DE GRUYTER
OPEN

JAISCR, 2015, Vol. 5, No. 1, pp. 59 – 70
10.1515/jaiscr-2015-0019

A CLASS OF NEURO-COMPUTATIONAL METHODS FOR ASSAMESE FRICATIVE CLASSIFICATION

Chayashree Patgiri[1], Mousmita Sarma[2] and Kandarpa Kumar Sarma[3]

[1]*Department of Applied Electronics and Instrumentation, GIMT*
al., Guwahati-781017, Assam, India

[2]*Department of Electronics and Communication Engineering, Gauhati University*
al., Guwahati- 781014, Assam, India

[3]*Department of Electronics and Communication Technology, Gauhati University*
al. Guwahati- 781014, Assam, India

Abstract

In this work, a class of neuro-computational classifiers are used for classification of fricative phonemes of Assamese language. Initially, a Recurrent Neural Network (RNN) based classifier is used for classification. Later, another neuro fuzzy classifier is used for classification. We have used two different feature sets for the work, one using the specific acoustic-phonetic characteristics and another temporal attributes using linear prediction cepstral coefficients (LPCC) and a Self Organizing Map (SOM). Here, we present the experimental details and performance difference obtained by replacing the RNN based classifier with an adaptive neuro fuzzy inference system (ANFIS) based block for both the feature sets to recognize Assamese fricative sounds.

1 Introduction

Over the years, soft computing tools like Artificial Neural Network (ANN), fuzzy systems and their combinations are used as effective tools for signal processing and pattern recognition. This is because these acquire knowledge from the environment, hold back the learning and use it subsequently. Due to its inherent capability of dealing with nonlinear, dynamic systems, Artificial Neural Networks (ANN) are used as reliable classifiers. Fuzzy systems are suitable for tracking minute variations in input patterns. While fuzzy systems generate qualitative and knowledge based mechanisms for decision making, ANNs are non- parametric classification with the ability to effectively learn given patterns. These two are combined to form hybrid systems like Fuzzy Neural System (FNS) and Neuro Fuzzy System (NFS). NFS blocks demonstrate the ability to acquire numeric-qualitative, expert level decision making and generate efficient adaptability and robustness while handling uncertain processes or situations. Advantages of ANN and fuzzy systems in form of NFS are found to be robust, adaptive and expert systems for decision making for uncertain processes like speech recognition. Here, we propose an approach for classifying fricative sounds using two different neuro computational techniques, one with a Recurrent Neural Network (RNN) and the other with an Adaptive Neuro Fuzzy Inference System (ANFIS). The RNN is a form of ANN with feedback connections between output, input and different layers. The ANFIS is an adaptive NFS system with a rule base supported inference engine. Our objective is to ascertain the suitability of application of an ANFIS based classifier for temporal signal classification. This is driven by the fact that fuzzy based systems are able to track minute variations in an environment invested with uncertainty. Fuzzy in combination with ANN acquires both qualitative and quantitative processing ability which is expected to aid speech based applications.

Among the supervised learning ANNs, the RNNs have the dynamic structure with a capability of learning temporal information and hence are suitable for speech based applications. The key difference of RNN compared to the Multi Layer Perceptron (MLP) (which only hass feedforward paths) is the contextual processing which circulates the most relevant portion of the information among the different layers of the network and the constituent neurons. Further, in many situations, due to inversion in the applied patterns while performing the contextual processing, differential mode learning in the local level of neurons enables the RNN to consider only the most relevant portion of the data. With different types of activation functions at different layers of the network, contextual and differential processing is strengthened. For example, in a three hidden layer RNN, if one layer has tan-sigmoidal, the next with log-sigmoidal and the last with tan-sigmoidal activation function, the combination enables better learning. The least correlated portion of the patterns are retained and circulated and the portions with similarity are discarded. As a result the RNN becomes a fast learner and tracks time dependent variations. The RNN uses feedforward and feedback paths to track finer variations in the input. The feedback paths are sometimes passed through memory blocks which enable delayed versions of the processed output to be considered for processing. These variations can be due to the time dependent behavior of the input patterns. So while the MLP is only able to do discrimination between applied samples, the RNN is able to distinguish classes that show time variations. For the above mentioned attributes, RNNs are found to be suitable for application like speech recognition [1] [2]. RNNs have been first applied to speech recognition in [3]. Other important works include [4], [5], [6], [7], [8], [9], [10].

Due to the uncertain nature of cognitive problems, fuzzy logic and fuzzy inference systems (FIS) are suitable tools for dealing with pattern recognition issues with subtle and random variations. ANFIS is one of the best tradeoff between ANN and fuzzy systems. Its capabilities are obtained from the smoothness due to the fuzzy clustering interpolation and adaptability provided by the backpropagation learning algorithm of ANN. The ANFIS employs a hybrid-learning algorithm, which is a combination of the recursive least-squares (RLS) method and the back propagation gradient descent method for training Sugeno-type FIS membership function parameters to replicate the given training data set [24]. Use of NFS approach for speech recognition has recently been reported in contribution like [11], [12], [13], [14] and [15].

Speech consists of sequences of sounds. Phonemes are the smallest distinguishable meaningful unit of the speech signal which is an abstract representation at some cognitive level. Fricatives are consonant phonemes produced with a very narrow constriction in the oral cavity. There is a rapid flow of air through the constriction, creating turbulence in the flow. The random velocity fluctuation in the flow can act as a source of sound. The sound generated in this way is called turbulence noise. Air turbulence produced in this way, by various kinds of constrictions in the vocal tract is the typical sound source for all fricatives [16]-[19] the position of which depends on the particular fricative. Assamese is a major language in the north eastern part of India spoken by over 30 million people. It has a rich linguistic diversity with vast dialectal and ethnographic tonal variations. In Assamese language, fricative forms a major group of speech sounds which has different phonemical characteristics. In Assamese language, voiceless alveolar fricative /s/ and velar fricative /x/ are observed. Further, voiced alveolar fricative /z/ and voiced glottal fricative /ɦ/ are also identified. Unlike other Indian languages, the presence of voiceless velar fricative /x/ is a specific feature of the language [20]-[22].

For classification of fricatives, we have performed experiments on two different feature sets. The first set of feature vectors are generated from the specific acoustic-phonetic characteristics i.e. centre of gravity (COG), standard deviation (SD), skewness and kurtosis and the second set of feature vector have been formulated using frame based Linear prediction cepstral coefficients (LPCC). The second feature used for the work is of temporal nature and here a Self Organizing Map (SOM) is used to reduce the dimension of features.

The rest of the paper is organized as follows. Section 2 provides a details of theoretical considerations, Section 3 describes the feature set used in the work and Section 4 provides the experimental details and result derived out of the work. Section 5 concludes the description.

2 Theoretical Consideration

Here, we briefly describe the related theoretical aspects. We focus on certain basic attributes of the RNN, ANFIS, SOM, LPCC and fricative sounds of Assamese language.

2.1 RNN

RNNs are types of ANNs with one or more feedback loops. The feedback can be of a local or global kind. The RNN may be considered to be an MLP having a local or global feedback in a variety of forms. It may have feedback paths from the output neurons of the MLP to the input layer or a global feedback from the hidden neurons of the ANN to the input layer [23]. RNN uses learning algorithm like Back Propagation Through Time (BPTT), Real-Time Recurrent Learning (RTRL), Decoupled Extended Kalman Filtering (DEKF) etc. The architectural layout of a recurrent networks takes many different forms like Input-Output Recurrent Model, State-Space Model etc which are commonly used. Each of these have a specific form of global feedback. RNN's design enables it to deal with dynamic variations in the input. It has feedforward and feedback back paths which contribute towards the temporal processing ability. The feedforward paths make it like the MLP hence is able to make non-linear discrimination between boundaries using a gradient descent based learning. Next, the feedback paths enable the RNN to generate contextual processing. The feedback paths are sometimes passed through memory blocks which enable delayed versions of the processed output to be considered for processing. These variations can be due to the time dependent behavior of the input patterns. So while the MLP is only able to do discriminations between applied samples, the RNN is able to distinguish classes that show time variations. For the above mentioned attributes, RNNs are found to be suitable for application like speech recognition.

2.2 ANFIS

ANFIS are a class of adaptive networks which combines the powerful description of fuzzy classification techniques with the learning capabilities of ANNs. Fundamentally, ANFIS is a FIS whose membership function parameters are adjusted using either a back propagation algorithm alone or in combination with a RLS type adaptive update running in recursion. This adjustment allows the fuzzy systems to learn from the data they are modeling. The primary aspect is that if the inputs have finer variations and the FIS has the appropriate rule set, the system is able to follow the changes appropriately. Here, applied patterns are mapped through input and then through output membership functions and associated parameters to the decision layer. An adaptive network is a multi-layer feedforward network in which each node performs a particular function based on incoming signals and a set of parameters pertaining to this node. The type of node function may vary from node to node and the choice of node function depends on the overall function that the network is designed to carry out. ANFIS implements Takagi Sugeno FIS and has a five layered architecture as shown in Figure 2.2. The first hidden layer is for fuzzification of the input variables and T-norm operators are deployed in the second hidden layer to compute the rule antecedent part. The third hidden layer normalizes the rule strengths followed by the fourth hidden layer where the consequent parameters of the rule are determined. Output layer computes the overall input as the summation of all incoming signals [24] [25].

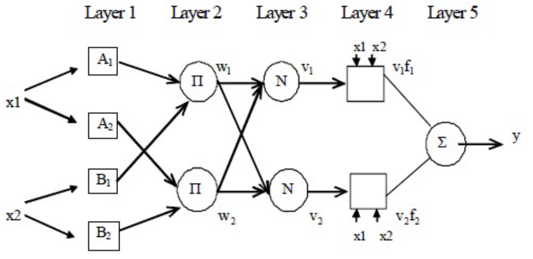


Figure 1. Architecture of ANFIS

2.3 SOM

The SOM is a method of data analysis used for clustering and projecting multi-dimensional data into a lower-dimensional space to reveal hidden structures of the data. The SOM [26] is a class of ANN capable of recognizing the main features of the data they are trained on. Kohonen proposed SOM architecture which can automatically generate self organization properties during unsupervised learning process. Kohonen SOM is unsupervised system which is based on the competitive learning. It means that a competition process takes place before each cycle of learning. In the competition process a winning processing element is chosen by some criteria. Usually this criteria is to minimize an Euclidean distance between the input vector and the weight vector. After the winning processing element is chosen, its weight vector is adapted according to the learning law used [23]. The basic network architecture of Kohonen's SOM is shown in Figure 2.

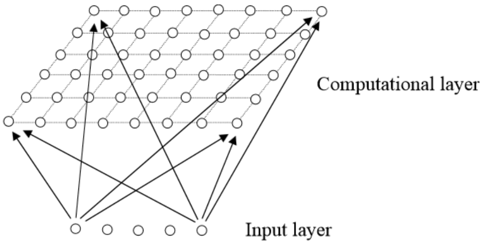


Figure 2. Generic Self Organizing Map

The learning procedure of Kohonen feature maps is similar to that of competitive learning networks. A similarity (dissimilarity) measure is selected and the winning unit is considered to be the one with the largest (smallest) activation. For Kohonen feature maps, the winning unit's weights and also all of the weights in a neighborhood around the winning units are updated [27].

2.4 LPCC feature extraction method

In the linear prediction analysis of speech, each sample is predicted as a linear weighted sum of the past p samples where p represents the order of prediction [28]. If $s(n)$ is the present sample, then it is predicted by the past p samples as

$$\hat{s}(n) = -\sum_{k=1}^{p} a_k s(n-k) \tag{1}$$

The difference between the actual and predicted sample value is termed as the prediction error or residual, which is given by

$$e(n) = s(n) - \hat{s}(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k) \tag{2}$$

where $\{a_k\}$ are the linear prediction coefficients. The linear prediction coefficients are typically determined by minimizing the mean square error (MSE) over an analysis frame. The coefficients can be obtained by solving the set of p normal equations,

$$\sum_{k=1}^{p} a_k R(n-k) = -R(n), n = 1, 2, \ldots, p \tag{3}$$

where

$$R(k) = \sum_{n=0}^{N-(p-1)} s(n)s(n-k), k = 0, 1, 2, \ldots, p \tag{4}$$

and $\{s(n)\}$ are the speech samples and N is the numbers of samples in one analysis frame.
In the frequency domain, the eq. (2) can be represented as,

$$E(z) = S(z) + \sum_{k=1}^{p} a_k S(z) z^{-k} \tag{5}$$

i.e.

$$A(z) = \frac{E(z)}{S(z)} = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{6}$$

A cepstrum is the result of taking the Inverse Fourier transform (IFT) of the logarithm of the estimated spectrum of a signal. The concept of cepstrum was defined in a 1963 paper by Bogert et al [29]. A short-time cepstrum analysis was proposed by Schroeder and Noll for application to pitch determination of human speech [30] [31]. Cepstral parameter extraction in speech recognizers system is based on converting LPC parameters to cepstral coefficients by utilizing the recursion relationship.
Cepstral coefficients of $A(z)$ is given by,

$$c(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log A(e^{j\omega}) e^{jn\omega} d\omega \tag{7}$$

The cepstrum parameters may be computed directly from the LP parameters using the following recursion.

$$c(k) = a(k) - \sum_{m=1}^{k-1} \frac{m}{k} c(k) a(k-m), 1 \leq k \leq p \quad (8)$$

In speech recognition systems, the cepstrum also plays a significant role. Specifically, the cepstral coefficients have been found empirically to be a more robust, reliable feature set for speech recognition than linear predictive coding (LPC) coefficients or other equivalent parameter sets [32]. Thus, the cepstral coefficient of the LPC obtained are applied to SOM for clustering which will form the feature vector for the RNN classifier in this paper.

2.5 Fricative Sounds of Assamese language

Assamese is a major language spoken in the North-Eastern part of India. It is the official language of state of Assam. It is an Indo-Aryan language originated from Vedic dialects but the exact nature of the origin and growth of the language is not very clear as yet [20]. It is supposed that like other Aryan languages, Assamese was also born from *Apabhraṁśa* dialects developed from *Mãgadhi* Prakrit of the eastern group of Sanskritic language [20]. Assamese phonemic inventory consists of eight vowels and twenty-one consonants. The consonants may be grouped into broad divisions: the stops and the continuants. There are eleven continuants out of which four spirants or fricatives /s/, /z/, /x/, /ɦ/ are identified [20] [33]. These four Assamese fricative as shown in Table 2.5 are described below [20]-

1 Voiceless alveolar sibilant, /s/: It is one of the most common sound cross linguistically. Its manner of articulation is sibilant fricative, which means it is generally produced by channeling air flow along a groove in the back of the tongue up to the place of articulation, at which point it is focused against the sharp edges of the nearly clenches teeth, causing high frequency turbulence. Its place of articulation is alveolar means it is articulated with tongue at the alveolar ridge. Its phonation is voiceless, which means it is produced without vibrations of the vocal cords.

2 Voiced alveolar fricative, /z/: Its manner of articulation is also sibilant. But its phonation is voiced, which means the vocal cords vibrate during the articulation.

3 Voiceless velar fricative, /x/: Its place of articulation is velar, which means it is articulated with the back of the tongue at the soft palate. Its phonation is voiceless. Assamese is unusual among eastern Indo-Aryan language for the presence of voiceless velar fricatives. It is similar to the velar sound in German of Europe. Phonetically, this /x/ sound is pronounced somewhat in between the sounds /s/, /kh/ and /h/ and is similar to the German sound /ch/ as pronounced in the word 'Bach' or the Scottish sound as found in the word 'Loch'. It may be an Indo- European feature, which has been preserved by ɑxɑmija [34] [35]. It is an important phoneme in the language.

4 Voiced glottal fricative, $/H/$: Its phonation is voiced, which means the vocal cords vibrate during the articulation.

Table 1. Assamese Fricative Phonemes

Phonation	Place of Articulation		
	Alveolar	Velar	Glottal
Voiceless	/s/	/x/	
Voiced	/z/		/ɦ/

3 Feature set used in the work

This section provides a brief description of the two different feature vectors used in this work. The first feature vector is characterized by acoustic phonetic characteristics while the other is formed by LPCC constituents.

3.1 Feature vector creation using acoustic phonetic characteristics

Fricatives can be clearly differentiated from one another using spectral features of them. Spectral features include determination of first, four spectral moments viz. center of gravity (COG), Standard deviation (SD), skewness and kurtosis. Overall noise amplitude is also investigated. A feature vector is formed using an acoustical study carried out on the spectral characteristics of the four fricatives.

A set of experiments are performed on the speech database mentioned in Sect. 4.1, to measure the acoustic features, mainly spectral moments, namely COG, SD, skewness and kurtosis etc. and thus a feature vector is created for each of the fricative examples used for pattern mapping to class codes using RNN. It is well known that the COG reflects average energy concentration and SD is the measure of how much the frequency in a spectrum can deviate from the COG. On the other hand, skewness refers to spectral tilt, the overall slant of the energy distribution. Positive skewness suggests a negative tilt with a concentration of energy in the lower frequencies and negative skewness is associated with a positive tilt and a predominance of energy in the higher frequencies. Finally, kurtosis is an indicator of the peakedness of the distribution. Positive kurtosis values indicate a relatively high peakedness (the higher the value, the more peaked the distribution), while negative values indicate a relatively flat distribution. Positive kurtosis thus suggests a clearly defined spectrum with well-resolved peaks, while negative kurtosis indicates a flat spectrum without clearly defined peaks. Amplitude of fricative noise is also useful for classification and recognition. The fricative samples used for these experiments are later used for RNN training. All these parameters are measured by writing a few simple scripts in the speech analysis software PRAAT. The feature vectors thus generated are of size 1×4, which has four parameters COG, SD, skewness and kurtosis on its four elements. Such a feature vector for fricative /s/ generated using the four parameter for a male and a female speaker is shown in the Table 2 and Table 3.1.

Table 2. Feature vector of fricative /s/ (male speaker)

COG (Hz)	SD (Hz)	Skewness	Kurtosis
8747	2034	-2.26	8.72

Table 3. Feature vector of fricative /s/ (female speaker)

COG (Hz)	SD (Hz)	Skewness	Kurtosis
9541	1166	-1.24	15.81

3.2 Feature vector creation using LPCC and SOM

Initially, LPCC is computed from the fricative sound for 20 mili second (ms) frame with a shift of 10 ms and after that a difference LPCC coefficient is computed. The difference coefficients for frame n are the difference between the coefficients of frame $n+\delta$ and $n-\delta$. In our implementation, a differential coefficient is computed every frame, with $\delta = 1$ frames. The feature vectors obtained for a particular fricative sound is presented to SOM for clustering. If a 13^{th} order LPCC is performed for every frame, the size of the feature matrix will be $N \times 12$, where N will be the number of frames present in the speech segment. SOM is used here to cluster the feature vectors extracted in that way since adjacent frames may possess less variation. SOMs role is to simply bring similar frames into one cluster. Thus vectors obtained after taking LPCC are fed to a SOM network with different cluster size, $M(M < N)$ for grouping the similar data. The cluster size, M used here are 8 and 10. The cluster provided by SOM is used as pattern vector for classification.

4 Experimental Details, Results and Discussion

In this section, we report the experimental details and the results of fricative classification obtained from a RNN and an ANFIS based classifiers. Individual results for the two classifiers using two different feature set are reported. The following sections provide description of speech database and experiments performed.

4.1 Speech Database

The speech database is created from speakers of four different dialects of Assamese language. A word list as shown in Figure 3 is prepared containing fricative-vowel-fricative (C_iVC_i and C_iVC_j) and vowel-fricative-vowel (V_iCV_i and V_iCV_j) syllables and are recorded by the trained speakers in a noise free environment. Each CVC and VCV token is repeated five times, yielding a total of 245 tokens per speaker (49 syllables × 5 repetitions). For recording, the speech analysis software Wavesurfer [36] and a PC headset is used with the following specification-

- Sampling frequency: 48000 Hz and

- Bit resolution: 16 bit per sample

Assamese has four different dialects namely Eastern, Central, Kamrupi and Goalpariya groups. From every dialect there are 3 speakers. After recording, fricatives are annotated and segmented in the speech analysis software PRAAT [37].

4.2 Classification using RNN with acoustic phonetic feature

A significant portion of the work is related to the design of the RNN classifier, its training, validation and testing for recognition of Assamese fricatives using acoustic phonetic features. The formulation of the features have already been discussed. The process logic of the design of classifier using RNN and acoustic phonetic feature is shown in Figure 4.

Acoustic-phonetic features of four fricatives are used to perform the RNN based fricative classification problem in order to recognize the fricatives. Initially, the gradient descent with adaptive learning rate backpropagation algorithm is used to train the RNN with 3 hidden layer and 15 feature vectors per fricative, but it requires more time and the recognition rate is somewhat lower. Then the Resilient Backpropagation (RB) and Levenberg-Marquardt (LM) training algorithms are adopted for training, which have provided 88% success rate, if the feature vector size is increased to 25 per fricative. Further, Scaled Conjugate Gradient (SCG) and Bayesian Regularization (BR) algorithms are also used, which further increases the success rate to 96% and is found to be the best among all the four algorithms in terms of success rate. Table 4 represents the success rate of Assamese fricatives using SCG training algorithm. The comparison between recognition success rate and training algorithms is shown in Table 5. It can be observed from the Table 5 that LM and RB algorithm give same recognition rate, but training speed is better for RB algorithm. SCG and BR algorithm give 96% success rate. But BR algorithm takes relatively more training time than that of SCG algorithm. Therefore, from success rate and computational time point of view, SCG algorithm is a better choice for the proposed work.
The experiments are repeated for the entire data set considered and at least ten trials are conducted for each of the algorithms considered during training. It makes the set up robust. The feature set considered and the RNN formulated turns out to be a robust combination for fricative recognition in Assamese.

Table 4. Classification result of RNN using scaled conjugate gradient training algorithm

Fricative	Correct Recognition	Faulty Recognition
/s/	98%	2%
/z/	88%	12%
/x/	98%	2%
/ɦ/	98%	2%

Table 5. Comparison of different training algorithms

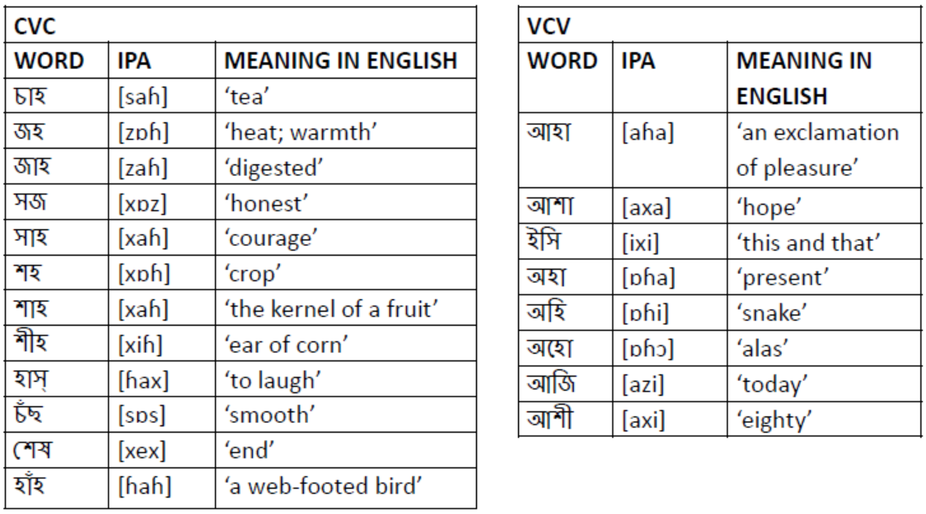
Training Algorithm	Recognition Rate (%)	Training Time (sec)
Levenberg-Marquardt	88	31.20
Resilient Backpropagation	88	7.46
Scaled Conjugate Gradient	96	174.75
Bayesian Regularization	96	250.90

4.3 Classification using RNN with LPCC and SOM based feature

The results obtained from the RNN-acoustic phonetic feature combination is next compared with that obtained using LPCC and SOM generated features applied to a RNN classifier. Figure 5 shows the classifier using RNN and LPCC and SOM based feature.

The steps involved on the process can be summarized below-

1 Recording of 245 number of fricative-vowel-fricative (C$_i$VC$_i$ and C$_i$VC$_j$) and vowel-fricative-vowel (V$_i$CV$_i$ and V$_i$CV$_j$) syllables



CVC		
WORD	**IPA**	**MEANING IN ENGLISH**
চাহ	[saɦ]	‘tea’
জহ	[zɒɦ]	‘heat; warmth’
জাহ	[zaɦ]	‘digested’
সজ	[xɒz]	‘honest’
সাহ	[xaɦ]	‘courage’
শহ	[xɒɦ]	‘crop’
শাহ	[xaɦ]	‘the kernel of a fruit’
শীহ	[xiɦ]	‘ear of corn’
হাস্	[ɦax]	‘to laugh’
চঁছ	[sɒs]	‘smooth’
শেষ	[xex]	‘end’
হাঁহ	[ɦaɦ]	‘a web-footed bird’

VCV		
WORD	**IPA**	**MEANING IN ENGLISH**
আহা	[aɦa]	‘an exclamation of pleasure’
আশা	[axa]	‘hope’
ইসি	[ixi]	‘this and that’
অহা	[ɒɦa]	‘present’
অহি	[ɒɦi]	‘snake’
অহো	[ɒɦɔ]	‘alas’
আজি	[azi]	‘today’
আশী	[axi]	‘eighty’

Figure 3. Wordlist Prepared for Recognition purpose

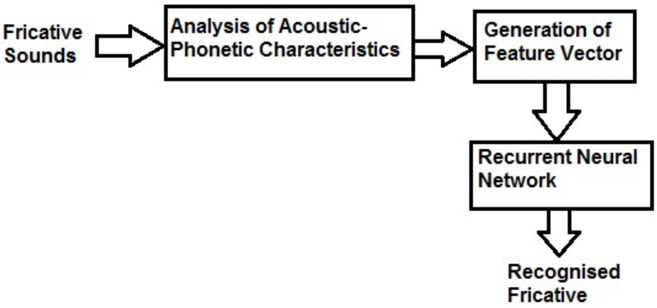

Fricative Sounds
Analysis of Acoustic-Phonetic Characteristics
Generation of Feature Vector
Recurrent Neural Network
Recognised Fricative

Figure 4. Process logic of the RNN and acoustic phonetic feature

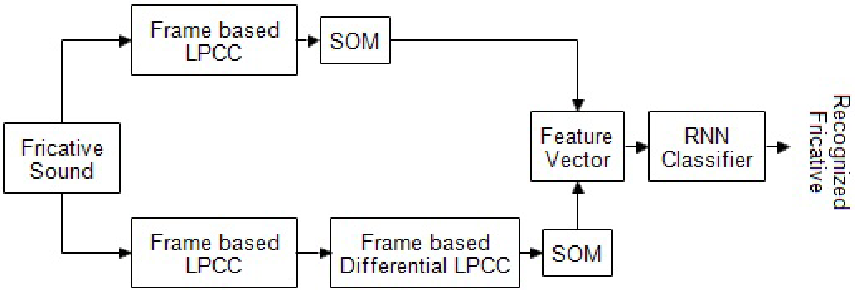


Figure 5. Process logic of the RNN classifier using LPCC and SOM based feature

from 12 speakers covering all the four dialects of Assamese language.

2 Extraction of features from recorded fricative sounds using frame-based LPCC and frame-based differential LPCC method.

3 Generation of hybrid feature vectors for both LPCC methods those will be used for training and testing of said classifiers.

4 Training each of the RNN with 80 fricative samples using those feature vectors.

5 Testing of the algorithm with 30 samples per fricative.

The feature vectors generated from the frame-based LPCC of the fricative speech are presented to the SOM for clustering the large dimensional data. Here, SOM is used to reduce the size of the feature vector which will form the pattern layer of the RNN classifier. A 12^{th} order linear prediction analysis is performed for every frame of 20 ms speech with overlap of 10 ms. So, the size of the vector after taking LPCC will be $N \times 12$, where N is the number of frames present in the speech segment. Here, a SOM block is used to cluster the feature vectors extracted in groups with commonality. This is required because of the fact that adjacent frames possess less variation hence have higher correlation which decelerates the learning of the classifier. SOM's role is to group similar frames into one cluster. To attain this objective, vectors obtained after taking LPCC are fed to a SOM network with different cluster sizes, M $(M < N)$ for grouping the similar data. The cluster size, M used here are 8 and 10. The clusters provided by SOM are used as pattern vectors for driving the training and testing of the RNN classifier. This way original feature vectors obtained from 20 ms frames are reduced into the differentially sized cluster centers.

Initially, a few training algorithms like gradient descent with adaptive learning rate backpropagation, Resilient Backpropagation (RBP) and Levenberg-Marquardt (LM) optimized backpropagation are used to train the RNN with 3 hidden layers and 40 feature vectors. But recognition rate observed is somewhat lower and requires more time to complete the learning cycle. Finally, Scaled Conjugate Gradient (SCG) algorithms is used with 80 feature vectors, which increases the success rate to an acceptable mark and is found to be the best among all the four algorithms in terms of recognition rate. Table 4.3 shows the percentage of correct recognition using frame-based LPCC with cluster size, M of value 8. It is observed that the overall recognition rate is 77% considering LPCC and RNN with three hidden layer and 80 feature vector combination trained with SCG learning algorithm where $M = 8$. The same RNN is now trained with $M = 10$. Table 4.3 shows the results which indicates an improvement of overall recognition rate for fricative inputs reaching levels upto 80%. The primary reason behind selection of different values of M (namely 8 and 10) is to ascertain the effect of cluster size on recognition. With $M = 10$, more information is extracted than is the case with $M = 8$, hence better success rate is obtained with the former.

Table 6. Classification result of RNN using frame-based LPCC (M=8)

Fricative	Correct Recognition	Faulty Recognition
/s/	86.67 %	13.33%
/z/	70 %	30%
/x/	70 %	30%
/ɦ/	80 %	20%

Table 7. Classification result of RNN using frame-based LPCC (M=10)

Fricative	Correct Recognition	Faulty Recognition
/s/	83.33%	16.67%
/z/	76.67 %	23.33%
/x/	70%	30%
/ɦ/	86.67%	13.33%

To increase the recognition rate further, we design a hybrid feature set to train the RNN classifier. For that purpose, difference of LPCC from one frame to another is used to create novel feature vector. This provides another set of feature vectors which is clustered using SOM. A 2^{nd} RNN classifier is trained using differential LPCC, which add knowledge to the main classifier. This way recognition rate improves. Frame-based differential LPCC is combined with a delayed version of the conventional frame-based LPCC which forms a hybrid feature vector set for the RNN to train efficiently. Table 4.3 shows the results of recognition rates obtained using the hybrid feature set with the RNN classifier. It has three hidden layers and $M = 10$ dealing with hybrid LPCC features. With these parameters, the overall recognition rate of Assamese fricatives is found to be approximately 82%.

Table 8. Classification result of RNN using frame-based LPCC and differential LPCC (M=10)

Fricative	Correct Recognition	Faulty Recognition
/s/	86.67 %	13.33%
/z/	76.67%	23.33 %
/x/	76.67 %	23.33%
/ɦ/	86.67%	13.33%

4.4 Classification using ANFIS with acoustic phonetic feature and LPCC and SOM based feature

The results of the RNN classifier derived using the multiple feature sets as described above is next compared with that obtained using the ANFIS classifier. The ANFIS block replaces the RNN classifier with the rest of the process logic remaining unchanged. ANFIS uses a fuzzified backpropagation learning to capture the variations in the inputs which are modified appropriately. The parameters related to membership functions and RLS estimation are fixed to enable proper real to fuzzy world mapping, learning, inference generation, decision making and back translation from the fuzzy to the real world. The learning procedure of the ANFIS has two parts. In the first part, the input patterns are propagated and the optimal consequent parameters are estimated by an iterative RLS procedure, while the premise parameters are assumed to be fixed for the current cycle through the training set. In the second part the patterns are propagated again and a fuzzified backpropagation is used to modify the premise parameters, while the consequent parameters remain fixed. This procedure is then iterated [24] [25]. We have used trapezoidal membership function to generate the fuzzification for use in the FIS. The classification results of ANFIS for acoustic phonetic feature is represented in Table 9 and for LPCC and SOM based feature is represented in Table 4.4. Table 4.4 shows a computational time required by the ANFIS and RNN based classifier to do classification of four fricatives with same number of input patterns for two types of feature. In comparison to RNN, the correct recognition rate of ANFIS is little less in the current database of fricative sounds for both the feature sets. However, ANFIS provides advantage in terms of computational time. The learning of the ANFIS is fast and computationally less demanding. The computational time shown in Table 4.4 is for a worst case situation where due to less optimized feature set, the training time is extended than the case with the inputs samples having low corellation content.

Table 9. Classification performance using ANFIS for acoustic phonetic feature

Fricative	Correct Recognition	Faulty Recognition
/s/	85.4%	14.6%
/z/	86.2%	13.8%
/x/	83.4%	16.6%
/ ɦ /	89.5%	10.5%

Table 10. Classification using ANFIS for frame-based LPCC and differential LPCC (M=10)

Fricative	Correct Recognition	Faulty Recognition
/s/	73.8 %	26.2%
/z/	72.57 %	27.43%
/x/	70.51 %	29.49%
/ɦ/	78.9%	21.1 %

Table 11. Computational load (worst case) for RNN and ANFIS

Feature	RNN	ANFIS
Acoustic Phonetic	162.28 sec	53.8 sec
LPCC and SOM	223.14 sec	83.23 sec

With acoustic features, the RNN and the SCG algorithm generates a highest correct recognition rate of around 98% (Table 4). SCG algorithm provides best performance of around 96% on an average for the entire data set and a worst case computational time of 174.75 seconds (Table 5). With LPCC features, RNN gives a highest performance of 86.67% with 8 clusters (Table 6) and a marginal improvement with 10 cluster numbers (Table 7).

Using frame based LPCC and differential LPCC, the 10 cluster configuration improves RNN performance (Table 8). With ANFIS using acoustic phonetic features, the performance is marginally lower. This is also observed with the frame based LPCC and differential LPCC when used with the ANFIS classifier. But the advantage is with respect to the computational time which is between 2.7 to 3 times lower than that required by the RNN. This aspect of the ANFIS can be used further to enhance the overall efficiency of the system.

5 Conclusion

Here, we have presented a comparative depiction of the experimental results derived from fricative classification of Assamese speech using RNN and ANFIS. Two types of feature sets are derived for the work, one using acoustic phonetic characteristics of fricative sounds and the other using LPCC and SOM based temporal feature. It is observed that the RNN provides better classification for both the feature sets but at the cost of higher computational speed. Although ANFIS performance is somewhat lower for the current set of data, computational time is significantly lesser than RNN. The work establishes the applicability of ANFIS based recognizer for fricative classification of a dialectically oriented and ethnographic diction containing language like Assamese, the vital aspects of which are captured using acoustic phonetic and temporal features.

References

[1] M. Sarma and K.K. Sarma, Phoneme-Based Speech Segmentation Using Hybrid Soft Computing Framework. Studies in Computational Intelligence vol. 550, Springer India, New Delhi, 2014.

[2] M. Sarma and K. K. Sarma, An ANN based approach to Recognize Initial Phonemes of Spoken Words of Assamese Language. Elsevier International Journal of Applied Soft Computing, vol. 13, no. 5, pp. 2281-2291, 2013.

[3] T Robinson, M Hochberg and S Renals, IPA: Improved phone modelling with recurrent neural networks. In Proceedings of IEEE ICASSP, 1994.

[4] T Lee, P C Ching and L W Chan, An RNN Based Speech Recognition System with Discriminative Training. In Proceedings of the 4th European Conferrence on Speech Communication and Technology, pp. 1667-1670, 1995.

[5] L. H. R. C. Jamieson, Experiments on the Implementation of Recurrent Neural Networks for Speech Phone Recognition. Proceedings of the Thirtieth Annual Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, pp. 779-782, November, 1996.

[6] T. Koizumi, M. Mori, S. Taniguchi and M. Maruya, Recurrent Neural Networks for Phoneme Recognition. In Proceedings Fourth International Conference, vol. 1, pp. 326 -329, 1996.

[7] L J M Rothkrantz and D Nollen, Speech Recognition Using Elman Neural Networks. Text, Speech and Dialogue, Lecture Notes in Computer Science, 1692: 146-151, 1999.

[8] Y Sun , L T Bosch and L Boves, Hybrid HMM/BLstm-Rnn for Robust Speech Recognition. In Proceedings of 13th International Conference on Text, Speech and Dialogue, Springer-Verlag Berlin, Heidelberg :400-407, 2010.

[9] O Vinyals , S V Ravuri and D Povey, Revisiting Recurrent Neural Networks For Robust ASR. In Proceedings of IEEE International Confrence on Acoustics, Speech, and Signal Processing (ICASSP), 2012.

[10] T Mikolov and G Zweig,Context Dependent Recurrent Neural Network Language Model. In Proceedings of the IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA : 234-239, 2012.

[11] S. Badura, M. Fnitrik, O. Skvarek, M. Klimo, Bimodal vowel recognition using fuzzy logic networks - naive approach. In Proceedings of ELEKTRO, Rajecke Teplice, 2014.

[12] R. Halavati, S. B. Shouraki, S. H. Zadeh, Recognition of human speech phonemes using a novel fuzzy approach. Applied Soft Computing, 7:828839, 2007 .

[13] I. B. Fredj, K. Ouni, A novel phonemes classification method using fuzzy logic. Science Journal of Circuits, Systems and Signal Processing vol. 2, no. 1, pp. 1-5, 2013.

[14] P. Melin, J. Urias, D. Solano, M. Soto, M. Lopez and O. Castillo, Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms, Engineering Letters, vol.3, no.2, 2006.

[15] A. Taleb, Speech Recognition by Fuzzy-Neuro ANFIS Network and Genetic Algorithms, In Proceedings of International Conference on Intelligent Computational Systems, Dubai, 2012.

[16] A. Jongman, R. Wayland, and S. Wong, Acoustic characteristics of English fricatives. Journal of Acoustical Society of America , vol. 108, no. 3, Sepember, 2000.

[17] D. O'Shaughnessy, Speech Communication Human and Machine, 2^{nd} Edition, IEEE Press, New York, 2000.

[18] Kenneth N. Stevens, Acoustic Phonetics, 1^{st} MIT Press paperback Edition, The MIT Press, Cambridge, Massachusetts, London, England, 2000.

[19] P. Ladefoged, S. F. Disner, Vowels and Consonants, 3^{rd} Edition, Wiley-Blackwell Publishing Ltd., West Sussex, UK, 2012.

[20] G. C. Goswami, Structure of Assamese, 1^{st} Edition, Department of Publication, Gauhati University, Guwahati, Assam, India, 1982.

[21] U. N. Goswami, An Introduction to Assamese, Mani-Manik Prakash, Guwahati, Assam, India, 1978.

[22] G. C. Goswami and J. P. Tamuli, "Asamiya", in G. Cardona and D. Jain (eds.), The Indo-Aryan Languages, London: Routledge, pp. 391-443, 2003.

[23] S. Haykin, Neural Networks:A Comprehensive Foundation, 2^{nd} Edition, Prentice-Hall of India Pvt. Ltd., Delhi, India, 2005.

[24] J. R. Jang, ANFIS : Adaptive Network-Based Fuzzy Inference System, IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 3, 1993.

[25] A. Abraham, Neuro Fuzzy Systems: State-of-the-art Modeling Techniques, Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence Lecture Notes in Computer Science, Vol.2084, pp 269-276, 2001.

[26] K. Haese, Self-organizing feature maps with self-adjusting learning parameters. IEEE Transactions on Neural Networks, vol. 9, pp. 1270-1278, 1998.

[27] J. S. R. Jang, C. T. Sun and E. Mizutani, Neuro-Fuzzy and Soft-Computing, 1^{st} Edition, Prentice-Hall of India Pvt. Ltd., Delhi, India, 2011.

[28] J. Makhoul, Linear prediction: A tutorial review, In Proceedings of IEEE, vol. 63, pp. 561-580, 1975.

[29] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking, Proceedings of the Symposium on Time Series Analysis, Chapter 15, pp. 209-243, 1963.

[30] A. M. Noll and M. R. Schroeder, Short-Time 'Cepstrum' Pitch Detection, Journal of the Acoustical Society of America, vol. 36, no. 5, pp. 1030-1036, 1964.

[31] A. M. Noll , Short-Time Spectrum and Cepstrum Techniques for Vocal-Pitch Detection, Journal of the Acoustical Society of America, vol. 36, no. 2, pp. 296-302, 1964.

[32] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Englewood Cliffs, NJ:Prentice-Hall, 1975.

[33] Sarma B D, Sarma M, Sarma M and Prasanna S R M, Development of Assamese Phonetic Engine: Some Issues, In Proceedings of INDICON-2013, IIT Bombay, Mumbai, India, 2013.

[34] Rajen Barua, The X sound in Assamese language, The Assam Tribune, Guwahati, Sunday, March 5, 2006.

[35] Prof. Gautam Baruah, Dept. of CSE, IIT Guwahati, Available via
*tdil.mit.gov.in/assamesecodechartoct*02.*pdf*

[36] *http : //www.speech.kth.se/wavesurfer/*

[37] P. Boersma and D. Weenink, Praat: doing phonetics by computer. Available via
http : //www.fon.hum.uva.nl/praat/