Arabic Speech Recognition Using Recurrent Neural Networks

M. M. El Choubassi, H. E. El Khoury, C. E. Jabra Alagha, J. A. Skaf and M. A. Al-Alaoui

Electrical and Computer Engineering Department

Faculty of Engineering and Architecture – American University of Beirut

Beirut 1107 2020, P.O. Box: 11-0236, LEBANON

adnan@aub.edu.lb

Abstract

In this paper, a novel approach for implementing Arabic isolated speech recognition is described. While most of the literature on speech recognition (SR) is based on hidden Markov models (HMM), the present system is implemented by modular recurrent Elman neural networks (MRENN).

The promising results obtained through this design show that this new neural networks approach can compete with the traditional HMM-based speech recognition approaches.

Keywords

Arabic speech recognition, cepstral feature extraction, vector quantization, isolated word, speaker independent, modular recurrent Elman neural network.

1 INTRODUCTION

The speech recognition problem may be interpreted as a speech-to-text conversion problem. A speaker wants his/her voice to be transcribed into text by a computer.

Automatic speech recognition has been an active research topic for more than four decades. With the advent of digital computing and signal processing, the problem of speech recognition was clearly posed and thoroughly studied. These developments were complemented with an increased awareness of the advantages of conversational systems. The range of the possible applications is wide and includes: voice-controlled appliances, fully featured speech-to-text software, automation of operator-assisted services, and voice recognition aids for the handicapped….

Different approaches in speech recognition have been adopted. They can be divided mainly into two trends: hidden Markov model (HMM) and neural networks (NN). HMMs have been the most popular and most commonly used approaches while NN haven't been used for SR until recently.

The NN approach for SR can be divided into two main categories: conventional neural networks (MLP, RBF, SOM/LVQ, etc.) and recurrent neural networks (RNN). Conventional neural networks have proven to be good pattern classifiers but they haven't been able to compete with the results obtained by HMM. RNNs have been widely used in various sequence processing tasks such as time-series prediction, grammatical inference, dynamic system identification, etc. However, they have not attained the same level of success in speech recognition as in other applications.

The novelty in our approach is the use of a small RNN for each word in the vocabulary set instead of a unique large RNN for the entire set.

There are many distinctive features in our speech recognition system. The system:

- is implemented using neural networks
- is designed for Arabic language recognition
- recognizes a limited set of isolated words
- is female speaker-independent and performs favorably for male speakers.
- is tolerant to moderate noise

In the following sections, we present the implementation stages of our system. In the first stage of the design, the speech is appropriately processed to be input to the neural networks. By this we imply feature extraction achieved through modeling the human vocal tract using linear predictive coding which is then converted to the more robust cepstral coefficients. To compress those features, vector quantization is used, and a codebook is created using the K-means algorithm. This is discussed in Section 2.

The second stage of the design is to train the system for different utterances of the words in the vocabulary set. These utterances should constitute a good sample set of the various conditions and situations in which the word may be pronounced.

This training was implemented on Elman neural networks using the back propagation algorithm with momentum and variable learning rate. This is discussed in Section 3.

The last stage of our project is testing. The system was tested under different conditions: noisy and clean environments, speakers who trained the system and new speakers. The results are presented in Section 4.

2 FEATURE EXTRACTION

Speech acquisition begins with a person speaking into a microphone or telephone. This act of speaking produces a sound pressure wave that forms an acoustic signal. The microphone or telephone receives the acoustic signal and converts it to an analog signal that can be understood by an electronic device. Finally, in order to store the analog signal on a computer, it must be converted to a digital signal.

2.1 Pre-emphasis

In general, the digitized speech waveform has a high dynamic range and suffers from additive noise. An example of such a waveform is shown in the upper part of Figure 1.

In order to reduce this range pre-emphasis is applied. By pre-emphasis [5], we imply the application of a high pass filter, which is usually a first-order FIR of the form:

$$H(z) = 1 - az^{-1} \quad 0.9 \le a \le 1.0 \tag{1}$$

The pre-emphasizer is implemented as a fixed-coefficient filter or as an adaptive one, where the coefficient a is adjusted with time according to the autocorrelation values of the speech. The pre-emphasizer has the effect of spectral flattening which renders the signal less susceptible to finite precision effects (such as overflow and underflow) in any subsequent processing of the signal. The selected value for a in our work was 0.9375.

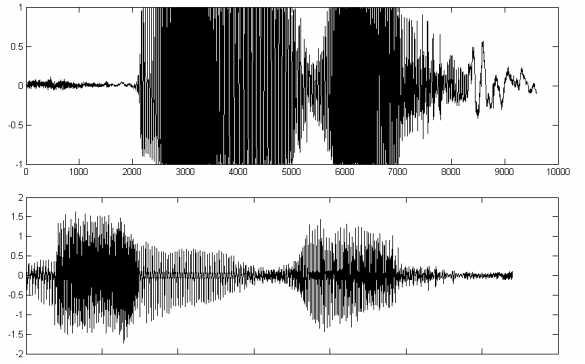

Figure 1. Speech Waveform of the word "*Manzel*" before and after pre-emphasis and endpoint detection

2.2 Endpoints detection

The goal of endpoint detection is to isolate the word to be detected from the background noise. It is necessary to trim the word utterance to its tightest limits, in order to avoid errors in the modeling of subsequent utterances of the same word. As we can see from the upper part of figure 1, a threshold has been applied at both ends of the waveform. The front threshold is of value 0.12 whereas the end threshold value is 0.1. These values have been obtained after observing the behavior of the waveform and noise in a particular environment.

2.3 Frame blocking

Since the vocal tract moves mechanically slowly, speech can be assumed to be a random process with slowly varying properties [5]. Hence, the speech is divided into overlapping frames of 20ms every 10ms. The speech signal is assumed to be stationary over each frame and this property will prove useful in the following steps.

2.4 Windowing

To minimize the discontinuity of a signal at the beginning and end of each frame, we window each frame to increase the correlation of the linear predictive coding (LPC) spectral estimates between consecutive frames [5]. The windowing tapers the signal to zero at the beginning and end of each frame. A typical LPC window is the Hamming window of the form:

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

2.5 LPC analysis

A speech recognizer is a system that tries to understand or "decode" a digitized speech signal. This signal, as first captured by the microphone, contains information in a form not suitable for pattern recognition. However, it can be represented by a limited set of features relevant for the task. These features more closely describe the variability of the phonemes (such as vowels and consonants) that constitute each word.

The feature measurements of speech signals are typically extracted using one of the following spectral analysis techniques: filter bank analyzer, LPC analysis or discrete Fourier transform analysis. Since LPC is one of the most powerful speech analysis techniques for extracting good quality features and hence encoding the speech signal at a low bit rate, we selected it to extract the features of the speech signal [5].

The LPC coefficients a_i are the coefficients of the all-pass transfer function $H(z)$ modeling the vocal tract, and the order of the LPC, p, is also the order of $H(z)$ defined as follows:

$$H(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{3}$$

LPC was implemented using the autocorrelation method. A drawback of LPC estimates is their high sensitivity to quantization noise; cepstral coefficients, which can be derived from the LPC coefficients, have lower susceptibility to noise, and were adopted instead as explained below.

2.6 LPC conversion to Cepstral coefficients

The features used in this system are the weighted LPC-based cepstral coefficients, which are the coefficients of the Fourier transform representation of the log magnitude spectrum.

Table 1. Cepstral coefficients determination

Iterations m=1,2,…..,p	$c_m = \alpha(m) + \sum_{k=1}^{m-1}\left[\left(\frac{k}{m}\right)\alpha(m-k)c_k\right]$
Iterations m=p+1,…..,q	$c_m = \sum_{k=m-p}^{m-1}\left[\left(\frac{k}{m}\right)\alpha(m-k)c_k\right]$

Table 1 shows an iterative algorithm for the determination of the cepstral coefficients from the LPC coefficients. The cepstral order q is generally chosen to be greater than the LPC order p. A rule of thumb is to set q to 3/2 of the LPC order p. In our system, we have chosen p to be 8, therefore q was set to 12 accordingly [5].

To decrease the sensitivity of high-order and low-order cepstral coefficients to noise, the obtained cepstral coef-

ficients are multiplied by an appropriate weighting which is a window with the following equation:

$$w_m = \left[1 + \frac{q}{2}\sin\left(\frac{\pi m}{q}\right)\right] \qquad 1 \le m \le q \qquad (4)$$

This results in what is known as the weighted cepstral coefficients [5].

Figure 2 illustrates clearly the advantage of weighted cepstral representation, i.e. its superior tolerance to noise when compared to LPC. The plots represent the weighted cepstral coefficients generated from seven distinct utterances of the sound "aa".

It is obvious from the figure that there is little variation between the extracted cepstral coefficients for the seven utterances. Hence, this demonstrates the reliability and consistency of these coefficients.

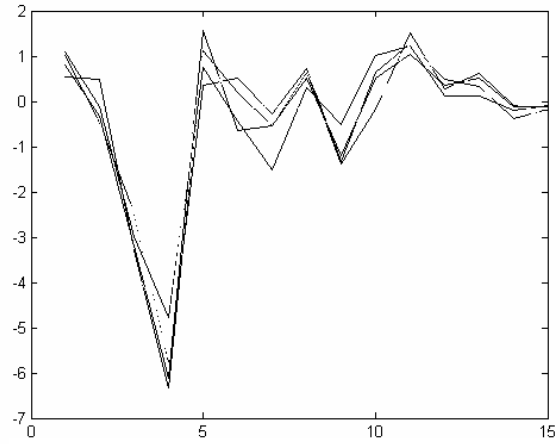

Figure 2. Weighted cepstral coefficients generated from seven distinct utterances of the sound "aa".

2.7 Vector quantization

Optimization of the system is achieved by using vector quantization in order to compress and subsequently reduce the variability among the feature vectors derived from the frames. In vector quantization, a reproduction vector (codevector) in a pre-designed set of K vectors (codebook) approximates each feature vector of the input signal: the feature vector space is divided into K regions and all subsequent feature vectors are classified into one of the corresponding codebook-elements (i.e.: the centroids of the K regions) according to the least distance criterion (Euclidian distance).

The best results were obtained using an 80-element codebook, generated by Lloyd's K-means algorithm applied on a long speech sample consisting of the words in the vocabulary set [5]. The output of this last stage is the final feature used throughout.

3 NEURAL NETWORKS IMPLEMENTATION

The training and classification of the extracted features can be implemented in several ways: using HMM, NN or a hybrid HMM-NN. One of the most successful and popular speech models discussed in the literature is the *first order* HMM, a simplified stochastic process model based upon the Markov chain. Despite the scarcity of the literature available on the implementation of SR using NN, we have adopted a MRENN model and we have found that it can achieve results as good as the HMM model.

Neural networks [2,3,4] attempt to mimic some or all of the characteristics of biological neurons that form the structural constituents of the brain.

A neural network can:

- Learn by adapting its synaptic weights to changes in the surrounding environments;
- Handle imprecise, fuzzy, noisy, and probabilistic information;
- Generalize from known tasks or examples to unknown ones.

3.1 Feedforward vs. recurrent networks

Neural network architecture can be divided into two principal types: recurrent and non-recurrent networks. An important sub-class of non-recurrent NN consists of architectures in which cells are organized into layers, and only unidirectional connections are permitted between adjacent layers. This is known as a feedforward multilayer perceptron (MLP) architecture. This architecture is shown in Figure 3.

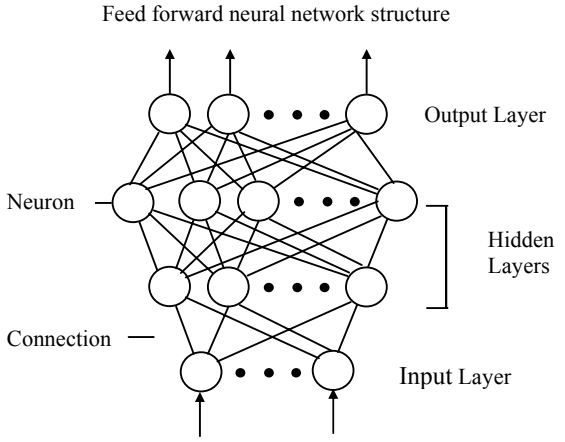

Figure 3. A possible architecture of a Neural Network (feedforward MLP)

On the other hand, recurrent neural networks are characterized by both feedforward and feedback paths between the layers. The feedback paths enable the activation at any layer to either be used as an input to a previous layer or be returned to that layer after one or more time steps.

It was believed that multilayered perceptrons are useful for SR because they can approximate the relationship between the inputs and outputs of a system. In a linear system, this would be described as the transfer function of the system. However, training a feedforward MLP consists of showing the network a set of input and output pairs of data, with no consideration given to their temporal relationship. Thus the data, and the resultant model, represent only the static model of the system. Of more use to a SR application is the dynamic model of the sys-

tem, which takes into account the way in which the system changes from one state to the next. While feedforward networks are useful for static data, the importance of recurrent networks lies within their ability to deal with dynamic and time-changing data.

3.2 Elman networks

In this paper, we used the Elman network [2,3,4], which is a special kind of a recurrent network. The Elman network, originally developed for speech recognition, is a two-layer network in which the hidden layer is recurrent. The inputs to the hidden layer are the present inputs and the outputs of the hidden layer which are saved from the previous time-step in buffers called context units.

Hence, the outputs of the Elman network are functions of the present state, the previous state (as supplied by the context units) and the present inputs. This means that when the network is shown a set of inputs, it can learn to give the appropriate outputs in the context of the previous states of the network.

The advantage of Elman networks over fully recurrent networks is that back propagation is used to train the network while this is not possible with other recurrent networks where the training algorithms are more complex and therefore slower.

In our SR system, we used a 24-10-1 Elman network. This network can be seen in Figure 4.

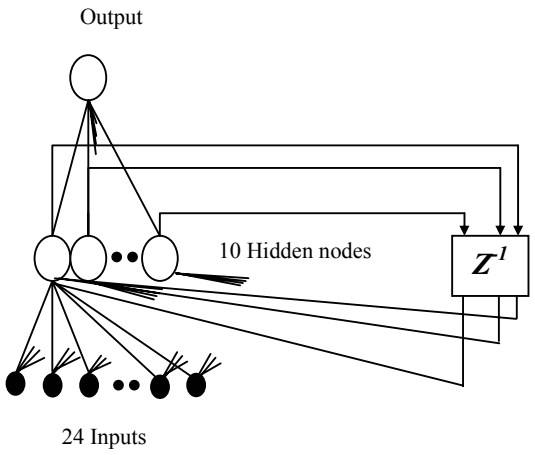

Figure 4. Architecture of an Elman Network

3.3 System architecture and training approach

Our SR system is modular, i.e. for each word in the vocabulary set, there is a separate Elman network. Modularity adopts a "divide-and-conquer" approach by dividing the complex problem at hand into many smaller and simpler problems [6].

The vocabulary set used is composed of 6 Arabic words: "manzel" (*house*), "hirra" (*cat*), "chajara" (*tree*), "tariq" (*road*), "ghinaa" (*singing*), "zeina" (*zeina)*.

The function of each network is to recognize its dedicated word only and to reject other words. This is why the training is divided into two steps: consistent training and discriminative training.

Consistent training is exposing the network to different utterances of the dedicated word, associated with linear targets with positive slope (as seen in Figure 5). Twelve utterances were obtained from each of four female speakers in a relatively clean environment.

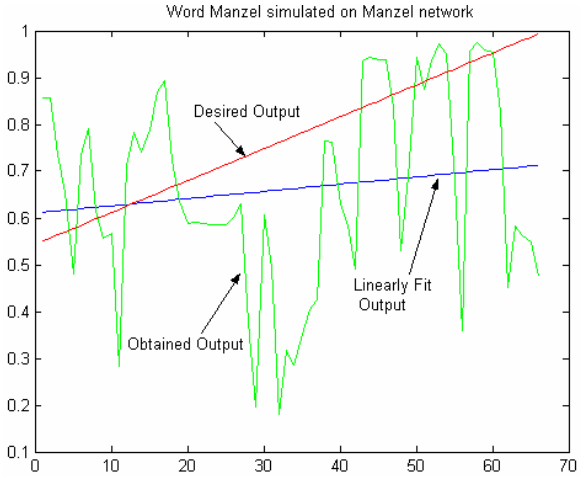

Figure 5. Outputs and target for the dedicated word "manzel" on its network

On the other hand, discriminative training is exposing the network to utterances other than that of the dedicated word, associated with linear targets with negative slopes (as seen in Figure 6). One utterance per word was obtained from each of four female speakers in a relatively clean environment.

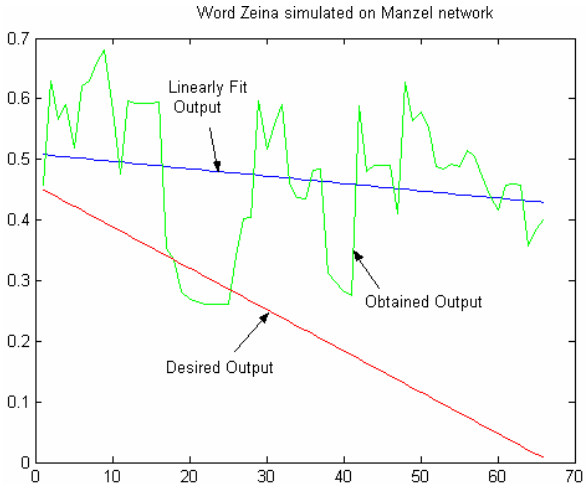

Figure 6. Outputs and target for the word "zeina" on the network dedicated to "manzel"

Hence, the training set of each network is composed of 48 consistent training utterances and 20 discriminative training utterances.

The training algorithm used is back-propagation with momentum and variable learning rate. Consistent training was performed after discriminative training because recurrent networks inherently "remember" the most recent training utterance applied to it.

After each training pass (100 epochs), the network is simulated on a validation set composed of 40 new utterances of the dedicated word and 80 new utterances from the remaining 5 words. The output obtained from the simulation of the network for an utterance is a non-linear curve. The decision-making criterion is the slope of the line obtained from the linear fitting of this curve.

The classification of an utterance, other than the dedicated word, is based on the comparison of its resulting slope s with the minimum slope s_m among the slopes obtained from all the utterances of the dedicated word.

- If $s > s_m$, then a classification error results because the network confused the tested utterance with the dedicated word.
- If $s < s_m$, no classification error occurred.

If the number of classification errors, i.e. the misclassified utterances of a word, is greater than a given threshold (taken to be 5), the network is retrained for the "worst-offender" (i.e. the utterance that resulted in the greatest slope) and for two consistent utterances selected randomly.

This iterative procedure is a variation of the "cloning" approach introduced by Al-Alaoui et al. in [1]. It converges to a network with a minimal number of classification errors.

After obtaining the six optimal networks, they are integrated into the final SR system.

When the SR system is exposed to any utterance of the vocabulary set, each network is simulated with this utterance. The network that results in the maximum slope is elected as the network of the resulting word.

4 RESULTS

The speech recognizer described in this paper was fully implemented in MATLAB, and was subjected to several test inputs. The obtained results are summarized in Table 2.

Sp.1, Sp.2, Sp.3 and Sp.4 are female speakers who provided the utterances for the training phase. They tested the system in moderate background noise.

Sp.5 is a female speaker whose utterances weren't used in the training phase. She tested the system in a relatively clean environment.

Sp.6 is a male speaker who tested the system in a relatively clean environment.

Table 2. Recognition rate for different speakers in different environments

	Manzel	*Hirra*	*Chajara*	*Tariq*	*Ghinaa*	*Zeina*
Sp.1	100%	90%	97%	99%	97%	95%
Sp.2	97%	97%	99%	99%	98%	98%
Sp.3	100%	90%	98%	99%	92%	98%
Sp.4	100%	95%	98%	99%	95%	97%
Sp.5	100%	95%	98%	98%	96%	97%
Sp.6	92%	85%	89%	91%	86%	87%

As can be seen from these results, the approach that we adopted gave promising recognition rates that can match, if not compete, with the ones usually obtained by HMM-based approaches.

ACKNOWLEGMENTS

We would like to thank Mr. Rony Ferzli and Mr. Mesrob Ohannessian for their constant help and support.

Our work would have been very difficult if it were not for the facilities provided to us by the American University of Beirut.

REFERENCES

[1] Al-Alaoui, M.A., Mouci, R., Mansour M.M., Ferzli, R., *A Cloning Approach to Classifier Training*, IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, vol.32, no.6, pp.746-752, (2002)

[2] Gurney, K., *An Introduction to Neural Networks*, UCL Press, University of Sheffield (1997).

[3] Morgan, D. and Scolfield, C., *Neural Networks and Speech Processing*, Kluwer Academic Publishers (1991).

[4] Picton, P. *Neural Networks*, Palgrave, NY (2000)

[5] Rabiner, L. and Juang, B. -H., *Fundamentals of Speech Recognition*, PTR Prentice Hall, San Francisco, NJ (1993).

[6] Tan Lee, P. C. Ching, L.W. Chan, *Isolated Word Recognition Using Modular Recurrent Neural Networks,* Pattern Recognition, vol. 31, no. 6, pp. 751-760 (1998)