Temporal Processing for Speech Recognition

Codrescu Cristinel
Department of Scientific Computing
University of Salzburg
Salzburg, Austria
E-mail: ccodres@cosy.sbg.ac.at

Abstract— **We have developed a java software application for speech recognition. This software offers the possibility to create different temporal processing neural network types based on FIR-MLP architecture. The FIR-synapse types we are using have been defined by means of differential operators [1]. These neural network architectures have been used for the task of speech recognition of nine german digits. The learning time and the recognition rates for one and for more speakers have been compared. The networks training was done by using temporal backpropagation as learning algorithm.**

I. Introduction

Temporal information plays an important role in speech recognition. Some available and often used methods to incorporate temporal information in speech recognition are:

- the use of the time derivatives, the so called delta and delta-delta features [2].
- the use of Cepstral Mean Subtraction [4]. This is a form of high-pass filtering that try to improve the recognition accuracy in noisy conditions, eliminating the slowly changing components from the time trajectory.
- the use of the Rasta-PLP filtering technique [8][9]. It consists of applying an IIR filter on the time trajectory of each spectral component. The frequency response of the RASTA filters approximates a bandpass filter from 1-10 Hz. This technique improves the recognition of speech in the presence of noise.
- the use of temporal streams (TRAPS) [10]
- the use of the modulation spectrogram, that is the result of applying spectral analysis to the temporal trajectory of each power spectral component of speech [11].

Perception experiments reveal that by applying such filtering processes the intelligibility of speech is not too much affected [3]. However, for automatic speech recognition (ASR) these filtering processes uses some ad hoc created filters (highpass, lowpass or bandpass) with no adaptation to the considered data.
The finite impulse response (FIR) neural network has been used successful for time series prediction and has been the winner at the Santa Fe competition [5][6]. It was reported that experiments for recognition of the phonemes b,d,g using a neural network consisting of a Time Delay Neural Network (TDNN) and one Dynamic Time Warping (DTW) module was very successfull too [12]. The TDNN is functionally identical to FIR just the training algorithms are different. In the present paper we shall develope a java software application for speech recognition based on a FIR-MLP neural network where the FIR processing units are defined by means of differential operators as defined in [1]. In our experiments we have used different types of weight functions as follows:
as linear synapse is defined

$$w_{ji}(t) = \lambda_{ji} \cdot \frac{t}{T} \tag{1}$$

as weak delay synapse is defined

$$w_{ji}(t) = \lambda_{ji} \cdot \frac{t}{T} \cdot \exp(-\frac{t}{T}) \tag{2}$$

and last as strong delay synapse is defined

$$w_{ji}(t) = \lambda_{ji} \cdot \frac{t}{T^2} \cdot \exp(-\frac{t}{T}) \tag{3}$$

with λ_{ji} real coefficients.

II. Neural network description

A. Introduction

In the static multi layer perceptron if x_i is the input to neuron j, w_{ji} is the synaptic coefficient on synapse j receiving input signal x_i then the potential and output from neuron j are described by:

$$\begin{cases} v_j = \sum_{i=1}^{N} w_{ji} \cdot x_i + \theta_j \\ u_j = \varphi(v_j) \end{cases} \tag{4}$$

where φ is the activation function.
If we consider that input signal is time dependent, $x_i(t)$, then the weight function , $w_{ji}(t)$, will be time dependent too. The answer v_j of the neuron will be modelled as a filter

$$v_j = \sum_{i=1}^{N} w_{ji}(t) \bullet x_i(t) + \theta_j$$

where $\bullet$ denotes the convolution product between two functions, defined in general by

$$f(t) \bullet g(t) = \int_{-\infty}^{+\infty} f(\tau) \cdot g(t-\tau) d\tau$$

The mathematical model of a FIR neuron will be

$$v_j = \sum_{i=1}^{N} \int_{-\infty}^{+\infty} w_{ji}(\tau) \cdot x_i(t-\tau) + \theta_j$$

and

$$x_j = \varphi(v_j)$$

B. FIR Synapse description

The signal flow for a FIR neuron is shown in fig.1

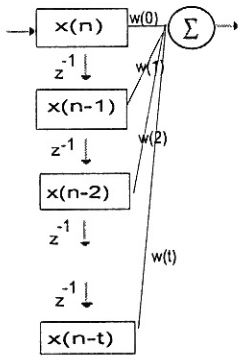

Fig. 1. FIR neuron

In [1] a new concept of defining the weights functions by means of differential operators has been introduced. We will follow here the main steps.

Let consider that the dynamic of synapses is determined by a linear differential operator

$$P_{ji}(D) = \frac{d^m}{dt^m} + a_1^{ji} \cdot \frac{d^{m-1}}{dt^{m-1}} + \ldots + a_{m-1}^{ji} \cdot \frac{d}{dt} + a_m^{ji} \ , a_i \in R$$

If an input signal $x_i(t)$ will be applied at the synapse i, of the neuron j, then

$$P_{ji}(D) \cdot u_{ji}(t) = x_i(t)$$

where u_{ji} is the response of neuron to the input signal $x_i(t)$. The impulse response $h_{ji}(t)$ is defined as the solution of the equation

$$P_{ji}(D) \cdot h_{ji}(t) = \delta$$

where

$$\delta(t) = \begin{cases} 0 & ,t \neq 0 \\ \infty & ,t = 0 \end{cases} \quad is\ the\ Dirac's\ distribution$$

The solution of this equation is represented by

$$h_{ji}(t) = P_{ji}^{-1}(\delta)$$

and characterize the FIR neuron. For different forms of the differential operator P_{ji}, one can compute different types of weight functions.

If we consider now that the dynamic of the synapse is defined by the differential operator

$$P_{ji}(D) = \frac{d}{dt} + \lambda_{ji}$$

then the weight function will be

$$h_{ji}(t) = Y(t) \cdot \exp(-\lambda_{ji} \cdot t) = \begin{cases} \exp(-\lambda_{ji} \cdot t) \ ,t \geq 0 \\ 0 \ ,t < 0 \end{cases}$$

If the differential operator is given by

$$P_{ji}(D) = \frac{d^2}{dt^2} + \lambda_{ji}^2$$

the synapses coefficients can be formulated as

$$h_{ji}(t) = Y(t) \cdot \frac{\sin(\lambda_{ji} \cdot t)}{\lambda_{ji}} = \begin{cases} \frac{\sin(\lambda_{ij} \cdot t)}{\lambda_{ji}}, t \geq 0 \\ 0, \ t < 0 \end{cases}$$

And finally considering the differential operator

$$P_{ji}(D) = \left(\frac{d}{dt} + \lambda_{ji}\right)^{(m)}$$

the solution is given by

$$h_{ji}(t) = \frac{1}{(m-1)!} Y(t)\, t^{m-1} e^{(-\lambda_{ji} t)}$$

For m=2 we will obtain

$$P_{ji}(D) = \left(\frac{d}{dt} + \lambda_{ji}\right)^{(2)} = \frac{d^2}{dt^2} + 2 \cdot \lambda_{ji} \cdot \frac{d}{dt} + \lambda_{ji}^2$$

The solution is

$$h_{ji}(t) = P_{ji}^{-1}(\delta) = Y(t) \cdot t \cdot \exp(-\lambda_{ji} \cdot t)$$

In our implementation we will use simplified versions of the weight functions as already defined in equations (1)(2)(3)

III. Software description

We have developed a java application with components for:

- recording speech in many formats and at different sampling rates
- signal and speech analysis
- filter design
- words database creation
- neural network creation and training
- voice activity detection and words recognition

A. Signal processing toolkit

1) Filter design module: With this module FIR and IIR filters can be designed using various methods. FIR filter can be designed with

- windowing method
- parks mc'clelland (remez exchange algorithm)

IIR filter

- butterworth
- chebischev

2) Signal and speech analysis module: Some algorithms for speech analysis have been implemented:

- FFT
- Cepstrum
- LPC
- MFCC
- pitch determination algorithms from [13]

The signal can be previously filtered using FIR or IIR filters.

3) Temporal pattern extraction module: We have developed modules for different feature extraction algorithms like FFT, Cepstrum, LPC, MFCC. Introduced in 1980 the algorithm of Davis and Mermelstein (MFCC) has become very popular and has been used in many speech recognition experiments. In our experiments we have used it too. Fig. 2 shows the developed tool for extraction of temporal MFCC features. It offers the possibilities:

- to mark a desired audio segment for further analysis and to display it in another window (using preemphasis,weighting)
- to chose different weighting windows types and sizes for FFT calculation
- to choose parameters like preemphasis value, the number of mel filters and of mfcc coefficients
- to display the spatial-temporal 3D pattern (fig. 6)

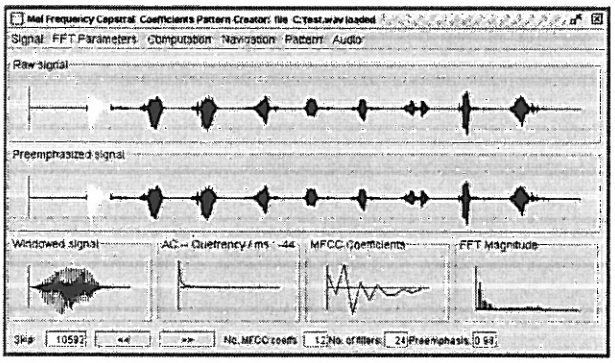

Fig. 2. MFCC Pattern creator

4) Database module: The pattern assignment to a desired class and database management is done with the module from fig. 3. As an optional step after database creation, cepstral mean subtraction can be applied, improving the robustness against convolutional noise.

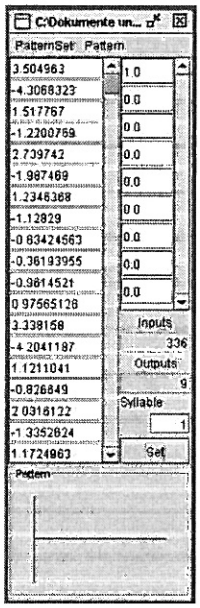

Fig. 3. Word to pattern assignment

B. Neural network toolkit

1) Neural network creator: FIR-MLP neural networks can be created with the options to chose:

- the size of the input pattern
- the number of neurons, the time delays for every layer, and the output functions for these neurons
- the type of synapses between layers and the time delays for performing the integration

2) Neural network trainer: In the training module one can chose the values of the following parameters:

- the number of training cycles
- the number of training steps per cycle
- the error to be reached
- the option to train the network progressive - by beginning with a small number of patterns and increase this number after reaching the desired error.

3) Recognizer: The recognition process is described in fig. 4 where the *feature extraction* is the same as in fig. 7. The recognizer module offers the options:

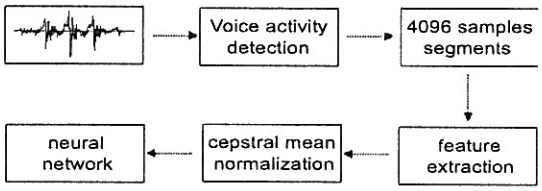

Fig. 4. Recognition flow module

- to load an audio file
- to load a neural network
- to detect speech segments and to mark them
- to recognize words from the detected speech segments

The recognition process starts first with loading a trained network and an audio signal file. The second step is to start the voice activity detection (VAD) module. This module uses a simple algorithm based on signal energy and threshold crossings as described in [13]. The founded speech segments are displayed and marked with another color in the VAD window. Finally the recognition process can be started and the recognized words are displayed and labelled in the recognition window.

IV. Experiment Description

All the steps described in this section have used the software modules described in III

A. Signal processing and feature extraction

The speech has been recorded with a microphone without noise reduction at a sampling rate of 11025 samples/s.
From one 4096 samples audio signal segment, representing 375 ms of speech, we have created a temporal pattern as follows:

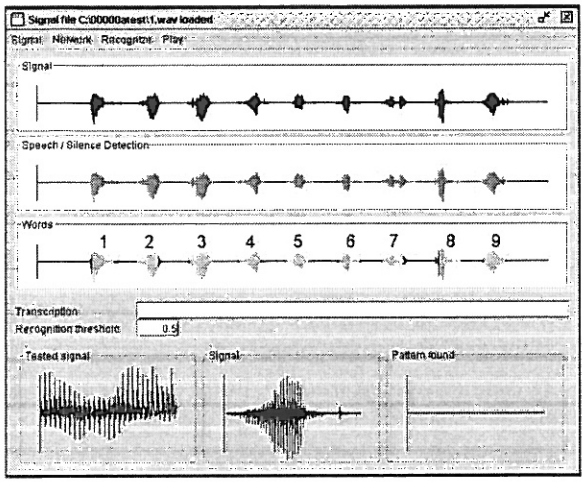

Fig. 5. Recognizer module

- from the first 512 samples we calculate 12 mfcc coefficients
- we shift the window with 128 samples and calculate 12 mfcc coefficients again
- we repeat the last step while samples exists

After this procedure we have obtained 28 feature vectors (28x12 matrix) that will represent a training pattern. Fig. 6 illustrates the temporal trajectories of the extracted mfcc coefficients. The static mfcc parameters have been obtained as

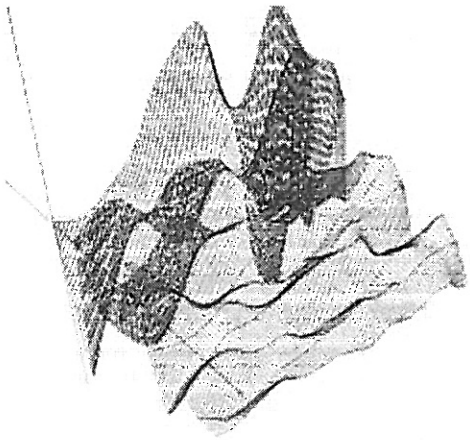

Fig. 6. Temporal trajectory of the 12 mfcc coefficients for german word eins (one). Window length is 1024 samples with hanning window, 32 samples step

described in fig. 7 by using a 0.98 preemphasis value, hanning window and 24 mel filter banks. In this step we have used an additional spectral subtraction component to remove the additive noise.

B. Databases

The training databases have been created by selecting the words manually using the module from section III-A.3 as described in IV-A. We have created a database of german digits from one to nine consisting of 90 patterns from one speaker and another database consisting of 450 mfcc patterns

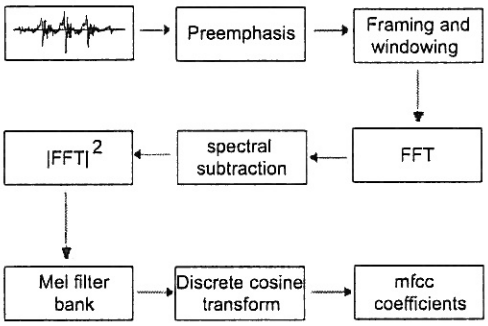

Fig. 7. Features extraction flow

from five speakers (every speaker 10 times every digit). Cepstral mean subtraction (CMS) have been applied to both databases.

From all speakers additional speech was recorded, indeed every speaker repeat 15 times all digits from one to nine. No test database have been created because in the recognition process the words are found automatically as described in section III-B.3

C. Neural networks description

We created different neural networks with 48 input, 96 hidden and 9 output FIR neurons, but with different neuron output functions and synapse types. The neurons in the input layer have 19 delays, in the hidden layer 10 and the output neurons have no delay. All FIR synapses have 10 delays. Such a neural network contains 6048 FIR synapses. If we look at the temporal resolutions in the network we see that 28 time intervals, corresponding to 375 ms of speech, are presented to the network, the input layer generates 19 time intervals and the hidden layer only 10. The neurons in different layers integrates for different time resolutions corresponding to different events level in the speech signals. The synapse types used are defined in equations (1)(2)(3). For a given neural network the synapses were chosen all from the same type, input and hidden neurons have all the same output function as listed in table I. The output layer neurons have had in all experiments as output the linear function. As training algorithm we used the standard online temporal backpropagation [5] [7] using coefficients from equations (1)(2)(3). The neural networks training was done using as error function the mean squared error, but the McClelland objective function is used as distance between the achieved network output and stored digits coding when classification is desired.

V. Main Results

We have investigated the training times, in terms of cycles that the networks needed to accomplish the task of learning the given mapping and their recognition performance. The times for propagating the signal through network and computing the new weights are recorded too. The results in table I and table II have been obtained in our experiments by using different network parameters, regarding different output functions and synapse types. These tables presents the different

combinations of the described processing elements for a FIR-MLP as described in IV-C. As it can be seen for a given synapse type, the change of the neurons output function from sigmoid to tanh reduces the needed training cycles. The same can be observed if we fix the neurons output function and change the synapse type. In recognition rates there are only

TABLE I

ONE SPEAKER EXPERIMENT - TRAINING CYCLES

synapse type	sigmoid output	tanh output	recognition rate
linear	170	34	95.3
weak delay	92	27	98.2
strong delay	400	75	97,5

minor variations (0.1) and we presented the results in only one column. By combining the tanh as neuron output function

TABLE II

FIVE SPEAKERS EXPERIMENT - TRAINING CYCLES

synapse type	sigmoid output	tanh output	recognition rate
linear	250	125	94
weak delay	159	70	97.5
strong delay	533	203	96.2

with FIR synapse defined by

$$h_{ji}(t) = \lambda_{ji} \cdot \frac{t}{T} \cdot \exp(-\frac{t}{T}) \qquad (5)$$

we obtained the best recognition rate with the fastest learning time (cycles). In table III is given the number of multiplications needed to calculate once the network and how many times neuron outputs are calculated.

TABLE III

OPERATIONS NEEDED FOR ONE NET COMPUTATION

multiplications needed	5.270.400
neuron outputs calc.	1881

TABLE IV

EXECUTION TIME

Processor	Net calculation	update weights
AMD AthlonXP 2 GHz	62,2 ms	16,1 ms
AMD AthlonXP 2.1 GHz	47,1 ms	12,1 ms
Intel Pentium4 2.8 GHz	210,3 ms	48,1 ms

The executions times listed in table IV were measured in different computer configurations.

VI. CONCLUSION

We have presented a java speech recognition application. We have created, trained and tested different FIR neural networks and those performances have been reported. The achieved results denotes that the FIR neural network architecture has a great potential for speech recognition. By combining the tanh as neuron output function with nonlinear FIR synapse we have obtained the best recognition rate with the fastest learning time (cycles) from all the networks presented here.
An apparent disadvantage of this model could be the relativelly high computational costs.
As application areas, if we consider that the knowledge is represented in the network and there is no need for maintaining a database for pattern matching, that implies that no additional storage and memory will be required, this networks could be used in devices with low memory requirements like mobile devices for the task of limited vocabulary word recognition. It should be mentioned that some of such devices have incorporated digital signal processors that are designed and optimized for filtering operations like our synaptic computations.
It should be mentioned that for feature extraction we have used the mfcc coefficients, that have been developed in 1980 and are not optimal for some reasons. In the last years computational models like meddis hair cell and lyon cochlear model have been introduced. These are clearly better modelling of the processes that takes place in human hearing as the mfcc or bark coefficients are. The beginning research that have been done using such models reveals that better results in speech recognition could be obtained in noisy conditions. A system composed of a such advanced auditory model coupled with a temporal processing neural network should be considered in future.

ACKNOWLEDGMENT

This paper presents the results obtained in a master thesis in computer sciences supervised by Prof. Badea Claudia Lidia from the University of Salzburg, Austria.

REFERENCES

[1] Badea Claudia Lidia,"ANN for time processing",*KES*, Osaka, Japan, 2001.
[2] S.Furui, "Cepstral analysis technique for automatic speaker verification", *IEEE Trans. Acoust. Speech Signal Process.*, 1981, vol. 29, pp. 254–272.
[3] R. Drullman, J.M. Festen, R. Plomp "Effect of temporal envelope smearing on speech reception", *Journal Acoust. Soc. Am.*,1994, vol. 95, pp. 1053–1064.
[4] R.Stern,A.Acero,F.H.Liu,Y.Oshima "Signal Prcessing for Robust Speech Recognition", *Automatic Speech and Speaker Recognition*, Kluwer Academic,1996.
[5] Eric A.Wan, "Finite impulse response neural networks with application in time series prediction", *PhD*, Stanford University,1993.
[6] Eric A.Wan, "Modelling nolinear dynamics with neural networks: examples in time series prediction".
[7] R.Timothy Eduards, "An overview of temporal backpropagation", Stanford University,1993.
[8] H.Hermansky,N. Morgan, "RASTA processing of speech", *IEEE Trans. Speech and Audio.*,1994, vol. 2, pp. 578–589.
[9] H.Hermansky "Perceptual linear predictive (PLP) analysis of speech", *Journal Acoust. Soc. Am.*,1990, vol. 87, pp. 1738–1752.
[10] H.Hermansky, S. Sharma "Temporal streams (TRAPS) in ASR noisy speech", *Proc. ICASSP.*,Mar.1999, Phoenix, vol. 87, pp. 289–292.

[11] H.Hermansky, "The modulation spectrum in automatic recognition of speech", *IEEE Workshop on Automatic Speech Recognition and Understanding.*,1997.
[12] A.Waibel,T.Hanazawa,G.Hinton,K.Shiano,K.Lang "Phoneme Recognition using Time-Delay Neural Networks", *IEEE Trans. Acoust. Speech Signal Process.*,March 1989.
[13] W.Hess"Pitch determination of speech signals", *Springer Verlag* ,1983.