Spoken Language Identification using Neural Network

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

- The problem of automatic language identification (LID) can be defined as the process of automatically identifying the language of a given spoken utterance.

- LID can be used by speech recognition systems, multilingual translation systems or call-centers(e.g., emergency calls) routing.

- Also LID can be used by intelligence and security, where the language identities of recorded messages need to be established before any information can be extracted.

Introduction

- Our project aims at spoken language identification in speech audio samples using Deep Learning Models (DNNs).
- We have used acoustic modelling for our LID system. Audio is divided into small chunks called frames and raw audio signal from each frame is transformed by applying the mel-frequency cepstrum.
- The coefficients from this transformation called MFCCs are used as input to the acoustic model.

Problem Statement

To construct efficient as well as effective Language Identification (LID) model using two stage Neural Network.

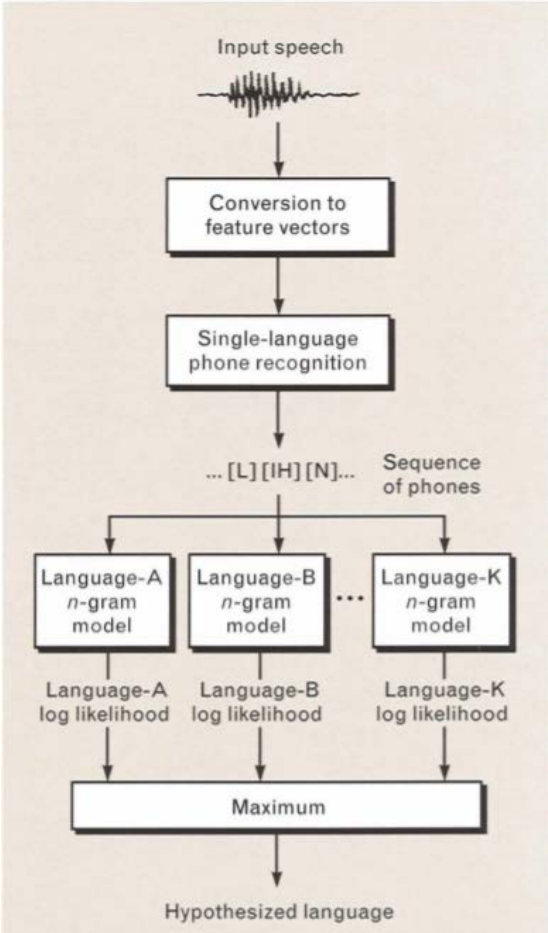
Related Work

1. **MA Zeissman in Lincoln Laboratory[1]**

— uses PRLM(Phone Recognization followed by language modelling)

— It consists of single language phone recognition followed by language modeling with an n-gram analyzer.

— PRLM can be further extended to Parallel PRLM and Parallel Phone Recognition.



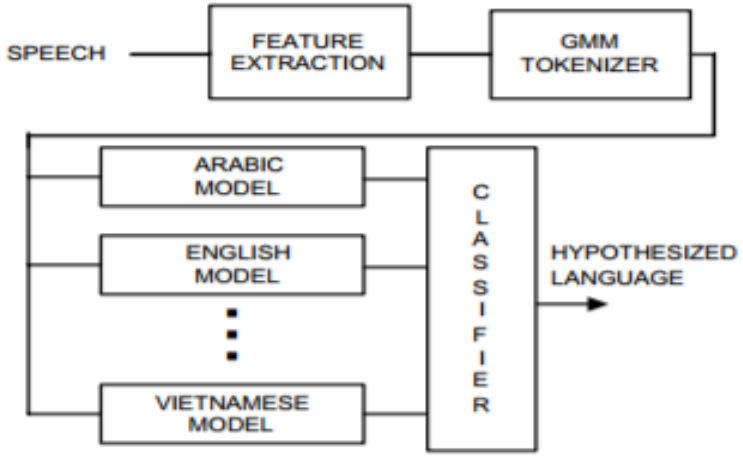
[1] M. Zissman, "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," IEEE Trans. Acoust., Speech, Signal Processing, vol. 4,no. 1, pp. 31–44, 1996.

2. KshirodSarmah and UtpalBhattacharjee[2]

— GMM based Language Identification using MFCC and SDC Features

3. Pedro A. Torres-Carrasquillo[3]

— Language Identification using Gaussian Mixture Model Tokenization



[2] Kshirod Sarmah, "GMM based Language Identification using MFCC and SDC Features", International Journal of Computer Applications (0975 – 8887) Volume 85 – No 5, January 2014

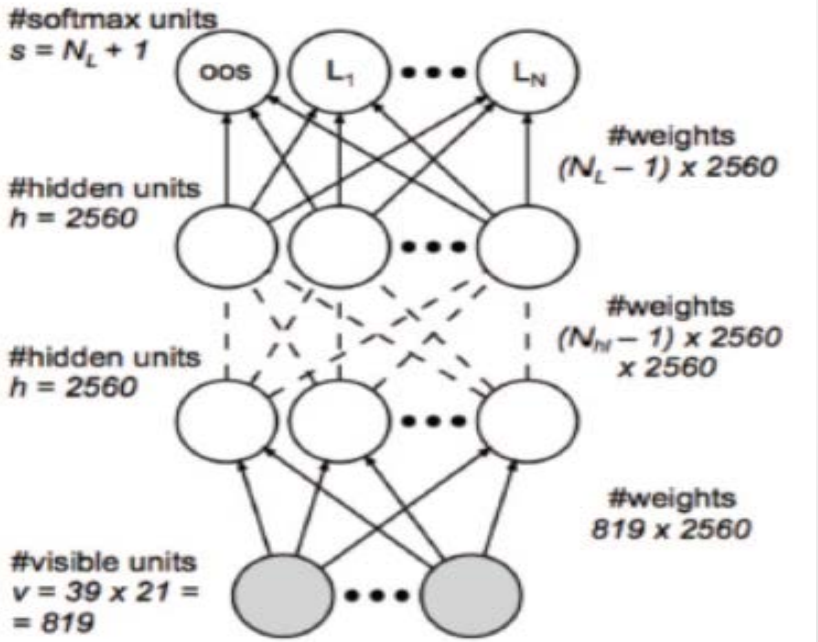
[3] Pedro A. Torres-Carrasquillo, "Language Identification using Gaussian Mixture Model Tokenization", Lincoln Laboratory, Massachusetts Institute of Technology

4. Julien De Mori[4]

- Presented and compared Support Vector Machine(SVM) based methods, Gaussian Mixture Model(GMM) based methods and Neural network.
- Used Neural Network and Computing features for each 25ms frame.

5. Ignacio Lopez-Moreno[5]

- Automatic Language Identification Using Deep Neural Networks
- Motivated by their recent success in acoustic modelling.
- The proposed approach is compared to state-of-the-art i-vector based acoustic systems.



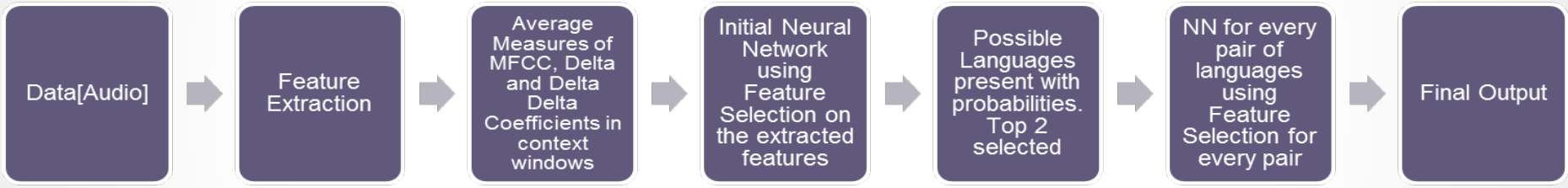
[4] Julien De Mori, "Spoken Language Classfication" , CS 229 – Machine Learning(Stanford)

[5] Ignacio Lopez-Moreno, "AUTOMATIC LANGUAGE IDENTIFICATION USING DEEP NEURAL NETWORKS", 2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP),2014 (context windows)

Our Proposal

- We propose a hybrid neural network that consists of two levels.
- An initial neural network gives the probability of the belongingness of data sample to all class of languages.
- The second level, or the binary classifier finally outputs the class to which the data sample belongs amongst the best two sub-candidates.
- The idea behind this hypothesis is the fact that different languages need different set of features to differentiate them.

Experimental Setup



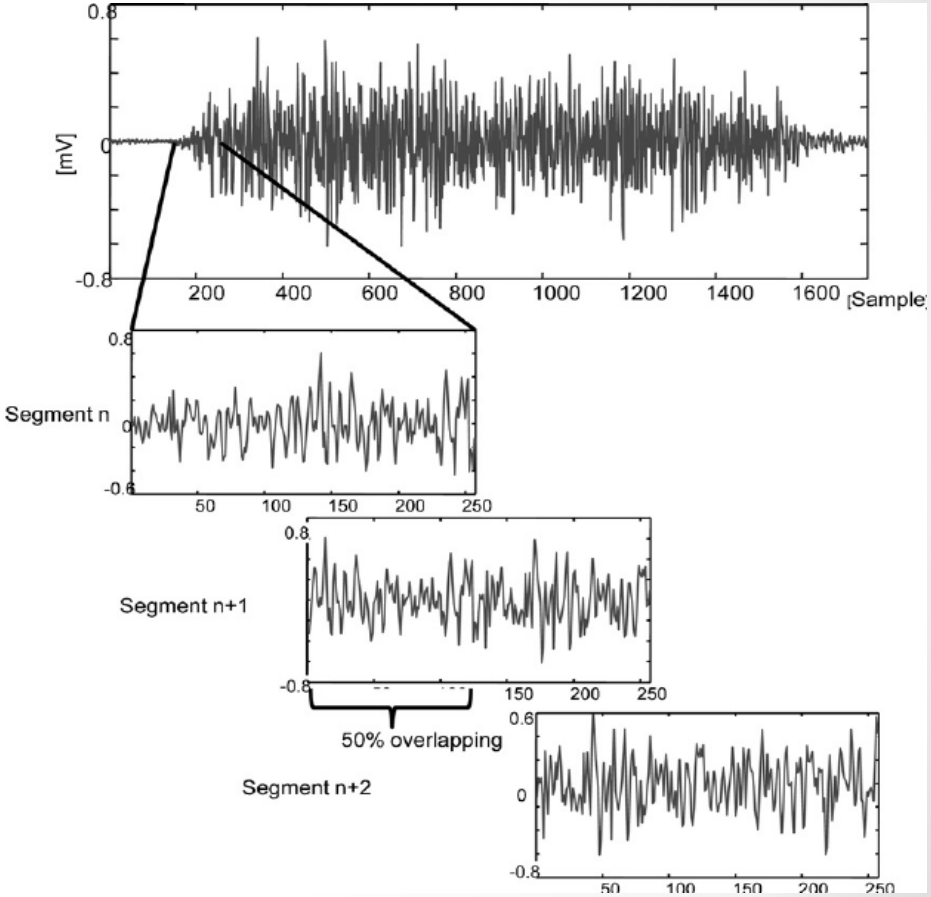
Workflow of project

Data Set

- We obtained our language dataset from Audio Lingua.
- The dataset includes 170 distinct speakers for each language.
- The dataset is assumed to be clean. It is pre-processed and split to multiple equal sized (5 sec) audio samples.
- One audio samples is treated as a single data point. This is done to average out any noise component in the speech.

Feature Extraction

- Audio Sample is divided into frames of 25ms(400 frames) with a window hop of 10ms(100 frames).
- Sliding Window is used processing the audio sample.
- This overlapping of windows prevents spectral loss that is caused by framing.
- For each frame 13 mfcc features are calculated.
- Using these mfcc features, delta mfcc, delta delta mfcc is calculated.



Sliding Window

Feature Vector

- For each audio sample of 5 sec we obtain a long term averaged feature vector with 390 features.
- These features include 13 MFCC+13 Delta MFCC+13 Delta Delta MFCC. Making a set of 39 sized feature vectors.
- Such consecutive features are stacked to create **context windows** and the number of consecutive features used is called context window size. We used a context window size of 5.
- Long term averaging is done by computing the mean and standard deviation over the entire duration giving a single feature vector of size of shape (1, 390)
- 39(mfcc + delta mfcc + delta delta mfcc) * 5 (context window size) * 2 (mean + standard deviation) = 390
- When x such samples are taken out feature vector has a shape of (x, 390)

Feature Selection

- Advantages: Shorter training and prediction time, weeds out irrelevant features, less resource are used, prevents overfitting and avoids curse of dimensionality.

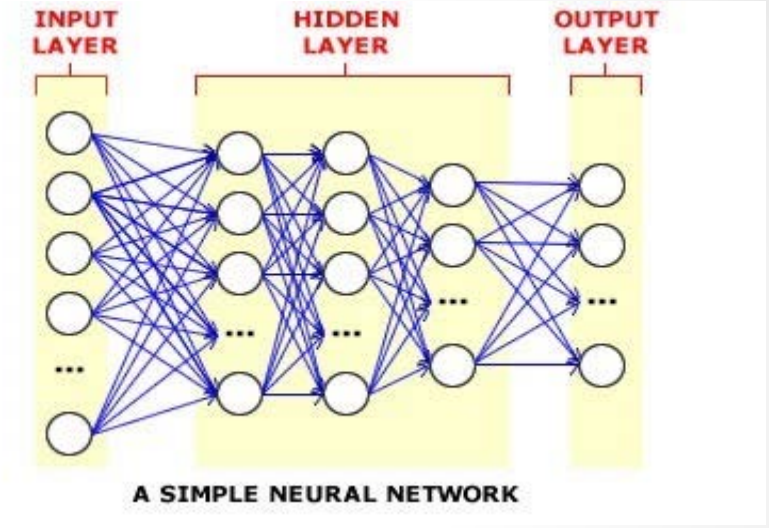
- Pearson's chi-squared statistics is used to calculate p-value of each feature using formula:

$$\chi^2 = \sum_{i=1}^{n} \frac{(\mathrm{O}_i - \mathrm{E}_i)^2}{\mathrm{E}_i}$$

- Feature are then selected on the basis of their respective p-values.

Deep Learning

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

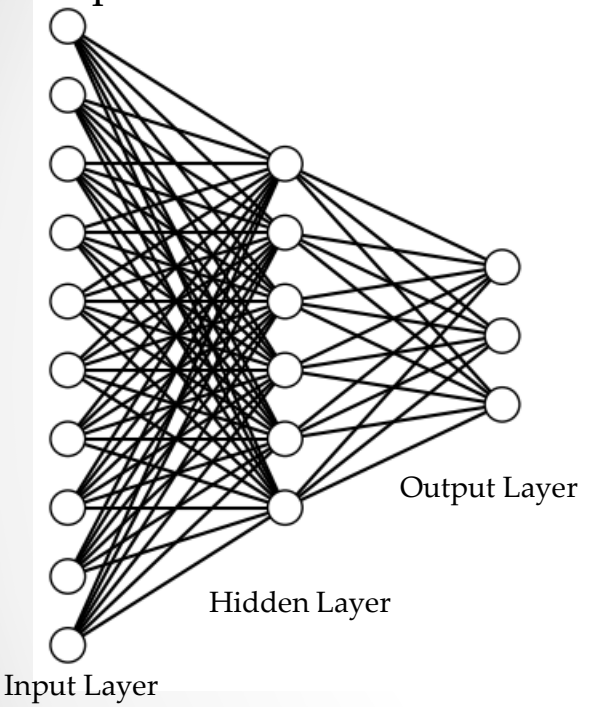


Why Use Deep Neural Networks?

- 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
- Surpass the performance of the other dominant paradigms

NN Architecture

Initial Neural Network
Representative Architecture



Common Parameters

Activation Function:

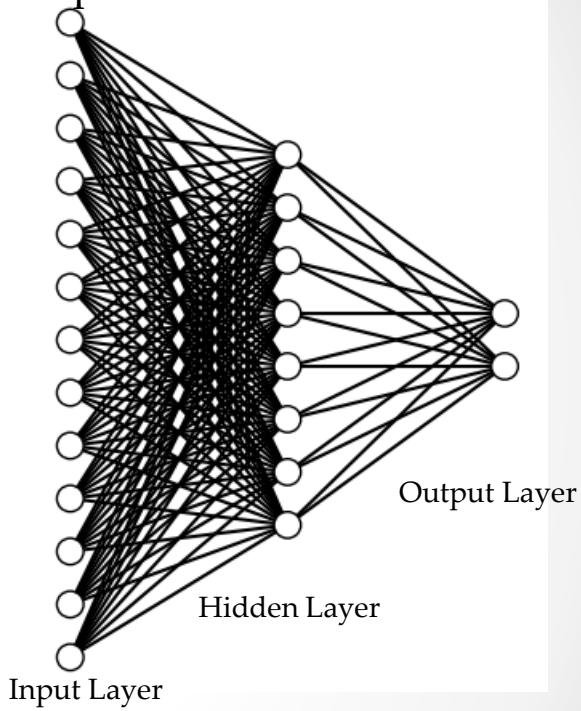
- Hidden Layer : ReLU
- Output Layer : Softmax

Regularizer : L1L2

Optimizer : Adadelta

Metric : Accuracy

Binary Neural Network(s)
Representative Architecture



Hidden Layers : 1
Input Dimension : Number of Selected Features : 180 Neurons
Hidden Layer : 12 Neurons
Output Layer : Number of Languages in set : 3 Neurons
Loss Function : Categorical Crossentropy

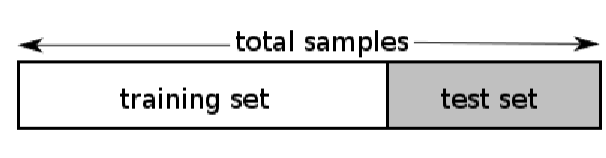
Hidden Layers : 1
Input Dimension : Number of Selected Features : 380 Neurons
Hidden Layer : 22 Neurons
Output Layer : 2 Neurons
Loss Function : Binary Crossentropy

Hybrid NN Model Training

- Both Initial and Binary are using **Keras** Library which is a high-level neural networks library, written in Python, capable of running on top of either Tensor Flow or Theano and enables fast experimentation. It allows easy specification of the characteristics of the network.
- The parameters associated with both NN's were found through extensive testing.
- The Initial NN is trained with audio speech samples for all languages and produces belongingness probability of the data sample to each class. It is used for classifying any number of languages in the set.
- Binary NN(s) are trained with audio speech samples for all language pairs and is used for classifying two different languages.
- Final output is produced by using the top 2 most probable languages and inputting them to the appropriate Binary Classifier from which the most probable language is selected.

Testing

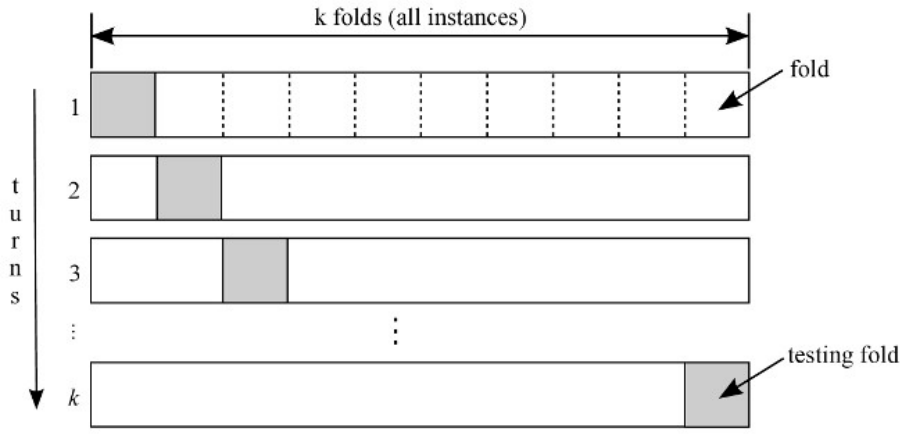
- **HoldOut Validation**: The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The function approximator fits a function using the training set only. Then the function approximator is asked to predict the output values for the data in the testing set (it has never seen these output values before).



In this type of validation, we set out 60% of the entire data for training and rest 30% for holdout test. The 30% test samples were independent of training samples, i.e. even the speakers were distinct in both the sample sets.

Testing

- **KFold Validation**: K-fold cross validation is one way to improve over the holdout method. The data set is divided into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all *k* trials is computed.



While doing K-Fold Validation we used k=10, thus the entire model had to be trained 10 times. The standard deviation and average accuracy is specified in the results section.

Results

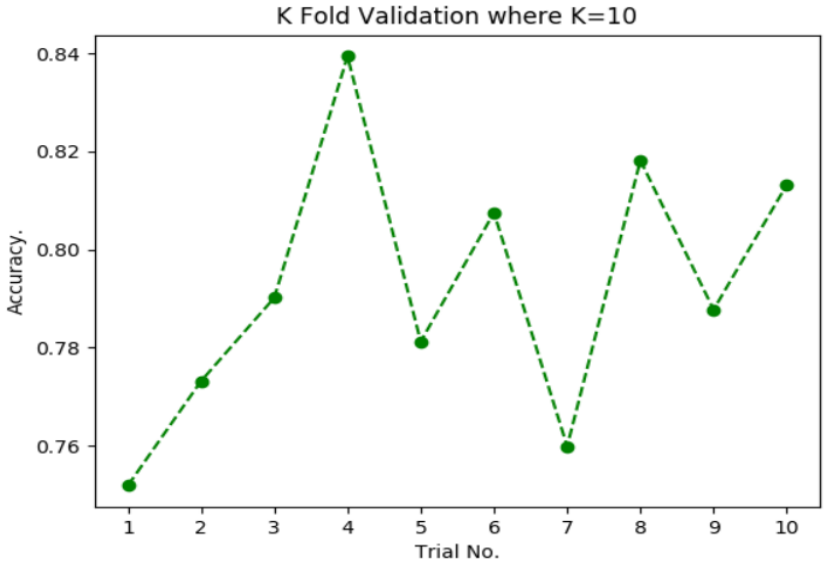
Overall accuracy of the system using holdout validation technique

	Chinese	French	German
Chinese	160	23	53
French	36	155	45
German	25	41	170

Accuracy: 79.20%

Results

Overall accuracy curve using KFold Validation Technique



Results of the K Fold Validation of the language detection system. The average accuracy is 79.22% and a low standard deviation of 2.59 in the percentage accuracy is indicative of a stable system.

Binary Classification Chinese vs French

	Chinese	French
Chinese	222	14
French	11	225

Accuracy: 94.70%

Binary Classification German vs French

	French	German
French	225	11
German	27	209

Accuracy: 91.95%

Binary Classification Chinese vs Germany

	Chinese	German
Chinese	192	44
German	27	209

Accuracy: 84.96%

Results

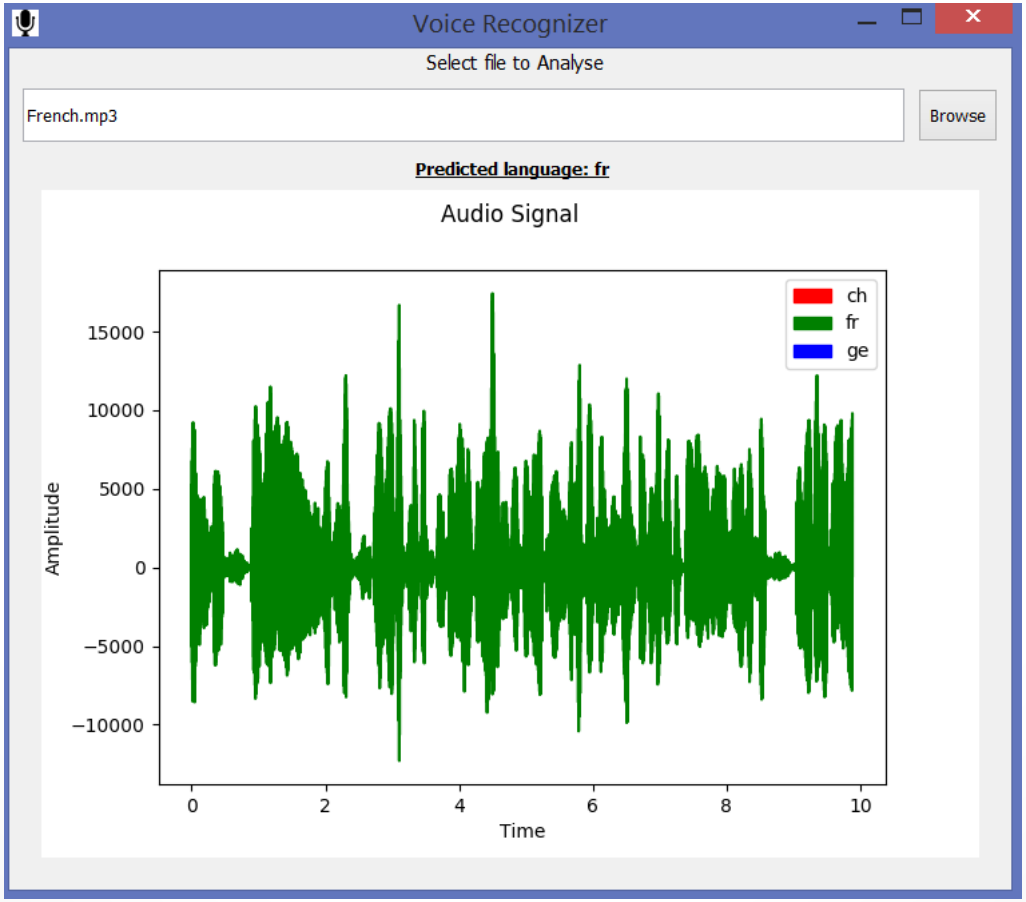
Comparison of the language identification system with SVM Based Baseline Method

	Baseline SGD Classifier	Hybrid Neural Network
Accuracy	72.60%	79.20%
Precision	62.26%	68.87%
Recall	58.90%	68.50%

Prediction Visualization through GUI

- GUI was made to illustrate the functioning of the Hybrid Neural Network Model.
- **PyQt5** was used to design the GUI.
- Allows the user to select any .mp3 file.
- Provides frame wise language prediction for the audio sample.
- Provides final language prediction for the audio sample.

GUI Sample


Voice Recognizer
Select file to Analyse
French.mp3
Browse
Predicted language: fr
Audio Signal
ch
fr
ge
15000
10000
5000
0
−5000
−10000
Amplitude
0
2
4
6
8
10
Time

Real Time Analysis

- Fast prediction time of hybrid model allowed us to do real time analysis of audio.
- pyAudio was used as audio I/O library.
- Thread were used to gather raw audio from source and pass that to our prediction model in chunks of 1 second.
- Processing time for 1s audio sample was observed to be around 0.2s.
- When using default device's microphone as source result was observed to be not so accurate due to noise being high in audio samples.

Conclusion

- Acoustic modelling together with DNN is powerful enough to identify languages.
- Languages are better resolved when considered pairwise as a feature which is useful in distinguishing some languages was irrelevant for some pair of languages.
- Our hybrid two stage classification model is definitely an improvement over normal single stage classifier.

Future Work

- Adding pre filter for noisy audio.
- Using Shifted Delta Cepstral(SDC) features
- Providing facility for incremental training.
- Adding out of set option as possible output.
- Improving real-time capabilities

Thank You

谢谢

Merci

Danke