Speaker Identification: Text Independent Context

Final Report: CS230

Course	**CS230: Deep Learning (Spring 2018)**
Project	**Speaker Identification: Text Independent Context**
Category	**Speaker recognition, voice recognition**
Team	**Rish Gupta (rishg), Manish Pandit (manish7), Sophia Zheng (xszheng)**
Submitted	**6th June 2018**
github	**https://github.com/manishpandit/speaker-recognition.git**

Abstract

This document provides the final report of our speaker identification project. We address the problem of identifying a speaker based on a short audio signals from among the known set of speakers enrolled during the model training, with an emphasis on text-independent speaker recognition. Traditional approaches based on Gaussian mixture model and Universal background model (GMM-UBM) have high success rate but with a higher computational cost during the GMM evaluation phase. We experiment with a deep learning architecture based on convolutional neural network (CNN). Our CNN model is trained and tested against freely available and comprehensive VoxForge (voxforge.org) dataset and provide constant evaluation cost. With our efforts through this quarter, we have successfully built a speaker identification algorithm with extraordinary accuracy (96%), and also developed an effective hyperparameter tuning algorithm to search for optimal hyperparameters during the model training. We submit a detailed summary of the project along with the methods and results.

Introduction

The human speech signal conveys many levels of information. At the base level it carries a message in words. But at other levels, it conveys information about language, dialect, emotion, gender and identity of the speaker. While the *speech recognition* systems aim to identify the words spoken in the speech, the goal of the *speaker recognition* system is to extract the identity of the speaker associated with the speech signal.

The broad area of speaker recognition emcompasses two more fundamental tasks. *Speaker verification* (also known as speaker authentication) is a task of determining whether a person is who she claims to be. *Speaker identification* is a task of determining who is speaking from a known set of speakers. The unknown speaker makes no identity claim so the system must perform a 1:N classification.

These tasks can be further divided into text dependent and text independent categories. In a *text dependent* system the recognition system has prior knowledge of the text been spoken to. In a *text independent* the recognition system is agnostic to the associated text.

Our focus is the problem of speaker identification in the text independent context. Further, we will concentrate this study on short speeches (usually 2-5 seconds) from large number of speakers.

Related Work

Research and development on speaker recognition methods and techniques has been undertaken for well over four decades and it continues to be an active area. Approaches have spanned from human aural and spectrogram comparisons, to simple template matching, to dynamic time-warping approaches, to more modern statistical pattern recognition approaches, such as neural networks, Gaussian Mixture Models (GMM) and Hidden Markov Models (HMMs). It is interesting to note that, although striving to extract and recognize different information from the speech signal, many of the same features and techniques successfully applied to speech recognition have also been used for speaker recognition.

Generally, in the speaker identification systems, the speech signal is first processed to extract features conveying speaker information. Then these features are compared to a repository of models, obtained from previous enrollment, representing the speaker set from which we wish to identify the unknown voice. For closed-set identification, the speaker associated with the most likely, or highest scoring model is selected as the identified speaker. This is simply a maximum likelihood classifier.

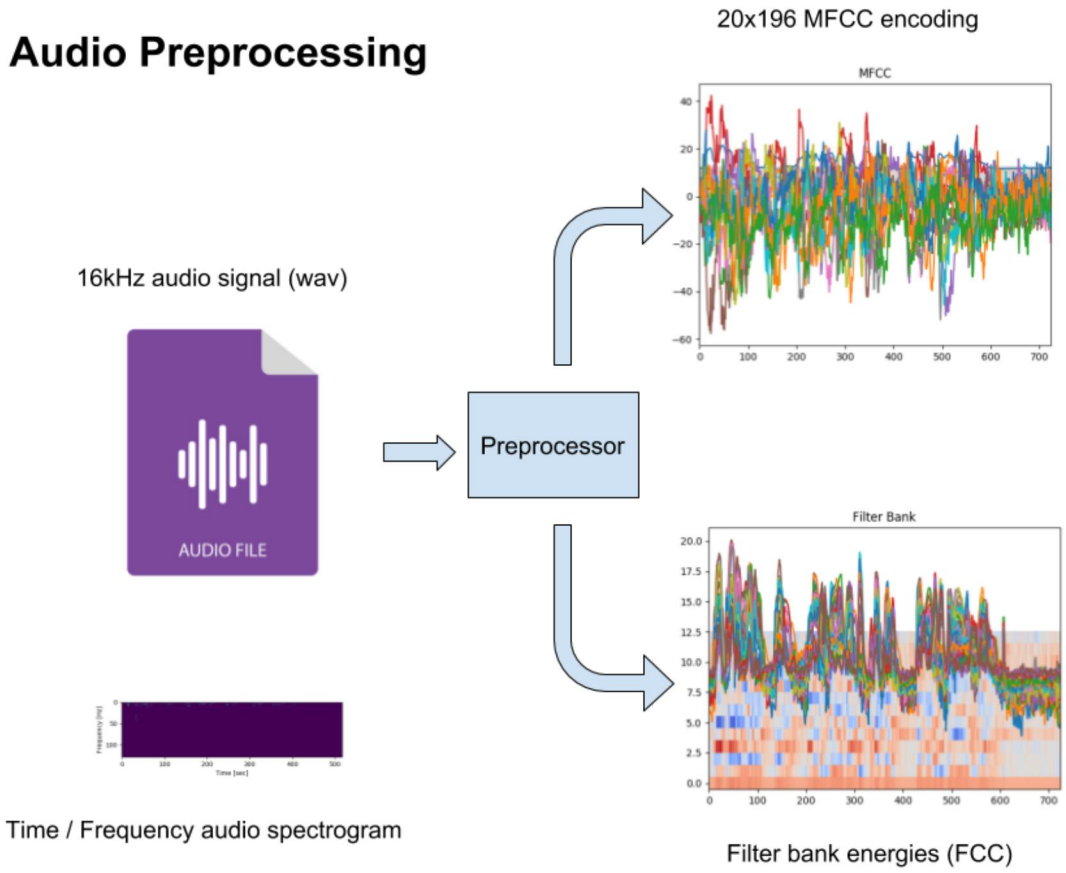
The mainstream neural net approach to speaker recognition is similar to face recognition, where models are explicitly trained to discriminate between the speaker being modeled and some alternative speakers. Training can be computationally expensive and models are sometimes not generalizable.

Our approach is to train the model for a known set of speakers. The prediction task is to identify an unknown audio signal from a set of known speakers in constant time. This approach yields highly performant speaker identification with the downside that any new enrollments are expensive.

Dateset

Our choice of audio dataset is open source VoxForge dataset. It is freely available under GNU General Public License. VoxForge was set up to collect transcribed speech for use in Open Source Speech Recognition Engines ("SRE"s). The dataset contains **1216** unique speaker's multiple audio files in wav format. Each speech is of short duration (2-10 seconds).

The voxforge dataset contains few samples where the speakers are not known and hence grouped under anonymous category. We decided to exclude these samples from our project since they just impede the learning. During the pre-processing stage, the wav files are converted to Mel-frequency cepstral coefficients (MFCCs) matrix of shape 20x196x1. MFCCs can approximate the human auditory system response more closely than the linearly-spaced frequency bands used in the normal cepstrum. We experimented with Filter Bank energies as alternate but our findings indicate that for the speaker recognition task, the MFCC provides better accuracy.

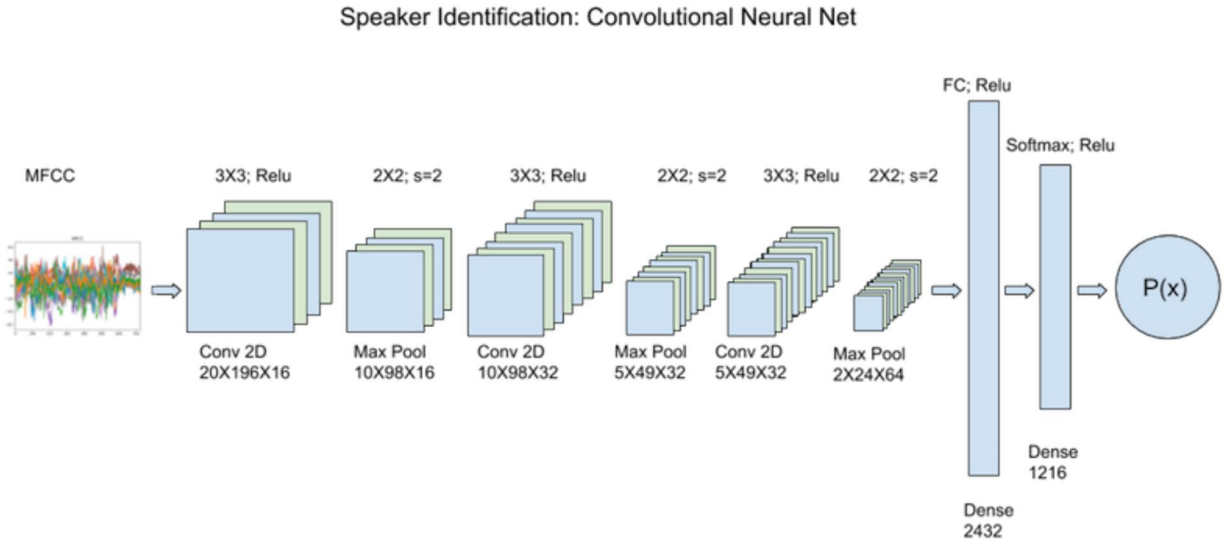


The pre-processing stage yields 4-dimensional matrix of input features (X) for M samples and a 2-dimensional one-hot encoded matrix of labels (Y). We split the data into train, dev and test sets with the ratio of **90%-5%-5%**. Finally, these 3 datasets are saved in h5 file format for efficient processing during training and test phase.

Speaker Identification CNN Model

We explored numerous public repo for a suitable model that we can leverage. The public repo we explored had one or more of the following limitations: incomplete or erroneous implementations; didn't align with our goals completely. We decided to implement from scratch a CNN on top of keras and tensorflow frameworks as described below.

The model employs multiple 2-D Convolutional layers accompanied by Max Pool and BatchNorm layers and finally followed by FC layers. The final FC layer has softmax activation. We use L2 regularization at the hidden FC layer to address the variance observed during initial training sessions. The dropout rate is 25%.



The model has **10,460,448** trainable parameters. The softmax activation of the output layer provides probability distribution over all known speakers:

$$\sigma : \mathbb{R}^K \to \left\{ \sigma \in \mathbb{R}^K | \sigma_i > 0, \sum_{i=1}^{K} \sigma_i = 1 \right\}$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

Training and Test

Optimal Hyperparameter search

The implementation begin with a simple model from scratch. A single convolutional layer followed by a FC with softmax activation. The model had high bias and also high variance. We experimented with increasing depth and layers to reduce bias. We implemented a hyperparameter tuner algorithm to search through a grid of hyperparameters. We compared tanh and relu for activation functions; adam, adadelta, rmsprop for optimizer. We tried a range for dropouts, L2 lambda and other hyperparameters. The hyperparameter search and training was done on **AWS GPU** to finalize the hyperparameters. In addition we experimented with varied depth of the network for optimal results.

Results

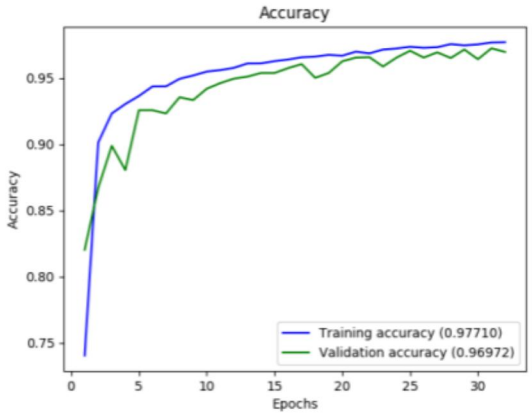
The model was trained with 58,854 samples for 32 epochs. The following table and charts summarize the results. Leveraging L2 regularization and dropout managed to keep variance very low. The depth and longer training ensured very low bias.

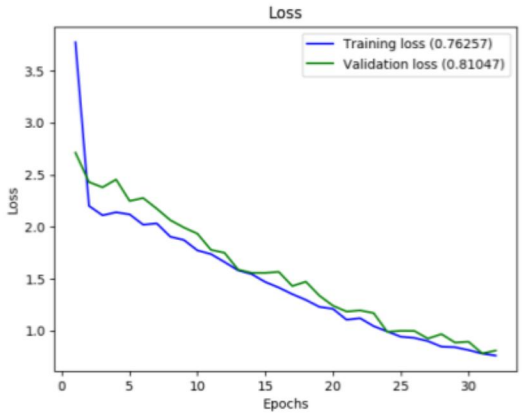
Model Hyperparameters

CNN Layers	*FC layers*	*Batch size*	*Optimizers*	*Activations*	*Dropout*
3	2	32	adam	ReLu + SoftMax	0.25

Model Performance

	Training	*Dev*	*Test*
Accuracy	97.71%	96.97%	96.33%
Loss	0.76	0.81	





Error Analysis

A close study of the samples which were incorrectly predicted by the model revealed a common pattern. Each incorrectly classified samples had high noise level or a very short burst of speech segment. The MFCC encoding were truncated to unify the sample sizes. We performed statistical analysis to find a reasonable truncation length. However, this could have also lead to some of the bais.

Conclusion/Future Work

In conclusion, we were able to achieve significant test set accuracy of over **96%** on a large dataset of thousands of different speakers. This result is encouraging and demonstrates that CNN is an effective choice for speaker recognition tasks.
The code and the trained model for this project is available at:
https://github.com/manishpandit/speaker-recognition.git

As for future work, we suggest to continue investigating the following:

1. **How does noise level affect the performance?** The voxforge dataset consists of audio samples with moderate amount of ambient noise. It will be instructive to explore noise reductions algorithms during the preprocessing stage and compare results.
2. **Are we overfitting to the audio source?** The voxforge dataset consists of audio samples recorded on computers. It will be interesting to test the model against audio recorded from various other sources.
3. **How can this algorithm be applied to real time streaming?** Currently the algorithm is trained and tested on existing audio clips; what changes would we have to make to use the algorithm in a real time scenario to identify speakers as they speak.

Contributions

- Manish: Github setup, Virtualenv setup, CNN Architecture and coding, Hyperparameter tuning, Documentation.
- Sophia: Analysis of various DNN models, coding of GMM for baseline, error analysis and testing, Documentation.
- Rish: Audio format research and pre-processing, AWS setup, Training executions, Documentation.

References and Research papers

1. Voxforge: http://www.voxforge.org/
2. GMM: https://en.wikipedia.org/wiki/Mixture_model
3. MFCC: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
4. Speaker Recognition: https://en.wikipedia.org/wiki/Speaker_recognition

There are several different sources of existing research done on speaker identification, below are some examples we referenced:

- *Baidu, Inc. Deep Speaker: an End-to-End Neural Speaker Embedding System*
  This research ResCNN and GRU architectures to process the audio data, and conducts training using triplet loss based on cosine similarity. This algorithm was tested on 3 different datasets and was presented to achieve around 95% accuracy for text-independent dataset.
- *X-VECTORS: ROBUST DNN EMBEDDINGS FOR SPEAKER RECOGNITION*
  This research uses x-vector model (maps variable length utterances to fixed-dimension features) with DNN, and uses data augmentation to supplement the dataset.
- *ROBUST TEXT-INDEPENDENT SPEAKER IDENTIFICATION USING SHORT TEST AND TRAINING SESSIONS*
  This research uses Gaussian Mixture Model (GMM) and investigated two approaches: Generalized Gaussian Density (GGD) and Sparse Representation Classification (SRC) method, particularly under noisy situations.
- *An Overview of Text-Independent Speaker Recognition: from Features to Supervectors*