Continuous Digit Recognition Using the Weight Initialization and LR Parser

Ki-Hoon Choi*, Seong-Kwon Lee*, Soon-Hyob Kim*

Abstract

This paper is a study on the neural network to recognize the phonemes, the weight initialization to reduce learning speed, and LR parser for continuous speech recognition. The neural network spots the phonemes in continuous speech and LR parser parses the output of neural network. The whole phonemes recognized in neural network are divided into several groups which are grouped by the similarity of phonemes, and then each group consists of neural network. Each group of neural network to recognize the phonemes consisits of that recognize the phonemes of their own group and VGNN(Verify Group Neural Network) which judges whether the inputs are their own group or not. The weights of neural network are not initialized with random values but initialized from learning data to reduce learning speed. The LR parsing method applied to this paper is not a method which traces a unique path, but one which traces several possible paths because the output of neural network is not accurate. The parser processes the continuous speech frame by frame as accumulating the output of neural network through several possible paths. If this accumulated path-value drops below the threshold value, this path is deleted in possible parsing paths. This paper applies the continuous speech recognition system to the continuous Korean digits recognition. The recognition rate of the phonemes is about 83% in speaker dependent. The recognition rate of isolated digits is 97% in speaker dependent, and 75% in speaker independent. The recognition rate of continuous digits is 74% in speaker dependent.

I. Introduction

In the research area related speech recognition system [38][39[40], there are several methods DTW(Dynamic Time Warping), HMM(Hidden Markov Model), ANN (Artificial Neural Network) ect[1][2][3]. DTW algorithm search the input utterance or candidate X, for optimal alignment paths with each word template C. There are many variations of the basic DTW and Dynamic Programming algorithms which use different distortion matrix, allowable paths, and search procedure. HMM models are a method for modeling system with discrete, time-dependent behavior characterized by common, short-time "process" and transitions between them. Recent, ANN is lively researched for speech processing. One of the goals of ANN is to process information in a manner similar to that of biological neural system[16][17][18].

Several connected speech recognition system were developed in America in 70's[4]. The Hearsay-I and Dragon system, which were developed in CMU(Carnegie Mellon University), performed a connected speech recognition experiment for 102 sentences which consists of 578 words. Sentence recognition rate of 31% and word recognition rate of 55% were obtained in the experiment which limited the number of speaker to 4 persons. Sentence recognition rate of 49% and word recognition rate of 83% were obtained in the Dragon system when the same data was used. Sentence recognition rate of 81% and word recognition rate of 97% were obtained in IBM system for 363 sentences. Dragon and IBM used finite state grammar and Hearsay-I did production rule. In early 80's, Hearsay-II speech understanding system showed the meaning understanding rate of 90% and the continuous speech recognition of 73% for the stiff grammar for 1011 words [5][6]. In 1980, connected speech recognition system, KEAL, was developed in France[7]. For the 324 sentences which consisted of 123 words, by transforming the input speech into phoneme symbol and thus by reducing the scanning range by using Context Free Grammar, recognition rate of 81.5% was obtained. In the early 80's SPHINX system was developed in CMU and it showed the word recognition rate of 94.1% and sentence recognition rate of 69.3 for 150 sentences which consist of 997 words by using

*Computer Engineering Department Kwang-Woon University
Manuscript Received April 18, 1996.
※This paper is supported by research fund, KOSEF 95-0100-22-01-3, 1996

HMM and bigram[8][9]. In the early 90's it achieved the word recognition rate of 95.5% and the sentence recognition rate of 78% by LR method to SPHINX system applying[10]. ATR laboratory in JAPAN did the connected speech recognition obtained the recognition rate of 88.4% in speaker dependent for 279 sentences which consisted of 1035 words by converging HMM and LR parsing method[11].

In this paper, our goal is to design a NN for continuous speech. For processing continuous speech, it is difficult for recognition system to use word-unit. Variability of speech and the number of vocabulary make it difficult to recognize the continuous speech. Therefore the system must be the system of phoneme unit[12][13]. The processing of transition region is the one of important thing in continuous speech by phoneme unit. In conventional NN, the transition region has been the one of main mis-recognition. In proposed paper, the additional VGNN(Verify Group Neural Network) can process this transition region [14][15]. Not using conventional method in weight initialization, VGNN initialize the weight from the learning data and this make it possible to shorten the learning time and decrease the error. The system processes the phoneme which is recognized in NN, in the continuous speech using LR parser.

Ⅱ. Continuous Speech Recognition System

The whole speech recognition system for continuous speech recognition is shown in Fig 1. CDR(Continuous Digit Recognition) system was divided into two parts; one is speech recognition part and the other language process part[41][42]. Speech processing part consists of 3 layers NN(Neural Network) which finds phonemes sequence from continuous speech input. We divided the phonemes into groups in order to consist the proposed NN and define the NN for each group. The NN of each group consists of two NN parts. first NN part verify the ponemes in group and the second NN part verify whether input phoneme belongs to its own group or not. In VGNN(Verify Group Neural Network) which was proposed in [14][15], there is the weight initialization which initialize the weights of NN from the learning data. Language process part consists LR paser which processes recognized phoneme in NN. LR parser begins with LR parsing table which was taken from CFG(Context Free Grammar).

2.1 VGNN(Verify Group Neural Network)

Continuous speech recognition system using conventional NN, TDNN[20][21][22], etc have the following problems. First, total recognition system extremely depends on the NN which verifies the groups. The second, each group was never learned for the data of others' group. When the random input which doesn't come under its own group comes into the NN, network may output high value. In this case, we can decrease the output value which increases the weight of each group's output. But if this process occurs, the total recognition system depends on the NN which verify the group more and more. To solve the above problem, we include the additional network, which verify whether the data of each gorup is included in its own group or not, inot the system. This additional network was called VGNN. The Fig.2 shows the NN which finds the phonemes in continuous speech. And an example of the NN which verify the k, p, k is shown in Fig 3. And weight Initialization too.

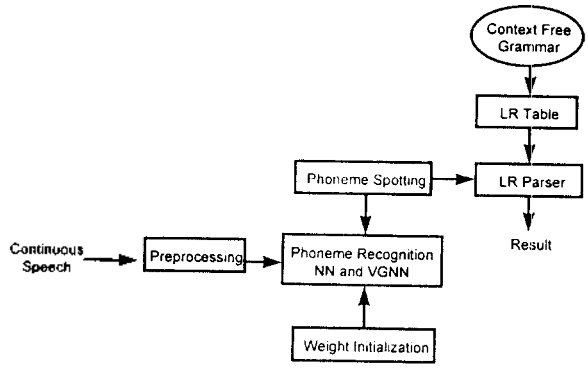


Fig 1. The block diagram of speech recognition system

Specially transition region, which is difficult to handle in continuous speech, can be processed through using VGNN. For example, when the input which was not learned comes especially in transition region, the system doesn's know if the input is included in correct phoneme group. Therefore the output of VGNN for the transition region is zero because the input is not included in its own group. The example of processing transition region is shown Fig 4.

the VGNN is very useful for processing continuous speech, but the amount of the learning data is very large. If the weight is initialized randomly by using conventional method, the error is not decreased to the level we want. To reduce the learning time of VGNN and to finish the learning within the level we want, we must initialize the weight from the learning data.

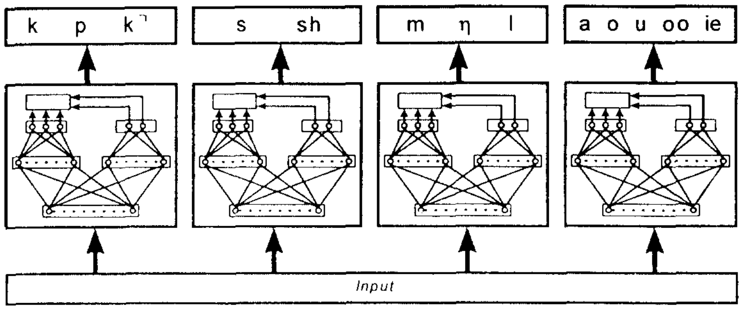


Fig 2. The total neural network architecture of spotting the phonemes in continuous speech

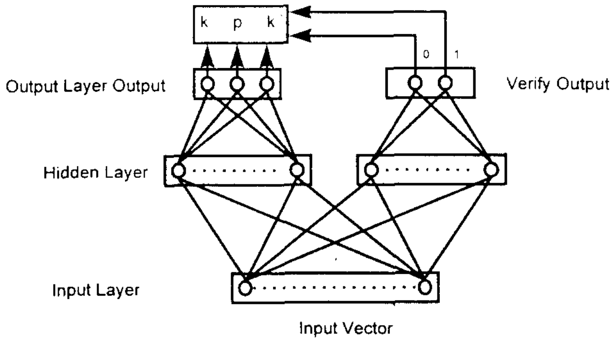
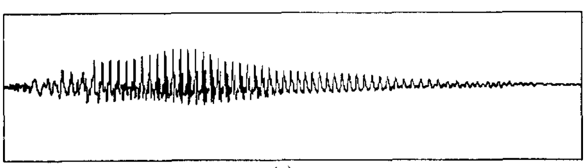


Fig 3. The neural network of recognizing the k, p, k group and VGNN

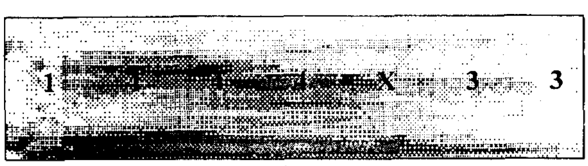
2.2 Weight Initialization for increasing the learning speed of NN

In the proposed system, the active function used by the nodes of each NN is the hyperbolic tangent function and is shown in Fig.5.

$$\tanh(X) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (2\text{-}1)$$



(a)



(b)

Fig 4. (a) The waveform /koŋ/ (b) spectrograph and the output of VGNN

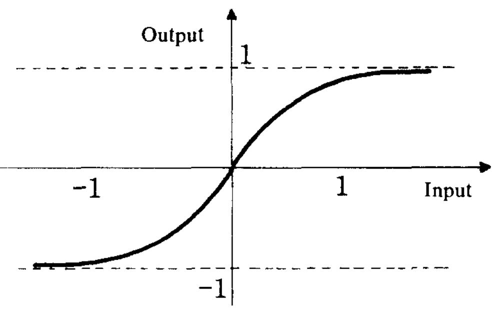


Fig 5. Hyperbolic tangent function

2.2.1 The method using SVD(Singular value Decomposition)

This method initialize the weight using SVD.[25][27][28]. In single hidden layer, the output is

$$Y = W2 \cdot f(W1 \cdot X + b1 \cdot u^{t}) + b2 \cdot u^{t} \quad (2\text{-}2)$$

$f(\cdot)$ is the tangent hyperbolic function in (3-1). X(n, X) and Y(m, N) is the matrix of the input and the output. W1(p, n) and b1(p, 1) represent the weight and the bias between input layer and hidden layer. W2(p, n) and b2(p, 2) represent the weight and bias between hidden layer and output layer. u(N, 1) represent the unit vector. In case that the NN operates in linear area of sigmoid function and $b1 = 0$, the equation (2-2) is simplified as

$$P = W2 \cdot W1 \cdot X + b2 \cdot u^{t} \quad (2\text{-}3)$$

in this equation is reformed for the **W1**$\cdot$***W2***, it is shown below

$$W1 \cdot W2 = (P - b1 \cdot u^{t}) \cdot X^{-1} \quad (2\text{-}4)$$

X is decomposited to $X = U \cdot \Sigma \cdot V^{t}$ by SVD and the equation (2-4) is represented as

$$W1 \cdot W2 = (P - b1 \cdot u^t) \cdot V \cdot \Sigma^{-1} \cdot U^{t} \quad (2\text{-}5)$$

If the number of hidden layer is H,

$$W1 \cdot W2 = (P - b1 \cdot u^t) \cdot V_H \cdot \Sigma_H^{-1} \cdot U_H^t \quad (2\text{-}6)$$

In this case, the possible decomposition is

$$W2 = (P - b2 \cdot u^t) \cdot V_H \cdot T^{-1} \; and \; W1 = T \cdot \Sigma_H^{-1} \cdot U_H^t \quad (2\text{-}7)$$

T is the intentional matrix which exists inverse matrix of(H, H)

2.2.2 The method using the center of cluster

This paper uses the weight initialization method which verifies the center of cluster between the hidden layer and the input layer, and initialize the weight between the hidden layer and the output layer using SVD(Singular Value Decomposition)[14][15][26]. The block diagram of the weight initialization method is shown in Fig.6.

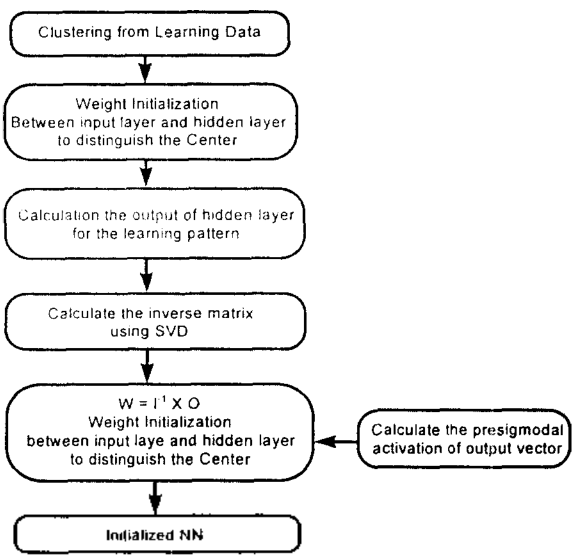
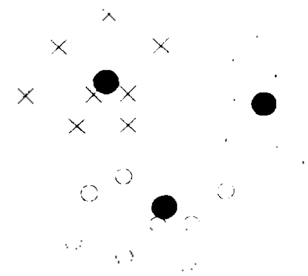


Fig 6. The block diagram of weight initialization

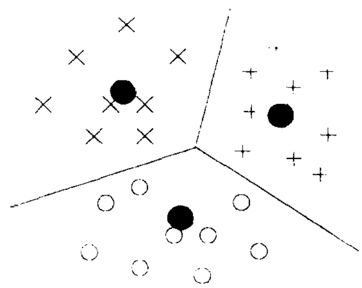
2.2.3 The weight initialization between Input layer and Hidden layer[19]

The weight initialization between Hidden and Input layer is processed to verify the cluster center of the learning data. First, the cluster center and the data for 3 group in 2ed dimension is shown in Fig. 7 (a). The thich point shows the center cluster. Fig.7 (b) shows the boundary line which is divided by two vertically for the center cluster. The equation of boundary line, which verify boundary between the center two cluster and represented as

$$X^t \cdot (P - Q) - \frac{1}{2}(P^t P - Q^t Q) = 0 \quad (2\text{-}8)$$



(a) 3 groups and each clusters of them



(b) A boundary lines for Fig (a)

Fig 7. Three groups and the their bisectional lines in two dimension

P and Q is the vector which represents the cluster center. For example, the two cluster center in 2cd dimension P(p1, p2), Q(q1, q2) are represented as

$$X_1 \cdot (p_1 - q_1) + X_2 \cdot (p_2 - q_2) + \frac{1}{2}(p_1^2 - q_1^2 + p_2^2 - q_2^2) = 0 \quad (2\text{-}9)$$

We define the hyperplant segment, which verify the center between two cluster. The sign of f(x) determine where the input x will be located in boundary and the quantity of f(x) represent the how far it is located from the hyperplane. We initialize the weight between hidden layer and input layer for the node in hidden layer to process the object of hyperplane segment which verify the two center cluster. First we represent the output function

$$f(X) = X^t \cdot (P - Q) - \frac{1}{2}(P^t P - Q^t Q) \quad (2\text{-}10)$$

of ith node in output layer,

$$f_i(\mathbf{X}) = O_i = g\left(\sum_j w_{ij} \cdot s_j(\mathbf{X}) + b_i\right) \quad (2\text{-}11)$$

represent the output of ith node in hidden layer, and represent tangent hyperbolic function

$$s_j(\mathbf{X}) = g(h_j(\mathbf{X})) = g\left(\sum_k w_{ik} \cdot x_j + b_j\right) \quad (2\text{-}12)$$

represent the input of ith hidden layer, hyperplane equation

$$h_j(\mathbf{X}) = \sum_k w_{ik} \cdot x_k + b_j = \mathbf{W1}_j^t \cdot \mathbf{X} + b_j \quad (2\text{-}13)$$

To classify the 2 center cluster for the hidden layer, equation (2-13) and (2-10) must be equal. Therefore the weight between input layer and hidden layer is initialized as (2-14)

$$\mathrm{W1}_j = \beta \cdot (\mathrm{P} - \mathrm{Q}) \qquad \beta:\text{scaling factor} \quad (2\text{-}14)$$
$$b_j = -\frac{\beta}{2} \cdot (\mathrm{P}^t \cdot \mathrm{P} - \mathrm{Q}^t \cdot \mathrm{Q})$$

2.2.4 The weight initialization between output layer and Hidden layer

The weight between output layer and Hidden layer is initialized with the output of hidden layer and presigmodial activation of desired output vector using SVD. The presigmodial activation is the value of input in output layer. The output value of hidden layer in kth pattern for initialized input and hidden layer and itsd equation is

$$I_k = (s(\sum w_{1i} \cdot p_{ki} + b_1), s(\sum w_{2i} \cdot p_{ki} + b_2), \ldots,$$
$$s(\sum w_{mi} \cdot p_{ki} + b_m), 1.0) \quad (2\text{-}15)$$

If the input of the output layer in kth pattern is Ok (presigmodial activation) the weight between hidden layer and output layer is represented by

$$\begin{pmatrix} \mathrm{I}_1 \\ \vdots \\ \mathrm{I}_n \end{pmatrix} \times \mathrm{W2} = \begin{pmatrix} \mathrm{O}_1 \\ \vdots \\ \mathrm{O}_n \end{pmatrix}, \quad \mathrm{I} \times \mathrm{W2} = \mathrm{O} \quad (2\text{-}16)$$

To solve the above equation, for I, we search the inverse matrix using SVD

$$\mathrm{W2} = \mathrm{I}^{-1} \times \mathrm{O} \quad (2\text{-}17)$$

3. LR Parser for the continuous speech recognition

3.1 LR Parsing

LR parsing was introduced by Kunth for the translating programming language and can be applied to context free grammar[29]. LR parser was parsed by LR parsing table which consists of two tables; action table and goto table. Action table determines the which action should be taken by parser next time, ACTION[s, a], from the present input symbol a and status s of the top in present stack. Action have four kinds of class; shift, reduce, accept and error. Shift means the action that moves the input symbol to, stack and reduce that replace the value of stack as much as the value of the left value in production rule and accept shows whether the input string is correct in grammar or not, error shows incorrect sentence because the present input symbol can not be represented in that status. Goto table determine the status of the next parser from status s and grammar symbol A. The basic LR parsing is summarized as follows.

Step 1. Initialization
: make the pointer p indicate the first symbol of the input

Stpe 2. Refer to ACTION[s, a]
: s is the status, which is represented by top of the stack, and a is the input s symbol, which was indicated by pointer p.

Step 3. If ACTION[s, a] = "shift s′"
then put the status s′ to the top of the stack and let the pointer p indicate the next input symbol.

Step 4. If ACTION[s, a] = "reduce $A \rightarrow \beta$"
then remove symbol $|\beta|$ in stack put the status indicated by Goto[s, A] to the stack

Step 5. If ACTION[s, a] = "accept"
then string sequence is recognized the correct sentence and end the parsing

Step 6. If ACTION[s, a] = "error"
then string sequence is misrecognized to correct sentence and ends the parsing fail.

Setp 7. return to 2 step.

3.2 Parsing method applied for speech[12][13][30][31][32][33][34]

Conventional LR parsing method, which was parsed by input character, traces the only path. This method can't be applied to speech because the phoneme symbol changed in speech is not always correct and the various possible paths must be parsed. One frame of phoneme is represented the value of the possibility for the each symbols by the output value of the NN which verifies the phonemes. The parser processes the uttered speech by one frame in adding the value of the possibility for these phoneme symbols through various paths. If the possibility of the added value is below some degree, the input symbol which has that value is out of the parsing path. In this paper, the proposed parser processes the speech by the frame. In this case, if the same phonemes are represented several time, we let the CFG(Context Free Grammar) to process theses cases. When the parser is processed by CFG, it can be processed for the repeated paths. In this case, parser take the path which has the maximum possibility and reject it for the repeated path. For example, the pronounced /il/ can be the phoneme sequence of /illll/ or /illll/. In these two cases, as the same continuous speech is processed, one path which has the high possibility value is taken and the others rejected. In this paper, the applied parser is processed by following steps[12][13].

In the above figure, parser initialized the M cells in order to parse for the possibility of M cells. And then each cell divided into N cell through the input symbol and each cell process the action by parsing table. For the each cell proposed by action from the cell in repeated paths, the cells which have maximum possibility value remains and the others are rejected and sorted descendant order. Finally, in the sorted cells, M cells are taken and the others rejected.

4.1 Experiment for phoneme

The number of continuous digit utterance are total 14. Total phoneme used in the proposed system is shown table 4-1 and this is the case for isolated digits. In continuous digits, because of the co-articulation effect, additive phonemes are needed as shown below

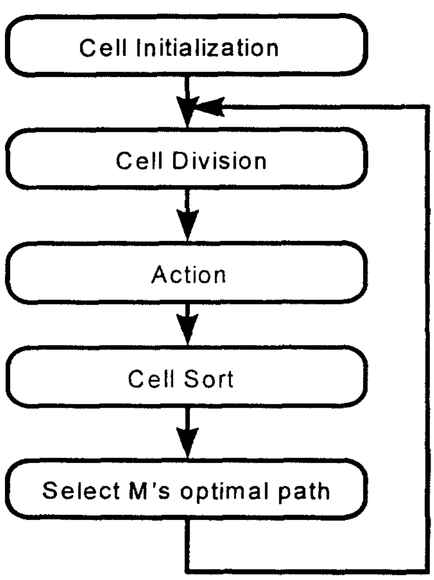


Fig 8. The procedure of LR parsing

Ⅳ. Simulation result and Discussion

For simulation test, the speech signal was sampled at 10Khz, low-pass filtered at 4Khz and digitized with a 16 bits A/D converter in TMS320C30 board. The simulation was performed by using personal computer (IBM-PC/Pentium 75Mhz). The procedure of extracting feature parameter for in put speech is shown in Fig. 9.

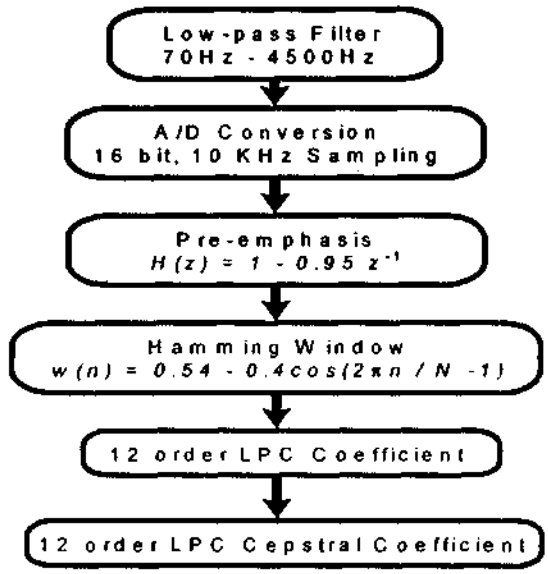


Fig 9. The extraction of feature parameter

Table 1. The phoneme generating in the continuous digit

CONSONANTS	Group 1	k	공구
		p	팔
		k┌	육
	Group 2	s	삼사
		ch	칠
	Group 3	m	삼
		ŋ	공
		l	팔
VOWELS	Group 4	a	삼사팔
		o	오공
		u	구
		oo	육
		i	일이칠
		e	에

Table 2. The additive phoneme by coarticulation

Additional phoneme	r	ㅆ	ㄲ
Utterance digit	/1에/, /7에/, /8에/	/13/, /63/, /73/, /83/	/60/, /69/

4.1.1 Experiment various number of hidden layer

Another important thing in structuring NN is the number of hidden layers. If the number of hidden layer is too small, it takes much time for learning and the error doesn't decrease the level to the degree we want. Namely, the NN has the small number of weight to learn whole learning data. If the number of hidden layer is too large, the number of learning time decrease but the large number of weight brings much calculation. The result of the number of hidden layer is shown in table 4-3.

Table 3. The phoneme recognition of the order of LPC cepstrum

Order	p=11	p=12	p=16	p=20
Learning rate	92.9	93.4	93.8	93.5
Recognition Rate	95.5	96.1	96.5	94.5

The experiment for the four weight initialization is shown table 4-4, too.

Table 4. The phoneme recognition of the number of hidden node

No. of hidden layer	20	48	78	90
Learning rate	93.4	93.8	94.1	96.4
Recognition rate	96.1	97.2	97.9	96.8

4.1.2 Experiment for the whole phonemes

The recognition rate is shown table 4-5 after initializing the weight for the center of cluster.

Table 5. The phoneme recognition experiment of the four weight initialization

	random	Nguyen-Widrow	SVD	Center of cluster
No. of ineration	3245	429	1	47
Error	0.005	0.005	0.0027	0.005
Learning Rate	100%	100%	100%	100%
Recognition Rate	92.1%	89.4%	68.6%	91.3%

4.1.3 Experiment for the VGNN

As shown above table, the recognition rate for VGNN is better than each phonemes. Because each group for the speech is grouped, the LPC cepstrum coefficient of input speech is little overlaid in vector dimension.

4.2 Experiment for isolated digit

In Fig 10, the recognition result is shown for isolated digit of LR parser in Top 3 and in table 4-7, the recognition result for isolated digit in speacker dependent and independent is shown.

Table 6. The recognition rate of total phoneme

	Phoneme	No. of hidden layer	Initial recognition rate(%)	Recognition rate after learning(%)
Group 1	k	27	75.8	82.1
	p		71.4	79.3
	k'		74.2	84.3
Group 2	s	9	83.3	91.2
	ch		79.3	89.7
Group 3	m	27	83.6	87.0
	ŋ		83.1	88.9
	l		73.7	82.5
Vowel	a	26	92.8	100.0
	o		94.2	94.2
	u		85.3	97.8
	oo		91.7	94.2
	i		84.5	92.5
	e		89.5	99.6

Table 7. The recognition rate of four VGNN

	Intial Recognition rate(%)	Recognition rate after learning(%)
Group 1(k, p, k$^{\Gamma}$)	94.5	96.5
Group 2(s, sh)	96.1	97.3
Group 3(m, η, l)	95.7	95.8
Group 4(a, o, u, oo, i, e)	89.1	97.8

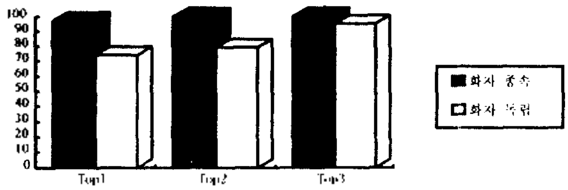


Fig 10. The recognition result of LR parser in Top 3

4.3 Experiment for continuous digit

The reference pattern used in continuous digit recognition experiment is shown table 4-8, and this consists of the whole case in continuous digit uttered. The 21 continuous digits pattern is shown below.

Table 8. The recognition result of isolated digit
(a) speaker dependent (b) speaker independent

Digit	0	1	2	3	4	5	6	7	8	9
0	10									
1		10								
2		1	9							
3				10						
4				1	9					
5						100				
6							10			
7								10		
8									10	
9	1									9

(a)

Digit	0	1	2	3	4	5	6	7	8	9
0	28									12
1		30	6					4		
2		6	34							
3				30	8			2		
4				10	26				4	
5		2	2			36				
6	2				1		26	3		8
7		8						31	1	
8	4			8			1		27	
9	10		4							26

(b)

6 male speakers are participated in recognition experiment. 3 male speaker's data is used for speaker dependent experiment and the other 3 male speaker's is for speaker independent. The result is shown in table 4-9.

Table 9. The reference pattern of continuous digit

	Telephone No.		Telephone No.
1	512에 0257	12	270에 9483
2	630에 1349	13	396에 0011
3	745에 6780	14	408에 6281
4	826에 9318	15	689에 6542
5	904에 0371	16	209에 1921
6	910에 2388	17	147에 3324
7	843에 4616	18	986에 5066
8	729에 5522	19	569에 1775
9	607에 7641	20	795에 9785
10	358에 8736	21	448에 1234
11	153에 0599		

4.4 Discussion

First, the size of the input layer is determined by input vector. Experimentally using 12 order LPC cepstrum coefficient, 96.1% recognition rate is gotten which is the best score and the size of hidden layer is determined by minimum number through the number of VQ(Vector Quantization)[44][45] cluster experimentally. There is much time for learning NN. But when the weight is initialized with the center of cluster, learning time is much decreased and good recognition result is made. In NN, the recognition of phoneme frame by frame has much bad result. But as using the restriction for the grammar of continuous speech, the continuous speech is processed effectively by parser. These parser is made by CFG which modeled continuous speech by grammar. There is a indistinct thing for the grammar recognizing continuous digit. This is shown in Fig. 11.

The above grammar, if the digits /22/ or /55/ come, the parser doesn't know when to reduce or shift after

Fig 11. The grammar which contains two shift/reduce problems.

which the phoneme symbol. In this paper, parser determines the shift or reduce by silence region. But when the input speech comes continuously, there isn't a silence region behind the /2/ or /5/. In practice, phoneme recognition system, after the silence region is removed, the phoneme symbol is given to the parser. This is the reason why the system recognizes the continuous digit real time. But for the silence region of the above two cases, tit is designed for the system give the silence region to the parser.

V. Conclusion

We proposed the NN and the LR parser for recognizing the continuous digit. For the continuous speech recognition, we proposed the VGNN, for the learning time, proposed weight initialization with the clustering of center, for the continuous speech recognition from the output of NN, proposed the LR parsing method. In this paper the VGNN consists of 3 layer NN and this processes well in continuous system for the transition region effectively. When the weight is initialized with the center of cluster, we got the decrement of learning time and the good recognition result. The proposed continuous speech system of phoneme unit using VGNN and LR parser determines the vocabulary learns the phoneme and designs the LR parser with CFG. This can be extended easily not only continuous digit but continuous speech.

In this paper, the recognition rate of continuous digit is 74% in speaker dependent and 59% in speaker independent.

Reference

1. Hiroaki Sakoe, Seibi, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 26, No. 1, Feb. 1978.
2. L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *IEEE Proceedings*, 1989.
3. R.P.Lipmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, April, 1987.
4. D. R. Reddy, "Speech Recognition by Machine: A review," *Proceedings of the IEEE* Vol. 64, No. 6, April, 1976.
5. A. J. Gray, J. D. Marker, "Distance Measures for Speech Processing," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 24, No. 5, Oct., 1976.
6. B. Lowerre, "The Harpy Speech Understanding System," *Trends in Speech Recognition*, Prentice Hall, 1980.
7. G. Mercier, D. Bigorgne, L. Miclet, L. Le Guennec and M. Guerre, "Recognition of Speaker-Dependent Continuous Speech with KEAL," *IEEE Proceedings* Vol. 136, No. 2, April, 1989.
8. Kai-Fu Lee, "Automatic Speech Recognition," *Kluwer Academic Publishers*, 1989.
9. Kai-Fu Lee, R. Reddy, An Overview of the SPHINX Speech Recognition System," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Jan., 1990.
10. Kenji Kita, Wayne H. Ward, "Incorporation LR Parsing into SPHINX," *ICASSP-91*, 1991.
11. Toshiyuki Hanazawa, Kenji Kita, Satoshi Nakamura, Takeshi Kawabata, Kiyohiro Shikano, "ATR HMM-LR Continuous Speech Recognition System," *ICASSP-90*, 1990.
12. Kang-sung, Lee, "A study on the phoneme classification neural network and LR parsing for connected word recognition," *Ph. D thesis, Kwang Univ.*, 1992.
13. Hyung-jun Lee, "A study on the Phoneme Based Connected Word Recognition Using Glue Modified Block Windowed Neural Network" *Ph. D thesis, Kwang woon Univ.* graduate, 1994.
14. Ki-hoon, Choi, Lee-hyung Kim, Seong-kwon Lee, Soon-hyob Kim, "A study on the Verify Group Neural Network and weight initialization for the continuous speech recognition," *The 12th Speech communication and signal processing workshop proceeding* 1995.
15. Ki-hoon, Choi, Seong-kwon Lee, Kang-sung Lee, Hyung-jun Lee, Soon-hyob Kim, "A study on the for continuous speech recognition for Voice Dialing System" *Acoustical society of Korea Conference Proc.* 1995.
16. Laurene Fausett, "Fundamentals of Neural Networks: Archietecutre, Algorithms, and Applications," Prentice Hall, 1994.
17. S. Y. Kung, "Digital Neural Networks," *Prentice Hall*, 1993.
18. James A. Freeman, David M. Skapura, "Neural Networks," Addison-Wesley Publishing Company, 1991.
19. Bunpeilric, Sei Miyake, "Capabilities of Three-layered Perceptrons," *IJCNN-88*, 1988.
20. Alex Waible, Hiderfumi Sawai and Kiyohiro Shikano, "Consonant Recognition by Modular Construction of Large Phoneme Time-Delay Neurl Networks," *ICASSP-89*, 1989.
21. Hidefumi Sawai, Alex Waibel, Masanori Miyatake and Kiyohiro, "Spotting Japanese CV-Syllables and Phonemes Using Time-Dealy Neural Network," *ICASSP*-89, 1989.
22. Masanori Miyatake, Hidefumi Sawai, "Yasuhiro Minami and Kiyohiro Shikano, "Interfrated Training for Spotting Japanese Phonemes Using Large Phonemic Time-Dealy Neural Networks," *ICASSP*-90, 1990.
23. Kishan G. Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka, "Bounds on the Number of Samples Needed for

Neural Learning," IEEE Transaction on Neural Network, 1991.
24. Derrick Nguyen and Bernard Widrow, "Improving the Learning Speech of 2-layer Neural Networks by Choosing Values of the Adaptive Weights," *IJCNN-89.*, 1989.
25. Pascale Hirschauer, Pascal Larzabal and Henri Clergot, "Design of Neural Estimators: Second Order Backpropoagation, Initialization and Generalization," *IJCNN-94.*, 1994.
26. S. Gavin Smyth, "Designing Multilayer Perceptrons from Nearest-neighbor Systems," *IEEE Transaction on Neural Network*, 1992.
27. Howard Anton, Chris Rorres, "Elementary Linear Algebra," *John Wiley & Sons, Inc.*, 1994.
28. Gilbert Strang, "Linear Algebra and its Applications," *Jarcout Brace Jovanovich, Inc.*, 1988.
29. Margaret W. "Parsing Natural Language," *Academic Press*, 1983.
30. Kenneth W. Church, "Phonological Parsing in Speech Recognition," *Kluwer Academic Publishers*, 1987.
31. Masaru Tomita, "An efficient word lattice parsing algorithm for continuous speech recognition," *ICASSP-86.*, 1986.
32. Hermann Ney, "Dynamic Programming Speech Recognition Using a Context-Free Grammar," *ICASSP-91*, 1991.
33. Kenji Kita, Takeshi Kawbata, Hiroaki Saito, "HMM continuous speech recognition using predictive LR parsing," *ICASSP-89.*, 1989.
34. Hidefumi Sawai, "TDNN-LR Continuous Speech Recognition System Using Adaptive Incrental TDNN Training," *ICASSP-91.*, 1991.
35. Keninosuke Fukunaga, "Introduction to Statistical Pattern Recognition," *Academic Press*, 1990.
36. Charles W. Therrien, "Discrete Random Signals and Statistical Signal Processing," *Prentice Hall*, 1992.
37. Nicholas Kalouptsidis and Sergios Theodoridis, "Adaptive System Identification and Signal Processing Algorithms," *Prentice Hall*, 1993.
38. Lawrence Rabiner, Biing-Hwang Juang, "Fundamentals of Speech Recognition," *Prentice Hall*, 1993.
39. Sadaoki Furui, "Digital Speech Processing, Synthesis, and Recognition," *Marcel Dekker, Inc.*, 1992.
40. Shuzo Saito, Kazuo Nakata, "Fundamentals of Speech Signal Processing," *Academic Press, Inc.*, 1981.
41. C. Wheddon, R. Linggard, "Speech and Language Processing," *Chapman and Hall*, 1990.
42. Douglas O'Shaughnessy, "Speech Communication," *Addison-Wesley Publishing Company* 1987.
43. J.D. Markel, A.H. Gray, "Linear Prediction of Speech," *Springer-Verlag Berlin Heidelberg*, 1976.
44. R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, Vol. 1, Apr., 1984.
45. Allen Gersho, Robert M. Gray, "Vector Quantization and Signal Compression," *Kluwet Academic Pulbishers*, 1992.

▲ Ki-Hoon, Choi

1994. 2.: Dept. od Computer Engineering, KwangWoon Univ. (B.S.)
1996. 2.: Dept. od Computer Engineering, KwangWoon Univ. (M.S.)
1996. 3.~present: Research Engineer, Switching Group Dept. 5, R&D Center, DAEWOO TELECOM LTD.

▲ Seong-Kwon, Lee

1988. 2.: Dept. of Computer Engineering, KwangWoon Univ.(B.S.)
1990. 2.: Dept. of Computer Engineering, KwangWoon Univ.(M.S.)
1993.~present: Enrolled in a Ph.D degree of KwangWoon Univ.

▲ Soon-Hyob, Kim

Professor.
Dept. of Computer Engineering, Kwangwoon Univ.
See Vol. 14, No. 3