World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:4, 2008

Efficient System for Speech Recognition using General Regression Neural Network

Abderrahmane Amrouche, and Jean Michel Rouvaen

***Abstract*—**In this paper we present an efficient system for independent speaker speech recognition based on neural network approach. The proposed architecture comprises two phases: a preprocessing phase which consists in segmental normalization and features extraction and a classification phase which uses neural networks based on nonparametric density estimation namely the general regression neural network (GRNN). The relative performances of the proposed model are compared to the similar recognition systems based on the Multilayer Perceptron (MLP), the Recurrent Neural Network (RNN) and the well known Discrete Hidden Markov Model (HMM-VQ) that we have achieved also.
Experimental results obtained with Arabic digits have shown that the use of nonparametric density estimation with an appropriate smoothing factor (spread) improves the generalization power of the neural network. The word error rate (WER) is reduced significantly over the baseline HMM method. GRNN computation is a successful alternative to the other neural network and DHMM.

***Keywords*—**Speech Recognition, General Regression Neural Network, Hidden Markov Model, Recurrent Neural Network, Arabic Digits.

I. Introduction

DURING the two past decades various systems have been tested in automatic speech recognition (ASR). They are generally based on the stochastic approach using Hidden Markov Models (HMM) [1]. Markov Models provide mathematically rigorous approach to developing robust statistical signal models. This approach gave some success in clean speech and rapidly became the most used model for various speech recognition systems for many languages. The best performances are obtained with isolated words where discrete models (DHMM) using vector quantization (VQ) are normally sufficient although many works using continuous model are proposed.

Another promising technique for speech recognition is the neural network based approach. Artificial Neural Networks (ANN) [2], [3] are biologically inspired tools for information processing. The multilayer feed-forward networks, the recurrent network… etc. can be trained to associate input data, to learn unknown words.

Manuscript received May, 4, 2006. This work was supported in part by the Algerian Ministry of Research under CNEPRU Project J 1602/02/09/04

Amrouche is with the Faculty of Electronics and Computer Sciences at USTHB, P.O. Box 32, Bab Ezzouar, Algiers, Algeria (phone: 213-20472022; fax: 213-21247187; e-mail: namrouche@ usthb.dz; abderrahmane .amrouche@caramail.com).

J.M. Rouvaen is with the OAE Department (UMR 8520 CNRS) of IEMN, Valenciennes University, P.O Box 304, Le Mont Houy 59300, Valenciennes cedex, France (phone: 33-327511365; fax: 33-327511189; e-mail: rouvaen@univ-valenciennes.fr).

Speech recognition modelling by artificial neural networks (ANN) [4], [5] doesn't require a priori knowledge of speech process and this technique quickly became an attractive alternative to HMM. The first work applied to consonant phonemes recognition using Time-Delay Neural Network (TDNN) by Waibel [6], [7] has been followed by many attempts based on neural network classifiers. ANNs have been shown to yield good performances (something's better than HMM) on short isolated speech units [8].

In order to combine the function approximations capabilities of Artificial Neural Networks with the modelling power of Hidden Markov Models, ANNs have been integrated into HMM/ANN models where the neural networks are often used to compute emission posterior state probabilities [5], [8], [9].

Recently the works are focussed on continuous speech recognition whose applications are much broader but the performances are still weak. We note that the problem of the isolated word recognition is not solved for all the languages in particular for the Arabic language.

Arabic is currently one of the most widely spoken languages used in the world with an estimated number of 300 million speakers and it covers a large geographical area. Despite this, few studies have been reported on the automatic speech recognition of this language [10], [11], [12]. The first motivation of this work is to perform an independent speaker speech recognition system for the Arabic language.

Nonparametric techniques in pattern recognition can be used when no functional form for the density function is assumed. The density estimates is driven by the data without making any assumptions about the form of the distribution. Due to this interesting property we propose a new integrated approach of the neural network adaptation scheme based on nonparametric regression for independent speaker speech recognition.

This paper is organised as follows: a brief introduction outlines the motivation of this work. The nonparametric regression and neural network implementation are presented in section 2. Section 3 describes the proposed model and its performances. The next section describes the comparative study using MLP, RNN and HMM baseline system.

II. General Regression

A. Theoretical Fundament

The regression function performed on an independent variable X computes the most probable value of the dependent variable Y based on a finite set of observations of X and the associated values of Y.

Let *f(x,y)* be the joint continuous probability density function of a vector random variable, *x,* and a scalar random variable *y*. Let *X* be a particular measured value of the random variable *x*. The regression of *y* given *X*, is given by the conditional expectation of *y* on *X*.

$$E[y/X] = \frac{\int_{-\infty}^{+\infty} y.f(X,y)dy}{\int_{-\infty}^{+\infty} f(X,y)dy} \quad (1)$$

The regression will be parametric if the relationship between *x* and *y* is expressed in a functional form with parameters. But the joint density *f(x,y)* is usually unknown. Without any real knowledge of the functional form between the dependent and the independent variables it is more appropriate to use nonparametric estimation methods. In nonparametric density estimation, no fixed parametrically-defined form for the estimated density is assumed. Then, the probability density function must be estimated empirically from a sample of observations (data points) of *x* and *y*. The general form of the estimator is given by the following equation:

$$f_n(x) = \frac{1}{n\lambda}\sum_{i=1}^{n}\varphi(\frac{x-x_i}{\sigma}) \quad (2)$$

where the x_i are independent, identically distributed random variables with absolutely continuous distribution function. The weighting function φ must be bounded and satisfy the following conditions:

$$\int_{-\infty}^{+\infty}|\varphi(y)|dy < \infty \quad (3)$$

$$\lim_{y\to\infty}|y\varphi(y)| = 0 \quad (4)$$

and

$$\int_{-\infty}^{+\infty}\varphi(y)dy = 1 \quad (5)$$

The function $\sigma = \sigma(n)$ must be chosen with

$$\lim_{n\to\infty}\sigma(n) = 0 \quad (6)$$

$$\lim_{n\to\infty}n\sigma^2(n) = \infty \quad (7)$$

One useful form of the weighting function φ is the Kernel density function (Gaussian). Parzen has shown that these estimators are consistent. They asymptotically converge to the underlying distribution at the sample point when it is smooth and continuous. Parzen's results have also been extended to the multivariate distribution case [13]. Based upon sample values X^i and Y^i of the random variable *x* and *y*, a good choice for the probability estimator, as in [14], [15] is given by:

$$\hat{f}(X,Y) = \frac{1}{(2\pi)^{(p+1)/2}\sigma^{p+1}}.$$
$$\frac{1}{n}\sum_{i=1}^{n}\exp\left[-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}\right].\exp\left[-\frac{(Y-Y^i)^2}{2\sigma^2}\right] \quad (8)$$

where p is the dimension of the vector variable *x* ,
n is the number of observations (pattern sample points),
σ: the width (spread) of the estimating kernel or smoothing factor,
Y^i: the desired scalar output given the observed input X^i.

Defining the scalar function D_i^2

$$D_i^2 = (X-X^i)^T(X-X^i) \quad (9)$$

Combining (8) and (9) and interchanging the order of integration and summation, yields the desired conditional mean, designated as following:

$$\hat{Y}(X) = \frac{\sum_{i=1}^{N}Y^i\exp(-\frac{D_i^2}{2\sigma^2})}{\sum_{i=1}^{N}\exp(-\frac{D_i^2}{2\sigma^2})} \quad (10)$$

The resulting regression, (10), known also as Nadaraya-Watson kernel regression estimator, is directly applicable to problems involving numerical data. The estimate $\hat{Y}(X)$ can be visualised as a weighted average of all the observed values, Y^i, where each observed value is weighted exponentially according to its Euclidean distance from *X* [14].

B. Neural Network Implementation

General regression neural network implementation was firstly proposed by D. Specht [14], [15]. Let be w_{ij} the target output corresponding to input training vector x_i and j^{th} output. Equation (4) can be expressed as follows.

$$y_j = \frac{\sum_{i=1}^{n}w_{ij}.h_i}{\sum_{i=1}^{n}h_i} \quad (11)$$

$$\text{with } h_i = \exp(-\frac{D_i^2}{2\sigma^2}) \quad (12)$$

According to (11) and (12) the topology of a GRNN is described in Fig. 1 and it consists in:

1) The input layer (input cells), which is fully connected to the pattern layer
2) The pattern layer which has one neuron for each pattern. It computes the pattern functions $h_i(\sigma, C_i)$ which is expressed in (12) using the centres C_i .
3) The summation layer which has two units N and D. The first unit has input weights equal to X^i, then it computes the numerator N by summation of the exponential terms multiplied by the Y^i associated with X^i. The second unit has input weights equal to 1, then the denominator D is the summation of exponential terms alone

4) Finally the output unit divides N by D to provides the prediction result.

The choice of the smoothing factor is very important. When σ is small only a few samples plays a role. If it is large then even distant neighbours affect the estimate at X, leading to a very smooth estimate. In the extreme case, when σ goes to infinity $\hat{Y}(X)$ is the average of all Y^i independently of the input [14].

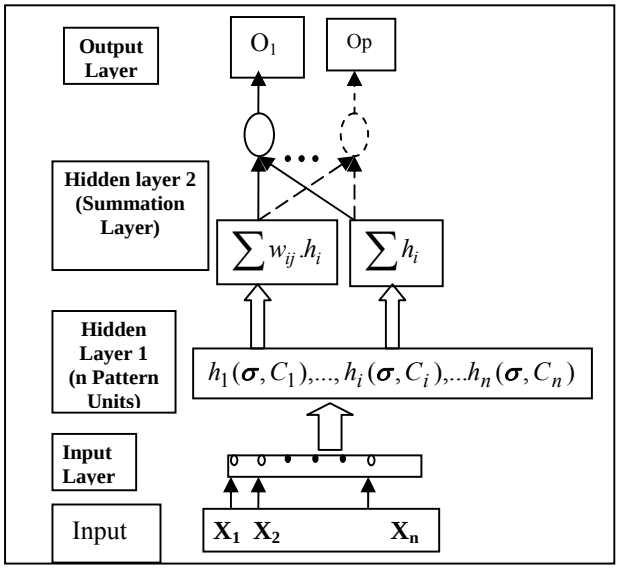

Fig. 1 General regression neural network implementation

III. Speech Recognition using General Regression Neural Network

A. Proposed Model

The general scheme of the proposed speech recognizer is depicted in Fig. 2. It consists in two parts: preprocessing phase and classification phase.

In the preprocessing phase the speech signal is firstly digitized and end pointed. Endpoint detector separates speech word and background noise using energy threshold level, zero crossing rate and temporal criteria especially for stopped consonants.

The digitized speech signal is pre-emphases by a first-order digital network in order to spectrally flatten the signal.

$$\hat{S} = S(n) - \mu S(n-1) \quad (13)$$

Where μ = 0.98.

The signal is fragmented into frames by using a Hamming window (256 points with half covering). As defined by ETSI for each frame the features extraction consists of 12 Mfcc (Mel frequency cepstrum coefficients) + log(energy) with their first and second derivatives (delta and acceleration). Each frame is then represented by an acoustic vector x_t as follow:

$$x_t = \{Mfcc,\ \Delta Mfcc,\ \Delta(\Delta Mfcc), Log\ (Energy)\} \quad (14)$$

These features constitute the input vectors to the neural network used as classifier. The classification phase uses general regression neural network as defined above.

B. Data Collection

The speech data used for training and testing the recognition system is a part of ARADIGITS, the Arabic speech data base collected from 200 Algerian natives aged from 18 to 50 and recorded in a large auditory room which was very quiet. Then, the speech segments have been downsampled at 16 kHz.

In the training phase a total of 1800 utterances pronounced by 90 speakers of both sex (equally distributed)

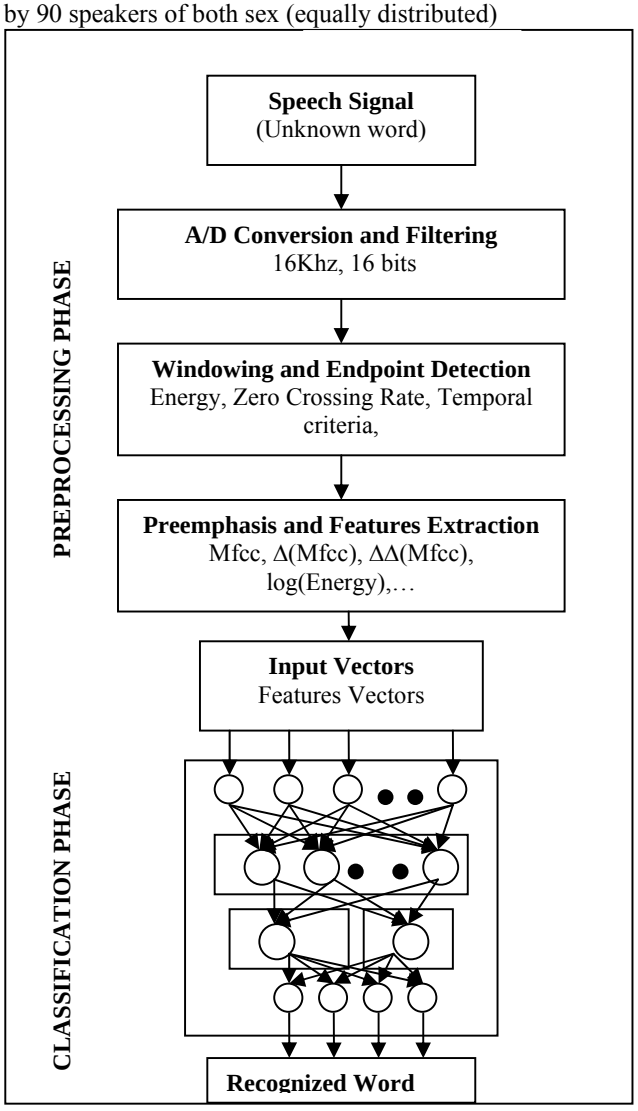

Fig. 2 Speech recognition system using general regression neural network proposed for isolated word recognition

were used. In testing phase 1000 utterances pronounced by 50 others speakers of both gender (25 males and 25 females) were used. The data in the testing set do not intersect those in the training set.

C. Optimization of Smoothing Factor or Spread

At the heart of GRNN is the kernel function. The output of kernel function is an estimate of how likely the unknown pattern of word belongs to that distribution. The optimization of the smoothing factor is critical to the performance of the GRNN and is usually found through iterative adjustment and the cross-validation procedure.

Several experiments were carried out with various values of the spread σ (smoothing factor). Fig. 3 shows the variation of word error rate (WER) versus σ. All recognition results are given in term of Word Error Rate defined as:

$$WER = \frac{S + D + I}{N} \times 100\% \quad (15)$$

where N is the total number of word in the test set, S denotes the total numbers of substitutions errors, D the total numbers of deletions errors and I the total number of insertions errors.

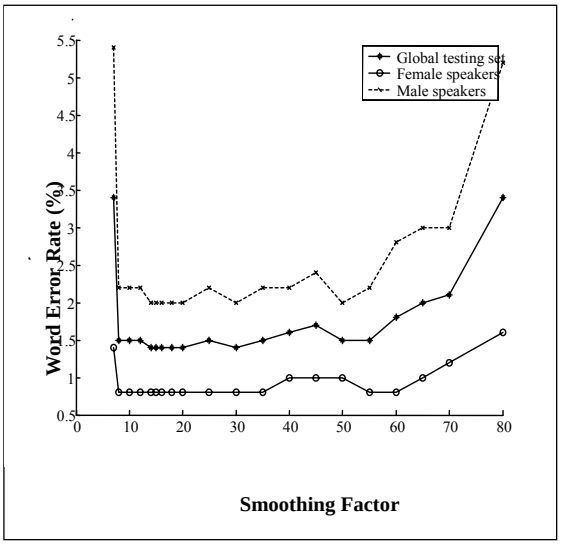

Fig. 3 Word error rate versus spread (smoothing factor) for Arabic digits recognition using GRNN

It was showed that for smoothing values less than 6, the majority of the words in testing set remain undetermined, some words have been recognized and a few words have been confused with others. For smoothing values between 7 and 12 the error rate decreases, very little set of words remain unrecognized and some words are confused with others. Fig. 3 shows that best results are obtained from σ between 14 and 20. For the following experiments smoothing factor was set to 15.

IV. Comparative Study

A. The HMM Baseline System

The bloc diagram of the baseline system achieved is proposed in Fig. 4. It is an isolated word recognition system based on discrete hidden Markov model with conventional vector quantization (CVQ). The model used is a stationary first order hidden Markov. Two steps could be distinguished:

- Training step, which permits the codebook generation and parameters model estimation.
- Recognition step in which the system selects the most probable model to issue the unknown word.

1. Training step

Vector quantization (VQ) is a method of compressing vector data by partitioning the continuous vector space into non-overlapping subsets and representing each subset with a unique codeword. The set of available codeword is termed the codebook. In discrete HMM system, the procedure of generating codebook is often associated to the unsupervised cluster algorithm like MKM (Modified K-means) or LBG (Linde–Buzo-Gray). In our work we have used the LBG algorithm. In the training step also, an HMM model is built for each word vocabulary. The estimation of model's parameter is done by the Baum-Welch algorithm as follows.

Models Training: Baum-Welch algorithm

Before we start the training, we must specify the model topology, the transition parameters and the output distribution parameters of the HMMs. A stationary first order hidden Markov model is characterized by the following elements:

- Set of hidden states $S = \{S_1, S_2 \ldots S_N\}$, (16)

where N is the number of states in the model.

- Set of observation symbols: $Y = \{Y_1, Y_2 \ldots Y_M\}$, (17)

where M is the codebook size.

- State transition probability matrix $A = \{a_{ij}\}$:

$$\text{where } a_{ij} = P(S_{t+1} = j \mid S_t = i), \quad 1 \le i, j \le N. \quad (18)$$

- observation symbol probability matrix $B = \{b_j(k)\}$: where

$$b_j(k) = P(Y_t = k \mid S_t = j), \quad 1 \le j \le N \text{ and } 1 \le k \le M. \quad (19)$$

- initial state probability $\Pi = \{\Pi_i\}$:

$$\text{where } \Pi_i = P(S_i, \ t=1) \quad (20)$$

Let $Y = \{y_1, y_2, \ldots . y_T\}$ be an observation sequence for training and $\lambda = (A, B, \Pi)$ a given model. The aim of the training is to find the model, say λ^*, such that:

$$\lambda^* = \arg\max_{\lambda} P(Y|\lambda) \quad (21)$$

where $P(Y|\lambda)$ is the likelihood of the sequence Y given the model λ. The procedure we used to find the model which gives us the maximum likelihood is the Baum-Welch algorithm [1]. To describe this algorithm, we need to define the forward variable $\alpha_t(i)$, which is the probability of having generated the partial observation sequence $(y_1 y_2 \ldots . y_t)$ with

state i at time t, given the model λ, and the backward variable $\beta_t(i)$, which is the probability of generating the partial observation sequence $(y_{t+1}\ y_{t+2}\ldots y_T)$, given the model λ, and that the state sequence starts from state i at time t. The forward and backward variables can be calculated by the following recursions:

$$\alpha_t(i) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(Y_t), \quad (22)$$

$1 \le t \le T,\ 1 \le j \le N$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j), \quad (23)$$

$1 \le t \le T\text{-}1,\ 1 \le i \le N$

From the definition of the forward $\alpha_t(i)$ and backward $\beta_t(i)$ variables, the variable $\varepsilon_t(i, j)$ which is the probability of being in state i at time t and state j at time $t+1$ given the observation sequence Y and model λ can be written in the following form:

$$\varepsilon_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j)} \quad (24)$$

The variable $\gamma_t(i)$ which is the probability of being in state i at time t given the observation sequence Y and the model λ is given by the following equation:

$$\gamma_t(i) = \sum_{j=1}^{N} \varepsilon_t(i, j) \quad (25)$$

If we assume that the individual observation sequence are independent of each other, the formulas to reestimate the model parameters a_{ij}, $b_j(k)$ and Π_i for a multiple observation sequences $Y^{(L)} = \{y_1^{(L)}, y_2^{(L)}, \ldots\ldots y_{TL}^{(L)}\}$ where L is the number of observation are given by:

$$\hat{a}_{ij} = \frac{\sum_{l=1}^{L} \frac{1}{P_l} \sum_{t=1}^{T_l - 1} \varepsilon_t^l(i,j)}{\sum_{l=1}^{L} \frac{1}{P_l} \sum_{t=1}^{T_l - 1} \gamma_t^l(i)} \quad (26)$$

$$\hat{b}_j(k) = \frac{\sum_{l=1}^{L} \frac{1}{P_l} \sum_{t=1, y_t = Y_k}^{T_l - 1} \gamma_t^l(j)}{\sum_{l=1}^{L} \frac{1}{P_l} \sum_{t=1}^{T_l - 1} \gamma_t^l(j)} \quad (27)$$

$$\hat{\pi}_i = \frac{1}{L} \sum_{l=1}^{L} \gamma_1^l(i) \quad (28)$$

A.2 Recognition Step

In the recognition step, for each unknowed sequence X_1^T of length T frames we have the observation sequence Y. To compute the probability $P(X_1^T, \lambda)$ of generating the sequence by the model, we can use the forward (backward) procedure or the Viterbi algorithm [1]. In our case we have used the Viterbi algorithm particularly the logarithm of the maximum likelihood given by the following equation (10)

$$P(X_1^T|\lambda) = \alpha(t,i) = \max_j \left[\alpha(t-1,j) + Loga_{ji}\right] + Logb(Y_t) \quad (29)$$

where: $1 \le t \le T$ and $1 \le i, j \le N$

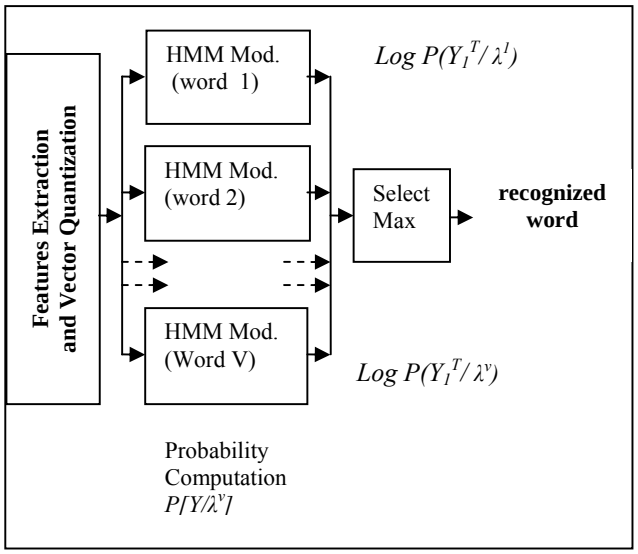

Fig. 4 Bloc diagram of VQ-HMM baseline system

B. Comparative Study

We have also achieved word recognition system using the well known MLP and the Elman Recurrent Neural Network classifiers (with a conjugate gradient training algorithm) including 300 neurons in the hidden layer. Tables I and II give the word error rate for comparative study using general regression approach versus the other methods used. Fig. 5 shows the efficiency of the general regression neural network based system.

TABLE I
COMPARATIVE STUDY FOR ARABIC DIGITS RECOGNITION SYSTEMS (FEMALE SPEAKERS)

		Word Error Rate (%)			
	Digit	DHMM	MLP	RNN	GRNN
0	şifr	12	6.0	6.0	4.0
1	wa:ħid	2.0	0.0	0.0	0.0
2	?iθna:n	16	0.0	0.0	0.0
3	θala:θa	10	6.0	4.0	2.0
4	?arbaça	8.0	0.0	2.0	0.0

5	χamsa	2.0	0.0	2.0	0.0
6	sitta	10	0.0	0.0	2.0
7	Sabça	16	0.0	0.0	0.0
8	θama:nija	10	4.0	0.0	0.0
9	Tisça	12	0.0	0.0	0.0
Overall		9,8	1.6	1.4	0.8

TABLE II
COMPARATIVE STUDY FOR ARABIC DIGITS RECOGNITION SYSTEMS (MALE SPEAKERS)

		Word Error Rate (%)			
	Digit	DHMM	MLP	RNN	GRNN
0	şifr	6	6	8.0	4.0
1	wa:ħid	12	2	2.0	0.0
2	?iθna:n	24	8	6.0	2.0
3	θala:θa	16	8	14.0	6.0
4	?arbaça	16	2	4.0	0.0
5	χamsa	0.0	0	4.0	0.0
6	sitta	12	0	0.0	0.0
7	Sabça	26	0	0.0	2.0
8	θama:nija	46	12	6.0	6.0
9	Tisça	14	0.0	0.0	0.0
Overall		17.2	3.8	4.4	2

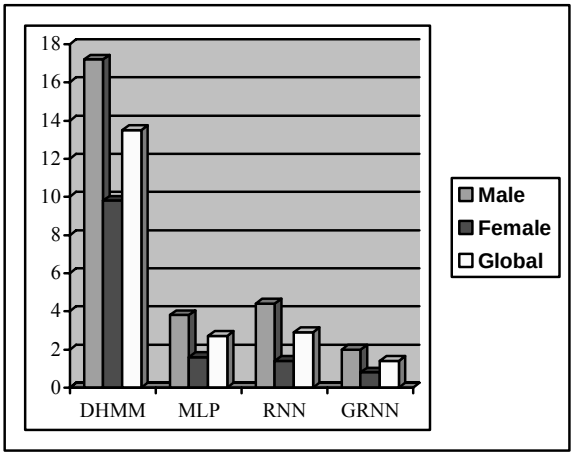

Fig. 5 Comparative performances (WER in %)

V. RESULTS AND DISCUSSION

At the heart of GRNN is the kernel function. The output of kernel function is an estimate of how likely the unknown pattern of word belongs to that distribution. The training data are simply copied into the hidden layer contain one word features from the training set. When presented with features of unknown word, the distance between the unknown word and each word in the hidden layer is computed and passed through a kernel function.

Arabic digits are polysyllabic. They are more lengthened than the English or the French digits. Because intra- and inter-speaker variability some digits have been pronounced in various manners. This causes a significant error in discrimination phase. As it is shown in tables I and II, some digits have been recognized without error by GRNN based system. The HMM is unable to correctly recognize the digit "7" and "8" which are pronounced in various manners. The error rate was respectively 8%, 8% and 12% for the digit "2", "3" and "8" pronounced by male speakers using MLP and respectively 6%, 14% and 6% using RNN. This error is less significant (respectively 2%, 6% and 6%) if we use nonparametric regression (see Table II).

The WER is reduced by 15.2% over the HMM baseline system for male speakers and by 9% for female speakers. The WER could be reduced respectively from 1% to 2.2 % for female and male speakers over the MLP and from 0.6% to 2.2% over the RNN based systems.

It is noticed that the error rate for female speakers is lower than the error rate for male speakers in the both neural networks classifiers. This difference is smaller with GRNN.

The optimization of the smoothing factor is critical to the performance of the GRNN and is usually found through iterative adjustment and the cross-validation procedure. Small values ($\sigma<10$) make the neural network not able to recognize large testing sets. In this case, the majority of digits have been unrecognized. In the other hand when the smoothing factor increases above a certain value ($\sigma>55$ in our case) the error rate increases significantly because the output is made independently of the new input presented in testing phase. Smooth factor σ was determined experimentally. Suitable interval for our application is $14<\sigma<20$ and $\sigma=15$ is a convenient value.

The primary advantage to the GRNN is the speed at which the network can be trained. The training of the GRNN is performed at one pass. The GRNN give the best recognition rate and it is the fastest algorithm though a large dimension of input vectors has been used. It requires a large size of memory because all new examples presented in the training set are memorized.

VI. CONCLUSION

In this work we have proposed GRNN adaptation scheme for classification task in ASR. The efficiency of this choice has been shown in comparative study with the MLP, the RNN and the Discrete Hidden Markov Model. The use of a nonparametric density estimator with an appropriate smoothing factor improves the generalization capability of the neural network. Experimental results obtained with large corpora have shown that the proposed model has several advantageous characteristics such as fast learning capability, flexibility network size, and robustness to speaker variability (ability to recognize the same words pronounced in various manners). GRNN is a successful alternative to the other neural networks and DHMM. It is therefore suitable to be applied in ASR systems.

REFERENCES

[1] L. Rabiner, "A Tutorial on hidden Markov model and selected applications", *in Proc. of IEEE*, Vol. 77, n°2, 1989.
[2] C. M. Bishop, *Neural Networks for Pattern Recognition,* Oxford University Press, 1995.
[3] S. Haykin , *Neural Networks: A Comprehensive Foundation*", 2nd ed., Cliffs, NJ,1999.
[4] R. P. Lippman, "Review of Neural Networks for Speech Recognition" *Neural Computation*, n°1, pp.1-38, 1989.
[5] F. Jelinek, *Statistical Methods for Speech Recognition*, Cambridge, Massachusetts, MIT Press, 1997.
[6] A. Waibel, T. Harazawa, G. Hinton, K. Shakano and K.J. Lang, "Phoneme recognition using Time-Delay Neural Networks," *IEEE Trans. On ASSP,* vol. 37, n°3, pp. 328–339, March 1989.
[7] K. Lang, A. Waibel, and G. Hinton, "A Time Delay Neural Network architecture," *Neural Networks*, vol. 3, pp. 333-34, 1990.

[8] H. Bourlard, and N. Morgan "Connexionnist techniques", available: http://cslu.cse.ogi.edu/HLT survey/ch11node7.html, March 2003.
[9] H. Bourlard and C.J. Wellekens "Links between Markov models and multilayer perceptrons" *in IEEE Trans on Pattern Analysis and Machine Intelligence,* Vol 2, pp. 1167-1178, 1990.
[10] K. Kirschoff et al., "Novel approach to Arabic speech recognition," Final Report from the JHU Summer School Workshop, 2002.
[11] S.A. Selouani and J. Caelen "Arabic word recognition by classifiers and context", *Journal of Computer Science and Technology*, Vol.20, N°3, pp.402-410. May 2005.
[12] H. Bahi and M. Sellami,"Combination of vector quantization and HMM for Arabic speech recognition ", *ACS/ IEEE Int. Conf. on Computer System and Applications AICCSA'01*, pp.96-101, Beirut, Lebanon, 2001.
[13] T. Cacoulos "Estimation of a multivariate density" *Ann. Inst. Math. Tokyo,* Vol. 18, n°2, pp. 179–189, 1966.
[14] D. F. Specht "A General Regression Neural Networks" *IEEE Trans. on Neural Networks,* Vol. 2, n°6, pp. 568–576, Nov. 1991.
[15] D.F. Specht, *Probabilistic Neural Networks and General Regression Neural Network*s, FuzzyLogic and Neural Network Handbook, Chap3. Mac Grow Hill inc. 1995.

A. Amrouche was born in Algeria. He received the diploma of electronics engineer "Ingenieur d'Etat" from National Polytechnic School from Algiers in 1980 and the "Magister" degree in 1995. Since 1982, he is in USTHB and currently he is senior lecturer at the Faculty of Electronics and Computer Sciences and scientist researcher in Speech communication laboratory. His research interests include pattern recognition, speech processing, multilingual speech recognition, neural networks…

J. M. Rouvaen was born in 1947 in France. He received M.S degree in 1968 and his Ph.D "Doctorat d'Etat" in 1971 from the University of Valenciennes (France). He his now Professor of electronics at ENSIAME, an engineering school of university of Valenciennes and he is the head of Radio-communications, Detection and Signal processing research group at OAE–IEMN Institute (France). His primary interests are in nonlinear phenomena, speech processing, and signal processing for communication systems…