DBpia

論文 95-3-3-21

시공간 신경회로망을 이용한 한국어 숫자음 인식

學生會員 李 鍾 植*, 正會員 鄭 在 皓*

Korean Digit Recognition Using Spatio-Temporal Neural Network

Jong Sik Lee* Student Member, Jae Ho Chung* Regular Members

본 연구는 94년도 인하대학교 연구비 지원에 의하여 수행 되었음

要 約

본 논문에서는, 시공간 신경회로망을 이용하여 한국어 숫자음의 인식을 시도하였다. 시공간 신경회로망을 이용하여 한국어 숫자음 인식을 시도한 기존의 논문에서는, 음성신호의 특징을 나타내는 특징벡터로서 LPC (Linear Predictive Coding) 계수를 입력패턴으로 사용하였다. 이에 반하여, 본 논문에서는, LPC-cepstrum 계수를 입력패턴으로 사용하였으며, LPC 계수를 입력패턴으로 사용한 경우와의 인식률을 비교하였다. 입력패턴을 LPC 계수로 선택하였을때는 83.5%의 인식률을, LPC-cepstrum을 계수로 사용하였을 때는 90.0%의 인식률을 얻었다. 또한, 본 논문에서는 LPC-cepstrum 계수를 입력패턴으로 선택한 시공간 신경회로망에서, 최종 판별이 어려운 두 경쟁 단어 사이의 구별을 위하여, 두 경쟁 단어를 나타내는 각각의 LPC-cepstrum들과 입력패턴을 나타내는 LPC-cepstrum 사이의 Euclidean cepstral distance를 구하고, 이들을 비교하여 최종인식 판별을 하였다. 이와같은 아주 간단한 Euclidean cepstral distance measuring을 시도하였을 경우, 시공간 신경회로망의 성능이 90.0%에서 95.0%로 크게 향상됨을 보였다.

ABSTRACT

In this paper, a new approach for Korean digit recognition using the Spatio-Temporal Neural Network (STNN) is reported. Three different approaches are analyzed, and the digit recognition rate of 95% is achieved. In the first approach, the Linear Predictive Coding (LPC) coefficients extracted from the vocal tract analysis are used as STNN's input pattern vector. In the second approach, the LPC-cepstrums derived from the LPC coefficients are used as the input pattern vector. For the first and the sec-

*인하대학교 전자공학과 디지탈 신호처리 연구실
Department of Electronics Engineering, Digital Signal Processing Laboratory, Inha University
論文番號 : 94366
接受日字 : 1994年 12月 23日

www.dbpia.co.kr

ond systems, we got the digit recognition rates of 83.5% and 90.0%, respectively. Using the LPC-cepstrums as the input pattern vector, in the third system, when the difference between the highest two scores of STNN's output neurons is less than the predefined threshold value, the distortions of two digit candidates from the input vector are computed and compared using the Euclidean cepstral distance measure. This simple added feature improves the performance of the system dramatically from 90.0% to 95.0%.

I. 서 론

신경회로망이 대량의 복잡한 데이타를 병렬처리할 수 있을 뿐만 아니라 학습능력이 있다는 사실에 근거를 두고, 신경망을 이용한 음성인식에 대한 연구가 활발히 진행되고 있다[1,2,3,4,5].

그러나 기존의 신경망들, 특히 일반화된 MLP (Multi-Layer Perceptron)나 CPN (Counter Propagation Network)은 정적 패턴의 인식에는 우수한 성능을 보이지만, 음성신호와 같이 시간에 따라 그 특성이 변하는 동적 패턴의 인식에는 취약점이 있는 것으로 알려져 있다. 이와같은 음성신호의 특성, 즉 시간에 따라 순차적으로 변화하는 신호의 동적 특성을 효과적으로 인식할 수 있는 신경회로망으로서, 시공간 신경회로망 즉 STNN (Spatio-Temporal Neural Network)이 1980년대 중반 Kosko와 Klopf에 의하여 제안되었다[6,7,8].

음성신호는 시간에 따라 특성이 변화하는 현상이외에, 같은 화자에 의하여 발음된 같은 음성신호라 하더라도 발음속도가 때에 따라 일정하지 않다. 또한 문맥의 전후관계에 의하여 같은 음성이 조금씩 다르게 발음된다. STNN은 입력되어 들어오는 신호의 시퀀스 패턴들 상호간의 연관성을 고려하면서 인식을 하기 때문에, 위에 언급한 음성신호의 부분적인 시간 변화와 주파수 변화의 영향을 인식과정에서 감소시킬 수 있는 특징을 갖고 있다. 또한, STNN은 인식과정에서 화자의 발성 길이의 20% 정도의 증감은 인식결과에 크게 영향을 미치지 않는다는 큰 장점을 갖고 있다. 본 연구에서는 STNN을 사용하여 한국어 숫자음의 인식을 시도하였다.

STNN을 이용하여 한국어 숫자음 인식을 시도한 기존의 논문에서는, 음성신호의 특징을 나타내는 특징벡터로서 LPC (Linear Predictive Coding) 계수를 입력패턴으로 사용하였다[7]. 이에 반하여, 본 논문에서는, LPC-ceptsrum 계수를 STNN의 입력패턴으로 선택하였으며, LPC 계수를 입력패턴으로 선택하였을 경우와의 인식률을 비교하였다. 입력패턴을 LPC 계수로 선택하였을때는 82.5%의 인식률을, 입력패턴을 LPC-cepstrum 계수로 하였을 경우에는 90.0%의 인식률을 얻었다. 따라서, LPC-cepstrum 계수를 사용하였을 경우에 더 좋은 인식률을 얻을 수 있음을 보였다.

또한, 본 논문에서는 LPC-cepstrum 계수를 입력패턴으로 선택한 STNN에서, 최종 판별이 어려운 두 경쟁 단어 사이의 구별을 위하여, 두 경쟁 단어를 나타내는 각각의 LPC-cepstrum들과 입력패턴을 나타내는 LPC-cepstrum 사이의 Euclidean cepstral distance를 구하고, 이들을 비교하여 최종 인식 판별을 하였다. 즉, 입력신호를 나타내는 LPC-cepstrum의 값들을 0부터 9까지의 해당 숫자음에 맞게 각각 디자인한 10개의 STNN에 통과시킨 후, 각 STNN의 최종 출력 neuron들의 값을 비교하여 상위 두 출력값의 차이가 극히 적어서 두 경쟁 단어 사이의 최종 인식 결정이 곤란할 경우, 상위 두 경쟁 단어를 나타내는 각각의 LPC-cepstrum 값들과 입력신호의 LPC-cepstrum 사이의 Euclidean cepstral distance를 구하여 비교한후, 작은 distance 값을 배출한 경쟁단어를 최종 승리자로 결정한다. Euclidean distance measure를 사용하여 두 cepstrum 벡터들 사이의 차 (즉, 각 성분의 차의 제곱의 합)를 구할 경우, 이는 주파수 영역에서 logarithm을 취한 각각 (즉, 입력패턴과 templete 패턴)의 스팩트럼들 사이의 평균-제곱의 차이 (mean-squared difference)라는 직접적인 의미를 갖는다[9,10]. 본 논문에서 행한 실험에서는, Euclidean cepstral distance measuring을 시도하였을 경우, STNN의 성능이 90.0%의 인식률에서 95.0%의 인식률로 향상됨을 보였다.

2절에서는 STNN의 구조와 동작원리에 대하여 간단

www.dbpia.co.kr

하게 언급 하였으며, 3절에서는 본 논문에서 숫자음 인식을 위하여 입력패턴으로 사용한 LPC-cepstrum에 대하여 설명하였다. 4절에서는 본 논문에서 행한 실험들과 그 결과들에 대하여 설명하였고, 마지막으로 5절에서는 결론을 내렸다.

Ⅱ. STNN(시공간 신경 회로망)

2.1 구조와 동작원리

STNN은 여러 개의 층으로 구성되어 있으며, 각 층은 서로 다른 단어에 대한 weight (가중치 또는 특징값)의 connection line들로 연결된 neuron들로 구성되어 있다. 각 층의 구조는 동일하다. 각 층의 neuron의 수는 입력 신호 전체의 길이를 선형적으로 나눈 구간의 갯수와 같다. 한 단어가 입력으로 들어오면, 시간적인 순서에 따라 첫번째 구간의 LPC-cepstrum 계수들이 층 전체의 neuron들을 활성화시키고, 출력값을 낸다. 시간이 지남에 따라 두번째 구간의 입력신호가 들어오면, 두번째 구간의 LPC-cepstrum 계수들을 계산하고, 다시 층 전체의 neuron들을 활성화하여 출력값을 계산한다. 이와같은 과정을 마지막 입력신호의 구간까지 반복 수행하여, 최종 출력값을 구한다. 각 층의 동작 원리가 그림 1에 설명되어져 있으며, 각 층을 구성하고 있는 i번째 neuron의 입력값을 수식적으로 나타내면 다음과 같다.

$$I_i = \overline{Q_j} \cdot \overline{W_i} + d\sum_{k=1}^{i-1} x_k \qquad (1)$$

식 (1)에서, I_i는 i번째 neuron의 입력값을, $\overline{Q_j}$는 j번째 입력구간의 LPC-cepstrum 벡터를, $\overline{W_i}$는 i번째 neuron의 weighting 벡터를, x_k는 k번째 neuron의 출력값을 나타낸다. 또한, 식 (1)에서 $d(0<d<1)$는 i번째 neuron에 전달되는 i번째 이전 neuron들의 출력량의 크기를 조절하는 상수이다.

한편, i번째 neuron의 출력값 x_i는 다음과 같은 미분방정식을 통하여 나타내어 진다.

$$\dot{x}_i = A(-ax_i + b[I_i - \Gamma]^+) \qquad (2)$$

식 (2)에서, a와 b는 양의 상수값이다. 식(2)에 포함되어 있는 함수 $[I_i-\Gamma]^+$는 다음과 같이 정의된다. 즉

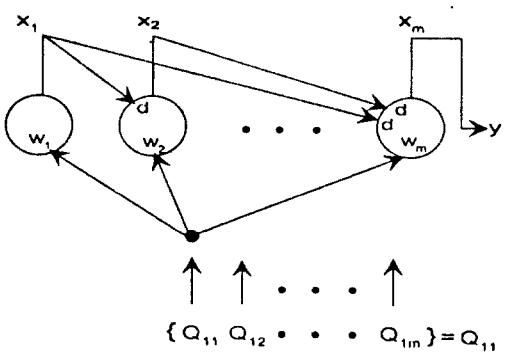

그림 1. STNN을 구성하는 각 층의 구조.

$$[I_i - \Gamma]^+ = \begin{cases} I_i - \Gamma, & \text{if } I_i - \Gamma > 0 \\ 0, & \text{if } I_i - \Gamma \le 0 \end{cases} \qquad (3)$$

따라서, 식 (3)에서 Γ는 임계값(threshold)의 역할을 한다. 식 (2)에서 함수 A(•)는 attack function이라 부르며, 다음과 같이 정의 된다.

$$A(u) = \begin{cases} u, & \text{if } u > 0 \\ cu, & \text{if } u \le 0 \end{cases} \qquad (4)$$

Attack function은 식 (2)에 소개한 i번째 neuron의 출력값이, 균형상태(equilibrium state)에 이르기까지의 상승시간(rising time)과 균형상태 후의 하강시간 (falling time)의 길이를 조정하는 효과를 갖는다. 특히, 파라메타 $c(0<c<1)$는 이전 neuron들의 weight 벡터들이 대응하는 입력구간들의 입력 벡터들과 일치되었을 경우, 이러한 기억들이 현재 neuron의 출력값에 영향을 미치게 하는 특성을 지닌다. 따라서, 입력신호의 일부에 noise가 섞이거나, 발음이 다소 변하였더라도 이를 극복할 수 있게 된다.

그림 2에 attack function의 시간에 따른 변화, 즉 neuron의 시간에 따른 출력값이 설명되어져 있다. 균형상태에 이르기까지의 과정을 살펴보면, 우선 식 (3)과 식 (4)에서 조건들이 식 (5)와 같을때, 즉,

$$\begin{cases} [I_i - \Gamma] > 0, \\ \quad \& \\ (-ax_i + b[I_i - \Gamma]) > 0, \end{cases} \qquad (5)$$

www.dbpia.co.kr

이와같은 경우, 식 (2)로부터 neuron의 출력값은 식 (6)과 같은 시간함수로 표현된다.

$$x_i(t) = \frac{b}{a}[I_i - \Gamma](1-e^{-at}) \quad (6)$$

식 (6)은, 그림 2에서 보여 주듯이, neuron의 출력값이 $\frac{1}{a}$의 기울기를 갖고 상승하는 것을 말해준다. 반면에, 균형상태를 거친후에는 식 (3)과 식 (4)가 식 (7)과 같은 조건을 만족한다. 즉

$$\begin{cases} [I_i - \Gamma] \le 0, \\ \quad \& \\ (-ax_i + b[I_i - \Gamma]) \le 0, \end{cases} \quad (7)$$

따라서, 균형상태를 거친후의 neuron의 출력은 식 (2)로부터, 식 (8)과 같은 시간함수로 표현된다. 즉

$$x_i(t) = x_{i_{max}} e^{-act} \quad (8)$$

따라서 식 (8)은, 그림 2에서 보여 주듯이, neuron의 출력값이 $\frac{1}{ac}$의 기울기를 갖고 감소함을 나타내고 있다. 식 (6)과 식 (8)로 부터 각 neuron의 출력값이 균형상태에 이르기까지의 상승 과정에서는 빠르게, 그리고 균형상태 후의 하강 과정에서는 천천히 하강함을 볼 수 있다. 따라서, 한 구간의 시간적 패턴이 일치하였을 경우, 이에 대응하는 출력은 짧은 시간내에 빨리 증가시키며, 다음 구간의 시간적 패턴이 비록 불일치하더라도, neuron의 출력값이 급격히 감소하지 않도록 한다. 이는 앞에서 언급하였듯이, 일부의 noise나 다소의 발음 변화가 있더라도 전반적으로 일치하면, 이를 인식하는 효과를 갖고 있음을 보여 준다.

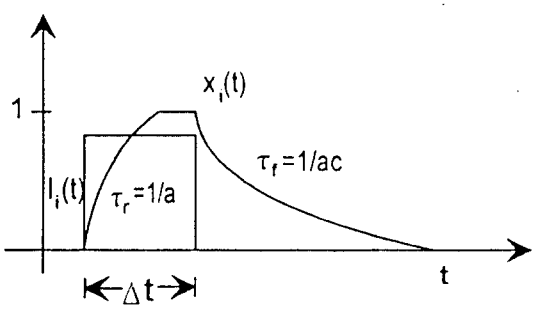

그림 2. Neuron의 시간에 따른 출력값의 변화.

그림 3은 시간이 진행하면서 과거, 현재 neuron 출력값들의 크기 변화를 보여주고 있다. 시간이 완료되었을 때에, 마지막 neuron의 최종 출력값이 입력신호와 입력신호를 적용한 STNN 사이의 닮은 정도를 나타낸다. 따라서, 입력된 숫자음을 인식하기 위하여서는, 입력신호의 LPC-cepstrum을 10개의 서로 다른 STNN들, 즉 0부터 9까지 각각의 숫자음에 해당하는 STNN에 각각 입력신호로 적용하고, 10개의 최종 출력값을 얻은 후, 이들을 비교하여 최종 winner를 결정한다.

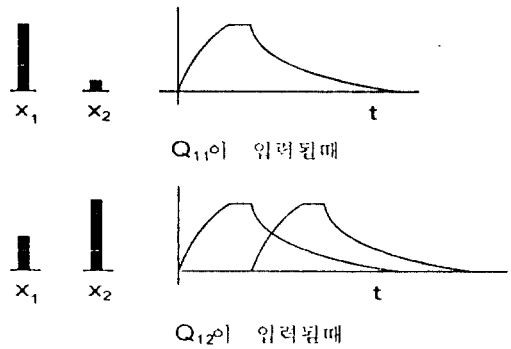

그림 3. 시간의 흐름에 따른 neuron 출력값들의 변화과정.

2.2 STNN (시공간 신경 회로망)의 학습 방법

STNN의 입력 패턴은 각 구간 단위로 크기가 1인 단위 벡터로 정규화 한다. 또한, weight 벡터도 크기가 1인 단위 벡터로 한다. 따라서, 학습과정에서의 각 neuron의 출력값은 식 (9)와 같이 표현될 수 있다.

$$\overline{Q} \cdot \overline{W} = |\overline{Q}|\,|\overline{W}| \cos\theta = \cos\theta \quad (9)$$

식 (9)에서 $\overline{Q}$는 입력 벡터이고, $\overline{W}$는 weight 벡터이다. 이와같은 경우, 학습의 목적은 weight 벡터를 원하는 입력 벡터와 일치시키는 것이므로, 학습과정에서 요구하는 weight의 변화량 $\Delta\overline{W}$는 다음과 같은 과정을 통하여 얻는다.

$$\Delta\overline{W} = \alpha(\overline{Q} - \overline{W}) \quad (10)$$

Weight 벡터의 변화량 $\Delta\overline{W}$이 결정되면, weight 벡터의 값을 변경시킨다. 즉

www.dbpia.co.kr

$$\overline{W}(t+1) = \overline{W}(t) + \alpha\ \Delta\overline{W}(t)\ , \quad 0<\alpha<1 \tag{11}$$

STNN의 학습과정에 있어서, 적절한 학습 횟수는 실험적으로 결정한다. 학습과정이 완료되었을 경우, 각각의 weight 벡터들의 위치는 학습에 사용된 입력신호들의 평균값을 나타내는 위치에 도달한다. 학습 과정중, weight 벡터의 변경 과정이 그림 4에 나타나 있다.

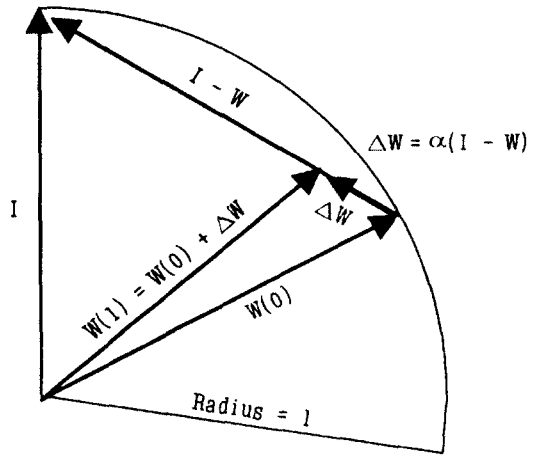

그림 4. STNN에서 weight 벡터의 변경.

Ⅲ. LPC-cepstrum

본 논문에서는 STNN의 입력패턴으로서 LPC-cepstrum 계수를 사용하였다. LPC-cepstrum 계수 c[n]은 LPC 계수 a [n]으로부터 다음과 같은 순환 (recursive) 방법을 통하여 얻어진다[11].

$$c[1] = -a[1], \tag{12}$$

$$c[n] = -a[n] - \sum_{k=1}^{n-1}\left(1-\frac{k}{n}\right) a[k]\ c[n-k], \quad 1< n \le p\ . \tag{13}$$

식 (13)에서 p는 LPC 계수의 차수를 나타내며, 본 논문에서는 16차를 사용하였다.

두 LPC-cepstrum 벡터들 사이의 차를 Euclidean cepstral distance measure를 사용하여 구할경우, 구한 차는 주파수 영역에서 logarithm을 취한 각각의 스펙트럼들 사이의 평균-제곱의 차이 (mean-squared difference)라는 매우 바람직한 해석을 갖는다 [9,10]. 먼저, 두 cepstrum 벡터 사이의 Euclidean cepatral distance measure는 다음과 같이 정의 된다.

$$d_{ECEP}(c,\tilde{c}) = \sum_i |c[i]-\tilde{c}[i]|^2 \tag{14}$$

Parseval의 정리를 사용하여, d_{ECEP}는 주파수 영역에서, 다음과 같이 해석 되어진다.

$$d_{ECEP}(c,\tilde{c}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\log S_c(\omega) - \log S_{\tilde{c}}(\omega)|^2 d\omega \tag{15}$$

식 (15)에서,

$\log S_c(\omega)$와 $\log S_{\tilde{c}}(\omega)$는 각각 LPC-cepstrum 벡터들 c와 $\tilde{c}$의 log를 취한 스펙트럼을 나타낸다. 따라서, Euclidean cepstral distance measure는 log를 취한 두 스펙트럼 사이의 차를 구하는 매우 효과적인 measure이다. 본 논문에서는 d_{ECEP}의 이와같은 성질을 이용하여 STNN의 숫자음 인식율을 더욱 효과적으로 향상시킬 수 있었다. 자세한 내용은 다음 절에서 언급하기로 한다.

Ⅳ. 실험 구성 및 결과

4.1 실험구성

본 연구에서 구성한 음성 인식 시스템은 그림 5와 같다. 마이크를 통해 받아 들여진 음성은 12bit 양자화 레벨을 갖는 A/D converter를 통하면서, 표본화 주파수 8KHz로 sampling된다. 디지탈로 바뀐 음성신호에 Rabiner와 Sambur가 제안한 average magnitude와 zero-crossing measurement algorithm[11,12]을 적용하여 시작점과 끝점 검출을 한다. 시작점과 끝점이 검출된 신호는 시간 영역에서 전체의 길이를 10개의 선형적 프레임으로 나누어 분석한다. 각 프레임으로부터 autocorelation 방법을 이용한 Durbin algorithm을

www.dbpia.co.kr

통해서 16차 LPC 계수를 추출한다[11,13]. 계산된 LPC 계수들로부터 식 (12)와 식 (13)을 사용하여 16차 LPC-cepstrum 계수들을 계산한다. 추출된 LPC-cepstrum 계수들을 STNN의 입력으로 사용하기 위해서 0과 1사이의 값들로 정규화한다.

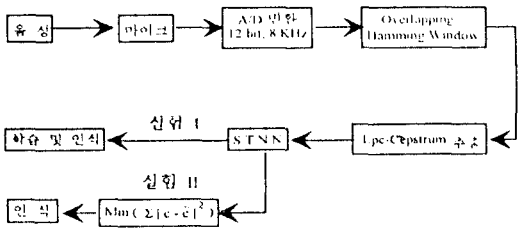

그림 5. 음성 인식 시스템

본 논문에서 시도한 시스템은 화자종속 시스템이며, 각 숫자에 대해 20번씩 발음한 총 200개의 data를 사용하였다. 1회 발음한 10개의 데이타는 network에 초기 weight로 사용하였다. 5회씩 발음한 50개의 데이타를 가지고 학습시켰으며, 나머지 140개의 학습에 참여하지 않은 데이타를 가지고 인식실험을 하였다.

STNN에서의 단어 인식 과정이 그림 6에 설명되어져 있다. 그림 6에서 세로축은 각 neuron의 출력값을 나타내고, 가로축은 입력 패턴의 구간수를 5로 간주했을 경우에 각 구간별로 그 때에 입력되는 음소를 시간에 따라 표시한 것이다. 입력 단어가 시간에 따라 STNN으로 들어올때 STNN의 각 층의 출력값은 패턴의 일치여부에 따라 변한다. 식 (4)의 attack function의 영향으로 전단계의 neuron에서 패턴이 일치되면 다음 단계에 큰 값을 넘겨 주어서 전체적으로 일치도가 가장 큰 단어를 인식하게 된다. 그림 6에서는 "일"을 인식하는데 유사단어인 "이"와 "칠"이 경쟁을 하며 출력값을 높여가지만 결국에는 전체적으로 일치도가 제일 큰 "일"에서 가장 큰 출력값을 내며 인식하게 된다.

4.2 실험 I

실험 I에서는, STNN의 입력벡터로서 LPC-cepstrum 계수를 사용했을때의 인식률을 입력벡터로서 LPC 계수를 사용했을때의 인식률과 비교하였다. 두 경우의 인식률이 표 1의 둘째줄과 셋째줄에 나타나 있다.

1.0				일	
0.8			일	칠	
0.6			칠 이		
0.4		일 이			
0.2		이 일	칠		
0.0	이 일	칠			

그림 6. 단어 "일" 의 인식 과정

표 1에서 보여주듯이, 특징 vector를 LPC 계수로 택했을 때에는 82.5%의 인식률을, LPC-cepstrum 계수로 택했을 때에는 90.0%의 인식률을 얻었다. 따라서, STNN에서는 입력 패턴을 LPC로 선택하는 것보다는 LPC-cepstrum을 사용하는 것이 인식률을 높이는 데에 기여할 수 있다는 것을 보였다.

표 1. STNN에서의 input pattern에 따른 인식률

Input pattern	인식률
LPC 계수	83.5%
LPC-CEP 계수 (실험 Ⅰ)	90%
LPC-CEP 계수 (실험 Ⅱ)	95%

표 2와 표 3에는 LPC 계수를 특징 벡터로 사용했을 경우와 LPC-cepstrum을 특징 벡터로 사용했을 경우의 인식률이 각 숫자음에 따라 각각 정리되어 있다. 표 2에서 보듯이, 특징 vector로 LPC 계수를 사용할 경우, 1과 8, 그리고 5와 9사이의 구별이 크게 혼동된다. 반면에, 특징 vector로 LPC-cepstrum 계수를 사용할 경우, 표 3에서 보듯이, 1과 8, 그리고 5와 9사이의 구별이 용이함을 알 수 있다. 상대적으로, LPC-cepstrum의 경우에서는, 1과 7, 그리고 6과 9 사이에 구별이 어려움을 알 수 있다. 따라서, 각 단어의 input vecter 공간으로부터 output neuron 공간으로 mapping 되는 과정이, 특징 vector를 LPC 계수로 택했을 때와 LPC-cepstrum 계수로 택했을 때에 서로 크게

www.dbpia.co.kr

다름을 알 수 있다. 전체적인 인식률에 있어서는, LPC-cepstrum의 사용이 LPC 계수의 사용보다 앞서는 것을 알 수 있다.

표 2. LPC 계수를 특징 vecter로 사용했을 때의 인식률

	0	1	2	3	4	5	6	7	8	9	#error	인식률
0	14											100%
1		13	1								1	92.8%
2			14									100%
3				14								100%
4					14							100%
5						11				3	3	78.5%
6							14					100%
7		4						10			4	71.4%
8		11							3		11	78.5%
9						3	2			9	5	64.2%
총 Error 수/총 인식률											24	83.5%

표 3. LPC-cepstrum 계수를 특징 vecter로 사용했을 때의 인식률

	0	1	2	3	4	5	6	7	8	9	#error	인식률
0	14											100%
1		14										100%
2			14									100%
3				14								100%
4					14							100%
5						13				1	1	92.8%
6							14					100%
7		5						9			5	64.2%
8									14			100%
9							8			6	8	57.1%
총 Error 수/총 인식률											14	90%

4.3 실험 Ⅱ

실험 Ⅰ에서 LPC-cepstrum의 사용이 LPC 보다 인식률에서 앞섬을 볼 수 있었다. 하지만, LPC-cepstrum의 경우 1과 7 그리고 6과 9 사이의 판별이 어려움을 지적하였다. 본 실험에서는 LPC-cepstrum 사용의 인식률을 더욱 향상시키기 위하여, Euclidean cepstral distance measure의 사용을 첨가하였다.

Euclidean cepstral distance measure의 사용 첨가를 위하여, 우선 각각의 STNN을 통하여 최종 출력값들을 얻는다. 최종 출력값들 10개 중 1, 2위의 차이를 미리 정해져 있는 threshold 값과 비교한다. 차이가 threshold 값보다 크면 STNN의 출력값 1위에 해당하는 단어를 최종 인식 숫자음으로 한다. 반면에, 차이가 threshold 값보다 작으면 출력값 1위와 2위에 해당하는 단어에 한하여 입력신호의 LPC-cepstrum과 각각 Euclidean cepstral distance를 계산한다. 두 Euclidean cepstral distance의 값들을 비교한후, 작은 distance 값을 배출한 경쟁단어를 최종 인식 숫자음으로 결정한다.

식 (14)에서와 같이, Euclidean distance measure를 사용하여 두 cepstrum 벡터들 사이의 차(즉, 각 성분의 차의 제곱의 합)를 구할 경우, 이는 주파수 영역에서 logarithm을 취한 각각 (즉, 입력패턴과 template 패턴)의 스펙트럼들 사이의 평균-제곱의 차이 (mean-squared difference)라는 직접적인 의미를 갖는다. 이와같은 Euclidean cepstral distance 계산의 첨가로 인식률을 95%까지 높일 수 있었다. 실험의 결과가 표 4에 정리되어 있다.

표 4에는 각 숫자음에 대한 인식률이 정리되어 있다. 표4에서 보여 주듯이, STNN에서 특징 vector로 LPC-cepstrum 계수를 선택하였을 때, 구별하기 어려웠던 6과 9 사이의 혼동을, Euclidean cepstral distance 계산 방법을 첨가하면 크게 높일 수 있음을 알 수 있다. Euclidean cepstral distance 계산의 첨가는 요구되어지는 계산량이 매우 적으므로, STNN을 이용한 음성인식의 실시간 처리에 있어서, 인식율을 높이면서 시간적 불리함을 주지 않는 매우 효과적인 방법이라 할 수 있다.

표 4. LPC-cepstrum 계수를 특징 vecter로 사용하고 Euclidean distance measure를 첨가했을때의 인식률

	0	1	2	3	4	5	6	7	8	9	#error	인식률
0	14											100%
1		14										100%
2			14									100%
3				14								100%
4					14							100%
5						14						100%
6							14					100%
7		5						9			5	64.2%
8									14			100%
9						2				12	2	85.7%
총 Error 수/총 인식률											7	95%

www.dbpia.co.kr

V. 결 론

기존의 신경망인 MLP (Multi Layer Perceptron)나 CPN (Counter Propagation Network)에서는 음성신호의 막대한 데이타량을 처리하기 위해 입력 노드의 수를 크게할 수 밖에 없다. 이로써 network이 커지며, 계산량이 많고 학습시간이 긴 단점이 있다. 그러나, STNN은 음성신호를 구간 별로 연속적으로 입력하므로 입력 노드의 수를 제한할 수 있고, 또한 network의 크기가 상대적으로 작고, 계산량이 적은 장점이 있다. 이 모델은 신경회로망의 고유한 특징인 잡음 극복 능력과 연상 기억 능력을 갖고 있으면서도, 막대한 데이타량에 따른 문제점이 적으므로 하드웨어로 구성하기가 용이하며, 대용량의 시스템을 구성할 수 있다.

본 논문에서는 LPC-cepstrum 계수들을 입력 패턴으로 하여 STNN을 통해 한국어 숫자음 인식 실험을 시도하였으며, 인식율을 LPC 벡터를 입력 패턴으로 한 기존의 논문과 비교 하였다. 입력 패턴을 LPC 계수로 선택하였을 때는 82.5%의 인식률을, 입력 패턴을 LPC-cepstrum 계수로 선택하였을 때는 90.0%의 인식률을 얻었다. 실험 결과에서 보여주듯이, 입력 패턴이 LPC 계수일 때보다는 LPC-cepstrum 계수일 때에 더 좋은 인식률을 얻었다. 따라서, LPC-cepstrum 계수가 STNN에서는 인식률을 향상시키는 입력 패턴임을 보였다.

또한, 본 연구에서는 LPC-cepstrum 계수를 입력 패턴으로 선택한 STNN에서, 최종 판별이 어려운 두 경쟁 단어 사이의 구별을 위하여, 두 경쟁 단어를 나타내는 각각의 LPC-cepstrum들과 입력 패턴을 나타내는 LPC-cepstrum 사이의 Euclidean cepstral distance를 구하고, 이들을 비교하여 최종 인식 판별을 하였다. STNN에서 입력 패턴을 LPC-Cepstrum 계수로 선택 하였을때의 90.0%의 인식률을 Euclidean cepstral distance 계산을 추가하여 인식율을 95%로 높였다. Euclidean cepstral distance 계산의 첨가는 요구되어지는 계산량이 매우 적으므로, STNN을 이용한 음성 인식의 실시간처리에 있어서, 인식률을 높이면서도 시간적 불리함을 주지 않는 매우 효과적인 방법이라 할 수 있다.

참고문헌

1. P. Demichelis, "On the Use of Neural Networkks for Speaker Independent Isolated Word Recognition," Int. Conf. on Acoust., Speech, and Signal Proc., May 1989.
2. 이종석, 이상욱, "신경망과 구문 분석을 이용한 한국어 연결 숫자음 인식," 대한 전자공학회 논문지, pp. 21-30, 1993.
3. 이영호, 정홍, "음절을 기반으로한 한국어 음성인식," 대한 전자공학회 논문지, pp. 11-22, 1994.
4. R. Cshalkof, Pattern Recognition, Statistical, Structural and Neural Approaches, Jone Wiley & Sons Inc., pp. 194-195, 1992.
5. R. Hetch-Nielson, Neuro Computing, Addison Wesley, 1990.
6. J. A. Freeman and D. M. Skapura, Neural Networks, Addison Wesley, 1990.
7. 백승우, 홍승홍, "시공간 패턴인식 신경회로망을 이용한 격리단어의 인식," 인하대학교 석사 졸업 논문, 1993.
8. J. Zurada, Artificial Neural Systems, West Info Access, 1992.
9. Y. Tohkura, "A Weighted Cepstral Distance Measure for Speech Recognition," IEEE Trans. on Acoustics, Speech, and Signal Proc., vol. ASSP-35, pp. 1414-1422, Oct. 1987.
10. L. R. Rabiner and F. Soong, "Single Frame Vowel Recognition Using Vector Quantization with Several Distance Measures," AT&T Technical Journal, vol. 64, pp. 2319-2330, 1985.
11. L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, pp.396-452, 1978.
12. L. R. Rabiner and M. R. Samber, "An Algorithm for Determining the Endpoints of Isolated Utterances," Bell Tech. Journal, vol. 54, no. 2, pp. 297-315, Feb. 1975.
13. P. Strobach, Linear Prediction Theory, a Mathematical Basis for Adaptive System, Springer-Verlag, pp. 13-36, 1990.

www.dbpia.co.kr

李鍾植 (Jong Sik Lee) 학생회원

1993년 : 인하대학교 전자공학과(학사)
1993년~현재 : 인하대학교 전자공학과(석사과정)

鄭在皓 (Jae Ho Chung) 정회원

1982년 : 美國 Univ. of Maryland (학사)
1984년 : 美國 Univ. of Maryland (석사)
1990년 : 美國 Georgia Institute of Technology (박사)
1984년~1985년 : 美國 Naval Surface Warfare Center, Electronic Engr.
1991년~1992년 : 美國 AT&T Bell Laboratories, 연구원
1992년~현재 : 인하대학교 공과대학 전자공학과, 조교수

www.dbpia.co.kr