 수 있어, 환경의 변화가 심하여 높은 적응력이 필요한 분야나 패턴 인식 분야에 효율적이다. 또한 퍼셉트론으로 구현이 가능하여 하드웨어 구현이 용이하므로 실용성이 높은 신경망이라 할 수 있다[5, 6]. 최근 이러한 SDM의 성능 개선 및 이론에 관한 연구가 다양하게 계속되고 있다[7-9]. 그러나 축약 분산 기억 장치는 일반적인 문제 해결이 가능한 다층 인식자와는 달리 단조 증가 또는

※ 본 연구는 한국과학재단 박사후연수(98후반기)지원으로 수행되었음.

† 공신회원 : 안양대학교 컴퓨터학과 교수

†† 준 회원 : 호서대학교 대학원 전자공학과

††† 공신회원 : 이화여자대학교 컴퓨터공학과 교수

†††† 정 회원 : 호서대학교 전자공학과 교수

논문접수 : 2000년 7월 20일, 심사완료 : 2001년 3월 6일

감소하는 결정 함수에 의해 해 공간이 양분되는 특정 문제에서는 매우 비효율적일 수 있다[3, 4]. 따라서 본 논문에서는 기존의 축약 분산 기억 장치의 위치 주소 선택 과정에 크기 관계를 비교할 수 있는 메커니즘을 부가함으로써, 문제의 해 공간이 단조 결정 함수에 의해 양분되는 특수한 경우 다층 인식자와 같은 일반성을 지니면서 적응성이 뛰어난 새로운 축약 분산 기억 장치의 모델을 제시하고자 한다. 또한 제안된 모델의 유효성을 입증하기 위해 새로운 모델을 ATM망에서의 호 수락 제어에 적용한 예를 보인다. 이하 2장에서 축약 분산 기억 장치의 구조 및 동작 원리를 보이고, 3장에서는 기존 축약 분산 기억 장치의 문제점을 제시하고 그 개선 방안을 제안한다. 4장에서는 이것을 호 수락 제어에 적용한 예를 비교 실험을 통해 보이고 결론을 맺는다.

2. 축약 분산 기억 장치

축약 분산 기억 장치는 기존의 RAM과 같이 임의의 주소로 지정된 장소에 데이터를 읽고 쓴다. 그러나 축약 분산 기억 장치와 RAM은 데이터를 읽고 쓰는 방법에 중요한 차이가 있다. 기존의 RAM에서는 연속적으로 부여되어 있는 주소들 중에서 액세스하고자 하는 단 한 개의 주소를 선택한다. 다시 말해서 이미 부여되어 있는 주소들 중 한 개의 주소를 결정하여 액세스하는데 반하여 축약 분산 기억 장치의 주소는 연속성이 없는 주소를 갖고 있고 특정 데이터의 주소는 그 데이터의 메모리 상의 위치와는 아무런 연관이 없다. 즉 주어진 주소에 대하여 일정 기준을 만족하는 임의의 데이터의 주소를 선택하여 데이터를 읽거나 쓰게 된다. 그러므로 각각의 데이터의 주소를 나타내는 또 다른 저장소가 필요하다. 이같은 구조는 방대한 주소 공간을 갖는 메모리 구성을 가능케한다. 가령 1000비트 이상으로 주소를 지정할 경우 기존의 RAM은 매우 비효율적이거나 비현실적이다. 축약 분산 기억 장치에서는 주소를 저장하는 저장소의 각각의 내용을 위치 주소라고 하고, 이 주소를 참조하여 특정 데이터에 접근할 수 있다[3]. RAM은 액세스하려는 주소(축약 분산 기억 장치에서 참조 주소에 해당)와 액세스되는 주소(축약 분산 기억 장치에서 위치 주소에 해당)가 일대일 대응하고 액세스되는 주소는 변경될 수 없으며 미리 연속적으로 부여되어 있다. 이에 반해 축약 분산 기억 장치는 참조 주소와 위치 주소가 일대일 대응이거나 일대다 대응하고 위치 주소를 저장하는 별도의 저장소가 필요하며 위치 주소의 값은 연속적이지 않고 변경 가능하다. 또 축약 분산 기억 장치에서 각 위치 주소에 대응하여 내용을 저장하는 데이터 부분은 축약 분산 기억 장치의 동작에서 계수기로서 동작한다. (그림 1)은 축약 분산 기억 장치의 일반적인 구조와 RAM의 구조를 나타내고 있다.

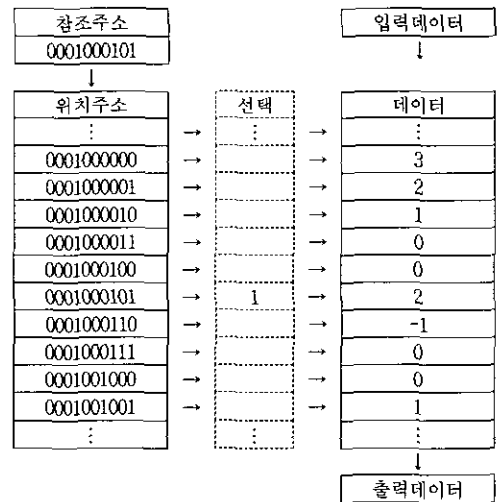
RAM이 액세스하려는 주소와 같은 단 한 개의 데이터

위치를 선택하는 것과는 달리, 참조 주소에 따른 위치 주소의 선택 방법은 참조 주소와 위치 주소간의 해밍 거리를 구하는 것으로부터 시작된다. 즉, 입력되는 참조 주소와 저장되어 있는 모든 위치 주소를 비교하여 각각의 해밍 거리를 구하고, 이 거리가 미리 정해져 있는 반경 내에 들어 있는 위치 주소들을 선택한다. n차원인 참조 주소(x)와 위치 주소(y)의 구성이 x_1, x_2, \dots, x_n 와 y_1, y_2, \dots, y_n 이라 할 때 해밍 거리를 구하는 식은 아래와 같다.

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n|$$



(a) 축약 분산 기억 장치의 구조



(b) RAM의 구조

(그림 1) 축약 분산 기억 장치와 RAM의 구조

예를 들어 호 수락 제어의 경우 주소의 전체 크기를 10bit로 구성하고 상위 6bit는 호 유형1이고, 나머지 4bit는 호 유형 2를 표시한다고 하면 <표 1>과 같은 경우 해밍 거리는 2가 된다.

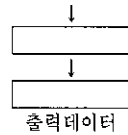
〈표 1〉 비트별 해밍 거리의 계산

	호 유형 1	호 유형 2
참 조 주 소	001101	0100
위 치 주 소	001001	0110
비트별 차이(XOR)	000100	0010
각 비트별 합(해밍거리)	0+0+0+1+0+0+0+0+1+0 = 2	

이와 같은 방법으로 해밍 거리를 구하면 미리 정해진 반경보다 작거나 같은 경우의 모든 위치 주소가 선택되어진다. 여기서 미리 정해진 반경은 입력 패턴과 참조 패턴 사이의 유사 정도의 허용 범위를 의미한다. 이렇게 선택되어진 메모리의 내용(데이터)들을 모두 활성화시켜 쓰기 및 읽기를 실시한다. (그림 2)는 축약 분산 기억 장치의 쓰기 및 읽기

참조주소	반 경	입력데이터
4(n0),5(n1)	3	수락 (1)

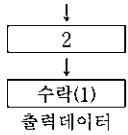
주소	거 리	선택	데이터	
4, 5	→ 0	1	→ +1	계
5, 8	→ 4		→ 0	
0, 1	→ 8		→ 0	치
5, 7	→ 3	1	→ +1	
5, 10	→ 6		→ 0	주
9, 7	→ 7		→ 0	
4, 12	→ 7		→ 0	소
10, 5	→ 6		→ 0	
5, 6	→ 2	1	→ +1	기
8, 9	→ 8		→ 0	
2, 4	→ 3	1	→ +1	기
1, 5	→ 3	1	→ +1	



(a) 쓰기

참조주소	반 경	입력데이터
4(n0),5(n1)	3	

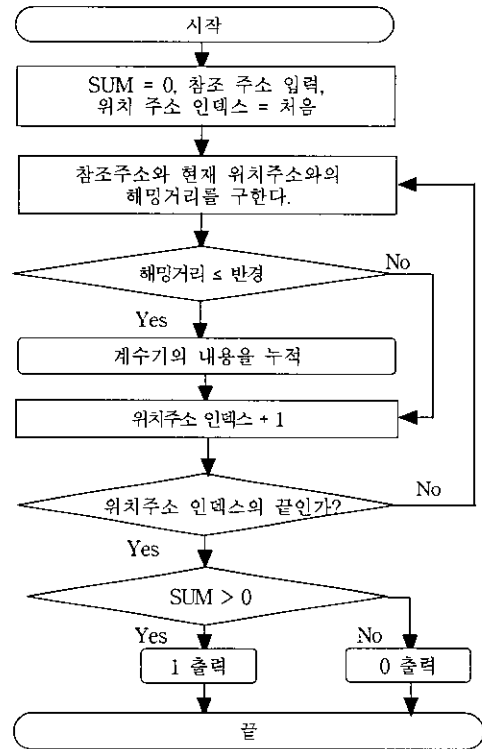
주소	거 리	선택	데이터	
4, 5	→ 0	1	→ 0	계
5, 8	→ 4	0	→ 0	
0, 1	→ 8	0	→ 2	치
5, 7	→ 3	1	→ -1	
5, 10	→ 6	0	→ 0	주
9, 7	→ 7	0	→ 1	
4, 12	→ 7	0	→ 0	소
10, 5	→ 6	0	→ 0	
5, 6	→ 2	1	→ 1	기
8, 9	→ 8	0	→ 0	
2, 4	→ 3	1	→ 0	기
1, 5	→ 3	1	→ 2	



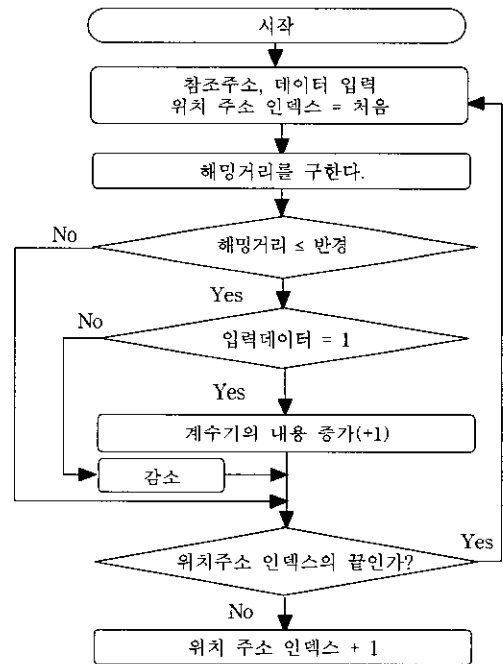
(b) 읽기 동작

(그림 2) 축약 분산 기억 장치의 쓰기 및 읽기 동작

동작을 나타낸 예로 n0과 n1의 두 가지 유형으로 축약 분산 기억 장치를 구성하였고, 이해를 돕기 위해 비트 값 대신에



(a) 읽기 동작



(b) 쓰기 동작

(그림 3) 축약 분산 기억 장치의 동작 알고리즘

정수로 변환하여 나타내었다. 그리고 반경을 3으로 설정하여 입력된 참조 주소와 모든 위치 주소 사이의 해밍 거리를 구

해 반경 3보다 작거나 같은 위치주소를 선택하도록 했다. 쓰기 동작은 쓰기를 원하는 참조 주소로 선택된(반경 내에 위치한) 위치 주소에 입력하려는 데이터가 '1'이면 원래의 내용을 증가(+1)시키고, '0'이면 감소(-1)시킨다.

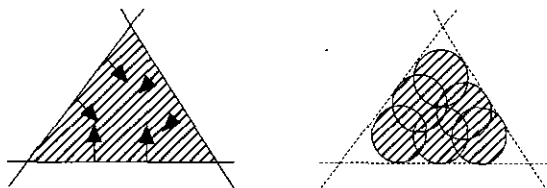
또 읽기 동작은 그림 (b)와 같이 데이터가 계수기로 동작하여 선택된 데이터의 내용들의 비트별 합을 구해서 그 합이 '0'보다 크면 출력을 '1'로 하고 '0'보다 작으면 출력을 '0'으로 한다. 아래의 (그림 3)은 축약 분산 기억 장치의 읽기 및 쓰기 동작 알고리즘을 나타낸다.

이러한 축약 분산 기억 장치의 쓰기 동작은 신경망의 학습 과정(Learning)에 해당하고 읽기 동작은 회상(Recall) 동작에 해당한다. 특히 다층 인식자 신경망(MLP)의 구조와 동작을 축약 분산 기억 장치를 비교하면 각 위치 주소는 다층 인식자의 개별 뉴런에 해당하며 각 위치 주소에 저장되어 있는 내용은 가중치(Weight value)에 해당한다. 다층 인식자에서는 임의의 입력 패턴에 대하여 각 뉴런이 갖고 있는 값과 곱해지는 가중치를 원하는 출력이 되도록 변경하는 과정이 신경망의 학습 과정이다. 이에 비해 축약 분산 기억 장치는 선택된 위치 주소에 저장된 내용, 즉 '계수기' 블록의 값(가중치)을 변경함으로써 학습을 하게 된다. 또 다층 인식자의 회상은 학습에 의해 고정된 가중치와 각 뉴런에 입력되는 패턴을 곱하여 합하고 그것을 활성화함으로써 출력 값을 얻어낸다. 축약 분산 기억 장치의 회상은 선택된 위치 주소에 저장되어 있는 내용을 모두 합하고 그 값을 thresholding하여 출력 값을 구한다.

3. 축약 분산 기억 장치의 문제점과 개선

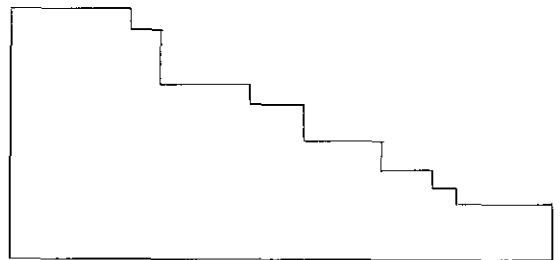
3.1 기존의 축약 분산 기억 장치의 문제점

현재 신경망 모델 중에서 가장 널리 사용되고 있는 다층 인식자는 (그림 4)의 (a) 예에서와 같이 개별 뉴런이 선형 또는 비선형의 결정 함수로 해 공간을 이분하고 그들이 층별로 AND 및 OR 연산으로 다양하게 결합함으로써 임의의 해 공간을 생성하는 능력을 갖고 있어 일반적인 문제에 대한 해결이 가능하다. 이에 비해 축약 분산 기억 장치의 뉴런은 자신을 중심으로 한 일정 반경 영역을 안과 밖으로 이분하고 이들을 단순하게 합함으로써 그림 4에서 다층 인식자에 의해 생성된 그림 (a)의 해 공간은 축약 분산 기억 장치를 이용하여 그림 (b)와 같이 생성할 수 있다.



(그림 4) MLP와 기존 SDM에 의해 생성되는 해 공간의 비교

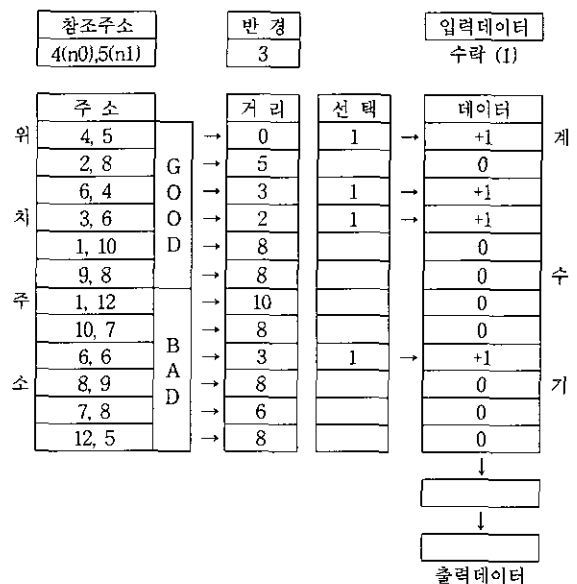
하지만 축약 분산 기억 장치에서는 각 뉴런을 중심으로 일정한 반경에 의한 영역만이 해 영역이 되므로 (그림 5)와 같이 해 영역이 단조 증가 또는 단조 감소 형태로 생성되는 경우에 그 영역이 커지면 뉴런의 수를 늘리거나 반경을 크게 하여야 한다. 이때 뉴런 수의 증가는 시스템 구성에 있어서 기억 장소의 증가와 더불어 처리 속도의 저하와 같은 부정적인 요소가 발생하게 된다. 또 다른 방법으로 축약 분산 기억 장치의 위치 주소를 선택하는 반경의 크기를 크게 하는 방법을 생각해 볼 수 있는데 반경의 크기가 커지면 결정 경계의 정확도가 떨어져 해 공간을 생성하는 과정에서 오차가 커지게 되어 역시 문제가 된다. 따라서 기존의 다층 인식자의 범용성과 축약 분산 기억 장치의 적응성 및 하드웨어 구현의 용이성을 살리기 위해 기존의 축약 분산 기억 장치 뉴런에 반경에 의한 안과 밖 구분 기능과 더불어 크기 관계에 따른 영역 구분 기능을 더할 필요가 있다. 이 경우 반경의 확장 없이 광범위한 해 영역을 하나의 뉴런이 담당할 수 있어 적은 수의 뉴런으로도 정밀한 결정 경계선을 만들 수 있다.



(그림 5) 문제의 해 공간 예

3.2 축약 분산 기억 장치의 개선

본 논문에서 제시하는 새로운 축약 분산 기억 장치는 기



(그림 6) 새로운 축약 분산 기억 장치의 변경된 구조

존 모델의 기본적인 구조와 동작에는 변함이 없다. 그러나 기존 장치의 문제점을 해결하기 위하여 반경을 구하는 부분과 기존의 참조 주소와 위치 주소를 해밍 거리에 의해 비교 선택하는 부분을, 반경 안팎의 비교 뿐 만 아니라 참조 주소와 위치 주소의 차원별 크기 관계까지 비교하도록 수정하였다. 이를 위하여 기존의 축약 분산 기억 장치의 구조를 (그림 6)과 같이 변경하였다.

(그림 6)에서는 축약분산 기억 장치의 패턴 테이블인 위치 주소의 구성이 GOOD과 BAD의 두 개의 부 영역으로 분리되어 있는 것을 볼 수 있다. 이는 결과적으로 해 공간을 구성하는데 있어서 결정 경계 부분에 위치 주소를 중점적으로 저장하게 하여 뉴런들이 결정 경계를 이루게 하도록 크기 관계에 따라 영역을 분리하기 위함이다.

구조의 변경에 따른 새로운 축약 분산 장치의 동작은 다음과 같다. 기존의 축약 분산 기억 장치의 문제를 해결하는데 결정적인 역할을 하는 부분은 새로 입력되는 참조 주소와 각 위치 주소와의 해밍 거리에 의해 위치 주소를 선택하는 방법이다. 기존 축약 분산 기억 장치의 위치 주소 선택 방법과 이에 대한 새로운 축약 분산 기억 장치의 위치 주소 선택 방법을 부 영역이 m 개인 경우로 비교하여 기술하였다. 이외의 읽기 동작과 쓰기 동작은 기존과 동일하다.

<기존 축약 분산 기억 장치의 위치 주소 선택 방법>

1. 참조 주소 $x(x_1, \dots, x_m)$ 와 위치 주소 $y(y_1, \dots, y_m)$ 간의 해밍 거리를 구한다.

$$d(x, y) = \sum_{i=1}^m (|x_i - y_i|)$$

2. 미리 정해진 반경 R과 $d(x, y)$ 를 비교하여 위치 주소를 선택한다.

if $d(x, y) \leq R$ then Select

<새로운 축약 분산 기억 장치의 위치 주소 선택 방법>

1. 참조 주소 $x(x_1, \dots, x_m)$ 와 위치 주소 $y(y_1, \dots, y_m)$ 간의 해밍 거리를 구한다.

$$d(x, y) = \sum_{i=1}^m (|x_i - y_i|)$$

2. 미리 정해진 반경 R과 $d(x, y)$ 를 비교하고 참조 주소와 각 부 영역의 위치 주소간의 대소 관계에 따라 선택한다.

case i) GOOD 영역의 위치 주소 선택

if $((d(x, y) \leq R) \text{ or } (x_i \leq y_i, \text{ for all } i))$ then Select

case ii) BAD 영역의 위치 주소 선택

if $((d(x, y) \leq R) \text{ or } (x_i > y_i, \text{ for all } i))$ then Select

두 가지 변수(부 영역이 2인 경우)를 갖는 임의의 패턴 (n_0, n_1) 에 대한 주소 선택 과정은 다음과 같다. GOOD 영역에 (1,2), (2,2), (3,1)이 BAD 영역에 (2,4), (3,3), (4,2)이 저장되어 있고 반경 R=1 일 경우 (1,1)이 입력되면 ;

1. 새로 입력되는 참조 주소 (1,1)에 대하여 해밍 거리를 구한다.

$$d((1,1),(1,2))=|1-1|+|1-2|=1, d((1,1),(2,2))=|1-2|+|1-2|=2$$

$$d((1,1),(3,1))=|1-3|+|1-1|=2, d((1,1),(2,4))=|1-2|+|1-4|=4$$

$$d((1,1),(3,3))=|1-3|+|1-3|=4, d((1,1),(4,2))=|1-4|+|1-2|=4$$

2. 반경과 각 해밍거리 값의 비교 및 참조 주소와 위치 주소와 대소 비교

case i) GOOD 영역의 선택

(1,2) : $d((1,1),(1,2))=R$ 이므로 선택

(2,2) : $d((1,1),(2,2))>R$ 이지만 $1 < 2, 1 < 2$ 이므로 선택

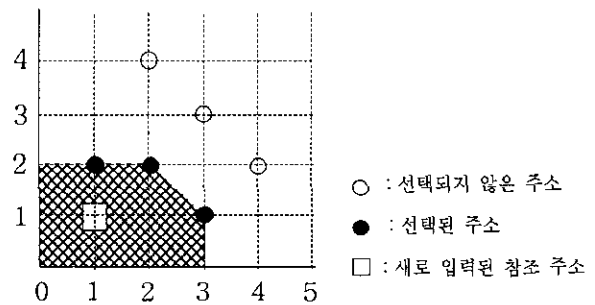
(3,1) : $d((1,1),(3,1))>R$ 이지만 $1 < 3, 1=1$ 이므로 선택

case ii)

(2,4) : $d((1,1),(2,4))>R$ 이고 $1 < 4, 1 < 2$ 이므로 선택
하지 않음

(3,3) : $d((1,1),(3,3))>R$ 이고 $1 < 3, 1 < 3$ 이므로 선택
하지 않음

(4,2) : $d((1,1),(2,4))>R$ 이고 $1 < 2, 1 < 4$ 이므로 선택
하지 않음.

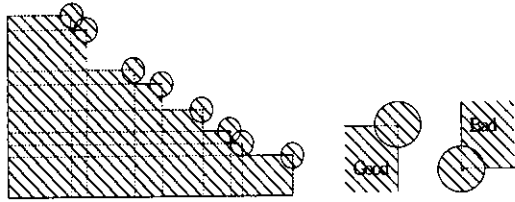


(그림 7) 축약 분산 기억 장치의 위치 주소 선택의 예

(그림 7)은 위의 예에 의해 나타난 결과를 도식화한 것이다. 새로운 참조주소 (1,1)에 의해 선택되는 주소들은 GOOD 영역에 저장되어 있던 (1,2), (2,2), (3,1)이다. 만약 축약 분산 기억 장치의 쓰기 동작이라면 GOOD 영역과 BAD 영역에 관계없이 선택된 위치 주소에 대응한 데이터를 증가하게 되고, 즉 축약 분산 기억 장치의 읽기 동작이라면 선택된 위치 주소에 대응한 데이터를 모두 더하고 그 결과를 thresholding 함으로써 출력 값을 얻어낸다. 여기서 thresholding은 GOOD 영역의 데이터 값들의 누적 값은 양수로 더해지고, BAD 영역의 데이터 값들의 누적 값은 음수로 더해져 그 결과가 양수이면 1을, 그 외에는 0을 출력하게 된다. 그림 7의 빗금 친 부분은 이미 기억된 GOOD 영역의 위치 주소에 의해 회상 과정에서 해 공간으로 인식되는 부분을 표시한 것이다.

이와 같이 축약 분산 기억 장치의 구조와 동작 알고리즘의 변경으로 (그림 5)의 문제 해 공간은 아래 (그림 8) (a)와 같이 해결된다. 그림 (b)는 해 공간에 있어서 새로운 축약 분산 기억 장치에 의한 개별 뉴런의 표현 영역을 나타

낸다. 그림 (b)에서 Good으로 표기된 뉴런은 GOOD 영역의 위치 주소에 저장된 것으로 '1' 영역을 포함함을 알 수 있다. 마찬가지로 Bad로 표기된 뉴런은 BAD 영역의 위치 주소에 저장된 것으로 '2' 영역을 포함한다.



(a) 해 공간 생성 (b) 개별 뉴런이 표현하는 영역

(그림 8) 새로운 SDM에 의한 해 공간 생성

이와 같이 위치 주소 선택 방법을 수정함으로써 단조 증가 또는 감소 형태로 해 공간이 양분되는 경우 적은 수의 뉴런으로 정밀한 결정 경계를 생성하여 효율적으로 문제를 해결할 수 있다.

4. 실험

기존의 방법과 제안된 방법의 성능 비교를 위해 해 공간이 단조 결정 함수로 양분되는 특수한 경우의 예로 ATM망에서의 호 수락 제어에 적용하였으며, 각각에 대해 2종류의 서비스 트래픽에 대한 학습 과정과 호 수락/거절의 결정 영역을 실험하였다.

4.1 축약 분산 기억 장치에 의한 ATM망에서의 호 수락 제어

ATM망에서의 호 수락 제어의 목표는 가능한 한 많은 호를 연결하면서도 새로운 호의 설정에 의해 기존의 호들에 대한 Quality of Service(QoS)를 악화시킬 경우 새로운 호 설정 요구를 거절함으로써 이미 연결되어 있는 여러 호들의 서비스 품질을 유지하는 것이다[12]. 이러한 제어의 핵심은 끊임없이 전송 지연, 셀 손실률 등의 망 상태를 감시하여 새로운 호 설정 가능성을 판단하는데 있다. 이때 신경회로망의 “학습 능력”은 자율적으로 입력들을 선택하고 입력 데이터로부터 특징 변수들을 이끌어 냄으로써 관측된 트래픽으로부터 자동적으로 QoS를 예측하는데 사용할 수 있다[13, 14]. 이는 ATM의 호 수락 제어에 신경망을 사용함으로써 결정 경계선으로 이루어진 해 공간을 생성하게 된다. 이렇게 생성된 해 공간은 앞서 살펴본 (그림 8)과 유사한 결정 경계선을 갖는다. 또 축약 분산 기억 장치는 과거에 발생한 적이 없는 데이터가 발생하면 이전의 학습했던 데이터와 상관없이 독립적으로 기억된다. 즉 선택된 각 위치 주소의 데이터를 증·감하여 학습함으로써 다층 인식자와 같이 별도의 학습 주기와 적응 기간이 필요하지 않다. 그러므로 트래픽 환경의 변화가 심한 ATM망에 대해 적응

력이 우수하다[4, 5]. 이에 새로운 축약 분산 기억 장치를 ATM망에서의 호 수락 제어에 적용하여 실험하였다.

축약 분산 기억 장치를 이용한 ATM망에서의 호 수락 제어에서는 호 연결 요구가 들어오면, 먼저 호 유형을 분류(Class0, Class1)하고 현재 연결된 유형별 호 수에 추가하여 새로운 호 수를 계산함으로써 참조 주소를 생성한다. 이때 새로운 호에 대한 수락 여부는 새로운 참조 주소에 대한 축약 분산 기억 장치의 출력이 0보다 크면 수락, 작으면 거절하게 된다.

학습은 주기적으로 일정 기간 동안의 특정 호 유형 별 셀 손실률의 평균값을 측정하여, 이 값(L(t))을 미리 정해 둔 임계값과 비교하여 실시한다. 즉 특정 호 유형을 참조 주소로 하여 반경 내에 있는 각 위치 주소의 데이터들이 아래 식에 따라 내용을 증·감을 하게 된다.

$$V(t) = \begin{cases} 1, & \text{if } L(t) < L_0 \\ -1, & \text{if } L(t) > L_0 \end{cases}$$

(※ L₀: 임계값, L(t): 시간 t에서의 평균 셀 손실률, V(t): 수락 여부)

다음은 축약 분산 기억 장치에 의한 ATM망에서의 호 수락 제어 알고리즘이다.

- 단계 1. 호의 연결 요구(새로운 호의 발생).
- 단계 2. 현재 ATM의 호 연결 상태 점검(연결이 포화 상태인가). 포화 상태이면 단계 1로.
- 단계 3. 호의 특성에 따른 유형 분류(Class0, Class1).
- 단계 4. 새로운 참조 주소의 생성.
- 단계 5. 축약 분산 기억 장치에 의한 호의 연결 수락 및 거절. 거절할 경우 단계 1로.
- 단계 6. 미리 정의한 반경 R 내에 있는 위치 주소의 내용을 증가.
- 단계 7. 4.항의 새로운 참조 주소가 축약 분산 기억 장치에 존재하는가? 존재하면 단계 1로.
- 단계 8. 축약 분산 기억 장치에 새로운 참조 주소를 추가할 공간이 있는가? 있으면 단계 10으로.
- 단계 9. 가장 적게 나타나는(출현 빈도수가 적은) 위치 주소를 제거.
- 단계 10. 새로운 참조 주소를 축약 분산 기억 장치에 저장.
- 단계 11. 호의 소멸이 발생했는가? 아니면 단계 13으로.
- 단계 12. 현재 연결된 호원 수를 감소.
- 단계 13. 전체 트래픽의 수정. 단계 1로.

여기서 단계 7부터 단계 10까지는 새로운 호의 연결을 수락함으로써 생성된 새로운 위치 주소를 추가시키는 루틴이고, 단계 11부터 단계 13까지는 각 cycle마다 새로운 호를 연결함으로써 부가된 트래픽과 호의 소멸, 그리고 시간에 따른 호의 트래픽 변화를 반영하기 위한 트래픽 수정 루틴이다. 실험에서는 '단계 1의 호 연결 요구' 부분을 두 가지 종류의 호를 비주기적으로 랜덤하게 발생시켜 실제

ATM망에서 일어나는 것과 같이 구현하여 실험하였다.

4.2 2가지 서비스 트래픽에 대한 실험

본 논문의 실험에서는 ATM 망에서 서비스할 수 있는 트래픽 모델 중에서 음성 및 데이터 호원에 대한 모델을 수락 제어하는 실험하였다. 일반적으로 음성 및 데이터의 경우 On 구간과 Off 구간을 각각 352ms와 650ms를 평균으로 하는 지수 분포로 모델링한다. 이와 같은 신호원이 여러 개 다중화 되면 상관성(Correlation)이 매우 큰 복잡한 모델로 나타나는데 이렇게 중첩된 상황을 모델링하는데에는 2-state MMPP (Markov Modulation Poisson Process)를 사용한다 [10]. 또한 2-state MMPP 모델에서 셀이 발생하지 않는 침묵구간을 포함하는 특수한 경우에 IPP 모델로 나타난다[11, 12]. 따라서 실험에서 트래픽 특성이 동일한 호원들을 하나의 트래픽 클래스로 정의하였고 트래픽 클래스의 종류를 크게 2-state MMPP와 IPP로 분류하였다. 그리고 단위 cycle당 시간을 고려하여 초당 발생 비트를 고려할 수 있도록 반영하였고 기존 방법과 새로운 방법의 입력 변수, 즉 축약 분산 장치의 참조 주소로서 ATM 노드의 출력 링크에 정규화되지 않은 평균 부하로 표현되는 트래픽 클래스의 호원 수를 사용하였다. 또 해밍 거리와 비교하는 축약 분산 기억 장치의 반경은 1로 하여 학습에 사용될 위치 주소를 선택하도록 설정하였다. 그 외 축약 분산 장치를 사용한 ATM 호 수락 제어 실험에서 사용한 각 파라미터는 <표 2>에 나타내었다.

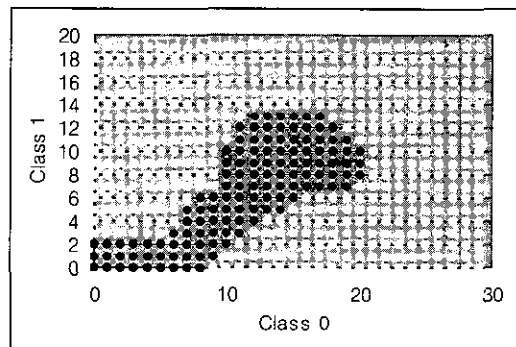
<표 2> ATM 호 수락 제어 실험 파라미터

파라미터	단위	Class0	Class1
트래픽 모델링		IPP	MMPP
평균 비트율	Mbps	1.925	15.33
평균 비트 변동율	%	1.18	0.68
비스트 구간의 평균 데이터 발생	Mbps	12.6	20.4
침묵 구간의 평균 데이터 발생	Mbps	0	5.5
비스트 구간의 평균 지속시간	ms	650	
침묵 구간의 평균 지속시간	ms	340	
trunk capacity	Mbps	250	
1 cycle 당 단위 시간	ms	100	
trunk에 접속 가능자 수	명	30	20
Good Table 수	개	100	
Bad Table 수	개	100	
해밍거리 비교를 위한 반경		1	

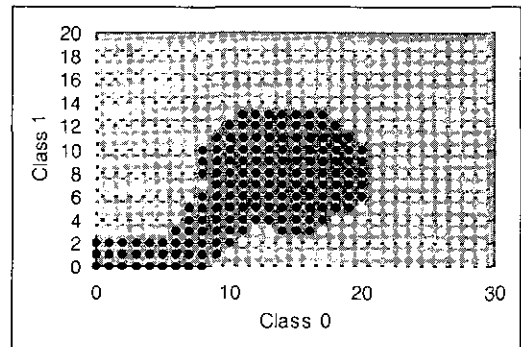
또 실험에서 새로운 데이터 학습시 오래된 패턴들의 영향을 효과적으로 없애기 위해 Leaky pattern table[13, 14]을 사용하였다. 이 방법은 새로운 입력 패턴에 대하여 수락/거절의 결과에 의해 Good table과 Bad table로 나누어 저장하는데 각 테이블에 빈곳이 있다면 그곳에 새로운 패턴을 저장하고, 빈곳이 존재하지 않는 경우에는 가장 오래된 패턴(예를 들면 수락의 경우, Good table에서 가장 오랫동안

참조되지 않은 패턴)을 삭제하여 그곳에 저장한다. 이것은 결국 결정 경계선에서 멀리 떨어진 곳의 패턴을 삭제하게 되는 것으로 결정 경계선에 의해 수락/거절의 영역으로 나누는 해 공간을 형성하는데 그다지 영향을 끼치지 못하는 패턴 대신 새로운 패턴을 대체하는 것을 의미한다.

(그림 9)와 (그림 10)은 각각 기존의 방법과 새로운 방법의 학습 결과를 보여준다. (그림 9)의 (a)와 (b)는 각각 기존의 축약 분산 기억 장치에 의한 10000 cycle까지 학습한 결과와 50000 cycle까지 학습한 결과인데 학습이 진행되면서 그림 (a)에 비해 그림(b)는 중앙 쪽의 수락 영역이 넓어졌고 반면에 좌측 하단 쪽이 약간 좁아진 것을 볼 수 있다. 그런데 그림에서 class0이 15이고 class1이 10일 때 수락 영역에 들어있지만 그 보다 훨씬 작은 class0이 2이고 class1이 4일 때는 거절하는 현상이 나타난다. 이러한 현상은 축약 분산 기억 장치가 학습하는 과정에서 최근의 자료에 의존하여 과거의 자료를 잃어버리기 때문이다. 다시 말해서 최근에 자주 발생한 패턴이 중앙 부분이기 때문에 중앙부분에서 확장이 일어났고 반면에 좌측 하단은 자주 발생하지 않아 축약 분산 기억 장치가 저장하고 있는 위치 주소 테이블에서 참조 회수가 적게 되므로 제거되어 그 영역이 감소되었다. 이때 위치 주소 테이블이 충분히 많았거나 반경이 컸다면 발생 빈도가 적다고 하여 삭제되는 일은

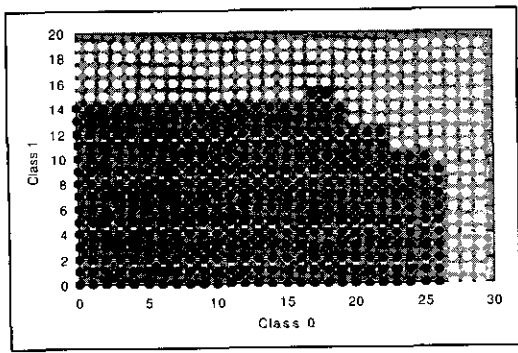


(a) 10000 Cycle(Old SDM)

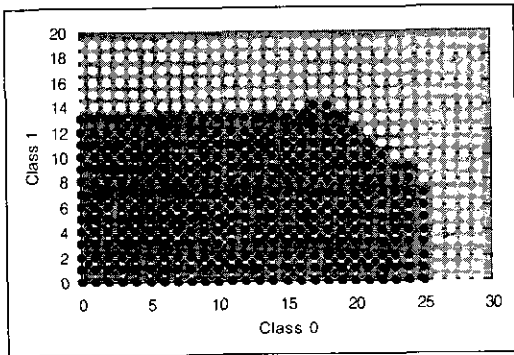


(b) 50000 Cycle (Old SDM)

(그림 9) 기존 축약 분산 기억 장치에 의한 호 수락 제어의 결정 경계선



(a) 10000 Cycle(New SDM)



(b) 20000 Cycle(New SDM)

(그림 10) 새로운 축약 분산 기억 장치에 의한 호 수락 제어의 결정 경계선

일어나지 않는다. 하지만 이것은 앞서 살펴본 바와 같이 시스템 구성에 있어서 부정적인 요소가 된다. 보다 근본적인 원인은 바로 반경을 구하는 데 있다. 축약 분산 기억 장치의 학습은 반경에 의해 위치 주소를 저장하는 테이블을 추가 또는 삭제를 함으로써 이루어지기 때문이다. 이와 같이 기존의 축약 분산 기억 장치는 ATM망에서의 호 수락 제어에서 한정된 위치 주소 테이블 수에 의해 학습 결과가 과거의 내용을 잃어버림으로 갑자기 일어나는 큰 변동에 적절히 대처할 수 없다. 이러한 문제를 해결한 새로운 축약 분산 기억 장치에 의한 ATM호 수락 제어의 결과는 (그림 10) (a)와 (b)에 나타나 있다. 그림 (a)와 (b)는 각각 새로운 축약 분산 기억 장치를 10000 cycle과 20000 cycle까지 수행 후의 결과를 나타낸 것이다. 그림 (a)와 그림 (b)를 보면 학습 과정에서 결정 경계선이 변화된 것을 볼 수 있다. 기존 축약 분산 기억 장치에서도 마찬가지로이지만 학습을 통하여 적절한 해 공간을 형성하는 과정에서 생긴 변화인데, 그 결정 경계선을 보면 기존 축약 분산 기억 장치와는 달리 결정 경계선을 기준으로 좌 하단 방향은 모두 수락 영역을 이루고 있다. 때문에 좌 하단 끝부분을 나타내는 새로운 참조 주소의 입력이 되었을 때, 즉 ATM에서 사용하던 호의 연결 사용이 급격 떨어지는 상황에서 ATM에서 수락을 하는 결과를 보이고 있다. 따라서 기존의 축약 분산 기억 장치의 해 공간 영역 생성에 대한 문제점이 해결되었음을 알 수 있다.

5. 결 론

본 논문에서는 해 공간이 단조 결정 함수로 양분되는 특수한 경우 기존의 축약 분산 기억 장치의 성능을 개선하는 방안을 제시하였다. 다층 인식자를 이용한 기존의 신경망은 범용성과 문제 해결능력에 있어서 뛰어나지만 새로운 데이터가 발생할 경우 이전의 데이터들도 모두 함께 학습을 다시 하여야 하는 불편한 점이 있었다. 또한 그로 인해 작업 속도가 늦어지고 ATM망과 같은 응용에서처럼 급격한 환경 변화에 대한 적응력이 유연하지 못했다. 이러한 문제를 해결하기 위해 축약 분산 기억 장치가 제안되었지만, 이 모델은 학습이 진행됨에 따라 결정 경계가 반경에 의해서 정해지게 되므로 해 공간이 뉴런의 수 또는 반경의 크기에 의존하게 되어 과도한 뉴런을 요구하거나 결정 경계의 정밀도가 떨어지는 문제가 있었다. 이것은 실험 예와 같이 해 공간에 크기 관계가 존재하는 경우 기존에 존재(수락)하던 상태를 기억하는 데 한계를 갖게 하였다. 본 논문에서는 실험 예와 같이 해 공간이 단조 결정 함수로 양분되는 특수한 경우 축약 분산 기억 장치의 뉴런 반경을 반경뿐만 아니라 크기 영역으로까지 확장하여 기존의 문제점들을 해결하였고, ATM의 호 수락 제어 실험을 통해 그 효용성을 보였다. 따라서 새로운 축약 분산 기억 장치는 주어진 제약 조건하에서는 효율적으로 문제 해결이 가능하면서도 적응성이 뛰어난 신경망 모델이라 할 수 있다.

본 논문에서는 기존의 축약 분산 기억 장치를 ATM에 적용하는 데 있어서 발견된 해 공간 생성에 대한 문제점을 해결하였다. 그러나 현재 제안된 모델은 결정 경계가 단조 증가 또는 감소로 나타날 때 유효하다. 따라서 향후 더욱 일반적인 해 공간을 형성할 수 있는 모델로의 개선이 필요하다.

참 고 문 헌

- [1] J. L. McClelland, D. E. Rumelhart, *Parallel Distributed Processing, Vol.1 : Foundations*, Cambridge, MA, MIT Press, 1986.
- [2] A. Jain, B. Wah, "Artificial Neural Networks : A Tutorial," *IEEE Computer*, pp.31-44, Mar., 1996.
- [3] H. Y. Kwon et al, "ATM Call Admission Control Using SDM (II)," *ICNN'98, Proc.*, pp.1799-1803, May, 1998.
- [4] 권희용 외, "축약 분산 기억 장치의 개선", *한국정보과학회 '98 가을 학술발표논문집(II)*, Vol.25, No.2, pp.354-356, 1998.
- [5] Pentti Kanerva, *Sparse Distributed Memory*, Cambridge, MA, MIT Press, 1988.
- [6] Paolo Antognetti, Veljko Milutinovic, *Neural Networks, Concepts, Applications and Implements*, Vol.IV, pp.275-289, Prentice Hall, 1991.
- [7] Karlsson Roland, "A Fast Activation Mechanism for the

Kanerva SDM Memory," SICS Technical Report R95-10, 1995.

[8] S.Gunnar et al, "Algorithms for efficient SDM," Real World Computing Symposium(RWC'97) Proc., pp.215-222, 1997.

[9] S. Gunnar, "The Sparchunk Code : A Method to Build Higher-level Structures in Sparsely Encoded SDM," IJCNN'98 Anchorage Proc., 1998.

[10] H. Heffes and D. M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexers Performance," IEEE JSAC, Vol.4, No.6, pp.856-868, 1986.

[11] H. Arai, M. Kawarasaki and S.Nogami, "An Analysis of Call Admission Control in the ATM Network," 전자정보통신학회 논문지, J72-B-1, pp.1000-1007, 1989.

[12] 강상혁, "ATM 트래픽 제어기술," 전자공학회지, 제19권 제 8호, pp.659-671, 1992.

[13] 이두현, 신요안, 김영한, "다중 서비스 트래픽을 위한 신경회로 망 기반의 ATM 호 수락 제어," 한국통신학회논문지, Vol.21, No.8, pp.1958-1969, 1996.

[14] A. Hiramatsu, "ATM Traffic Control using Neural Networks," Neural Networks in Telecommunications, pp.63-89, 1994.



권희용

e-mail : hykwon@aycc.anyang.ac.kr
 1983년 서울대학교 공과대학 전자계산기 공학과(공학사)
 1985년 서울대학교 대학원 전자계산기공학과(공학석사)
 1993년 서울대학교 대학원 컴퓨터공학과(공학박사)

1986년~1995년 한국통신 연구개발원 선임연구원
 1995년~1997년 안양대학교 컴퓨터학과 전임강사
 1997년~현재 안양대학교 컴퓨터학과 조교수
 관심분야 : 신경망, 소프트웨어컴퓨팅, 패턴인식



장정우

e-mail : jfox@shinbiro.com
 1998년 호서대학교 수학과 졸업(이학사)
 2000년 호서대학교 전자공학과 대학원 졸업(공학석사)
 2001년 현재 (주)테크노믹스에 근무
 관심분야 : 신경회로망, 이미지 프로세싱



임성준

e-mail : hokahie@dreamwiz.com
 1998년 호서대학교 전자공학과 (공학사)
 2001년 호서대학교 대학원 전자공학과 (공학석사)
 2001년 현재 호서대학교 대학원 전자공학과 박사과정 재학 중

2000년~현재 (주) C.I.brain 연구원
 관심분야 : Linux를 이용한 Embedded System, Device Driver Programming(System Programming), Realtime OS.



조동섭

e-mail : dscho@ewha.ac.kr
 1979년 서울대학교 공과대학 전기공학과 (공학사)
 1981년 서울대학교 대학원 전기공학과(공학석사)
 1986년 서울대학교 대학원 컴퓨터공학과 (공학박사)

1985년~1990년 이화여자대학교 전자계산학과 조교수
 1990년~1995년 이화여자대학교 컴퓨터학과 부교수
 1995년~현재 이화여자대학교 컴퓨터학과 교수
 1996년~1997년 Univ. of California, Irvine 객원교수
 관심분야 : 컴퓨터 시스템구조, 웹기반 지식공학, 멀티미디어 응용, 전자상거래 시스템



황희용

e-mail : hyhwang@office.hoseo.ac.kr
 1960년 서울대학교 공과대학 전기공학과 (공학사)
 1964년 서울대학교 대학원 전기공학과 (공학석사)
 1974년 서울대학교 대학원 박사과정졸업 (공학박사)

1976년~1980년 서울대학교 공과대학 전자계산소장
 1980년~1983년 서울대학교 전자계산소장
 1983년~1984년 미국FIT 공과대학 객원교수로 연구
 1992년~1993년 서울대학교 컴퓨터공학과 교수
 1993년~현재 호서대학교 전자공학과 교수
 관심분야 : MP3 Player/MIDI, Image Retrieval, SDM, Computer interfacing