한국통신학회지 제 11 권 제 9 호 1994년 9월

《主 題》

음성인식 기술 현황

김 순 협
(광운대 컴퓨터 공학과)

■차 례■

Ⅰ. 서 론

언어 소통은 인간이 가지고 있는 기본적이며 가장 필수적인 능력들 가운데 하나이며, 음성은 사람들이 쉽게 어떤 다른 수단없이 정보를 전달할 수 있는 가장 중요하고도 유일한 방법이라고 말할 수 있다. 음성 파형 자체는 화자의 어조 그리고 화자의 감정같은 언어학적 정보를 전달한다. 음성에 의한 정보 교환은 우리들의 삶에서 매우 중요한 역활을 한다. 따라서 인간과 기계간의 자연스러운 통신을 위한 음성인식 시스템 개발은 인간이 기계를 사용하여 여러가지 일을 수행하기 시작할 때부터 제기된 과제로서 컴퓨터 및 통신정보 기술의 발전에 힘입어 많은 연구가 집중되고 있다.

그리고, 자연스러운 음성에 의해 맨-머쉰 인터페이스가 이루어지기 위해서 기계로 하여금 그 능력을 갖도록하기 위한 대어휘-연속음성-불특정화자-실시간 처리라는 목표아래 새로운 모델이나 기법의 연구가 계속되고 있다.

Ⅱ. 음성 인식 시스템

음성 인식 시스템은 형태, 화자의 수, 인식 방법등에 따라 다음 표 1과 같이 분류할 수 있다.

일반적인 음성인식 시스템은 그림 1에서 보는 것처럼 음성으로부터 음성패턴의 특징을 추출하여 표준패턴을 만든 후 미지의 음성이 입력되면 저장된 기준패턴과 비교하여 가장 유사한 표준패턴을 찾아 인식하는 과정으로 나누는데 이러한 알고리즘을 패턴 매칭 알고리즘이라고 부른다. 이러한 음성인식을 위한 패턴매칭(pattern matching) 알고리즘은 음성 패턴의 특징이 발성자 및 발음시간에 따라 변하는 것이 아니라 음성의 의미에 따라서만 변한다는 가정을 전제로 한다. 그러므로 동일한 의미를 갖는 음성을 여러사람이 발음하더라도 각 음성으로부터 추출한 음성 특징은 동일하다고 가정한다.

음성으로부터 특징을 추출하는 방법 중에 하나는 음성이 성도(Vocal tract)를 통하여 발성된다는 사실에 근거하여 성도를 필터로 가정한 후 그 필터 계수를 음성의 특징으로 삼는 것이다. 대개 필터는 AR (Auto Regressive) 혹은 ARMA(Auto Regressive moving average)모델에 의해 구성되는데 AR 모델의 대표적인 것으로 LPC(Linear Predictive Coding) 방식을 사용한다. 또한 사람의 청각특성을 이용한 분석 방식이 있는데 이는 저주파에서는 민감하게, 고주파 영역에서는 개략적으로 측정하는 것으로 Bark scale 혹은 mel scale 이라 한다. 이 scale에 따라 음성의 특징을 FFT에 의하여 주파수 영역으로 변환시킬 때 가중치를 가하

표 1. 음성인식 시스템의 분류
Table 1. Classification of speech recognition system

분 류	종 류	비 교
음성의 형태	격리단어	단어의 앞뒤에 묵음이 있다고 가정한 음성
	연결단어	격리단어가 자연스럽게 발음된 음성
	연속음성	연속단어가 자연스럽게 발음된 음성
화자의 수	화자종속	훈련된 화자의 음성으로 test 하는 실험
	화자독립	훈련하지 않은 화자의 음성으로 test 하는 실험
인식 방법	패턴매칭	음성의 특징을 표준패턴과 비교하여 인식하는 실험
	확률적 방법	음성의 발생 확률을 이용하는 방법

는 방법과 LPC에 의해 추출된 파라미터에 가중치를 가해서 시켜 파라미터로 구하는 방법이 있다. 최근에는 시간 영역의 동적특징을 주파수 영역의 특징들과 함께 사용한 연구 결과에 따르면 mel scale된 LPC cepstrum의 차를 음성 특징으로 사용하였을때 높은 인식율을 보이고 있다.

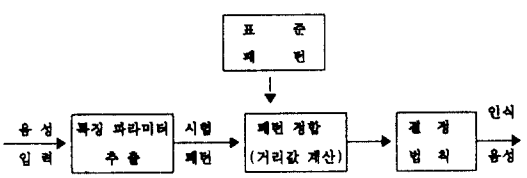


그림 1. 음성 인식 시스템
Fig 1. Diagram of speech recognition system

Ⅲ. 음성 인식 기술

현재 음성인식에 사용되는 알고리즘은 대체로 고전적인 DP(Dynamic Program) 알고리즘과 확률적 방법에 의한 HMM, 그리고 인공 신경망을 이용한 방법으로 나눌 수 있으며 각 알고리즘에 대한 설명은 다음과 같다.

1. DTW 알고리즘

인간이 발성하는 음성은 사람마다 발성 속도가 다르며, 같은 단어를 반복하여 발성하여도 발성 지속 시간의 차이가 생기게 되고 이로 인한 발성율의 변화는 음성 패턴에서 시간축의 비선형적 변동을 일으킨다.

그래서 이러한 변동 요인을 제거하는 시간축의 정규화 기법은 단독음의 인식수행에 있어서 매우 중요한 역활을 해왔다. 초기에 음성 패턴들 사이의 시간적 차이는 시간축을 선형적으로 변형(압축 또는 패창)함에 의해서 해결될 수 있다는 가정하에 선형적 정규화 기법을 사용하여 시간축의 정규화를 실행하였다. 그 결과 선형적 정규화 방법은 인식율을 크게 향상시킬 수 있었지만 시간축을 선형적으로 변형할 때 음성 정보가 손실되기 쉬운 단점과 비선형적인 시간축의 변동 패턴을 선형적으로 정규화 시키려는데 따르는 어려움 때문에 인식 알고리즘으로는 부적당함을 보였다. 따라서 비선형적인 시간축의 변동 패턴을 선형적으로 정규화 시키기 위해 패턴 정합 방식을 기본 이론으로 설정한 DTW 알고리즘을 사용하게 되었다. 이 알고리즘에서 시간축 변동은 몇가지 특정한 성질을 가진 비선형성 워핑(warping) 함수로 이루어져 있고 또 비교되는 두 음성의 패턴 사이의 시간적 차이는 워핑 함수를 이용하여 표준 패턴을 이 알고리즘에 적용하여 분석된 특징 벡터의 최대 일치점이 시험 패턴인 미지 단어의 특징벡터의 일치점에 도달하도록 표준 패턴의 시간축을 변화시켜 해결할 수 있다. 이때의 정규화 거리는 두 패턴들 사이의 오차 거리로부터 계산되고, 이 계산 거리의 최소화 처리는 DTW인식 알고리즘의 사용에 의해 매우 효율적으로 수행된다. 다음은 DP알고리즘 및 패턴 정합의 예를 보여주고 있다.

≪ DP matching algorithm ≫

Step 1) Initialization

$$g(1, 1) = 2d(1, 1) \qquad (1\text{-}1)$$

Step 2) DP－equation

for i = 1,...,I

for j = i − r,...,j + r

$$g(i, j) = \min \begin{matrix} g(i, j-1) + d(i, j) \\ g(i-1, j-1) + 2d(i, j) \\ g(i-1, j) + d(i, j) \end{matrix} \qquad (1\text{-}2)$$

Step 3) Time-normalized distance

$$D(A, B) = \frac{1}{N} g(I, J) \quad N = I + J \qquad (1\text{-}3)$$

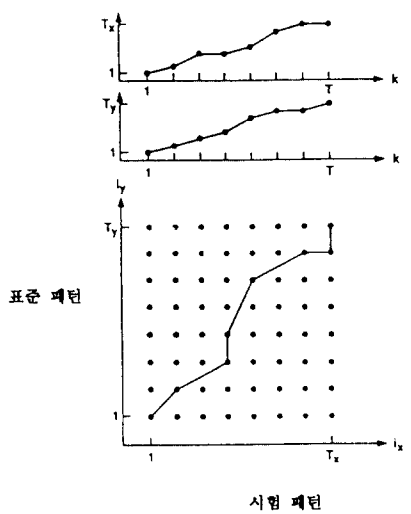


그림 2. 패턴 매칭의 예

Fig 2. Example of pattern matching

1-1. Level Building DTW 알고리즘

연결단어 인식방법은 Sakoe가 Two-level DP 방법이라 불리우는 알고리즘을 개발하면서부터 시작되었는데 이 알고리즘은 두 단계로 이루어진다. 이중 첫단계에서는 모든 표준패턴 단어들을 test 하고자 하는 문자열의 각 부분들과 정합을 시켜본다. 이 정합 과정은 단독어 인식에서의 비선형 time alignment 과정과 동일하다. 두번째 단계에서는 첫번째에서 구해진 거리값 정보를 토대로 미지의 입력 단어열과의 전체 거리값이 최소로 되는 단어들의 열을 찾아낸다.

LB DTW 알고리즘은 Sakoe가 제안한 two-level DP 계산량의 과도에 의한 단점을 극복한 방법으로서 입력 단어수를 지정할 수 있으며, 계산량을 줄이기 위해 여러가지 DP 범주 감축을 시도할 수 있는 장점이 있다. LB 알고리즘의 설명은 다음과 같다.

그림 3에서는 Super 표준패턴 R^S와 시험패턴 T간의 동적 워핑을 설정하는 예로서, 미지의 시험 패턴을 T(m), m = 1,...,M이라 하면 이 T는 L개의 표준 패턴 $R_{q(1)}(n)$, $R_{q(2)}(n)$,...,$R_{q(L)}(n)$의 연속과 대응된다고 할 수 있다. 여기서, $R_{q(k)}(n)$, k = 1,...,L은 V개의 표준패턴 R_v, v = 1,...,V중의 하나를 나타낸다. 또한 $\phi(l)$은 연결된 표준 패턴 l의 길이를 나타내며 v번째 표준패턴의 길이는 N_v로 표시한다.

$$\phi(l) = \sum_{k=1}^{l} N_{q}(k) \qquad (1\text{-}4)$$

$$\phi(0) = 0 \qquad (1\text{-}5)$$

이때 이 L개의 표준패턴의 연결을 $R_{q(1)q(2)}$,.....,$R_{q(L)}(n)$ 줄여서 R^S라고 하는 super 표준패턴을 정의 하면

$$R^S = R_{q(1)} + R_{q(2)} + \cdots + R_{q(L)} \qquad (1\text{-}6)$$

로 표현되며 DTW의 목적은 시험패턴의 시작과 끝이 각각 super 표준패턴의 시작과 끝이 대응되어야 하는 조건 (1-7)과 (1-8) 하에서

$$w(1) = 1 \qquad (1\text{-}7)$$

$$w(M) = \phi(L) \qquad (1\text{-}8)$$

$$D = \min_{w \in W} \left[\sum_{m=1}^{M} d(m, w(m)) \right] \qquad (1\text{-}9)$$

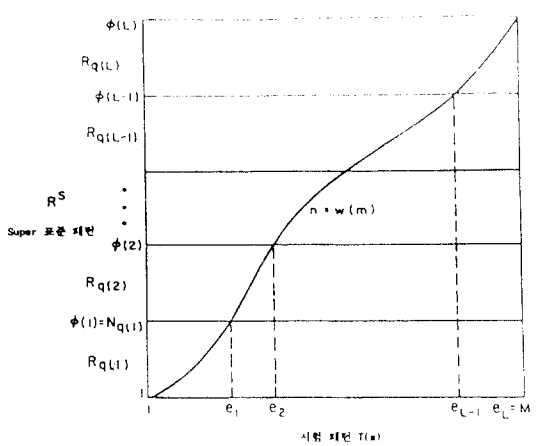


그림 3. LB DTW의 예

Fig 3. Example of LB DTW

을 만족하는 워핑 경로 n = w(m)을 구하는 것이다. 여기서, n은 super 표준패턴의 프레임(frame) index이다. 이때 Level Building의 관점은 super 표준패턴과 시험패턴을 격리단어 인식시 사용되는 DP의 경우와 같이 각각 한개의 표준패턴으로 보고 워핑하는 것을 super 표준패턴의 각 단어까지의 워핑을 연속으로 하는 것으로, 즉 level을 두어 워핑하는 방식으로 대체할 수 있다는 점이다. 이 생각은 super 표준패턴이 하나 이상인 경우에도 적용할 수 있다.

1-2. One Stage DP 알고리즘

미지의 입력패턴이 i = 1,...,N의 시간 프레임으로 구성되고 k = 1,...,K로 구분되는 표준패턴의 시간 프레임을 j = 1,...,J(k)로 표시한다. J(k)는 표준패턴 k의 길이이다. 이때 연결단어 인식의 목적은 입력패턴을 가장 잘 정합시키는 표준패턴 열 q(1),...,q(R)을 결정하는 것이다. 입력패턴의 i 프레임과 표준패턴 k의 j 프레임간의 국소 거리를(i, j, k)라고 표시하면 이 문제는 입력패턴과 미지의 표준패턴 열 사이를 가장 적절히 정합 시키는 grid point(i, j, k)의 집합으로 구성되는 워핑 경로를 찾는 것이다. 즉

$$\min_{w} \sum_{f}^{All} d(w(f)) \qquad (1\text{-}10)$$

를 구한다. 여기서 w(f) = (i(f), j(f), k(f)), f는 path element의 index이다. 시간 워핑 경로는 정합되는 패턴의 물리적 성질에 의해 여러가지 연속성 제한을 지켜야 한다. 연결단어 인식에서는 표준패턴 내부 천이 규칙과 표준패턴 경계 규칙을 구별하는 것이 편리하다. 표준패턴 내부천이 즉, W(f) = (i, j, k), j>1에서는

$$W(f-1) \in \{(i-1, j, k), (i-1, j-1, k), (i, j-1, k)\} \qquad (1\text{-}11)$$

또, 표준패턴 경계 즉, W(f) = (i, 1, k)에서는

$$W(f-1) \in \{(i-1, 1, k), (i-1, J(k^*), k^*)\ k^* = 1,...,K \qquad (1\text{-}12)$$

이런 제한을 가지는 연결단어인식 알고리즘은 다음과 같다.

$$\text{단계 1 : } D(1, j, k) = \sum_{n=1}^{j} d(1, n, k) \text{ 초기화} \qquad (1\text{-}13)$$

단계 2 : for i = 2 to N

for k = 1 to K

$$D(i, 1, k) = d(i, 1, k) + \min\{D(i-1, 1, k), D(i-1, J(k^*), k^*) : k^* = 1,...,k\} \qquad (1\text{-}14)$$

for j = 2 to J(k)

$$D(i, j, k) = d(i, j, k) + \min\{D(i-1, j, k), D(i-1, j-1, k), D(i, j-1, k)\} \qquad (1\text{-}15)$$

next j

next k

next i

단계 3 : 축적거리 array D(i, j, k)를 이용하여 표준패턴 끝 프레임에서 최소 전체거리를 가지는 표준패턴 끝 프레임으로부터의 최적 경로를 역추적한다.

이상과 같은 알고리즘의 단점은 1보다 큰 기울기를 가진 경로에 대해 입력 프레임 당 국소 거리의 수가 증가하는 반면 1보다 작은 경우에는 일정하게 유지된다는 점이다. 이점을 보완하기 위해 수평 방향, 대각선 방향, 수직 방향에 대해 각각 1+a, 1, b의 가중치를 곱하도록 한다. a, b는 실험에 의해 결정하고 전형적인 값은 a = 1, b = 1/2이다. 알고리즘의 실제적 수행에 있어서는 D(i, j, k)를 모두 저장할 필요없이

I. 축적거리의 column array D(j, k)
II. backpointer column array B(j, k)
III. "from template" array T(i)
IV. "from frame" array F(i)

만 있으면 되므로 메모리 저장량이 현저히 줄어든다. 즉 N*J*K 대신 2(J*K+N)개만 필요하다. 이외에도 최종 상태 문맥을 이용할 수 있도록 문맥 해석을 포함시킬 수 있다. H. Ney의 비교에 따르면 계산량에 있어 제한조건을 둔 Level-building DP에 비해 1.2배, 메모리 양에서는 약 1/9배 정도 소요된다. 이때 계산량은 One-Stage DP 계산시 제한조건을 두지 않은 경우이므로 적절한 방법을 취한다면 상당량을 감축할 수 있다.

- 천이 법칙

$$D(i, j, k) = d(i, j, k) + \min\{D(i-1, j, k), D(i-1, j-1, k), (D(i, j-1, k)\}$$

$D(i, 1, k) = d(i, 1, k) + \min\{D(i-1, 1, k), D(i-1, j(k^*), k^*) : K^* = 1,...,k\}$

k : word template

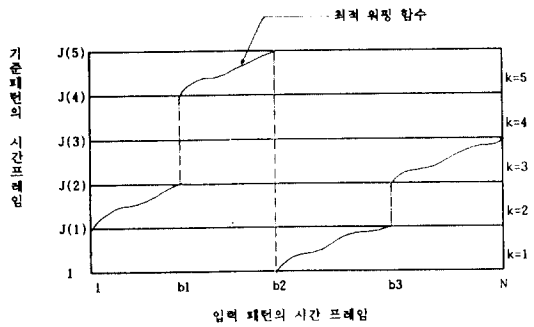


그림 4. one-stage DP의 예

Fig 4. Example of one-stage DP

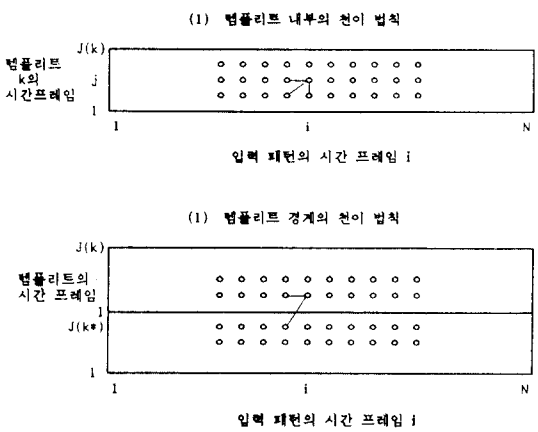


그림 4. (b) OSDP 알고리즘의 두가지 천이 규칙의 예

Fig. 4 (b) Example of two transition rule in One-Stage DP algorithn

2. VQ 이론

2-1-1. VQ의 개요

VQ(Vector Quantization)란 벡터의 열을 통신이나 디지탈 채널에 저장하기에 적당한 디지탈 열과 매핑하기 위한 시스템이다. VQ의 가장 큰 목적은 데이타 압축으로 데이타의 신뢰성을 잃지 않으며, 최대 한도로 데이타량을 줄이는데 있다. 데이타 압축에 기여한 Shannon의 이론에 의하면 스칼라 대신에 벡터를 코딩함으로써 더 좋은 성능을 얻을 수 있다는 것이다. 따라서, 음성 인식에 있어서 데이타 압축이라는 측면에서 표준 패턴을 생성하는데 VQ를 이용한다. 즉, 음성 인식에서의 VQ는 입력된 음성의 특징 벡터를 미리 저장해둔 특징벡터 중에서 장 잘 매칭되는 하나의 벡터와 매핑시켜 주는 것이다.

학습용 데이타들은 집단화(clustering) 기법에 코드북(codebook)으로 만들어지며, 입력 데이타는 코드북의 벡터들 중에서 최소의 거리값을 갖는 벡터로 양자화 된다. 그림 5에 VQ를 이용한 음성 인식 시스템을 나타낸다.

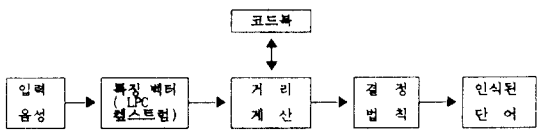


그림 5. VQ를 이용한 음성 인식 시스템

Fig 5. Speech recognition system using VQ

2-1-2. 거리값 계산

VQ 음성 인식 시스템에 있어서 거리 계산이란 학습용 데이타와 표준 데이타간의 거리 차이를 구하는 일이다. LPC 켑스트럼 학습용 데이타 집합$\{t_i\}, i = 1...I$라 하자. 이 벡터들은 어휘에서의 단어들이 다양한 화자에 의해서 발음 될 때 일어나는 LPC 켑스트럼이다. 표준 패턴이 코드북으로 구성되어 있으므로 학습용 데이타 T와 표준 패턴 $M = \{m_j : j = 1,...,J\}$와의 거리는 다음과 같이 정의 된다.

a) 학습용 데이타 T와 표준 패턴의 코드워드 m_j와의 거리 $d(t, m_j)$
$\{j = 1,...,J\}$ (J는 코드북 크기)를 구한다.

b) 학습용 데이타 T와 표준 패턴과의 거리 D는 다음의 $D = \min d(t, m_j) \{j = 1,...,J\}$으로 정의한다.

VQ의 주요 개념은 주어진 M에 대하여 가장 가까이에 있는 코드북 크기 $\{m_j\}$에 의해 학습 데이타 집합 벡터$\{t_i\}$의 각각에 대한 평균 거리가 최소가 되도록 LPC 켑스트럼 벡터로 최적의 코드북 집합을 결정하는 것이다. 공식적으로는 두 LPC 켑스트럼 벡터 m_j과 t_i간의 거리로서 $d(t_i, m_j)$라고 정의한다. 최적의 코드북의 집합을 구하는 식은,

$$|D_M| = \min\{1/I \sum_{i=1}^{I} \min_{1 \le j \le J} [d(t_i, m_j)]\} \quad (2\text{-}1)$$

로 나타낼 수 있다. 위 식 (2-1)에서 M 값(코드북 크기)에 대한 최적의 해를 발견할 때까지 학습을 반복한다.

2-1-3. 단어 인식

시험 음성 패턴과 모든 표준 음성 패턴과의 거리값을 구한 후 시험 음성 패턴은 가장 작은 거리값을 갖는 표준 패턴의 단어를 인식단어로 인식한다.

2-2. MSVQ (Multi Section VQ)에 의한 음성 인식

2-2-1 MSVQ의 개요

단어 단위의 음성 인식에서는 발성 속도에 따른 시간변동을 줄이기 위해 DP 정합이 많이 이용되고 있으나, 시간축의 선형변환에 따른 계산량이 증가한다. 그러므로, 시간 정규화가 필요 없는 단어 별로 작성된 VQ 코드북에 의해 단어들의 음향적인 특성만을 비교하는 방법을 이용한다. 그러나, VQ 코드북에는 시간적 정보가 포함 되어 있지 않아서 음향적 특성이 유사한 단어들 사이에 부정확한 인식이 일어난다. 따라서 한 단어를 발성순서에 따라 몇개의 구간(section)으로 나누고 구간 별로 독립된 코드북을 작성함으로써 시간적 정보를 포함시키는 MSVQ(Multi-Section VQ) 방법이 Burton등에 의해 제안되었다.

2-2-2. MSVQ 코드북 작성

VQ 코드북의 계열로써 시간 변화 패턴을 고려하는 방법을 MSVQ 코드북이라고 한다. MSVQ 코드북은 그 단어를 동일 길이의 구간으로 나누고 각 구간마다 집단화 기법을 써서 작성한다. 음성 데이타의 전체 프레임을 구한후 일정한 구간으로 나누어 코드북을 작성 하였다.

일반적인 MSVQ 코드북을 작성하는 과정에 있어서 한 단어 T를 1회 발성한 음성을 학습열로 사용한다. 한 프레임을 LPC로 분석하여 얻은 LPC 켑스트럼 벡터를 T라 하면 1회 발성한 음성은

$$T = \{t_1, t_2, t_3, ..., t_I\} \quad (2\text{-}2)$$

와 같이 나타낼 수 있다. 인식 대상 어휘가 모두 V개의 단어로 되어 있을때 각 단어마다 N회 발성된 음성으로 MSVQ 코드북을 구성하기 위해 이들을 J개의 구간으로 나눈다.

$$T_n^v = (T_n(1)\ T_n(2) ... T_n(J))^v \ (v = 1, 2, ..., V) \quad (2\text{-}3)$$

만약, 전 프레임을 J개의 구간으로 등간격으로 나눈 후 j번째 구간의 마지막 프레임 값을 e(j)라 하면, e(0) = 0이고 e(J) = I이다. 따라서,

$$T_n(j) = \{t_m\}\ (i = e(j-1)+1, e(j-1)+2, ..., e(j)) \quad (2\text{-}4)$$

와 같이 각 구간을 벡터 시퀀스로 표시할 수 있다.

그림 6에는 4-MSVQ 코드북을 작성하는 과정이 나타나 있다. 그림 6에서 보는 바와 같이 각 단어 마다 6회 발음되었으며, 각 구간은 프레임수가 같고 4구간으로 되어 있으므로 4개의 독립된 VQ 코드북의 조합에 의해 MSVQ 코드북이 구성 된다. 구간 j에 해당하는 학습열의 집합을 $T_s(j)$라 하면

$$T_s(j) = \{T_n(j)\}\ (j = 1, 2, 3, 4)\ (n = 1, ..., 6) \quad (2\text{-}5)$$

이 된다. 각 구간에 대한 코드북 m_j는 $T_s(j)$를 학습열로 하여서 집단화 기법에 의해 작성된다. 이 과정을 통해 작성된 코드북의 계열

$$M = \{m_1, m_2, m_3, m_4\} \quad (2\text{-}6)$$

는 MSVQ 코드북을 의미한다.

8-MSVQ 코드북을 만드는 과정은 4-MSVQ 코드북을 만드는 과정과 동일하나 단지 구간의 수를 4에서 8로

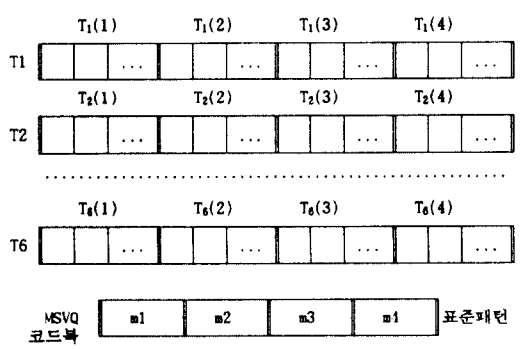


그림 6. 4-MSVQ 코드북 작성
Fig 6. MS VQ codebook generation

늘리는 것만이 다르다. 8-MSVQ의 경우, 각 단어마다 6회 발성된 음성으로 코드북을 작성하기 위하여 8개의 구간으로 나눈다.

$$T_n = \{T_n(j)\}\ (j = 1,...,8)\ (n = 1,...,6) \qquad (2\text{-}7)$$

식에서처럼 8 구간의 코드북이 다음과 같이 만들어진다.

$$M = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\} \qquad (2\text{-}8)$$

2-2-3. 단어 인식 방법

일반적인 VQ방법에 의한 거리값 계산과 동일하나 MSVQ에서는 구간 별 거리 값을 합하여 총거리 값으로 최적의 코드북을 찾는 것이 다르다. 인식 절차를 보이면 다음과 같다. 인식하고자 하는 시험 입력 음성 T는 먼저 전체 프레임을 구한후 J개의 구간으로 나눈다. 이것을 벡터 계열로 나타내면

$$T = \{T(1), T(2), T(3),...,T(J)\} \qquad (2\text{-}9)$$

가 되고, T(j)는 j번째 구간을 구성하는 프레임들로부터 LPC 켑스트럼 분석을 통해 구간 특징 벡터의 계열이다. 그래서 T(j)는

$$T(j) = \{t_{e(j-1)+1}, t_{e(j-1)+2},\cdots,t_{e(j)}\} \qquad (2\text{-}10)$$

로 표시된다. 이들 각 구간에 대한 특징 벡터들을 표준 패턴에 대응하는 위치에서의 구간과 그 앞뒤 구간과의 거리 비교를 통해 최소인 코드워드와 일치한 것으로 보고 거리 계산을 하며, 전체 평균거리인 D_{av}^{v} 를 구한다. 어떤 단어 v에 대한 표준 패턴과 전체 평균 거리 D_{av}^{v} 는

$$D_{av}^{v} = \frac{1}{I}\sum_{j=1}^{J} d_j(T(j), m_j) \qquad (2\text{-}11)$$

이다.

$$d_j = \sum_{i=e(j-1)+1}^{e(j)} \min_{l} d(t_i, m_{jl}^{v}) \qquad (2\text{-}12)$$

식 (2-12)에서 m_{jl}^{v}는 단어 v의 j번째 구간의 l번째 코드워드값으로 나타낸다. 이상의 과정을 모든 단어의 표준 패턴에 대하여 반복하여 최종 적으로 전체 평균 거리가 최소인 단어가 인식된 것으로 한다.

3. HMM(Hidden Markov Model)

Markov 체인(chain)의 기본적인 이론이 수학자들에 의해 알려진 것은 80여년전부터 알려져 있었지만 1960년대에 와서야 Markov 모델(model)의 파라메타 최적화 방법이 발견되어 1975년 Carnegie-Mellon 대학의 Baker와 IBM의 Jelinek 등에 의하여 음성 신호처리 분야에 처음으로 도입되었다. 현재까지 HMM은 음성인식에서 주도적인 역할을 해온 DTW 알고리즘의 단점인 많은 계산량과 연속 음성인식에 적용이 곤란함을 보완할 수 있는 방법으로 각광받아 오고 있다.

HMM은 관측이 불가능한 과정을, 관측이 가능한 다른 과정을 통해 추정하는 이중 확률 처리로서 음성과 같이 다변성이고 발생 과정을 알 수 없는 process를 표현하는데 적당한 방법이다. 그리고, HMM은 안정된 구간이나 특이한 구간의 판별, 구간에 따른 연속적 변화특성의 표사 및 각 구간에 대한 공통적인 단구간(short term) 모델을 효과적으로 처리할 수 있다.

3-1. Markov 과정

영어의 알파벳이 단어를 이룰때 q 다음에는 반드시 u가 나오기 때문에 단어에서 u가 나올 확률은 그 앞에 어떤 알파벳이 오느냐에 영향을 받는다. 이와 같이 현재 상태의 확률이 바로 앞 상태의 영향을 받는 확률적 과정을 Markov 과정이라하고 다음과 같이 N개의 상태들의 집합과 각각에 존재할 확률이 주어진 시스템으로 설명된다. 간단한 예로 이와같은 Markov 체인에서 한 상태에 존재할 확률은 다음과 같이 표시할 수 있다.

$$P[q_t = S_i | q_{t-1} = S_j, q_{t-2} = S_k,\cdots] = P[q_t = S_i | q_{t-1} = S_j] \qquad (3\text{-}1)$$

여기서 윗식의 우측항은 시행 횟수에 대해 독립적이므로 천이 확률 a_{ij}는

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i],\ 1 \le i, j \le N \qquad (3\text{-}2)$$

으로 표시할 수 있고

$$a_{ij} \ge 0,\ \sum_{j=1}^{N} a_{ij} = 1 \qquad (3\text{-}3)$$

의 특성을 갖고 있기 때문에 일반적인 확률적 제한에

따른다. 이와같은 확률처리 출력은 각 시행에서의 상태집합이므로 관측 가능한 Markov 모델이라고 한다.

3-2. HMM 알고리즘

HMM의 일반형을 예를 들어 설명한다. 지금 N개의 상자가 있고 각각의 상자안에 M 종류에 구슬이 들어있다. 실험자가 볼 수 없는 N개의 상자중 하나의 상자를 선택하고, 그 상자 안에서 1개의 구슬을 꺼낸다. 구슬의 종류를 기록하고 그 구슬을 꺼낸 상자에 다시 넣는다.(복원추출) 이러한 일련의 자업을 T회 반복한다. 이때 t회째의 작업에서 상자 q_i를 선택할 확률 a_{ij} (이 확률은 t-1번째에 선택한 상자 qi에 영향을 받음 : Markov 과정)와, 그 상자 q_i 중에서 구슬 v_k를 선택할 확률 $b_i(k)$로 부터 T회 반복하여 얻은 구슬열 O_1, O_2,...,O_T가 나올 확률을 얻을 수 있다.

확률 $A=\{a_{ij}\}$, $B=\{b_i(k)\}$와 제일 처음 시행시에 상자를 선택할 확률 $\pi=\{\pi_i\}$를 모델 파라메타로 하여 이 모델을 HMM 이라 부른다.

여기서, H(Hidden) '숨겨진'이 의미하는 것은, 구슬의 종류의 기록만을 볼 수 있으며 어느 상자를 선택하여 나온 구슬인지는 알 수 없다는 것이다.

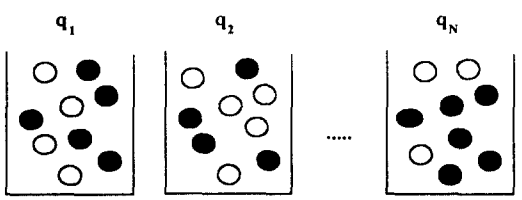


그림 7. HMM의 일반형
Fig 7. HMM diagram

T : 관측열의 길이 ; 시행 횟수
O_1,...,O_T : 관측열(observation sequence) ; 구슬열
N : 상태(state)수 ; 상자 수
M : 관측심볼수 ; 구슬 종류의 수
$Q=[q_1,...,q_n]$: 상태(state) ; 상자 이름
$V=\{v_1,...,v_M\}$: 관측 심볼(observation symbol) ; 구슬 이름

$A=\{a_{ij}\}$, $a_{ij}=P(q_j \text{ at } t+1 \mid q_i \text{ at } t)$: 상태 q_i에서 상태 q_j로 전이 확률 : t번째 시행에서 상자가 q_i가 선택되었을 때 t+1번째에 상자 q_j를 선택할 확률

$B=\{b_i(k)\}$, $b_j(k)=P(v_k \text{ at } t \mid q_j \text{ at } t)$: 상태 q_j에서 관측심볼 v_k의 출력확률
: t번째 시행에서 상자 q_j가 선택되고 그 상자에서 구슬 v_k가 선택될 확률 $\pi=\{\pi_i\}$, $\pi_i=P(q_i \text{ at } t=1)$: 초기 상태 전이 확률
: 최초 시행에서 상자 q_i가 선택될 확률

이상의 정의를 이용하면 HMM은 $\lambda=\{A, B, \pi\}$로 표시할 수 있는데 이 모델을 실제 응용하는데는 다음과 같은 세가지 해결해야 할 문제가 있다.

1. 관측열 $O=O_1, O_2,...,O_T$,와 모델이 주어졌을 때 관측열이 나올 확률 $P(O|\lambda)$를 계산하는 문제 (forward backward algorithm)
2. 관측열 $O=O_1, O_2,...,O_T$,와 모델이 주어졌을 때 최적의 상태열 $I=i_1, i_2,...,i_t$를 구하는 문제(Viterbi algorithm)
3. $P(O|\lambda)$를 최대화 하기 위해 모델 파라메터 $\lambda=(A, B, \pi)$를 조정하는 문제(Baum-Welch reestimation algorithm)

HMM은 인식과 관련된 forward 알고리즘, backward 알고리즘, Viterbi 알고리즘과 학습관련된 Baum-Welch reestimation 알고리즘이 있다.

3-2-1. forward 알고리즘

① $\alpha_1(i)=\pi_i b_i(O_1)$, $1 \le i \le N$

② For $t=1, 2,...,T-1$, $1 \le j \le N$

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(O_{t+1})$$

③ $$P(O \mid \lambda)=\sum_{i=1}^{N}\alpha_T(i)$$

$\alpha_t(i)=P(O_1, O_2,...,O_t, q_t=S_i|\lambda)$: 1~t까지 부분 관측열(obervation sequence)의 확률

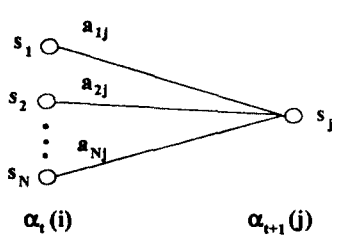


그림 8. Forward 알고리즘의 개념도
Fig 8. Diagram of forward algorithm

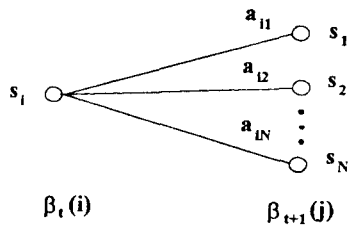


그림 9. Backward 알고리즘 개념도
Fig 9. Diagram of backward algorithm

3-2-2. backward 알고리즘

① $\beta_T(i) = 1,\ 1 \le i \le N$

② For $t = T-1, T-2, \ldots, 1\ 1 \le i \le N$

$$\sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

③ $$P(O|\lambda) = \sum_{i=1}^{N} \pi_j b_j(O_1) \beta_1(j)$$

$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T, q_t = S_i|\lambda)$

: t+1 ~ T까지 부분 관측열의 확률

3-2-3. Viterbi 알고리즘

① Initialization

$\delta_1(i) = \pi_i b_i(O_1),\ 1 \le i \le N$

$\Psi_1(i) = 0$

② Recursion

For $2 \le t \le T,\ 1 \le j \le N$

$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(O_t)$

$\Psi_t(j) = \operatorname{argmax}_i [\delta_{t-1}(i) a_{ij}]$

③ Termination

$P^* = \max_i [\delta_T(i)]$

$i^*_T = \operatorname{argmax}_i [\delta_T(i)]$

④ Path backtracking

For $t = T-1, T-2, \ldots, 1$

$i^*_t = \Psi_{t+1}(i^*_{t+1})$

3-2-4. Baum-Welch reestimation 알고리즘

① $\overline{\pi_i} = \gamma_1(i),\ 1 \le i \le N$

② $$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

③ $$\overline{b_j(k)} = \frac{\sum_{t=1,\ O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)}$$

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P(q_1 q_2 \cdots q_t = i, O_1, O_2 \cdots O_t|\lambda)$$

: 시간 t에서 최대 경로 확률

여기서,

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = P(q_t = S_i|O, \lambda)$$

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}$$

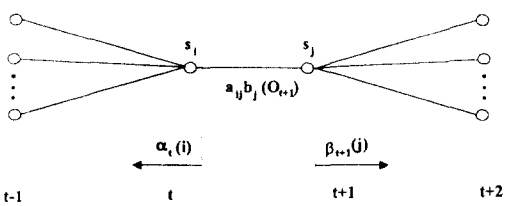


그림 10. Baum-Welch reestimation 알고리즘 개념도
Fig 10. Diagram of Baum-Welch reestimation

※ 확률적 제한

$$\sum_i \pi_i = 1,\ \sum_j a_{ij} = 1,\ \sum_{k=1}^{M} b_j(k) = 1,\ \gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j)$$

3-3. HMM을 이용한 음성 인식

HMM을 이용한 음성 인식 시스템에서 각 영역에 대한 HMM들은 블랙 박스와 같은 역할을 하여 미지

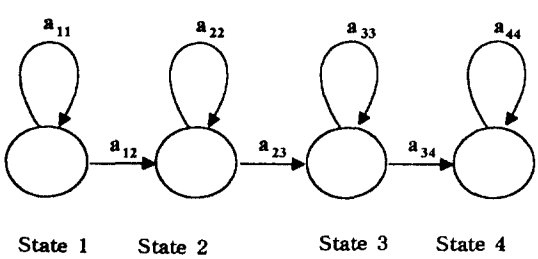


그림 11. Left-to-Right 상태 전이 모델
Fig 11. Model of Left-to-Right state transition

의 데이타를 각 모델에 적용했을 때 가장 큰 값의 확률을 나타내는 것을 인식된 것으로 한다. 시간에 따른 음성의 특성변화를 효과적으로 모델링하기 위해 left-to-right 상태 전이(state transition) 모델을 사용하여 상태 자체에 시간적 순서를 내포하도록 하며, 상태수는 음성 신호에서 순간적으로 크게 변화하는 부분(temporal event set)의 수를 고려하여 설정한다.

◆음성 인식에 적용 예◆

▶ 대상어 : 숫자음 10개('영'~'구'), 3 화자, 각각 2회 발성

▶ 특징벡터 : 10차 LPC 켑스트럼(1 프레임 = 128 샘플)

▶ 코드북 생성 알고리즘 : K-means
코드북 크기 : 64

▶ HMM 상태(state) 수 : 4, 전이 : left-to-right

□코드북 생성과정

① 전처리
화자 A⇒1회 : '일'~'구' 2회 : '일'~'구'
화자 B⇒1회 : '일'~'구' 2회 : '일'~'구'
화자 C⇒1회 : '일'~'구' 2회 : '일'~'구'
총 2*2*10 = 60개의 10차 LPC 벡터

②Clustering
K-means⇒64개의 코드워드 생성
(각 코드북에 1~64로 번호매김)

□HMM 생성(학습)
숫자음 '일'에 대한 모델 생성
표준패턴
화자 A : 1회 : '일', 2회 : '일',
화자 B : 1회 : '일', 2회 : '일'
화자 C : 1회 : '일', 2회 : '일' 총 6개

① 전처리
⇒ 한 프레임 당 10차 LPC 계수 생성

② 벡터 엔코딩
시험패턴과 가장 가까운 코드북의 번호 출력
⇒ 관측열 생성($O_1, O_2, ..., O_{30}$)

④ 모델 파라메터 초기화

▶ 초기 상태 전이확률
$\pi_1 = 1, \pi_2 = 0, \pi_3 = 0, \pi_4 = 0$: $\sum_i \pi_i = 1$

▶ 상태 전이확률 (N = 4, x = random)
a_{11} = x a_{12} = x $a_{13} = 0$ $a_{14} = 0$
a_{21} = x a_{22} = x $a_{23} = 0$ $a_{24} = 0$
a_{31} = x a_{32} = x $a_{33} = 0$ a_{34} = x
a_{41} = x a_{42} = x $a_{43} = 0$ $a_{44} = 1$
: $\sum_j a_{ij} = 1$

▶ 심볼의 출력 확률(N = 4, M = 64, x = random)
$b_1(1)$ = x, $b_1(2)$ = x, ~, $b_1(64)$ = x
$b_2(1)$ = x, ~,$b_2(64)$ = x
$b_3(1)$ = x, ~,$b_3(64)$ = x
$b_4(1)$ = x, ~,$b_4(64)$ = x

: $\sum_{k=1}^{M} b_j(k) = 1$

Forward, Backward 알고리즘으로 α, β 값 계산
Baum-Welch reestimation 알고리즘 반복수행
⇒숫자음 '일'에 대한 HMM(A, B, π) 생성
같은 방법으로 '이'~'구' 각각의 모델생성

□인식

① 미지의 시험패턴 입력

② 전처리및 벡터 엔코딩⇒시험패턴에 대한 관측열 생성

③ 각 모델에 적용하여 각각의 Viterbi score 계산
⇒가장 큰 값을 나타내는 모델의 인덱스 출력

HMM의 개선방법에는 정적 스펙트럼과 동적 스펙트럼의 특징 벡터를 조합여 모델링하는 DHMM(Dynamic HMM)과 인식률이 출력 확률에 의해 많은 영향을 받는다는 점을 이용하여, 자기 영역의 데이타 영향은 물론 다른 영역의 데이타들의 그 영향까지 고려하여 출력 확률 값을 재조정하는 Corrective training 방법, 그 밖에 뉴럴 네트웍과 조합등의 여러가지 방법이 있다.

4. 신경망(NN)을 이용한 음성 인식

최근 인간의 뛰어난 인지능력을 음성인식에 적용하기위해 인공신경망을 음성인식시스템에 적용한 연구가 많은 연구가들에 의해 시도되고 있다. 인공신경망을 이용한 음성인식 시스템을 구현하기 위해서는 음성신호로부터 인식하고자 하는 패턴에 대한 안정된 특징패턴을 추출할 수 있어야 한다. 그리고 음성신호의 시간축 왜곡 현상(Time Warpping), 즉 동일한 음성을 발성할 때 각 발성 시 음성의 길이가 다르다는 특징을 처리해야 하므로 음성인식기는 시간축 왜곡을 흡수할 수 있어야 한다. 그리고 인식하고자 하는 음성패턴의 시작과 끝을 미리 알 수가 없으므로 분석구간 내에 어느 부분에 단어가 존재한다 하더라도 찾

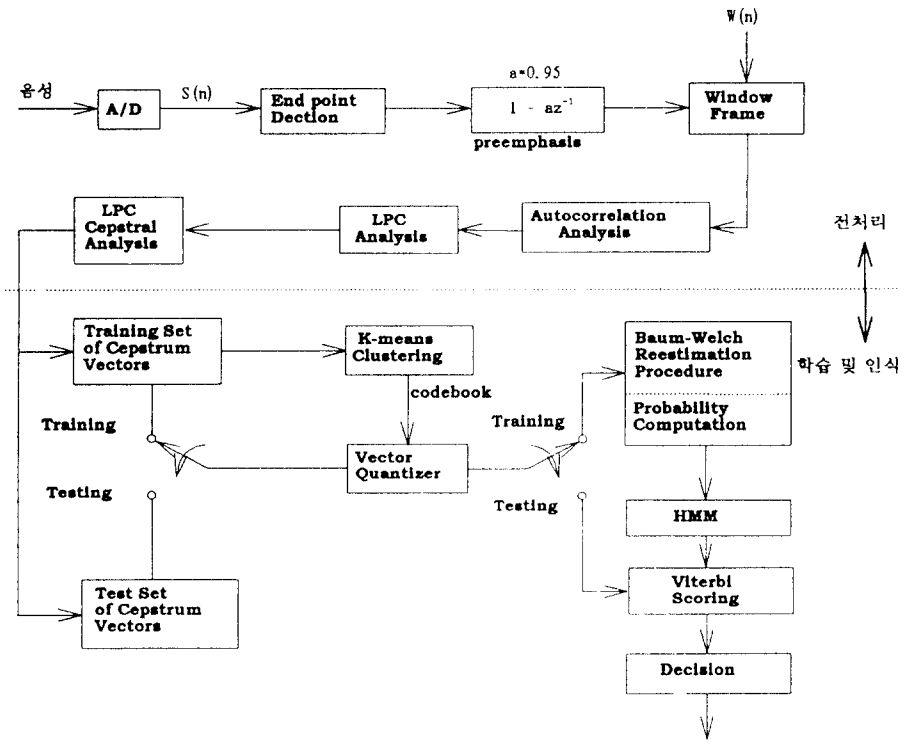


그림 12. HMM을 이용한 음성인식 블록도
Fig 12. Blockdiagram of speech recognition system using HMM

야낼 수 있어야 하므로 음성의 시변특성을 흡수 할 수 있어야 한다. 또한 화자에 따른 주파수 변화를 흡수할 수 있어야 한다. 이런기능을 갖도록 다양한 신경망들이 연구되고 있다. 이들 중 가장 많은 연구가 진행되는 것은 TDNN(Time Delay Neural Network)을 이용한 것과 작은 시간 슬라이스 단위의 음성인식 신경망이 있다. 또한, 최근 Context 정보를 이용할 수 있는 신경망들이 활발히 연구되고 있다.

음성의 시변특성을 흡수하기위한 신경망으로는 TDNN이 있고 각 발성화자에 따라 주파수 특성, 속 포만트의 변화와 음성의 시변특성을 흡수하기 위해서 시도된 FTDNN(Frequency TDNN)과 BWNN(Block Window NN)이 있다. 그리고 시간지연 신경회로망의 단점을 해결하고 Context 정보를 이용하여 연속어 인식에서도 쓰일 수 있도록 설계된 시간 슬라이스 단위의 TSRR (Time-Slice Recurrent Recognizer)이 있다. 본 논문에서는 TDNN을 이용한 각 신경회로망과 시간 슬라이스 단위의 인식 신경회로망인 LVQ2, TSRR, 그리고 기존의 음성인식 알고리즘과 신경망이 결합을 이용한 음성인식에 대하여 알아본다.

4-1. TDNN

음성에 있어서 주파수에 의한 특성, 음성 발성 시간을 모두 포함할 수 있는 MLP(Multi Layer Perceptron)의 변형으로서 TDNN이 waibel에 의해 개발되었고 특히 시간적인 특징이 포함되도록한 알고리즘으로 음소나 단어에 있어 높은 인식율을 보이고 /b/, /d/, /g/의 3개의 음소를 인식하는 계층형 네트웍의 구조는 다음과 같다.

입력층은 15프레임의 특징벡터(15프레임 × 16차 = 240개)를 받고 출력층의 각 유니트는 /b/, /d/, /g/의 3음소를 나타낸다. 출력층의 유니트가 3음소에 대한 것밖에 없는 것은 다음과 같은 이유때문이다. 모든 음절을 하나의 신경회로망으로 인식한다는 것은 식별

해야 하는 영역수가 너무 많기 때문에 어렵다. 그래서 전음소를 몇개의 음소그룹으로 나누어 그룹마다 음소인식을 행하는 네트웍(서브 네트웍)을 구성하여 학습한 후 통합하여 전체음소를 인식하는 방법을 사용하고 있다. 이와같은 방식을 채용하면 각 서브네트웍이 문제를 나누어 해결하기 때문에 사용이 용이하다. 시간지연 신경회로망은 다층 퍼셉트론을 음성에 적용하기위해 변형한 네트웍이다. 보통의 다층 네트웍이 인접하는 층 사이의 유니트간의 결합이 모두 독립적인데 비하여 시간지연 네트웍은 다음과 같은 제약이 도입되고 있다. 입력층의 유니트로부터 제1은닉층의 Unit로의 결합은 입력층의 3프레임분 $3 \times 16 = 48$개의 유니트로부터 1은닉층의 종축으로, 위의 그림에서는 8개의 유니트로 완전연결 되어있다. 여기서 입력층의 창을 3프레임으로 잡은 이유는 3프레임이면 한 음소의 안정상태를 충분히 나타낼 수 있기 때문이다. 그리고 그림 13에서와 같이 1프레임씩 시간방향으로 이동하면서 정보전달이 상위 층으로 전달된다. 이 네트웍은 역전파 학습알고리즘에 의해 학습이 진행된다. 이때 실제로 존재하는 연결강도는 각 층의 시간창의 모든 유니트와 상위층의 1프레임의 종축 유니트와의 연결만이 존재하기 때문에 각 층의 시간창과 상위층의 종축과의 연결강도 창이 시간 방향으로 이동하면서 연결강도의 평균 변화율을 변화시키면서 학습되어진다. 이와같은 시간지연 신경망의 서브네트웍은 각 음소 그룹에 대하여 비교적 높은 인식률을 얻을 수 있음을 실험적으로 입증되었다. 이 신경회로망을 전체 음소인식에 적용하기 위해서는 우선 각 음소그룹에 서브네트웍을 학습시킨 후 미지의 입력된 음소가 어느 음소그룹에 속하는가를 식별할 수

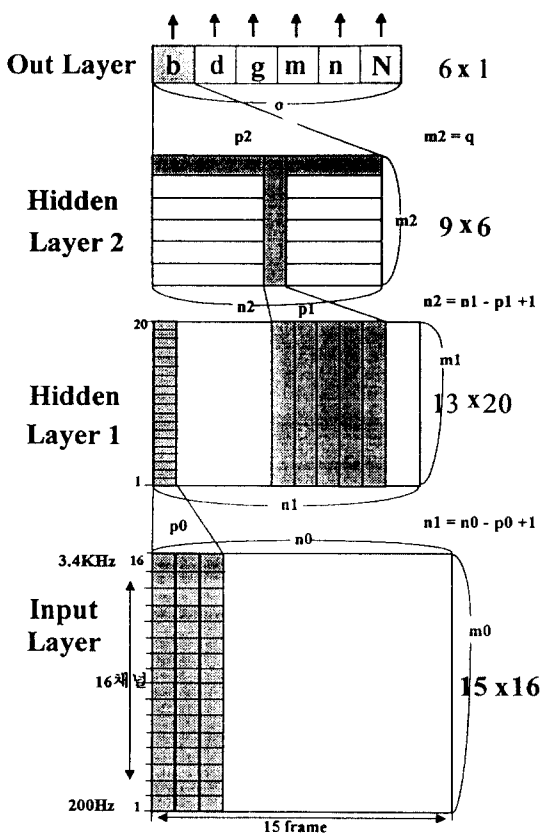


그림 13. TDNN의 구조
Fig 13. Structure of TDNN

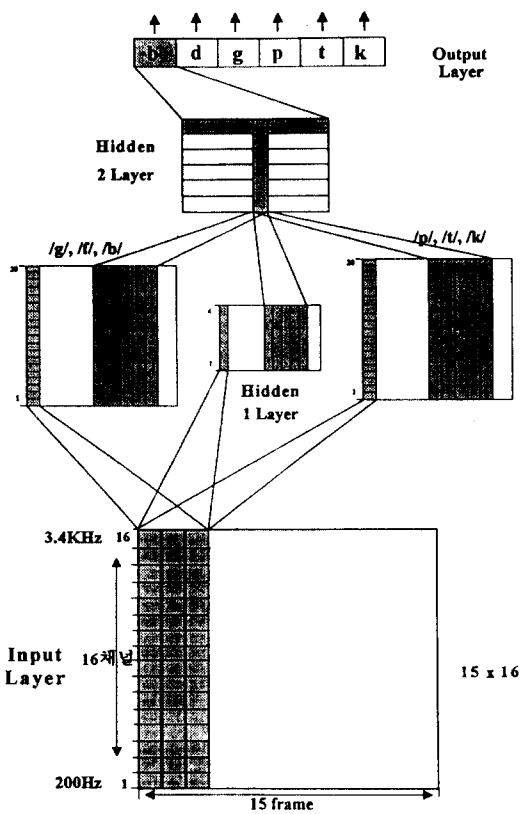


그림 14. Glue TDNN의 구조
Fig 14. Structure of Glue TDNN

있는 그룹식별 네트웍을 학습시킨다. 그리고 전체음소를 인식하기 위해서는 모든 서브네트웍을 결합한 통합네트웍을 구성한 후 학습된 각 서브 네트웍의 연결강도를 2은닉층까지 복사하고 2은닉층으로부터 출력층사이의 연결강도에 대하여 통합네트웍을 학습을 시킴으로서 음소인식 시스템을 생성시킬 수 있다.

4-2. FTDNN

시간지연 신경회로망에서는 시간지연의 문제점을 해결하지만 화자에 따른 주파수 변화를 흡수하지는 못하는 단점이 있다. 이 문제점을 해결하기 위해서 이 입력층에 4개의 채널로 이루어진 주파수 창과 3개의 프레임으로 이루어진 시간창을 두어 음성의 시간변화와 주파수 변화를 흡수하도록 구성된 신경회로망이 FTDNN이다. FTDNN의 구조는 다음과 같다.

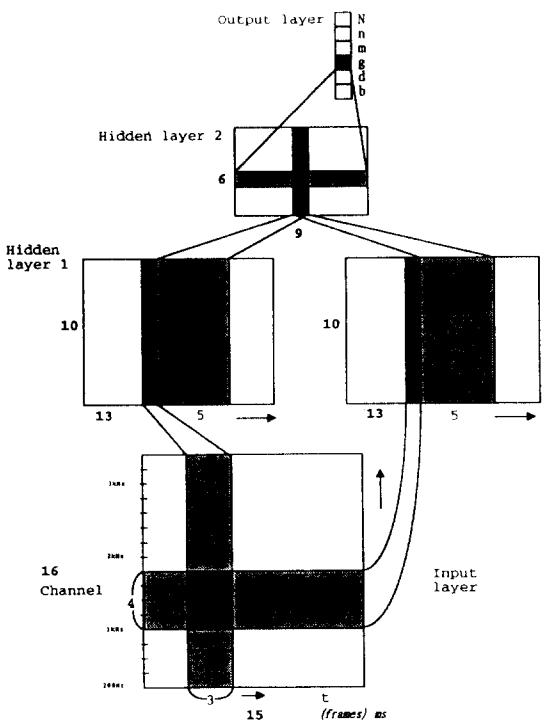


그림 15. FTDNN의 구조
Fig 15. Structure of FTDNN

음성의 특징 파라메터로는 시간지연 신경회로망과 같은 것을 사용하고 입력층도 동일하게 구성되어 입력으로 240개(15프레임×16차원)의 특징이 입력된다. 그러나 1은닉층은 시간창에 의한 모듈과 주파수창에 의한 모듈로 나누어진다. 시간모듈은 입력층의 3프레임의 시간창이 하나의 종축 유니트들과 완전연결되어 시간방향으로 이동하면서 정보의 압축이 일어난다. 주파수 모듈은 4채널의 창이 하나의 종축 유니트들과 완전연결되어 주파수 방향으로 이동하면서 정보의 압축이 일어난다. 입력층의 주파수 창의 주파수 구간을 4채널로 잡은 이유는 화자에 따른 주파수 변화가 4채널내에서 일어난다는 점에 기인한다. 전체 음소 인식기를 구성하기 위해서는 TDNN과 같이 각 서브 네트웍과 각 그룹식별 네트웍을 먼저 학습시킨 후 통합 학습을 함으로서 구성할 수 있다.

4-3. 시간 슬라이스 단위의 음성 인식기

4-3-1. LVQ

T. Kohonen은 작은 시간 슬라이스 단위의 인식기법으로 LVQ(Learning Vector Quantization)1, LVQ2 알고리즘을 제안했다. 이 알고리즘은 최적의 Bayes 결정법칙에 가능한한 근접한 Hyper plane을 발견하기 위해 제안된 것이다. 이 알고리즘들은 기존의 VQ 알고리즘이나 자기 조직화 알고리즘을 이용하여 각 클래스를 대표하는 코드북 벡터, M_i를 작성한 후 각 코드북 벡터에 각 클래스를 레이블링함으로써 초기화된다. LVQ1은 다음과 같다.

$$m_i(t+1) = m_i(t) + \alpha(t)[X_i(t) - m_i(t)] \quad (4\text{-}1)$$

만일, X_i가 올바르게 분류되었을 경우

$$m_j(t+1) = m_j(t) - \alpha(t)[X_i(t) - m_j(t)] \quad (4\text{-}2)$$

만일, X_i가 B 클래스로 오분류되었을 경우

$$m_i(t+1) = m_i(t) \quad (4\text{-}3)$$

다른 코드북 벡터

여기서, $\alpha(t)$는 학습률이고 LVQ1 알고리즘은 미세조정 부분이므로 0.01또는 0.02로 초기화되어 단조감소 한다. 그러나, LVQ1은 학습패턴 중 특별한 한개가 패턴이 코드북 벡터와의 거리가 클경우 각 클래스를 경계짓는 Hyper plane을 흐트려 뜨리는 결과를 낳는다. 이 결과 인식 시스템은 오히려 LVQ1 알고리즘으로 인해 분류에러가 증가되는 단점이 있다. 그래서, 이와같은 단점을 해결한 LVQ2 알고리즘이 다시 제안되어 있다.

오분류된 데이타에 대해서 다음과 같은 제약이 따른다.

1) 가장 가까운 Class j가 오분류된 Class이다.
2) 다음으로 가까운 Class i가 맞는 Class이다.
3) 학습 벡터, X_t가 m_i와 m_j의 중간지점에서 적당한 창안에 있어야 한다.

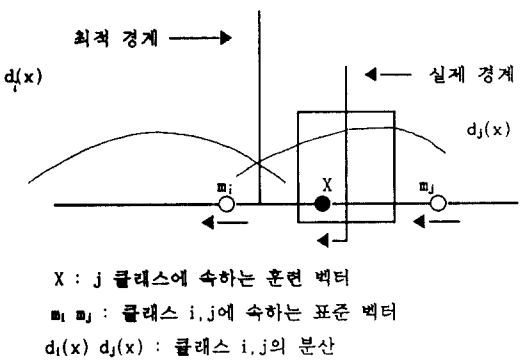


그림 16. LVQ2 모델
Fig 16. Model of LVQ2

$$m_i(t+1) = m_i(t) + \alpha(t)[X_t(t) - m_i(t)] \quad (4\text{-}4)$$

만일, X_t가 올바르게 분류되었을 경우

$$m_j(t+1) = m_j(t) - \alpha(t)[X_t(t) - m_j(t)] \quad (4\text{-}5)$$

만일, X_t가 j 클래스로 오분류되었을 경우

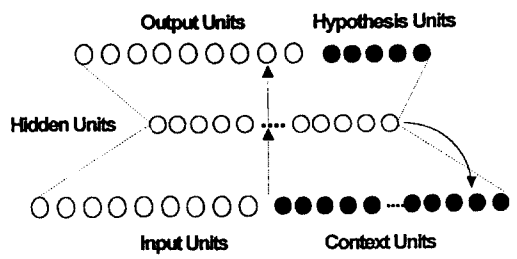
여기서, X_t는 클래스 i에 속한 학습벡터이며 m_j는 틀린 클래스의 코드북 벡터, m_i는 맞는 클래스의 코드북 벡터이다. 그리고 $\alpha(t)$는 단조 감소함수이다. 이 알고리즘을 사용함으로서 두 벡터 사이의 Hyper plan을 거의 최적 상태까지 점차적으로 수렴시킬 수 있다.

4-3-2. TSRR

신경회로망 중 음성인식에서 가장 각광을 받는 신경회로망은 TDNN이다. 그러나 TDNN은 입력으로서 음성신호의 15 프레임을 사용하기 때문에 입력 프레임내에 복수의 음소가 존재할 가능성이 있다. 이때, 각 서브네트웍이 15 프레임을 동시에 스캔(scan)할 경우 각 서브네트웍에서는 자신이 학습한 음소를 찾아 출력할 것이다. 이와같은 문제점을 해결하기 위해 그룹식별 서브네트웍을 추가한 통합(glue) 네트웍을 구성하여 전체음소를 인식한다. 그러므로 글루네트웍은 그룹식별 모듈의 성능에 거의 의존하게 되어 연속어 인식에서 많은 오차를 발생시킨다. 이와같은 단점을 해결하기 위해 제안된 인식기가 TSRR 이다. TSRR의 구조는 다음과 같다.

입력층은 입력 유니트들과 문맥(context) 유니트들로 구성되어 있다. 인식할 음성은 각 시간 슬라이스별로 분리되고 각 슬라이스 단위로 TSRR에 입력된다. 입

◆TSRR의 구조



◆음소 /a/를 인식하는 예

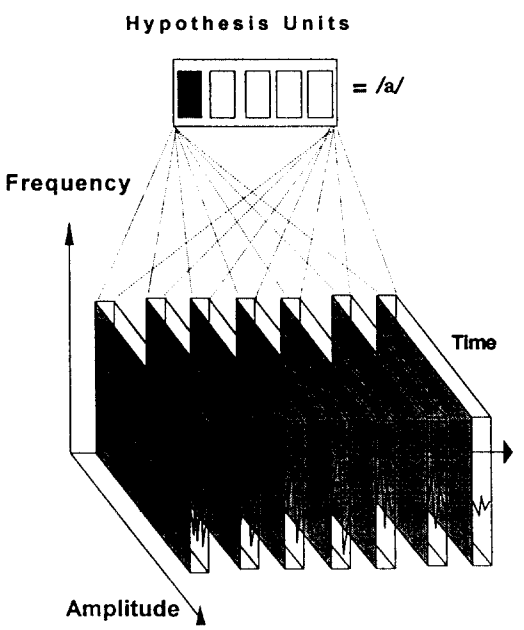


그림 17. (a) TSRR의 구조 (b) 음소 /a/를 인식하는 예
Fig 17. (a) Structure of TSRR (b) Example of recognition for phoneme /a/

력 유니트들로는 각 시간 슬라이스의 주파수 스펙트럼이 입력된다. 그리고 문맥 유니트들로는 전 슬라이스의 입력에 대한 은닉층의 출력을 복사해 현재 슬라이스와 함께 입력된다. 문맥 유니트의 입력은 현재 슬라이스를 인식하는데 전 슬라이스 정보를 사용하여 연속어 인식시 전후 관계정보를 이용할 수 있게 한다. 은닉층의 각 유니트는 입력 유니트들과 문맥 유니트들과 완전연결되어 정보전달을 한다. 출력층은 현재 입력된 시간 슬라이스가 어떤 음소인지를 출력하는 Hyphothesis 유니트들과 다음 슬라이스의 주파수 특성을 예측하도록 학습된 출력유니트로 이루어진다.

Ⅳ. 국내·외 음성인식 연구 동향

4-1. 국외 연구 동향

미국에서는 1971년에서 1976년까지 SUR(Speech Understanding Research)이라는 음성 이해 연구 프로젝트가 수행되었으며 최근에는 1984년부터 5년에서 10년 기간으로 음성 및 자연언어 처리에 관한 새로운 프로젝트가 수행되고 있다. 대용량 음성인식 시스템과 음성 언어 이해에 관한 연구를 주축으로 하여 특정 영역에서 자연스러운 음성을 실시간으로 인식하는 화자 독립 혹은 화자 적응 음성 인식 시스템을 개발하는 것을 목표로 한다. 프로젝트의 성공을 위해 매년 음성 및 자연언어 워크샵을 개최하여 연구에 참여하고 있는 연구원들이 최신 정보를 교류하고 앞으로의 연구방향을 모색하도록 하고 있다.

일본에서의 음성 인식 기술은 1982년부터 추진한 제5세대 컴퓨터 프로젝트의 일부인 "음성과 자연언어를 통한 컴퓨터 입출력"이라는 제목으로 연구가 진행되었으나 연구결과의 대외발표는 거의 없었다. 최근에의 음성인식 관련 프로젝트는 ATR(Advanced Telecommunication Reasearch institute) 산하 자동통역전화(automatic telephone interpretation) 프로젝트가 미국, 독일과 국제 공동 연구로 매우 발전적인 결과를 발표했다.

유럽에서는 범 유럽 국가들이 수행하는 ESPRIT(European Strategic Program for Research and developement in Inforamtion Technology)라는 정보 통신에 관련된 유럽국가들의 공동 프로그램이 있다. 한편 영국에서는 ITI(Information Technology Initiative) 프로젝트가 시작되어 음성 인식 및 데이타베이스 구축에 대한 연구가 진행되고 있다. 프랑스에서는 "Human machine communication"이라는 프로젝트에서 음성 통신, 자연어 처리에 관한 연구를 수행하고 있다. 독일에서는 SPICOS(Simens-Philips-IPO Continous Speech recognition)라는 대형 프로젝트에서 연속 음성 인식 기술에 관한 연구를 수행하였으며 최근 ASL(Architecture for Speech and Language research)라고 불리우는 새로운 프로젝트가 4년 계획으로 시작되어 음성 및 텍스트 데이타베이스 구성 및 대용량 음성인식 알고리즘 개발에 역점을 두고 있다.

4-2. 국내 음성인식 연구 동향

국내의 음성인식에 관한 연구는 1980년대 초부터 본격적으로 이루어졌으며 일부 대학이나 연구소에서 진행되던 연구도 이제는 많은 대학에 분포되어 진행되고 있다. 음성 인식 기술면에서도 초기의 DP 방법에서부터 VQ, HMM, 그리고 NN 방법을 이용한 인식실험 결과가 발표되고 있어 10여년의 역사로 볼 때 주목할만한 발전을 이룩하였다고 볼 수 있다. 또한 인식에 대상어휘가 소규모에서 중규모로 인식 대상도 특정화자에서 불특정 화자로, 단독어에서 연결어 또는 연속어등으로 확장되어 연구되면서 국외의 음성인식 기술 수준과 큰 차이없이 연구되고 있는 실정이다. 최근 3개년간 국내 학회지(한국 음향학회지 및 학술지, 대한 전자공학회지 및 학술지, 한국 통신학회지, 정보과학회지)에 게재된 음성인식 관련 논문들의 연구동향을 살펴보면, '80년대에는 VQ와 HMM, DP 방법을 이용한 음성인식 연구가 거의 비슷하게 진행되었다가 '90년도 들어와서 HMM방법을 이용한 연구가 많이 늘어났다. '91, '92년도에 이르러 DP와 VQ를 이용한 연구가 감소되면서 NN방법을 이용한 연구가 HMM을 이용한 연구와 동등하게 진행되어 국외의 연구 동향과 비슷한 경향을 보이고 있다. '94년을 전후로해서 HMM방식의 연구 및 HMM과 NN방식을 접목한 Hybrid 방식의 연구가 계속 나오고 있으며 HMM 방식에서도 파라메터의 개선에 관한 연구도 꾸준히 진행되고 있다.

Ⅴ. 음성인식의 기술 전망

앞으로 HMM의 성능을 향상시키는 방향의 연구가 계속될 것이다. 90년대에 들어 많은 HMM 알고리즘을 이용한 음성 인식 연구가 진행되었으며 또한 매우 좋은 결과를 얻었지만 아직도 보완되어야할 부분이 많다고 본다. 그러므로 우선 HMM 파라미터에 대한 개선이나 NN과 같이 사용하는 Hybrid방식이나 또는

HMM 알고리즘의 변형인 HMM-Net(Hidden Markov Model Network)등에 대한 연구가 깊이 진행될 것으로 본다.

다음으로는 음성처리(Spoken language processing) 연구이다. 음성 신호처리에서 얻은 지식뿐만 아니라 언어처리에서 얻은 지식을 효율적으로 결합시키는 방법이 병행하여 연구되어져야 할 것으로 본다. 그리고 회화체 음성에 관한 연구가 중심이 되어 Topic 위주의 시스템 개발에 더욱 박차를 가할 것으로 본다. 현재의 1000 단어 인식할 수 있는 시스템에서 수십만 단어를 인식할 수 있는 초 대용량 음성 인식 기술에 대한 연구가 계속 수행되어지며 고속 검색 알고리즘 및 유사단어 사이의 변별력 향상을 위한 알고리즘에 대한 연구도 함께 진행될 것이다. 마지막으로 잡음 환경에서의 인식을 위한 연구가 병행하여 진행될 것이다. 이러한 음성 인식 기술을 바탕으로 전화망을 통한 음성정보 검색 시스템이 실용화 되고 지능망에서 음성 인식 기술을 이용한 IP(Intellegent Peripheral) 시스템 개발이 많아질 것이다.

Ⅵ. 결 론

앞에서 언급했던 것처럼 선진국에서는 국가 주도하에 음성인식 분야에 대한 연구가 활발히 전개되고 있다. 미국, 일본, 그리고 유럽의 예를 보더라도 초 대용량 음성인식 시스템 개발, 다이얼링 서비스, 자동통역 전화 시스템, 다국적 언어인식이 가능한 실용적인 시스템 개발등에 중점을 두고 있다. 앞으로는 음성인식기술 분야는 응용 위주의 시스템 개발 즉, 실용화가 가능한 Topic 위주의 인식시스템 개발이 활발히 진행되어 일상생활에 많은 영향을 줄것으로 보며 단순히 음성 신호 처리의 측면을 벗어나 언어처리의 측면도 함께 고려한 회화체 중심의 인식시스템 개발에 연구가 활발히 전개될 것이다. 이를 위해서는 언어학자, 전자공학자, 심리학자, 컴퓨터 공학자등 공동 연구가 필수 불가결하다고 본다. 국내의 상황을 볼때 외국의 경우와 비교하여 기술력, 인력 및 연구비등 많은 부분이 부족한 실정이지만 관련업체 및 국가 기관으로부터의 보다 많은 관심과 투자가 병행되어 이 분야에 대한 연구를 지속적으로 전개해야 할 것으로 본다.

참 고 문 헌

1. D.S. Pallet, "DARPA resource management and ATIS bench mark test poster session," Proceedings of the DARPA speech and natural language workshop, pp 49-58, Feb., 1991.
2. C.Delogu et al, "New direction in the evaluation of voice input/output systems," IEEE Journal on selected Areas in comm., vol. 9, pp 566-573, May 1991.
3. K.F.Lee "Automatic speech recognition : the development of the SPHINX system," Kluwer Academic Publisher, 1989.
4. J.Takami and S. Sagayama, "A successive state splitting algorithm for efficient allophone modeling" proceedings of Int. Conf. on Acoustics Speech and Signal Processing, pp 573-576, Mar. 1992.
5. M. Ostendorf and S. Roukos, "A stochastic segment model for phoneme-based continous speech recognition" IEEE Trans. on Acoust., Speech and Signal Processing, vol 37, pp 1857-1869, Dec. 1989.
6. M. Lenning et al "Flexible vocabulary recognition of speech," Proceedings of an Int. Conf. on Spoken Lang. Processing, pp 93-96, Oct. 1992.
7. G.G. Matison, "Emerging voice services in the NyNEX network," Proceeding of voice systems worldwide 1992. pp 9-13. Feb. 1992.
8. S. Kuroiwa et al., "Architecture and algorithms of a real-time word recognizer for telephone input," Proceedings of an Int. Conf. on Spoken Lang. Processing, pp 1523-1526. Oct. 1992.
9. 구명완, "음성인식 기술의 현황과 전망," 정보 과학회지 제11권 5호 pp.21-34. 1993.
10. 한민수, 정유현, 이황섭, "음성처리 기술의 응용 현황및 전망" 전자 공학회지 제20권 5호 pp.542-547. 1993.
11. G. Elius et al. "Bellcore effects in applying speech technology to telephone network service," Int. Conf. on Speech lang. Processing., kobe, pp.20.21-20.24, 1991.
12. 김순협 "음성 인식 기술 현황및 실용화 전망," 한국음향학회지 pp.86-95 vol 13. No.2 1994.

김 순 협

- 1947년 12월 28일생
- 1974년 2월 : 울산공과대학 전기공학과(전자공학 전공)졸업
- 1976년 2월 : 연세대학교 대학원 전자공학과 석사 과정 졸업(공학석사)
- 1983년 2월 : 연세대학교 대학원 전자공학과 박사 과정 졸업(공학박사)
- 1979년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 교수
- 1986년 8월 ~ 1987년 7월 : Univ. of Texas at Austin 객원교수
- 1992년 1월 ~ 현재 : 한국음향학회 부회장
- 관심분야 : 음성인식, 신호처리, 신경회로망