“For Consideration of Young Scientist Award”
Category code: 8 (Data Mining, Neural Networks and Artificial Intelligence)

19th Annual Cum 4th International Conference of GAMS On “Advances in Mathematical Modelling to Real World Problems”

AUTOMATIC SPEECH RECOGNITION OF GUJARATI DIGITS USING ARTIFICIAL NEURAL NETWORK

Purnima Pandit[1], Shardav Bhatt[2], Priyank Makwana
pkpandit@yahoo.com, shardavb@gmail.com, priyankmak@gmail.com
[1 2] Department of Applied Mathematics, The M. S. University of Baroda, Vadodara.

Abstract

In this work we do Automatic Speech Recognition (ASR) for Gujarati digits using Artificial Neural Network (ANN). The feature extraction from the speech signals, for the required purpose, is done using Mel Frequency Cepstral Coefficients (MFCC). The extracted features are used in training of ANN, which later recognises the unknown speech signal. The recognised digit is outputted as an image of digit written in Gujarati language and its equivalent sign language image. Computations and experimental results are verified by the programs in MATLAB. Such an interface would be useful for person with disabilities and also for illiterate people.

Keywords

Automatic Speech Recognition, Gujarati digits, Mel Frequency Cepstral Coefficients, Artificial Neural Networks.

1. *Introduction*

Automatic Speech Recognition (ASR) is an interdisciplinary field and it is a process of automatically recognizing a natural human speech using a machine. The objective is to design an intelligent machine that can recognize the spoken word and comprehend its meaning. Enormous amount of research work has been done in this direction and we are still on the way [1]. Speech is a primary source of communication and language is a major tool. ASR is been done in various languages, but in local language like Gujarati it is in infant stages. Major motivation is to design a Speech User Interface in local language which can also be useful for people with disabilities. A Speech User Interface in local Indian Language will not required additional skills to use these modern gadgets. Even blind people, uneducated people can have advantage of technology due to Speech Recognition. Automatic Speech Recognition in local Indian languages can eliminate language barrier which majority of the Indian are facing and information can be shared easily by any person.

Automatic Speech Recognition of digits “One” to “Ten” spoken in Gujarati language using Dynamic Time Warping is done in [2]. In present paper, we do ASR using Artificial Neural Network (ANN). Significant features of the speech are extracted using Mel Frequency Cepstral Coefficients (MFCC). These feature vectors are used to train ANN for recognition of unknown speech.

The following section describes characteristics of Indian languages which are of vital significance in Speech Recognition for local languages. Section 3 and 4 describes MFCC and ANN respectively. Section 5 describes experimental results and section 6 concludes the paper with direction for future work.

2. *Characteristic of Indian Languages*

There are many scripts and dialects of Indian languages. Most of the scripts are phonetic in nature. According to Census 2001, India has 122 major languages and 2371 dialects. Out of these 122 languages, 22 languages are recognized constitutionally [3]. Indian languages are time-based syllabi like languages. Hence segmenting of Indian languages into phonemes is quite difficult task. Accent is not uniform within same languages. There are languages having more than one script and also there are languages having one script in common. Distinct articulatory places are needed to pronouns a word [4].

For phoneme based Speech Recognition, speech should be produced in a systematic way. As a result, better articulatory disciplines are required. As compared to the European languages, there are more fricatives and more retroflex consonants present in Indian Languages. The Gujarati language is based on Indo-Aryan language. Before going for phoneme based Speech Recognition we first experiment with isolated words recognition using Artificial Neural Network [4].

3. *Feature Extraction using Mel Frequency Cepstral Coefficients (MFCC)*

The first step in speech recognition is feature extraction. The feature from the speech signal is the initial process in the Speech Recognition. It reduces the dimensionality of the input vector and simultaneously discriminates the power of the signal. Main goal of feature extraction process is to obtain a sequence of vectors providing a compact representation of given input signal.

There are many feature extraction techniques like Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis, Linear Predictive Coding, Cepstral Analysis, Filter Bank Analysis, Mel Frequency Cepstral Coefficients (MFCC), Wavelets, Kernel based feature extraction etc [4]. Out of these, MFCC is most commonly used for extraction of features because it shows high accuracy results for clean speech. They were introduced by Davis and Mermelstein in 1980 [5]. MFCC are best approximations of human ears since they are based on human hearing perception. Human ears are more responsive to the lower frequencies, they cannot perceive frequencies higher than 1000 Hz. Lower frequency components of the speech signal are more important than the higher frequency components [5].

For feature extraction process, first we have to detect voiced or unvoiced part in the signal using short time energy and zero crossing rates. Then feature extraction is done for the voiced part of signal only. It is usually performed in three stages. In first stage, a spectro-temporal analysis is done. Here the raw features are generated which represents envelope of power spectrum of short speech interval. Second stage compiles the extended feature vector composed of static and dynamic features. In last stage, these extended feature vectors are transformed into more compact vectors and they are sent to the recognizer. The steps for finding MFCC are summarized in Fig. 1.

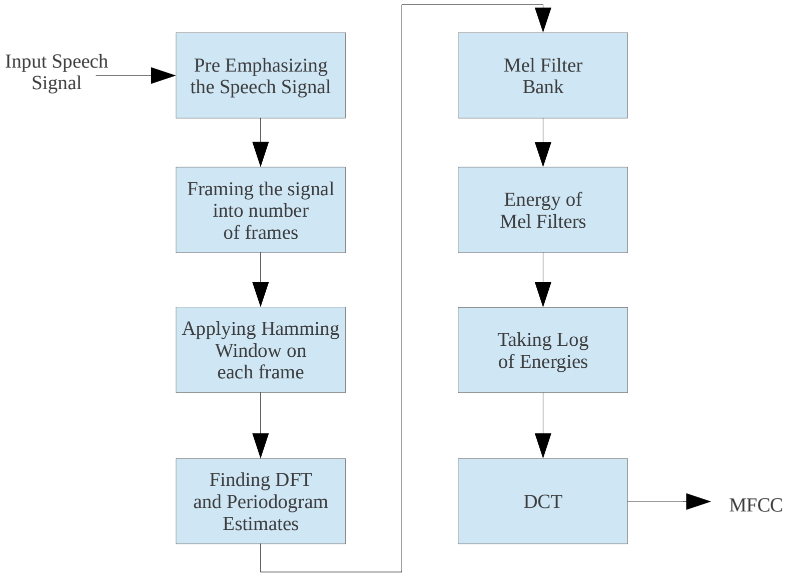

Input Speech Signal
Pre Emphasizing the Speech Signal
Framing the signal into number of frames
Applying Hamming Window on each frame
Finding DFT and Periodogram Estimates
Mel Filter Bank
Energy of Mel Filters
Taking Log of Energies
DCT
MFCC

Fig. 1: Steps to Calculate MFCC

Each of these steps is described below in detail. Let $x(n)$ be a voiced part of the recorded digital speech signal.

Step 1. Pre-Emphasizing

The recorded digital signal is passed through a first order FIR filter using the equation (1) below. So, we get an emphasized version of recorded signal $s(n)$.

$$s(n) \;=\; x(n) - Ax(n-1);\; 0 \leq A \leq 1 \tag{1}$$

Usually the value of A is taken as 0.95. It means that 95% of each sample is originating from its previous sample [1]. Due to this, the signal becomes spectrally flat and less susceptible to finite precision effect.

Step 2. Framing

It is difficult to analyze the speech signal entirely so we analyze it frame by frame. Signal is divided into number of frames. Usually frame size is 10-40 milliseconds (ms) in time domain. The framing is done in such a way that each frame overlaps on its adjacent frame. Hence now from a pre-emphasized signal $s(n)$ we have $s_i(n)$, where n is the number of sample in each frame varying from 1 to N in i^{th} frame.

Step 3. Windowing

The next step is to window each individual frame of the signal. This step minimizes the signal discontinuities at beginning and end of each frame. Usually Hamming window is used for this step [1]. A typical window used here is the hamming window defined by

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

The result of windowing is the signal

$$s_i(n)w(n), 0 \leq n \leq N-1 \tag{3}$$

Step 4. Discrete Fourier Transform (DFT) and Periodogram Estimates
The next step is to take Discrete Fourier Transform (DFT) of the windowed signal. We apply this step to transform each frame of the signal from time domain to the frequency domain. For this, the Fast Fourier Transform (FFT) algorithm is used. Hence for all i frames we have,

$$s_i(k) = \sum_{n=1}^{N} s_i\,(n)w(n)\; e^{\frac{-j\,2\pi kn}{N}};\; 1 \leq k \leq K \tag{4}$$

Here K is the number of samples is DFT. The Periodogram estimates of the power spectrum are

$$p_i(k) = \frac{1}{N}\;|s_i(k)|^{\,2} \text{ for each } i \tag{5}$$

Step 5. Mel Filter Bank
The signal is perceived by human ear linearly for the frequency less than 1000 Hz and on logarithmic scale for frequency greater than 1000 Hz. So we convert the frequency scale into Mel scale using equation (6).

$$M(f) = 1125\; ln\left(1 + \frac{f}{700}\right) \tag{6}$$

The Mel scale associates this perceived frequency $M(f)$ of the speech signal with the actual measured frequency f.

After converting frequencies into Mel scale, now we apply a Filter bank consisting of 20-40 triangular shaped filters on Mel Scale. They are non-uniformly spaced due to Mel scale, so there are more filters in low frequency region. The triangular filters used here are defined as

$$z_m(k) \;=\; \begin{cases} \dfrac{k - f(m-1)}{f(m) - f(m-1)};\; f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)};\; f(m) \leq k \leq f(m+1) \\ \qquad\quad 0 \qquad\quad ;\quad otherwise \end{cases} \tag{7}$$

Here m is the number of filters used and $f(m)$ are the Mel spaced filters. The energy of filter bank is calculated using,

$$\widetilde{p_m} = \sum_{k=0}^{K/2} p_i(k) z_m(k) \tag{8}$$

Here $\widetilde{p_m}$ is the Mel spectrum obtained from the original spectrum $p_i(k)$.

Step 6. Calculating MFCC
Now to come back to the time domain from the frequency domain, we take Discrete Cosine Transform (DCT). This gives a set of numbers for each frame called Cepstral Coefficients. They are used as input vectors to Artificial Neural Network for further processing. MFCC are obtained using,

$$\widetilde{c_n} \;=\; \sum_{k=1}^{m} (log\widetilde{p_k})\; cos\left\{n\left(k - \frac{1}{2}\right)\frac{\pi}{2}\right\} \tag{9}$$

Here n is the number of Cepstral Coefficients in each frame.

4. Artificial Neural Networks

Artificial Neural Networks (ANN) are massively parallel, distributed processing systems representing a computational technology built on the analogy to the human information processing system. They are massively connected networks of simple processing elements called neurons. They have a natural propensity to learn and save the knowledge to make it available for use. ANNs can be used for classification, pattern recognition and function approximation.

ANN plays a primary role in contemporary artificial intelligence and machine learning. The use of ANN in function approximation resulted from the following facts:

i. Cybenko and Hornik proved that the multi-layered Neural Network is universal approximator. It can approximate any continuous function defined on compact set.
ii. The Back-propagation algorithm used for the training of the feed-forward Neural Networks with the hidden layers.

In the real practical situations when there is the lack of the mathematical model for the system, the ANN can be trained with the help of Input-Output pairs without fitting any kind of model to the system.

The fundamental element of neural network is neuron. We will refer to neuron as an operator, which maps $\mathbb{R}^n \to \mathbb{R}$ and is explicitly, described by the equation (10).

$$x_j = \Gamma\left(\sum_{i=1}^{n} w_{ji} u_i + w_0\right) \tag{10}$$

Here, $U^T = [u_1, u_2, \dots, u_n]$ is input vector, $W_j^T = [w_{j1}, w_{j2}, \dots, w_{jn}]$ is weight vector to the j^{th} neuron and w_0 is the bias. $\Gamma(\cdot)$ is a monotone continuous function, $\Gamma: \mathbb{R} \to (-1,1)$ (For example $tanh(\cdot)$ or $signum(\cdot)$). Such neurons are interconnected to form a network.

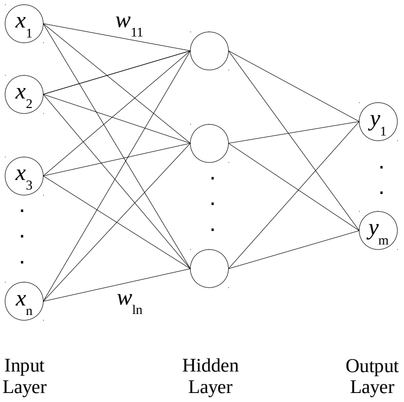


Fig. 2: A Multi layered Feed-forward Network

In the Feed-forward network, neurons are organized in layers $h = 0, 1, \dots, L$. It is common practice to refer to the layer $h = 0$ as the input layer and $h = L$ as the output layer and to all other as hidden layers. A neuron at layer h receives its inputs only from neurons in the $h - 1$ layer. The output of the i^{th} element in the layer h is given by

$$y_i^h = \Gamma\left(\sum_j w_{i,j}^h \, y_j^{h-1} + w_{i,0}^h\right) \tag{11}$$

Here, $\left[w_i^h\right]^T = [w_{i,0}^h, w_{i,1}^h, \dots, w_{i,n_{L-1}}^h]$ is the weight vector associated with i^{th} neuron at the h^{th} layer. Such a family of networks with n_i neurons at the i^{th} layer will be denoted by $N_{n_0, n_1, \dots, n_L}^L$.

The general learning rule for weight updating is $W(k+1) = W(k) + \Delta W(k)$.

Here, $W(k+1)$ is weight in $(k+1)^{th}$ step, $W(k)$ is weight in $(k)^{th}$ step and $\Delta W(k)$ is change in weight in $(k)^{th}$ step. The change in weight for supervised delta rule is given by

$$\Delta w_{lj} \;=\; \eta\, f^{'}\left(\sum_{i=0}^{n} w_{ji}\, x_i\right) f\left(\sum_{i=0}^{n} w_{ji}\, x_i\right) \qquad (12)$$

The training in the Feed-forward type Multilayered neural network is done using back-propagation algorithm. Back-propagation algorithm is an extension of delta rule for training for multilayer feed-forward networks.

5. *Experimental Work and Results*

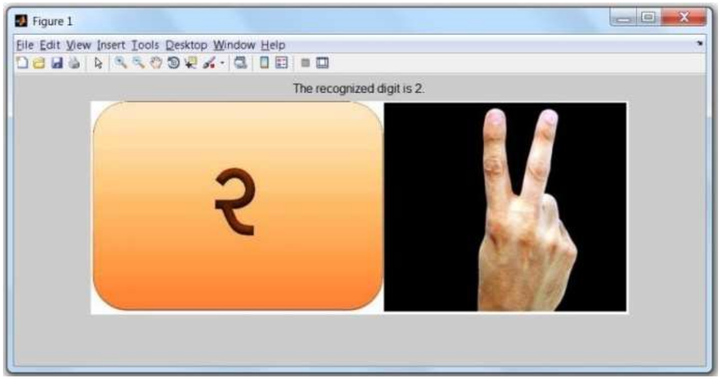
In our experiment, first we have recorded the voice of ten speakers, out of which 5 were male and 5 were female. The recording consists of spoken digits “One” to “Ten” in Gujarati i.e. “*Ek*” to “*Dus*”. So, we have 100 sound wave files. The recording was done in laboratory with almost nil environmental noise, using 'Audacity' software with sampling rate 16,000 samples per second and mono channel.

These 100 wave files are converted from analog to digital using inbuilt function *wavread*() of MATLAB. The output of this function is a column vector which has entries, representing the amplitude of recorded sound. The wave file for spoken Gujarati Digit “Two”, after using *wavread*() function, gave a column vector having 6971 elements for 2nd speaker. Thus, we have 100 such column vectors, 10 column vectors for each digit. The number of elements in each of these column vectors need not be equal so they are made equal in length using our defined insertion algorithm. The missing elements are inserted at equally spaced points and their values are the average value of their neighboring amplitudes. It is implemented in MATLAB by a function *myinsertion.m*. 10 column vectors for each digit are made equal in length using this function. The longest column vector for digit “Two” is a column vector having 9027 elements. The other nine column vectors for digit “Two” are less than 9027 and hence they are made of this length by including the missing elements.

Then the Mel Frequency Cepstral Coefficients (MFCCs) of each of these column vectors is obtained using function *mfcc.m*. This function gives column vector as an output having MFCCs of corresponding digit stored in it. For digit “Two”, all the column vectors having MFCCs are equal in length due to application of insertion algorithm. We have thus 100 column vectors (10 digits, 10 speakers) of MFCCs, which are not necessarily equal in length. In our experiment the longest column vector is corresponding to digit “One” having 603 elements, hence all remaining 99 column vectors of MFCC are made of length 603 by zero padding, i.e. including additional zeros at the end for missing values.

The MFCCs obtained in a manner described above were used as inputs to the network with architecture $N^{2}_{603,50,10}$. In input layer, we have 603 neurons, and 100 such input patterns are there all are having 603 neurons. By trial and error we came to selection of 50 neurons in hidden layer giving best performance. The desired output of this network has 10 neurons having binary values. Out of these 10 neurons, only one neuron has value ‘1’. For e.g. the value of second neuron, in output layer, for digit “Two” is ‘1’, rest of all neurons has value ‘0’. Such network was trained using back propagation algorithm with thumb rule of 80:20 ratio for training and testing patterns. The output for the recognized digit is the image of corresponding digit in Gujarati language as well as its equivalent Sign language form. The sample for the recognized digit “Two” is shown in Fig. 3.

We achieved 100% success for recognition of known speech which is present in neural network training. For an unknown speech which is not present in training set, we achieved 74% success of recognition.



The recognized digit is 2.

Fig. 3: The Sample for the recognized digit "Two" in Gujarati and Sign language respectively

6. *Conclusion and Future Work*

Our work shows that, the approach of Artificial Neural Network for Speech Recognition of isolated digits is worth. Since it generalizes and gives good results for testing unknown speech patterns. However, literature mentions difficulty in using Artificial Neural Network for continuous sentence, where the Speech recognition is phoneme based. Hence other techniques will have to be explored for phoneme based ASR.

References

1. Rabiner, L., Juang, B. H., Yegnanarayana, B., 2010, "Fundamentals of Speech Recognition", Second Edition, *Pearson Education*.

2. Pandit, P., Bhatt, S., 2014, "Automatic Speech Recognition of Gujarati Digits using Dynamic Time Warping.", *International Journal of Engineering and Innovative Technology*, 3(12), pp. 69-73.

3. Census India Data 2001 *http://www.censusindia.gov.in/Census_Data_2001/Census_data_online/Language/data_on_language.html*

4. Hemakumar, G., Punitha, P., 2013, "Speech Recognition Technology: A Survey on Indian Languages", *International Journal of Information Science and Intelligent Systems"*, 2(4), pp. 1-38.

5. Davis, S., Mermelstein, P., 1980, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 357-366.

6. Zurada, J. M., 2004, "Introduction to Artificial Neural Systems", First Edition, *Jaico Publishing House*.

7. Jain, A. K., 1996, "Artificial Neural Networks: A Tutorial", *Michigan State University.*