

---

碩士學位論文

神經回路網을 이용한 비터비 復號  
알고리즘 變換에 관한 研究

濟州大學校 大學院

通信工學科



1995年 12月

神經回路網을 利用한 비터비 復號  
알고리즘 變換에 關한 研究

指導教授 李 鎔 鶴

全 英 姬

이 論文을 工學 碩士學位 論文으로 提出함

1995 年 12 月

全 英 姬의 工學 碩士學位 論文을 認准함



제주대학교 중앙도서관  
JEJU NATIONAL UNIVERSITY LIBRARY

審査委員長 張 宰 成

印

委 員 金 興 洙



委 員 李 鎔 鶴



濟州大學校 大學院

1995 年 12 月

---

**A Study on the Viterbi Decoding Algorithm  
Transform Using Neural Networks**

**Young – Hee Jun**

**(Supervised by professor Yong – Hak Lee)**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF MASTER  
OF ENGINEERING**



**DEPARTMENT OF COMMUNICATION ENGINEERING  
GRADUATE SCHOOL  
CHEJU NATIONAL UNIVERSITY**

**1995. 12.**

# 목 차

Abstract .....	1
I. 서 론 .....	2
II. 콘블루션 부호기와 비터비 복호 알고리즘 .....	5
1. 콘블루션 부호기 .....	5
2. 비터비 복호 알고리즘 .....	6
3. 격자도 통합 .....	8
4. 비터비 복호 알고리즘의 행렬표현 .....	10
III. 이진 패턴 분류를 위한 신경 회로망 .....	16
1. 신경 회로망 .....	16
2. 호필드망 .....	19
3. 해밍망 .....	21
4. 비터비 복호 알고리즘에 해밍망 적용 .....	25
IV. 결과 및 고찰 .....	27
1. 기존의 비터비 복호 결과 .....	27
2. 해밍망을 적용한 비터비 복호 결과 .....	36
V. 결 론 .....	38
참 고 문 헌 .....	40

---

## Abstract

A great number of papers have introduced implementation of Viterbi decoding algorithm for convolution codes. But it is problems that long constraint length codes need many storages and hardware implementation is complicated.

In this paper, we analyzed the Viterbi algorithm and represented it with simple matrix operations. We also analyzed the Hamming neural network which had a parallelism. The procedure computing the Hamming distance between the received codes and codewords applied to lower subnet and selection of the minimum Hamming distance used upper subnet in the Hamming neural network. We proved that the information sequences decoded with conventional Viterbi decoding Algorithm were equal to their with Viterbi decoding Algorithm using Hamming neural network.

## I. 서 론

위성 통신이나 이동 통신 등에서 이용되고 있는 아날로그 시스템은 채널 용량의 한계성과 정보의 신뢰성이라는 관점에서 많은 문제가 있었다. 디지털 통신 시스템은 채널 용량의 한계를 극복하고 정보의 신뢰성을 보장하기 위하여 아날로그 신호를 이산화하는 소스코딩과 채널상의 에러를 정정하기 위한 채널 코딩의 과정을 거쳐 데이터를 전송한다.(Clark와 Cain, 1981)

채널상에서 발생하는 에러 정정을 위한 채널 코딩은 메모리가 필요없는 블록 코딩과 메모리를 필요로 하는 콘볼루션 코딩으로 크게 나눌 수 있다. 블록코딩은 패리티를 추가하는 방법으로 선형 블록 코드, 순환 코드, BCH 코드 등이 제안되었으며, 콘볼루션 코딩은 입력 데이터를 메모리에 있는 기존의 데이터와 상관 관계를 갖게 부호화하는 방법이다.(Rhee, 1989), (Lin과 Costello, 1983) 블록 코딩에 비하여 에러 정정 효율이 우수한 콘볼루션 부호가 주로 사용되고 있다. 콘볼루션 부호는 복호시에 많은 양의 메모리를 필요로하며, 복호화 지연이 발생하기 때문에 실시간 처리가 가능한 복호 알고리즘이 필요하게 되었다. 이에 따라 콘볼루션 부호를 복호하기 위한 최우 복호 알고리즘인 비터비 복호 알고리즘이 제안되었는데, 이 알고리즘은 위성 통신, 문자와 음성 인식 등의 여러 분야에서 응용되고 있으며, 그 응용 범위가 증가되고 있는 추세이다. 그러나 비터비 복호 알고리즘의 문제점은 구속장의 길이가 긴 코드에서 지나치게 많은 기억장소를 필요로하고 하드웨어의 구현이 복잡하다는 것이다. 하드웨어의 구현을 효율적으로 하고 계산시간을 줄이기 위한 복호 알고리즘이 필요하게 되었다.

신경 회로망은 생명체의 신경조직으로 부터 착안하여 모델화한 정보 처리 시스

템으로서 단순한 소자들의 병렬, 분산 연결구조를 가지고 있으며, 외부로부터 여러 개의 입력을 받아들여 동적반응을 일으킴으로서 필요한 출력을 생성시키는 것이다. 신경 회로망은 인공두뇌를 의미하는 것이 아니라 단지 종래의 노이만 (Von Neumann)식의 순차 처리 시스템인 디지털 컴퓨터의 한계를 극복하여 병렬 분산 처리 시스템으로 접근하는 것이라 할 수 있다.(Lippman, 1989) 신경 회로망의 여러 모델들은 단순 노드들간의 결선방법과 학습방식에 따라 구별된다. 인공 신경 회로망은 음성과 영상 인식 분야에서 데이터나 패턴들을 기억하고 회상하기, 패턴들을 분류하기, 입력패턴과 출력패턴들을 대응시키기, 유사한 패턴끼리 모으기, 구속하고 있는 최적 문제들에 대한 해를 찾기와 같은 다양한 문제들에 적용될 수 있다.

디지털 통신 시스템은 유한 이진 패턴 집합을 전송하는 것으로 간주할 수 있으므로 신경회로망의 분류기능을 콘볼루션 부호를 복호하는데 응용할 수 있다. 호필드망과 해밍망은 불규칙적으로 일그러진 이진 패턴들을 분류하는 최적 분류기로 구현된 신경 회로망이다. 무기억 이진 대칭 채널을 통해서 이진 고정 길이 신호를 보낼 때 최적 최소 에러 분류기는 각 클래스의 표본에 대한 해밍거리를 계산하여 최소 해밍거리를 가진 클래스를 선택한다.(Lippman, Gold, Malpas, 1987)

본 논문은 콘볼루션 부호와 비터비 복호 알고리즘을 분석하여 비터비 복호 알고리즘의 계산횟수와 계산시간을 줄이기 위한 격자도 단계를 통합하고, 비터비 복호 알고리즘을 간단한 행렬식으로 유도한다. 그리고 이진 패턴을 분류하는 데 이용되고 있는 대표적인 신경 회로망에 대하여 기술하고, 기존의 비터비 복호 알고리즘에서 격자도의 부호어와 수신어 간의 해밍거리가 최소인것을 결정하는 비터비 복호 알고리즘에 해밍망을 적용하여 하부망에서 해밍거리를 계산하고 상부망에서 최단경로를 가진 노드를 선택하도록 제안한다.

본 논문의 구성은 I장에서 신경 회로망을 이용한 비터비 복호 알고리즘에 관한 연구의 필요성과 방향을 제시하고, II장에서 콘볼루션 부호와 비터비 복호 알고리즘의 원리와 격자도 통합에 따른 변형된 비터비 복호 알고리즘에 관하여 설명하고, III장에서 이진 패턴 분류에 응용되고 있는 인공 신경 회로망에 대하여 기술하고 비터비 복호 알고리즘에 적용하기 위한 식을 제시한다. IV장에서 행렬식으로 표현된 비터비 복호 알고리즘으로 복호된 결과와 신경 회로망을 적용하여 복호된 결과가 동일함을 보이고 격자도를 통합한 후의 비터비 복호 계산시간이 줄었음을 입증한다. 마지막으로 V장에서 본 논문의 결론을 맺는다.





## II. 콘볼루션 부호기와 비터비 복호 알고리즘

디지털 통신 시스템에서 송신측은 정보 데이터에 여분의 부호를 첨가하여 전송하는 방법으로서 메모리에 기억된 기존의 데이터와 상관관계를 갖고 있는 콘볼루션 부호로 부호화한 후 채널을 통하여 전송한다. 수신측에서는 수신된 부호를 가지고 채널상에서 발생한 에러를 정정하여 정확한 데이터를 복호하는 비터비 복호 알고리즘을 많이 사용하고 있다. 그러나, 비터비 복호 알고리즘은 구속장의 길이가 긴 코드의 경우 많은 기억장소를 필요로 하고 하드웨어 구현이 복잡하며 복호화 지연이 발생하는 문제점이 있다. 1절에서 콘볼루션 부호기, 2절에서 비터비 복호 알고리즘, 3절에서 통합된 격자도에서 비터비 복호 알고리즘을 분석하고, 4절에서 비터비 복호 알고리즘을 행렬표현식으로 유도한다.

### 1. 콘볼루션 부호기



제주대학교 중앙도서관  
JEJU NATIONAL UNIVERSITY LIBRARY

Fig. 1에 보이는 바와 같이  $(n,k,m)$ 인 콘볼루션 부호기는  $k$ 개의 입력,  $X_1, X_2, \dots, X_k$  으로 하여  $n$ 개의 출력,  $Y_1, Y_2, \dots, Y_n$ 을 보내는 유한 상태기다. 각각의 입력노드와 연결된 부호기는 쉬프트 레지스터와 Modulo-2 덧셈기로 구성되어 있다.(Rhee, 1989) 입력 비트수인 구속장  $k$ 와 입출력 비트수  $n$ 의 비율인 코드비  $R$ 에 의하여 부호기의 출력과 구조가 결정된다. 덧셈기의 연결점은 생성벡터나 생성다항식으로 표현하는 대수학적 방법과 상태도나 격자도 등으로 나타내는 위상학적 방법이 있다.(Clark와 Cain, 1981) 상태도는 콘볼루션 부호기의 레지스터 값을 하

나의 상태로 하여 각 입력 값에 따른 상태의 변화를 나타낸 것이며, 상태의 변화를 이산적인 시간축의 함수로 나타낸 것이 격자도이다.

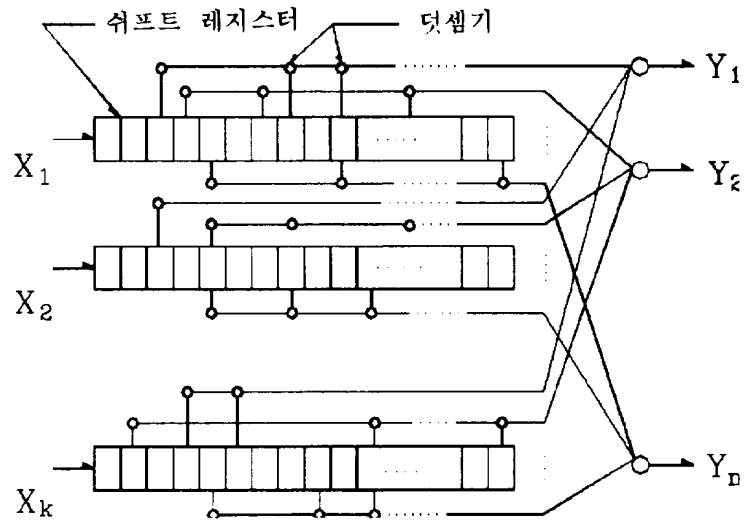


Fig. 1 Convolutional encoder

## 2. 비터비 복호 알고리즘

(n,k,m)인 콘볼루션 부호의 격자도에는  $N = 2^{km}$ 개의 상태가 있게 된다. 상태  $S_i$ 에서 다음 단계의 상태  $S_j$ 를 연결하는 가지를  $e_{ij}$ 로 나타낸다. Fig. 2에서 (n,1,3)인 콘볼루션 부호이므로 상태의 수  $N = 8$ 이다. 시점  $t$ 에서 어떤 상태  $S_j$ 에 연결되는  $2^k$ 개의 후보경로들이 있다. 각 경로는 시점  $t-1$ 의  $S_i$ 와 연결되는 생존 경로,  $S_i$ 와 시점  $t$ 의  $S_j$ 간의 연결가지  $e_{ij}$ 로 구성되어 있다.  $J(j)$ 는 격자도상의 각 상태의 가지  $e_{ij}$ 에 대하여  $S_j$ 와 연결되는 이전 단계의 상태  $S_i$ 의 첨자들의 집합으

로서 그 수는  $2^k$ 개가 된다. 예를 들면, Fig. 2에서 시점  $t+3$ 의 상태  $S_4$ 의  $J(4)=\{2,6\}$ 이다.

각 상태에서  $S_j$ 의 모든 후보경로에 대한 경로 평가량  $pm(i, j, t)$ 을 측정 함수  $f$ 를 거쳐서 계산하고( $i \in J(j)$ ), 그것들 중에서 최단 경로 평가량을 생존 평가량

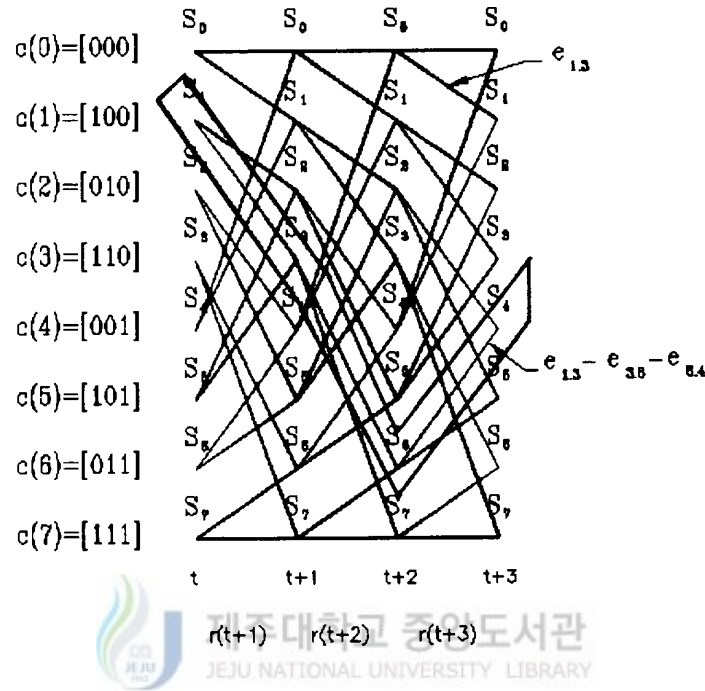


Fig. 2 Trellis diagram of an eight-state convolution code

$sm(j, t)$ 으로 선택한다.  $r(t)$ 를  $t$ 번째 시점에서 수신된  $n$ -비트부호어이고,  $w(i, j)$ 는 임의의 상태로 전이될 때의 부호기의 출력으로서  $e_{ij}$ 에 대응하는 부호어이다.  $pm(i, j, t)$ 와  $sm(j, t)$ 는 다음과 같은 관계가 성립한다.

$$pm(i, j, t) = f\{ r(t), w(i, j), sm(i, t-1)\} \tag{1}$$

$$sm(j, t) = X(pm(i, j, t)) \quad (= \max(pm(i, j, t); \forall i \in J(j))) \tag{2}$$

$X$ 는 최단경로 비교기이고,  $i \in J(j)$ 인 모든 상태  $S_i$ 에 적용된다.

식(2)에서,  $S_j$ 의 생존 평가량을 계산할 때 필요한 오퍼랜드는  $r(t)$ ,  $\{w(i,j) | i \in J(j)\}$ ,  $\{sm(i,t-1) | i \in J(j)\}$ 이다.

즉, 비터비 알고리즘은 복호기의 입력  $r(t)$ 와 부호기의 격자도 기준값  $w(i, j)$  사이의 해밍거리인 가지 평가량을 누적하여 경로 평가량을 구한다. 각 상태에서 경로 평가량은 들어오는 가지 중에서 전 단계의 생존 평가량과 현단계의 가지 평가량을 합한 것 중 가장 작은 것을 선택하여 얻어진다. 선택된 가지를 생존자라고 하며 가장 작은 경로 평가량을 갖는 상태에서 시작하여 생존자를 역추적함으로써 복호기의 출력을 결정한다.

예를 들어 시점  $t+3$ 에서  $S_4$ 의 경로 평가량  $pm(6,4,t+3)$ 은 다음과 같이 단계마다 반복해서 계산해야 한다.

$$\begin{aligned}
 pm(6,4,t+3) &= f(r(t+3),w(6,4),sm(6,t+2)) \\
 &= f(r(t+3),w(6,4),X(pm(i,6,t+2))) \\
 &= f(r(t+3),w(6,4),pm(3,6,t+2)) \\
 &= f(r(t+3),w(6,4),f(r(t+2),w(3,6),sm(3,t+1))) \\
 &= f(r(t+3),w(6,4),f(r(t+2),w(3,6),X(pm(i,3,t+1)))...).
 \end{aligned}$$

각 단계마다 동적처리 하여,  $i \in J(j)$ 를 만족하는 모든  $S_i$ 중에서 생존 평가량은  $S_j$ 의 생존 평가량을 갱신하는데 이용해야 한다. 이것은 VLSI회로에서 데이터를 역세스할 때 가장 많은 시간을 소비하게 하는 주요인이 된다.

### 3. 격자도의 통합

위상학적으로 수정 보완하여, 여러 단계의 격자도를 한단계의 격자도로 통합할

수 있다.  $(n,1,3)$ 인 콘블루션 부호의 통합 격자도를 Fig. 3(a)와 (b)에 나타내었다. Fig. 3(a)는 두 단계씩 통합한 것이고 각 가지는 Fig. 2의 격자도에서 2-가지 경로에 대응된다. Fig. 3(b)는 통합된 3-가지 경로를 설명하고 있다. Fig. 2의 테두리 속에 있는 경로들을 참조해 보면,  $pm(1,3,t+1)$ 을  $(t+1)$ 시점에서  $S_3$ 에 대한 생존 평가량으로 결정하고  $pm(3,6,t+2)$ 를  $(t+2)$ 시점에서  $S_6$ 에 대한 생존 평가량으로 결정한다. 그리고  $e_{1,3}$ ,  $e_{3,6}$ ,  $e_{6,4}$ 로 구성된 3가지 경로,  $p_{6,4,t+3}$ 는  $(t+3)$ 시점의  $S_4$ 의 후보경로 중 하나이다.

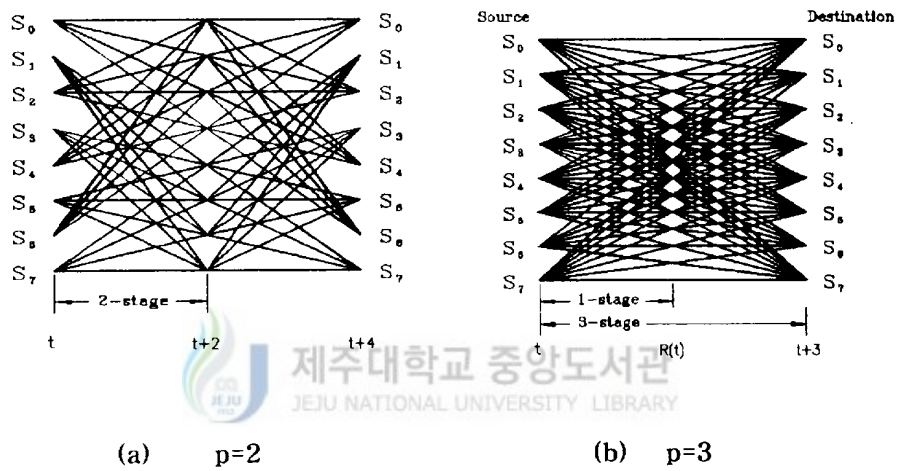


Fig. 3 The integrated trellis diagram

$p$  단계를 하나의 단계로 통합하면, 격자도에서  $p$ 개의 연속하는 가지들은 한 개의 통합된 가지로 변형되고, 한번의 계산과정으로  $p$ 개의 연속하는 가지들의 평가량 합을 계산할 수 있다. 새로운 경로 평가량을 형성하기 전에  $p$ 개의 단계들에서 계산된 생존 평가량과 합쳐질 수 있다면, 그때 각 상태에 대한 생존 평가량은 각  $p$ 단계마다 갱신될 것이고 데이터 액세스의 수는  $p$ 번 감소된다.


그러나, 통합 격자도와 원래의 격자도에서 가지/경로/생존자 계산들이 호환성을 유지하기 위하여, 통합된 격자도의 가지는 원래 격자도의 p-가지 경로에 1대 1 로 대응해야 한다는 제한이 있다.

#### 4. 비터비 복호 알고리즘의 행렬표현

기존의 비터비 복호 알고리즘과 격자도 통합에 따른 비터비 복호 알고리즘을 간단한 행렬식으로 표현하고자 한다.

##### 1) 상태 코드 정의

(n,l,m)인 코드는 하나의 입력, m개의 레지스터, n개의 출력이 있다. 레지스터의 수가 m이기 때문에, 부호기의 조합 가능한 상태수는  $N = 2^m$ 이다. 각 상태  $S_j$  를 j의 이진표현인 m-비트 상태 코드로 나타낸다. 즉, 다음의 관계로 표현할 수 있다.



제주대학교 중앙도서관  
 $C(j) = S_j$ 의 상태코드 = j의 이진표현

##### 2) 가지 부호어 생성

###### (1) 1단 가지 부호어 생성 공식

부호기의 주요 기능은  $S_i$ 에서  $S_j$ 로 가능한 상태 전이들에 따라서 가지 부호어  $w(i, j)$ 를 생성하는 것이다. 부호과정을 기초로 하여, 가지 부호를 생성하기 위하여 오퍼랜드벡터  $in(i, j)$ 는 m개의 레지스터 내용(C(i))과  $S_i$ 에서  $S_j$ 로 상태 전이를 야기시키는 입력비트의 내용으로 구성되어 있다. 입력비트가 부호기의 왼쪽부터 적재되므로 m개의 레지스터의 비트 조합은 C(i)에서 C(j)로 변환된다. C(j)의 가장

왼쪽 비트 LMB(j)는 S<sub>i</sub>에서 S<sub>j</sub>로 상태 전이를 일으키는 입력비트이다. 즉,

$$\text{in}(i, j) = [\text{LMB}(j)\backslash\text{C}(i)] \quad (3)$$

여기서  $[\ ]$ 는 연속을 의미하고  $w(i, j)$ 는 다음과 같이 일반화될 수 있다.

$$w(i, j) = \text{in}(i, j) * G \quad (4)$$

여기서  $*$ 는 GF(2)에서 정의한 행렬곱이다.  $G$ 는 부호를 정의하는  $k(m+1) \times n$

행렬이다.

$$G = \begin{bmatrix} G_0 \\ \dots \\ G_i \\ \dots \\ G_m \end{bmatrix} \quad G_r = \begin{bmatrix} g_{1,r}^1 & g_{1,r}^2 & g_{1,r}^3 & \dots & g_{1,r}^n \\ g_{2,r}^1 & g_{2,r}^2 & g_{2,r}^3 & \dots & g_{2,r}^n \\ \dots & \dots & \dots & \dots & \dots \\ g_{k,r}^1 & g_{k,r}^2 & g_{k,r}^3 & \dots & g_{k,r}^n \end{bmatrix} \quad (5)$$

요소  $g_{i,r}^j$ 는  $g_i^j$ 의  $r$ 번째 비트,  $X_i$ 와  $Y_j$ 에 관한 생성기열이다.(Rhee, 1989)

예를 들어 (3,1,3)콘볼루션 부호를 다음과 같이 가정하면,

$$G = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

가지 부호어  $w(i, j)$ 는 다음과 같이 된다.

$$w(1, 3) = [\text{LMB}(3)\backslash\text{C}(1)] * G = [1\backslash100] * G = [101] \quad (7a)$$

$$w(3, 6) = [\text{LMB}(6)\backslash\text{C}(3)] * G = [0\backslash110] * G = [110] \quad (7b)$$

$$w(6, 4) = [\text{LMB}(4)\backslash\text{C}(6)] * G = [0\backslash011] * G = [100] \quad (7c)$$

(2)  $m$ 단 가지 부호어 생성 공식

통합된  $m$ 개의 단계들과  $m$ 개의 레지스터의 내용이  $C(j)$ 로 변하는 동안  $m$ 개의





w(3, 6), w(6, 4)들이 역으로 연결되어 있다. 즉

$$W(1, 4) = [w(6, 4) \setminus w(3, 6) \setminus w(1, 3)] \quad (12)$$

가지 부호어 생성을 위한 공식을 좀 더 축소하기 위하여  $G_m$ 을  $m \times nm$ 행렬인  $G_{m_u}$ 와  $G_{m_L}$ 로 분리한다.  $G_{m_u}$ 와  $G_{m_L}$ 은 각각  $G_m$ 의 상반과 하반부이다.

상태  $S_j$ 로 들어오는 안쪽가지에 대한  $W(i, j)$ 를 계산할 때  $C(j)$ 와  $G_{m_u}$ 의 부분 행렬 곱은 고정되고, 식(9)에서  $T(j)$ 로 분리할 수 있다. 즉,

$$\begin{aligned} W(i, j) &= IN(i, j) * G_m \\ &= [C(j) \setminus C(i)] * \begin{pmatrix} G_{m_u} \\ G_{m_L} \end{pmatrix} \\ &= [C(j) * G_{m_u}] \oplus [C(i) * G_{m_L}] \quad (13) \\ &= T(j) \oplus [C(i) * G_{m_L}] \end{aligned}$$

위 식에서  $\oplus$ 는 베타적 OR연산자이다. 증가된 차원을 피승수벡터에 더해서 식(13)을 식(14)와 같이 간단하게 나타낼 수 있다.

$$W(i, j) = [C(i) \setminus 1] * \begin{pmatrix} G_{m_L} \\ T(j) \end{pmatrix} \quad (14)$$

### 3) 가지 평가량 계산

이진 대칭 채널의 경우, 부호어  $w(i, j)$ 와 수신어  $r(t)$ 는  $n$ 차 벡터이다. 가지 평가량  $bm(i, j, t)$ 를  $w(i, j)$ 와  $r(t)$ 간의 해밍거리로부터 구할 수 있다.  $bm(i, j, t)$ 가  $r(t)$ 와  $t$ 시점의 모든 부호어 간의 해밍거리들 중에서 최소 해밍거리라면 최우 복호

기는  $w(i, j)$ 를 전송 부호어로 선택한다. 가지 평가량은 다음과 같이 표현할 수 있다.

$$\begin{aligned}
 b_m(i, j, t) &= r(t) \otimes w(i, j) \times I^T \\
 &= r(t) \otimes [in(i, j) * G] \times I^T \\
 &= [in(i, j) \setminus 1] * \left( \frac{G}{r(t)} \right) \times I^T
 \end{aligned} \tag{15}$$

$I$ 는  $n$ 차 단위 벡터  $I = [111 \dots 1]$ 이다. 식(15)을  $m$ 단 통합 가지 평가량을 구하는 데 적용하려면, 연속하는  $m$ 단에 수신된 부호를 고려해야 한다.  $R(t) = [r(t) \setminus r(t-1) \dots r(t-m+1)]$ 라고 하면,  $m$ 단 통합 가지 평가량은 다음과 같이 쓸 수 있다.

$$\begin{aligned}
 B_m(i, j, t) &= (W(i, j) \otimes R(t)) \times I_m^T \\
 &= \left[ [C(i) \setminus 1] * \left( \frac{Gm_L}{T(j)} \right) \otimes R(t) \right] \times I_m^T
 \end{aligned} \tag{16}$$

$$= [C(i) \setminus 1] * \left( \frac{Gm_L}{T(j) \setminus R(t)} \right) \times I_m^T$$

여기서  $I_m$ 은  $I$ 로 부터 확장된  $nm$ 차원 단위벡터이다.

#### 4) 경로 평가량 계산

가지 평가량을 계산한 후, 경로 평가량  $pm(i, j, t)$ 은  $b_m(i, j, t)$ 와  $sm(i, t-1)$ 의 합으로 다음과 같이 나타낸다.

$$sm(j, t+1) = X(pm(i, j, t+1)) \tag{17}$$

$$pm(i, j, t+1) = f(r(t), w(i, j), sm(i, t)) \tag{18}$$

$$= bm(i, j, t+1) + sm(i, t)$$

m단으로 통합되었을 때, 각 상태에 대한 J(j)는 0에서 N-1까지의 모든 가지수를 포함하고 있다. 통합된 격자도의 경로 평가량과 생존 평가량은 각각 Pm(i, j, t)와 Sm(j, t)로 나타낸다. Pm(i, j, t)를 계산하고 나서 연속으로 Sm(j, t)을 구하는데 i는 0에서 N-1까지의 X연산을 한다.

$$Sm(j, t) = X(Pm(i, j, t)) \quad (19)$$

$$Pm(i, j, t) = Bm(i, j, t) + Sm(i, t-m) \quad (20)$$

$$\begin{aligned} &= [C(i)|I1] * \begin{pmatrix} \mathbf{Gm}_L \\ \mathbf{T}(j) \\ \mathbf{Re}(t) \end{pmatrix} \times I_m^T + Sm(i, t-m) \\ &= [C(i)|I1] * \begin{pmatrix} \mathbf{Gm}_L & | & 0 \\ \mathbf{T}(j) & | & 0 \\ \mathbf{Re}(t) & | & 1 \end{pmatrix} \times [I_m | Sm(i, t-m)]^T \end{aligned}$$

행렬 연산  $A[BC] = A * B \times C^T$  라고 정의하면, 식(20)은 다음과 같이 쓸

수 있다.



$$Pm(i, j, t) = [C(i)|I1] \begin{pmatrix} \mathbf{Gm}_L & | & 0 \\ \mathbf{T}(j) & | & 0 \\ \hline \mathbf{Re}(t) & | & Sm(i, t-m) \end{pmatrix} \quad (21)$$

그러므로, 부호어 생성, 통합된 격자도의 각 상태에 대한 가지 평가량과 경로 평가량은 식(21)에 주어진 간단한 행렬 연산에 의해 구해진다.

### 5) 역추적 과정 제거

기존의 비터비 복호에서, 역추적 과정을 통해 최종 복호된 부호를 추적해야 한다. sm(j, t)를 계산할 때, 행렬식으로 표현된 비터비 복호는 가장 큰 경로 평가량

으로 전이하도록 했던 이전 상태의 전이 입력 비트를 기록해 둔다.

$$U(j, t) = C(q) \text{ iff } X(pm(i, j, t)) = pm(q, j, t) \quad (22)$$

여기서  $U(j, t)$ 는 최우 복호 부호이고  $C(q)$ 는 이전 상태 비트이다. 이와 같이  $sm(j, t)$ 가 모든 상태들에 대한 생존자 평가량 중에서 최대값이라면,  $U(j, t)$ 로부터 직접 읽어낼 수 있다. 따라서 기존의 비터비 복호 알고리즘의 역추적 과정을 생략할 수 있다.



### Ⅲ. 이진 패턴 분류를 위한 신경 회로망

신경 회로망은 데이터나 패턴들을 기억하여 회상하기, 패턴들을 분류하기, 입력 패턴과 출력패턴들을 대응시키기, 유사한 패턴끼리 모으기, 구속하고 있는 최적 문제들에 대한 해를 찾기 등과 같은 다양한 문제들에 적용될 수 있다. 반면, 디지털 통신 시스템은 유한 이진 패턴 집합을 전송하는 것으로 간주할 수 있으므로 신경 회로망의 분류 기능을 콘볼루션 부호를 복호하는데 적용할 수 있다. 따라서 1절에서 신경 회로망의 단순 소자인 인공 뉴런의 동작을 생물학적 뉴런의 활동과 대조하면서 설명하고, 이진 패턴 분류를 위한 신경망 분류기로서 가장 많이 이용되고 있는 신경 회로망인 호필드망과 해밍망에 관하여 2, 3절에서 기술한다. 그리고 4절에서 호필드망과 해밍망을 비교분석한 후 해밍망에 비터비 복호 알고리즘을 적용한다.



#### 1. 신경 회로망

신경 회로망이란 생명체의 신경조직으로부터 착안하여 모델화한 정보 처리 시스템으로서 단순한 소자들의 병렬, 분산 연결 구조를 가지고 있으며, 외부로부터 받아들이는 입력에 의하여 동적반응을 일으킴으로서 필요한 출력을 생성시키는 것이다. 신경 회로망은 단지 종래의 노이만 (Von Neumann)식의 순차 처리 시스템인 디지털 컴퓨터의 한계를 극복하여 병렬 분산 처리 시스템으로 접근하는 것이라 할 수 있다.(Lippman, 1989) 생물학적 뉴런은 인공 뉴런을 이해하는 데 중요한 세가지 유형의 요소들로 구성되어 있다. 그것은 Fig. 4에서 처럼 수상돌기, 세포

체, 시냅스, 축삭돌기이다. 수상돌기는 다른 뉴런으로부터 신호를 받는다. 그 신호는 화학적 과정에 의하여 시냅스의 틈을 가로질러 전송되는 전기적 임펄스들이다. 화학적 전달자들의 활동은 인공 신경 회로망의 가중치들의 활동과 유사한 방법으로 입력신호들을 수정한다. 세포체는 입력신호들의 합이다. 충분한 입력이 수신되면 세포는 흥분한다. 즉, 세포는 축삭돌기를 거쳐서 다른 세포들로 신호를 전달한다. 일반적으로 세포가 흥분하거나 전혀 흥분하지 않는다고 가정하기 때문에 전송 신호는 이진으로 처리되고 있다.

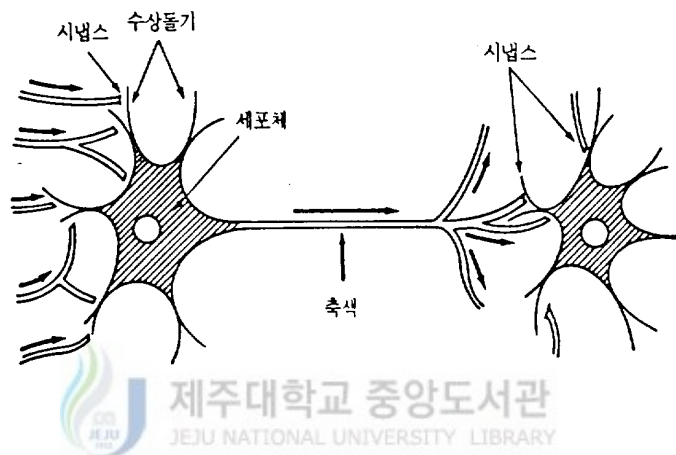


Fig. 4 Biological neuron

인공 뉴런(노드)은 생체의 세포체와 마찬가지로 인접한 많은 뉴런으로부터 시냅스를 통해 입력신호들을 받아서 저장되어 있거나 공급되는 가중치를 곱해서 다른 뉴런으로 보낸다. 이 때의 가중치를 연결강도라고 부른다. 즉, 가장 간단한 노드의 출력은 Fig. 5에서 보는 바와 같이  $N$ 개의 입력을 받아  $N$ 개의 연결강도의 벡터를 각각 곱하여 더하고 특정한 전달함수를 통과시킴으로서 얻어진다.(Lippman, 1989)

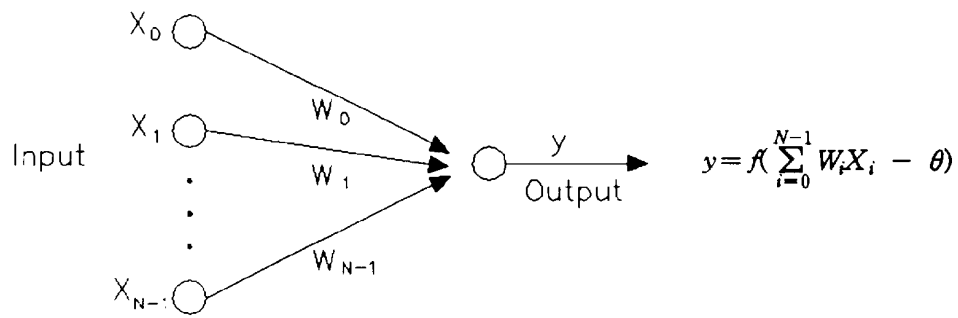


Fig. 5 Simple neural node

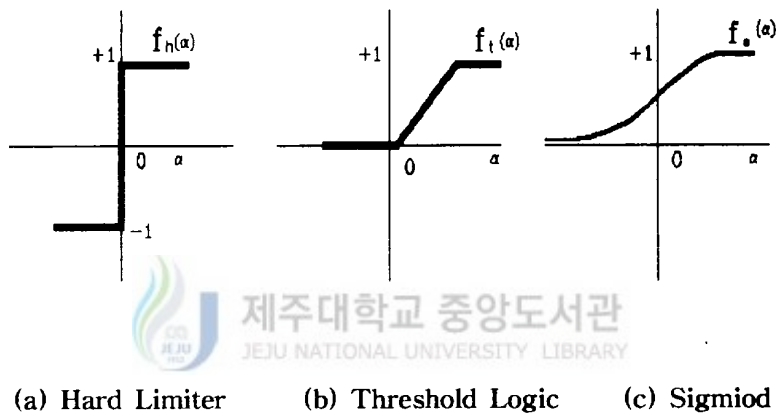


Fig. 6 Nonlinearity Function

노드는 내부적인 임계값이나 오프셋  $\theta$ , 그리고 비선형 함수의 형태에 따라 특징지어진다.(Lippman, 1989) 신경 회로망에서 많이 사용되는 대표적인 비선형 함수로는 Fig. 6처럼 계단 함수, 임계 논리 함수, 그리고 S자 형태의 시그모이드 함수 등이 있다. 이러한 노드들이 복합적으로 구성되어 신경 회로망을 형성한다. 신

경 회로망의 여러 모델들은 이들간의 결선방식과 학습법칙에 따라 구별한다. 신경 회로망은 데이터나 패턴들을 기억하고 회상하기, 패턴들을 분류하기, 입력패턴과 출력패턴들을 대응시키기, 유사한 패턴끼리 모으기, 구속하고 있는 최적 문제들에 대한 해를 찾기와 같은 다양한 문제들에 적용될 수 있다.

## 2. 호필드망

호필드망은 노드(뉴런)의 작용을 단지 임계값의 작용으로 보고 훈련에 의한 정보가 연결강도에 의해 표현된다는 간단한 이론에 기초하고 있으며, 연상기억 (associative memory), 최적화(optimization)문제, 순회판매원 문제(Traveling Salesman Problem)를 해결하는데 매우 유용하다. 호필드망은 자기자신은 연결되어 있지 않은 피드백 네트워크이다. Fig. 7에 보이는 이 망은 N개의 노드를 가지고 있다.

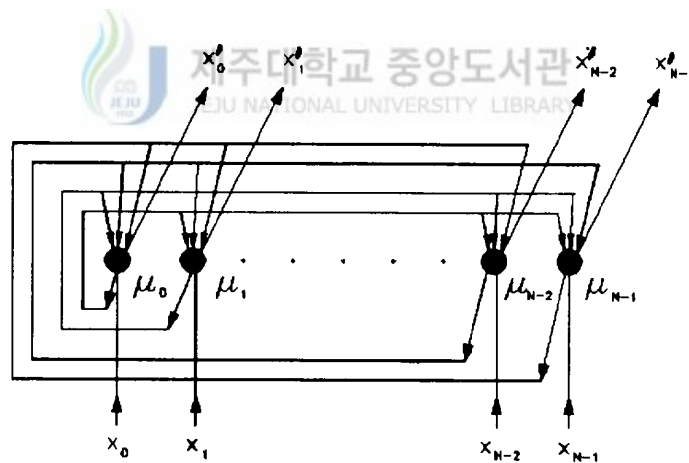


Fig. 7 Hopfield network



호필드망은 화소의 값이 흑과 백 두가지인 이진 영상이나 8비트의 ASCII문자와 같이 정확히 이진 표현이 가능한 분야에 적합하다. 호필드는 망의 연결강도가 대칭적이고 노드의 출력값이 비동기적으로 변할 때 수렴함을 증명하였다.(Hopfield, 1982) 노드  $i$ 와 노드  $j$ 간의 연결강도  $t_{ij}$ 는 다음과 같다.

$$t_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s, & i \neq j \\ 0, & i = j \end{cases} \quad (23)$$

$x_i^s$ 는 표본패턴  $s$ 의  $i$ 번째요소이다. 각 노드의 연결강도는 대칭적이고( $t_{ij} = t_{ji}$ ), 자기 자신의 노드를 제외한 모든 노드와 연결되어 있다. 시간이 0일 때의 노드의 출력(초기입력)  $\mu_i(0)$ 는 입력패턴으로 초기화 한다.

$$\mu_i(0) = x_i, \quad i=0, 1, 2, \dots, N-1 \quad (24)$$

그리고 식(23)을 이용하여 출력이 변하지 않을 때까지 호필드망은 반복한다.

$$\begin{aligned} \mu_j(t+1) &= f_h \left[ \sum_{i=0}^{N-1} t_{ij} \mu_i(t) \right], \quad 0 \leq j \leq M-1 \\ &= f(a) \end{aligned} \quad (25)$$

이 식에서  $f(a)$ 는 계단 함수로서 식(26)으로 정의한다.

$$f(a) = \begin{cases} 1 & (a > 0) \\ -1 & (a \leq 0) \end{cases} \quad (26)$$

호필드망의 동작을 요약하면, 먼저 모든 표본패턴으로 부터 식(23)을 이용하여 연결강도  $t_{ij}$ 를 정한다. 그리고 나서 미지 입력패턴을 시간이 영일 때( $t=0$ ) 그 망의 초기 출력으로 이용한다. 호필드망은 더 이상 출력이 변하지 않을 때 수렴한

것으로 간주한다. 수렴한 후 노드 출력에 의해 명시된 표본패턴을 망의 출력으로 결정한다. 호필드망을 연상 기억장치로 사용할 때 다음과 같은 두가지 단점이 있다. 첫째는 저장하고 다시 상기해 낼 수 있는 패턴의 갯수에 심각한 제한이 있다는 것이다. Abu-Mostafa가 호필드망의 기억용량에 대해 증명한 바에 의하면 입력 원소의 갯수를  $N$ 이라 할 때 학습 가능한 패턴의 갯수는  $0.15 \cdot N \log(N)$ 보다 작아야 한다. 이를 역으로 계산하면 10가지 부류의 패턴을 기억하기 위해서는 70개 이상의 노드와 5,000개가 넘는 연결선이 필요하다는 결론이 나온다. 두번째는 서로 다른 부류의 패턴들이 공유하는 비트수가 많아질수록 불안정해진다는 것이다. 이와 같은 문제점을 해결하기 위해서는 복잡한 직교과정을 도입해야 한다.

### 3. 해밍망

해밍망은 패턴인식에 이용되는 신경망 분류기로서 이진 패턴을 위한 최적 최소 에러 분류기로 이용되고 있다. 해밍망은 Fig. 8에서 처럼 하부망과 상부망 두개의 계산 동작망으로 구성되어 있다. 하부망은 전방향 네트워크인 반면, 상부망은 완전 연결된 피드백 형태의 winner-take-all(WTA) 네트워크이다.(Lippman, 1989)

$N$ 개의 입력과 하부망의 꼭대기에 있는  $M$ 개의 중간노드 간의 상호 연결강도는 표본패턴  $p^j$ 의 각 요소의  $1/2$ 이다. 분류기는 하부망의 밑부분에서 입력패턴  $x$ 를 제시하여 매칭스코어(Matching Score)를 계산하고 상부망으로 그 값을 전달한다. 그리고 나서 입력패턴은 제거되고, 상위망은 최대의 매칭스코어를 가진 표본패턴을 선택한다.

입력패턴  $x(x_0, x_1, \dots, x_{N-1})$ 을 해밍망에 제시하면, 해밍망은 우선 앞서

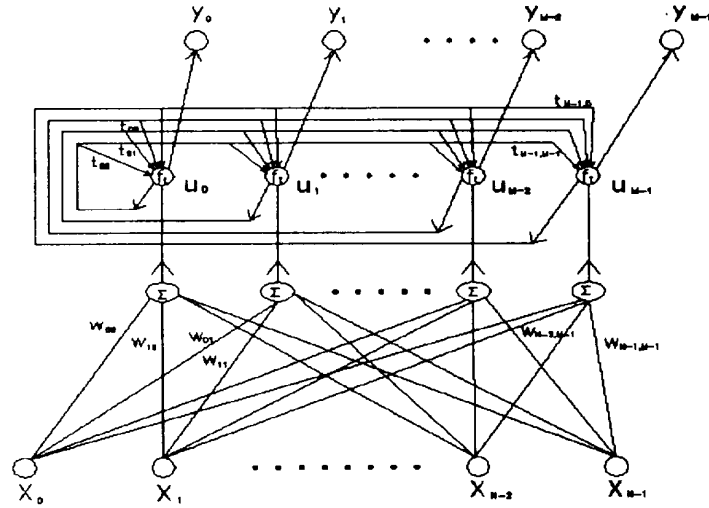


Fig. 8 Hamming Network

기억된 표본패턴  $p^j (p_0^j, p_1^j, \dots, p_{M-1}^j)$ ,  $0 \leq j < M$ 과 입력패턴 간의 매칭스 코어를 계산한다. 매칭스코어는 입력패턴과 표본패턴 간의 각 요소를 비교하여 일치하는 비트수로 정의된다.

$$\begin{aligned}
 MS_j(x) &= N - HD(x, p^j) \\
 &= N - \sum_{i=0}^{N-1} |x_i - p_i^j|, \quad 0 \leq j < M
 \end{aligned}
 \tag{27}$$

여기서  $x_i$ 은 입력패턴의  $i$ 번째요소이고,  $p_i^j$ 는 표본패턴  $j$ 의  $i$ 번째요소,  $HD(x, p^j)$ 는 표본패턴과 입력패턴 간의 해밍거리이다. 해밍거리는 입력패턴과 표본패턴 간의 각 요소가 서로 다른 비트수를 말한다.  $x$ 와  $p^j$ 간의 모든 비트가 일치할 경우 매칭스코어는 최대값  $N$ 이 되고( $MS = N$ ),  $x$ 와  $p^j$ 간의 모든 비트가 상이할 경우

매칭스코어는 최소값 0이 된다.(MS = 0) M개의 매칭스코어를 계산하고 난 후, 해밍망은 매칭스코어가 최대인 표본패턴을 선택하게 된다. 즉 표본패턴들 중에서 입력패턴과 일치하는 표본패턴을 분류하게 된다. 이진패턴에서  $x_i$ 와  $p_i^j$ 는 두개의 값(1,0)중 하나를 취하기 때문에 HD를 배타적-OR(XOR)연산으로 표현할 수 있다.

$$HD = \sum_{i=0}^{N-1} x_i \oplus p_i^j \quad (28)$$

여기서  $\oplus$ 와 그보수  $\overline{\oplus}$ 은 다음과 같이 나타낼 수 있다.

$$\begin{array}{r} x_i \quad 0 \ 0 \ 1 \ 1 \\ p_i^j \quad 0 \ 1 \ 0 \ 1 \\ x_i \oplus p_i^j \quad 0 \ 1 \ 1 \ 0 \\ \overline{x_i \oplus p_i^j} \quad 1 \ 0 \ 0 \ 1 \end{array}$$

위의 정의를 이용하여 식 (27)을 다음과 같이 쓸 수 있다.

$$\begin{aligned} MS_j &= N - \sum_{i=0}^{N-1} x_i \oplus p_i^j \\ &= \sum_{i=0}^{N-1} \overline{x_i \oplus p_i^j} \end{aligned} \quad (29)$$

요약하면, 이진 표본패턴의 매칭스코어는 각 표본패턴과 입력패턴 사이의 일치하는 요소수로 정의된다. 해밍망은 하부망에서 입력패턴과 표본패턴 간의 배타적-NOR(XNOR)연산을 수행하여 매칭스코어를 계산하고 상부망의 초기출력(상부망의 입력)으로 넘겨준다. 상부망에서 피드백하면서 하나의 노드만 최대가 될 때까지 반복한다.

위와 같이 매칭스코어를 직접 구하는 대신, 입력요소와 연결강도의 곱들을 합한 것으로 계산할 수 있다. 이진패턴의 요소 0을 -1, 1을 1로 나타내면 다음과 같다.

$$MS = \mu_j(0) = f_i\left(\sum_{i=0}^{N-1} W_{ij}x_i + \theta_j\right), \quad 0 \leq j \leq M-1 \quad (30)$$

$$W_{ij} = \frac{x_i^j}{2}, \quad \theta_j = \frac{N}{2} \quad (0 \leq i \leq N-1, \quad 0 \leq j \leq M-1) \quad (31)$$

$W_{ij}$ 는 하부망에서  $i$ 번째 입력요소와 노드  $j$ 간의 연결강도이고,  $\theta_j$ 는 노드  $j$ 의 임계값이다. 상부망의 연결강도와 임계값은 고정되어 있다. 임계값은 0으로 정하고, 각 노드 자신에 대한 연결강도는 1, 다른 노드 간의 연결강도는  $-\varepsilon$ 이다. ( $\varepsilon < 1/M$ ) 상부망의 연결강도  $t_{kl}$ 은 다음과 같다.

$$t_{kl} = \begin{cases} 1, & k=l \\ -\varepsilon, & k \neq l, \quad \varepsilon < \frac{1}{M} \end{cases} \quad (0 \leq k, \quad l \leq M-1) \quad (32)$$

$t_{kl}$ 는 상부망의 노드  $k$ 와 노드  $l$ 사이의 연결강도이다. 상부망에서 초기입력은 하부망의 MS값으로 초기화된다. ( $U_j(0) = MS_j$ ). 그리고 나서 두개의 부속망은 분리되고 상부망은 최대값을 발견할 때까지 몇번 반복한다.

상부망의 출력은 다음과 같다.

$$\begin{aligned} \mu_j(t+1) &= f_i\left(\sum_{i=0}^{M-1} \mu_i(t)t_{ij}\right) \\ &= f_i\left(\mu_j(t) - \varepsilon \sum_{k \neq j} \mu_k(t)\right) \\ &= f(a) \quad (0 \leq j, \quad k \leq M-1) \end{aligned} \quad (33)$$

여기서  $f(a)$ 는 임계논리함수로서 다음과 같이 정의된다.

$$f(a) = \begin{cases} a & (a > 0) \\ 0 & (a \leq 0) \end{cases} \quad (34)$$

#### 4. 비터비 복호 알고리즘에 해밍망 적용

앞 절에서 기술한 해밍망은 호필드망에 비하여 다음과 같은 몇가지 특징이 있다.(Lippman, Gold, Malpas, 1987) 첫째, 해밍망은 이론적으로 최적 최소 에러 분류기이므로 호필드망의 성능은 해밍망의 성능을 능가할 수 없다. 실제로 같은 패턴을 가지고 실험을 했을 때 호필드망은 30%이내의 잡음을 갖은 벡터에 대하여 동작을 했지만 해밍망은 48%의 잡음에도 정확히 동작한다. 두번째, 호필드에 비하여 상대적으로 적은 수의 연결강도를 필요로 한다. 예를 들면 10가지 부류의 입력 100개를 인식하기 위해서 해밍망은 1,100개의 연결선을 필요로 하는데 반해서 호필드망은 거의 10,000개에 달하는 연결선을 필요로 한다. 또한 해밍망의 연결강도 수는 입력에 비례하여 선형적으로 증가하고 호필드망은 입력의 제곱에 비례하여 증가한다.

따라서 본 논문에서는 해밍망에 비터비 복호 알고리즘을 적용한다. 기존의 비터비 복호 알고리즘에서 격자도의 부호어와 수신어 간의 해밍거리가 최소인 것을 결정한다는 것과 해밍망에서 표본패턴과 입력패턴 간의 매칭스코어가 최대인 값을 선택한다는 개념이 동일하다는 데서 착안하여, 해밍망의 입력패턴을 복호기 입력(수신부호어)으로 하고 격자도의 기준값(상태전이 부호어)을 표본패턴으로 하여 해밍망을 적용하면 다음과 같다.

$$X = r'(t) \quad (35)$$

$$P = w'(i, j) = \{in(i, j) * G\}' \quad (36)$$

위 식에서 '은 벡터요소 0을 -1로, 1을 1로 변환했음을 의미하며  $r(t)$ 는 수신부호어 벡터이고  $X$ 는 입력패턴 벡터이다.  $P$ 는 상태  $i$ 에서  $j$ 로 전이될 때 부호어, 즉 표본패턴 벡터이다. 식(35), (36)을 식(30) - (34)에 대입하여 격자도의 각 단계마

다 생존경로를 결정하는 데 응용할 수 있다.

격자도의 각 단계마다 해밍망을 이용하여 선택된 생존경로의 부호어(표본패턴)와 이전 단계의 상태에서 현 상태로 전이하도록 했던 입력비트를 기억시켜 둔다. 이는 기존의 비터비 복호 알고리즘에서 역추적 과정을 거쳐 최종 복호된 부호를 찾아내야 하는 처리 과정을 생략할 수 있다.



## IV. 결과 및 고찰

디지털 통신 시스템에서는 채널상에서 발생하는 에러를 정정하기 위하여 정보를 부호화하여 전송한다. 에러 정정 효율이 우수한 콘볼루션 부호는 복호시에 많은 양의 메모리를 필요로 하며 복호화 지연이 발생하기 때문에 실시간 처리가 가능한 복호 알고리즘인 비터비 복호 알고리즘을 적용하여 복호한다. 그러나 비터비 복호 알고리즘은 구속장의 길이가 긴 코드에서 지나치게 많은 기억장소를 필요로 하고 하드웨어 구현이 복잡하다. 따라서 신경 회로망의 병렬성이라는 특성을 적용하여 복호 시간을 줄이기 위한 복호 알고리즘으로서 해밍망을 적용한 비터비 복호 알고리즘을 적용한다. 기존의 비터비 복호 결과와 해밍망을 적용한 비터비 복호 결과를 비교분석한다.

### 1. 기존의 비터비 복호 결과

(4,1,2)콘볼루션 부호기의 생성행렬  $G$ 를 다음과 같이 정하고,

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

정보열  $u = 101110$ 을 채널코딩한 부호열  $c(t) = 1111, 0111, 0000, 1000, 0111, 1000, 1111, 0000$ 를 이진 대칭 채널을 통하여 전송했을 때 수신열  $r(t) = 1011, 1111, 0000, 0000, 0111, 1000, 1111, 0000$ 이라 두고 기존의 비터비 복호 알고리즘의 격자도를 이용하여 복호하는 방법과 행렬로 표현된 식으로 시뮬레이션 하였다.



Fig. 9는 (4,1,2)콘볼루션 부호를 기존의 비터비 복호 알고리즘을 사용하여 복호하는 것을 격자도로 표현하였다. 굵은 선은 각 시점에서의 생존경로를 의미하고

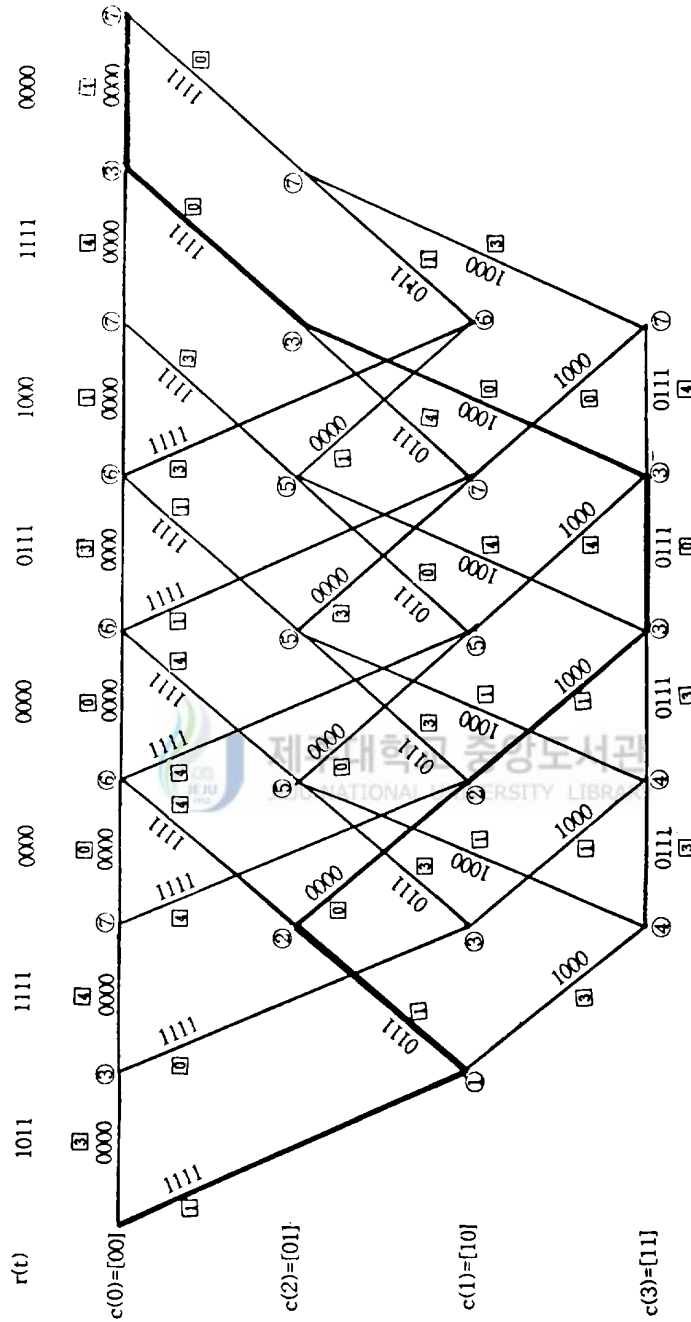


Fig. 9 Viterbi Algorithm of (4,1,2)Convolutional Code

있다. 사각문자는 가지 평가량을 원문자는 경로 평가량을 나타낸다. 가지 위의 부호어는 콘볼루션 부호기의 출력인 가지 부호어를 나타낸다.

비터비 복호 알고리즘을 행렬식으로 표현하여 시뮬레이션하면, 가지부호어  $w(i,j)$ 는 다음과 같다.

$$\begin{aligned}
 w(0, 0) &= [0,0,0] * G = 0000 & w(2, 0) &= [0,0,1] * G = 1111 \\
 w(0, 1) &= [1,0,0] * G = 1111 & w(2, 1) &= [1,0,1] * G = 0000 \\
 w(1, 2) &= [0,1,0] * G = 0111 & w(3, 2) &= [0,1,1] * G = 1000 \\
 w(1, 3) &= [1,1,0] * G = 1000 & w(3, 3) &= [1,1,1] * G = 0111
 \end{aligned}$$

격자도의 가지를 두 단계씩 통합할 경우 생성행렬  $G_m$ 은 다음과 같고,  $T(j)$ 는 다음과처럼 계산이 된다.

$$G_m = \begin{pmatrix} 1 & 1 & 1 & 1 & & & & & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & \\ & & & & 1 & 1 & 1 & 1 & \end{pmatrix}$$

$$T(0) = [00] * \begin{pmatrix} 1 & 1 & 1 & 1 & & & & & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ & & & & & & & & \end{pmatrix} = [00000000]$$

$$T(1) = [10] * \begin{pmatrix} 1 & 1 & 1 & 1 & & & & & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ & & & & & & & & \end{pmatrix} = [11110000]$$

$$T(2) = [01] * \begin{pmatrix} 1 & 1 & 1 & 1 & & & & & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ & & & & & & & & \end{pmatrix} = [01111111]$$

$$T(3) = [11] * \begin{pmatrix} 1 & 1 & 1 & 1 & & & & & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ & & & & & & & & \end{pmatrix} = [10001111]$$

각 시점마다의 가지, 경로, 생존 평가량은 Table 1과 같고 Table 2는 복호된 결과이다. 그리고 Table 3은 격자도를 통합했을 경우의 비터비 복호 알고리즘을 시뮬레이션 한 결과를 보이고 있다. Table 1과 Table 3을 비교해 보면 각 단계의

가지 평가량, 경로 평가량, 생존 평가량을 계산하는 횟수가 2배 만큼 줄었음을 알 수 있다.

Table 1 Computation Process of Conventional Viterbi Algorithm on (4,1,2)convolutional code

t	i, j	1	2	3	4	5	6	7	8
bm	0, 0	3	4	0	0	3	1	4	4
	2, 0			4	4	1	3	0	0
	0, 1	1	0	4	4	1	3		
	2, 1			0	0	3	1		
	1, 2		1	3	3	0	4	1	
	3, 2			1	1	4	0	3	
	1, 3		3	1	1	4	0		
	3, 3			3	3	0	4		
pm	0, 0	3	7	7	6	9	7	11	7
	2, 0			6	9	6	8	3	7
	0, 1	1	3	11	10	7	9		
	2, 1			2	5	8	6		
	1, 2		2	6	5	5	11	7	
	3, 2			5	5	7	3	10	
	1, 3		4	4	3	9	7		
	3, 3			7	7	3	7		
sm	0	3	7	6	6	6	7	3	7
	1	1	3	2	5	7	6		
	2		2	5	5	5	3	7	
	3		4	4	4	3	3	7	

Table 2 Result of Conventional Viterbi Decoding on (4,1,2)convolutional code

t	1	2	3	4	5	6	7	8
0	0	00	100	1000	10100	101000	1011100	<u>10111000</u>
1	1	01	101	1101	10001	110101		
2		10	110	1010	11010	101110	110101	
3		11	011	1011	10111	101111		

Table 3 Result of Viterbi Algorithm on integrated trellis of (4,1,2) convolutional code

t	i, j	1	2	3	4	5	6	7	8
P <sub>m</sub>	0, 0		7		7		10		15
	1, 0				10		8		7
	2, 0					6		7	7
	3, 0						9		10
	0, 1		3		11		12		
	1, 1					8		6	
	2, 1						10		9
	3, 1							5	8
	0, 2		2		14		11		
	1, 2					5		9	
	2, 2						5		12
	3, 2							8	3
	0, 3		4		12		7		
1, 3					7		13		
2, 3								10	7
S <sub>m</sub>	0		7		6		7		
	1		3		5		6		
	2		2		5		3		
	3		4		3		7		

또 다른 한 예로서 다음과 같은 생성행렬을 가진 (3,1,2)콘볼루션부호에서

$$\text{생성행렬 } G = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

정보열 0 1 0 0 1 0 0 0에 대응하는 콘볼루션 부호 c(t) = 011, 110, 001, 100, 110, 000, 101, 001, 011, 101이고 이를 이진 대칭 채널을 통하여 전송했을 경우



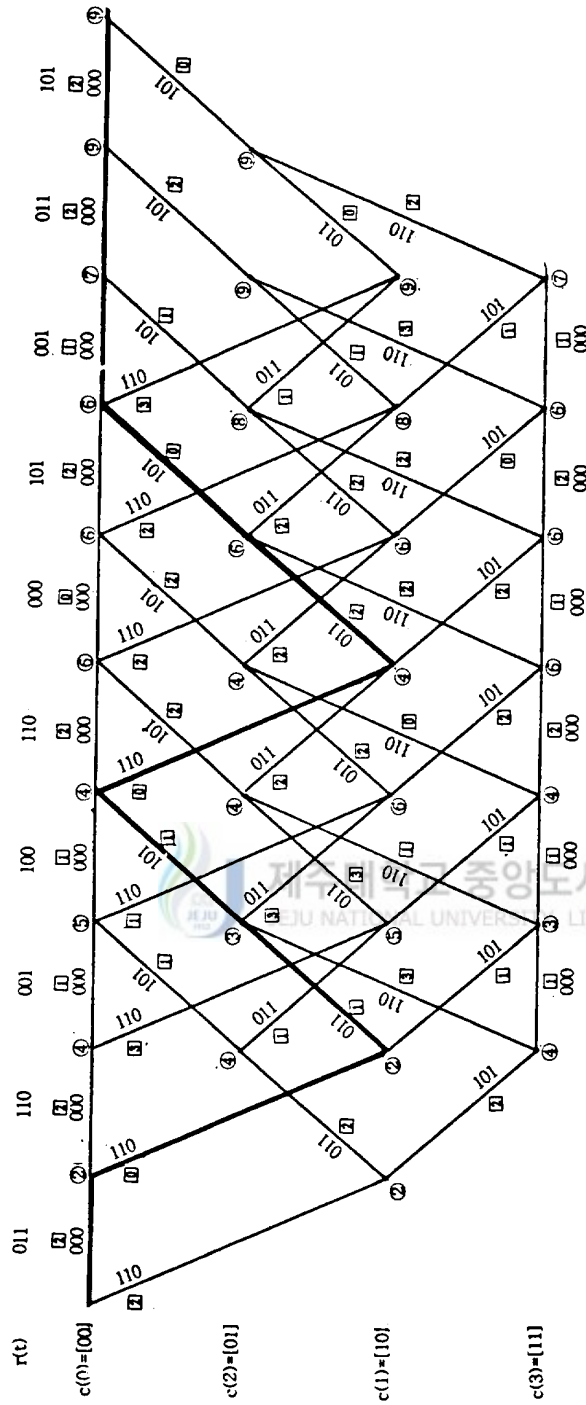


Fig. 10 Viterbi Algorithm of (3,1,2)Convolutional Code

각 시점마다의 가지, 경로, 생존 평가량은 Table 4과 같고 Table 5는 복호된 결과이다. 그리고 Table 6은 격자도를 통합했을 경우의 비터비 복호 알고리즘을 시뮬레이션 한 결과를 보이고 있다. Table 4과 Table 6을 비교해 보면 각 단계의 가지 평가량, 경로 평가량, 생존 평가량을 계산하는 횟수가 2배 만큼 줄었음을 알 수 있다.

Table 4 Computation Process of Conventional Viterbi Algorithm on (3,1,2)convolutional code

t	i, j	1	2	3	4	5	6	7	8	9	10
bm	0, 0	2	2	1	1	2	0	2	1	2	2
	2, 0			1	1	2	2	0	1	2	0
	0, 1	2	0	3	1	0	2	2	3		
	2, 1			1	3	2	2	2	1		
	1, 2		2	1	3	2	2	2	1	0	
	3, 2			3	1	0	2	2	3	2	
	1, 3		2	1	1	2	2	0	1		
	3, 3			1	1	2	0	2	1		
pm	0, 0	2	4	5	6	6	6	8	7	9	11
	2, 0			5	4	6	6	6	9	11	9
	0, 1	2	2	7	6	4	8	8	9		
	2, 1			5	6	6	6	8	9		
	1, 2		4	3	8	8	6	8	9	9	
	3, 2			7	4	4	8	8	9	9	
	1, 3		4	3	6	8	6	6	9		
	3, 3			5	4	6	6	8	7		
sm	0	2	4	5	4	6	6	6	7	9	9
	1	2	2	5	6	4	6	8	9	9	
	2		4	3	4	4	6	8	9		
	3		4	3	4	6	6	6	7		

Table 5 Result of Conventional Viterbi Decoding  
on (3,1,2)convolutional code

t	1	2	3	4	5	6	7	8	9	10
0	0	00	000	0100	01000	010000	0100100	01001000	010010000	<u>0100100000</u>
1	1	01	101	0001	01001	011101	0100001	01001001		
2		10	010	0110	01110	010010	0111010	01000010	010010010	
3		11	011	0111	01111	010011	0111011	01110111		

Table 6 Result of Viterbi Algorithm on integrated  
trellis of (3,1,2)convolutional code

t	i, j	1	2	3	4	5	6	7	8	9	10
P <sub>m</sub>	0, 0		4		6		6		9		11
	1, 0				4		10		9		
	2, 0					6	6		7		9
	3, 0					8	6		9		
	0, 1		2		6		8		11		
	1, 1				6		10		9		
	2, 1					6	8		9		
	3, 1					10	6		9		
	0, 2		4		10		6		9		
	1, 2				4		10		9		
	2, 2					8	8		9		
	3, 2					6	8		11		
	0, 3		4		8		6		9		
	1, 3				4		8		7		
	2, 3					6	8		9		
3, 3					6	6		9			
S <sub>m</sub>	0		4		4		6		7		9
	1			2		6		6	9		
	2			4		4		6	9		
	3			4		4		6	7		



## 2. 해밍망을 적용한 비터비 복호 결과

해밍망을 적용하기 위하여 벡터요소의 1을 1, 0을 -1로 변환하여 생존경로를 선택한 후, 생존경로의 부호어를 기억시킬 때는 -1을 0, 1을 1로 복원하였다. IV 장 1절에서 예를 들었던 (4,2,1)컨볼루션 부호를 해밍망에 적용해 보면, 시점 1에서 연결강도  $W_{ij}$ , 임계값  $\theta_j$ , 상부망 초기입력  $u_j$ , 상부망 연결강도  $t_{kl}$ 은 다음과 같이 계산된다.

$$\begin{aligned}
 t = 1 \text{ 일때,} \quad & r[1] = 1 \ -1 \ 1 \ 1 \text{ 이고} \\
 W_{00} = \frac{1}{2} \quad & W_{10} = -\frac{1}{2} \quad W_{20} = \frac{1}{2} \quad W_{30} = \frac{1}{2} \\
 W_{01} = \frac{1}{2} \quad & W_{11} = -\frac{1}{2} \quad W_{21} = \frac{1}{2} \quad W_{31} = \frac{1}{2} \\
 \theta_j = \frac{4}{2} \quad & j = 0, 1 \quad t_{00}=1 \quad t_{01}=-0.25 \\
 u_{j}[0] = 1.0 \quad & u_{j}[1] = 3.0
 \end{aligned}$$

위와 같이 반복 계산한 결과는 Table 7이 되며 여기서 밑줄은 생존노드를 의미한다.

Table 7 Result of Viterbi Decoding using Hamming Network on (4,1,2)convolutional code

t	1	2	3	4	5	6	7	8
$u_j[0]$	1.0	<u>3.0</u>	0.0	1.0	0.0	<u>4.0</u>	<u>4.0</u>	<u>4.0</u>
$u_j[1]$	<u>3.0</u>	1.0	<u>4.0</u>	<u>3.0</u>	<u>4.0</u>	0.0	0.0	0.0
u	1	0	1	1	1	0	0	0

Table 2와 Table 7을 비교해 보면 기존의 비터비 복호 알고리즘으로 복호된 정보열과 해밍망을 적용한 비터비 복호 알고리즘으로 복호된 정보열  $u = 1 \ 0 \ 1 \ 1 \ 1$

0 으로서 동일함을 알 수 있다.

IV장 1절의 두번째 예로서 (3,1,2)콘볼루션 부호로 전송된 부호열을 복호하기 위하여 해밍망을 적용해 보면, 시점 1에서 연결강도  $W_{ij}$ , 임계값  $\theta_j$ , 상부망 초기 입력  $u_j$ , 상부망 연결강도  $t_{kj}$ 은 다음과 같이 계산된다.

$$\begin{aligned}
 t = 1 \text{ 일때,} \quad & r[1] = -1, 1, 1 \text{ 이고} \\
 W_{00} = \frac{1}{2} \quad & W_{01} = -\frac{1}{2} \quad & W_{10} = -\frac{1}{2} \\
 W_{11} = \frac{1}{2} \quad & W_{20} = -\frac{1}{2} \quad & W_{21} = -\frac{1}{2} \\
 \theta_j = \frac{3}{2} \quad & j = 0, 1 \quad & t_{00} = 1 \quad t_{01} = -0.3 \\
 u_j[0] = u_j[1] = & 1.0
 \end{aligned}$$

위와 같이 반복 계산한 결과는 Table 8이 되며 여기서 밑줄은 생존노드를 의미한다.

Table 8 Result of Viterbi Decoding using Hamming Network on (3,1,2)convolutional code

t	1	2	3	4	5	6	7	8	9	10
$u_j[0]$	<u>1.0</u>	1.0	<u>2.0</u>	<u>2.0</u>	1.0	<u>1.0</u>	<u>3.0</u>	<u>2.0</u>	<u>1.0</u>	<u>1.0</u>
$u_j[1]$	1.0	<u>3.0</u>	2.0	0.0	<u>3.0</u>	1.0	1.0	0.0	1.0	1.0
u	0	1	0	0	1	0	0	0	0	0

Table 5와 Table 8을 비교해 보면, 기존의 비터비 복호 알고리즘으로 복호된 정보열과 해밍망을 적용한 비터비 복호 알고리즘으로 복호된 정보열  $u = 0 1 0 0 1 0 0 0$  으로서 동일함을 알 수 있다.

#### IV. 결 론

디지털 통신 시스템은 채널상에서 발생한 에러를 찾아 정정하기 위하여 정보를 콘볼루션 부호로 채널코딩하여 전송한다. 에러 정정 효율이 우수한 콘볼루션 부호는 복호시에 많은 양의 메모리를 필요로 하며 복호화 지연이 발생하기 때문에 실시간 처리가 가능한 복호 알고리즘으로서 비터비 복호 알고리즘을 이용한다. 그러나 비터비 복호 알고리즘은 구속장의 길이가 긴 코드에서 지나치게 많은 기억장소를 필요로하고 하드웨어 구현이 복잡하다.

본 논문에서는 채널상에서 수신된 부호어를 복호하기 위한 알고리즘으로서 기존의 비터비 복호 알고리즘에 관하여 분석하였고, 비터비 복호 알고리즘을 행렬식으로 유도했으며 격자도의 단계를 통합하여 계산횟수와 계산시간을 단축시켰다. 또한 이진 패턴 분류기로 가장 많이 이용되고 있는 신경 회로망 중에서 호필드망과 해밍망에 관하여 기술하였고, 비터비 복호 알고리즘에 해밍망 적용이 적합함을 입증하였다. 그리고 신경 회로망의 병렬성이라는 특성을 응용하여 복호시간을 줄이기 위한 복호 알고리즘으로서 해밍망을 적용한 비터비 복호 알고리즘을 제안하였다. 해밍망을 적용한 비터비 복호 알고리즘은 수신어와 부호어 간의 해밍거리를 구하고 최소거리를 갖는 경로를 선택하는 과정을 해밍망을 적용하는 새로운 알고리즘이다. 제안한 복호 알고리즘으로 복호된 정보열은 기존의 비터비 복호 알고리즘으로 복호된 것과 동일함을 확인하였다. 그리고 제안한 복호 알고리즘은 전체 메모리로 접근하는 것이 아니라 부분 메모리로 접근하도록 하였고, 역추적 과정을 생략함으로써 데이터 전송과 액세스 시간을 감소시켰다. 이와 덧붙여서 해밍망은 신경망의 병렬 분산 연결 구조를 가지고 병렬적으로 계산되는 특성을 갖고 있으므로

로 경쟁하는 노드의 수가 증가하더라도 계산시간이 급증하지 않았다.

격자도의 여러 단계를 통합하여 계산횟수와 콘블루션부호의 복호시간을 줄이고, 격자도의 통합된 가지의 수가 늘어나더라도 해밍망의 병렬성을 이용하여 효과적으로 복호할 수 있도록 연구할 필요가 있다.



## 참 고 문 헌

- Clark, G. and Cain, J., 1981, *Error-Crrrection Coding for Digital Communication*, Plenum Press, New-York.
- David Forney G, JR, 1973, "The Viterbi Algorithm", proceedings of the IEEE, VOL.61, NO.3.
- Hopfield J.J., 1982, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proc. National Acad. Sci. USA, Vol. 79, pp.2554-2558.
- Hopfield J.J., 1984, "Neurons with Graded Response Have Collective Computational Abilities", Proc. National Acad. Sci. USA, Vol.81, pp.3088-3092.
- Lin S. and Costello D.J. Jr., 1983, *Error Control Coding : Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, N.J. 07632, ch 11.
- Lathi B.P., 1989, *Modern Digital and Analog Communication Systems*, Saunders College Publishing
- Laurene Fausett, 1994, *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*, Prentice Hall
- Lippman R.P., 1989, "Introduction to computing with neural networks", IEEE ASSP Magazine. Nov., pp. 54-66.
- Lippman R.P., Gold B., and Malpas M.L., 1987, "A comparison of Hamming and Hopfield neural nets for parrern classification", Tech.

- 
- Rep. 769. Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, May.
- Man young Rhee, 1898, *Error Correction Coding Theory*, McGraw-Hill.
  - Moises E. Robinson, Hideki Yoneda, and Edgar Sanchez-Sinencio, 1992, "A Modular CMOS Design of a Hamming Network", *IEEE Transaction on neural networks*, May, VOL.3, NO.3.
  - Simon Haykin, 1994, *Neural Networks : A Comprehensive Foundation*, Macmillano
  - Tank D.W. and Hopfield J.J., 1986, "Simple"Neural"Optimization Networks:An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit", *IEEE Trans. on Circuits and Systems*, Vol. CAS-33, pp.533-541.
  - Tzi-Dar Chiueh and Rodney Goodman, 1987, "A Neural Network Classifier based on Coding Theory", *IEEE Conference on Neural Information Processing Systems*
  - Viterbi A.J. and Omura J.K., 1979, "Principle of Digital Communication and Codeing", New York:McGraw-Hall.

## 본 논문과 관련된 저자의 발표 논문

- 1) 전영희, 김장형, 이용학, “콘블루션 부호를 위한 비터비 복호 변환 알고리즘”, 산업 기술 연구 보고 제5권, pp. 33-42, 1994.
- 2) 전영희, 김장형, 이용학, “신경망을 이용한 비터비 알고리즘”, 한국 통신 학회·대한 전자 공학회 제주지부 합동 학술 발표 논문집, pp. 24-29, 1995.
- 3) 전영희, 이용학, “신경회로망을 이용한 비터비 변환 알고리즘”, 산업 기술 연구 보고 제6권, 1995.



## 감사의 글

희망과 기대감을 가득 안고 대학원 생활을 한 지도 벌써 2년이 되어갑니다. 그동안 친해지게 된 밤공기와 새벽이슬이 그리워질 것 같습니다. 저에게 힘과 용기를 주신 분들은 물론이고 가끔은 힘들게 했던, 그럼으로써 내 자신을 더욱 성숙하게 만들어 주셨던 분들께 진정으로 감사를 드립니다.

무엇보다도 연구 주제에서부터 완성에 이르기까지 2년동안 지도해 주시고 격려해 주신 이용학 교수님께 깊은 감사를 드립니다. 그리고 친절하게 세심한 조언을 아끼지 않으신 김홍수교수님과 바쁘신 중에도 선뜻 논문 심사를 맡아주신 장재성 교수님께 감사를 드립니다. 그동안 가르침을 주신 임재운 교수님, 양두영 교수님, 강진식 교수님을 비롯한 통신공학과 모든 교수님께 감사를 드립니다. 또한 학부 4년동안 세심한 지도와 격려의 말씀을 아끼지 않으신 김장형 교수님, 김두경 교수님, 안기중 교수님, 최연성 교수님, 오성보 교수님 그리고 곽호영 교수님께도 깊은 감사를 드립니다.

배움의 길로 한걸음 더 나아갈 수 있도록 배려해 주신 한라병원 김병찬 원장님과 인생에 대하여 여러가지 좋은 말씀을 아끼지 않으신 김성수 부원장님께 감사를 드립니다. 세심한 관심을 보여주신 허성록 재단사무국장님, 김달영 사무국장님, 김기남 행정관리부장님, 진선미 간호부장님, 윤동현 임상병리과장님, 송민관 과장님, 김종완 과장님, 조숙희 과장님, 김영진 과장님께도 깊은 감사를 드립니다. 부러울 만큼이나 여성스럽던 형숙과 언제나 명랑한 영실에게도 고마운 마음을 전합니다.

처음 어색함과 긴장감으로 연구실로 입문할때부터 지금까지 배움의 자세에 관하여 많은 조언을 아끼지 않으신 기태선배, 부식선배께 고마움을 전합니다. 지금은 졸업했지만 특이한 경험을 할 수 있도록 했던 청민, 삶의 방향을 바꾸고 싶어하던



승남, 아이아빠가 된 것을 자랑하던 덕주, 입학동기이고 힘들어 할 때 따뜻한 말을  
잊지 않은 창진, 성욱에게 무엇과도 바꿀수 없는 고마움을 전합니다. 보기만 해도  
믿음직스런 권익, 올바른 학문의 길을 걷고 싶어하는 재신, 한걸음 더 성숙하게 해  
주었던 영희에게도 고마움을 전합니다.

한술이를 유능한 프로그래머로 키우셨다시던 김대영 선생님, 영상처리에 많은  
관심을 보이시던 문석환 선생님, 바쁜 스케줄에도 관심을 보여주신 양행진 선배님  
께 고마움을 전합니다. 집 지키는 강아지처럼 제주대학교를 항상 지켜주던 회국, 항  
상 아껴주고 싶었던 여운, 정희, 영민, 영호 그리고 행운을 전해주는 데레사에게  
이 마음을 전합니다. 바쁜 핑계로 연락도 제대로 못했지만 소중한 친구들에게도  
이 마음을 전합니다.

힘겨워하는 모습을 바라보며 안타까워하시던 큰오빠, 논문 작성법이란 하면서  
약올렸던 작은오빠, 자신에게 관심을 보여달라고 투정부리던 동생, 학술 발표 연습  
을 지켜보며 내용에 관계없이 말버릇을 지적해 주던 사랑하는 열세살짜리 미영,  
여덟살짜리 은성, 여섯살짜리 지한, 마지막으로 지금 이 순간에도 당신의 자식을  
걱정하느라 노심초사하시고 세상에서 그 무엇과도 바꿀 수 없는 어머님께 이 작은  
논문을 바칩니다.

1995년, 마무리를 하면서 . . . . .