신경 회로망을 이용한 음성 신호의 벡터 양자화

백 승 복* , 김 상 희**
*금오공과 대학교 전자공학부 대학원
**금오공과 대학교 전자공학부 부교수

Speech Signal Vector Quantization Using Neural Network

Seong-Bok Baek and Sang-Hee Kim
School of Electronics Engineering, Kumoh Nat. Univ. of Tech., Korea
E-mail: bsb@knut.kumoh.ac.kr

Abstract

This paper describes a vector quantization for speech signal coding using neural networks.
We processed speech signal using LPC method that extracts speech signal feature, and speech signal feature is quantized using competitive neural network kohonen self-organization feature map.

I. 서론

최근 음성인식이나 분석에 있어서 각광을 받고 있는 음성파라미터 추출방법으로 LPC를 이용하고 있다. 선형 예측 분석법(linear prediction analysis)이라는 용어가 N.Wiener에 의해서 처음 소개된 이후, 선형 예측 분석법은 여러 분야에서 널리 쓰이고 있다. 선형 예측 분석법은 음성 분석 및 합성에서 Itakkura와 Saito에 의해서 사용되었으며 Atal과 Schroeder에 의해서 음성 압축에 사용되는 등 음성 처리의 여러 분야에서 광범위하게 사용되는 기법이다. 선형 예측 분석을 사용하면 음성 신호, 혹은 음성 스펙트럼이 가진 특성을 상대적으로 적은 수의 파라미터만으로 정확하게 표현할 수 있다는 장점이 있다. 또한, 선형 예측 분석 자체가 그리 많은 연산량을 요구하지 않는 장점도 갖는다.

벡터 양자화는 음성 인식, 음성 압축, 화상 처리 등에서 널리 쓰이며, 입력값의 차원이 너무 크거나 그 값의 범위가 매우 큰 경우, 대표 패턴이 저장된 코드북으로부터 이에 대응되는 양자화값으로 차원 수를 줄이거나 범위를 줄이는 방법이다. 일반적인 벡타 양자화 방법인 K-means 알고리즘과 Linde-Buzo-Gray(LBG)알고리즘 여러개의 확률 분포로서 벡터를 양자화하는 혼합분포를 이용한 양자화 방법이 있다. 최근의 연구로는 신경망을 이용하여 양자화 하는 방법도 제시 되고 있다. 본 논문에서는 코호넨 신경망을 이용하여 벡터 양자화를 수행했다.

스스로 학습을 할 수 있는 능력을 이용한 신경망 모델이 튜보 코호넨(Teuvo Kohonen)에 의해서 제안 되었고, 그가 제안한 신경망은 자기 조직화(self-organizing)의 특성을 이용하여 스스로 학습할 수 있도록 하였다.[8]

Kohonen이 제안한 SOFM(Self Organizing Featured Maps)[3]의 특성은 무감독 학습으로서 양자화를 이용한 패턴 인식 분야에서 널리 쓰여진다. Competitive Net은 Kohonen learning 알고리즘을 따르고 무감독 학습으로서 학습이 진행되어진다.

본 논문에서는 신경망을 이용하여 벡터 양자화기를 구현하였다. 음성신호의 특징 벡타 추출후, 코넨이 제시한 SOFM 특성을 가지는 신경망을 이용하여 양자화기를 학습하고, 구현된 양자화의 성능을 테스트 하였다.

II. LPC

음성의 특징 추출은 음성 생성 원리를 이용한 선형 예측방법을 이용한다. 음성신호는 5에서 10ms 사이의 충분히 짧은 시간 주기에 대해 조사하면, 그 특성이 상당히 정지적(stationary)이라는 것과, 음성신호는 느리게 변하는 시변 신호라는 것을 알 수 있다. 따라서, 음성 신호는 서로 상관적으로 변화하고, 약 20ms 내에서는 선형적으로 변하는 특성이 있으므로, 선형 시스템으로 가정할 수 있다.

p개의 과거의 값 $x_{n-1}, x_{n-2}, \cdots, x_{n-p}$이 주어지면 현재의 예측신호는 다음과 같이 주어진다.

$$\hat{x}_n = a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_m x_{n-m} = \sum_{k=1}^{p} a_k \cdot x_{n-k} \quad (1)$$

음성의 선형예측 모델은 그림[2-1]과 같다.

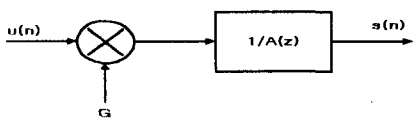


Fig 2-1. Linear prediction model of speech

전극모델(all-pole model)에 대한 전달함수는 다음식과 같다

$$H(Z) = \frac{1}{A(Z)} = \frac{G}{1+\sum_{k=1}^{p} a_k Z^{-k}} \quad (2)$$

여기서 a_k는 선형예측계수(Linear predictive coefficient)라 한다. 원 신호와 예측신호와의 차를 예측오차(predictive error)라 하고 다음과 같이 표현되어진다.

$$e(n) = x(n) - \hat{x}(n) = x(n) + \sum_{k=1}^{p} a_k\, x(n-k) \quad (3)$$

여기서 예측계수 a_k는 다음과 같은 자승오차(square-error)를 최소화 하는 방향으로 결정되어진다.

$$E = \sum_n e^2(n) = \sum \left\{ x(n) + \sum_{k=1}^{p} a_k\, x(n-k) \right\}^2 \quad (4)$$

최소 오차를 구하기 위해서 E를 예측계수에 대하여 편미분 하면

$$\frac{\partial E}{\partial a_i} = \sum_{k=1}^{p} a_k \sum_n x(n-k)x(n-i) + \sum_n x(n)x(n-i) = 0 \quad (5)$$

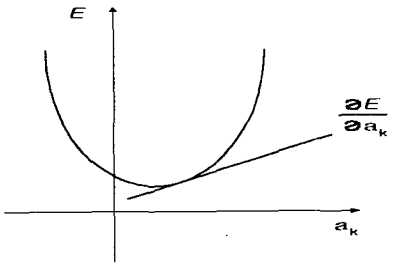


Fig 2-2. Error function according to prediction coefficient

이고, 윗 식들을 정리해서 다시 쓰게 되면 다음과 같다

$$E_p = \sum_n x^2(n) + \sum_{k=1}^{p} a_k \sum_n x(n)x(n-k) \quad (6)$$

또한 $e(n) = Gu(n)$의 관계를 이용하여 쓰면 다음식을 얻게 되며,

$$E_p = \sum_n e^2(n) = G^2 \sum_n u^2(n) \quad (7)$$

결과적으로 $E_p = G^2$가 된다. 위와 같은 방법으로 이득(G)과 예측오차(a_k)가 구해진다. 예측계수를 구하기 위해서 두 가지 방법 중 자기상관계수법을 이용하였다.

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i) \qquad (1 \le i \le p),$$

$$R(i) = R(-i) = \sum_{n=-\infty}^{\infty} x(n)x(n+i) \quad (8)$$

주어진 신호에 대한 자기상관계수는 다음과 같다.

$$r_{ij} = \sum_{n=0}^{N-1-(i-j)} x_n x_{n+(i-j)} \quad (9)$$

이것을 자기상관 함수 형태로 다시 표현하면 다음과 같다.

$$r_{ij} = r_{i-j}$$
$$r_k = \sum_{n=0}^{N-1-k} x_n x_{n+k} \quad (10)$$

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdots \\ r_p \end{bmatrix} = \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & r_0 & r_1 & \cdots & r_{p-2} \\ r_2 & r_1 & r_0 & \cdots & r_{p-3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdots \\ a_p \end{bmatrix} \quad (11)$$

이 행렬의 효율적인 풀이 방법에는 Durbin's algorithm을 이용한다.

III. Vector Quantization

일반적으로 벡타 양자화란 N의 크기를 갖는 k 차원의 유클리드 공간 R^k 에서 같은 크기의 제한된 집합 C로의 멥(map)을 말한다.[7]

$$Q: R^k \rightarrow C \quad (12)$$

여기서 $C = \{y_0, y_1, \ldots, y_{N-1}\}$이고, N개의 원소로 구성된 코드북(codebook)을 나타내며, $y_i \in R^k$ 이다.

벡터 양자화에서 가장 중요시되는 코드북의 설계를 위해서 코호넨이 제안한 *SOFM*(Self Organizing Featured Maps)의 특성을 가지는 Network를 적용하였다.

코호넨의 네트워크는 먼저 각 뉴런은 연결강도 벡터(Z)와 입력벡터(X)가 얼마나 가까운지 계산한다. n차원의 입력벡터 X가 주어졌을 때 Z(network parameter)와의 유클리디안 거리 계산은 다음과 같다.

$$X = (x_1, x_2, \ldots, x_n)$$
$$D = \|X - Z\| = \left[\sum_{i=1}^{n} (x_i - z_i)^2 \right]^{\frac{1}{2}} \quad (13)$$

유클리디안 거리 계산 후 D의 값이 가장 최소인 뉴런이 승자뉴런으로 결정되고 이후 학습에 있어서 기준뉴런으로 결정된다.

네트워크의 학습규칙(Learning Rule)은 먼저 승자 뉴런을 결정하고, 그 후에는 학습 규칙에 따라 뉴런의 연결강도를 재 조정한다. 네트워크의 학습 규칙은 단순히 연결강도 벡터와 입력벡터의 차이(D)를 구한 다음 그것의 일정한 비율을 원래의 연결강도 벡터(Z)에 더하는 것으로 학습이 진행된다.

$$X \in C_j \quad then \quad Z(t+1) = Z(t) + \alpha(t) \cdot [X - Z(t)]$$
$$X \notin C_j \quad then \quad Z(t+1) = Z(t) \qquad (14)$$

여기서 $\alpha(t)$는 네트워크의 학습률이고 시간에 따라 감소하게 된다.[6]

승자 뉴런만이 그것과 관련된 연결강도 벡터를 조정하는 것이 아니라 그의 이웃 반경안에 드는 모든 뉴런들도 유사한 조정을 하게 된다. 학습의 처음 단계에서는 층내의 모든 뉴런들이 포함될 수 있으며, 훈련이 진행됨에 따라서 이웃 반경은 서서히 줄어듦으로서 점점 적은 개수의 뉴런들이 학습을 하게 된다. 최종적으로는 승자 뉴런만이 혼자 그것의 연결강도를 조정하게 된다.

최종적으로 학습이 끝난후의 각 뉴런의 연결강도 벡터(Z)는 기하학적으로 입력패턴들과 유사성을 가지게 된다.

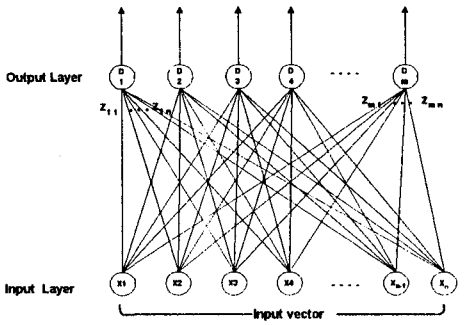


그림 3-1. LVQ competitive network structure

SOFM 구조는 기존의 신경망과는 달리 계층적(Hierahical)이지 않고 2개의 층으로 이루어져 있다. 이 네트워크의 첫번째 층은 입력층(input layer)이고 두 번째 층은 경쟁층(competitive layer)로 구성되어져 있며, 모든 연결들은 첫 번째 층에서 두 번째 층의 방향으로 연결되어 있고 그 연결은 Fully-connected 되어 있는 구조이다. 이를 그림[3-1]에 나타냈다. 그림[3-2]는 벡터 양자화기 생성과 그후 양자화기의 성능테스트 과정을 도시한 블록 다이어그램이다.

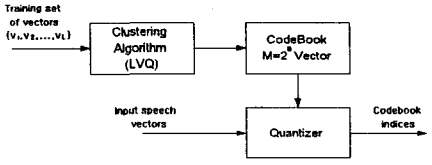


그림 3-2. Block diagram of the basic VQ training and classification structure.

IV. 시뮬레이션 결과 및 고찰

먼저 네트워크의 학습을 위한 데이터로는(training data)는 한국여성이 발음한 음성신호를 사용하였다. 이 음성은 11.025kHz로 샘플링 됐으며 8bit의 샘플비트를 갖는 데이타이다. 총 샘플된 개수는 229,408개이며 전체 음성의 길이는 약 22.80sec이다. 이 음성을 이용하여 만든 프레임은 각 프레임당 256개의 샘플수를 갖고 overlap이 64 point인 프레임을 사용하였다. LPC 차수는 10차이고, 정규화 시킨 LPC 신호를 네크워크의 입력으로 사용하였다. 네트워크의 입력벡터 차원은 128이며, 전체 프레임 수는 1194개이다. 네크워크의 Cluster 개수는 64(2^6)로 하여 codebook을 완성하였다.

이후 동일 여성이 발성한 음성신호를 이용하여 구성된 양자화기의 특성을 검토하였다. 이 테스트 데이터로는 700개의 프레임을 사용하였으며, 총 샘플수는 178,927개 전체 음성의 길이는 약17.78sec 분량의 길이이다. 양자화기의 학습률은 다음과 같이 조정하였다.

$$\alpha = e^{-\frac{1}{100}n} \qquad (15)$$

여기서 α는 학습률이고, n은 네트워크의 학습회수를 나타낸다. 본 모의 실험에서는 최대300회까지의 학습을 허용하였으며, 적정 허용왜곡까지의 학습이 진행되면 더 이상의 진행없이 바로 codebook을 생성하도록 구성하였다. 그림[4-1]은 코드북을 생성하기 위한 네크워크 전체 흐름도를 도시하였다.

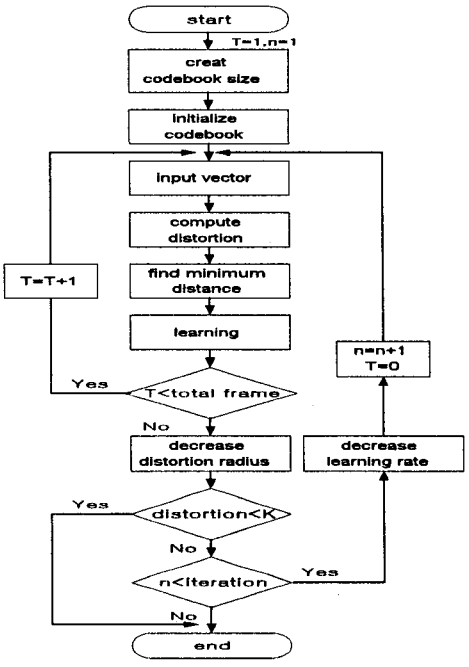
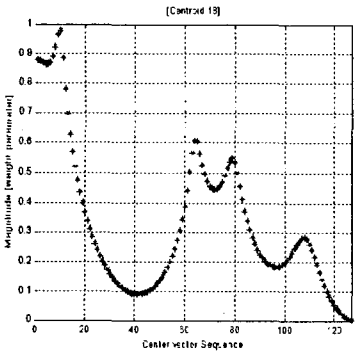


그림 4-1. Flow diagram of codebook generation algorithm

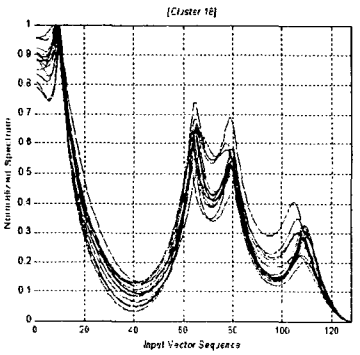
모의 실험결과에 따른 64개의 클러스터중 18번째 클러스터의 결과를 그림[4-2]에 도시하였다.

본 실험결과에서 양자화에 따른 distortion은 training data기준으로 testing data가 약 15.74%의 증가가 있었고, testing data 에서 입력벡터와 codebook사이에 각 원

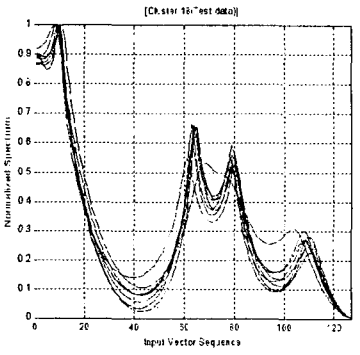
소간의 평균 거리는 약 0.09로 얻어졌다. 동일한 작업환경에서 동일 인물이 발성한 training data와 testing data를 이용한 관계로 비교적 좋은 결과를 얻었다.



(a)



(b)



(c)

Fig 4-2. (a)center vector (b)cluster(training data) (c) cluster(testing data)

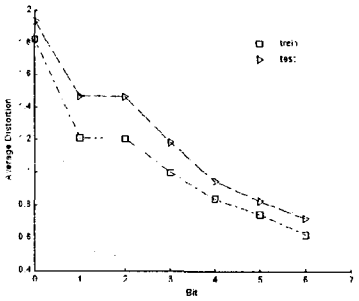


Fig 4-3. Overall average distortion *according to increase bit*

<표 1> Train data 와 Test data의 양자화 왜곡

	전체 프레임 개수	전체 평균왜곡
TRAIN	1194	0.624254
TEST	700	0.722525

V. 결론

본 논문에서는 음성신호의 특징 추출 후 얻어진 특징 벡터로 신경망을 이용한 벡터 양자화를 수행하였다. 제안된 코호넨 신경망의 무감독 학습방법으로서 벡터 양자화기의 구현을 하였고, 이 양자화기의 성능을 테스트 하였다.

비교적 좋은 성능을 가지는 양자화기를 구현이 가능하였지만, 많은 양의 training data를 이용하여 양자화기 학습시간의 증가가 있었다. 보다 향상된 양자화기를 구현하기 위해서는 여러 종류의 음성신호로 양자화기의 학습이 필요하며, 학습시간을 줄이기 위해서는 입력벡터의 차원을 제한하는 방법등의 개선이 필요하다.

참고문헌

[1] Rabiner, L.R., Juang, B-H, "Fundamentals of Speech Recognitiion," Prentice Hall, 1993.
[2] Sadaoki Furui, Digital Speech Processing, Synthesis, and Recognition, Marcel Dekker, Inc, pp. 225-289, 1991.
[3] T. Kohonen, Self-Organization and Associative Memory, 3rd ed. Berlin: Springer-Verlag, 1989.
[4] T. Kohonen, "The self-organizing map", Proc. IEEE, vol. 78, no. 9, pp.1464-1480, 1990.
[5] R. Pal, J. C. Bezdek, and E. C.-K Tsao, "Generalized clustering networks and Kohonen's self-organizing scheme", IEEE Trans. Neural Networks, vol. 4, pp. 549-557, 1993.
[6] Kohonen, T.: "The Neural Phonetic Typewriter," IEEE Computer, 21, pp. 11-22 1988.
[7] R. M. Cray, "Vector quantization," IEEE Acoust., Speech, Signal Processing Mag., pp. 4-29, Apr. 1984.
[8] S. Nakagawa, K. Shikano, Y.Tohkura, "Speech, Hearing and Neural Network Modls", IOS press, 1995