Inter-Ing 2007 UNIVERSITATEA Petru Maior TÂRGU-MUREŞ

„INTERDISCIPLINARITY IN ENGINEERING"
SCIENTIFIC INTERNATIONAL CONFERENCE,
TG. MUREŞ – ROMÂNIA, 15 -16 November 2007.

FRAME-BY-FRAME PHONEME CLASSIFICATION USING MLP

DOMOKOS JÓZSEF, SAPIENTIA UNIVERSITY
TODEREAN GAVRIL, TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

Key words: MFCC, delta, double delta, MLP, OASIS Numbers, continuous speech recognition;

Abstract: In this paper, we present some practical experiments for continuous speech frame-by-frame phoneme classification using Multi Layer Perceptron (MLP) neural networks. We used to train and test our software application, the the OASIS Numbers speech database. In our experiments, we tried to classify all the existing 32 phonemes together, from OASIS Numbers database dictionary. We also used different MLP configurations to compare the achieved results. For classification, we used 13 MFCC coefficients and their first and second order derivatives (delta parameters) extracted from speech signal using our Matlab based feature extractor software application.

I. INTRODUCTION

A standard statistical speech recognition system is based on Hidden Markov Models (HMM's). In fact the entire system is constructed using multiple HMM's (for acoustic modeling and lingvistic modeling) linked together. Mathematically such a system can be described as fallows: given a set $A = \{a_1, a_2, ..., a_n\}$ of acoustic vectors after the feature extraction stage, we are searching for the most probable word sequence $W^* = \{w_1, w_2, ..., w_m\}$.

$$W^* = \arg\max_{w}\{P(W \mid A)\} \qquad \textbf{Eq. 1}$$

The above ecuation can be transformed using Bayes rule as fallows:

$$W^* = \arg\max_{w}\{P(A \mid W) \cdot P(W)\} \qquad \textbf{Eq. 2}$$

In the above ecuation the probability P(A|W) represents the acoustic model and P(W) represents the language model part of the system [2][7][8]. We try to examinate in this paper an alternative way for acoustic modeling instead of Hidden Markov Models, one which is based on neural networks.

Neural networks have been successfully applied to pattern recognition in the past years. Most conventional neural networks used in pattern recognition belong to classification type, that is, a pattern is used as an input and a category symbol given as output target. These networks can classify input patterns by complex nonlinear decision surfaces. The only thing they need is a large training set consisting in correct input and output target pairs.

Such a problem is also the phoneme classification. In this case the input consists in one or more feature vectors extracted from speech signal. Most commonly used features are the MFCC (Mel Frequency Cepstral Coefficients) and the PLPC (Perceptual Linear Prediction Coefficients) [7] [8] [13]. The outputs correspond to each phoneme needed to be recognized. In this way just one output needs to be active (its value equals to one, or close to one), the one who represents the recognized phoneme. The other outputs need to be equal to zero or close to zero to represent likelihood values.

Usually the number of input features are 13x3 coefficients (MFCC, delta and double delta features), and the number of outputs is between 31 – 61 (depending on number of phonemes from a language or a database)

II.SOFTWARE APPLICATION ARCHITECTURE

Our software application is built from two different parts:

- speech preprocessor, feature extractor and feature vector generator;
- MLP builder, trainer and tester;

We developed our software applications considering the Matlab platform, and we use in specially the Signal Processing Toolbox and Neural Networks Toolbox [4].

III. FEATURE EXTRACTOR

Our feature extractor calculates MFC coefficients with their delta and double delta parameters. The steps for extracting these features are as follows. Before computing the MFCC parameters, we make a preemphasis of high frequencies by filtering the speech signal with a first order FIR (Finite Impulse Filter) filter [2].

$$H(z) = 1 - az^{-1}, a = 0.95 \qquad \textbf{Eq. 3}$$

In the second step, a Graphic User Interface let us set the feature extraction parameters: frame length, window types and MFCC coefficients number. For this experiment the speech signal was windowed using a 256 sample length Hamming window with 15% overlap. Next we perform a Discrete Fourier Transform (DFT) analysis and a mel-scale based filtering, using a combination of two computationally inexpensive methods presented in [2] and [7]. The filters frequency domain responses are simply shifted and frequency warped versions of a triangular window presented in Fig. 1.

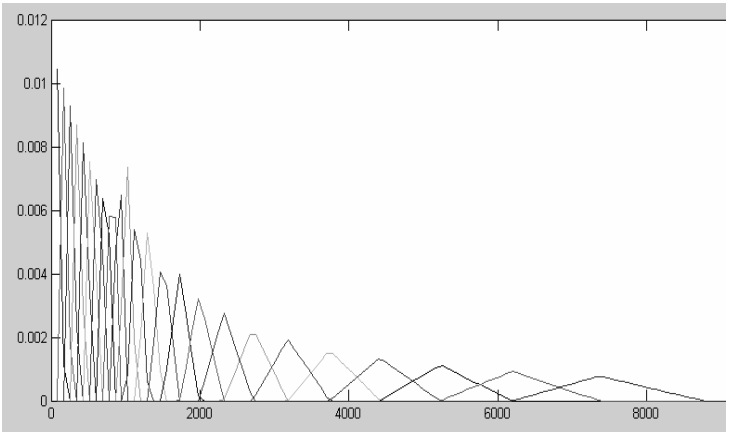

Fig. 1.Mel-scale based filters frequency domain responses

Filter m is given by Eq. 4, where f[m] is the central frequency of filter m [7]. The first ten filters are equally spaced from 0 to 1 kHz frequency range and the next 14 follows the eq. 5 [2] where B[m] is the filter m bandwidth, such we have totally 24 filters depicted in Fig. 1.

$$H_m[k]=\begin{cases}0, k<f[m-1] \\ \dfrac{2(k-f[m-1])}{(f[m+1]-f[m-1])(f[m]-f[m-1])}, f[m-1]\le k\le f[m] \\ \dfrac{2(f[m+1]-k)}{(f[m+1]-f[m-1])(f[m+1]-f[m])}, f[m]\le k\le f[m+1] \\ 0, k>f[m+1]\end{cases} \quad \textbf{Eq. 4}$$

$$B[m]=1.2\cdot B[m-1] \quad \textbf{Eq. 5}$$

The last two steps to achieve the MFCC coefficients are log energy computation and Inverse Discrete Cosine Transform (IDCT).

To take into account the dynamic evolution of speech signal, we compute the first and second order derivatives of MFCC coefficients, also called delta (Δ) and double delta (ΔΔ) parameters. The derivatives are computed as time differences, like presented in [2] and [7], using Eq. 6 and 7.

$$\Delta[i]=MFCC[i+1]-MFCC[i-1] \quad \textbf{Eq. 6}$$

$$\Delta\Delta[i]=\Delta[i+1]-\Delta[i-1] \quad \textbf{Eq. 7}$$

The complete architecture of feature extractor module is presented in Fig. 2.

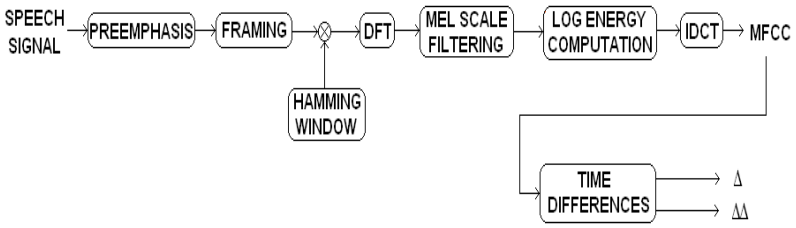

Fig. 2. The feature extractor module

Using the GUI, we can load manually one by one the speech files to perform feature extraction, or we can load a text file with a list of speech file paths and the application will process them all. The extracted features are finally saved in text file format with the same name and path as the original speech file, and with .fea extension, and also they are exported in .mat format for further usage.

IV. MLP TRAINING AND TESTING

We have studied different architectures and reach the conclusion that different phoneme classes can be recognized in a relatively easy way using a set of 13x3 features [5][7] and a three or four layered feed-forward neural network architecture [3]. The state of the art phoneme classification systems use recurrent neural networks with around 100.000 – 200.000 weights [6].

In our experiment we want to classify the 32 phones from OASIS dictionary file together, and the results are good enough just if we use a great MLP architecture with one hidden layer consisting of 512 perceptrons. Our tested architecture was: 351x512x32. The networks is presented in Fig. 3.

The MLP has 351 = 9x39 inputs because we use contextual information [3]. That means that for one frame we introduce the surrounding 8 frames (4 frames before and 4 frames after the current frame) in MLP input layer. We use feed-forward networks with tansig transfer function in the hidden layers and logsig transfer function in the output layer. We trained the MLP over 450 training epochs using trains - sequential order incremental training

function [5] and we used overtraining criterion to stop the training process [4]. This training method gives the best results and the fastest training times for pattern recognition problems where there is a lot of training data [4]. However system training takes about four weeks on a 3Gz PC with 1GB DDR.

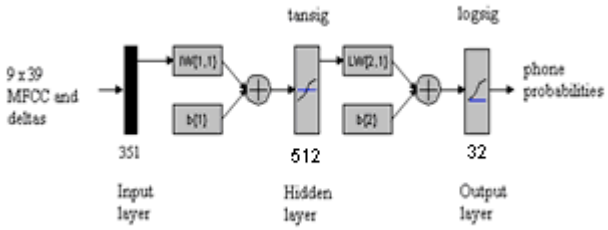

Fig. 3. 351X512X32 MLP architecture

V. EXPERIMENTAL RESULTS

In the experiment we try to use the OASIS Numbers database [9]. This database was developed at University of Szeged, by Artificial Intelligence Research Group for training and testing speaker independent number recognition systems. The database contains 26 short numbers and 20 long numbers each of them uttered two times by 66 speakers. All the short utterances are manually segmented and annotated. These utterances can be used to train the system. The train and test sets were the recommended ones. The phoneme set consists of 32 phonemes marked using SAMPA standard.

The results presented in the following tables were good enough to take the conclusion, that our phoneme classifier can be used as part of a further developed hybrid continuous speech recognition system.

Table 3: Classification results on OASIS Numbers database

Train epochs	*Test percentage[%]*
11	67.74
50	74.06
100	78.22
150	79.69
200	81.63
250	82.93
300	83.48
350	83.77
450	84.25

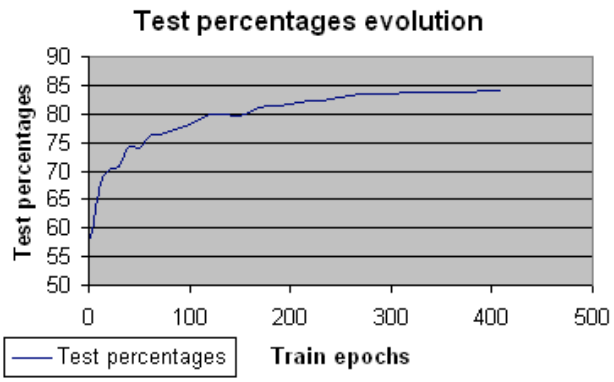

Fig. 4. Test percentage evolution on OASIS Numbers database

Table 4: Phoneme classification results on OASIS Numbers

Phoneme	*Total nr. of*	*Recognition [%]*	*Misrecognized frames*
E	2625	79,58	536
+	312	64,42	111
d'	292	67,12	96
z	765	75,82	185
r	322	53,42	150
h	918	82,79	158
A:	953	46,27	512
o	561	74,69	142
m	259	55,6	115
O	1053	77,21	240
i	592	68,58	186
n	1324	50,68	653
-	1252	72,44	345
ts	680	74,71	172
t	525	66,67	175
v	492	62,6	184
e:	913	65,5	315
u:	293	90,44	28
s	738	77,1	169
u	236	80,08	47
k	295	85,42	43
-:	237	18,57	193
2:	228	70,61	67
l	433	24,94	325
o:	199	62,81	74
l:	147	43,54	83
J	125	43,2	71
j	128	64,84	45
2	345	75,36	85
i:	329	91,49	28
~	19192	98,47	293
X	0	0	0

VI. CONCLUSIONS AND FURTHER DEVELOPEMENTS

Our results, encourage us to use this classifier software as phonetic modeling part of further continuous speech recognition systems. There are many hybrid artificial neural network (ANN) – HMM (Hidden Markov Model) approaches for continuous speech recognition which provide very good results [3], [8]. Our final scope is to develop, considering the HTK Toolkit and our phoneme recognizer, a continuous speech recognition system.

In comparison to a Gaussian Mixtures Model based phoneme classification method which result were a recognition rate of 89,51% on the same database [1], our model provides fairly good results.

In the future we want to calculate deletion, insertion and substitution errors for each phoneme.

Other works, like [6], [10], [11] and [12] use a reduced phoneme set or groups to achieve better results on phoneme recognition. We intend to follow this simplification within our tests to increase the recognition rates.

The tests made on the OASIS Number database shows us that the application performs well on small databases. We want to try our system on some bigger databases like TIMIT database.

In the future we'll try to use recurrent neural networks, to achieve better results in phoneme classification and recognition [6] [10] [11] [12].

VII. ACKNOWLEDGEMENTS

Research and conclusions of this paper were achieved as part of Sapientia - Research Programs Institute sponsored Phd. grant nr. 8/2006 - 2007.

VIII. REFERENCES

1. ANTAL M., Phoneme recognition for ASR, Proceedings of the 6th International Conference COMMUNICATIONS, 2006, Bucharest, Romania, pp. 123-126.
2. BECCHETTI, C., RICOTTI, L. P., Speech recognition. Theory and C++ implementations. J. W. & sons, 1999.
3. BOURLARD, H, MORGAN, N., Connectionist speech recognition, Kluiwert Academic Publishers, 1994.
4. DEMUTH H., BEALE M., Neural Network Toolbox. For Use with MATLAB®. Math Work Inc, 2005.
5. DOMOKOS J., Phoneme classification using MLP, research raport, Sapientia - Research Programs Institute, 2006.
6. GRAVES, A., SCHMIDTHUBER, J., Framewise phoneme classification with bidirectional LSTM networks, Proceedings of IEEE International Joint Conference on Neural Networks, 2005, pp. 2047- 2052, vol. 4.
7. HUANG X., ACERO A., HON H. Spoken Language Processing, Prentice Hall, 2001.
8. JURAFSKY D., MARTIN H. J., Speech and language processing. Prentice Hall, 2000.
9. MTA-SZTE, Mesterséges Intelligencia Tanszéki Kutatócsoport, OASIS Numbers adatbázis, 2002.
10. ROBINSON, T., Phoneme recognition from the TIMIT database using recurrent error propagation networks, Technical Report, Cambridge University, 1990.
11. ROBINSON, T., Several improvements to a recurrent error propagation network phone recognition system, Technical Report, Cambridge University, 1991.
12. ROBINSON, T., Recurrent nets for phone probability estimation, Proceedings of ARPA Continuous Speech Recognition Workshop, 1992.
13. TODEREAN G., CĂRUNTU A., Metode de recunoaştere a vorbirii, Editura Risoprint 2005.