Speaker Identification System based on PLP Coefficients and Artificial Neural Network

Fatma zohra Chelali, Amar.Djeradi, Rachida.Djeradi

***Abstract*—Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves.**

Feature extraction for speech recognition is a subject of a major interest today; different features have been investigated in speech recognition systems. The perceptual linear predictive PLP: this technique uses three concepts from the psychophysics of hearing to derive an estimate of the auditory spectrum: (1) the critical-band spectral resolution, (2) the equal loudness curve, and (3) the intensity-loudness power law. This paper discusses the development of a speaker identification and phoneme classification system. In particular, we develop an artificial neural network: multilayer perceptron MLP using PLP coefficients of voice signal. The performance of the system has been tested in experiments using 14 Arabic phonemes, specifically the Arabic fricatives uttered by 4 Algerian native speakers.

Our results demonstrates the efficiency of the PLP-MLP algorithm, a good recognition rate was obtained.

Keywords- PLP, LPC, cepstrum, neural network, MLP.

I. INTRODUCTION

Speaker recognition can be classified into identification and verification. Speaker identification is the process of determining which registered speaker provides a given utterance. Speaker verification, on the other hand, is the process of accepting or rejecting the identity claim of a speaker.

Speaker recognition methods can be divided into text-independent and text-dependant methods. In a text-independent system, speaker models capture characteristics of somebody's speech which show up irrespective of what one is saying. In a text-dependant system, on the other hand, the recognition of the speaker's identity is based on his or her speaking one or more specific phrases or words[1][2].

Speaker recognition systems contain two main modules: feature extraction and feature matching. Feature extraction is the process that extracts a small amount of data from the voice signal that can later be used to represent each speaker. Feature matching involves the actual procedure to identify the unknown speaker by comparing extracted features from his/her voice input with the ones from a set of known speakers[1][2].

The speaker recognition systems are presented in two phases: training phase and recognition phase. In the training phase, each registered speaker has to provide samples of their speech so that the system can build or train a reference model for that speaker. In the testing phase, the input speech is matched with stored reference model(s) and a recognition decision is made.

Speaker recognition is a difficult task. The principle source of variance is the speaker himself/herself. Speech signals in training and testing sessions can be greatly different due to many facts such as people voice change with time, health conditions, speaking rates, and so on. There are also other factors, beyond speaker variability, that present a challenge to speaker recognition technology. Examples of these are acoustical noise and variations in recording environments [1] [2].

One of the first decisions in any pattern recognition system is the choice of what features can be used and how exactly to represent the basic signal that is to be classified, in order to make the classification task easiest[3]. Through more than 30 years of recognizer research, many different feature extraction of the speech signal have been suggested and tried.

The most popular feature representation currently used is the Mel-frequency Cepstral coefficients (MFCC). Another popular feature representation is known as perceptual linear predictive (PLP) [3].

The PLP analysis technique was originally designed to suppress speaker dependent components in features used for automatic speech recognition, but later experiments demonstrated the efficiency of their use for speaker recognition tasks [4].

The system that we will describe is classified as text-dependent (or phoneme) speaker identification system since its task is to identify the person who speaks regardless of what is saying.

The article is presented as follows: section II presents an overview of our recognition system; the results of our experiments are reported in section III, followed by a conclusion in section IV.

II. OVERVIEW OF THE SYSTEM

A. *Feature extraction*

A wide range of possibilities exist for parametrically representing the speech signal for the speaker recognition task, such as Linear Prediction Coding (LPC), Mel-Frequency Cepstrum Coefficients (MFCC), Perceptual linear Predictive coefficients(PLP).

The authors are with Speech communication and signal processing laboratory, Faculty of Electronics and computing, University of Science and Technology Houari Boumedienne (USTHB),Algiers.ALGERIA Box n°:32 El Alia, 16111, Algiers, Algeria
Fax: (213) 21247187 (emails: Chelali_zohra@yahoo.fr adjeradi05@yahoo.com r_djeradi@yahoo.fr)

Perceptual linear predictive analysis (PLP) was proposed by Hynek Hermansky in 1989 [3]. PLP analysis is similar to linear predictive coding (LPC), except that the PLP technique also uses three concepts from the psychophysics of hearing. These three concepts are the critical-band spectral resolution, equalloudness curve, and intensity-loudness power law [5].

Both LPC and PLP use the autoregressive all-pole model to estimate the short-term power spectrum of speech. However, as pointed out by Hermansky, the LPC all-pole model is not consistent with human auditory perception because it does not consider the nonuniform frequency resolution and intensity resolution of hearing. PLP alleviates this problem by applying the all-pole model to the auditory spectrum. The auditory spectrum is designed to be an estimate of the mean rate of firing of auditory nerve fibers [5].

B. PLP Algorithm

In the PLP technique, several well-known properties of hearing are simulated by practical engineering approximations, and the resulting auditorylike spectrum of speech is approximated by an autoregressive all-pole model [6] [11]. A block diagram is shown in figure (1).

Spectral analysis

The speech segment is weighted by the Hamming window

$$w(n) = 0.54 + 0.46\cos[2\pi n/(N-1)]$$

(1)

Where N is the length of the window.

The typical length of the window is about 20ms.The discrete Fourier transform(DFT) transforms the windowed speech segment into the frequency domain. Typically, the fast fourier transform (FFT)is used here [6].

The real and imaginary components of the short-term speech spectrum are squared and added to get the short term power spectrum [6].

$$P(w) = \mathrm{Re}[s(w)]^2 + \mathrm{Im}[s(w)]^2$$

(2)

Critical-band spectral resolution

The spectrum P(w) is warped along its frequency axis w into the bark frequency Ω by

$$\Omega(w) = 6\ln\left\{w/1200\pi + [(w/1200\pi)^2 + 1]^{0.5}\right\}$$

(3)

The resulting warped power spectrum is then convolved with the power spectrum of the simulated critical-band masking curve $\Psi(\Omega)$. This step is similar to spectral processing in mel cepstral analysis, except for the particular shape of the critical-band curve. In PLP technique, the critical-band curve is given by:

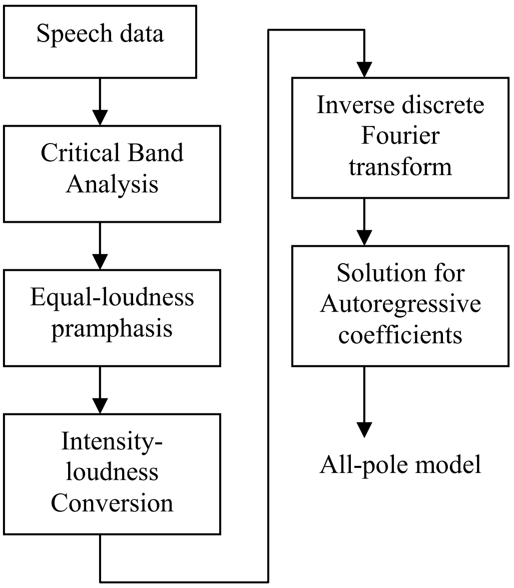

Speech data
Critical Band Analysis
Equal-loudness pramphasis
Intensity-loudness Conversion
Inverse discrete Fourier transform
Solution for Autoregressive coefficients
All-pole model

Figure 1. Block diagram of PLP speech analysis (hermansky)[6]

$$\Psi(\Omega) = \begin{cases} 0 & for\ \Omega < -1.3, \\ 10^{2.5(\Omega+0.5)} & for\ -1.3 \le \Omega \le -0.5, \\ 1 & for\ -0.5 \le \Omega \le 0.5, \\ 10^{-1.0(\Omega-0.5)} & for\ 0.5 \le \Omega \le 2.5, \\ 0 & for\ \Omega \succ 2.5. \end{cases}$$

(4)

The discrete convolution of $\Psi(\Omega)$ with (the even symmetric and periodic function) P(w) yields samples of the criticl-band power spectrum.

$$\theta(\Omega_i) = \sum_{\Omega=-1.3}^{2.5} P(\Omega - \Omega_i)\Psi(\Omega)$$

(5)

The convolution with the relatively broad critical-band masking curves $\Psi(\Omega)$ significantly reduces the spectral resolution of $\theta(\Omega)$ in comparison with the original P (w). This allows for the down-sampling of $\theta(\Omega)$.

Equal-loudness preemphasis

The sampled $\Theta[\Omega(w)]$ is preemphasized by the simulated equel-loudness curve:

$$\Xi[\Omega(w)] = E(w)[\Theta(w)]$$

(6)

The function E (w) is an approximation to the non equal sensitivity of human hearing at different frequencies and simulates the sensitivity of hearing at about the 40-dB level.

The particular approximation is adopted from makhoul and Cosell(1976) and is given by:

$$E(w) = \left[(w^2 + 56.8*10^6)w^4\right] / \left[\begin{matrix}(w^2 + 6.3*10^6)^2 * \\ (w^2 + 0.38*10^9)\end{matrix}\right]$$

(7)

Finally, the values of the first (0bark) and the last (Nyquist frequency) samples (which are not well found) are made equal to the values of their nearest neighbors. Thus $\Xi[\Omega(w)]$ begins and ends with two equal-valued samples [6].

Intensity-loudness power law

The last operation prior to the all-pole modelling is the cubic-root amplitude compression

$$\Phi(\Omega) = \Xi(\Omega)^{0.33}$$

(8)

This operation is an approximation to the power law of hearing (Stevens1957) and simulates the nonlinear relation between the intensity of sound and its perceived loudness. Together with the psychophysical equal-loudness preemphasis, this operation also reduces the spectral amplitude variation of the critical band spectrum so that the following all-pole modelling can be done by a relatively low model order [6].

Autoregressive modelling

In the final operation of PLP analysis, $\Phi(\Omega)$ is approximated by the spectrum of an all-pole model using the autocorrelation method of all-pole spectral modelling. We give here only a brief overview of its principle: the inverse DFT (IDFT) is applied to $\Phi(\Omega)$ to yield the autocorrelation function dual to $\Psi(\Omega)$. The first M+1 autocorrelation values are used to solve the Yule-Walker equations for the autoregressive coefficients of the Mth-order all-pole model. The autoregressive coefficients could be further transformed into some other set of parameters of interest, such as cepstral coefficients of all-pole model [6].

III. Phoneme Classification and Speaker Identification

The text-dependent speaker identification system that we have developed can be divided into two “subsystems” or, in other words, has to accomplish two tasks: Digitize the spoken utterance; divide it into frames and compute features (PLP coefficients+ first and second derivate) for each frame; classify each frame as belonging to a specific speaker with a neural network; and, finally, given the neural network’s outputs for each frame, determine who the speaker is [5].

For each frame nine (9) Perceptual Linear Prediction (PLP) features are computed. The PLP analysis technique was originally designed to suppress speaker dependent components in features used for automatic speech recognition, but later experiments demonstrated the efficiency of their use for speaker recognition tasks. For each frame a 27-dimensional vector is constructed.

A. Neural Network

A neural network is used to classify each frame as belonging to a specific speaker. The network has a three-layered architecture and is trained using the back-propagation algorithm [8]. The number of the input nodes is equal to the size of the input vectors. The number of the output nodes is equal to the number of the registered to the system speakers. Finally, the number of the hidden nodes is chosen by the user.

Multilayer Perceptron

Multilayer Perceptron Neural Networks are feed-forward and use the Back-propagation algorithm. We imply feed-forward networks and Back-propagation algorithm (plus full connectivity). A typical topology of a fully connected feed-forward network is shown in Figure 2. While inputs are fed to the ANN forwardly, the ‘Back’ in Back-propagation algorithm refers to the direction to which the error is transmitted.

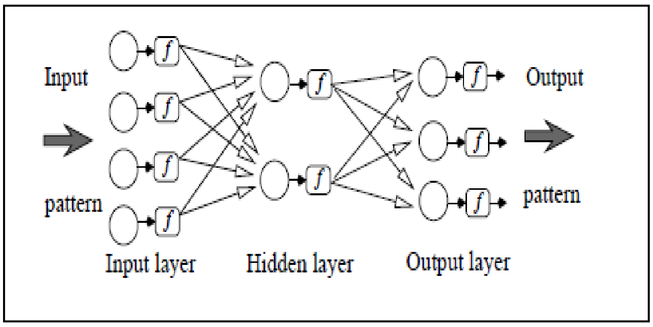


Figure 2. Architecture of FFNN for classification [7]

Learning process in Backpropagation requires providing pairs of input and target vectors. The output vector y of each input vector is compared with target vector d. In case of difference the weights are adjusted to minimize the difference. Initially random weights and thresholds are assigned to the network.

The logistic function $f(x) = \dfrac{1}{1+\exp(-x)}$ which maps the real numbers into the interval [−1 + 1] and whose derivative, needed for learning, is easily computed $\{f'(x) = f(x)[1 - f(x)]\}$. The reason for its popularity is the ease of computing its derivative [7].

Learning rules

Neural networks are adaptive statistical devices. This means that they can change iteratively the values of their parameters (i.e., the synaptic weights) as a function of their performance. These changes are made according to learning rules which can be characterized as supervised (when a desired output is known and used to compute an error signal) or unsupervised (when no such error signal is used) [7].

Backpropagation consists of measuring the error term between target output d (n) and the observed output y (n).

Case of the output unit:

If ej(n) is the observed error for the neuron j defined by the equation

$$e_j(n) = d_j(n) - y_j(n)$$

(9)

yj(n) and dj(n) represents the real output and the target output at neuron j in the output layer respectively[8].

The weights wji are updated every iteration in order to minimize the cost function or the mean square error between the output vector and the target vector [8].

We need to update weight of the network to minimize the output unit error.

$$E(n) = \frac{1}{2}\sum_{j \in c} e_j^2(n)$$

(10)

C: the total output neurons.

The output yj(n) of the neuron j is calculated by the equation

$$y_j = \varphi[v_j(n)] = \varphi\left[\sum_{i=0}^{r} w_{ji}(n) y_i(n)\right]$$

(11)

$\varphi[.]$ Represents the transfer function

$w_{ji}(n)$: Weights of network

$w_{j0}(n)$: biais of neuron j.

For each network output, we calculate its error term $\dfrac{\partial E(n)}{\partial w_{ji}(n)}$ and also $\Delta w_{ji}(n) = \_\eta \dfrac{\partial E(n)}{\partial w_{ji}(n)}$

$0 \le \eta \le 1$ is called learning rate of the backpropagation algorithm . finally, we obtain:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n) y_j(n)[1 - y_j(n)]\, y_i(n)$$

(12)

$$\Delta w_{ji}(n) = \_\eta \frac{\partial E(n)}{\partial w_{ji}(n)} = \eta \delta_j(n) y_i(n)$$

(13)

$$\delta_j = e_j(n) y_j(n)[1 - y_j(n)]$$

$$\delta_j = (d_j(n) - y_j(n)) y_j(n)[1 - y_j(n)] \qquad (14)$$

Case of hidden unit:

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_{k \in C} \delta_k(n) w_{kj}(n)$$

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -y_j(n)[1 - y_j(n)]\left[\sum_{k \in C} \delta_k(n) w_{kj}(n)\right] y_i(n)$$

$$\Delta w_{ji}(n) = \_\eta \frac{\partial E(n)}{\partial w_{ji}(n)} = \eta \delta_j(n) y_i(n)$$

(15)

With:

$$\delta_j(n) = y_j(n)[1 - y_j(n)] \sum_{k \in C} \delta_k(n) w_{kj}(n) \qquad (16$$

We can summarize all the operations by the following equation:

$$w_{ji}(n) = w_{ji}(n-1) + \Delta w_{ji}(n)$$

$$= w_{ji}(n-1) + \eta \delta_j(n) y_i(n)$$

(17)

The local gradient is defined:

$$\delta_j(n) = \begin{cases} e_j(n) y_j(n)[1 - y_j(n)] & \textit{if } j \in \textit{outputunit} \\ y_j(n)[1 - y_j(n)] \sum_{k \in C} \delta_k(n) w_{kj}(n) & \textit{if } j \in \textit{hiddenunit} \end{cases}$$

(18)

We can also define the generalized delta rule as follows:

$$w_{ji}(n) = w_{ji}(n-1) + \eta \delta_j(n) y_i(n) + \alpha\, \Delta w_{ji}(n-1)$$

where $0 \le \alpha \le 1$ is the *momentum* term.

Standard $\Delta w_{ji}(n) = \_\eta \dfrac{\partial E(n)}{\partial w_{ji}(n)}$ (19)

WithMomentum

$$w_{ji}(n) = w_{ji}(n-1) + \eta \delta_j(n) y_i(n) + \alpha\, \Delta w_{ji}(n-1)$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) + \alpha\, \Delta w_{ji}(n-1)$$

(20)

IV. Result

A. Feature extraction

To evaluate the performance of the proposed method, we collected a large number of speech signal of different speakers male and female at different moments pronouncing 14 Arabic syllabus (short vowels)), we choose in our experiments the voiced and the unvoiced fricatives phonemes:

س, ش ع , غ ف ح خ ز ظ ض ص ج ذ ه ث with their API representation[12].

The database includes 700 speech signals from four (4) different subjects. The speech signals are acquired during different sessions with a sampling frequency of 22 KHz.

The speech input is typically recorded at a sampling rate 22 KHz. This sampling frequency was chosen to minimize the effects of aliasing in the analog-to-digital conversion. These sampled signals can capture all frequencies up to 5 kHz, which cover most energy of sounds that are generated by humans.

Table 1. API representation of The 14 Arabic phonemes

ARABIC ALPHABET	ث	ج	ح	خ	د	ذ	ز	ش	ص	ظ	ع	غ	ف	ه
Phonetic Transcription(A P I)	θ	d z	ħ	x	d	δ	Z	š	ś	δ	ς	γ	f	h

Frame Blocking

In this step, the continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M (M < N). The first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by N - M samples and so on. This process continues until all the speech is accounted for within one or more frames. Typical values for N and M are N = 256 (which is equivalent to ~ 30 msec windowing and facilitate the fast radix-2 FFT) and M = 100[10].

Windowing

Typically the Hamming window is used, which has the form:

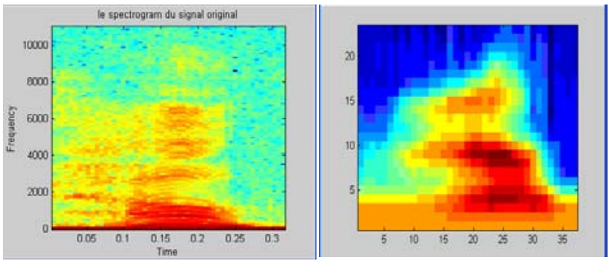
$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

$w(n),\ 0 \le n \le N-1$, where N is the number of samples in each frame.

Feature extraction (PLP coefficients)

This operation is done for every individual and for all the phonemes used (700 speech signals). For good word/speaker recognition accuracy, nine (9) PLP coefficients per frame are necessary.

The following shows the spectrum representation of phonem غ /γ /in frequency domain(linear scale) and its representation with PLP technique in bark scale.



-a- -b-
Spectrum representation of غ /γ /
a-linear scale b-Bark scale

Therefore, dimensionality reduction or speech parameterisation is a very important step which will greatly improve the performance of the speaker recognition system. The input matrix (the voice print matrix) has a dimension of 270 real values corresponding to 9 coefficients calculated for the 30 frames of each signal.

B. *Speaker Recognition system using MLP*

A neural network is used to classify each frame as belonging to a specific speaker. The network has a three-layered architecture and is trained using the back-propagation algorithm [9]. The number of the input nodes is equal to the size of the input vectors. The number of the output nodes is equal to the number of the registered to the system speakers. Finally, the number of the hidden nodes is chosen by the user.

The network will receive an input layer having a matrix of size (270*20), twenty corresponds to five (5) training signals for the four (4) speakers. The features test matrix is defined with variables called target, the target matrix has the same dimension as the training matrix. The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the output vector with 0’s.

Fourteen (14) neural networks were constructed for each specified phoneme. All the NNs trained present fast convergence and the training process terminated within 100 or 200 epochs, with the summed squared error (SSE) reaching the pre-specified goal (10-5) .

We used log-sigmoid functions as a transfer function at all neurons (In hidden layer and output layer).log-sgmoid is ideal for our system.

The speaker recognition system is initially trained with artificial neural network for a maximum of 4000 epochs or until the network sum-squared error falls below 0.0001(figure 3).

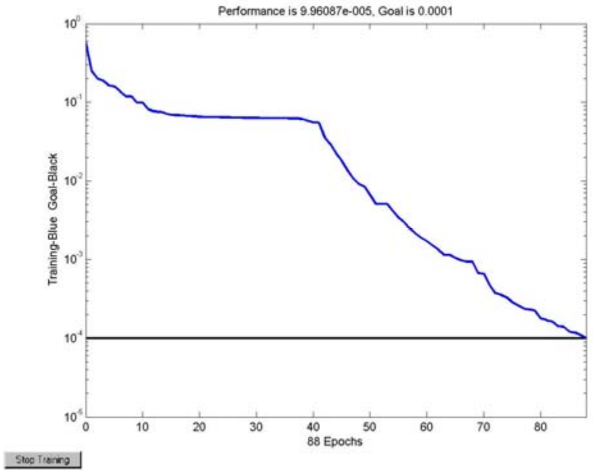


Figure 3: training phase : the MSE MLP for 4000 iterations

In order to show the importance of processing elements, we trained our MLP classifier with variable hidden unit from 5 to 45. The example showed in figure (4) is for the phoneme /š/ ش

For a small number of neurons (5 to 10) in the hidden layer we observed large MSE, solow accuracy. The MLP generalize poorly. After ~25 neurons, MSE came back to the levels of a system with only 5 neurons in the hidden layer. by adding more and more units in the hidden layer the training error can be made as small as desired but generally each additional unit will produce less and less benefit. When too many neurons, poor performance is a direct effect of overfitting. The system overfits the training data and does not perform well on novel patterns. The result of our speaker recognition depending on phonemes is presented in table 2.From this table, we can see that most of tested phonemes has an accuracy of 90 to 100 %, but only when the tested phoneme /ς/ ع and /dz/ ج the accuracy rate is 75 %.

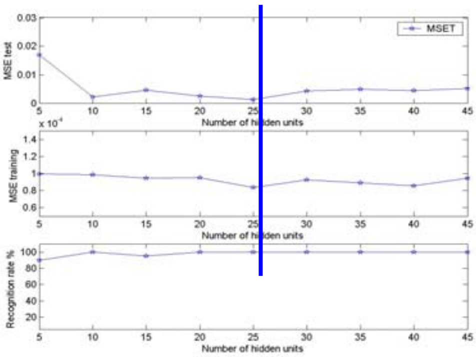


Figure 4. the optimal NN architecture when varying neurons in hidden layer

Table 2. Recognition Accuracy for each phoneme

Phoneme	training MSE	test MSE	Recognition rate %	#Neurons in hidden layer
/ς/ ع	$9.30\ 10^{-5}$	$6.9\ 10^{-3}$	100	20
/š/ ش	9.60e-5	1.863e-4	100	15
/d/ د	9.87e-5	4.710^{-3}	95	25
/δ/ ذ	9.47e-5	$2.3\ 10^{-3}$	100	20
/δ/ ظ	9.96e-5	5.910^{-3}	100	15
/dz/ ج	9.16e-5	$15.6\ 10^{-3}$	95	20
/f/ ف	9.24e-5	$18.9\ 10^{-3}$	90	25
/γ/ غ	9.80e-5	$16.5\ 10^{-3}$	95	30
/ħ/ ح	9.64e-6	$1.00\ 10^{-5}$	100	20
/h/ ه	8.65e-5	$5.4\ 10^{-3}$	100	20
/x/ خ	8.9e-5	$2.6\ 10^{-5}$	100	20
/ś/ ص	9.77e-5	$4\ 10^{-4}$	100	15
/θ/ ث	7.56e-5	1.88e-4	100	20
/Z/ ز	9.15e-5	1.74e-4	100	20

V. CONCLUSION

A speaker-dependent phoneme recognition system using PLP analysis and the MLP algorithm has been examined in this work. The PLP technique uses engineering approximations for three basic concepts from the psychophysics of hearing: (1) the critical-band resolution curves, (2) the equal-loudness curve and (3) the intensity-loudness power-law relation.

Artificial neural network and especially MLP are widely used in pattern recognition, experimental results showed that an accuracy rate of 100% can be achieved by using PLP features and MLP classifier.

The MLP classifier gives better recognition rate; the network was trained several times in order to find the optimal topology or architecture.

REFERENCES

[1] Cole R.A, Mariani J, Uszkoreit H., Zaenen A., Zue V. (eds.): Survey of the State of the Art in Human Language Technology. Cambridge University Press, 1997.

[2] Marcos Faundez-Zanuay, Enric monte-moreno, "state-of-the Art in speaker recognition", IEEE A&E Systems magazine, May 2005.

[3] M jamaati,H. Marvi and M. Lankarany, "vowels recognition using mellin transform and plp-based feature extraction", acoustics-08 Paris.

[4] Asterios Toutious, K.G.Margaritis,"Development of a text dependent Speaker Identification System with the OGI Toolkit", 2nd hellenic Conference on AI, SETN-2002,11-12 April 2002, Thessaloniki Greece, proceedings, Companion Volumenpp. 525-530.

[5] Wira Gunawan and Mark hasegawa-Johnson, "PLP coefficients can be quantizied at 400 BPS", department of electrical and Computer Engineering,University of Illinois at Urbana-Champaign,USA.

[6] H. hermansky,"perceptual linear predictive(PLP) analysis of speech", journal of the Acoustical Society of America, vol 87 no.4,pp 1738-1752,1990.

[7] Herve Abdi,"Neural networks",Program in Cognition and neurosciences,MS:Gr.4.1,The university of Texas at Dallas.

[8] Marc Parizeau ,"le perceptron multicouche et son algorithme de rétropropagation des erreurs",département de génie électrique et de génie informatique, Université de laval, 10 septembre 2004.

[9] N. Morgant, H. Hermansky,H. Bourlardt,P. Kohnt &C. Wooterst,"Continuous Speech Recognition using PLP Analysis with Multilayer perceptrons" , CH2977-719110000-0049,1991 IEEE.

[10] "An Automatic Speaker Recognition system" , available at: http://www.IFP.UIUC.EDU/~MINHDO/TEACHING/SPEAKER_RECOGNITION.

[11] Sid Ahmed Selouani and Jean Caelen, "un système connexionniste modulaire pour la reconnaissance des traits phonétiques de l'Arabe ».

[12] Phonologie Arabe, available at : http://fr.wikipedia.org/wiki/Arabe_(phonologie)