TR-I-0174

Study on Combining HMMs and Neural Network Models

TDNN-HMM for Phoneme recognition

Alain BIEM & Masahide SUGIYAMA

Abstract

This report focuses on the combination of Hidden Markov Models (HMMs) and Neural networks. Hidden Markov Models are a stochastic approach for continuous speech well suited to cope with the variability of speech. On the other hand, neural networks have shown high classification power for short speech utterances. Therefore, we can try to build a system with the advantage of Hidden Markov models and neural networks. We first review some of the most interesting works on this field recently and secondly we discuss a new idea: To build a codebook from the TDNN output units and train HMMs using the Fuzzy-VQ algorithm. We trained several discrete HMMs for the recognition task of /b/, /d/, /g/ using just one TDNN-generated codebook and we achieved a recognition rate of 97.2%. Close Interpretation reveals that the quality of extracted feature representations could be more important than the amount of data used for training as well as the discrimination power of the Neural Networks used here as a preprocessor.

ATR Interpreting Telephony Research Labs.

Contents

1 Introduction 1

2 Combination of NN and HMM for Speech Recognition 1
- 2.1 Description of HMM . . . 1
  - 2.1.1 The theory of HMM and Its Application to Speech Recognition . . . 1
- 2.2 Description of Neural Network Model . . . 2
  - 2.2.1 An Introducing to computing with Neural Nets . . . 2
  - 2.2.2 Review of Networks for speech recognition . . . 2
  - 2.2.3 Large vocabulary recognition using linked predictive neural networks . . 2
- 2.3 Combination of HMMs and Neural Network Models . . . 2
  - 2.3.1 Continuous speech recognition using Multilayer Perceptrons . . . 2
  - 2.3.2 TDNN for HMM Recognizer . . . 3
  - 2.3.3 Combining HMM and Neural Nets . . . 3
  - 2.3.4 Word Recognition using Hidden control neural architecture . . . 3
  - 2.3.5 Alpha-nets:A recurrent "neural" networks architecture with an HMM interpretation . . . 3
  - 2.3.6 Competitive Training:A connectionnist Approach to the Discriminative Training of Hidden Markov Models . . . 3
  - 2.3.7 Neural Network Driven Hidden Markov Model . . . 4

3 TDNN-HMM for Phoneme Recognition 4
- 3.1 Architecture . . . 4
- 3.2 TDNN . . . 5
- 3.3 Phoneme Models . . . 7

4 Phoneme Recognition 7
- 4.1 Speech data . . . 8
- 4.2 Label generation . . . 8
- 4.3 VQ Codebook Generation . . . 10
- 4.4 Preliminary experiment on the /b/, /d/, /g/ task . . . 15
  - 4.4.1 VQ distortion and Recognition Rate . . . 16
- 4.5 Recognition of all Japanese Phonemes . . . 17
- 4.6 Discussion . . . 21

5 Conclusion 21

6 Future works 21

A Regular Talk 23

List of Tables

1 Number of VQ training samples 10
2 Decreasing VQ distortion 10
3 Number of training data 15
4 Recognition Data 15
5 Recognition rate using Fuzzy-VQ 16
6 Recognition rate using Hard VQ 16
7 Comparison of recognition rate 17
8 Recognition result for all the phonemes using Fuzzy VQ 18
9 Recognition result for phoneme categories using Fuzzy VQ 19
10 Recognition rate of all the phoneme using Hard VQ 20
11 Comparison of four recognition methods 20

List of Figures

1 The architecture of recognition system 5
2 The architecture of TDNN 6
3 The architecture of HMM 7
4 Label generation 9
5 Decreasing VQ distortion 11
6 Values of VQ codes (32 codes) 12
7 Values of VQ codes (64 codes) 13
8 Values of VQ codes (128 codes) 14

1 Introduction

Pattern classification using neural networks has proved to be a powerful way to classify speech units.The spotted features by the TDNN can enable one to study the well-suited parameters examine the way to use them in a link with Hidden Markov modeling. The already well-trained TDNN for large vocabulary speech recognition showing a good discrimination for the stop consonant /b/, /d/, /g/ and also high performance of the recognition all phonemes, have lead us to use this type of networks for our experiment. The HMM capability to deal with the time warping of speech, and the flexibility that enables it to model a pre-chosen unit of speech makes it suitable for continuous speech recognition. The question is how to use the high discrimination power of a neural network and the HMM in a way to take both advantages of these two speech recognition methods. Some studies have been made in this subject and many proposals to improve the discriminant ability of the HMM.

We first present an overview of other researchers' works on this area. These are the main points from these reports. The summary of these reports could be useful to those that have an interest in building a hybrid system of HMM and neural networks. Some reports are very similar, so we will not mention all of them. Nowadays, the main part in the combination of Hidden Markov models and neural is just theory, as it looks obvious that the building of an hybrid system that will conserve the main advantages of Hidden Markov models and Neural Networks could be reached in several ways. Studies could be in the definition of some new error criterion linked with the forward-backward algorithm or just a cooperation of the two methods to increase the recognition rate.

These reports will then show the theoretical aspect of the HMM and neural net hybridization. We also use one of those ways to make some experiments. The TDNN was used as a preprocessor for a Hidden Markov Model. The most common acoustic parameters are being cepstral coefficients, their time derivatives, the signal energy and its derivatives. Here is an experiment in which we do not include derivatives information about spectral transition in our training data as we hope this will be taken in account by the TDNN delays. This shows the good integration of the two models.

2 Combination of NN and HMM for Speech Recognition

2.1 Description of HMM

2.1.1 The theory of HMM and Its Application to Speech Recognition

L.R.Rabiner[9]

This is an essential introducing report for HMM. The presentation is very clear and the vocabulary quite simple to understand. The theoretical and the technical aspects of HMM are presented as well but the author does not mention the drawbacks of HMM. This reports is a reference about HMM.

2.2 Description of Neural Network Model

2.2.1 An Introducing to computing with Neural Nets

R.Lipmann[12]

The author introduces us what exactly a neural networks is. After reviewing all the type of neural networks, their main uses, their advantages and drawbacks, the author ends with quite an optimistic conclusion. This report is useful in introducing the main ideas about NN, although it lacks some mathematical demonstrations that would help to compare the performance of the main algorithms.

2.2.2 Review of Networks for speech recognition

R. Lipmann[11]

The performance of many type of neural networks on speech recognition are presented. The author agrees with the fact that the performance of the multilayer perceptrons, TDNN and recurrent networks, whenever they show some good power of classification, is not as good as the HMM for the continuous speech recognition.

2.2.3 Large vocabulary recognition using linked predictive neural networks

J.Tebelski & A. Waibel[8]

The idea is simple: a neural network is trained to predict the next sample. For each phoneme a network is build as a predictor of the frames of the phoneme. During the recognition, the input frame is compared to the prediction of the network and the network with the closest output is chosen. At the end we have a set of neural nets, each one being equivalent to a unit of speech. The idea is interesting but there is no clear link with HMM.

2.3 Combination of HMMs and Neural Network Models

There are several approaches that combine HMM and Neural Network model. These include the interpretation of Hidden Markov Model into connectionnist models as well as the complementary use of the two models. HMM is well suited to represent global time sequence pattern. On the other hand, neural nets are good in representing local static pattern.

2.3.1 Continuous speech recognition using Multilayer Perceptrons

N.Morgan & Boulard[1]

The authors present a development of a hybrid system of a phoneme-based, speaker-dependent system in continuous speech recognition using multilayer perceptron(MLP) and HMM. The output of the MLP are considered as an estimation of the maximum a posteriori probabilities for an HMM. In other words, they use the output of an MLP (produced by the input of one frame) as emitted probabilities for a discrete HMM.

2.3.2 TDNN for HMM Recognizer

Weiye MA & D.V.Compernolle[10]

The main topic is the use of a TDNN as a preprocessor instead of using a VQ system. The TDNN is trained to generate phone-like labels. The labels are computed from a TDNN network and passed on to an HMM recognizer. They assume that the number of phonetic models in the HMM system is equal to the number of labels, therefore the HMM can be used to produce the training data for another HMM by the use of the Viterbi algorithm. This system is interesting, although the results are not as good as a usual HMM. This is close to our idea, but this was only used to train new HMM by giving TDNN segmentation.

2.3.3 Combining HMM and Neural Nets

Les T.Niles and Harvey F.Silverman[2]

The purpose of the report is how to build an HMM net. An HMM net is a neural network whose purpose is to compute the forward-backward algorithm. The structure of such a net is proposed and also how to use those nets to make some classifier networks. The main advantage is that you do not have to consider the probability constraints. The main result is that an HMM is just a certain type of neural net. But this does not help us to increase the power of discrimination added to the adaptation of the speech continuity.

2.3.4 Word Recognition using Hidden control neural architecture

Esther Levin[3]

This is one attempt to find a hybrid model between HMM & NN. The model is called HCNN. It combines non-linear prediction with HMM. The HCNN is trained to predict the next sample. The original fact is that the input is separated into two part: the sample X and the control input C. This control is equivalent to a state of an HMM and the variation of this state is driven by noise. This idea is interesting.

2.3.5 Alpha-nets:A recurrent "neural" networks architecture with an HMM interpretation

J.S.Bridle[4]

This paper starts by the comparison of the HMM recognizer and recurrent NN. The alpha calculation in HMM is viewed as a recurrent neural network and an error criterion is proposed for applying the back propagation algorithm. This is quite similar to the HMM nets discussed above. There is no practical result.

2.3.6 Competitive Training:A connectionnist Approach to the Discriminative Training of Hidden Markov Models

S.J.Young[5]

The paper presents a view of HMM as a connectionnist model and shows how the error back propagation method can be used to train HMM. Thus, there is a definition of an error that is used to train the HMM net. For the experiments, they use one HMM per phoneme. But training takes many iterations. The results are almost satisfying.

2.3.7 Neural Network Driven Hidden Markov Model

E.Tsuboka[6]

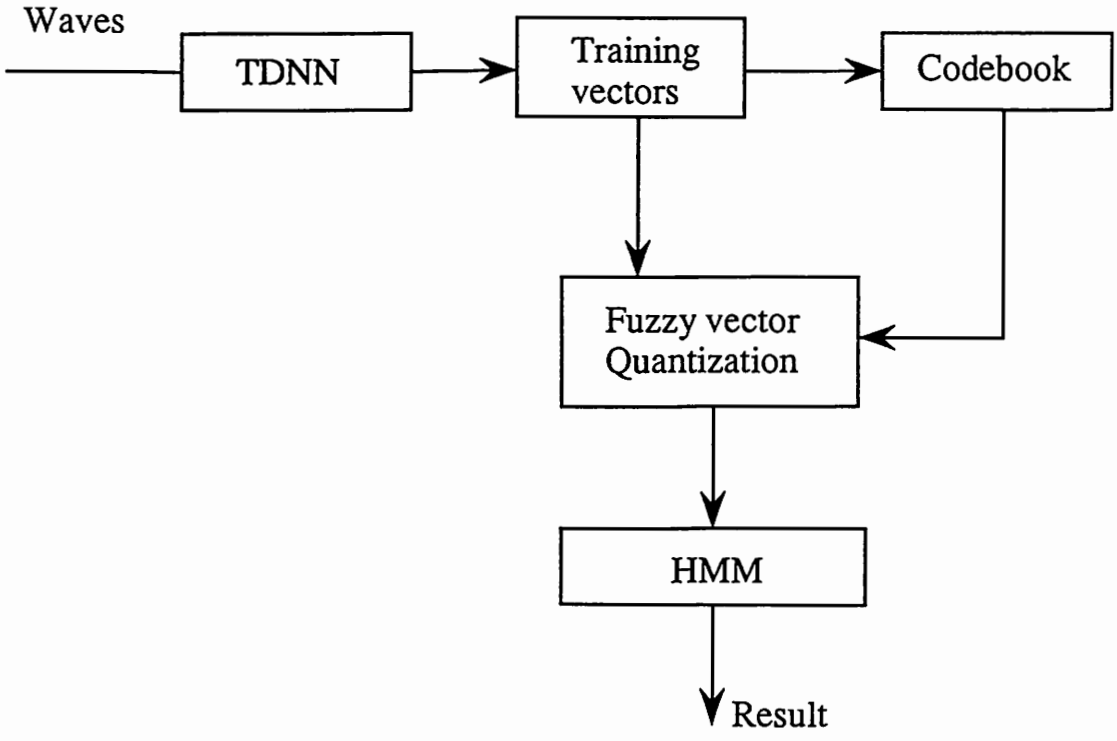
This aim is to represent the time variant phenomena in each state in HMM. The author proposed linear and non-linear models. For a non-linear model, parameter estimation is so difficult that non-linear mapping by neural network is applied. So, each state in HMM is represented with one NN. In conventional continuous HMM, probability b is Gaussian. In predicted one, the mean vector in Gaussian model is estimated from the previous vector. This model is a extension of Iso's Neural Prediction DP to HMM. Modified Baum-Welch algorithm for HMM parameter estimation includes Error Back-Propagation algorithm.

3 TDNN-HMM for Phoneme Recognition

Here is a presentation of the way we have taken to improve the recognition rate of the HMM linked with the TDNN. It is to use a TDNN output features to train Hidden Markov Models.

3.1 Architecture

Speech wave is preprocessed by the TDNN and the result is 24 dimensional vector whose elements are activations values of the output units of the TDNN. So, each value in the vectors represents an activation of the output unit of the TDNN. We then use those vectors to produce the codebook and to train the HMMs. The procedure (block diagram) is shown in Fig.1.



Architecture

Figure 1: The architecture of recognition system

3.2 TDNN

We use the Large-Phonemic-TDNN (Fig.2) for our experiment. It has 24 output units that correspond to the activation of the 24 Japanese phonemes i.e /b/, /d/, /g/, /p/, /t/, /k/, /m/, /n/, /N/, /s/, /sh/, /h/, /z/, /ch/, /ts/, /r/, /w/, /y/, /a/, /i/, /u/, /e/, /o/, /Q/. The notation /Q/ represents the silence. The input layer is composed of 240 units. The input data are 15 frames of 16 melscale spectrum coefficients (with 10 ms frames rate) computed from FFT coefficient and normalized between -1 and +1 among 15 frames. [7].

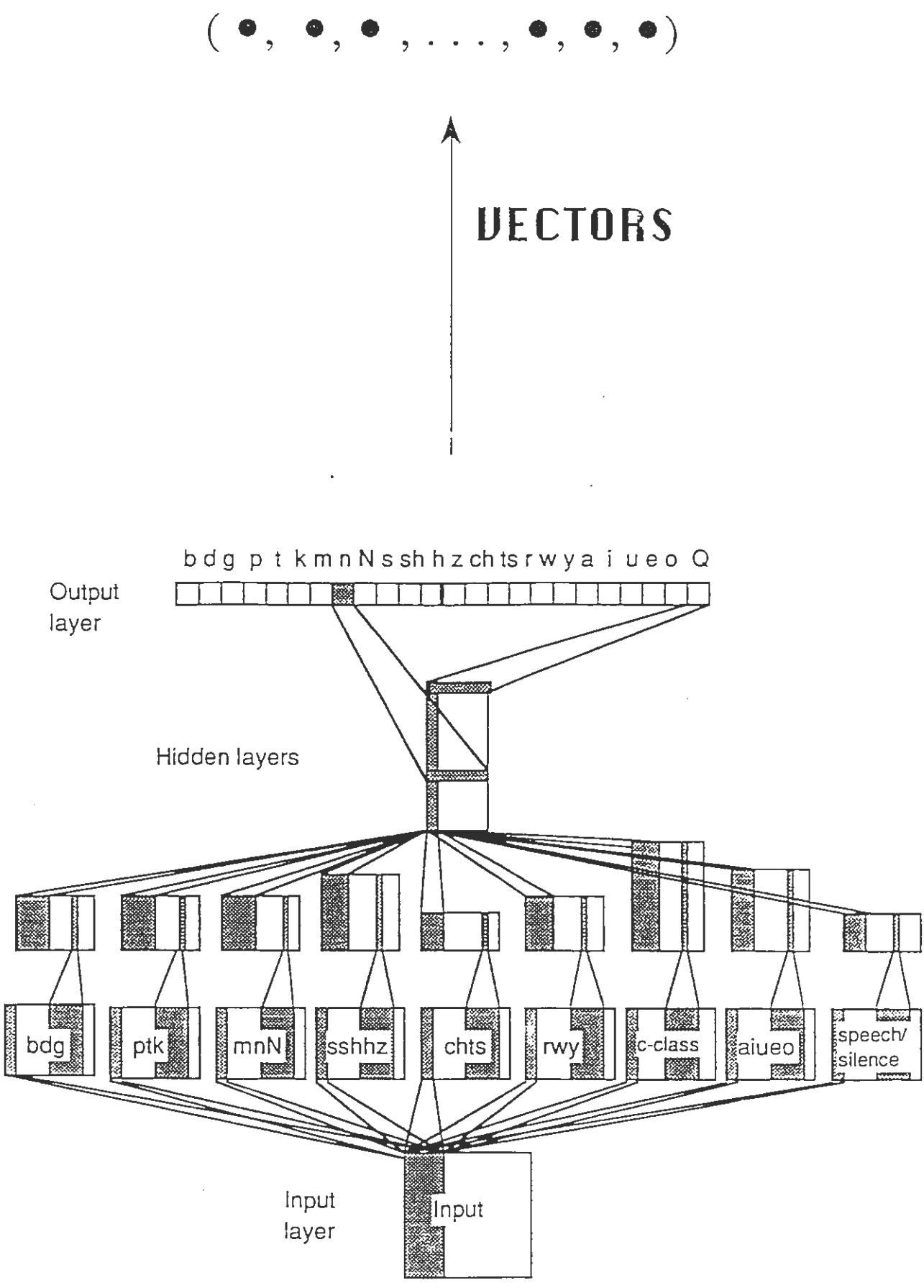


Figure 2: The architecture of TDNN

The training algorithm is the widely used backpropagation method, with the connection weight initialized between -0.5 and +0.5. As the output values are continuous parameters between -1 and +1, we have considered the strongest unit as a result.

3.3 Phoneme Models

An important choice concerns the topology of the Hidden Markov Model (Fig.3) and the tying of probability distribution [9]. We choose a discrete HMM for our experiment. An HMM consisted of 4 states (including the last state), with loops in the three first states.

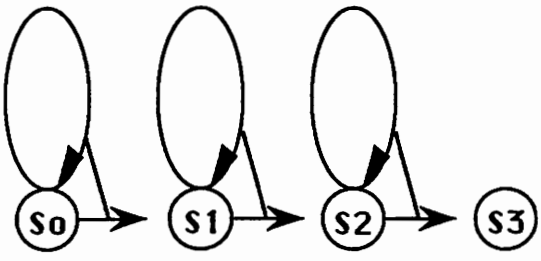


Figure 3: The architecture of HMM

These models were trained using several codebooks of different size (64,128,256) in order to check out the recognition rate as a function of the codebook size.

We initialized the transition probability to be equal to 0.5 and the initial values of the observation probabilities are set to be equal to the number of observation of one code divided by the total number of code. We used the Baum-Welch algorithm to train the model. We set the number of iteration at 7, using a logarithmic compression to prevent the probabilities of becoming zero.

4 Phoneme Recognition

The general procedure of the experiment are as follow:

1. Sample speech at 12 kHz
2. Compute a 256 FFT coefficients every 5ms (average by every two frames, so frames rate is to 10ms)

3. Compute the 16 melscale coefficients from the FFT coefficients
4. Normalize all the values between -1 and +1 among 15 frames
5. Pass the 15 frames of 16 melscales as input in the TDNN networks
6. Get the set of training and the recognition data from the TDNN output
7. Process the sequence of training vectors according to the phoneme label file
8. Train the HMM by the available data

One problem was the time spent at step 6 as it takes in the average one minute to scan all the phonemes contained in a word. The raw speech taken from the database (one male speaker) was first sampled at 12kHz, hamming windowed and 256 FFT coefficients were computed every 5ms. From this FFT coefficients, we computed the melscale spectrum coefficients and use 15 frames of them (one frame is equivalent to 16 melscale coefficients). We used the normalized values as input in the TDNN and we collected the output result in the form of a vector of 24 dimension. We then use one part of the stored vectors to produce the codebook, another part for training and the others for recognition. To process the sequence of vectors corresponding to phoneme we used hand segmented information of the database.

4.1 Speech data

The data was taken from the ATR database(MAU). That means a set of common japanese words, uttered in a sound-proof booth by a male professional announcer. We used 216 words to generate several codebooks of different sizes. The phoneme tokens (24 phoneme categories, one of them including the silence) consisted of 15 time frames of 16 melscale spectrum coefficients with a frame rate of 10ms.

The input speech of the TDNN was obtained from the speech wave by sampling at 12kHZ and using a hamming window to process a 256-point FFT every 5ms. So, one output vector of the TDNN is the result of 150ms of the speech wave.

4.2 Label generation

Building a new label file for our data is as follows (Fig.4). To process the right frames corresponding to our data, we used this formulas:

$$f_1 = ((b-a)/s) + w/2s) \quad (1)$$
$$f_2 = ((e-a)/s) + w/2s) \quad (2)$$

- f_1: beginning of the frame corresponding to the phoneme.

- f_2: end of the frame corresponding to the phoneme.
- b: beginning of the speech boundaries corresponding to the phoneme.
- e: end of the speech boundaries corresponding to the phonemes.
- a: end of the silence.
- w: size of TDNN window (150ms).
- s: window shift size (10ms).

We took these information into consideration:

1. 150ms of the speech is equivalent to one output token from the TDNN.
2. Tokens are extracted every 10ms.

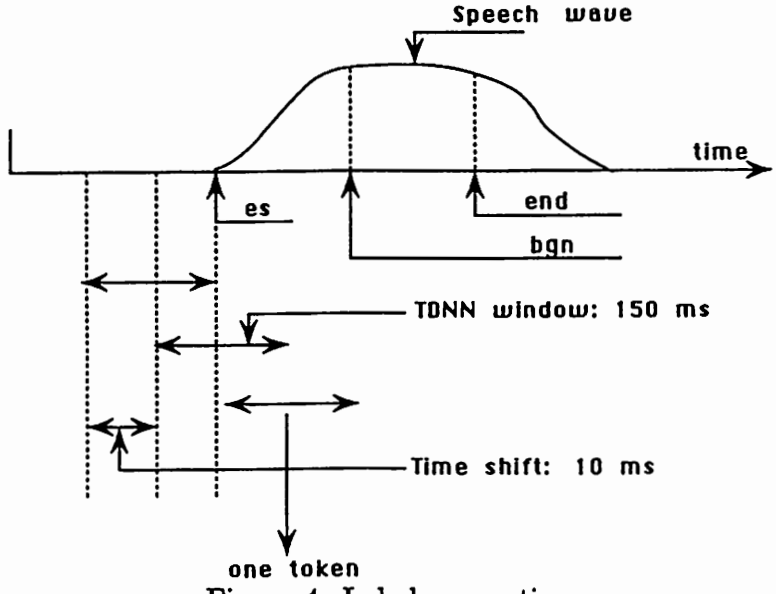


Figure 4: Label generation

An output vector of TDNN is equivalent to 150ms of speech wave. During the recognition, the window is shifted every 10ms and starts at the end of the silence. Our file data contained the vectors scanned from the TDNN output units. So,among these vectors we have to process the right sequence corresponding to the phoneme only by taking the raw speech into consideration. That leads to the formulas given above.

4.3 VQ Codebook Generation

We produced several codebooks of different size using the Euclidean distance:

$$D = \sum_{i=1}^{24} (v_i^1 - v_i^2)^2$$

v_i: the i-th element of the vector that corresponds to the activation of the i-th phoneme. This distance is equal to error in backpropagation during TDNN training. The codebook was processed from 216 common Japanese words taken from the database. The algorithm used to produce the codebook is close to the LBG (Linde, Buzo, Gray). It splits the training data into 2,4,..., 256 partitions, generating a centroid for each partition as the average of the vector in the partition.

Table 1: Number of VQ training samples

training words	216 words
training frames	18281 frames

We then produced several codebooks of different size. The decreasing distortions are shown in the Table 2 and Fig.5 .

Table 2: Decreasing VQ distortion

codebook size	distortion
2	0.7858
4	0.5379
8	0.4886
16	0.3550
32	0.2264
64	0.1675
128	0.1250
256	0.0971

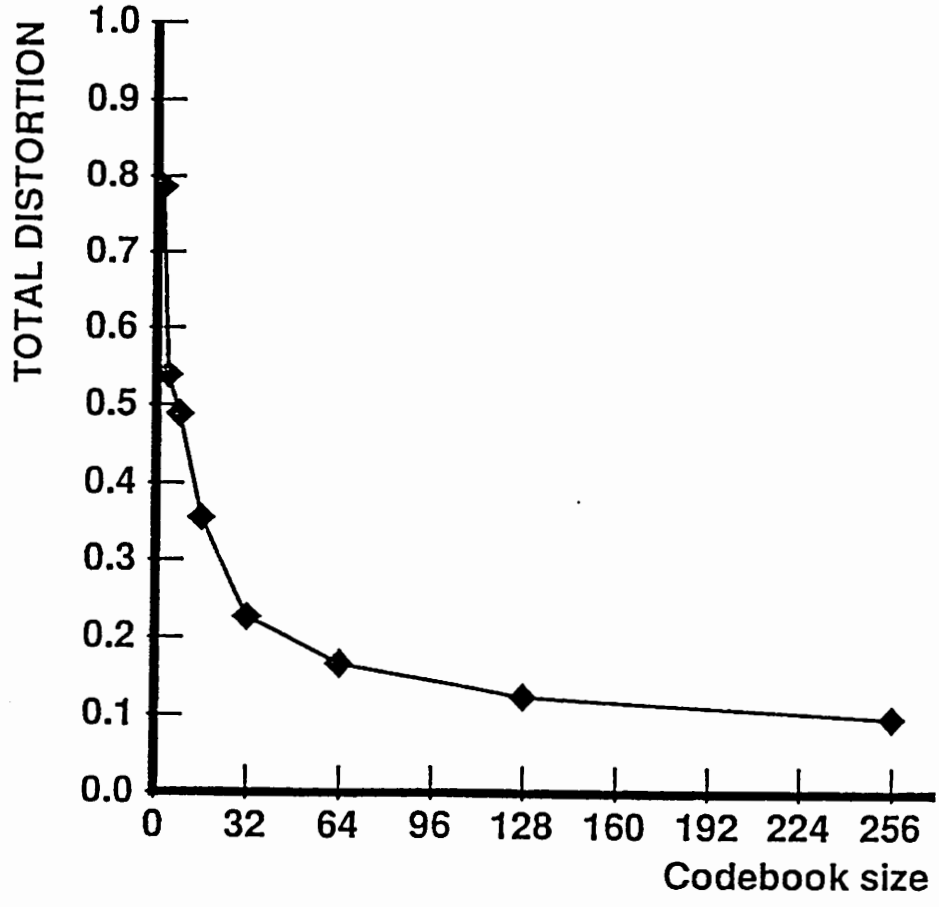


Figure 5: Decreasing VQ distortion

We first think that a codebook of size 32 will be sufficient as there will be enough vectors stored in this codebook representing one of the Japanese phoneme. But, if we examine the vectors stored [see the Fig.6, 7, 8], we could notice that the first vector stored correspond to the activation of the /m/. There is no stored vector corresponding to /d/, or /p/. We then go ahead in increasing the codebook size from 32 to 64. The codebook of size 64 containing the codes of all the Japanese Phoneme seemed sufficient. We also produced the codebook of size 128 and 256.

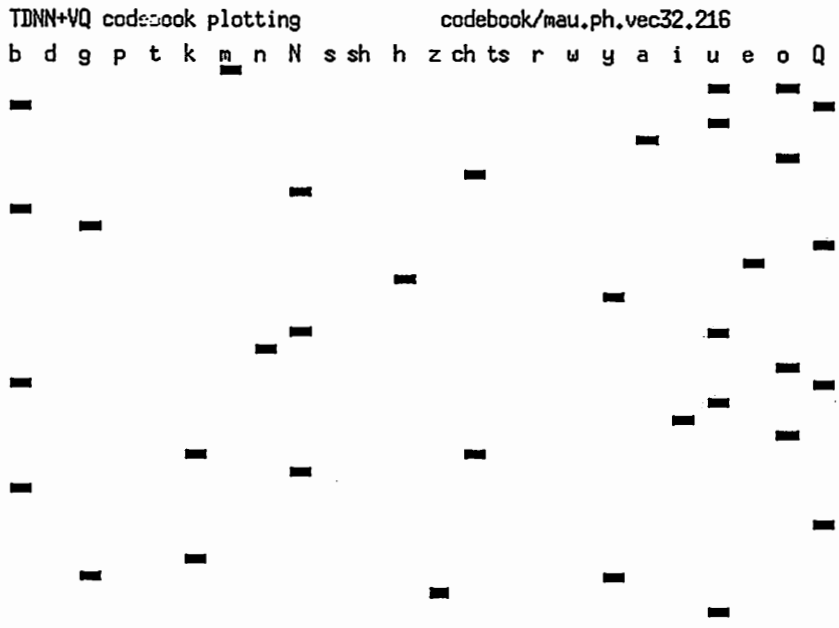


Figure 6: Values of VQ codes (32 codes)

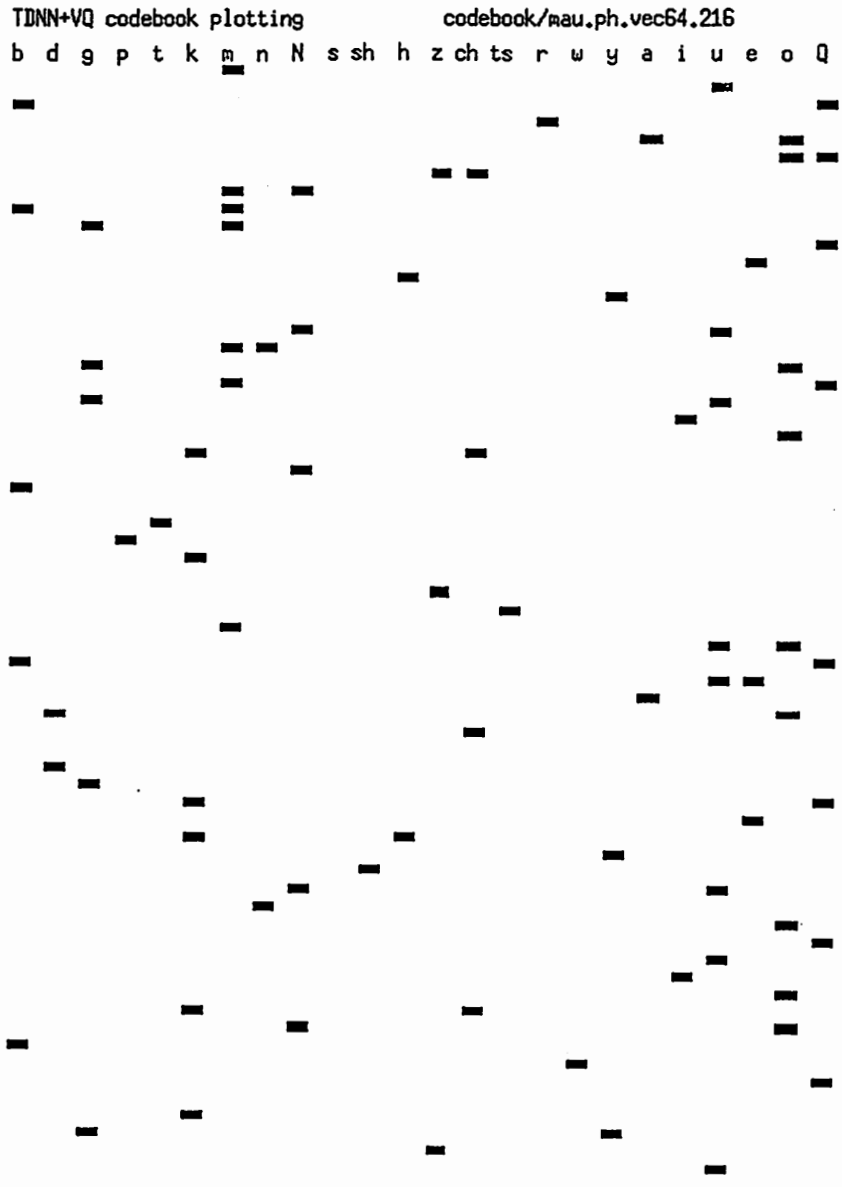


Figure 7: Values of VQ codes (64 codes)

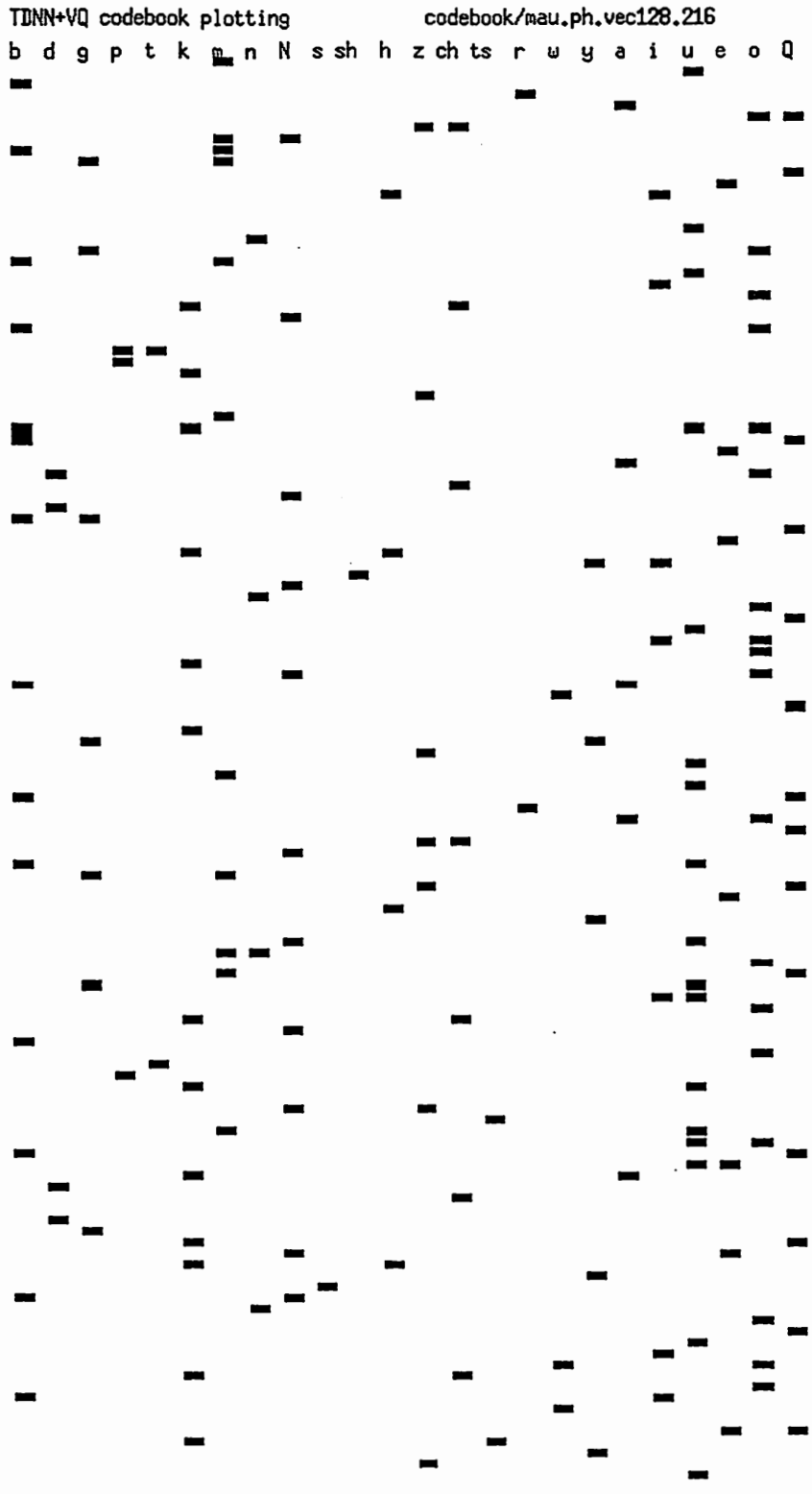


Figure 8: Values of VQ codes (128 codes)

4.4 Preliminary experiment on the /b/, /d/, /g/ task

We first trained the system on the recognition of /b/, /d/, /g/ phonemes. We collected from the database all the Japanese words containing one of these phonemes. The work was done on a Dec station 3100 and we had to care about the available diskspace. Therefore, we could not collect all the Japanese words together and trained the system for the recognition of all the phoneme at the same time. It took about 20 hours to collect all the data from the TDNN output, but the training of the Hidden Markov Model is relatively faster. For each phoneme, we divided the number of samples into two parts. One part for the training (even number) and another part for the recognition (odd number). Table 3 and the Table 4 show the number of data used both for training and for recognition.

Table 3: Number of training data

Phoneme	number of tokens	number of frames
b	227	1632
d	202	1475
g	259	1877

Table 4: Recognition Data

Phoneme	number of tokens	number of frames
b	218	1558
d	179	1330
g	251	1785

The two sets of data are on the average acoustically identical as we picked up sequentially one word for the training an the next for recognition from the database. We trained the model using the Fuzzy-VQ (FZVQ)algorithm and the Hard-VQ(HDVQ). For the FZVQ, we choose the number 1.6 as the fuzziness value, setting the number of fuzzy at 6. Different codebooks were used to train the models. The results are shown in Table 5 and 6 for different codebooks of size 64, 128, and 256.

Table 5: Recognition rate using Fuzzy-VQ

Iterations = 7
Fuzziness = 1.6
Number of fuzzy = 6.

codebook size	b	d	g	average rate
64	98.6	96.6	95.6	96.9
128	98.2	97.2	95.6	97.0
256	98.6	97.2	96.0	97.2

Table 6: Recognition rate using Hard VQ

codebook size	b	d	g	average rate
64	97.7	96.6	95.6	96.6
128	98.2	97.2	96.0	97.1
256	97.7	96.6	96.8	97.0

4.4.1 VQ distortion and Recognition Rate

According to the algorithm used (HDVQ or FZVQ), the performance are not the same. The lowest average distortion was processed in the codebook of size 256. But, we could notice that the average recognition rate, when using HDVQ happened to on the codebook of size 128, not on the codebook of size 256. It is the reverse when using FZVQ with the higher recognition rate (97.27%).

These results are satisfying and show the effective cooperation of the two methods. It is noticed that the error function in the back propagation algorithm and the distance in TDNN-VQ are the same the Euclidean distance. Table 7 shows the comparison with a standard HMM.

Table 7: Comparison of recognition rate

HDVQ: Hard-VQ
FZVQ: Fuzzy-VQ

method	b	d	g	average rate
HDVQ-256	97.7	96.6	96.8	97.0
FZVQ-256	98.6	97.2	96.0	97.2
HMM only	95.6	97.2	94.4	95.6
TDNN	98.2	98.3	99.6	98.7

The standard HMM was trained using 3 codebooks with different parameters: the first for the WLR of size 256, the second for power with size 64, and the third for Δcepstral with size 256.

4.5 Recognition of all Japanese Phonemes

After the first experiment on the recognition of /b/, /d/, /g/, encouraged by the result, we went ahead in applying the system to additional consonant clusters: the unvoiced stop /p/, /t/, /k/; the nasals /m/, /n/, /N/; the fricatives /s/, /sh/, /h/, /z/; the affricatives /ch/, /ts/; the liquid and glides /r/, /w/, /y/; and the vowels /a/, /i/, /u/, /e/, /o/. This is almost the entire phoneme set for Japanese.

Each of the phoneme cluster was treated the same way as the Phonemes /b/, /d/, /g/. The recognition rate are shown in the Table 8 and Table 9 below. The number of samples used for training and for recognition are not the same.We used more training data than recognition.

Iterations	7
Fuzziness	1.6
Number of fuzzy	6

Table 8: Recognition result for all the phonemes using Fuzzy VQ

phoneme	TDNN-HMM			HMM	
	errors /token	recognition rate(%)	average rate(%)	recognition rate(%)	average rate(%)
b	8/218	96.3		88.5	
d	9/179	95.0	92.6	97.2	88.3
g	34/251	86.5		79.4	
p	3/33	90.9		60.0	
t	29/400	92.8	94.3	94.5	81.9
k	3/400	99.3		91.4	
m	6/400	98.5		75.5	
n	0/260	100.0	99.5	86.4	85.9
N	0/151	100.0		92.2	
s	0/400	100.0		95.2	
sh	0/310	100.0	100.0	99.4	88.3
h	0/214	100.0		91.3	
z	0/116	100.0		67.4	
ch	1/134	99.3	99.4	97.2	97.2
ts	1/212	99.5		97.2	
r	130/400	67.5		78.8	
w	0/72	100.0	88.9	98.7	92.3
y	1/159	99.4		99.4	
a	1/400	99.8		98.2	
i	4/122	96.7		84.2	
u	7/223	96.2	98.4	72.7	89.4
e	4/400	99.0		94.7	
o	1/400	99.8		97.2	

Table 9: Recognition result for phoneme categories using Fuzzy VQ

phoneme	TDNN-HMM			TDNN
	errors /token	recognition rate(%)	average rate(%)	average rate(%))
b	3/218	98.6		
d	5/179	97.2	97.2	98.6
g	10/251	96.0		
p	2/33	93.9		
t	24/400	94.0	95.8	98.7
k	2/400	99.5		
m	3/400	99.3		
n	0/260	100.0	99.7	96.6
N	0/151	100.0		
s	0/400	100.0		
sh	0/310	100.0	100.0	99.3
h	0/214	100.0		
z	0/116	100.0		
ch	0/134	100.0	99.7	100.0
ts	1/212	99.5		
r	120/400	70.0		
w	0/72	100.0	90.0	99.9
y	0/159	100.0		
a	1/400	99.8		
i	1/122	99.2		
u	4/223	98.2	99.3	98.6
e	2/400	99.5		
o	0/400	100.0		

We set the number of sample to be below 400, because the software used did not enable us to use larger number of samples. The recognition result using hard VQ HMM is shown in Table 10. This results are little bit worse than the results using FZVQ HMM.

Table 10: Recognition rate of all the phoneme using Hard VQ

phoneme	TDNN-HMM			TDNN
	errors /token	recognition rate(%)	average rate(%)	average rate(%)
b	8/218	96.3		
d	9/179	95.0	92.7	98.6
g	33/251	86.9		
p	5/31	84.8		
t	30/400	92.5	92.2	98.7
k	3/400	99.3		
m	11/400	97.3		
n	0/260	100.	98.6	96.6
N	2/151	98.7		
s	0/400	100.0		
sh	0/310	100.0	99.4	99.3
h	1/214	99.5		
z	2/116	98.3		
ch	1/134	99.3	99.4	100.0
ts	1/212	99.5		
r	132/400	67.0		
w	0/72	100.0	99.9	99.9
y	4/159	100.0		
a	1/400	99.8		
i	3/122	97.5		
u	8/223	96.4	98.5	98.6
e	4/400	99.0		
o	1/400	99.8		

Table 11: Comparison of four recognition methods

TDNN-HMM-HDVQ	TDNN-HMM-FZVQ	HMM	TDNN
95.6	96.1	89.0	98.6

Table 11 shows the comparison of TDNN-HMM-HDVQ, TDNN-HMM-FZVQ, HMM and TDNN. New methods provide the higher recognition rate than the conventional HMM by 7.1%, and just below the TDNN for 2.5%. Taking on account the ability of HMM for continuous speech

recognition, this enables us to use the discrimination power of the TDNN in continuous speech recognition.

4.6 Discussion

One of our interests in this report is the choice of the best parameters to use in training HMM. It seems obvious to used time derivatives in order to take in account the high variability of speech in time axis. This experiment shows that the performance could depend on the quality of the features. We manage to represent one speech unit (a phoneme) by a vector. Of course, the performance still depend on the codebook size. We decided here to used the conventional VQ algorithm so as to prove its efficiency. We could have also, store in a codebook, some pre-chosen vectors and used them in HMM task. In this way, the choice of the vectors is easy as we could just choose the output corresponding to the phoneme and store 24 of them.

Another question could be about the capacity of the HMM to correct a high sequences of errors occurring from the TDNN. We did not face this case in our experiment as the TDNN was well trained for better performance. Let us suggest that the performance of this method is intended to take the advantages of both TDNN and neural networks.

A decisive idea will be to train the HMM using the FFT coefficients as data (the TDNN input coefficients), in order to check out the part of the TDNN in the process. So, further research can be done in this field.

5 Conclusion

The combination of Neural Networks and Hidden Markov Models in the way to take the advantage of both Model for speech recognition seems to have a promising avenue. The powerful learning algorithm used in Hidden Markov modeling added to the good quality as classifier of the neural networks could be used together in order to increase the speech recognition performance. As we have shown, several researchers are now working on various ways to combine Hidden Markov Modeling and Neural Nets. We proposed a simple method for the cooperation of the both methods that was to use a TDNN codebook to train discrete Hidden Markov Models.

6 Future works

Others approaches can be attempted.Here are several proposals to go ahead in this work.

- Train the system with phonemes taken from phrases
- TDNN-HMM-LR system to continuous speech
- Build a codebook from the second Hidden layer of the TDNN

- Try the system on Fuzzy Trained Neural Network
- Evaluation of speaker independency of HMM part.

ACKNOWLEDGEMENT

We specially want to thank the following people of the speech processing department of ATR Interpreting Telephony Research Laboratories for their help and their comprehension about this research: Keiji Fukuzawa , Hiroaki Hattori, Hidefumi Sawai, Jun,Ichi Takami for good explanation on the sofwares. David Rainton for providing useful discussions and explanations, E.Mc Dermott and S. Katagiri of ATR Visual Perception and Auditory Research Laboratories for valuable advice.

References

[1] N.Morgan and H.Bourland, Continuous Speech Recognition using Multilayer Perceptrons with Hidden Markov Models, Proc. ICASSP90, S8.1, pp.413-416 (Apr. 1990).

[2] Les T. Niles and H.F. Silverman, Combining Hidden Markov Model and Neural Network Classifiers, Proc. ICASSP90, S8.2, pp.417-420 (Apr. 1990).

[3] Esther Levin, Word Recognition using Hidden Control Neural Architecture, Proc. ICASSP90, S8.6, pp.433-436 (Apr. 1990).

[4] J.S.Bridle, Alpha-Nets: A recurrent "neural" network architecture with a Hidden Markov Model interpretation, RSRE Research Report (Oct. 1989).

[5] S.J.Young, Competitive Training: A Connectionnist Approach to the Discriminative Training of Hidden Markov Models, CUED/ F-INFENG/ TR.41 (Mar. 1990).

[6] E.Tsuboka, Neural Network Driven Hidden Markov Model, SP89-83 (1989).

[7] A.Waibel & H.Sawai, Integrated Training for Spotting Japanese Phonemes Using Large Phonemic Time-Delay Neural Networks, ICASSP 90.

[8] J.Tebelskis & A.Waibel, Large Vocabulary Recognition using linked predictive Neural Networks, IEEE Magazine (1990).

[9] L.Rabiner, Tutorial on Hidden Markov Modelling, IEEE Magazine (Feb. 1989).

[10] Weiye and Compornelle, TDNN Labeling for HMM Recognizer, IEEE Magazine (1990).

[11] R.Lipmann, Review of Neural Networks for Speech Recognition.

[12] R.Lipmann, An Introducing to computing with Neural Nets, IEEE Magazine (Apr. 1987).

A Regular Talk

Study on Combination of HMM and Neural Network Model

Alain BIEM & Masahide SUGIYAMA

ATR Interpreting Telephony Research Laboratories

Sanpeidani, Inuidani, Seika-cho, Soraku-gun,
Kyoto 619-02, Japan

historicnl_review.tex

Overview of the work on this Field

- N.Morgan & Boulard *(ICASSP90, S8, PP.413-416, 1990-04)*
  Suggestion: Neural Nets outputs could be considered as the estimation of the maximum a posteriori probabilities for Hmm.

- T.Niles and Harvey F.Silverman*(ICASSP90, S8.2, pp.417-420, 1990-04)*
  Suggestion: Build a Neural Net to process the forward backward algorithm

- Esther Levin*(ICASSP90, S8.6, pp.433-436, 1990-04)*
  Suggestion: Implementation of a non-linear predictive network where the input is separated into two part:
  - The input speech x
  - The control input c driven by the noise

- J.S.Bridle*(RSRE Research Report, Oct-1989)*
  Suggestion: Alpha calculation viewed as a reccurent neural net and a new error criterion is proposed to apply back propagation

- S.J.Young*(CUED/F-INFENG/ TR.41, 1990-03)*
  Suggestion: HMM is viewed as a connectionnist model where error back propagation could be used to train HMM.

- S.Katagiri & H. Iwamida & Y.Tohkura &E.Mc Dermott *(Reading in SR, pp425, 1990)*
  Suggestion: Use LVQ generated codebook to train HMM.

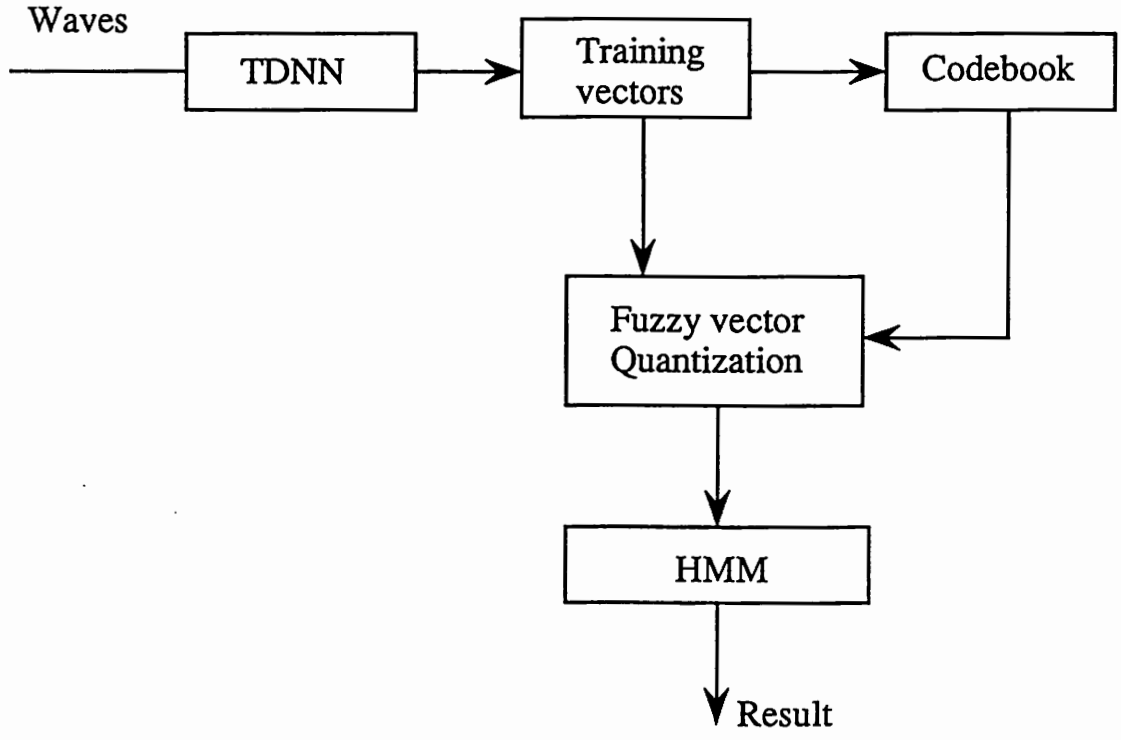
- E.Tsuboka*(SP89-03, 1989)*
  Suggestion: Aim to represent time variant phenomena in each state of an HMM by using Neural nets.

TDNN-HMM for Phoneme Recognition

Alain BIEM & Masahide SUGIYAMA

ATR Interpreting Telephony Research Laboratories

Sanpeidani, Inuidani, Seika-cho, Soraku-gun,
Kyoto 619-02, Japan



Architecture

experiment_step.tex

Experiment steps

- Sample speech at 12khz
- Compute a 256 FFT coefficients every 5ms
- Compute melscale coefficients from the FFT coefficients
- Normalise all the values between -1 and +1
- Pass the melscale coefficients as input in the TDNN.
- Get the set of training and recognition Data from the output units of the TDNN.
- Process the sequence of training vectors from the label files
- Train the HMM with the available Data

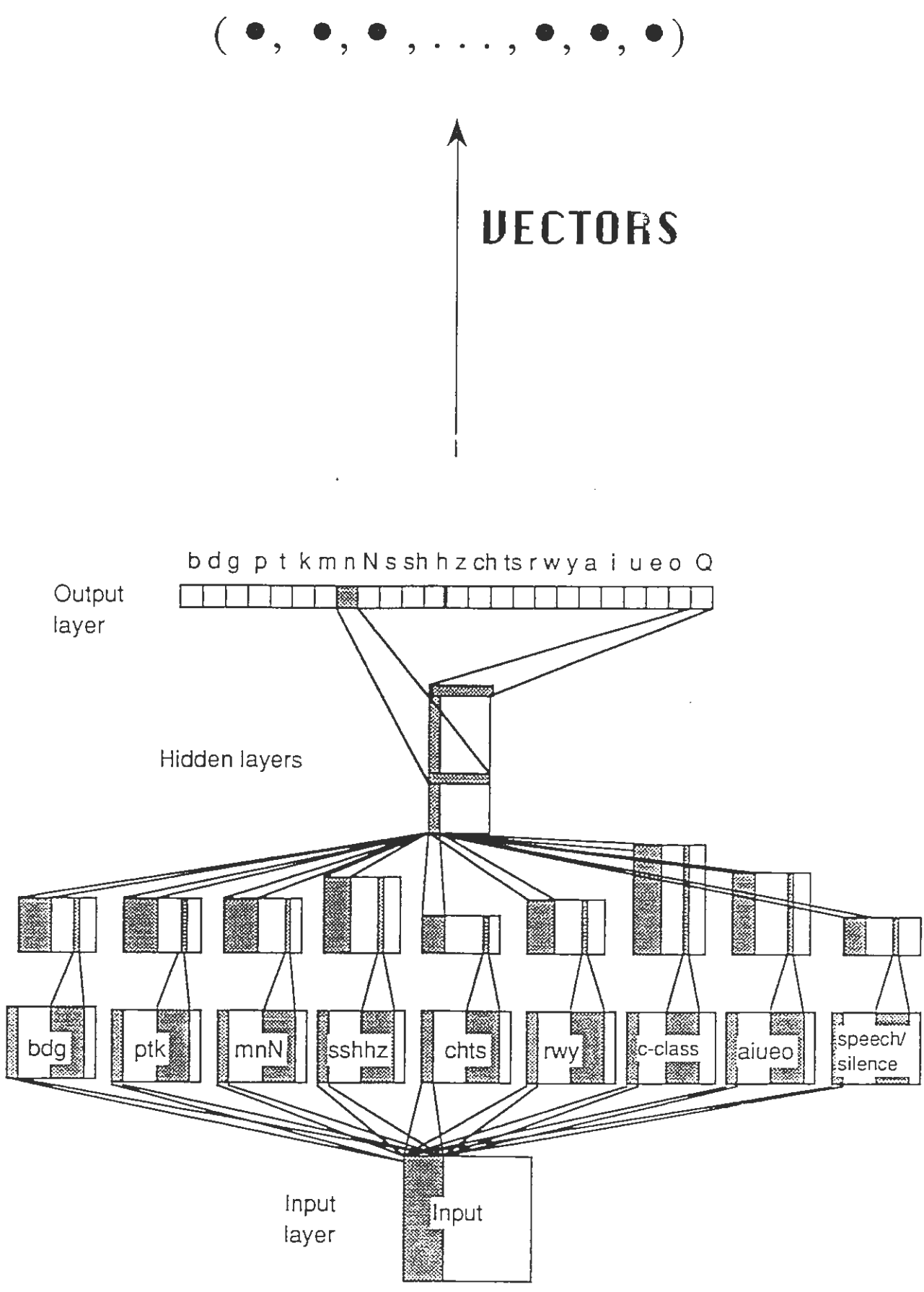
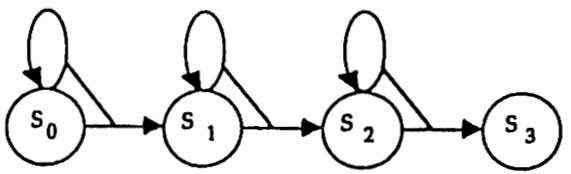


Fig. 1 A Large Phonemic TDNN Architecture

PHONE MODELs

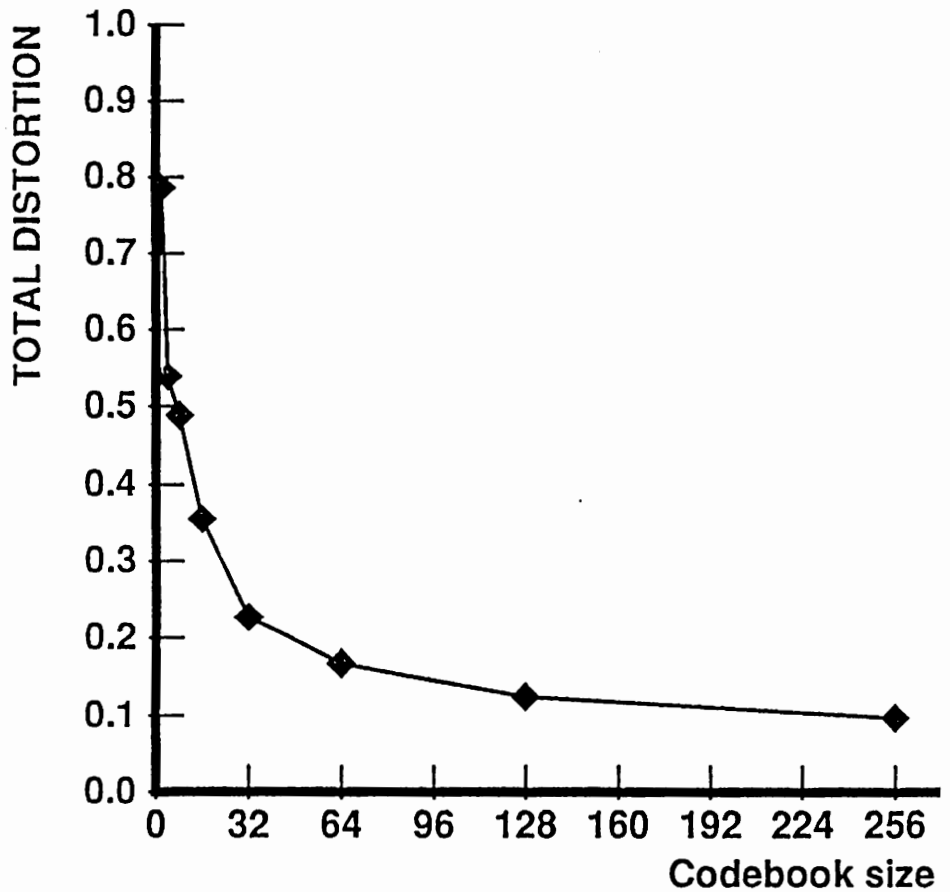
- Left-to-right discrete HMMs



- One Codebook
- Euclidian Distance

Table 2. The relationship between VQ distortion and codebook size

codebook size	VQ distortion
2	0.7858
4	0.5379
8	0.4886
16	0.3550
32	0.2264
64	0.1675
128	0.1250
256	0.0971


TOTAL DISTORTION
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
32
64
96
128
160
192
224
256
Codebook size

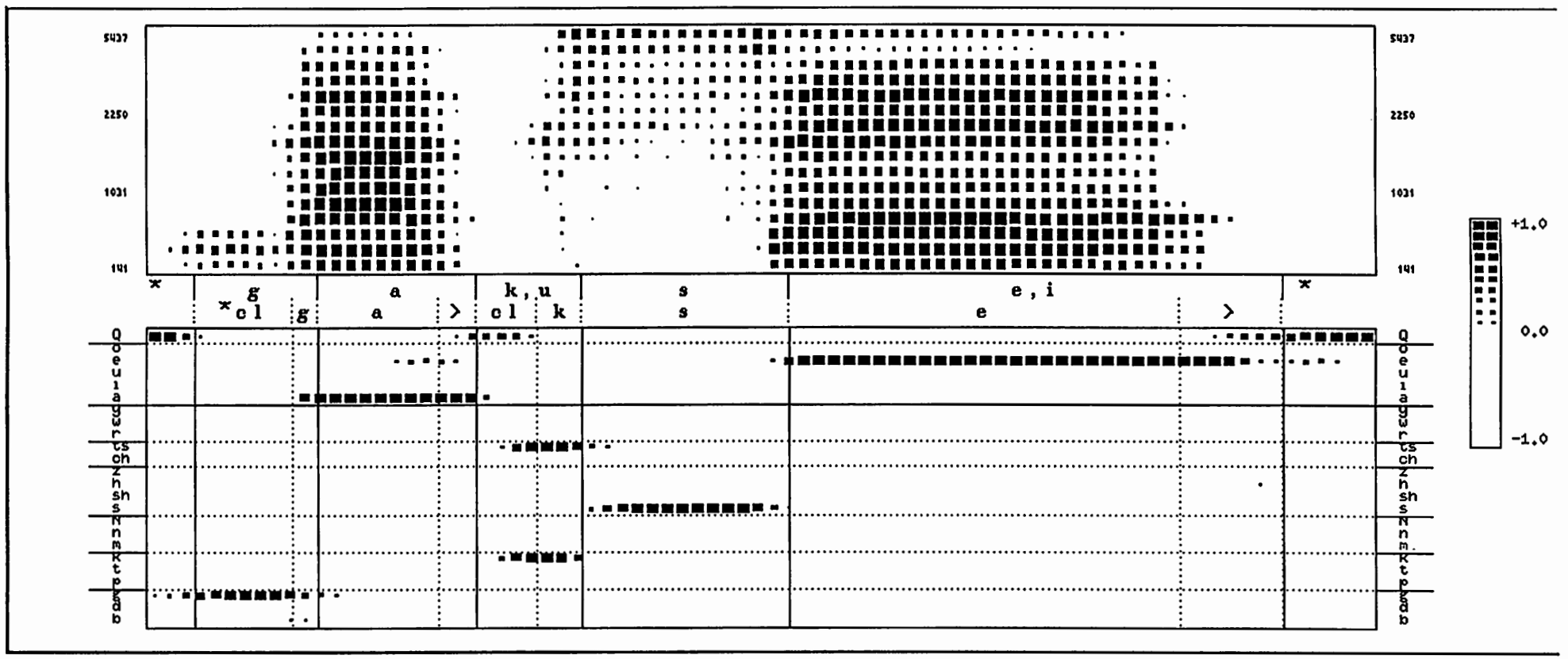

5437
2250
1031
141
+1.0
0.0
-1.0
x
g
cl
g
a
a
>
k, u
cl
k
s
s
e, i
e
>
x
Oberlippe
Zunge
Nasal
Velum
Pharynx

Table 4. Recognition rate of /b/,/d/, /g/ using Hard VQ

codebook size	b	d	g	average rate
64	97.7	96.6	95.6	96.6
128	98.2	97.2	96.0	97.1
256	97.7	96.6	96.8	97.0

Table 5. Recognition rate of /b/,/d/, /g/ using Fuzzy VQ

Iterations	7
Fuzziness	1.6
Number of fuzzy	6

codebook size	b	d	g	average rate
64	98.6	96.6	95.6	96.9
128	98.2	97.2	95.6	97.0
256	98.6	97.2	96.0	97.2

Table 4. Comparison of recognition rates

method	b	d	g	average rate
HDVQ-256	97.7	96.6	96.8	97.0
FZVQ-256	98.6	97.2	96.0	97.2
HMM only	95.6	97.2	94.4	95.6
TDNN	98.2	98.3	99.6	98.7

- HDVQ: Hard VQ
- FZVQ: Fuzzy VQ

Standart HMM

- Four States with Loops in the first three states

- 3 Codebooks :
  -Size 256 for WLR
  -Size 64 for Power
  -Size 256 for Δcep.

Table 3. Experiment condition

Training Data

Phoneme	Number of samples	Number of frames
b	227	1632
d	202	1475
g	259	1877

Recognition Data

Phoneme	Number of samples	Number of frames
b	218	1558
d	179	1330
g	251	1785