Spoken Language Identification using Neural Networks
BTP Mid Term Presentation

Prepared under the guidance of
Dr. Shampa Chakraverty

by:

Aditya Jain 210/CO/13

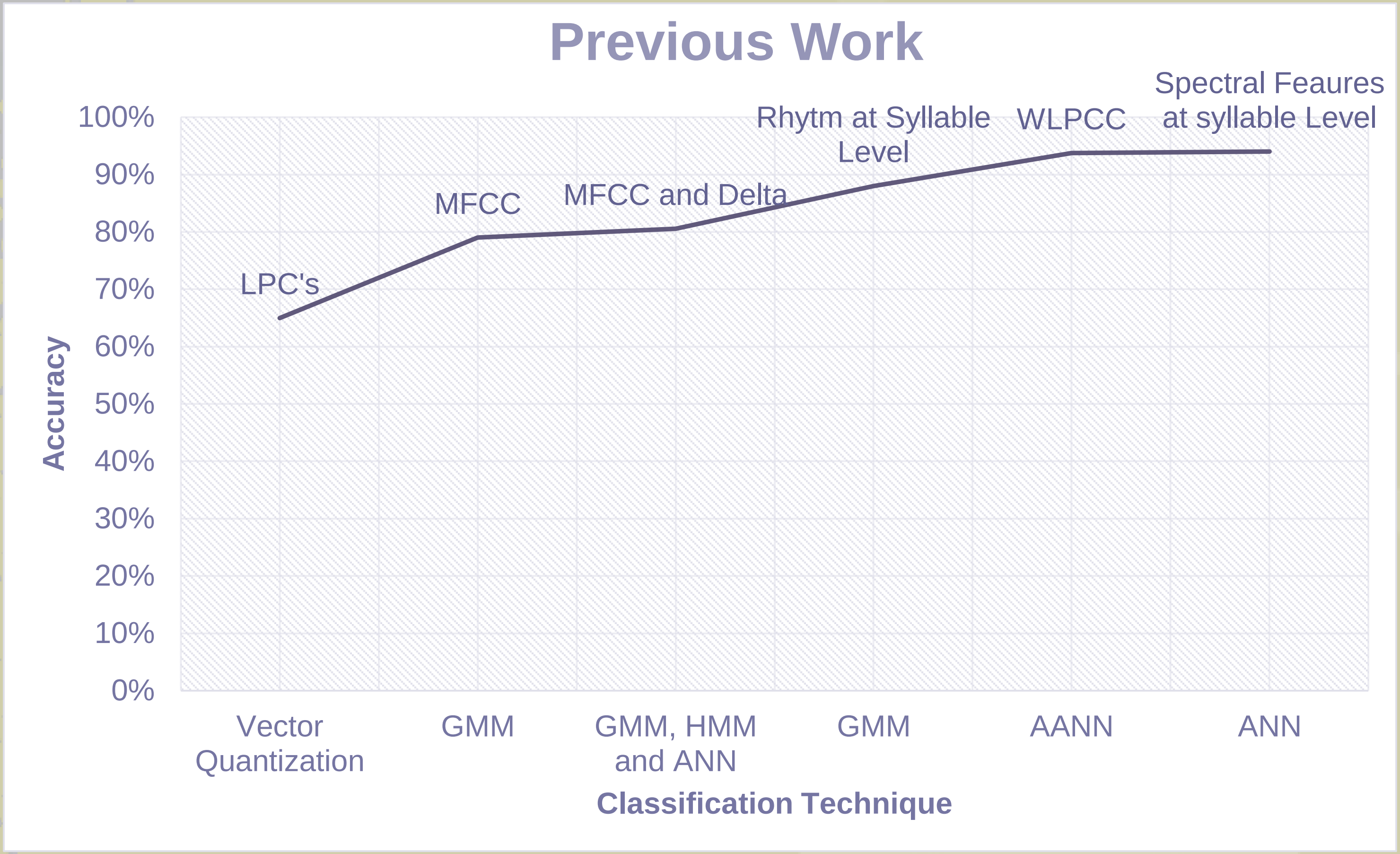
Anmol Pandey 233/CO/13

Anmol Varshney 234/CO/13

Introduction

- Objective
  - The problem we will tackle is Spoken Language Identification using Neural Networks
  - We intend to build a model which achieves the best accuracy by using a general Neural Network and a set of other Neural Networks aimed at Binary Classification.

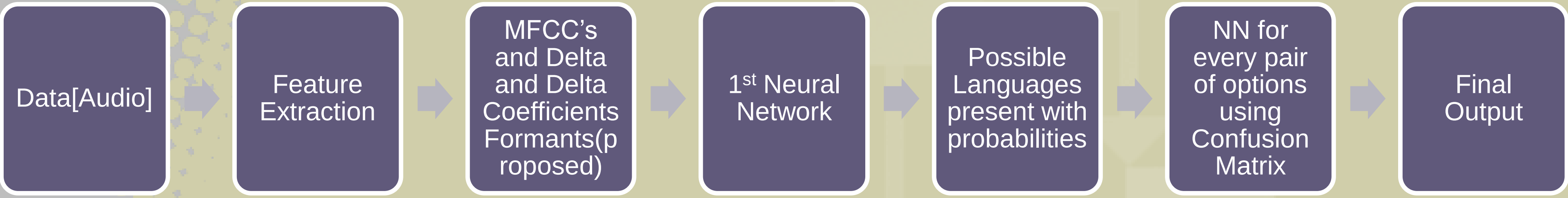
History of Spoken Language Identification

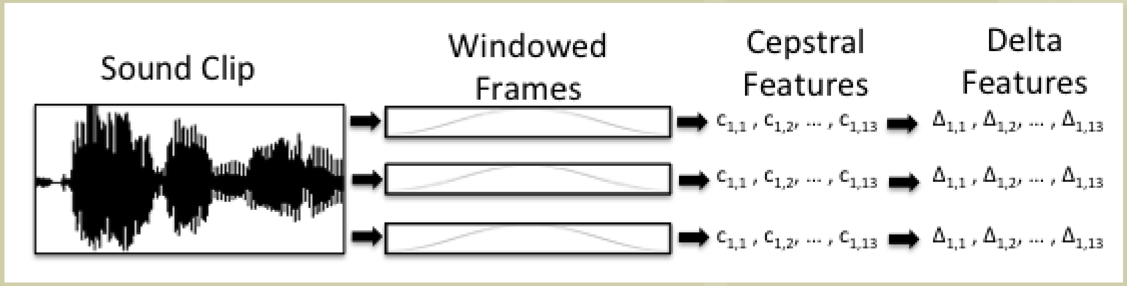


Our Proposal

- We aim to improve the accuracy further by using 2 layers of Neural Networks
- The 1st NN aims at general discrimination of languages at the frame level
- It generates probabilities of a frame being of a certain language and these are then averaged over for all the frames of an audio clip
- The 2nd NN applies nC2 classifiers aimed at binary classification, where n is the number of options that the 1st NN suggests
- The 2nd NN uses features selected by using LDA
- The selected features for each pair of languages is stored in a confusion matrix

Spoken Language Identification Neural Network Model

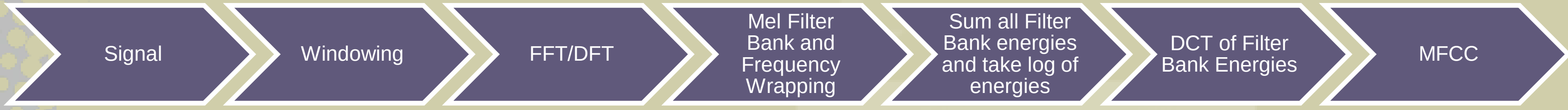

Data[Audio]
Feature Extraction
MFCC’s and Delta and Delta Coefficients Formants(proposed)
1st Neural Network
Possible Languages present with probabilities
NN for every pair of options using Confusion Matrix
Final Output


Sound Clip
Windowed Frames
Cepstral Features
Delta Features
$c_{1,1}, c_{1,2}, \dots, c_{1,13}$
$\Delta_{1,1}, \Delta_{1,2}, \dots, \Delta_{1,13}$

The feature extraction process

MFCC Features

- The mel-frequency cepstrum is a representation of an audio signal on the mel scale, a nonlinear mapping of frequencies that down-samples higher frequencies to imitate the human ear’s ability to process sound.
- In our implementations, we used the first 13 cepstral coefficients as our primary features, as is common in similar applications


Signal
Windowing
FFT/DFT
Mel Filter Bank and Frequency Wrapping
Sum all Filter Bank energies and take log of energies
DCT of Filter Bank Energies
MFCC

Delta Features

- We also investigated using delta features. Delta features are the first and second time derivatives of the cepstral coefficients, capturing the change of the cepstral features over time, which we hypothesize will be useful in classifying language, since pace is an important factor in language recognition by humans.
- We can calculate these features as the central finite difference approximation of these derivatives

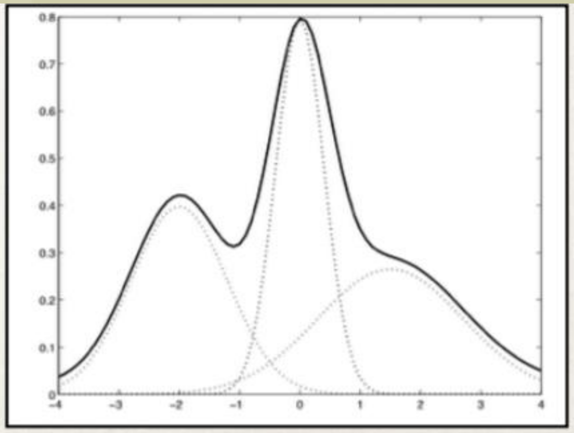
$$\Delta_{t,i} = \frac{c_{t+1,i} - c_{t-1,i}}{2} \qquad \text{and} \qquad \Delta^2_{t,i} = \frac{c_{t+1,i} - 2c_{t,i} + c_{t-1,i}}{4},$$

where $c_{t,i}$ denotes the ith coefficient of the tth frame of the clip.

Gaussian Mixture Model

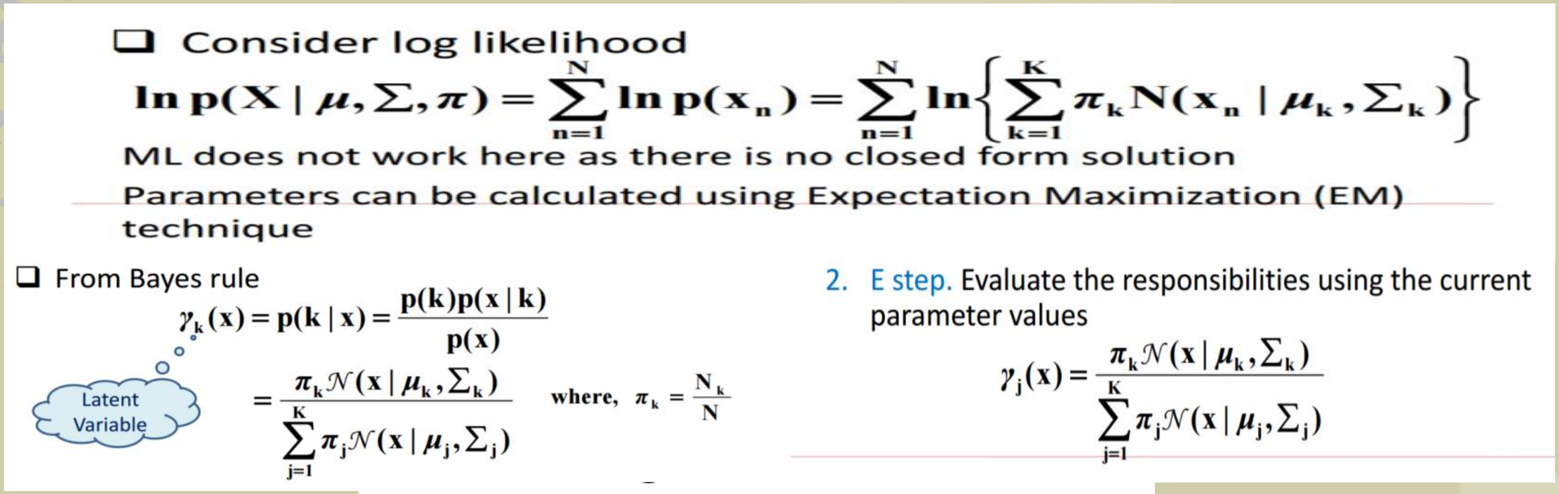
- GMMs are used to represent frame-based speech features
- Used for estimating acoustic likelihoods
- GMMs are a weighted sum of multivariate Gaussians

$$f(x|\mu,\Sigma) = \sum_{k=1}^{M} c_k \frac{1}{\sqrt{2\pi|\Sigma_k|}} exp[(x-\mu_k)^T \Sigma^{-1} (x-\mu_k)]$$



Gaussian Mixture Model

- The basic aim of Using Gaussian Mixture Model is to minimize the log likelihood function



❑ Consider log likelihood

$$\ln p(X \mid \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln p(x_n) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}$$

ML does not work here as there is no closed form solution

Parameters can be calculated using Expectation Maximization (EM) technique

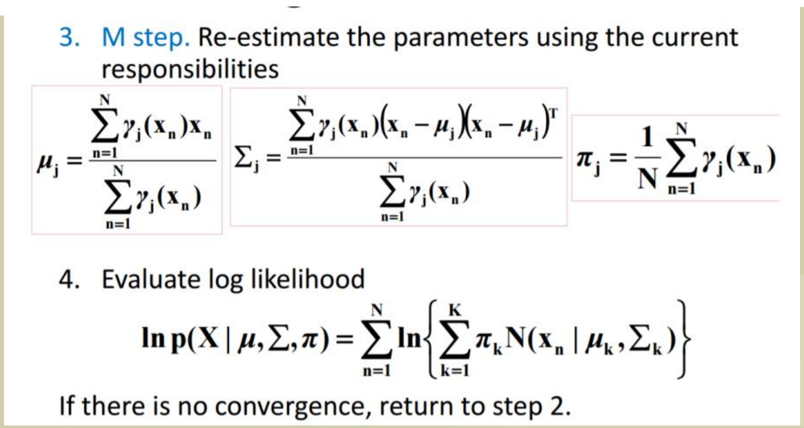
❑ From Bayes rule

$$\gamma_k(x) = p(k \mid x) = \frac{p(k)p(x \mid k)}{p(x)}$$

$$= \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)} \quad \text{where, } \pi_k = \frac{N_k}{N}$$

2. E step. Evaluate the responsibilities using the current parameter values

$$\gamma_j(x) = \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)}$$



3. M step. Re-estimate the parameters using the current responsibilities

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)} \qquad \Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)} \qquad \pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n)$$

4. Evaluate log likelihood

$$\ln p(X \mid \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k) \right\}$$

If there is no convergence, return to step 2.

Linear Discriminant Analysis(LDA)

- It is a composite, 2-stage technique, first reduce dimensionality, then classify.
- The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible
- Why to use LDA:
  - Shorter training time
  - Avoids the curse of dimensionality
  - Enhances generalization by reducing overfitting

Linear Discriminant Analysis(LDA)

- The generalization of the within-class scatter matrix

$$S_W = \sum_{i=1}^{C} S_i$$

$$\text{where } S_i = \sum_{x\in\omega_i} (x-\mu_i)(x-\mu_i)^T \text{ and } \mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x$$

- The generalization for the between-class scatter matrix

$$S_B = \sum_{i=1}^{C} N_i(\mu_i-\mu)(\mu_i-\mu)^T$$

$$\text{where } \mu = \frac{1}{N}\sum_{\forall x} x = \frac{1}{N}\sum_{x\in\omega_i} N_i\mu_i$$

- For the (C-1) class problem we will seek (C-1) projection vectors w_i, which can be arranged by columns into a projection matrix W=[w1|w2|…|wC-1] so that

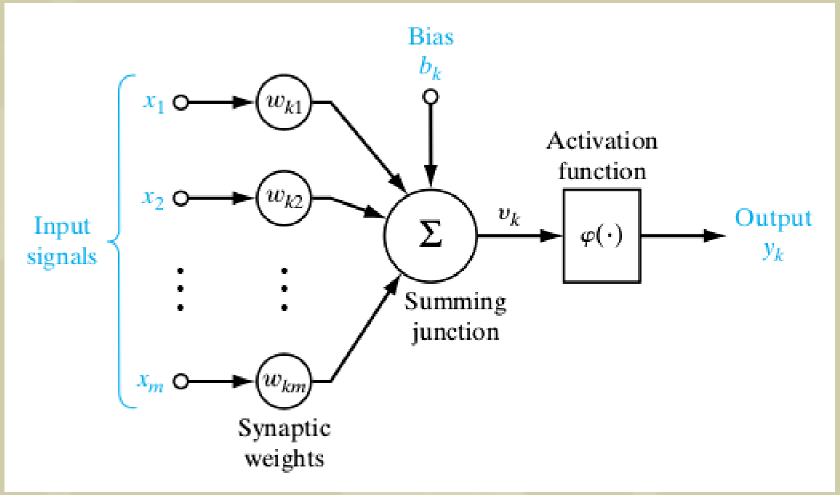
$$y_i = w_i^T x \Rightarrow y = W^T x$$

- Optimal projection matrix W* is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the given generalized eigenvalue problem. The new vector y can be used as our set of reduced feature.

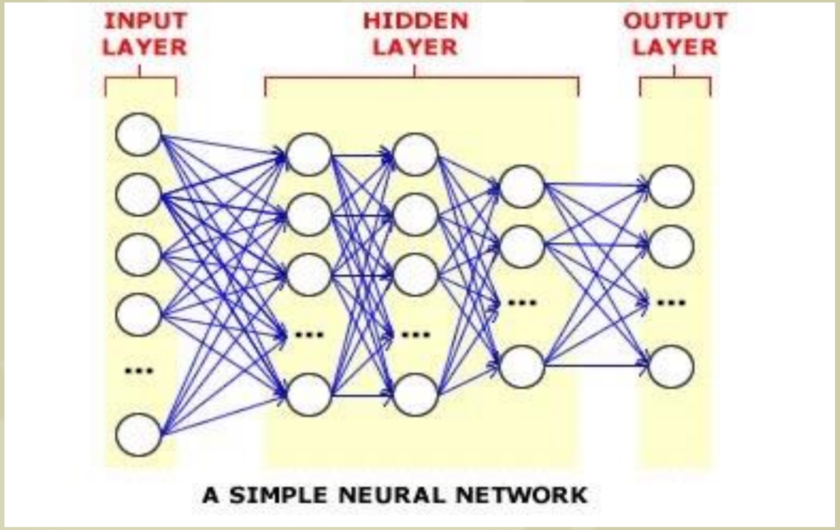
$$W^* = \left[w_1^* \mid w_2^* \mid \cdots \mid w_{C-1}^*\right] = \operatorname{argmax}\left\{\frac{W^T S_B W}{W^T S_W W}\right\} \Rightarrow (S_B - \lambda_i S_W)w_i^* = 0$$

Deep Learning

Model of a “Neuron”:



Deep neural network is a feed-forward artificial neural network with multiple hidden units between input and output



Why Deep Learning?

- DNNs have ability to learn complicated feature representations and classifiers jointly
- Learn much better models of data that lie on or near a non-linear manifold
- Performance does not saturate with increase in training data
- Deep learning proven to be better than other Models

DNN Proposed Structure

- Our proposed DNN at the 1st stage will contain 1 hidden layer with 13 MFCC features, 13 Delta MFCC and 13 Delta Delta MFCC.
- The number of outputs is equal to the number of languages to be identified and the outputs are the respective probabilities of the languages.
- We intend to add Formants as our input features to learn from speech pronunciations
- The DNN at the second layers will be trained on features selected for binary classification b/w each pair of languages using LDA and they will be stored in a confusion matrix
- These binary classifiers will have 2 outputs for each language.

DNN Parameters

- The DNN at 1st layer uses sigmoid function as it’s activation function
- The outputs are converted into probabilities by averaging over all activation values.
- DNN uses Categorical Crossentropy as it’s cost function which it tries to minimize.
- The Binary Classifiers use similar activation functions but uses binary crossentropy as it’s cost.

Tools and Libraries used

pyAudio Analysis A Python library for audio feature extraction, classification, segmentation and applications

pyAudioAnalysis

- pyAudioAnalysis is a Python library covering a wide range of audio analysis tasks.
- It is used for framing and audio extraction. The python library is used to obtain both short term and averaged mid term features
- We are using it to extract the MFCC Features. It takes the audio as wav file and depending on the window size and hop length returns the MFCC coefficients of each feature window

Tools and Libraries used

LibROSA

- LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.
- We have used it to calculate Delta and Delta Delta MFCC coefficients for each frame.

Tools and Libraries used

scikit-learn

- Used for Machine Learning in Python
- Provides simple and efficient tools for data mining and data analysis
- We have used it for extracting the Gaussian Mixture Models

Tools and Libraries used

Keras

- Keras is a high-level neural networks library, written in Python and capable of running on top of either Tensor Flow or Theano. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*
- We used it to model our Neural Network efficiently and set it various parameters.