

神經回路網을 利用한 비터비 變換 알고리즘

이 용 학* · 전 영 회**

Transform Viterbi Algorithm Using Neural Networks

Yong-Hak Lee* and Young-Hee Jun**

ABSTRACT

A great number of papers have been 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 analized the Viterbi algorithm and represented it with simple matrix operations. We also analized the Hamming neural network which was a parallelism. The procedure computing the Hamming distance between the received codes and codewords was applied to lower subnet in the Hamming neural network and selection of the minimum Hamming distance used MAXNET in the Hamming neural network.

Key words : Viterbi decoding algorithm, Hamming neural network, Hamming distance

1. 서 론

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⁽¹⁾. 채널상에서 발생하는 에러 정정 효율이 우수한 콘볼루션 부호를 많이 이용하고 있다. 콘볼루션 부호는 입력 데이터를 메모리에 있는 기존의 데이터와 상관 관계를 갖게 부호화하는 방법이다^(2,3). 그러나 콘볼루션

부호는 복호시에 많은 양의 메모리를 필요로하며, 복호화 지연이 발생하기 때문에 실시간 처리가 가능한 복호 알고리즘이 필요하게 되었다.

이에 따라 콘볼루션 부호의 최우 복호 알고리즘인 비터비 알고리즘이 제안되었는데, 이 알고리즘은 위성 통신, 문자와 음성인식등의 여러 분야에서 응용되고 있으며, 그 응용범위가 증가되고 있는 추세이다. 그러나 비터비 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.

신경회로망은 생명체의 신경조직으로 부터 착안하여 모델화한 정보처리시스템으로서 단순한

* 제주대학교 통신공학과
Dept. of Telecommunication Eng., Cheju Nat'l Univ.
** 제주대학교 대학원
Graduate school, Cheju Nat'l Univ.

소자들의 병렬, 분산 연결구조를 가지고 있으며 이진패턴들을 분류할 수 있다. 또한 디지털 통신 시스템을 유한 패턴집합을 전송하는 것으로 간주할 수 있으므로 신경회로망의 분류기능을 콘볼루션 부호를 복호하는데 응용할 수 있다. 해밍망은 불규칙적으로 일그러진 이진 패턴들을 분류하는 최적 분류기를 구현한 신경회로망이다. 무기억 이진 대칭채널을 통해서 이진 고정길이 신호를 보낼 때 최적 최소 에러 분류기는 각 클래스의 표본에 대한 해밍거리를 계산하고 최소 해밍거리를 가진 클래스를 선택한다⁽⁵⁾.

본 논문은 콘볼루션부호와 비터비 알고리즘을 분석하였고, 비터비 알고리즘을 간단한 행렬식으로 표현하였다. 또한, 수신어와 부호어간의 해밍거리를 계산하여 가지평가량을 구하고 전단계의 생존평가량과 합함으로서 경로평가량을 구한 후 최단경로 값을 가진 상태를 현 단계의 생존경로로 선택하는 비터비 알고리즘에 해밍망을 적용하여 하위망에서 해밍거리를 계산하고 상위망에서 최단경로를 가진 노드를 선택하도록 제안했다. 제안된 알고리즘은 전체 메모리를 접근하는 것이 아니라 부분 메모리를 접근하도록 제한하고 역추적 과정을 제거하기 때문에, 데이터 전송과 역세스로 야기되는 VLSI회로의 시간 소비를 감소시킬 수 있다.

II. 비터비 알고리즘과 신경회로망

2.1 콘볼루션 부호기와 비터비 알고리즘

(n,k,m)인 콘볼루션부호기는 m개의 기억소자를 가지고 k개의 입력, n개의 출력을 보내는 유한상태기이다. 각각의 입력노드와 연결된 부호기는 쉬프트레지스터와 Modulo-2 덧셈기로 구성되어 있다⁽²⁾. 부호율 R은 입력 비트수 k와 출력 비트수 n의 비로 나타내고 생성다항식에 따라서 부호기의 출력과 구조가 결정된다^(1,2).

Fig. 1은 생성다항식 $g^{(1)} = 1 + D$, $g^{(2)} = D + D^2$, $g^{(3)} = 1 + D^2$ 인 (3,1,2) 콘볼루션부호기를 나타낸 것이다. Fig. 2에서 처럼 격자도는 콘볼루션 부호기의 레지스터 값을 하나의 상태로 하

여 각 입력 값에 따른 상태의 변화를 이산적인 시간축의 함수로 나타낸 것이다.

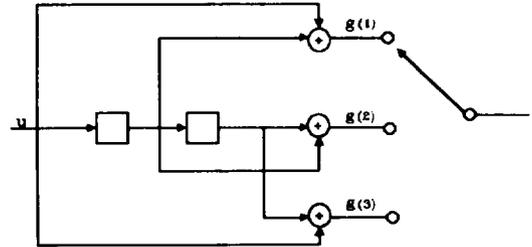


Fig. 1 (3,1,2)Convolution encoder

(n,k,m)코드의 격자도에는 $N = 2^{km}$ 개의 상태가 있다. 상태 S_i 에서 다음 단계의 상태 S_j 를 연결하는 가지를 e_{ij} 로 나타낸다. 시점 t에서 어떤 상태 S_i 에 연결되는 가지들의 집합을 $J(j)$ 라고 하며 2^k 개가 있다. 각 상태에서 필요한 연산은 경로평가량을 찾아서 S_i 에 대한 모든 후보경로들 중에서 생존경로를 결정하는 것이다. S_i 의 모든 후보경로에 대한 경로평가량 $pm(i,j,t)$ 은 유사측정 함수 f를 거쳐서 계산하고($i \in J(j)$), 그것들 중에서 최단 경로평가량을 생존평가량 $sm(j,t)$ 로서 선택한다. $pm(i,j,t)$ 와 $sm(j,t)$ 는 다음과 같은 관계가 성립한다.

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

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

$r(t)$ 는 t번째 시점에서 수신된 n-비트코드어이고, $w(i,j)$ 는 임의의 상태로 전이될 때의 부호기의 출력으로서 e_{ij} 에 대응하는 코드어이다. X는 최단경로 비교기이다.

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

즉, 비터비 알고리즘은 복호기의 입력과 부호기에 의하여 주어지는 격자도의 기준값과 해밍거리인 가지평가량을 누적한 경로평가량을 이용한다. 각 상태에서 경로평가량은 들어오는 가지 중

에서 전 단계의 생존평가량과 현단계의 가지 평가량을 합한 것중 가장 작은 것을 선택하여 얻어진다. 선택된 가지를 생존자라고 하며 가장 작은 경로평가량을 갖은 상태에서 시작하여 생존자를 역추적함으로써 복호기의 출력을 결정한다.

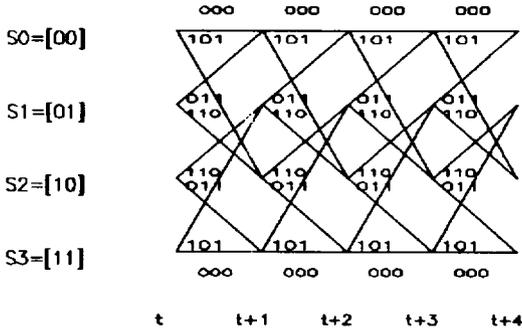


Fig. 2 Trellis of (3,1,2)convolution encoder

2.2 비터비 알고리즘의 행렬표현

2.2.1 상태 코드 정의

(n,1,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,

$$C(j) = S_j \text{의 상태코드} = j \text{의 이진표현}$$

2.2.2 가지 코드 생성

부호기의 주요 기능은 S_i 에서 S_j 로 가능한 상태 전이들에 따라서 가지코드 $w(i,j)$ 를 생성하는 것이다. 부호과정울 기초로 하여, 가지 코드를 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_i 에서 S_j 로 상태 전이를 일으키는 입력비트이다. 즉,

$$in(i,j) = [LMB(j) \setminus C(i)] \quad (3)$$

여기서 \setminus 는 연속을 의미하고 $w(i,j)$ 는 식(4)와 같이 일반화될 수 있다.

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

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

$$G = \begin{bmatrix} G_0 \\ \dots \\ G_i \\ \dots \\ G_m \end{bmatrix} \quad (5)$$

$$G_r = \begin{bmatrix} g_{1,r}^1 & g_{1,r}^2 & g_{1,r}^3 & \dots & g_{1,r}^n \\ g_{2,r}^1 & g_{2,r}^2 & g_{2,r}^3 & \dots & g_{2,r}^n \\ \dots & \dots & \dots & \dots & \dots \\ g_{k,r}^1 & g_{k,r}^2 & g_{k,r}^3 & \dots & g_{k,r}^n \end{bmatrix}$$

요소 $g_{i,r}^j$ 는 g_i^j 의 r번째 비트, X_i 와 Y_j 에 관한 생성기열이다⁽²⁾. 예를 들어 (3,1,3)콘볼루션코드를 다음과 같이 정의하면,

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

$$w(1,3)=[LMB(3) \setminus C(1)] * G = [1 \setminus 100] * G = [101] \quad (7a)$$

$$w(3,6)=[LMB(6) \setminus C(3)] * G = [0 \setminus 110] * G = [110] \quad (7b)$$

$$w(6,4)=[LMB(4) \setminus C(6)] * G = [0 \setminus 011] * G = [100] \quad (7c)$$

2.2.3 가지 평가량 계산

이진 대칭 채널(BSC)의 경우, 부호어 $w(i,j)$ 와 수신어 $r(t)$ 는 n차 벡터이다. 가지평가량 $bm(i,j,t)$ 를 $w(i,j)$ 와 $r(t)$ 간의 해밍거리로부터 구할 수

있다. $bm(i,j,t)$ 가 $r(t)$ 와 t 시점의 모든 코드어 간의 해밍거리들 중에서 최소 해밍거리라면 최우 복호기는 $w(i,j)$ 를 전송 코드어로 선택한다. 가지평가량은 식(8)과 같이 표현할 수 있다.

$$\begin{aligned} bm(i, j, t) &= r(t) \oplus w(i, j) \times I^T \\ &= r(t) \oplus [in(i, j) * G] \times I^T \\ &= [in(i, j) \setminus 1] * \left(\frac{G}{r(t)}\right) \times I^T \end{aligned} \quad (8)$$

I 는 n 차 단위 벡터 $I = [111 \dots 1]$ 이다.

2.2.4 경로 평가량계산

가지평가량을 계산한 후, 경로평가량을 계산해야 한다. 경로평가량, $pm(i,j,t)$ 은 $bm(i,j,t)$ 와 $sm(i,t-1)$ 의 합으로 다음처럼 정의한다.

$$sm(j, t+1) = X(pm(i, j, t+1)) \quad (9)$$

$$\begin{aligned} pm(i, j, t+1) &= f(r(t), w(i, j), sm(i, t)) \\ &= bm(i, j, t+1) + sm(i, t) \end{aligned} \quad (10)$$

2.2.5 역추적과정 제거

일반적인 비터비 복호에서, 역추적과정을 통해 최종 복호된 부호를 추적해야 한다. $sm(j,t)$ 를 계산할 때, 가장 큰 경로 평가량으로 전이하도록 했던 이전상태의 전이입력비트를 기록해둔다.

$$U(j,t)=C(q) \text{ iff } X(pm(i,j,t))=pm(q,j,t) \quad (11)$$

여기서 $U(j,t)$ 는 최우 복호 부호이다. 이와같이 $sm(j,t)$ 가 모든 상태들에 대한 생존자 평가량 중에서 최대값이라면, $U(j,t)$ 로부터 직접 읽어낼 수 있다. 일반적인 비터비 알고리즘의 역추적과정을 제거할 수 있다.

2.3 신경회로망

신경회로망이란 생명체의 신경조직으로부터 착안하여 모델화한 정보처리시스템으로서 단순한 소자들의 병렬, 분산 연결구조를 가지고 있으며, 외부로부터 받아들이는 입력에 의하여 동적반응을 일으킴으로서 필요한 출력을 생성시키는 것이다. 신경회로망 연구자들은 생체신경계통 구조를 모방한 인공신경회로망을 만들어서 생명체의 특수한 정보처리기능을 부분적이거나 모방하고자 하는 의도에서 신경회로망을 개발, 응용하고 있다. 그러나 신경회로망의 개발이 인공두뇌를 만드는 것을 의미하지는 않는다. 두뇌에 대해서는 구조와 작용원리 등이 피상적인 것 외에는 잘 알려져있지 않다. 신경회로망은 단지 종래의 노이만 식의 순차처리 시스템인 디지털 컴퓨터의 한계를 극복하여 병렬 분산 처리 시스템으로 접근하는 것이라 할 수 있다⁽⁴⁾.

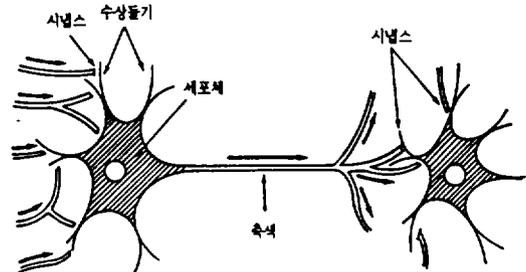


Fig. 3 Biological neuron

신경회로망을 이루는 기본소자는 Fig. 3과 같이 신경세포(뉴런)와 연결부(시냅스)로 구성되어 있다. 여러모델들은 이들 간의 결선방식과 학습법칙에 따라 구별된다. 뉴런이란 생체의 신경세포와 마찬가지로 인접한 많은 뉴런으로부터 입력신호들을 받아서 하나의 결론을 출력한다.

연접부는 뉴런의 출력을 받아서 이에 저장되어 있거나 공급되는 가중치를 곱해서 다른 뉴런으로 보낸다. 이 때의 가중치를 연결강도라고 부른다. 뉴런의 출력은 그 뉴런에 연결된 모든 뉴런의 출력값 또는 외부로부터의 입력값에 해당 가중치를 곱한 값들을 모두 더해서 뉴런의 전달함수를 통과시킴으로서 얻어진다⁽⁴⁾.

신경회로망의 간단한 노드는 Fig. 4에서 보는 바와 같이 N개의 입력을 받아 N개의 연결강도의 벡터들과 곱해져서 특정한 전달함수를 거쳐 출력을 내게 된다.

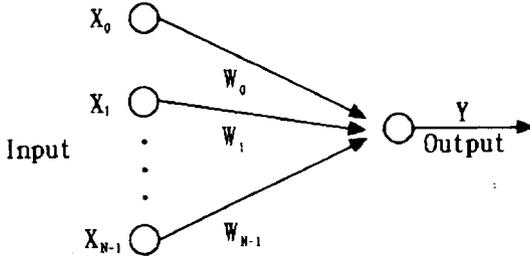
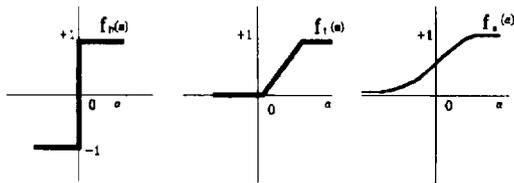


Fig. 4 Simple neural node



(a)Hard limiter (b)Threshold logic function (c)Sigmoid Function

Fig. 5 Nonlinearity function

노드는 내부적인 임계값이나 오프셋 θ , 그리고 비선형 함수의 형태에 따라 특징지워진다⁽⁴⁾. 신경회로망에서 많이 사용되는 대표적인 비선형 함수는 Fig. 5에서처럼 계단함수, 임계논리함수와 S자형태의 시그모이드함수 등이 있다.

2.4 해밍망

해밍망은 패턴인식에 이용되는 신경망 분류기로서 이진 패턴을 위한 최적 최소 에러 분류기로 이용되고 있다. 해밍망은 Fig. 6에서처럼 하부망, 상부망 두개의 계산동작망으로 구성되어 있다. 하부망은 전방향네트워크인 반면, 상부망은 완전 연결된 피드백형태의 winner-take-all (WTA) 네트워크이다^(4,5).

N개의 입력과 하부망의 상부에 있는 M개의 중간노드 간의 상호연결강도는 표본패턴 p^j 의 각 요소의 절반이다. 분류기는 하부망의 밑부분에서 입력패턴 x 를 제시하여 매칭스코어를 계산

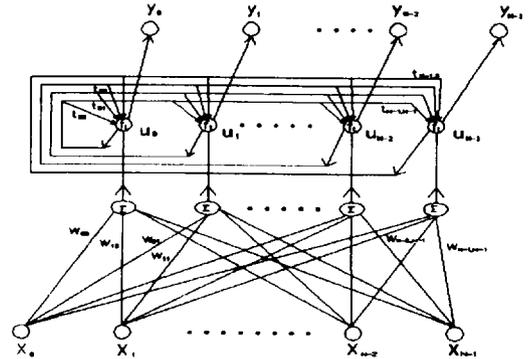


Fig. 6 Hamming network

하고 상부망으로 그 값을 전달한다. 그리고 나서 입력패턴은 제거되고, 상위망은 최대의 매칭스코어를 가진 표본패턴을 선택한다.⁽⁶⁾ 입력패턴 $x (x_0, x_1, \dots, x_{n-1})$ 을 해밍망에 제시하면, 해밍망은 우선 앞서 기억된 표본패턴 $p (p_0^j, p_1^j, \dots, p_{m-1}^j)$, $0 \leq j < M$ 과 입력패턴 간의 매칭스코어를 계산한다. 매칭스코어는 입력패턴과 표본패턴간의 각요소를 비교하여 일치하는 비트수로 정의된다.

$$MS_j(x) = N - HD(x, p^j)$$

$$= N - \sum_{i=0}^{N-1} |x_i - p_i^j|, \quad 0 \leq j < M \quad (12)$$

여기서 x_i 은 입력패턴의 i 번째요소이고, p_i^j 는 표본패턴 j 의 i 번째요소, $HD(x, p^j)$ 는 표본패턴과 입력패턴간의 해밍거리이다. 해밍거리는 입력패턴과 표본패턴간의 각요소가 다른 비트수를 말한다. x 와 p^j 간의 모든 비트가 일치할 경우 매칭스코어는 최대값 N 이 되고, x 와 p^j 간의 모든 비트가 상이할 경우 매칭스코어는 최소값 0 이 된다. M 개의 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.

$$HD = \sum_{i=0}^{N-1} x_i \oplus p_i \quad (13)$$

식(12)에 식(13)을 대입하여 식(14)로 다시 쓸 수 있다.

$$\begin{aligned} MS_j &= N - \sum_{i=0}^{N-1} x_i \oplus p_i \\ &= \sum_{i=0}^{N-1} \overline{x_i \oplus p_i} \end{aligned} \quad (14)$$

요약하면, 이진 표본패턴의 매칭스코어는 각 표본패턴과 입력패턴사이의 일치하는 요소수로 정의된다. 해밍망은 하부망에서 입력패턴과 표본패턴간의 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 식(15), (16)와 같다.

$$MS = \mu_j(0) = f_j(\sum_{i=0}^{N-1} W_{ij}x_i + \theta_j) \quad (15)$$

$0 \leq j \leq M-1$

$$W_{ij} = \frac{x_i}{2}, \quad \theta_j = \frac{N}{2} \quad (16)$$

$0 \leq i \leq N-1, 0 \leq j \leq M-1$

W_{ij} 는 하위망에서 i 번째 입력요소와 노드 j 간의 연결강도이고, θ_j 는 노드 j 의 임계값이다. 상부망의 연결강도와 임계값은 고정되어 있다. 임계값은 0으로 정하고, 각 노드자신에 대한 연결강도는 1, 다른 노드간의 연결강도는 $-\epsilon$ 이다. ($\epsilon < 1/M$) 상부망의 연결강도 t_{kl} 를 수식으로 전개하면 식(17)이 된다.

$$t_{kl} = \begin{cases} 1, & k=l \\ -\epsilon, & k \neq l \end{cases} \quad \frac{\epsilon < 1}{M}$$

$$0 \leq k, \quad l \leq M-1 \quad (17)$$

t_{kl} 는 상부망의 노드 k 와 노드 l 사이의 연결강도이다. 상위망에서 초기입력은 하위망의 MS값으로 초기화된다. 그리고 나서 두개의 부속망은 분리되고 상위망은 최대값을 발견할 때까지 상위망은 몇번 반복한다.

상위망의 출력 u_j 는 식(18)과 같다.

$$\begin{aligned} \mu_j(t+1) &= f_j(\sum_{i=0}^{M-1} \mu_i(t)t_{ij}) \\ &= f_j(\mu_j(t) - \epsilon \sum_{k \neq j} \mu_k(t)) \\ &= f(a) \end{aligned} \quad (18)$$

$$0 \leq j, \quad k \leq M-1$$

여기서 $f(a)$ 는 임계논리함수로서 다음과 같이 정의된다.

$$f(a) = \begin{cases} a & (a > 0) \\ 0 & (a \leq 0) \end{cases} \quad (19)$$

2.5 해밍망을 적용한 비터비 알고리즘

일반적인 비터비 알고리즘에서 격자도의 부호어와 수신부호어간의 해밍거리가 최소인 것을 결정한다는 것은 해밍망에서 표본패턴과 입력패턴간의 매칭스코어가 최대인 값을 선택하는 것과 동일하다. 이러한 원리에 의하여 복호기 입력을 입력패턴으로 하고 격자도의 기준값을 표본패턴으로 하여 해밍망을 적용하면 식(20), (21)과 같다.

$$X = r'(t) \quad (20)$$

$$P = w'(i,j) = \{in(i,j)*G\}' \quad (21)$$

식(20)에서 '은 요소 0을 -1로, 1을 1로 변환했음을 의미하며 $r(t)$ 는 수신부호어 벡터이고 X 는 입력패턴 벡터이다. P 는 상태 i 에서 j 로 전이

될 때 부호어, 즉 표본패턴 벡터이다. 식(20), (21)을 식(15) - (19)에 대입하면 격자도의 각단계 생존경로를 결정하는 데 응용할 수 있다.

격자도의 각 단계에 해밍망을 적용하여 선택된 생존경로의 부호어(표본패턴)와 이전 단계의 상태에서 현 상태로 전이하도록 했던 입력비트를 기억시켜 둔다. 이는 기존의 비터비 알고리즘에서 역추적 과정을 거쳐 최종 복호된 부호를 찾아내야 하는 처리과정을 제거할 수 있다.

III. 결과 및 고찰

예로서 (3,1,2)콘볼루션 부호의 생성행렬 G와 수신 부호열 r(t)는 다음과 같고, 기억소자의 초기상태는 00이며 각 단계에서 들어오는 가지의 해밍거리가 동일한 경우 입력비트 0에 의해 전이되는 경로를 생존경로로 선택하도록 했다.

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

$$r(t) = [011 \ 110 \ 001 \ 100 \ 110 \ 000 \ 101 \ 001 \ 011 \ 101]$$

가지코드 w(i,j)는 다음과 같다.

$$\begin{aligned} w(0,0) &= [0,0,0]*G=000 & w(2,0) &= [0,0,1]*G=101 \\ w(0,1) &= [1,0,0]*G=110 & w(2,1) &= [1,0,1]*G=011 \\ w(1,2) &= [0,1,0]*G=011 & w(3,2) &= [0,1,1]*G=110 \\ w(1,3) &= [1,1,0]*G=101 & w(3,3) &= [1,1,1]*G=000 \end{aligned}$$

3.1 기존의 비터비 알고리즘 결과

기존의 비터비 알고리즘을 행렬식으로 시뮬레이션한 결과 각 시점의 가지, 경로, 생존평가량은 Table 1과 같고 Table 2는 복호과정을 나타내고 있다.

3.2 해밍망을 적용한 비터비 알고리즘 결과

해밍망의 특성 때문에 1을 1, 0을 -1로 변환하여 생존경로를 선택한 후, 생존경로의 부호어

를 기억시킬 때는 -1을 0, 1을 1로 복원하였다. 시점 1에서 연결강도 W_{ij} , 임계값 θ_j , 상부망 초기입력 u_j , 상부망 연결강도 t_{ki} 은 다음과 같이 계산된다.

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

위와 같이 반복 계산한 결과는 Table 3이 되며 여기서 밑줄은 생존노드를 의미한다.

기존의 비터비 알고리즘과 해밍망을 적용한 비터비 알고리즘을 비교해 보면 Table 2와 Table 3과 같이 복호된 정보열은 동일하게 나타난다.

Table 1 Computation process of conventional Viterbi algorithm

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		
	0, 0	2	4	5	6	6	6	8	7	9	11
	2, 0			5	4	6	6	6	9	11	9
pm	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	9	9
	sm	0	2	4	5	4	6	6	6	7	
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 2 Results of conventional Viterbi decoding

	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 3 Results of Viterbi decoding using Hamming network

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

IV. 결 론

본 논문에서는 패턴분류기로 이용되고 있는 신경회로망 중에서 해밍망을 이용하여 수신어와 부호어간의 해밍거리를 구하고 최소거리를 갖는 경로를 선택하는 비터비 알고리즘을 제안하였다. 해밍망을 적용한 비터비 알고리즘으로 복호된 정보열은 기존의 비터비 알고리즘으로 복호된 것과 동일하게 나타남을 확인 하였다.

따라서 제안된 알고리즘은 전체 메모리로 접근하는 것이 아닌 부분 메모리로 접근하도록 하였고, 역추적 과정을 제거했기 때문에 데이터 전송과 액세스 시간을 감소시킬 수 있다. 또한 해밍망은 병렬적으로 계산되기 때문에 경쟁하는 노드의 수가 증가하더라도 노드수가 증가한 만큼 계산시간이 증가하지 않는다.

추후 연구과제는 격자도의 여러 단계를 통합하여 계산회수와 콘볼루션부호의 복호시간을 줄이고, 해밍망의 병렬성을 이용하여 격자도의 통합된 가지의 수가 증가해도 효율적으로 복호 할 수 있도록 한다.

참고문헌

1. Clark, G. and Cain, J., 1981, *Error-Crrrection Coding for Digital Communication*, Plenum Press : New-York.
2. Man young Rhee, 1898, *Error Correction Coding Theory*, McGraw-Hill.
3. Lin S. and Costello D.J. Jr., 1983, *Error Control Coding : Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, N.J. 07632, ch 11.
4. Lippman R.P., 1989, "Introduction to computing with neural networks", IEEE ASSP Magazine. Nov., pp. 54-66.
5. 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.
6. 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.