Identifying gender using Speech Patterns

- Sooraj Mangalath Subrahmannian

Agenda

- Data Preparation
- Feature engineering and data augmentation
- Model development
- Results
- Future works

Data preparation

The data preparation was the crucial step. Here I have prepared the data for both the current model development and for future works.

Precautions:

- The data was initially segregated such that there is no overlap of speakers in the train and test data to avoid data leakage
- The data was split into train and development (validation) set using Stratified sampling with respect to gender

Two types of input data was created for modeling the speech patterns

- MFCC (**Mel-frequency cepstral coefficients**)
- Mel spectrogram

Data preparation (contd)

After a literature study on the experiments done by others before me, it was noted that Mel spectrogram were much effective in classification of speech pattern while using a deep learning model.

The reason for such a decision was because the MFCCs are obtained from Mel spectrograms after removing correlation using a linear transformation technique known as DCT.

Hence, if we use MFCC, we will end up losing non linear relations in the input data.

Since deep learning perform very well with correlated data, we need not convert Mel spectrogram to MFCCs.

Mel spectrograms from speech samples

Windowing → Fourier Transform → Filter banks → Mean normalization

Windowing: To perform Fourier transform, speech sample needs to be stationary, Hence speech samples are made into short overlapping window

Fourier Transform: To make the data more denser

Filter banks: Features

Mean normalization: improve Signal to noise ratio

Feature engineering/ Data augmentation

- Padding to make every sample has a stationary length
- Chopped the sample when the length of sample is greater than 95 percentile of all the lengths
- Label encoding of all y labels
- Data augmentation was performed such that, when a sample is longer, a random portion of speech sample spectrogram was used for training

Model Architecture and training

2 convolutional layers
Batch normalization post every layer
Dropouts after each layer
4 Fully connected layers

All intermediate activation functions had RELU and final layer had sigmoid

Training was done using Adam optimizer with a learning rate of 0.01

Results

The model had a log loss of 0.83

Future works

- Add more data and perform
- Try more architectures
- Try more labels on this data set

THANKS!

Any questions?

You can find me at

smsubrahmannian@usfca.edu
soorajviraat@gmail.com
https://github.com/smsubrahmannian