Language Identification Using Acoustic Features

Research and Development Project (EE 691)

Kalpesh Patil (130040019)

Guide: Prof. Preeti Rao

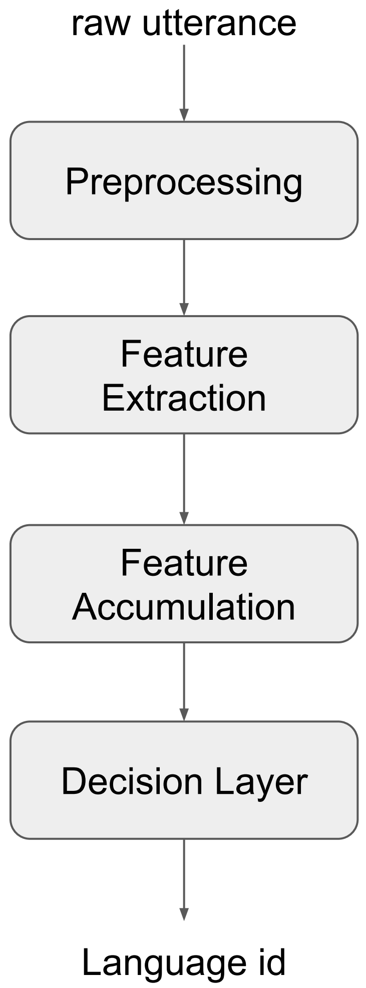
Outline

- Introduction
- Dataset Description
- Feature Extraction
- GPPS
- DNN
  - Context-free DNN
  - Context DNN
- Bottleneck Features
  - BNF and Fine Tuning
  - BNF and GPPS
- Other Approaches
  - LSTM
  - 1D-CNN
- Conclusion and Future Work

Introduction

- Preprocessing
  Noise removal, silence removal

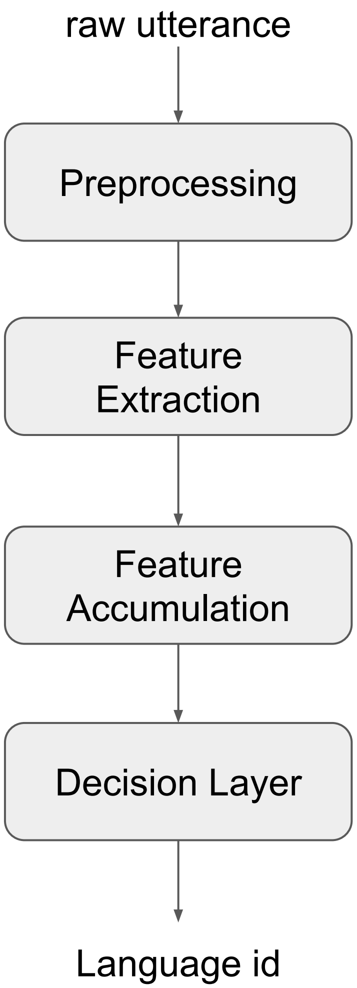
- Feature Extraction
  - Acoustic features
    MFCC, MFBE, SDC etc.
  - Phonotactic
    Discriminate language based on phones
    e.g. N-grams, C-V features
  - Prosodic features
    F0 contours, Stress, Intonation



Introduction

- Feature Accumulation
  Sequence of features to single representation

- Decision Layer
  Top end classifier
  e.g. Neural network, SVM



Dataset Description

- CallFriend datasets: half an hour long telephone conversations

- Data preprocessing
  - Silence removal
    Energy based thresholding
  - Speaker Independent splitting
    No common speakers in train and test splits
  - Small chunks of 10sec as a unit utterance

Language	Train	Test
Hindi	3388	813
Tamil	2356	767
Total	5744	1580

Feature Extraction

- MFCC features with delta and delta-delta coefficients
- CMVN normalization to remove bias due to local environmental conditions

Vector Quantization

- Codebook generation for each language
  - MFCC features of each frame of each language as input
  - MiniBatchKMeans for clustering

- A test utterance is classified based on the average distance of its frames from the nearest clusters of each genre

- Performance improves as number of clusters increases

Cluster size	200	300	400	500
Accuracy	75.57	76.77	77.08	77.91

GPPS

Gaussian Posterior Probability Supervector

- GMM-UBM training
  - GMM trained on MFCC features of entire training data
  - Diagonal covariance matrices for reduced no. of parameters

- GPPS extraction

$$Pr(o_t|\lambda) = \sum_{j=1}^{J} w_j Pr(o_t|\mu_j, \Sigma_j)$$

$$\lambda = \{w_j, \mu_j, \Sigma_j\}, j = 1, 2..J$$

$$\kappa_j = \frac{1}{T} \sum_{t=1}^{T} \frac{w_j Pr(o_t|\mu_j, \Sigma_j)}{\sum_{j=1}^{J} w_j Pr(o_t|\mu_j, \Sigma_j)}$$

$$\kappa = [\kappa_1, \kappa_2, ...\kappa_J]$$

GPPS

- Classifier

  NN with architecture

  InputLayer(J), Dense(100,relu), Dropout(0.5), Dense(10,relu), Dropout(0.5), Dense(2)

- Performance improves as number of GMM components increase

GMM components	64	128	256	512
Accuracy	83.35	84.81	88.10	90.06

DNN

- Context-free DNN

- Input: MFCC feature vectors of entire training data
  Output: Language id

- Final decision making
  - Majority rule
  - Maximum Likelihood

$$\hat{L} = \underset{i}{argmax} \prod_{n=1}^{N} Pr(y_n = i|x_n)$$

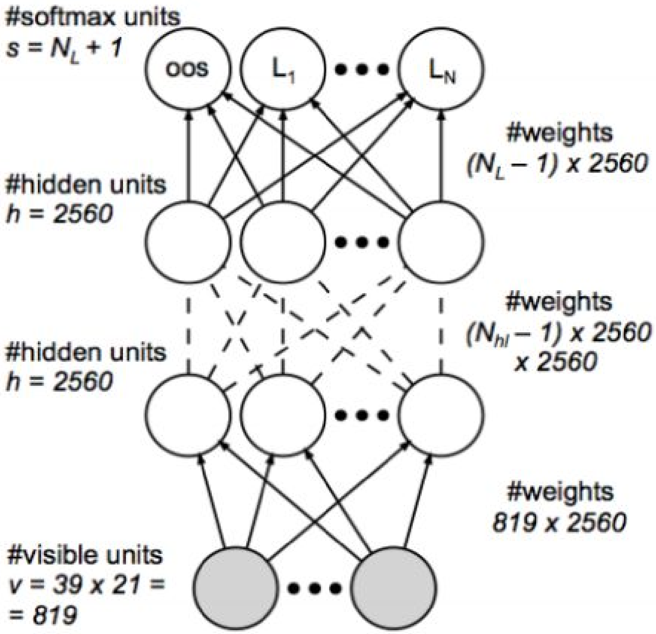
$$\therefore \hat{L} = \underset{i}{argmax} \sum_{n=1}^{N} log(Pr(y_n = i|x_n))$$

- Drawbacks
  Didn’t consider temporal connections across frames

DNN

- Context DNN
  - Input: a frame with Nc left and right context frames
    Output: Language id
  - Final decision making
    Majority rule
    Maximum likelihood

	Majority vote	Maximum total likelihood
Context-free DNN	64.34	65.21
Context DNN	78.57	80.12



Representational image for network topology of context-DNN

Source: I. Lopez-Moreno, et al, "Automatic language identification using deep neural networks"

Bottleneck Features

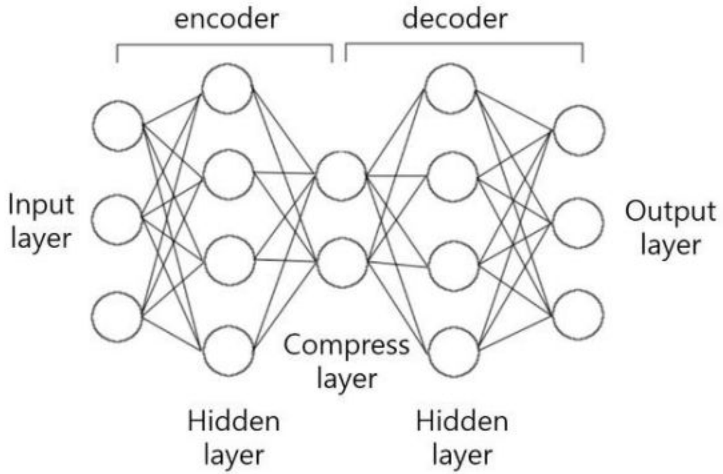
- Drawbacks of traditional features
  - Hand-crafted features
  - Fixed method of extraction without considering end-goal
  - Language information is latent

- Deep Bottleneck Features
  - Restricted Boltzman Machines
  - Autoencoders
  - Stacked Autoencoders

Bottleneck Features

Autoencoders

- Reconstruct input at the output layer
- Encoder
- Decoder
- Features at the bottleneck layer



BNF and Fine Tuning

- Input: (39(2Nc+1)) dimensional context frames
  Output: same as input
  Objective: Minimize mean squared loss

- Architecture
  InputLayer(429), Dense(1000,relu), Dropout(0.5), Dense(200,relu), Dropout(0.5), Dense(50,relu)(also the bottleneck layer), Dense(200,relu), Dropout(0.5), Dense(1000,relu), OutputLayer(429)

- Fine Tuning
  - Cut decoder part and add softmax layer
  - Fine tune network with small learning rate

	Majority vote	Maximum total likelihood
Context DNN	80.27	82.46

Results after pretraining with autoencoder

BNF and GPPS

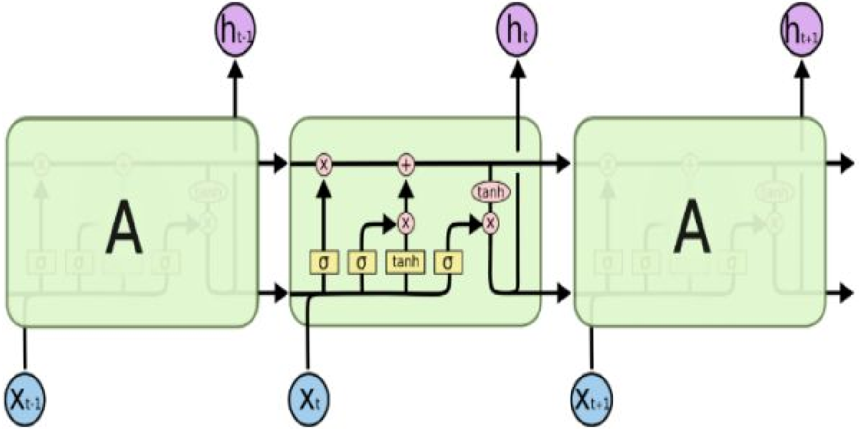
- BNF
  - Better representation of data
  - More discriminative power than MFCC

- GPPS
  - Better classifier

- Best of both worlds
  - GPPS with BNF instead of MFCC
  - Accuracy **92.39%** (best so far)

RNN-LSTM

- Sequence classifier
- RNN
  - Unrolling network in time
  - Suffer from vanishing gradients
- LSTM
  - Memory cells (analogous to conveyer belt)
  - Input gate, forget gate
  - Cell state are expected to contain information related to language
  - Accuracy: **70%***

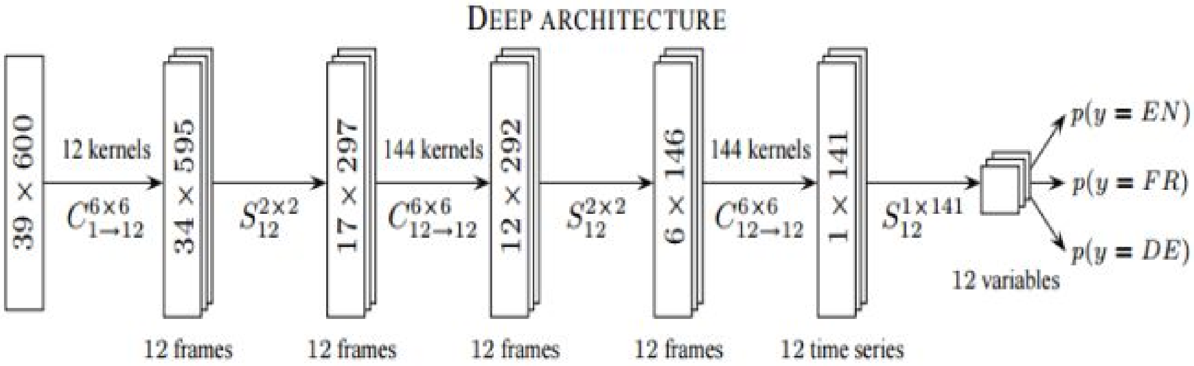


Representational figure for LSTM

Source: "Understanding lstm networks," http://colah.github.io/posts/2015-08- Understanding-LSTMs

*preliminary result, more experimentation required

CNN



Representational diagram of 2D-CNN

Source: G. Montavon, "Deep learning for spoken language identification", NIPS

- Local connectivity
  Weight sharing

- 2D-CNN has been explored for language classification

- 1D-CNN more intuitive than 2D-CNN because local connectivity in temporal dimension only (and not along MFCC feature dimension)

- Accuracy: 78.32%*

*preliminary result, more experimentation required

Mandi Dataset

- Dialects identification for Marathi language
- Coastal vs Eastern dialects
- GPPS with 512 clusters
  - Accuaracy: **66.57%**
- Possible reasons for low performance
  - High background noise
  - Not enough discrimination in dialect captured in MFCC features

Accent	Train	Test
Coastal	567	170
Eastern	568	180
Total	1135	350

Conclusion and Future Work

- BNF found to be better features than MFCC

- BNF + GPPS performed better than all other techniques in terms of accuracy

- Future Work
  - Transfer learning
    Exploit well trained model trained on larger datasets like CallFriend to be able to use them on smaller datasets like Mandi
  - Dwelling more into advanced methods like LSTM, 1D-CNN etc.

Thanks

Questions?